SIMP Documentation

THE SIMP TEAM

Apr 20, 2023

CONTENTS

This is the documentation for the 6.6.0-2 release of SIMP.

SIMP is generally compatible with Red Hat Enterprise Linux (*RHEL*) and RHEL-derived distributions.

Microsoft Windows support is available via the commercial *Sicura* product.

This guide walks users through the process of installing and managing a *SIMP* system. It also provides a mapping of security features to security requirements, which can be used to document a system's security conformance.

CHAPTER

ONE

SPONSORS

We would like to thank the following for sponsoring the continued development of the *SIMP* project:

1.1 Sicura

Sicura is on a mission to automate security and compliance so that the organizations that power the world's critical technology systems can operate securely and efficiently.

https://www.sicura.us

CHAPTER

TWO

THE GUIDE

Warning: Be **EXTREMELY CAREFUL** when performing copy/paste operations from this document!

Different web browsers and operating systems may substitute incompatible quotes and/or line endings in your files.

The System Integrity Management Platform (*SIMP*) is an *Open Source* framework designed around the concept that individuals and organizations should not need to repeat the work of automating the basic components of their operating system infrastructure.

Expanding upon this philosophy, SIMP also aims to take care of routine policy compliance to include *NIST 800-53*, *FIPS* 140-2, the *DISA STIG*, and the *SCAP Security Guide*.

By using the *Puppet* automation stack, SIMP is working toward the concept of a self-healing infrastructure that, when used with a consistent configuration management process, will allow users to have confidence that their systems not only start in compliance but remain in compliance over time.

Finally, SIMP has a goal of remaining flexible enough to properly maintain your operational infrastructure. To this end, where possible, the SIMP components are written to allow all security-related capabilities to be easily adjusted to meet the needs of individual applications.

CHAPTER THREE

LEVEL OF KNOWLEDGE

SIMP targets users with a strong background in Linux systems management. The core technologies that require prerequisite knowledge are:

- *Puppet*
- *Domain Name System* (DNS) - *BIND*
- *Dynamic Host Configuration Protocol* (DHCP) - Internet Systems Consortium (ISC) DHCP
- *Lightweight Directory Access Protocol* (LDAP)
  - *389-DS* for *EL*8
  - *OpenLDAP* for *EL*7
- RedHat *Kickstart*, including all technologies involved: *Trivial File Transfer Protocol* (TFTP), *PXE*, PXELinux, etc.
- The Apache HTTP Server
- The *Yellowdog Updater, Modified* (YUM) package manager
- *Rsyslog* 8+
- *Firewalld* or *IPTables*
- *Auditd*, Basic knowledge of how the daemon works
- *Advanced Intrusion Detection Environment* (AIDE), basic knowledge of the rules
- Basic *X.509*-based *PKI* Key Management

SIMP handles as much of the initial setup and management of these tools as possible However, you will need at least some understanding of them in order to tailor a SIMP system to fit the desired environment. You will also need a general understanding of how to control and manipulate these tools from the *command line interface* (CLI); SIMP does not provide a *graphical user interface* (GUI).

Knowledge of scripting and *Ruby* programming will also help to further customize a SIMP install but is not required for routine use.

CHAPTER

FOUR

CONTENTS

4.1 Quick Start

4.1.1 What is SIMP?

The System Integrity Management Platform (*SIMP*) is an *Open Source* framework designed around the concept that individuals and organizations should not need to repeat the work of automating the basic components of their operating system infrastructure.

Expanding upon this philosophy, SIMP also aims to take care of routine policy compliance to include *NIST 800-53*, *FIPS* 140-2, the *DISA STIG*, and the *SCAP Security Guide*.

By using the *Puppet* automation stack, SIMP is working toward the concept of a self-healing infrastructure that, when used with a consistent configuration management process, will allow users to have confidence that their systems not only start in compliance but remain in compliance over time.

Finally, SIMP has a goal of remaining flexible enough to properly maintain your operational infrastructure. To this end, where possible, the SIMP components are written to allow all security-related capabilities to be easily adjusted to meet the needs of individual applications.

4.1.2 Diving Right In

The fastest way to get started with *SIMP* is to use one of the following two guides:

1. You need an ISO for bare metal or VM installation

    - *Installing SIMP from an ISO*

2. You have an existing system

    - *Installing SIMP From A Repository*

You should then follow the *SIMP User Guide* to start configuring the system.

4.2 Changelogs

This contains all SIMP changelogs for reference.

Important: Please read the intermediary changelogs if you are jumping versions during an upgrade!

4.2.1 SIMP Community Edition (CE) 6.6.0

Contents

OS compatibility

This release is known to work with:

- CentOS 7.0 2009 x86_64
- CentOS 8.5 2111 x86_64
- CentOS 8 Stream 20220423 x86_64
- OEL 7.9 x86_64
- OEL 8.5 x86_64
- RHEL 7.9 x86_64
- RHEL 8.5 x86_64

Full support for EL8

This release introduces **full** EL8 support for the SIMP Puppet server and agents across the entire SIMP framework.

EL8 support *was previously limited* to managing Puppet agents with the core SIMP Puppet modules.

EL6 support has been removed

EL6 is EOL and is no longer supported by SIMP CE.

All logic and testing in support of EL6 has been **completely removed** from the entire SIMP framework.

If you require further support for EL6 systems, consider purchasing commercial support.

Breaking Changes

- *ISOs Unpack into Unique Repository Paths*
- *Rsyslog < 8.24.0 is no Longer Supported*
- *SSSD < 1.16.0 is no Longer Supported*

ISOs Unpack into Unique Repository Paths

The directory structure of yum repositories unpacked from SIMP ISOs has changed.

Previously, all SIMP RPMs were placed into a single yum repository on the SIMP server, under `/var/www/yum/SIMP/` . This directory structure wasn't flexible enough to serve multiple operating systems/releases simultaneously without significant customization.

Starting from this release, repositories will be placed under the directory structure `/var/www/yum/SIMP/<os name>/<os version>/<arch>/`, which mirrors the layout of the base operating system repositories.

The **unpack_dvd** script has been updated to ensure that only compatible items are unpacked into the underlying repository. If the script detects incompatibilities, it will fail and provide guidance.

Rsyslog < 8.24.0 is no Longer Supported

Due to vendor recommendations, simp/rsyslog no longer supports **rsyslog** versions under 8.24.0

If you need to support older versions of **rsyslog**, please use simp/rsyslog 7.6.4 in an alternate *Puppet environment*.

SSSD < 1.16.0 is no Longer Supported

There are multiple issues in versions of **sssd** prior to 1.16.0. Users should upgrade to the latest release.

Significant Updates

SIMP Server Support on EL8

This release provides full support for managing SIMP Puppet servers on EL8.

Puppet 7 Support

All SIMP Puppet modules now work with both Puppet 6 and Puppet 7.

Puppet 5 Support Removed

Puppet 5 is EOL and support for it has been removed from all modules.

PuppetDB no Longer Configured by Default

A review of the newer **puppetserver** defaults as well as the concept of “only run what you require” led to the removal of **puppetdb** as a default installed/configured application.

This change should make it easier to run in resource-limited environments.

Existing systems will not be affected, but new systems will need to enable **puppetdb** per *HOWTO Enable PuppetDB*.

389 DS replaces OpenLDAP on EL8

On EL8, *389 Directory Server* replaces the (deprecated) *OpenLDAP* server as the default LDAP service.

Existing infrastructures will not be affected on upgrade, but new environments will need to configure correctly for their environment’s LDAP server.

LDAP Clients are still able to connect to either OpenLDAP server or 389 DS as necessary. *Please read the upgrade guide* if you are switching from OpenLDAP to 389 DS. New systems will require no additional configuration.

Switch from Cron to Systemd

With the deprecation of EL6, all supported OSes use systemd. The framework is now in a position to take advantage of systemd-specific features that improve system maintenance and administration.

Where possible, all SIMP puppet modules have been updated to replace old `cron` jobs with `systemd` timers. This enhances execution control and reporting for the scheduled jobs.

This practice may eventually enable systems to opt out of installing `cron` altogether, to the benefit of certain compliance profiles. It also has the benefit of being easier to manage.

Switch from Iptables to Firewalld

All SIMP modules now use `firewalld` by default instead of directly managing `iptables`. In general, the transition should be seamless for users unless advanced `iptables` rulesets were being managed (NAT, etc…).

Users still have the ability to directly manage `iptables` rules, but should be aware that there will be no further development on simp/iptables outside of maintaining the shims that hook it into `firewalld`.

Security Announcements

RPM Updates

Puppet RPMs

The following Puppet RPMs are packaged with the SIMP 6.6.0 ISOs:

Package	Version
puppet-agent	6.27.1-1 or 7.16.0-1
puppet-bolt	3.22.1-1
puppetdb	6.21.0-1 or 7.10.1-1
puppetdb-termini	6.21.0-1 or 7.10.1-1
puppetserver	6.19.0-1 or 7.7.0-1

Removed Puppet Modules

The following modules were removed from the release:

- simp_pki_service
- simp_bolt

Replaced Puppet Modules

Original	Replacement
aboe/chrony	puppet/chrony
camptocamp/kmod	puppet/kmod

Fixed Bugs

- *pupmod-simp-auditd*
- *pupmod-simp-aide*
- *pupmod-simp-cron*
- *pupmod-simp-fips*
- *pupmod-simp-gdm*
- *pupmod-simp-haveged*
- *pupmod-simp-incron*
- *pupmod-simp-libreswan*
- *pupmod-simp-libvirt*
- *pupmod-simp-network*
- *pupmod-simp-nfs*
- *pupmod-simp-ntpd*
- *pupmod-simp-openscap*
- *pupmod-simp-pam*
- *pupmod-simp-pupmod*
- *pupmod-simp-resolv*
- *pupmod-simp-rkhunter*
- *pupmod-simp-rsync*
- *pupmod-simp-rsyslog*
- *pupmod-simp-selinux*
- *pupmod-simp-simp*
- *pupmod-simp-simp_apache*
- *pupmod-simp-simp_gitlab*
- *pupmod-simp-simp_grub*
- *pupmod-simp-simp_nfs*
- *pupmod-simp-simp_openldap*
- *pupmod-simp-simplib*
- *pupmod-simp-ssh*
- *pupmod-simp-sssd*
- *pupmod-simp-stunnel*
- *pupmod-simp-svckill*
- *pupmod-simp-swap*
- *pupmod-simp-tlog*
- *pupmod-simp-tpm2*

- *pupmod-simp-vsftpd*
- *simp-doc*
- *simp-environment*
- *simp-gpgkeys*
- *simp-rsync*
- *simp-utils*
- *rubygem-simp-cli*

pupmod-simp-auditd

- Aligned the EL8 STIG settings
- Always add the `head` rules since they are required for proper functionality of the system
- Use `-F key=` instead of `-k` to match the STIG recommendations
- Switched the audit rules to `always,exit` instead of `exit,always` to match the man pages

pupmod-simp-aide

- Changed to using `--check` instead of `-C` by default to match the expectation of most security scanners
- Randomized the scheduling `minute` field so that I/O load is reduced on hosting platforms

pupmod-simp-cron

- Manage the **cron** packages by default

pupmod-simp-fips

- Use the **simplib__crypto_policy_state** fact instead of **crypto_policy__state**
- Ensure that **dracut_rebuild** is called when the `fips` kernel parameter is changed

pupmod-simp-gdm

- Fixed minor errors in the `compliance_markup` data
- Properly handle integration of **systemd-logind** with the `hidepid` flag on `/proc`
- Added a `pam_access` entry for the **gdm** user so that the greeter session can start

pupmod-simp-haveged

- Mask the **haveged** service when disabling it so that it is not restarted on reboot
- Ensure that **haveged** does not start if **rngd** is running

pupmod-simp-incron

- No longer pin the version of **incron** since the upstream versions have been fixed

pupmod-simp-libreswan

- Removed obsolete configuration items that prevented functionality on EL8:
  - `libreswan::ikeport`
  - `libreswan::nat_ikeport`
  - `libreswan::klipsdebug`
  - `libreswan::perpeerlog`
  - `libreswan::perpeerlogdir`

pupmod-simp-libvirt

- Removed ipxe-roms from the OEL package lists since they are now optional

pupmod-simp-network

- Ensure that the `network::eth` defined type honors the `network::auto_restart` parameter

pupmod-simp-nfs

- Added `_netdev` to the default mount options
- Ensure that `remote-fs.target` is enabled

pupmod-simp-ntpd

- Fixed a bug where `ntp::allow::rules` was not being honored
- Added `simp_options::ntp::servers` to the default lookup list for `ntpd::servers`

pupmod-simp-openscap

- Fixed the default data stream name in EL7

pupmod-simp-pam

- Silenced unnecessary TTY messages
- Added default Hiera deep merges for `pam::access::users` and `pam::limits::rules`
- Fixed a bug in `system-auth` where **`pam_tty_audit`** was not skipped if the login did not have a TTY. This prevented the GDM service login from succeeding.
- Set **`quiet`** on **`pam_listfile`** so that warnings do not get logged that look like authentication failures

pupmod-simp-pupmod

- Changed all instances of setting items in the `master` section to use `server` instead
- Updated `pupmod::conf` to automcatically switch `master` to `server`
- Automatically remove items from the puppet config in the `master` section that are set in the `server` section
- Added `pupmod::master::sysconfig::use_code_cache_flushing` to reduce excessive memory usage
- Removed SHA1 ciphers from the server cipher list
- Disconnected the puppetserver from the system FIPS libraries since it causes conflicts with the vendor provided settings
- Allow `pupmod::puppet_server` to accept Arrays
- Properly configure the server list when multiple puppet servers are specified
- Converted all **`cron`** settings to **`systemd`** timers
- Converted the ‘cleanup’ jobs to **`systemd.tmpfile`** jobs
- Fixed a bug where the `pupmod::master::sysconfig` class was not being applied
- Get **`certname`** from trusted facts only for authenticated remote requests
- Fix bolt compatibility

pupmod-simp-resolv

- Fixed bugs in the Augeas template
- Use configuration files to manage the global **`NetworkManager`** configuration

pupmod-simp-rkhunter

- Changed the minute parameter on scheduled tasks to a random number to reduce I/O load on hosting platforms
- Updated to use **systemd** timers instead of **cron** by default
- Added default user_fileprop_files_dirs to covert he puppet applications
- Ensure that the initial **propupd** command runs after the puppet run is complete
- Added a rkhunter::propupd class to ensure that the first cut of properties is updated after all packages have competed in the puppet run

pupmod-simp-rsync

- Fixed the documentation
- Noted that **sebool_use_nfs** and **sebool_cifs** will be deprecated in the future

pupmod-simp-rsyslog

- Fixed a bug where the **rsyslog** service would start without errors but fail to log when rsyslog::config::default_template was set to traditional

pupmod-simp-selinux

- Fixed a dependency cycle when using vox_selinux::boolean
- Fixed a bug where the module would attempt to create selinux_login resources when selinux::login_resources was set but **selinux** was disabled

pupmod-simp-simp

- Updated simp::yum::repo::local_os_updates to use the gpg keys installed into <yum directory>/SIMP/GPGKEYS to work around changes in EL8
- Corrected the HeapDumpOnOutOfMemoryError setting for **puppetdb**
- Ensure that **nsswitch** SSSD options for sudoers do not stop on files
- Do not include the auditors **sudo** user specification if the aliases have not been included
- Added the following to the sudoers defaults:
  - !visiblepw
  - always_set_home
  - match_group_by_gid
  - always_query_group_plugin
- Now use relative paths for the location for the SIMP GPG keys on YUM servers by default
- Support all valid values for simp::pam_limits::max_logins::value
- Added additional parameters to simp::admin to allow for more fine-grained control of global admin and auditor **sudo** rules

pupmod-simp-simp_apache

- Ensure that all `file` resources that manage more than permissions have an `ensure` attribute
- Moved the `magic` file into an EPP template to work better with **`bolt`**
- Use **`systemd`** to reload/restart the **`httpd`** service

pupmod-simp-simp_gitlab

- Fixed a bug where the **`change_gitlab_root_password`** script did not work with GitLab after 13.6.0

pupmod-simp-simp_grub

- Updated the documentation to better reflect GRUB2

pupmod-simp-simp_nfs

- Fixed a bug in **`create_home_directories.rb`** where EL8 systems could not talk to EL7 LDAP servers when the servers were in FIPS mode

pupmod-simp-simp_openldap

- Fixed `pki::copy` since the **`ldap`** group is no longer created by the OpenLDAP client packages
- Fixed `Float` to `String` comparison error in `simp_openldap::server::conf::tls_protocol_min`
- Deprecated parameters only applicable to EL6:
  - `simp_openldap::client::strip_128_bit_ciphers`
  - `simp_openldap::client::nss_pam_ldapd_ensure`

pupmod-simp-simplib

- Fixed the call to *klist* to properly handle cache issues
- Increased randomization in `simplib::gen_random_password`
- `simplib::cron::hour_entry` now supports comma separated lists
- `simplib::cron::minute_entry` now supports comma separated lists
- Fixed the **`simplib__networkmanager`** fact
- Fixed a bug where the **`ipa`** fact did not detect when an EL8 client was joined to an IPA domain
- Ensure that the **`puppet_settings`** fact supports both the `server` and `master` sections for backwards compatibility
- Added a tertiary check to the **`grub_version`** fact

pupmod-simp-ssh

- Fixed a bug where some changes to the **sshd** configuration did not cause a service restart
- Fixed a bug that caused a compilation error when `ssh::conf::ensure_sshd_packages` was set to `true`
- Ensure that `vox_selinux` is included prior to calling `selinux_port`
- Ensure that parameters that do not apply to EL8+ systems are not set on the target system
- No longer set `HostKeyAlgorithms` on the client configuration by default

pupmod-simp-sssd

- Added an option to `sssd::install` to prevent installation of the **sssd** client to increase compatibility with other operating systems
- Fixed multiple compatibility issues with non-OpenLDAP LDAP servers
- No longer use `concat` but instead drop configuration items into the `/etc/sssd/conf.d` directory
- Ensure that systems bound to FreeIPA, but not connected, do not cause compilation issues

pupmod-simp-stunnel

- Worked around a bug in EL7 where a connection denied by **tcpwrappers** would cause **stunnel** to hang and spike to 100% CPU usage indefinitely. All connections are still blocked by the firewall but now are always allowed in **tcpwrappers**.

pupmod-simp-svckill

- Added **rngd** to the default list of services to never be killed
- Removed obsolete documentation

pupmod-simp-swap

- Disable `dynamic_swappiness` by default
- Set static system swappiness to 60 by default

pupmod-simp-tlog

- Add a `file` resource if the file writer is specified
- Corrected the login in `tlog.sh.epp` in the case where a user does not have a login shell

pupmod-simp-tpm2

- Overrode the `systemd` unit file for `tpm2-abrmd` for TCTI compatibility

pupmod-simp-vsftpd

- Fixed `sysctl` updates on service restart

simp-doc

- Added HOWTO for managing PuppetDB
- Added HOWTO for enabling client reports
- Corrected SSL recovery documentation
- Corrected documentation relating to using `sudo` in STIG mode
- Added documentation for using EYAML in SIMP environments

simp-environment

- Add the EYAML hierarchy to the default `hiera.yaml`

simp-gpgkeys

- Fixed the target location for copying the GPG keys into the YUM repository
- Removed EL6 keys
- Updated the Red Hat release key

simp-rsync

- Removed dynamic BIND files from the list of files to `rsync`

simp-utils

- Fixed the `puppetlast` script and enabled it to read from filesystem reports
  - You will need to follow the instructions in *HOWTO Enable Client Reporting*

rubygem-simp-cli

- Changed set/get from **`master`** to **`server`** when updating the puppet configuration
- Use the status endpoint instead of a CRL query to validate the puppetserver status
- Use puppet to set the GRUB password
- Ensure that updating entries in `/etc/hosts` is idempotent
- Removed the **`LOCAL`** domain from the default **`sssd`** configuration
- No longer use the deprecated `simp_options::ntpd::servers` setting
- Simplified the instructions for the 'local user lockout' warning

New Features

The following items are common to most module updates and do not warrant specific inclusion below. For full details, see the `CHANGELOG` of all delivered packages.

- Removal of old Puppet version support
- Removal of EL6 support
- Addition of EL8 support
- Puppet module dependency updates

pupmod-simp-ds389

- New module for managing 389 DS

pupmod-simp-simp_firewalld

- Added the simp/simp_firewalld module and set it to the default on EL8+

pupmod-simp-gnome

- Removed support for GNOME2 since EL6 is no longer supported
- Also removed all gconf parameters and settings since they no longer have any use

pupmod-simp-logrotate

- Allow all log size configuration parameters to be specified in bytes, kilobytes, megabytes, or gigabytes

pupmod-simp-pam

- Added *dictcheck* and *faillock_log_file* parameter support
- Added Amazon Linux 2 support
- Added a **pre** section for setting auth file content to work with third party plugins
- Added the ability to set extra content in the **su** configuration

pupmod-simp-resolv

- Added the ability to precisely update the `resolv.conf` contents
- Added the ability to specify the entire contents of `resolv.conf`
- Added the ability to remove `resolv.conf` completely

pupmod-simp-rsyslog

Please read the module documentation and `CHANGELOG` since there were numerous changes!

- Dropped support for **`rsyslog`** < 8.24.0
- Added the ability to set the default template used for forwarding via `rsyslog::config::default_forward_template`
- Added parameters to allow additional configuration of the modules and main queue
- Added `Direct` and `Disk` to the allowed main message queue types
- Removed parameters only relevant to **`rsyslog`** < 8.6.0
  - `rsyslog::config::host_list`
  - `rsyslog::config::domain_list`
- Replaced obsolete parameters with modern replacements:
  - `rsyslog::config::action_send_stream_driver_mode` => `rsyslog::config::imtcp_stream_driver_mode`
  - `rsyslog::config::action_send_stream_driver_auth_mode` => `rsyslog::config::imtcp_stream_driver_aut`
  - `rsyslog::config::disable_remote_dns` => `rsyslog::config::net_enable_dns`
  - `rsyslog::config::suppress_noauth_warn` => `rsyslog::config::net_permit_acl_warning`
- Deprecated `rsyslog::config::default_template` for `rsyslog::config::default_file_template`
- Updated various parts of the configuration from legacy to RainerScript format

pupmod-simp-simp

- Added EL8 support
- Added `simp::puppetdb::disable_update_checking` to disable default analytics in accordance with NIST guidance
- **`puppetdb`** now sets `UseCodeCacheFlushing` by default
- The **`sssd`** client configuration now sets the LDAP schema based on the `simp::sssd:; client::ldap_server_type`
- The `simp::sssd::client` no longer creates a `LOCAL` provider

pupmod-simp-simp_ds389

- New module providing SIMP-specific settings for 389 DS for providing a suitable replacement for OpenLDAP

pupmod-simp-simp_gitlab

- Now default simp_gitlab::allow_fips to true which works with GitLab 14.0.0+

pupmod-simp-simp_nfs

- Provide host PKI information to upstream LDAP servers

pupmod-simp-simp_options

- Added simp_options::ntp for more generalized configuration of both **ntpd** and **chronyd**

pupmod-simp-simpkv

- Added an LDAP backend plugin

pupmod-simp-simplib

- Added simplib::cron::to_systemd() to convert **cron** resource parameters to **systemd** timespec format
- Added simplib::cron::expand_range() to expand ranges into comma separated strings
- Added simplib::params2hash() to return all of the calling scope's parameters as a Hash
- Added **net.ipv6.conf.all.disable_ipv6** to the **simplib_sysctl** fact
- Added a **simplib__cryhpto_policy_state** fact

pupmod-simp-ssh

- Added an option to turn off managing the AuthorizedKeysFile parameter in /etc/ssh/sshd_config

pupmod-simp-sssd

- Made installing the **sssd** client optional (enabled by default)
- No longer support **sssd** < 1.16.0
- Users can now set sssd::custom_config to a string that will be placed into /etc/sssd/conf.d/zz_puppet_custom.conf
- Users can optionally purge the /etc/sssd/conf.d directory if they want puppet to be authoritative

pupmod-simp-sudo

- Added the ability for users to create `include` clauses in `/etc/sudoers`

pupmod-simp-tpm2

- Updated `tpm2::ownership` and the **`tpm2`** fact to support tpm2_tools version 4
- Added a provider for the **`tpm2_changeauth`** functionality to provide ownership update capabilities

simp-environment

- No longer configure **`puppetdb`** by default

simp-gpgkeys

- Added the EL8 GPG keys
- Added the new Puppet signing key

simp-utils

- Updated the **`unpack_dvd`** scripts to work with EL8 ISOs
- Added transition scripts for upgrading from 6.5.0 to 6.6.0

rubygem-simp-cli

- Removed management of **`puppetdb`** components since it is no longer enabled by default
- Removed support for EL6
- Use OpenLDAP by default on EL7 and 389 DS otherwise
- Set the defaults for both **`ntpd`** and **`chronyd`**

Known Bugs and Limitations

Below are bugs and limitations known to affect this release. If you discover additional problems, please submit an issue to let use know.

- **`sssd`** does not always start the **`ds389`** LDAP server immediately after kickstarting an EL8 system. An additional puppet run clears the problem. The error in the log is

  sssd.dataprovider.getDomains: Error [1432158215]: DP target is not configured

4.2.2 SIMP Community Edition (CE) 6.5.0

Contents

OS compatibility

This release is known to work with:

- CentOS 6.10 x86_64 — no server ISO or release tarball
- CentOS 7.0 2003 x86_64
- CentOS 8.2 2004 x86_64 — *client systems only*
- OEL 6.10 x86_64 — no server ISO or release tarball
- OEL 7.8 x86_64
- OEL 8.2 x86_64 — *client systems only*
- RHEL 6.10 x86_64 — no server ISO or release tarball
- RHEL 7.8 x86_64
- RHEL 8.2 x86_64 — *client systems only*

Important OS compatibility limitations

OS compatibility is subject to the following limitations:

EL8 support is CLIENT ONLY

This release introduces client-only EL8 support in the core Puppet modules.

- EL8 support is limited to managing EL8 Puppet *agents* with the core Puppet modules.
- All Puppet modules provided as core dependencies of the simp RPM support EL8.

This release does NOT support EL8 for:

- Managing an EL8 SIMP Server
- Installing SIMP from an EL8 ISO.
- Using the **`unpack_dvd`** script on modular yum repositories found on EL8 OS ISOs

Additional limitations with EL8

- Not all modules provided by the simp-extras RPM have been updated for EL8.
- EL8 updates to the remaining simp-extras modules will be phased in over future SIMP releases.
- Support for managing an EL8 SIMP/Puppet server and installing from EL8 ISOs will be provided in a later SIMP release (SIMP 6.6.0).
- In SIMP 6.5.0, *there are known issues* with PXE kickstarting and unpacking ISOs as yum mirrors for EL8 clients. These issues particularly affect network-isolated environments.
  - For details, see: *Special considerations with EL8 clients*.

EL6 support is EOL

- EL6 maintenance support is EOL for both RHEL 6 and CentOS 6.
  - Upstream vendor support ended on 30 November 2020.
- SIMP tarball and ISO releases for EL6 **have been discontinued.**
  - There will be **no EL6 tarball or server ISO release** for SIMP 6.5.0.
- New SIMP Puppet modules **may not support EL6.**
- EL6 *has* been tested with SIMP 6.5.0's Puppet modules, the `unpack_dvd` script, and PXE kickstarts.
  - However: **No further EL6 support** is planned for SIMP Puppet modules.
  - EL6 support **may be removed completely** in each module's next release.
- Some optional Puppet modules (provided by the simp-extras RPM) no longer support EL6.
  - In particular, this affects simp/autofs, simp/nfs, and simp/simp_nfs.
  - If you need these capabilities on EL6, use earlier versions of these modules in an EL6-specific Puppet environment.

Breaking Changes

IPTables Rule Refinement

Important: IPTables does NOT have breaking changes out of the box.

A new parameter, `iptables::precise_match` was added that performs higher precision matching on **`iptables`** rules to detect the need to restart **`iptables`**.

It is highly recommended that you set `iptables::precise_match: true` in *Hiera* so that minor changes, such as subnet updates or single port changes, will appropriately restart **`iptables`**.

If you enable precision matching, do so with care since you may find that **`iptables`** rule updates are propagated that you thought had previously been applied.

It is highly recommended that you migrate to `firewalld` if at all possible. See the relevant section below for more details.

Deprecated Puppet 3 API Functions Removed

All SIMP-provided Puppet 3 API functions (originally deprecated in SIMP 6.4.0) have now been removed in order to fully support Puppet 6. The affected functions and their replacements (when available) are listed in sub-sections below.

Puppet 3 Functions Removed from simp/compliance_markup

Puppet 3 API Function	Replacement	Replacement Source
`compliance_map`	`compliance_markup::compliance_map`	simp/compliance_markup >= 3.0.0

Puppet 3 Functions Removed from simp/simp_apache

Puppet 3 API Function	Replacement	Replacement Source
`apache_auth`	`simp_apache::auth`	simp/simp_apache >= 6.0.1
`apache_limits`	`simp_apache::limits`	simp/simp_apache >= 6.0.1
`munge_httpd_networks`	`simp_apache::munge_httpd_networks`	simp/simp_apache >= 6.0.1

Puppet 3 Functions Removed from simp/simplib

Important: Most (but not all) of the Puppet 3 API functions in the table below have replacements. If any function that has been removed without a replacement is essential to you, let us know by submitting a feature request at https://simp-project.atlassian.net.

Puppet 3 API Function	Replacement	Rep
`array_include`	Puppet language in operator *or* Puppet built-in functions `any` or `all`	Pup
`array_size`	Puppet built-in function `length`	Pup
`array_union`	Puppet language + (concatenation) operator, combined with Puppet built-in function `unique`	Pup
`bracketize`	`simplib::bracketize`	sim
`generate_reboot_msg`	None	N/
`get_ports`	None	N/
`h2n`	None	N/
`host_is_me`	`simplib::host_is_me`	sim
`inspect`	`simplib::inspect`	sim
`ipaddresses`	`simplib::ipaddresses`	sim
`ip_is_me`	`simplib::host_is_me` (checks hostnames and IP addresses)	sim
`ip_to_cron`	`simplib::ip_to_cron`	sim
`join_mount_opts`	`simplib::join_mount_opts`	sim
`localuser`	None	N/
`mapval`	None	N/
`nets2cidr`	`simplib::nets2cidr`	sim
`nets2ddq`	`simplib::nets2ddq`	sim
`parse_hosts`	`simplib::parse_hosts`	sim
`passgen`	`simplib::passgen`	sim
`rand_cron`	`simplib::rand_cron`	sim
`simp_version`	`simplib::simp_version`	sim
`simplib_deprecation`	`simplib::deprecation`	sim
`slice_array`	Puppet built-in `slice`	Pup
`strip_ports`	`simplib::strip_ports`	sim
`to_integer`	Puppet built-in `Integer` *or* `simplib::to_integer`	`Int`
`to_string`	Puppet built-in `String` *or* `simplib::to_string`	`Str`
`validate_array_member`	`simplib::validate_array_member`	sim
`validate_array_of_hashes`	Use a custom Puppet data type such as `Array[Hash]`	Pup
`validate_between`	Puppet data types `Integer` or `Float` *or* `simplib::validate_between`	sim
`validate_bool_simp`	Use Puppet `Boolean` data type *or* `simplib::validate_bool`	Pup
`validate_deep_hash`	`simplib::validate_deep_hash`	sim
`validate_float`	Use Puppet `Float` data type *or* a check using `is_float` from puppetlabs/stdlib	Pup
`validate_macaddress`	Use `Simplib::Macaddress` data type	sim

Table 1 – continued from previous page

Puppet 3 API Function	Replacement	Rep
`validate_net_list`	Use `Simplib::Netlist` data type *or* `simplib::validate_net_list`	sim
`validate_port`	Use `Simplib::Port` data type *or* `simplib::validate_net_list`	sim
`validate_re_array`	`simplib::validate_re_array`	sim
`validate_sysctl_value`	`simplib::validate_sysctl_value`	sim
`validate_umask`	Use `Simplib::Umask` data type	sim
`validate_uri_list`	`simplib::validate_sysctl_value`	sim

Puppet 3 Functions Removed from simp/ssh

Puppet 3 API Function	Replacement	Replacement Source
`ssh_autokey`	`ssh::autokey`	simp/ssh >= 6.2.0
`ssh_global_known_hosts`	`ssh::global_known_hosts`	simp/ssh >= 6.2.0

Primary API Changed in Optional Modules

The following SIMP modules from the simp-extras RPM have had breaking API changes:

- simp/autofs
- simp/nfs
- simp/simp_nfs
- simp/simp_snmpd

The specific changes made are described in detail in the *New Features section*.

EL6 Support Dropped from Some (Optional) Puppet Modules

The following optional SIMP modules have dropped support for EL6:

- simp/autofs
- simp/nfs
- simp/simp_nfs

If you need EL6 for a client node, place it in an environment with older versions of the appropriate modules.

Significant Updates

- *EL8 SIMP Client Node Support*
- *Full Puppet 6 Support and Puppet 6 Default Components*
- *firewalld Support*
- *Optional Dependency Handling*
- *Dependent Module Updates*

EL8 SIMP Client Node Support

This release provides support for managing software on EL8 agents.

This includes all (appropriate) Puppet modules provided by the simp RPM, and a subset of the Puppet modules provided by the simp-extras RPM.

- The remaining changes required for an EL8 SIMP server and ISO will be available in the next SIMP minor release.
- EL8 updates to the remaining, optional, Puppet modules will be phased in over future SIMP releases. This includes the following SIMP modules:
  - simp/gdm
  - simp/gnome
  - simp/hirs_provisioner
  - simp/mate
  - simp/simp_pki_service
  - simp/tuned
  - simp/vnc
  - simp/x2go

Full Puppet 6 Support and Puppet 6 Default Components

All SIMP Puppet modules now work with both Puppet 5 and Puppet 6, and the SIMP-6.5.0 ISOs deliver Puppet 6 application RPMs.

firewalld Support

As of SIMP 6.5.0, **firewalld** support is available within the SIMP and is the default for all new installations on platforms that support it.

- **New simp/simp_firewalld module**: SIMP now includes simp/simp_firewalld which provides a profile class and defined type to manage the system's **firewalld** with "safe" defaults and safety checks for **firewalld** rules.
- **firewalld support in simp/iptables for backward compatibility**: The simp/iptables module has preliminary support for acting as a pass-through to various **firewalld** capabilities using the simp/simp_firewalld module.
  - To enable 'firewalld' mode on supported operating systems, simply set `iptables::use_firewalld` to `true` via *Hiera*.
  - EL8 systems enable 'firewalld' mode by default.
  - Use of any of the `iptables::listen::*` defined types will work seamlessly in 'firewalld' mode, as long as IP addresses are used in their `trusted_net` parameters.
  - Direct calls to `iptables::rule` in 'firewalld' mode will emit a warning notification that directs the user to convert their rules to `simp_iptables::rule` types.

Important: Be aware that **firewalld** rules do not support hostnames; IP addresses must be used. This may impact any manifests that contain `iptables::listen` resources, including resources from some SIMP modules. You will have to change any hostnames to IP addresses for the affected resources when using **firewalld**.

The table below is a list of the SIMP resource parameters impacted by the lack of hostname support by **`firewalld`**.

- Many of these parameters default to `simp_options:trusted_nets`, when it is available.
- Each network element can be specified as a network (CIDR notation), an IP address, `'ALL'` or `'any'`.
- ‘or’ in the table below indicates the default value that will be used if the previous value is not defined.

Parameter	Default Value
freeradius::v3::conf::trusted_nets	simp_options::trusted_nets or ['127.0.0.1', '::1']
krb5::kdc::firewall::trusted_nets	simp_options::trusted_nets or ['127.0.0.1','::1']
krb5::kdc::realm::trusted_nets	krb5::kdc::trusted_nets or simp_options::trusted_nets or ['127.0.0.1']
libreswan::trusted_nets	simp_options::trusted_nets or ['127.0.0.1/32']
nfs::client::mount::nfs_server	N/A
nfs::server::trusted_nets	simp_options::trusted_nets or ['127.0.0.1']
ntpd::trusted_nets	simp_options::trusted_nets or ['127.0.0.1','::1']
postfix::server::trusted_nets	simp_options::trusted_nets or ['127.0.0.1']
pupmod::master::trusted_nets	simp_options::trusted_nets or ['127.0.0.1','::1']
rsync::server::trusted_nets	simp_options::trusted_nets or ['127.0.0.1']
rsyslog::trusted_nets	simp_options::trusted_nets or ['127.0.0.1/32']
simp::puppetdb::trusted_nets	simp_options::trusted_nets or ['127.0.0.1']
simp_apache::ssl::trusted_nets	simp_options::trusted_nets or ['127.0.0.1','::1']
simp_apache::conf::allowroot	['127.0.0.1','::1']
simp_nfs::home_dir_server	N/A
simp_nfs::mount::home::nfs_server	N/A
simp_openldap::server::conf::trusted_nets	simp_options::trusted_nets or ['127.0.0.1']
ssh::server::conf::trusted_nets	['ALL']
stunnel::connection::trusted_nets	simp_options::trusted_nets or ['127.0.0.1']
stunnel::instance::trusted_nets	simp_options::trusted_nets

Optional Dependency Handling

In SIMP 6.5.0, optional dependency handling has been integrated into ~20 additional SIMP Puppet modules. These modules explicitly identify optional, dependent modules, all while providing safeguards to ensure the user is notified of any such missing dependencies at compilation time. This feature allows the user to minimize installation of unused modules in an environment, when the user is not using SIMP to manage specific capabilities.

Key details about this feature are as follows:

- Optional module dependencies are indicated in the `metadata.json` file using an 'optional_dependencies' key within a 'simp' key. For example, simp/rsyslog's metadata.json.
- The user has complete control over installation of the optional dependency modules. These dependencies will not be installed automatically when the module using them is installed via `puppet module install`.
- Modules that use this feature will fail manifest compilation, if the user enables the optional capabilities, but the optional dependencies required to implement that capability are not installed in the environment.

Dependent Module Updates

SIMP updated as many dependent modules as possible. This included major version bumps for several of the dependent modules. These changes did not have a significant impact on the SIMP infrastructure. The dependency version bumps did, however, require some of the SIMP modules to update their respective `metadata.json` files. These metadata changes, in turn, required SIMP module version updates.

Security Announcements

SIMP 6.5.0 Added mitigations for the following CVEs:

- CVE-2020-7942
- CVE-2019-14287
- CVE-2019-6477

RPM Updates

Puppet RPMs

The following Puppet RPMs are packaged with the SIMP 6.5.0 ISOs:

Package	Version
puppet-agent	6.18.0-1
puppet-bolt	2.29.0-1
puppetdb	6.12.0-1
puppetdb-termini	6.12.0-1
puppetserver	6.13.0-1

> **Warning:** You do **NOT** need to update your version of Puppet from 5.X to use the modules supplied with this version of SIMP.
>
> If you decide to update from 5.X, back up your server and test the upgrade carefully.

Removed Puppet Modules

Unused Augeasproviders Modules

The following packages for unused Augeasproviders Puppet modules and one dependency were removed from the SIMP ISOs:

- pupmod-herculesteam-augeasproviders_apache
- pupmod-herculesteam-augeasproviders_mounttab
- pupmod-herculesteam-augeasproviders_nagios
- pupmod-herculesteam-augeasproviders_pam
- pupmod-herculesteam-augeasproviders_postgresql
- pupmod-herculesteam-augeasproviders_puppet
- pupmod-herculesteam-augeasproviders_shellvar
- pupmod-puppetlabs-mount_providers

Docker Modules

The packages for the following Docker Puppet modules have been permanently removed from the SIMP ISOs, because SIMP is moving towards **podman** support over **docker**.

- pupmod-puppetlabs-docker
- pupmod-simp-simp_docker

pupmod-simp-journald

The pupmod-simp-journald package has been removed from SIMP ISOs, because the functionality the simp/journald module provided is now provided by the camptocamp/systemd module. If you used simp/journald, you will need to update your manifests to use camptocamp/systemd.

Fixed Bugs

- *pupmod-simp-freeradius*
- *pupmod-simp-iptables*
- *pupmod-simp-libvirt*
- *pupmod-simp-logrotate*
- *pupmod-simp-network*
- *pupmod-simp-nfs*
- *pupmod-simp-pam*
- *pupmod-simp-polkit*
- *pupmod-simp-pupmod*
- *pupmod-simp-rsyslog*
- *pupmod-simp-selinux*
- *pupmod-simp-simp*
- *pupmod-simp-simplib*
- *pupmod-simp-simp_options*
- *pupmod-simp-simp_snmpd*
- *pupmod-simp-stunnel*
- *pupmod-simp-tftpboot*
- *pupmod-simp-tlog*
- *pupmod-simp-tpm2*
- *pupmod-simp-xinetd*
- *rubygem-simp-cli*
- *simp-environment-skeleton*
- *simp-utils*

pupmod-simp-aide

- Fixed a bug in Compliance Engine data.

pupmod-simp-auditd

- Fixed a bug in which the module could not enable auditing on a system with auditing already disabled in the kernel, when replication of the audit logs to syslog was required.
- Fixed a bug in which the **`auditd`** service was managed when the kernel was not enforcing auditing.
- Fixed a bug in which the facts were not properly confined.
- Fixed a bug in which `/etc/audit/audit.rules.prev` caused unnecessary flapping.
- Fixed regex substitution for bad path characters.
- Added missing herculesteam/augeasproviders_grub module dependency.

pupmod-simp-dconf

- Fixed a bug in `ensure = absent` in `dconf::settings`.

pupmod-simp-compliance_markup

- Fixed merging bugs introduced in interim versions of the module.
- Fixed a regression introduced in interim versions of the module in which compliance reports were missing 'controls', 'identifiers', and 'oval-ids'.

pupmod-simp-freeradius

- Fixed missing 'group_filter' option in LDAP.

pupmod-simp-iptables

- Fixed a bug in which the **`iptables`** services and rules were not managed when `iptables::use_firewalld` was set to `true` on an EL6 system.
- Fixed an ordering issue with setting `xt_recent` parameters that could occur on OEL7 nodes. However, there are other issues with `xt_recent` on OEL that may prevent this module from working on OEL in some circumstances.
- Fixed a bug in which the module did not check for **`firewalld`** availability when `iptables::use_firewalld` was set to `true`.
  - The module now ensures that systems that do not have `firewalld` do not attempt to configure it.
- Fixed bugs in **`iptables`** rule address normalization:
  - Ensured that all addresses are normalized when rules are processed.
  - Removed nested looped rule normalization of addresses since it is no longer required.
  - Fixed `normalize_addresses()` so that it simply grabs the netmask if present and slaps on the appropriate one if not.
- Fixed some bugs in the `munge()` portions of the native types.

pupmod-simp-libvirt

- Fixed issues with module data.

pupmod-simp-logrotate

- Fixed a bug in which the 'size' parameter in the global **`logrotate`** configuration file was specified more than once.

pupmod-simp-network

- Fix a bug where both the legacy network and **NetworkManager** were activated in all cases.

pupmod-simp-nfs

- Fixed a bug in which IPv6 '::1' network entries were not being created in `/etc/exports`. This could cause connections over **stunnel** to fail under certain conditions.
- **rpc.rquotad** service configuration was erroneously written to `/etc/sysconfig/nfs` for EL7. It is now written to the correct file, `/etc/sysconfig/rpc-rquotad`.
- Fixed **idmapd**-related bugs:
  - **idmapd** was erroneously only enabled when NFSv3 was allowed. **idmapd** is an NFSv4 service.
  - The **idmapd** client was not configured to use **nfsidmap**. An **nfsidmap** entry has now been added to `/etc/request-key.conf`.
- Fixed bugs in which bidirectional communication for NFSv3 was not properly configured.
  - NFSv3 lockd ports on the NFS client were not explicitly configured and thus not allowed through the firewall. This would have affected file locking using NLM.
  - **rpcbind**, **statd**, and **lockd** service names were not allowed by TCP Wrappers for the NFS client. This would have affected server to client NFSv3 NSM and NLM protocol messages over TCP.
- Fixed bugs in mount options
  - Previously used the deprecated 'nfs4' fstype. This has been replaced with the 'nfs' fstype and use of the 'nfsvers' option to specify the version of NFS to use.
  - The mount option 'proto' is now set to 'tcp' when `stunnel` is enabled.
- Fixed a bug with a duplicate exec resource in `nfs::client::mount` when **stunnel** was enabled.
- Fixed erroneous server-only/client-only configuration that appeared to be able to be set independently for the NFS client and NFS server on the same node, but because of shared services, actually applied to the node as a whole.
  - Removed `nfs::client::firewall` and `nfs::server::firewall`. Use `nfs::firewall` instead.
  - Removed `nfs::server::tcpwrappers`. Use `nfs::tcpwrappers` instead.
  - Removed `nfs::server::nfsv3`, `nfs::server::lockd_arg`, `nfs::server::statdarg`, `nfs::server::statd_ha_callout`, `nfs::server::rpcgssdargs`, and `nfs::server::rpcsvcgssdargs`. Use appropriate parameters in the `nfs` class instead.

pupmod-simp-pam

- Fixed a bug in which a local user password could not be set.
  - Moved the 'pam_unix.so' check before the 'pam_sss.so' check in the password section of the auth files otherwise it returns an `authentication token manipulation` error and local passwords cannot be changed.

pupmod-simp-polkit

- Fixed issue with `basic_policy` template that resulted in malformed rules.

pupmod-simp-pupmod

- Fixed a bug in which the module could not determine the appropriate Puppet configuration for Puppet >= 6.19.0 from the internal `Puppet.settings` method, because the 'master' section was renamed to 'server'.
- Fixed a bug on EL6 nodes in which setting `pupmod::master::generate_types` to `false` caused the catalog compilation to fail.
- Fixed a bug in **`puppetserver`** configuration in which the 'profiler-output-file' parameter was incorrectly specified as 'profiling-output-file'.
- Fixed a bug in managing group ownership of `puppet.conf` when using Puppet Enterprise.
  - Ensured that `pupmod::pass_two` does not conflict with the internal *PE* configuration code for group ownership of `puppet.conf`.

pupmod-simp-rsyslog

- Fixed the default security collection string for **`firewalld`** rules.
- Fixed a bug where the 'IncludeConfig' directive for `/etc/rsyslog.d` allowed more than just `.conf` files to be parsed.

pupmod-simp-selinux

- Fixed a bug in which the module would attempt to create `selinux_login` resources when `selinux::login_resources` was set but selinux was disabled. This resulted in an error message *Could not find a suitable provider for selinux_login* during catalog compilation.

pupmod-simp-simp

- Ensure that the **`sudoers`** rule for removing the Puppet SSL directory is not created when running from Bolt, since the directory target is changed at each Bolt run and will result in non-idempotency.
- Fixed a bug in which the 'gpgkey' and 'baseurl' configuration strings were required for the local YUM repositories managed by `simp::yum::repo::local_os_updates` and `simp::yum::repo::local_simp`.
  - Both are optional in the `yumrepo` type if they already exist on disk.
- Removed the broken `tasks/` directory.

pupmod-simp-simplib

- Fixed the `simplib::puppet::metadata::os_support` data type to allow `operatingsystemrelease` to be optionally defined.
- Added Amazon Linux support
- Fixed the use of `simplib::debug::inspect` when using Bolt.
- Fixed bugs in the `grub_version` and `init_systems` facts.
- Fixed the `simplib__auditd` fact so that it detects the state of the running **`auditd`** process.
- Fixed `Simplib::Systemd::ServiceName` to accept instance services.
- Fixed an issue in the `simplib__sshd_config` fact that would cause the daemon to start on an EL6 system that did not already have it running.
- Fixed a bug in which `simplib__firewalls` fact was not properly confined and would trigger on Windows+ systems.
- Fixed an issue in `simplib::ip::family_hash` where the 'unknown' entries were not properly populated.
- Fixed bug in which `simplib::simp_version` did not work on Windows.
- Fixed `uninitialized constant` error with the `reboot_notify` custom type.

pupmod-simp-simp_options

- Fixed *PE* detection in `simp_options::puppet::server_distribution`.

pupmod-simp-simp_snmpd

- Fixed a bug in which the PID file option was missing from the default options for the **`snmpd`** daemon in EL6. The daemon failed to start without this option.
- Fixed a bug where the default for client security level was incorrectly set.
  - The default access security level is now by the new parameter `simp_snmpd::defvacmlevel` instead of `simp_snmpd::defsecuritylevel`.
  - `simp_snmpd::defsecuritylevel` sets the default security level for the client.
- Added a missing dependency on simp/tcpwrappers.

pupmod-simp-stunnel

- Added the `stunnel::instance_purge` class to remedy the 'floating services' issue.

pupmod-simp-tftpboot

- Fixed a bug in which the internal **rsync** operation did not match the documentation.
- Fixed a manifest ordering issue.

pupmod-simp-tlog

- Fixed a bug in the **tcsh** template.
- Added a workaround to scripts in `/etc/profile` to handle a bug in **tlog** that would prevent logins if the system hostname could not be found.

pupmod-simp-tpm2

- Fixed a bug where the **tpm2_*** commands could return nothing which would trigger an error in further logic.

pupmod-simp-xinetd

- Removed 'TRAFFIC' from the default `log_on_success` list since it may cause information leakage and is not supported by all service types.

rubygem-simp-cli

- Fixed a bug in which **simp config** did not allow DNS domains that did not include at least one dot character. Domains are now validated per RFC 3696.
- Fixed a bug where **simp config** recommended the wrong SSSD domain, when the SIMP server was not the LDAP server. It recommended the 'Local' domain, when the appropriate SIMP-created domain with the 'local' (EL6) or 'files' (EL7) provider is named 'LOCAL'.
- Fixed a bug in **simp environment new** in which the actual failure messages from a failed **setfacl --restore** execution were not logged.
- Fixed a bug where **simp config --dry-run** would prompt the user to apply actions instead of skipping them and then writing the `~/.simp/simp_conf.yaml` file.
  - Users would answer 'no' to the unexpected apply query and then **simp config** would only persist the answers to the interim answers file (`~/.simp/.simp_conf.yaml`).
- Fixed Puppet Enterprise support for **simp config** and **simp bootstrap**.
  - Fixed a fact-loading bug that prevented the *PE* fact (`is_pe`) from being available.
  - Hardened PE-detection logic for cases in which the `is_pe` fact is not yet available during **simp config**.
  - Added support for SIMP server template Hiera data that is PE-specific.
  - Fixed a bug in which the module paths containing PE modules were not excluded when **simp config** checked for modules in the 'production' Puppet environment. This forced the user to remove the skeleton 'production' environment installed by the puppet-agent RPM, in order to get **simp config** to run on a freshly installed PE system.

simp-environment-skeleton

- When running FakeCA certification-generation scripts in batch mode, do not request input from the user.
- Fixed a bug in which some non-script files were installed with executable permissions.

simp-utils

- Fixed minor bugs in **`unpack_dvd`**.

New Features

pupmod-simp-aide

- Updated the EL8 ciphers to be safe on FIPS systems by default.
- Removed overrides for `aide::aliases` on EL8 since it works properly in FIPS mode.
- Automatically add '@@include' lines to `aide.conf`. Previously, when declaring `aide::rule` resources, it was also necessary to add the rule name to the `aide::rules` array.
- Moved the default rules to data in modules.

pupmod-simp-auditd

- Allow `auditd::space_left` and `auditd::admin_space_left` to accept percentages on supported versions.
- Added 'INCREMENTAL_ASYNC' to possible values for `auditd::flush`.
- Added a `built_in` audit profile to the subsystem that provides ability to include and manage sample rulesets to be compiled into active rules.
- Ensured that **kmod** is audited in all STIG modes on EL7+.
- Allow users to knockout entries from arrays specified in Hiera.
- Added rules based on best practices mostly pulled from `/usr/share/doc/auditd`:
  - Audit 32 bit operations on 64 bit systems
  - Audit calls to the **auditd** CLI commands
  - Audit IPv4 and IPv6 inbound connections
  - Optionally audit IPv4 and IPv6 outbound connections
  - Audit suspicious applications
  - Audit systemd
  - Audit the **auditd** configuration space
  - Ignore time daemon logs (clutter)
  - Ignore 'CRYPTO_KEY_USER' logs (clutter)
  - Add ability to set the 'backlog_wait_time'
  - Set 'loginuid_immutable'
- Set defaults for syslog parameters if **auditd** version is unknown.
- Added a fact that determines the major version of **auditd** that is running on the system, `auditd_major_version`. This is used in the `hiera.yaml` hierarchy to add module data specific to the versions.
- Added support for **auditd** v3.0 which is used by RedHat 8. Most of the changes in **auditd** v3.0 were related to how the plugins are handled but there are a few new parameters added to `auditd.conf`. They are set to their defaults according to **man** page of `auditd.conf`.
  - **auditd** V3.0 moved the handling of plugins into **auditd** from **audispd**. The following changes were made to accommodate that:
    - To make sure the parameters used to handle plugins where defined in one place no matter what version of **auditd** was used, they were moved to `init.pp` and referenced from there by the `audisp` manifest. For backwards compatibility, they remain in `audisp.conf` and are aliased in the Hiera module data.

    - For backwards compatibility `auditd::syslog` remains defaulting to the value of `simp_options::syslog` although the two are not really the same thing. You might want to review this setting and set `auditd::syslog` to a value that is appropriate for your system.

      - To enable **auditd** logging to syslog, set the following in Hiera

        ```
        ---
        auditd::syslog: true
        auditd::config::audisp::syslog::enable: true.
        # The drop_audit_logs is still there for backwards compatibility and
        # needs to be disabled.
        auditd::config::audisp::syslog::drop_audit_logs: false
        ```

      - To stop **auditd** logging to syslog set the following in Hiera

        ```
        ---
        auditd::syslog: true
        auditd::config::plugins::syslog::enable: false.
        ```

      - Setting `auditd::syslog` to `false` will stop Puppet from managing the `syslog.conf`, it will not disable **auditd** logging to syslog. Disable the syslog plugin as described above.

    - The settings for `syslog.conf` were updated to work for new and old versions of **auditd**.

    - Added installation of audisp-syslog package when using **auditd** V3.

- Added rules to monitor `/usr/share/selinux`.

pupmod-simp-autofs

This module was extensively refactored. Please read the updated `README.md` to understand the current usage. Notable feature/API changes:

- Updated **autofs** service configuration to use `/etc/autofs.conf` in addition to `/etc/sysconfig/autofs`.
- Updated `/etc/autofs.master` to load content from `/etc/auto.master.simp.d/` and `/etc/auto.master.d/` in lieu of specifying map entries directly.
  - ‘auto.master’ entries are now written to files in `/etc/auto.master.simp.d`, a directory fully managed by this module.
  - `/etc/auto.master.d` is left unmanaged by Puppet.
  - Auto-converts from old maps directory to current maps directory and emits a warning. This is to help the 90% of the users who aren’t doing anything special with this module.
- Added a `autofs::map` defined type that allows the user to specify all the parameters for a ‘file’ map in one place. This resource will generate the appropriate resources to create both the ‘auto.master’ entry file and the map file.
- Added `autofs::masterfile` defined type to replace deprecated `autofs::master::map`.
  - `autofs::masterfile` creates an ‘auto.master’ entry file in `autofs::master_conf_dir`.
  - Unlike `autofs::map::master`, `autofs::masterfile` does not have a `content` parameter, because a user can simply use a `file` resource to specify a custom ‘auto.master’ entry file.
- Added `autofs::mapfile` defined type to replace deprecated `autofs::master::entry`.
  - `autofs::mapfile` creates a mapfile for a direct mapping or one or more indirect mappings.

    - Unlike `autofs::master::entry`, it does not have duplicate resource naming problems (wildcard or otherwise).

- `autofs` class changes
    - Added the following new **`autofs`** service configuration parameters:
        - `master_wait`
        - `mount_verbose`
        - `mount_nfs_default_protocol`
        - `force_standard_program_map_env`
        - `use_hostname_for_mounts`
        - `disable_not_found_message`
        - `sss_master_map_wait`
        - `use_mount_request_log_id`
        - `auth_conf_file`
        - `custom_autofs_conf_options`
    - Added `master_conf_dir` and `master_include_dirs` parameters to allow users to specify directories containing 'auto.master' entry files.
    - Added `maps_dir` to specify the location of SIMP-managed maps and changed the directory name from `/etc/autofs` to `/etc/autofs.maps.simp.d` for clarity.
    - Added `maps` to allow users to specify 'file' type maps in Hiera data.
        - Each map specifies the contents of a 'auto.master` entry file and its corresponding mapping file.
    - Renamed `options` to `automount_options` for clarity.
    - Renamed `use_misc_device` to `automount_use_misc_device` for clarity.
    - Removed `autofs::master_map_name`.
        - This parameter is not exposed in `/etc/autofs.conf` and does not look like it is intended to be changed.
    - Changed permissions of `/etc/auto.master` and `/etc/sysconfig/autofs` to match those of the delivered RPM.
- `autofs::ldap_auth` class changes
    - `autofs::ldap_auth` is now a private class to ensure the name of the configuration file created by this class matches the 'auth_conf_file' setting in `/etc/autofs.conf`.
    - Added `encoded_secret` optional parameter. This parameter takes precedence when both `secret` and `encoded_secret` parameters are specified.
- `autofs::map::master` has been deprecated by `autofs::map` or `autofs::masterfile`. Its behavior has changed from writing a section of `/etc/auto.master` to writing an *auto.master* entry file in `autofs::master_conf_dir`.
- `autofs::map::entry` has been deprecated by `autofs::map` or `autofs::mapfile`. Its behavior has changed from writing a file in `/etc/autofs` to writing a file in `autofs::maps_dir`.

pupmod-simp-clamav

- Updated documentation to clarify what `simp_options::clamav` actually does and to note that `clamav` was removed from the SIMP's default class list in SIMP 6.5.
- Set the default for `clamav::set_schedule::enable` to lookup `clamav::enable`, so that the class will remove the 'clamscan' **`cron`** job if management of ClamAV is disabled.
- Disable SIMP's **`rsync`** pulls by default.

pupmod-simp-compliance_markup

- Deep merge hash values in the Hiera backend.
- Improved confinement
  - Added support for confinement in 'profiles', 'controls' and 'ces' (as well as 'checks').
  - Added support for arrays of potential matches in confinement blocks.
  - Added support for structured facts in confinement.
  - Updated confinement logic to ensure that all possibilities are collected.
  - Apply confinement before merging values.
- Improved performance:
  - Reduced the amount of data passed around in the Hiera backend.
  - Ensured that the Hiera backend recurses as little as possible.
  - Removed useless loops in `list_puppet_params()`.
- Improved error handling and debugging:
  - Ignore undefined 'ces' when correlating checks and profiles.
  - Raise errors on malformed data.
  - Added debugging logs to enforcement logic.
- Removed all support for v1 data since it was experimental and removed in 3.0.0.
- Load data from the `compliance_markup::compliance_map` Hiera key after compliance profiles in modules to allow for profile tailoring via Hiera. This means that uses may now override all settings from the underlying compliance maps across all modules to fit their environment specifics.

pupmod-simp-cron

- Manage **`cron`** packages by default.

pupmod-simp-crypto_policy

This is a new module to manage, and provide information about, the system-wide crypto policies.

pupmod-simp-dconf

- Allow users to set custom settings via Hiera.

pupmod-simp-deferred_resources

- Remove 'ftp' and 'games' users and groups when enforcing STIG compliance.

pupmod-simp-dhcp

- Made use of SIMP's **`rsync`** operation optional (enabled by default for backwards compatibility).
- Added support for passing in a full `dhcpd.conf` entry.
- Ensured that the SELinux user and type are set for the configuration files.
- Switched to using `iptables::listen::udp` for **`firewalld`** compatibility.

pupmod-simp-fips

- Ensured that EL8 updates trigger updating the global system crypto policy, since some subsystems now ignore the local configuration by default.

pupmod-simp-freeradius

- Added support for overriding 'post-auth' in LDAP.
- Added support for overriding 'accounting' in LDAP.
- Added support for specifying the entire file content.
- Removed `simp_options::puppet::server` from the default lookup logic for `freeradius::v3::modules::ldap::server`. In systems that use Bolt to compile and apply manifests, that setting will not be available.

pupmod-simp-incron

- Remove pinned versions of incron, since the upstream packages have been fixed.

pupmod-simp-iptables

- Added preliminary support for acting as a pass-through to various **firewalld** capabilities using the simp/simp_firewalld module.
    - Using any of the `iptables::listen::*` defined types will work seamlessly in 'firewalld' mode but direct calls to `iptables::rule` will fail.
    - Calls to any of the native types included in this module will result in undefined behavior and is not advised.
    - To enable 'firewalld' mode on supported operating systems, simply set `iptables::use_firewalld` to `true` via Hiera.
    - EL8 systems will enable 'firewalld' mode by default.
- Improved the internal rule matching to handle most netmask and port updates.
- Added a `exact_match` Boolean to the `iptables_optimize` and `ip6tables_optimize` native types to allow for more aggressive rule matching.
    - This change requires that inbound rules match whatever is returned by **iptables-save** and/or **ip6tables-save** to prevent **iptables** flapping.
- Allow 'LOCAL-INPUT' jump rule in 'FORWARD' and 'INPUT' chains to occur last as a default action through the addition of an `iptables::rules::base::force_local_input` parameter.
- Allow users to disable adding the 'SIMP:' prefix to the rule comment.
- Allow users to disable comments on rules completely.

pupmod-simp-krb5

- Updated SELinux hotfix for EL8.
- Migrated SELinux hotfix to `vox_selinux::module`.

pupmod-simp-libreswan

- Removed unused `libreswan::use_certs_parameter` parameter.
- Added support for IKEv2 Mobility (RFC-4555) and mobile client connections.
- Added additional settings for DNS and Domains for Libreswan v3.23+.

pupmod-simp-libvirt

- Split out install and service into separate classes to give users more flexibility on what they manage with the module.

pupmod-simp-logrotate

- Allow all log size configuration parameters to be specified in bytes, kilobytes, megabytes, or gigabytes.
- Added ability to specify 'maxsize' configuration for specific **logrotate** rules.

pupmod-simp-named

- Allow users to force enabling/disabling of the **chroot** settings.
- Allow users to easily set the named_write_master_zones SELinux boolean in case they need to support dynamic DNS or zone transfers.

pupmod-simp-nfs

This module was extensively refactored. Read the updated README.md to understand the current usage. Notable feature/API changes:

- Overall changes
  - Dropped **stunnel** support for NFSv3. This tunneling did not work because:
    - The NFS client sends the NFS server Network Status Manager (NSM) notifications via UDP, exclusively.
    - At multi-NFS-server sites, a unique rpcbind port per server is required in order for a NFS client to be able to tunnel its server-specific RPC requests to the appropriate server.
  - nfs class
    - Reworked parameters to reflect configuration of /etc/nfs.conf and, for limited EL7-only configuration, /etc/sysconfig/nfs. See the class documentation for full details.
  - Removed stunnel_systemd_deps and stunnel_tcp_nodelay parameters throughout the module.
    - These parameters were not consistently used in the manifest code (i.e., declared but not used) and were confusing.
    - The corresponding stunnel_socket_options and stunnel_wantedby parameters in classes/defines now use defaults that were intended to be set by those parameters.
  - Now masks NFS services that are not needed, so they are not unnecessarily started when the **nfs-server.service** or **nfs-client.target** are restarted.
- nfs::client changes
  - Added support for pNFS: Set blkmap to true to enable the pNFS service, **nfs-blkmap.service**.
  - Added nfs::stunnel_socket_options and stunnel_wantedby parameters which provide the defaults for all nfs::client::mount instances.
- nfs::client::mount define changes
  - nfs_server must now be specified as an IP address. This change was necessary for **firewalld**.
  - In options, changed the default mount type to 'soft' instead of 'hard'. Also removed deprecated 'intr' option, as it has no effect.
  - Reworked the remote autodetect logic to detect a local mount based on IP address instead of simply whether the node is also configured to be an NFS server.

  - Added support for direct autofs mounts and simplified specification of indirect mounts. When `autofs_indirect_map_key` is not specified, a direct mount is specified by `name`. When `autofs_indirect_map_key` is specified, an indirect mount is specified with `name` as the mount point and `autofs_indirect_map_key` as the mount key.
  - Renamed `autofs_map_to_user` to `autofs_add_key_subst` to better reflect automount terminology. This parameter simply adds key substitution to the remote location, which although can be used for user home directories, is not restricted to that use case.
  - Renamed `port` to `nfsd_port` to be consistent with the name of that parameter throughout the entire module.
  - Renamed `v4_remote_port` to `stunnel_nfsd_port` for clarity and to be consistent with the name of that parameter throughout the entire module.
  - Exposed client **`stunnel`** configuration that was scattered throughout the module to this API. User can now specify `stunnel_socket_options` and `stunnel_verify` for each mount. When unspecified, the defaults from the `nfs` class are used.

- `nfs::server` class changes
  - Exposed server **`stunnel`** configuration that was scattered throughout the module to this API. User can now specify `stunnel_accept_address`, `stunnel_nfsd_accept_port`, `stunnel_socket_options`, `stunnel_verify`, and `stunnel_wantedby` in this class. When unspecified, the defaults for all but `stunnel_accept_address` and `stunnel_wantedby` are pulled from the `nfs` class.
  - Added the following parameters: `nfsd_vers4`, `nfsd_vers4_0`, `nfsd_vers4_1`, `nfsd_vers4_2`, and `custom_rpcrquotad_opts`.
  - Renamed `nfsv3` to `nfsd_vers3` to reflect its use in `/etc/nfs.conf`.
  - Moved `nfs::rpcquotad_port` to this class and renamed `rpcrquotadopts` to `custom_rpcrquotad_opts` for clarity.
  - Moved `nfs::mountd_port` to this class and removed `rpcmountdopts`. Custom configuration for that daemon should now be made via `nfs::custom_nfs_conf_opts` or `nfs::custom_daemon_args` as appropriate.
  - Removed the obsolete `nfsd_module` parameter.
- `nfs::server::export` define changes
  - Added `replicas`, `pnfs`, and `security_label` parameters to support additional export configuration parameters.
- `nfs::idmapd` class changes
  - Refactored into 3 classes to support distinct NFS server and client configuration
  - Added `no_strip` and `reformat_group` to `nfs::idmapd::config` to support additional `/etc/idmapd.conf` configuration parameters.

pupmod-simp-oath

- Allow `oath::config::user` to be any string.
- Disabled `show_diff` option in `concat` for `/etc/liboath/users.oath` to prevent that information from being exposed in logs.

pupmod-simp-pam

- **Ensured that 'pam_tty_audit' is optional if auditing is not enabled on the**
  system.
- Added the ability to specify `pam::limits::rules` via Hiera.
- Ignore **`authconfig`** disable on EL8. Authconfig was replaced with **`authselect`** and **`authselect`** does not overwrite settings unless you select the `--force` option.
- Remove installation of pam_pkcs11 and fprintd-pam by default, since they aren't actually required for basic functionality.

pupmod-simp-polkit

- Added the following classes:
  - `polkit::install`
  - `polkit::service`
  - `polkit::use`
- Ensured that the polkit user is managed by default and placed into the supplementary group bound to the 'gid' option on `/proc`, if one is set. This is necessary to work around issues with 'hidepid' > 0.
- Made the entire main class inert on unsupported OSs; logs a warning on the server that can be disabled.

pupmod-simp-pupmod

- Default `pupmod::master::ssl_protocols` to TLSv1.2 only.
- Use `$facts['certname']`, when available, in the parameters below, because `$facts['fqdn` may not be appropriate when the system does not use its primary NIC/FQDN for its Puppet certificate.
  - `pupmod::certname`
  - `pupmod::master::ca_status_whitelist`
  - `pupmod::master::admin_api_whitelist`
- Set the default **`puppetserver`** ciphers to a safe set.
- Added better auto-tuning support for **`puppetserver`**, based on best practices.
- Added 'ReservedCodeCache' **`puppetserver`** support.
- Removed **`incron`** support in favor of using **`systemd`** path units to run **`simp_generate_types`**.
  - Attempts to activate the **`incron`** code will result in a warning message.
- Added mitigation for CVE-2020-7942
- Added optional management of the Facter configuration file.

- Removed the deprecated CA CRL pull **cron** job and the corresponding pupmod::ca_crl_pull_interval parameter.
- Removed deprecated auth.conf support for the legacy pki module and the corresponding parameters:
  - pupmod::master::simp_auth::legacy_cacerts_all
  - pupmod::master::simp_auth::legacy_mcollective_all
  - pupmod::master::simp_auth::legacy_pki_keytabs_from_host
- Removed the deprecated pupmod::master::simp_auth::server_distribution parameter.

pupmod-simp-resolv

- Added optional management of DNS servers via **nmcli**.

pupmod-simp-rsyslog

- Added support for 'KeepAlive' variables for 'imtcp' and 'omfwd' actions.
- Changed local rule defined type to use the same package defaults for action queues that are in the remote rule defined type.
- Changed remote rule defined type to use package defaults for action queues.
- Added a default rule to log packets dropped by **firewalld** to /var/log/firewall.log.
- Added /var/log/firewall.log to SIMP's 'syslog' **logrotate** rule.
- Added logrotate::rule options to rsyslog::conf::logrotate class.
- Removed the filter_ rules that were present for an old (and broken) version of the simp/simp_firewalld module.
- Removed params pattern and migrated to data in modules.

pupmod-simp-selinux

- Allow users to include selinux::install without needing full SELinux system management. This is particularly important when the native types are to be used in different modules but you don't want to include full management just to get the required packages
- No longer enable or install **mcstransd** by default. It is a user convenience feature and not required for core functionality.
- Ensured that **mcstransd** is added to the GID assigned to /proc if one is assigned on the system.

pupmod-simp-simp

- **sssd** configuration updates
  - Configure the 'files' provider in lieu of the 'local' provider for EL7 and later.
  - Deprecated the following parameters in simp::sssd::client: autofs, ssh and sudo. The simp/sssd module configures services in sssd::services. Use that parameter to configure those entries.
  - Configure **sssd** for EL8, even if the ldap_domain and local_domain parameters of simp::sssd::client are set to false.

- Updated `simp::mountpoints::proc` to ensure **`polkitd`** can be configured to have access to `/proc`:
  - Assign a group and gid by default.
  - Create a group by default.
  - Discover these values from the system if possible.
- Removed the following applications from the list of base OS applications installed automatically by simp/simp:
  - man
  - man-pages
  - vim-enhanced
  - dos2unix
  - elinks
  - hunspell
  - lsof
  - mlocate
  - pax
  - pinfo
  - sos
  - star
  - symlinks
  - words
  - x86info
- Deprecated the `simp::base_apps::manage_elinks_config` parameter.
  - It no longer has any effect.
- `simp::nsswitch` updates
  - Updated the `simp::nsswitch` class to have sane defaults.
    - Added support for 'mymachines' and 'myhostname' by default.
    - Removed all NIS references since NIS should not be in general usage any longer and was never natively supported by SIMP.
    - Configuration files are now common across all supported OSs since **`nsswitch`** "does the right thing" when it hits a module that it does not recognize.
  - Allow **`nsswitch`** overrides.
- Added **`chronyd`** support for EL8
  - Moved ntp to list of OS relevant applications for EL6 and EL7.
  - Added chrony for EL8.
- Updated the client kickstart scripts/configuration
  - Updated the **`bootstrap_simp_client`** script to use **`chronyd`** if the kernel version is 4 or later.
  - Deprecated the `simp::server::kickstart::runpuppet` parameter and removed the old, corresponding **`runpuppet`** kickstart scripts. The **`simp_bootstrap_client`** scripts should be used instead.

- ClamAV updates:
  - Removed `clamav` from the list of classes included by default in the SIMP scenarios.
    - This will not remove ClamAV from systems where it is installed; Puppet will simply stop managing it.
    - To continue managing ClamAV with Puppet, add `clamav` to `simp::classes` in the appropriate Hiera file for that SIMP client.
    - See the simp/clamav module for information on configuring or removing ClamAV on a system.
  - Deprecated `simp::server::clamav`.
    - This parameter will be removed in a future SIMP release.
    - To manage ClamAV on the SIMP server after the parameter is removed, manually add the `clamav` class to the `simp::classes` array in the SIMP server's Hiera file.
- `simp::yum::repo*` updates:
  - Added:
    - `simp::yum::repo::internet_simp` class:
      - Uses the SIMP yum repository package (simp-community-release) to configure yum for SIMP's internet public repositories at simp-project.com.
      - simp-project.com is the new host for SIMP's yum repositories.
      - packagecloud is no longer being updated.
    - `simp::yum::repo::simp_release_version` function: Returns the SIMP release version for use in the SIMP internet yum repositories.
    - `Simp::Version` data type alias for valid version strings for use in the SIMP internet repositories.
    - New parameters to `simp::yum::repo::local_simp` and `simp::yum::repo::local_os_updates`:
      - `relative_repo_path`, `baseurl`, and `gpgkey`.
      - `baseurl` and `gpgkey` allow complete `yumrepo` resource overrides.
  - Deprecated:
    - `simp::yum::repo::internet_simp_server` and `simp::yum::repo::internet_simp_dependencies` classes:
      - These resources are no longer useful because their API matches the OBE packagecloud SIMP repositories.
      - As a workaround, the classes have been modified to use `simp::yum::repo::internet_simp` to configure the correct repositories at simp-project.com.
      - You should switch to using `simp::yum::repo::internet_simp`, directly, as these classes will be removed in a future release.
    - `simp::yum::repo::sanitize_simp_release_slug` function: a function only useful to the deprecated classes.
- Added `simp::puppetdb::cipher_suites` parameter to manage the cipher suites supported by PuppetDB's HTTP interface (Jetty).
  - Used to set `puppetdb::cipher_suites`.
  - Value set to a safe set.
- Call `selinux::install` prior to using native types that require the packages to be installed.

pupmod-simp-simp_apache

- Default to only TLS1.2.

pupmod-simp-simp_banners

- Removed all OS support statements from `metadata.json`, since this is simply a data-only module.

pupmod-simp-simp_bolt

- Added plan to install puppet-agent on target nodes.
- Configured Bolt to request a pseudo TTY for SSH sessions if specified.
- Configured new logs to be appended to the log file instead of overwriting.

pupmod-simp-simp_firewalld

This is a new SIMP module that provides a profile class and defined type to manage the system's **firewalld** with "safe" defaults and safety checks for **firewalld** rules. It uses the puppet/firewalld module to update the system's **firewalld** configuration.

pupmod-simp-simp_gitlab

Updated for the latest GitLab application (13.5.x) and puppet/gitlab (6.0.1).

- Removed:
  - Support for GitLab < 12.3.0.
  - TLSv1.1 from the default for `simp_gitlab::ssl_protocols`.
- Changed:
  - Set the GitLab root password in a fashion that minimizes coupling of simp/simp_gitlab with the internals of puppet/gitlab.
    - Set a throw-away password during initial GitLab package installation using GitLab configuration in `/etc/gitlab/gitlab.rb`. Setting the password during initial install is the **only** way to ensure the password is not set by an external user. Otherwise, the first GitLab page that comes up is the page to reset the root password.
    - After GitLab initial configuration, set the real root password using a script that implements Gitlab-provided procedures for resetting the password.
  - Use **chronyd** instead of **ntpd**, as GitLab itself uses **chronyd** and **chronyd** is required for EL8.
  - Use puppet/gitlab for managing packages again.
  - Renamed the 'gitlab_monitor' key to 'gitlab_exporter' in the configuration hash.
    - The name change is required for GitLab >= 12.3.0.
  - No longer set `gitlab::external_port`
    - The custom port is already appropriately configured via the `gitlab::external_url`.

    - 'external_port' is no longer a supported GitLab configuration key and causes **`gitlab-ctl reconfigure`** to fail.
  - simp/simp_gitlab now fails to compile when the node is in *FIPS* mode, unless `simp_gitlab::allow_fips` (a new parameter) is set to `true`.
- Added:
  - Parameters to enable setting the GitLab root password
    - `simp_gitlab::set_gitlab_root_password`
    - `simp_gitlab::gitlab_root_password`
    - `simp_gitlab::rails_console_load_timeout`
  - A script to change the GitLab root password, **`/usr/local/sbin/change_gitlab_root_password`**.
  - Disabling of Let's Encrypt usage in GitLab, by default.
    - The integration of SIMP PKI management with with Let's Encrypt has not yet been done.
    - To use Let's Encrypt, disable SIMP management of PKI by setting `simp_gitlab::pki` to `false` and then manage the certificates manually.
  - `svckill::ignore` rule for the GitLab service. Since the service is no longer managed by default by `gitlab::service`, this prevents the service from being inadvertently killed when it is unmanaged.

Important: As a side effect of the changes related to setting the GitLab root password, upon module upgrade, the GitLab root password will be automatically set to the value of `simp_gitlab::gitlab_root_password`, unless the (empty) marker file `/etc/gitlab/.root_password_set` exists or the parameter `simp_gitlab::set_gitlab_root_password` is set to `false`. If you forget to disable this automation or just want to reset the GitLab root password, simply run

```
/usr/local/sbin/change_gitlab_root_password <new_password>
```

You do not need to know the previous password to set the new password.

pupmod-simp-simp_ipa

- Make the IPA server optional in the `join` task. It is perfectly valid to not specify a server when doing an IPA client install and instead rely on DNS auto discovery.

pupmod-simp-simp_nfs

- The following parameters had to be changed from hostnames or IP addresses to only IP addresses due to use of **`firewalld`** on EL8:
  - `simp_nfs::home_dir_server`
  - `simp_nfs::mount::home::nfs_server`

pupmod-simp-simp_options

- The simp_options::clamav catalyst has been deprecated.
  - As of SIMP 6.5, SIMP's clamav class is no longer included in the class list of the SIMP scenarios. So, this catalyst is not needed to disable it.
  - To have SIMP manage ClamAV on your system, add the clamav class to your system's class list.
  - See the simp/clamav module README.md for information on managing ClamAV.
- simp_options::puppet::server and simp_options::puppet::ca are now optional.
  - These are no longer required at all times due to support for Bolt. Code that used these parameters will correctly fail and require users to add them to their configuration.
- Updated simp_options::ldap to require the master and uri parameters if simp_options::puppet::server is not defined.

pupmod-simp-simp_rsyslog

- Added support for **firewalld** log message collection.
- Deep merge simp_rsyslog::log_collection.
- Removed the filter_IN_99_simp_DROP rules that were present for an old (and broken) version of the simp/simp_firewalld module.

pupmod-simp-simp_snmpd

- Changes:
  - Updated to use puppet/snmp version 5.1.2.
  - The default configuration for this module has not changed but some settings are now placed in the snmpd.conf file instead of in a subdirectory.
  - In the previous version the user directory was automatically included. Now the user must set simp_snmpd::include_userdir to true for files in the user directory to be included. The relevant parameters are as follows:
    - simp_snmpd::include_userdir
    - simp_snmpd::user_snmpd_dir
  - The configuration parameter simp_snmpd::snmpd_conf_file has been renamed to simp_snmpd::service_config. This is the location of the the snmpd.conf file.
  - The type of the simp_snmpd::services parameter has been changed from a String to an Integer.
  - The simp_snmpd::system_info parameter has been deprecated. puppet/snmp now includes these settings by default and they can't be removed. This means that **net-snmp** will set them as not writable and they can not be changed by a set call from an **snmpd** manager or client.
- New features:
  - Added settings to allow users to change owner/group and permissions on configuration files:
    - simp_snmpd::service_config_dir_owner
    - simp_snmpd::service_config_dir_group

    - `simp_snmpd::service_config_dir_perms`
    - `simp_snmpd::service_config_perms`
  - Added configuration of **`snmpd`** user and group IDs, as well as optional management of the user and group:
    - `simp_snmpd::snmpd_uid`
    - `simp_snmpd::snmpd_gid`
    - `simp_snmpd::manage_snmpd_user`
    - `simp_snmpd::manage_snmpd_group`
  - The SNMP trap daemon is still stopped by default. New parameters can be used to enable the daemon, set the command line options on the daemon and start it at boot. The default settings in puppet/snmp are used. Configuration files placed in a user directory can created by the user for any additional configuration. The following settings have been added to create this behavior:
    - `simp_snmpd::trap_service_ensure`
    - `simp_snmpd::trap_service_startatboot`
    - `simp_snmpd::trap_service_config`
    - `simp_snmpd::snmpdtrapd_options`
    - `simp_snmpd::user_trapd_dir`

pupmod-simp-simpkv

This is a new SIMP module that provides an abstract library that allows Puppet to access one or more key/value stores.

This module provides

- a standard Puppet language API (functions) for using key/value stores
- a configuration scheme that allows users to specify per-application use of different key/value store instances
- adapter software that loads and uses store-specific interface software provided by the simp/simpkv module itself and other modules
- a Ruby API for the store interface software that developers can implement to provide their own store interface
- a file-based store on the local filesystem and its interface software.
  - Future versions of this module will provide a distributed key/value store.

pupmod-simp-simplib

Facts Changes

Added the following facts:

Fact	Description
`simplib__auditd`	Returns a hash of **`auditd`** status.
`simplib__firewalls`	Return an array of known firewall commands that are present on the system.
`simplib__mountpoints`	Returns a hash of mountpoints of particular interest to SIMP modules.
`simplib__numa`	Returns a hash of NUMA values.
`simplib__efi_enabled`	Returns `true` if the host is using EFI.
`simplib__secure_boot_enabled`	Returns `true` if the host is using UEFI Secure Boot.

Deprecated the following facts:

- `tmp_mounts` fact. Use `simplib__mountpoints`, instead.

Function Changes

Added the following functions:

Function	Description
`simplib::debug::inspect`	Enhanced version of `simplib::inspect`.
`simplib::debug::classtrace`	Prints a trace of all catalog resources traversed to get to the current point.
`simplib::debug::stacktrace`	Prints a trace of all files traversed to get to the current point.
`simplib::ip::family_hash`	Takes an IP address or array of IP addresses and returns a hash with the addresses broken down by family. The returned hash also contains additional helpful metadata.
`simplib::module_metadata::os_blacklisted`	Determines if the passed metadata indicates that the current OS has been blacklisted.
`simplib::module_metadata::os_supported`	Determines if the passed module metadata indicates that the current OS is supported.
`simplib::module_metadata::assert`	Adds an assertion based on whether the OS is supported or blacklisted.
`simplib::caller`	Determines what called a function.
`simplib::passgen::gen_password_and_salt`	Generates a password and salt.
`simplib::passgen::gen_salt`	Generates a salt.
`simplib::passgen::get`	Retrieves a generated password and any stored attributes.
`simplib::passgen::list`	Retrieves the list of generated passwords with attributes and the list of sub-folders stored at a `simplib::passgen` folder.
`simplib::passgen::remove`	Removes a generated password, history and stored attributes.
`simplib::passgen::set`	Sets a generated password with attributes.
`simplib::safe_filename`	Convert a string into a is filename that 'path safe'.

Updated the following functions:

- `simplib::passgen`
  - Added 'simpkv' mode.
    - Runs in 'legacy' mode (default) or in a 'simpkv' mode.
    - 'simpkv' mode is **EXPERIMENTAL**.
    - When in 'simpkv' mode, `simplib:passgen` uses simp/simpkv for password persistence.
    - 'simpkv' mode is enabled by setting `simplib::passgen::simpkv` to `true` in Hiera.
    - If you enable 'simpkv' mode in a system that already has passwords generated via the legacy code, currently, **all passwords will be regenerated**.
    - Added `simpkv_options` parameter to `simplib::passgen` for use in 'simpkv' mode.
  - Enhanced `simplib::passgen` operation when in 'simpkv' mode
    - Stores `complexity` and `complex_only` setting in the password's simpkv metadata, so that the password can be regenerated with the same characteristics.
    - Regenerates the password if the requested 'complexity' or 'complex_only' setting differs from the setting used for the latest persisted password.
    - Stores up to the latest 10 <password,salt> pairs in the password's simpkv metadata.
  - Added a `gen_timeout_seconds` password option. Previously this was hardcoded to 30 seconds.
  - Added ability to set the user and group for legacy `simplib::passgen` files.

  - Changed the default permissions on legacy simplib::passgen files to the user running the catalog compile. This will allow bolt to set permissions correctly.

- simplib::gen_random_password:
  - Intersperse special characters among the alpha-numeric characters, when complexity is 1 or 2 and complex_only is false. Previously, this function grouped the all alpha-numeric characters together and grouped all special characters together. This generated passwords that were not suitable for user passwords, as they would fail the cracklib/libpwquality complexity checks.
- simplib::assert_metadata:
  - Added blacklist option. This allows functionality to deliberately fail on an OS that is listed in the module's metadata.json, but is not necessarily supported by all parts of the given module.

New data type aliases

Added Simplib::Systemd::ServiceName for valid **systemd** service names.

pupmod-simp-ssh

- Migrated to the updated version of simp/selinux that allows for isolated package installation in support of the SELinux native types.
- Allow users to use the puppet/selinux module instead of SIMP components.

pupmod-simp-stunnel

- Set default for stunnel::connection::ssl_version to TLSv1.2 for EL8 compatibility.
- Set default for stunnel::instance::ssl_version to TLSv1.2 for EL8 compatibility.
- Set the stunnel::connection::app_pki_crl parameter to undef by default due to issues with pointing the setting to an absent directory in EL8.
- Set the stunnel::instance::app_pki_crl parameter to undef by default due to issues with pointing the setting to an absent directory in EL8.
- Updated valid ssl_version entries.

pupmod-simp-sudo

- Added parameters for sudo::default_entry and sudo::alias defined types.
- CVE-2019-14287 mitigation.
  - Do not allow the use of user id or group id of '-1' when 'ALL' or '%ALL' are used in the runas section of a **sudo** user specification and the version of **sudo** is earlier than 1.8.28.
- Deep merge user_specifications by default.

pupmod-simp-svckill

- Updated the `svckill` provider to work with different Puppet `service` provider implementations.
  - If after a Puppet upgrade you find that `svckill` is trying to kill system services that it previously ignored, you need simp/svckill version 3.6.1 or later to fix the problem.
- Updated service lists.

pupmod-simp-swap

- Disable `dynamic_swappiness` by default.
- Set the static system swappiness to 60 by default.

pupmod-simp-tcpwrappers

- Enhanced behavior to do nothing when TCP Wrappers is not supported by the OS.

pupmod-simp-tpm2

- Removed the option for managing tools, `tpm2::manage_tpm2_tools`. Tools can be managed or not by removing them from the package list. Note that the tools package is needed to determine the status of the TPM.
- Added support for setting `tabrm_options` for connecting to the simulator.

pupmod-simp-useradd

- Added explicit support for setting the rescue/emergency shell on systemd systems.

rubygem-simp-cli

- Updated the instructions provided in the local user lockout warning message in the bootstrap lock file.
  - Simplified instructions to create resources via Hiera.
  - Tell the user to check that they can **`ssh`** into the server with the new user after bootstrap but before rebooting. This step is imperative to ensure that the user can also get through Puppet-managed authentication!
- Updated SIMP internet repositories configured by **`simp config`**.
  - Now uses simp-project.com repositories via the new `simp::yum::repo::internet_simp` class.
  - The packagecloud repositories are no longer being updated.
- Allow users to set the 'SIMP_ENVIRONMENT' environment variable to change the initial environment from 'production' to a custom value, when running **`simp config`** or **`simp bootstrap`**.
- **`simp config`** changes
  - Ensured that **`simp config`** uses the `simp::classes` parameter instead of `classes` by default, but accept both `simp::classes` and `classes` as valid existing configurations.
  - Removed deprecated `--non-interactive` option. Use `--force-defaults` instead.
- Added **`simpkv`** command family to allow users to manage and inspect entries in a simpkv key/value store

- **simp passgen** changes
  - Split into sub-commands for ease of use:
    - **simp passgen envs**: List environments that may have simplib::passgen passwords.
    - **simp passgen list**: List names of simplib::passgen passwords.
    - **simp passgen remove**: Remove simplib::passgen passwords.
    - **simp passgen set**: Set simplib::passgen passwords.
    - **simp passgen show**: Show simplib::passgen passwords and other stored attributes.
  - Updated to work with simpkv-enabled simplib::passgen. Automatically detects whether simplib::passgen is operating in 'legacy' mode or 'simpkv' mode in the specified environment, and then executes password operations using the appropriate mechanism for that mode.
  - When setting passwords, disabled libpwquality/cracklib validation of user-entered passwords, by default, because not all passwords managed by simplib::passgen are user passwords. This validation can be re-enabled with the --validate option of **simp passgen set**.
  - Added the following command line options when creating passwords
    - --[no-]auto-gen: Whether to auto-generate new passwords.
    - --complexity: Password complexity to use when a password is auto-generated. Corresponds to the complexity option of simplib::passgen.
    - --[no-]complex-only: Whether to only use only complex characters when a password is auto-generated. Corresponds to the complex_only option of simplib::passgen.
    - --[no-]validate: Enables validation of new passwords with libpwquality/cracklib.
    - --length: Password length to use when a password is auto-generated.
  - Added --[no-]details option when showing password information. When enabled, all available password information is displayed, not just the current and previous password values.
- Updated HighLine from version 1.7.8 to 2.0.3.

simp-adapter

- Removed logic to ensure any existing, global hiera.yaml.simp file is not removed on upgrade from simp-adapter <= 0.0.6.
  - This is not an issue when upgrading from SIMP 6.4.0 to SIMP 6.5.0 (i.e., simp-adapter version 1.0.1 to version 2.0.0).
  - If for some reason you are upgrading from simp-adapter version <= 0.0.6, manually save off /etc/puppetlabs/puppet/hiera.yaml.simp prior to the upgrade, and then restore that file after the upgrade is complete.

simp-environment-skeleton

- Ensure that **firewalld** is used by default in the applicable SIMP scenarios.
- Ensured that the server Hiera defaults have `simp::server` in the `simp::classes` array. Otherwise, it will never get picked up.
- Replace `classes` with `simp::classes` and `simp::server::classes` as appropriate in example Hiera YAML files.
- FakeCA updates
  - Added the CA code directly into the project to allow the code to work on newer OS versions
  - Allow users to specify an alternate output directory via a 'KEYDIST' environment variable.
  - Consolidate the certificate request and revocation code.
  - Certificate revocation now runs in linear time.
- Changed permissions for files and directories to be world readable.
- Add a PE-suitable Puppet server YAML data template.

simp-gpgkeys

- Added the CentOS 8 and EPEL 8 GPG keys.
- Removed Fedora 25 and 26 GPG keys.
- Updated puppetlabs GPG key.

simp-rsync-skeleton

- Added mitigation for CVE-2019-6477 to the sample, RedHat 7 `named.conf`.
- Removed `rndc.key` files from sample named configuration to prevent users from accidentally using a published, sample secret key.
  - The **named** service will create a key if one does not exist using the correct defaults for the system.
- Updated the `README` in `rsync/RedHat/Global/tftpboot/linux-install`.
  - It now explains which boot files for the *TFTP* boot server are required when `tftpboot::use_os_files` is set to `false`.

simp-utils

- Added sample kickstart files to `/usr/share/simp/` to allow users to have access to all OS-specific versions of the kickstart files.
- Added a check to the **unpack_dvd** script for dangerously unspecific OS versions (e.g., '7' instead of '7.0.2003').
  - This is common when **unpack_dvd** autodetects the OS version from the ISO's `.treeinfo` on some OSes (particularly CentOS).
  - It can result in clobbering of existing OS files, when the script unpacks files into a directory names for the major OS version.

  - The script will exit with an informative message and instructions for how the user can address the issue with the `-v` option.

- Added (optional) `--unpack-pxe [DIR]` option to the **unpack_dvd** script.
  - Added (optional) `--environment ENV` to set the PXE rsync environment.
  - Added a new `--[no-]unpack-yum` (enabled by default), to permit users to disable the RPM unpack.
  - To enable unpacking PXE tftpboot files, run with `--unpack-pxe`.
  - To disable unpacking RPMs/yum repos, run with `--no-unpack-yum`.
  - See **unpack_dvd --help** for details.
- Overhauled **unpack_dvd --help**; output now fits on 80-character PTY consoles.

SIMP ISO

- Fixed a bug in the instructions about enabling encryption in non-FIPS mode in the sample client kickstart files.
  - Following the erroneous instructions prevented automatic decryption from happening at client boot, because the encrypted disk credentials were not added to the **dracut** configuration.

Known Bugs and Limitations

Below are bugs and limitations known to affect this release. If you discover additional problems, please submit an issue to let use know.

Special considerations with EL8 clients

Network-isolated EL8 clients require EPEL8 and EL8 Base/Updates dnf mirrors

Because there is no SIMP 6.5 EL8 server release, there is no accompanying EL8 ISO or package tarball that can be used to create a self-hosted dnf repository for SIMP-specific EL8 packages.

In order to provide the necessary packages to EL8 agents on a network-isolated SIMP 6.5 infrastructure, admins must ensure that dnf repo mirrors are available for:

- EL8 Base/Updates
- EPEL 8
- Puppet EL8

unpack_dvd does not (re-)create modular repos for EL8 dnf repos (SIMP-8614)

EL8 introduces modular package repositories. When unpacking an EL8 ISO to populate a yum repository, SIMP 6.5.0's **unpack_dvd** script does not recognize or correctly package repository modules. Consequently, EL8 Puppet agents applying catalogs that require modular EL8 packages may encounter errors like the following:

```
Error: /Stage[main]/Simp_apache::Install/Package[httpd]/ensure: change from 'purged' to
↪'latest' failed: Could not update: Execution of '/usr/bin/dnf -d 0 -e 1 -y install␣
↪httpd' returned 1: No available modular metadata for modular package 'httpd-2.4.37-21.
↪module_el8.2.0+382+15b0afa8.x86_64', it cannot be installed on the system
Error: No available modular metadata for modular package
```

4.2.3 SIMP Community Edition (CE) 6.4.0-0

Contents

This release is known to work with:

- CentOS 6.10 x86_64
- CentOS 7.0 1810 x86_64
- OEL 6.10 x86_64
- OEL 7.6 x86_64
- RHEL 6.10 x86_64
- RHEL 7.6 x86_64

Warning: Local modules installed in the modules directory can get deleted during upgrade. Make sure you read the upgrade instructions carefully.

Breaking Changes

- Puppet 4 is no longer supported.

Significant Updates

The most significant change with SIMP 6.4.0 is a redesign of the packaging of SIMP RPMs, in order to support management of one or more *Puppet environments*.

The change allows SIMP users to easily use *r10K* or *Code Manager* to maintain these environments, even when their systems are on isolated networks.

Multiple environments are key to implementing standard *DevOps* work flows such as DTAP (Development > Testing > Acceptance > Production).

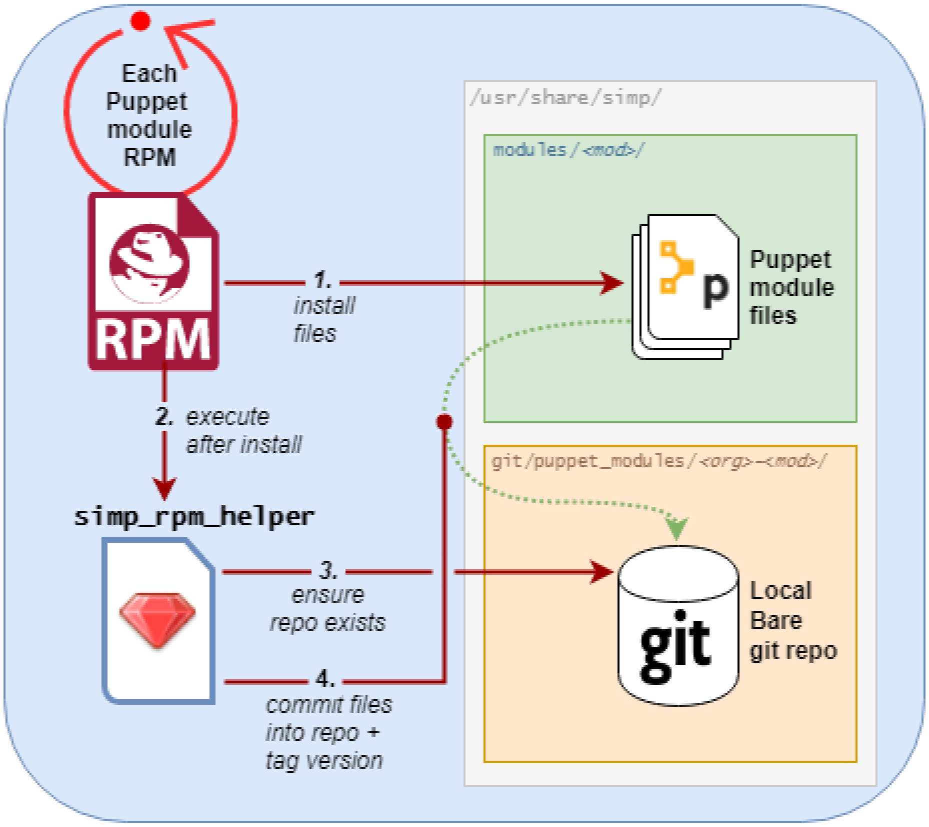
At the root of this change are the following major features:

- Puppet modules installed via SIMP-packaged RPMs are now imported into local, SIMP-managed Git repositories at `/usr/share/simp/git/puppet_modules`, in addition to being installed in `/usr/share/simp/modules`.
- No SIMP component RPM install/upgrade/erase operation will modify an active Puppet environment or SIMP secondary environment, `/var/simp/environments/simp`.
- SIMP's *CLI* provides commands that help users create and maintain one or more *SIMP Omni-Environments*, each of which is comprised of the following:
  - A Puppet environment in `/etc/puppetlabs/code/environments`
  - A *SIMP Secondary Environment* in `/var/simp/environments`
  - A *SIMP Writable Environment* in `/opt/puppetlabs/server/data/puppetserver/simp/environments`.
- The `simp-environment-skeleton` and `simp-rsync-skeleton` packages install a SIMP Omni-Environment skeleton at `/usr/share/simp/environment-skeleton`.
- Unless a `production` SIMP Omni-Environment already exists, `simp config` will create a `production` SIMP Omni-Environment from SIMP's skeleton and local Git repositories.

Module RPM Installation

In SIMP 6 versions <= 6.3.3, when a SIMP Puppet module RPM was installed, its contents could also be automatically copied into the `simp` Puppet environment, the standard active Puppet environment for a SIMP server. This copy was affected by the `simp_rpm_helper` script of the `simp-adapter`.

In SIMP 6.4.0, the `simp_rpm_helper`'s behavior has been redesigned as shown in the following figure:



In lieu of the auto-updating `/etc/puppetlabs/code/environments/simp`, the `simp_rpm_helper` now creates and maintains local Git repositories for Puppet modules installed via SIMP-packaged RPMs. This change allows SIMP users on isolated networks to manage one or more Puppet environments easily, using r10K or Code Manager. The use of r10K/Code Manager, in turn, provides Puppet module installation that aligns with current Puppet best practices.

Other SIMP Asset RPM Installation

The packaging of key non-module SIMP assets (previously named `simp-environment` and `simp-rsync`) has been redesigned to no longer modify directories actively being used by deployed Puppet environments. This change has several benefits:

- Asset RPM installs/upgrades/erases have no unintended consequences on active Puppet environments. These operations are completely safe! Previously, RPM erase operations could remove files or rename them (add '.rpm-save' suffixes), even when they were actively being used by the 'simp' Puppet environment.
- Sample files will always be available. Previously, for any asset file that was locally modified, changes to that file during RPM upgrade were hidden. The user had to manually extract files from the RPM to view the changes.

In addition, the naming and dependency tree of these assets have been reworked:

- The `simp-environment` package has been split into `simp-environment-skeleton` and `simp-selinux-policy` packages.
- The `simp-rsync` package has been deprecated. Its replacement is named `simp-rsync-skeleton`.
- The asset RPMs no longer require specific versions of `puppet-agent` and `puppetserver` RPMs. Users now have more flexibility in the versions of Puppet artifacts they can use in their infrastructure.

SIMP CLI Enhancements

New commands have been added to the SIMP CLI in order to support Git-based module deploy.

- `simp puppetfile generate` has been added to generate two different flavors of a SIMP-specific *Puppetfile*.
- `simp environment new` and `simp environment fix` have been added to help users manage the SIMP omni environment.

In addition, `simp config` has been updated to allow users to opt-out of SIMP-provided LDAP capabilities.

Puppet 3 API Function Deprecations

In preparation for full Puppet 6 support, all SIMP-provided Puppet 3 API functions are officially deprecated and **will be removed** in the next SIMP release. The affected functions and their replacements, as appropriate, are listed in sub-sections below.

In addition, SIMP deprecation warnings have been enabled by default. To disable SIMP deprecation warnings set environment variable `SIMP_NOLOG_DEPRECATIONS=true` on the Puppet server.

simp-compliance_markup Deprecated Functions

Puppet 3 API Function	Replacement	Replacement Source
compliance_map	**compliance_markup::compliance_map**	simp-compliance_markup >= 3.0.0

simp-simp_apache Deprecated Functions

Puppet 3 API Function	Replacement	Replacement Source
apache_auth	**simp_apache::auth**	simp-simp_apache >= 6.0.1
apache_limits	**simp_apache::limits**	simp-simp_apache >= 6.0.1
munge_httpd_networks	**simp_apache::munge_httpd_networks**	simp-simp_apache >= 6.0.1

simp-simplib Deprecated Functions

Important: Most but not all deprecated functions in the table below have replacements. If any deprecated function without a replacement is essential to you, please enter a feature request.

Puppet 3 API Function	Replacement	Replacement Source
array_include	Puppet language **in** *or* Puppet built-in **any** or **all**	Puppet >= 5.2.0
array_size	Puppet built-in **length**	Puppet >= 5.5.0
array_union	Puppet language **+** operator combined with Puppet built-in **unique**	Puppet >= 5.0.0
bracketize	**simplib::bracketize**	simp-simplib >= 3.15.0
generate_reboot_msg	None	N/A
get_ports	None	N/A
h2n	None	N/A
host_is_me	**simplib::host_is_me**	simp-simplib >= 3.15.0

Table 2 – continued from previous page

Puppet 3 API Function	Replacement	Replacement Source
inspect	**simplib::inspect**	simp-simplib >= 3.3.0
ipaddresses	**simplib::ipaddresses**	simp-simplib >= 3.5.0
ip_is_me	**simplib::host_is_me** (checks hostnames and IP addresses)	simp-simplib >= 3.15.0
ip_to_cron	**simplib::ip_to_cron**	simp-simplib >= 3.5.0
join_mount_opts	**simplib::join_mount_opts**	simp-simplib >= 3.8.0
localuser	None	N/A
mapval	None	N/A
nets2cidr	**simplib::nets2cidr**	simp-simplib >= 3.7.0
nets2ddq	**simplib::nets2ddq**	simp-simplib >= 3.8.0
parse_hosts	**simplib::parse_hosts**	simp-simplib >= 3.5.0
passgen	**simplib::passgen**	simp-simplib >= 3.5.0
rand_cron	**simplib::rand_cron**	simp-simplib >= 3.5.0
simp_version	**simplib::simp_version**	simp-simplib >= 3.15.0
simplib_deprecation	**simplib::deprecation**	simp-simplib >= 3.5.0
slice_array slice_array	Puppet built-in **slice** Puppet built-in **slice**	Puppet >= 4.0.0 Puppe
strip_ports	**simplib::strip_ports**	simp-simplib >= 3.5.0
to_integer	Puppet built-in **Integer** *or* **simplib::to_integer**	**Integer**: Puppet >= 4.0
to_string	Puppet built-in **String** *or* **simplib::to_string**	**String**: Puppet >= 4.0.
validate_array_member	**simplib::validate_array_member**	simp-simplib >= 3.8.0
validate_array_of_hashes	Use a custom Puppet data type such as **Array[Hash]**	Puppet >= 4.0.0
validate_between	**simplib::validate_between**	simp-simplib >= 3.8.0
validate_bool_simp	Use Puppet **Boolean** data type *or* **simplib::validate_bool**	Puppet: >= 4.0.0; simp
validate_deep_hash	**simplib::validate_deep_hash**	simp-simplib >= 3.8.0
validate_float	Use Puppet **Float** data type *or* a check using **is_float** from **puppetlabs-stdlib**	Puppet: >= 4.0.0; **is_fl**
validate_macaddress	Use **Simplib::Macaddress** data type	simp-simplib >= 3.7.0
validate_net_list	Use **Simplib::Netlist** data type *or* **simplib::validate_net_list**	simp-simplib >= 3.5.0
validate_port	Use **Simplib::Port** data type *or* **simplib::validate_net_list**	simp-simplib >= 3.5.0
validate_re_array	**simplib::validate_re_array**	simp-simplib >= 3.7.0
validate_sysctl_value	**simplib::validate_sysctl_value**	simp-simplib >= 3.7.0
validate_umask	Use **Simplib::Umask** data type	simp-simplib >= 3.7.0
validate_uri_list	**simplib::validate_sysctl_value**	simp-simplib >= 3.7.0

simp-ssh Deprecated Functions

Puppet 3 API Function	Replacement	Replacement Source
ssh_autokey	**ssh::autokey**	simp-ssh >= 6.2.0
ssh_global_known_hosts	**ssh::global_know-hosts**	simp-ssh >= 6.2.0

Other important changes for SIMP 6.4.0

- *Non-breaking dependent module updates.* SIMP updated as many dependent modules as possible. While numerous of these updates were major version bumps, the actual changes did not affect much of the SIMP infrastructure. The dependency version bumps did, however, require quite a few of the SIMP modules to update their respective `metadata.json` files. These metadata changes, in turn, required SIMP module version updates.
- *Non-breaking use of new* `Simplib::Cron::*` *types.* Many modules were converted to use the new `cron` types provided by `simp-simplib`. These changes allow more flexible `cron` scheduling.
- *Puppet 6 support for many of the modules.*

- *Added v2* `compliance_markup` *data to many of the modules.*
- *Increase in the minimum RAM requirement for a SIMP server.* It is now 3.4G. See *System Requirements* for more information.

Security Announcements

None at this time.

RPM Updates

rubygem-simp-cli 5.0.1

- Added `simp puppetfile generate`, `simp environment new` and `simp environment fix` commands.
- Changed the environment created by `simp config` to be `production`, not `simp` linked to `production`. The link is not appropriate for sites that use r10K or Code Manager.
- Changed `simp config` to create a new `production` SIMP omni-environment from the new environment skeletons installed in `/usr/share/simp/environment-skeleton`. This new environment includes Puppetfiles `simp config` used to deploy the modules into the environment.
- Reworked `simp config` queries to allow users to opt-out of SIMP-provided LDAP capabilities. Since this change affects the keys in the answers file, be sure to regenerate any answers files you routinely input to `simp config`.
- Added a `--force-config` option to `simp config` to minimize unexpected modification of an active Puppet environment. This option *must* be used when the user runs `simp config` and the `production` environment has already been populated with modules.
- Improved some documentation, logging, and validation in `simp config` and `simp bootstrap`.

simp-adapter 1.0.1

Beginning with `simp-adapter` 1.0.0, the (optional) auto-update to the `simp` Puppet environment has been replaced with creation/maintenance of a local Git repository for each Puppet module that SIMP packages as an RPM. The `simp_adapter`'s `simp_rpm_helper` now ensures that each Puppet module is imported from its RPM installation location, `/usr/share/simp/modules/<module name>`, into a local, SIMP-managed, Git repository, `/usr/share/simp/git/puppet_modules/<owner>-<module name>`. The name of the repository is the top-level `name` field from the module's `metadata.json`.

The specific behavior of `simp_rpm_helper` during RPM operations is as follows:

- Upon module RPM install/upgrade/downgrade, the `simp_rpm_helper`
  - Updates the master branch of the repository to be the contents of the RPM, excluding any empty directories.
  - Adds a Git tag to the repository that matches the version number in the module's `metadata.json` file, as necessary. If the tag for the version already exists but doesn't match the contents of the RPM, `simp_rpm_helper` will **overwrite** the tag with the correct content.
- Upon module RPM erase, the `simp_rpm_helper` does **NOT** remove the local module Git repo, but leaves it intact, in case it is still being used by r10K or Code Manager for an active Puppet environment.

In addition to the change in `simp_rpm_helper` behavior, `simp-adapter` 1.0.1 has three other significant packaging changes:

- It no longer has dependencies upon a specific version of `puppet-agent`, just a minimum version.

- It no longer has any dependency on the Puppet Enterprise or FOSS versions of `puppetserver`.
- Since it now works for both Puppet Enterprise and the FOSS editions of Puppet, `simp-adapter-pe` and `simp-adapter-foss` packages are no longer created.

Puppet RPMs

The following Puppet RPMs are packaged with the SIMP 6.4.0 ISOs:

Package	Version
puppet-agent	5.5.14-1
puppet-client-tools	1.2.6-1
puppetdb	5.2.8-1
puppetdb-termini	5.2.8-1
puppetserver	5.3.8-1

Removed Modules

Elasticsearch-Logstash-Grafana (ELG) Modules

The following modules were removed because they are significantly out of date and, in some cases, only work with application versions that are no longer supported:

- pupmod-elastic-elasticsearch
- pupmod-elastic-logstash
- pupmod-puppet-grafana
- pupmod-simp-simp_elasticsearch
- pupmod-simp-simp_logstash
- pupmod-simp-simp_grafana

The corresponding RPM GPG keys were also removed from `simp-gpgkeys` and the SIMP ISOs:

- RPM-GPG-KEY-elasticsearch
- RPM-GPG-KEY-grafana
- RPM-GPG-KEY-grafana-legacy

pupmod-simp-dirtycow

This module only applies to an old version of CentOS/RHEL 7.

pupmod-simp-simpcat

This module is no longer necessary and uses the obsolete Puppet 3 function API. Within SIMP, all uses of this module have been replaced with uses of the `puppetlabs-concat` module.

pupmod-simp-site

This module skeleton is no longer useful. While it had been useful for some beginning users, the roles and profiles pattern is generally stable and should be used instead.

Fixed Bugs

pupmod-simp-auditd

- The `/etc/audit` directory is now managed in the same way as the `/etc/audit/rules.d` directory. Both will have the same permissions, ownership, and utilize purge and recurse.
- Fixed a bug in which `auditd` failed to start because `space_left` configuration parameter was not ensured to always be larger than the `admin_space_left` configuration parameter.
- Fixed a bug in which `restorecon` was not explicitly audited in the STIG audit profile.
- Fixed a bug where trailing newlines may not be present on custom rule profiles, particularly with rules defined in an Array.

pupmod-simp-compliance_markup

- Fixed a bug in Array merging.
- Remove management of `simp::sssd::client::ldap_domain` from the mappings since use of LDAP is not guaranteed.
- Fixed a bug in the compliance report functionality that did not correctly record the percent compliant in each report summary.
- Fixed a bug in the call to `Hiera.warn` in the compliance mapper.
- Fixed a bug in compliance markup validation for Defined Types.

pupmod-simp-dconf

- Fixed an issue with duplicate resources when creating multiple `dconf::settings` resources under the same namespace.

pupmod-simp-deferred_resources

- Fixed an issue to allow users to set `absent` or `purged` when removing packages, working around a bug in the Puppet `package` provider on EL systems where `yum` is only used if you call `purged`.

pupmod-simp-gdm

- Fixed an issue where `gdm::settings` was not being applied to the system.

pupmod-simp-ima

- Fixed a bug in which the ```ima_security_attr `` fact was not properly confined to require the ``cmdline``` fact.

pupmod-simp-incron

- Fixed a bug in the `incrond_version` fact in which an error message was displayed during fact resolution, on systems for which `incron` was not installed.

pupmod-simp-iptables

- Fixed a bug in which port ranges specified by `iptables::listen::tcp_stateful::dports` or `iptables::listen::udp::dports` could be erroneously split over multiple `iptables` rules.

pupmod-simp-libvirt

- Fixed a bug in which `vm_create` did not work with Puppet > 4.

pupmod-simp-network

- Fixed a bug in which `network::global` could break networking.
- Fixed a bug in which failures of `exec` operations during network-related activity were hidden.

pupmod-simp-ntpd

- Fixed minor typos in some templates.

pupmod-simp-pam

- Fix bug where the ending faillock items were not being called due to the sufficient lines on pam_unix and pam_sssd.
- Fix bug where authsucc was not set at the end of the auth section for faillock.

pupmod-simp-pupmod

- Add missing gem-path setting to puppetserver.conf template.
- Fixed ordering issues with managing the installation of Puppet Server.
- Fixed an issue preventing Puppet from running in environments other than production.
- Fixed a bug where the Java memory maximum size setting on the puppetserver command line, -Xmx, could be empty and prevent the process from starting. This happened when the user specified a fixed RAM size for pupmod::master::sysconfig::java_max_memory, instead of a percentage.

pupmod-simp-postfix

- Fixed unrecognized escape error in /root/.muttrc.

pupmod-simp-rsync

- Fixed a template bug that prevented catalog compilation when rsync::server::section::user_pass was set.

pupmod-simp-simp

- Fixed a bug where the root password field was attempting to set an undef value as Sensitive.
- Fixed a bug in which the the simp_version fact was broken because puppet did not have access to /etc/simp and /etc/simp/simp/version.
- Fixed a bug on EL6 systems in which the puppetdb-dlo-cleanup cron job from the puppetdb module could not be created.
- Excluded the yum class from all Windows nodes.

pupmod-simp-simp_apache

- Fixed a bug in which the /etc/httpd/conf and /etc/httpd/conf.d directories were not being correctly purged.
- Fixed a bug in which the listen configuration parameter could only be specified as a port. It can now be specified as either a port or a host address with port.
- Fixed an issue where the simp_apache::munge_httpd_networks function was modifying passed parameters.

pupmod-simp-simp_gitlab

- Fixed a bug in which both `simp_gitlab` and the `gitlab` application were attempting to manage *CA* certificate hash links.

pupmod-simp-simp_ipa

- Fixed a bug in which `ntp-server` was not passed into the client install.

pupmod-simp-simplib

- Fixed a bug in the signature for the `simplib::ipaddresses` function.
- Fixed a bug in `simplib::nets2ddq` where it would incorrectly try to expand an IPv6 CIDR. It now passes IPv6 addresses through untouched.
- Fixed a bug in which `simp_version` function could cause a GLIBC error in JRuby 9K by using backticks.

pupmod-simp-ssh

- Fixed bug in which the `sshd` 'Subsystem' configuration specified by `ssh::server::conf::subsystem` was erroneously stripped of whitespace.

pupmod-simp-stunnel

- Fixed a bug in which the stunnel `RNDoverwrite` configuration parameter could not be set to `no`.

pupmod-simp-swap

- Fixed a bug in which `/usr/local/sbin/dynamic_swappiness.rb` pointed to system Ruby, which may not be installed, instead of Puppet-provided Ruby, which will always be installed.

pupmod-simp-tcpwrappers

- Fix template bug that prevented some IPv6 addresses from being properly formatted in `/etc/hosts.allow`.

pupmod-simp-tpm2

- Fixed a bug in which the `tmp2` fact was not confined to the presence of the tools required for the fact execution.

pupmod-simp-vsftpd

- Fixed an ordering issue between the kernel module loading for `iptables` and the `vsftpd` service being started.

pupmod-simp-xinetd

- Fixed bug in which the `xinetd::disabled` parameter would only be included in `xinetd.conf`, if the `xinetd::no_access` parameter was not empty.

rubygem-simp-cli

- Fixed a bug in which `simp config` failed to find the template SIMP server host YAML file, `puppet.your.domain.yaml`, from `/usr/share simp/environments/simp`. This bug caused subsequent `simp config` runs to fail, when the SIMP server hostname had changed from the hostname used in the first `simp config` run.
- Fixed bugs in which `simp config` and `simp bootstrap` did not ensure the server had a valid FQDN.
- Fixed a bug in which `simp bootstrap` could fail unless the `puppetserver` was reloaded after the port change to 8140.
- Fixed a bug where the web-routes.conf file was not being overwritten with a pristine copy. This meant that multiple calls to `simp bootstrap` would fail due to leftover CA entries in the file.
- Fixed a bug in log message processing that would cause `simp bootstrap` to fail if it had already been successfully run.
- Fixed a bug in which the tftpboot images copied into the rsync space in SIMP's secondary environment were not world readable, preventing clients from PXE booting.
- Fixed a bug in which `simp config`'s lockfile instructions on how to prevent possible root lockout from the SIMP server were incomplete.

simp-environment-skeleton

- Fixed a bug in which `simp_options::ldap` was to `true` in the `simp` and `simp_lite` scenarios. The use of LDAP is not required, and so these values should have been `false`.
- Fix a regression in which `simp_options::selinux` was inadvertently reinserted into `puppet.your.domain.yaml`.

simp-rsync

- Fixed a bug in which the permissions and owner on `/var/lib/clamav` were incorrect because they were incorrect on `/var/simp/environments/simp/rsync/Global/clamav`.

SIMP ISO

- Fixed a bug in which the generated disk encryption key used to encrypt the disk during a client kickstart was not being stored in the `/boot` partition. This rendered the client unusable, as the disk was effectively permanently locked with an unknown key. The source of the problem was a missing `SYSIMAGE` variable in the sample client kickstart file.

Modules Replacements

The following modules are current and actively maintained replacements for obsolete modules or temporary SIMP versions of modules.

pupmod-camptocamp-systemd

- Replacement for the `simp-systemd` module. SIMP had temporarily published `simp-systemd` while we were waiting for important updates to `camptocamp` to be published.

pupmod-puppet-gitlab

- Replacement for the OBE `vshn-gitlab` module. The Vox Pupuli project has assumed ownership of this module.

pupmod-puppet-snmp

- Replacement for the OBE `razorsedge-snmp` module. The Vox Pupuli project has assumed ownership of this module.

pupmod-saz-timezone

- Replacement for the `simp-timezone` module. SIMP had temporarily published `simp-timezone` while we were waiting for important updates to `saz-timezone` to be published.

pupmod-simp-journald

- Replacement for the OBE `cristifalcas-journald` module.

New Features

pupmod-simp-auditd

- Added a `custom` audit profile that accepts either an Array of rules or a template path for ease of setting full rule sets via Hiera.
- Allow users to optimize their audit processing by only collecting on specific SELinux types.

pupmod-simp-compliance_markup

- Improved migration capabilities of the `compliance_map_migrate` utility:
  - Added ability to merge values from multiple input files.
  - Made `check_header` consistent with other v2 data.
  - Reordered output to match other v2 data.
  - Fixed controls, oval-ids, and identifiers output.
  - Normalized identifier strings.
  - Added an option to supply confinement.
  - Added capability to avoid anchors in the output.
  - Added an option to append a string to the checks key.
  - Added additional helper scripts for v1 to v2 migration.
- Made the 'timestamp' in the client-side report optional to prevent puppet from triggering file resource changes every time.
- Removed experimental v1 data since it is no longer used.
- Removed partial v2 data sets.
- Added v2 data for the non-SIMP `puppet-yum` module.
- Added confinement on modules and facts to SIMP Compliance Engine.

pupmod-simp-deferred_resources

- Added `deferred_resources::files`, `deferred_resources::users`, and `deferred_resources::groups`.
- Added `override_existing_attributes` capability to the `deferred_resources` native type to allow file attributes to be overridden with more useful messages than what you get with resource collectors
- Ensure that an override attribute is defined prior to removal.
- Defined deep merge strategies for the Hash and Array class arguments.
- Switched all FIPS checks to use top level inherited checks.

pupmod-simp-freeradius

- Refactored the API for ease of use and for Puppet 5 compatibility.
- Created an `ldap` site and module to be used to integrate `freeradius` with an LDAP server.
- Added a failsafe for FIPS mode. The module now fails compilation if the server is in FIPS mode, because `freeradius` does not run when FIPS is enabled.
- Removed support for freeradius version 2 and CentOS 6.
- Removed addition of inter-tunnel and SQL sites by default.

pupmod-simp-hirs_provisioner

- New module to install the *HIRS* Provisioner and check-in with the Attestation Certificate Authority.

pupmod-simp-iptables

- Added a rule to allow outbound communication over OUTPUT to the loopback device by default.
- Added a rule to drop 127.0.0.0/8 addresses as defined in RFC 1122 - Section: 3.2.1.3(g). This will exclude 127.0.0.1 as it is allowed in an earlier rule.
- Added `iptables_default_policy` for modifying the default policy of the `filter` table on either IPv4 or IPv6.
- Added `iptables::rules::default_drop` to allow users to easily toggle the default drop behavior of the default filter policies.
- Allow the `proto` key for and entry in `iptables::ports` to be an array.

pupmod-simp-issue

- Added a `source` parameter which can be used to specify a file resource to use for the banner content.

pupmod-simp-journald

- Forked from `cristifalcas-puppet-journald`.
- Refactored the API for ease of use and for Puppet 5 compatibility.

pupmod-simp-network

- Added experimental `NetworkManager` support.

pupmod-simp-ntpd

- Ensure that users can change the default `restrict` parameters.
- Allow users to add arbitrary extra content to the `ntpd.conf`.
- Allow users to define the entire `ntpd.conf` content.
- Restrict `ntpd::allow::rules` to new `Ntpd::Restrict` data type.
- Added ability for users to enable a default server capability directly from Hiera.

pupmod-simp-oath

- New module to install oathtool, a command line utility for generating one-time passwords, and, optionally, install and configure the pam_oath and liboath packages.

pupmod-simp-pam

- Added pam::access::access_file_mode parameter to allow users to manage access.conf file permissions.
- Added *TOTP* support through pam_oath.
- Added option to allow users to disable faillock if desired.

pupmod-simp-pki

- Updated the pki_cert_sync type to also generate an aggregate CA certificates file with X.509 certificate headers removed, cacerts_no_headers.pem. This new CA certificates file is useful for applications that cannot handle certificate headers.
- Add a new option to the pki_cert_sync type, generate_pem_hash_links, which controls whether pki_cert_sync should generate and maintain the PEM file hash links in the target directory. Disabling automatic PEM hash link generation is useful if an application is managing those links in the target directory, itself.

pupmod-simp-pupmod

- No longer hardcode the puppet uid and puppet gid to 52.
- Added management of ssldir and rundir.
- Ensure that the puppet client environment is set to that which is provided by the server by default.
- Added a fact to list all the jruby jar files in the puppetserver installation directory.
- Allow user to change the jar file used to run puppetserver via pupmod::master::sysconfig::jruby_jar.
- Change the default jar file for puppetserver to jruby-9k.jar for the FOSS puppetserver. (It is already set to that for PE.)
- Added a pupmod::server_distribution function for more accurately determining the version of the puppet server installed on the target system.
- Refactored some of the underlying code to use the new function and be safer for PE installations.
- Added Certificate Authority parameters for authorization extensions and subject alternative names.

pupmod-simp-resolv

- Deprecated the 'spoof' option in `/etc/host.conf` since it has not done anything in recent history: https://bugzilla.redhat.com/show_bug.cgi?id=1577265

pupmod-simp-rkhunter

- New module to install and configure `rkhunter` and `unhide` for EL7 hosts.

pupmod-simp-rsync

- Added the ability to have a server support being both a SIMP rsync server and a SIMP rsync client of another server, simultaneously. This configuration is useful for a PE Compile Master <-> Master of Masters host configuration scenario.
- Added a `preserve_perms` attribute to disable preservation of server-side permissions.

pupmod-simp-rsyslog

- Changed default `rsyslog::rule::local::file_create_mode` to `0640`.
- Changed default `rsyslog::rule::local::dir_create_mode` to `0750`.
- Allow the following to be set directly via Hiera using `simplib::dlookup`:
  - `rsyslog::rule::local::file_create_mode`
  - `rsyslog::rule::local::dir_create_mode`
  - `rsyslog::rule::local::queue_max_file_size`

pupmod-simp-selinux

- Added `selinux::kernel_enforce` for toggling the enforcement of the selinux state at the kernel command line.
- Ensure that the `selinux::login_resources` Hash performs a deep merge by default.

pupmod-simp-simp

- Added SELinux login context management to `simp::admin`.
- Refactored the `simp::mountpoints::tmp` to use `systemd`'s `tmp.mount` target, if the system supports `systemd`.
- Added `net.ipv6.conf.all.accept_ra` and `net__ipv4__conf__default__log_martians` to `simp::sysctl` management.
- Added exceptions to SIMP's management of `filebucket` and `vardir` to support running puppet from Bolt.
- Switched out `chkrootkit` for `rkhunter` on El7 instances.
- Added the, inert by default, `deferred_resources` class to all class lists, to allow users to easily use this functionality to meet specific compliance requirements.

- Deprecated `simp::puppetdb::read_database_ssl`. Instead, use `simp::puppetdb::read_database_jdbc_ssl_propert` which maps directly to `puppetdb::server::read_database_jdbc_ssl_properties` (puppetdb version >= 7.0.0).
- Add the ability to set the root user password in `simp::root_user`
- Added a `sysctl` value to increase max number of `inotify` user watches.
- Added 2 configuration parameters to `simp::puppetdb`: `simp::puppetdb::automatic_dlo_cleanup` and `simp::puppetdb::dlo_max_age`.
- Update the URLS to the EPEL GPG keys. The URLS have changed.

pupmod-simp-simp_apache

- Split out service management into a `simp_apache::service` class and provided users with a Hiera-driveable option to not manage the service.

pupmod-simp-simp_bolt

- New module that provides a profile that allows Puppet *Bolt* to operate in a SIMP environment.

pupmod-simp-simp_gitlab

- Switched from `vshn-gitlab` 1.13.3 to its replacement `puppet-gitlab` 3.0.2 and reworked the manifests, as appropriate, to work with the newer module.
- Added `manage_package` and `package_ensure` parameters to `simp_gitlab`.

pupmod-simp-simp_grub

- New module that provides a Hiera-friendly interface to GRUB configuration activities. It currently supports setting administrative GRUB passwords on both GRUB 2 and legacy GRUB systems.

pupmod-simp-simp_ipa

- Added Puppet Tasks for joining and leaving an IPA domain.

pupmod-simp-simp_pki_service

- Modified API. These are minor breaking changes for this **experimental** module.

pupmod-simp-simplib

- Added a `simplib__sshd_config` fact to check the contents of the `sshd_config` file.
- Added a `simplib_networkmanager` fact to provide useful information about the state of NetworkManager components.
- Added a `simplib::in_bolt` function to detect if the current puppet run is happening during a Bolt run.
- Added a set of `Simplib::Cron::####` data types for specifying minute, hour, month, monthday, and weekday parameters for the `cron` resource.
- Removed `simplib`'s `deep_merge` 3.x function that conflicts with `stdlib`'s fully-equivalent `deep_merge` function.
- Removed `simplib`'s `validate_integer` 3.x function that conflicts with `stdlib`'s fully-equivalent `validate_integer` function.
- Converted the following Puppet 3 functions to Puppet 4:
  - `simplib::simp_version` replaces deprecated `simp_version`.
  - `simplib::host_is_me` replaces deprecated `host_is_me`.
  - `simplib::bracketize` replaces deprecated `bracketize`.
- Added a capability for `simplib::bracketize` to accept a string delimited by commas, spaces, and/or semicolons.
- Deprecated the remaining `simplib` Puppet 3.x functions and re-enabled `simplib` deprecation warnings by default.
- Defer to inbuilt `fips_enabled` fact if it exists.
- Added a check for the environment variable, `SIMPLIB_NOLOG_DEPRECATIONS`, to disable SIMP function deprecation warnings.

pupmod-simp-ssh

- Switched to `selinux_port` type for alternate SSH ports
- Added the ability for users to set custom sshd config entries via a Hash in Hiera.
- Made `ListenAddress` optional and documented the corresponding EL6 bug.
- Added *OATH* support
- Added support for the following SSH server configuration parameters:
  - `AllowGroups`
  - `AllowUsers`
  - `DenyGroups`
  - `DenyUsers`
  - `LoginGraceTime`
  - `LogLevel`
  - `MaxAuthTries`
- Added a new class, `ssh::authorized_keys`, that allows users to specify public keys in Hiera.
- Added multiple port support

  - The ssh::server::conf::port entry can now take an Array of ports
  - selinux_port resources are created for each non-standard entry

pupmod-simp-sssd

- Change the sssd::provider::ldap::ldap_access_order defaults to ['ppolicy', 'pwd_expire_policy_renew'] to prevent accidental user lockout.

pupmod-simp-sudo

- Allow additional options to be specified by a Hash in sudo::user_specification.

pupmod-simp-xinetd

- Added a capability to purge unknown xinetd services. This capability is similar to that of svckill, but for the xinetd subsystem.

simp-environment-skeleton

- Renamed the package from simp-environment to simp-environment-skeleton to more accurately portray its purpose.
- Moved all SELinux components to a new package, simp-selinux-policy, and then added a dependency on that package.
- Changed the install location to /usr/share/simp/environments-skeleton for all files. The files are now located within a puppet or secondary sub-directory.
- Removed use of the simp_rpm_helper script during the post-install, so that the potentially active /var/simp/environments/simp directory is no longer modified during initial install or erase.
- Removed the execution of SELinux fixfiles on /var.
- Removed creation of cacertkey in /var/simp/environments/simp/FakeCA.
- Replaced the sample environment.conf file with a template, environment.conf.template.

simp-rsync

This package has been deprecated. Its final release provides a stub with to ensure any files it delivered into /var/simp/environments/simp/rsync, a potentially active secondary environment, are not erased. You can remove this package if your site is not using a simp environment.

simp-rsync-skeleton

- Replacement moving forward for deprecated simp-rsync package.
- All files are now installed in /usr/share/simp/environments-skeleton/rsync.

simp-selinux-policy

New package containing policies originally packaged in the simp-environment package.

simp-utils

- Added simpenv script to facilitate creation of a new SIMP omni-environment when simp CLI cannot be used.
- Updated the updaterepo script to change permissions on the repo files as well as repodata.
- Updated the unpack_dvd script as follows:
  - Added an option to allow user to not link the extracted files to the major version.
  - Added an option to change what group is used to own the files.
  - Ensure permissions on all directories containing RPMs for the repo are correct.
  - Only attempt to change ownership of files if run as root.
  - Put noarch RPMs under the noarch directory for the SIMP repo.
  - Allow the user to specify the version directory for the OS, because the CentOS .treeinfo file only contains the major OS version number.
  - Use Puppet Ruby instead of system Ruby.

SIMP ISO

- Appropriate GPG keys delivered with simp-gpgkeys package are now imported into the rpm database during installation.
- The initial password validation settings now match the defaults provided the simp-pam module.

Known Bugs

Nothing significant at this time.

The SIMP project in JIRA can be used to file bugs.

4.2.4 SIMP Community Edition (CE) 6.3.3-0

Contents

Warning: Please see the *SIMP Community Edition (CE) 6.2.0-0* Changelog for general information, upgrade guidance, and compatibility notes.

This is a bug fix release in the 6.3.X series of SIMP to address the following issues:

- SIMP-6152: Change a new default that was introduced by a bug fix in *SIMP Community Edition (CE) 6.3.2-0* to SSSD that caused accounts with old `shadowLastChange` entries in LDAP to be unable to login to systems.

Fixed Bugs

pupmod-simp-sssd

- Change the `sssd::provider::ldap::ldap_access_order` defaults to `['ppolicy', 'pwd_expire_policy_renew']` by default to prevent accidental system lockouts on upgrade.

Known Bugs

Upgrading from previous SIMP 6.X versions

There are known issues when upgrading from Puppet 4 to Puppet 5. Make sure you read the *Upgrading SIMP* before attempting an upgrade.

Tlog

Tlog currently has a bug where session information may not be logged. The immediate mitigation to this is the fact that *pam_tty_audit* is the primary mode of auditing with `tlog` and/or `sudosh` being in place for a better overall tracking and behavior analysis experience.

Tlog has a second bug where the application fails if a user does not have a TTY. This has been mitigated by the SIMP wrapper script simply bypassing `tlog` if a TTY is not present.

4.2.5 SIMP Community Edition (CE) 6.3.2-0

Contents

Warning: Please see the *SIMP Community Edition (CE) 6.2.0-0* Changelog for general information, upgrade guidance, and compatibility notes.

This is a bug fix release in the 6.3.X series of SIMP to address the following issues:

- SIMP-5974: Ensure that the `incron` spawned `puppet generate types` would not overwhelm the puppet server due to an upstream bug in the incron package. This involved both pinning the incron version to a version that did not have bugs as well as reducing the footprint of the monitored files in the filesystem. See *When should I run puppet generate types?* for additional information.
  - The version of `incron` that shipped with SIMP 6.3.0 did not have issues, but the update in upstream EPEL did and affects **all uses of incron**, not just `pupmod::master::generate_types`. We strongly advise that you remove the `0.5.12-6` package from your upstream repositories and use the following Hiera configuration to ensure that your SIMP `6.3.0-0` installation does not upgrade.

    ```
    ---
    yum::config_options:
      exclude="incron"
    ```

    Warning: If you previously disabled `pupmod::master::generate_types` then be advised that you will need to manually run `puppet generate types` on your environments if you upgrade the `puppet` or `puppetserver` packages or if you add a new environment to your system.

    See the *When should I run puppet generate types?* for additional information.

- SIMP-5480: Fix a bug in the default `sssd settings` where the minimum allowed `uid/gid` is now `1` and the maximum allowed `uid/gid` is now `0` to align properly with the `sssd` functionality.
- SIMP-5932: Allow users to specify a timeout for `simp bootstrap` to address slow systems.
- SIMP-5975: Allow users to specify SSL settings for the puppet server.

Fixed Bugs

pupmod-simp-incron

- Add Incron::Mask Data Type denoting valid incron masks
- Added support for new options starting in 0.5.12
  - Automatically strip out options not supported by earlier versions for seamless backward compatibility
- Add ability to set max_open_files ulimit
- Pin incron to 0.5.10 via data in modules since 0.5.12 as currently published in EPEL can cause catastrophic system failure.

pupmod-simp-pupmod

- Fixed issues where a large number of incron watches may overload the system.
  - The module is now extensively tested against large numbers of environments but will still cause load if a large number of environments are created at once.
- Fixed a bug where some SSL settings could not be set in the puppetserver webserver components.
- Added the following *advanced usage* parameters in case users need to set parameters that are not presently managed to work around future issues:
  - pupmod::master::server_webserver_options
  - pupmod::master::ca_webserver_options

pupmod-simp-simplib

- Ensure that IPA fact does not hang indefinitely.
- Added 'defined type' lookup capability, simplib::dlookup that provides a consistent method for retrieving defined type parameters from Hiera in an opt-in manner. (Required for fixing the stunnel bug).
- Fixed YARD documentation issues

pupmod-simp-sssd

- Set the min_id settings across the board to 1 to match the sssd defaults, since they really have nothing to do with the target system's relationship with a centralized authentication service.
- The original setting of the min_id or max_id settings to the login.defs defaults was a bug since, per the man page, this would preclude sssd from recognizing items outside of that range at all. The relevance of the local login.defs settings (system specific) and the sssd settings (global authentication source) are completely irrelevant to one another and should not have been bound together.
- Updated the sssd::provider::ldap_access_order parameter to support the ppolicy related options that were added in sssd 1.14.0.
  - ppolicy
  - pwd_expire_policy_reject
  - pwd_expire_policy_warn

  - pwd_expire_policy_renew
- Added `pwd_expire_policy_reject` to the `sssd::provider::ldap::ldap_access_order` default. This will deny a locked account even it access is being attempted via a SSH key.

pupmod-simp-stunnel

- Add ability for users to override `stunnel::connection` and `stunnel::instance` options either globally or by specific identified instances using the new `simplib::dlookup` function.
- Fixed `stunnel::connection` and `stunnel::instance` bugs:
  - `sni` is not applicable on EL6
  - `retry` is only applicable when `exec` is specified and needed to be translated from a boolean to `yes/no`
  - `session` is only applicable on EL6

rubygem_simp_cli

- Added a `simp bootstrap` option to set the wait time for the puppetserver to start during the bootstrap process.
- Adjusted the help message so that it fits within a 80-character console window.

Known Bugs

Upgrading from previous SIMP 6.X versions

There are known issues when upgrading from Puppet 4 to Puppet 5. Make sure you read the *Upgrading SIMP* before attempting an upgrade.

Tlog

Tlog currently has a bug where session information may not be logged. The immediate mitigation to this is the fact that *pam_tty_audit* is the primary mode of auditing with `tlog` and/or `sudosh` being in place for a better overall tracking and behavior analysis experience.

Tlog has a second bug where the application fails if a user does not have a TTY. This has been mitigated by the SIMP wrapper script simply bypassing `tlog` if a TTY is not present.

4.2.6 SIMP Community Edition (CE) 6.3.0-0

Contents

This release is known to work with:

- CentOS 6.10 x86_64
- CentOS 7.0 1804 x86_64
- OEL 6.10 x86_64
- OEL 7.5 x86_64
- RHEL 6.10 x86_64
- RHEL 7.5 x86_64

The flagship feature for SIMP 6.3.0 is full compatibility with Puppet 5 and Hiera 5.

The versions of Puppet targeted are those delivered with Puppet Enterprise 2018:

- Puppet Server: 5.3.X
- Puppet Agent: 5.5.X

A much wider range of versions is used for unit and acceptance testing. See the `.gitlab_ci.yml` file in each module to see what versions it has been tested against.

> **Warning:** Puppet 4 is no longer supported as of SIMP 6.3. Users can continue with the SIMP 6.2 release and can obtain commercial support if further Puppet 4 support is required.
>
> From this point on, all components are tested against Puppet 5 and Puppet 4 support may be removed from any module as a non-breaking change at any time.

6.3.0-0 Errata

An upstream bug in the incron package, caused the `pupmod::master::generate_types` code to spin into an infinite loop if the `incron` package was updated to `0.5.12-6` as published in *EPEL*.

This bug affects **all uses of incron**, not just `pupmod::master::generate_types`. We strongly advise that you remove the `0.5.12-6` package from your upstream repositories and use the following Hiera configuration to ensure that your SIMP `6.3.0-0` installation does not upgrade.

```yaml
---
yum::config_options:
  exclude="incron"
```

> **Warning:** If you previously disabled `pupmod::master::generate_types` then be advised that you will need to manually run `puppet generate types` on your environments if you upgrade the `puppet` or `puppetserver` packages or if you add a new environment to your system.

See the *When should I run puppet generate types?* for additional information.

Breaking Changes

Upgrading from Puppet 4 and earlier versions of *Hiera* requires some preparation. Please be sure to read *Upgrading SIMP* carefully.

While *Hiera* 5 is fully compatible with Hiera 3, there have been some configuration changes to utilize new capabilities.

- The `/etc/puppetlabs/puppet/hiera.yaml` file, which defines the hierarchy used to search for parameter values, has been moved to the environment level to utilize the ability to have a unique `hiera.yaml` configuration per environment.
- The default data directory has been renamed from `hieradata` to `data` to match Hiera 5 conventions.

You should review the puppet documentation for upgrading to Hiera 5 to learn how to upgrade any custom modules or backends that you have created.

Significant Updates

The following items were removed as dependencies for the `simp` RPM and added as dependencies on the `simp-extras` RPM since they are not used by the default SIMP configuration:

- pupmod-richardc-datacat
- pupmod-simp-autofs
- pupmod-simp-krb5
- pupmod-simp-network

puppet-simp-tlog

Sudosh has been replaced by *Tlog* as the default for logging privileged user activities. The default command for a user to switch to privileged access is now:

```
sudo su - root
```

Package Installation Settings

Several of the SIMP modules have been updated to use the `simp_options::package_ensure` setting as the default for package resource ensure state. The default for `simp_options::package_ensure` is *installed*. This will change the default behavior of some modules that were previously hard-coded to *latest*. This will not affect anything that was explicitly set.

This change makes the SIMP modules consistent and allows the administrator to set the default across the system with one variable. Also, by setting the default to *installed* packages will be updated only if the administrator has explicitly set the variable to *latest*.

Note: This does **not** affect the nightly cron job that updates all packages on the system and it is recommended that you change this to `latest` and rely on prudent repository management.

See *Nightly Updates* for additional information.

The following modules were updated:

- pupmod-simp-aide
- pupmod-simp-auditd
- pupmod-simp-clamav
- pupmod-simp-dhcp
- pupmod-simp-fips
- pupmod-simp-iptables
- pupmod-simp-krb5
- pupmod-simp-mozilla
- pupmod-simp-oddjob
- pupmod-simp-openscap
- pupmod-simp-rsync
- pupmod-simp-rsyslog
- pupmod-simp-simp_apache
- pupmod-simp-simp_nfs
- pupmod-simp-simp_openldap
- pupmod-simp-ssh
- pupmod-simp-sudo
- pupmod-simp-sudosh
- pupmod-simp-tcpwrappers
- pupmod-simp-tuned
- pupmod-simp-vnc
- pupmod-simp-vsftpd
- pupmod-simp-xinetd

Oracle Enterprise Linux

The testing of the modules on Oracle Enterprise Linux was expanded and automated.

RPM Updates

ELG Stack

The application RPMs for *Elasticsearch*, *Logstash*, and *Grafana* will no longer be delivered with the SIMP ISO. Updates in the same major version of Elasticsearch and Logstash have been shown to randomly corrupt data and are therefore too dangerous to potentially drop into upstream repositories by default. Users must now download their own *ELG* packages.

Removed Modules

pupmod-simp-freeradius

There was not enough time to get the `freeradius` components updated sufficiently for Puppet 5 prior to release. This module may reappear in future releases if there is significant demand.

Security Announcements

None

Fixed Bugs

pupmod-simp-auditd

- Reverted back to using the native service provider for the auditd service
- Allowed users to opt-out of hooking the audit dispatchers into the SIMP rsyslog module using *auditd::config::audisp::syslog::rsyslog = false* or, alternatively, setting *simp_options::syslog = false*.
- Added a *write_logs* option to the *auditd_class* and multiplex between the *log_format = NOLOG* setting and *write_logs = false* since there were breaking changes in these settings after *auditd* version *2.6.0*.
- Added support for *log_format = ENHANCED* for *auditd* version >= *2.6.0*. Older versions will simply fall back to *RAW*.
- Removed unnecessary dependencies from metadata.json. Now, when users install auditd stand-alone i.e. *puppet module install*, they will not have extraneous modules clutter their environment.

pupmod-simp-nfs

- Allowed users to set the 'ensure' state of their client mount points in case they don't want them to be mounted by default.

pupmod-simp-rsyslogd

- Updated templates to use RainerScript for rsyslogd V8 and later
- Fixed the MainMsgQueueDiscardMark and MainMsgQueueWorkerThreads parameters
- Updated rsyslog::rule::remote to select a more intelligent default for StreamDriverPermittedPeers when TLS is enabled. This improvement fixes the bug in which forwarding of logs to servers in different domains was not possible within one call.
- Added logic to properly handle rsyslogd parameters for V8.6 and later as documented in CentOS 7.5 Release notes. These include moving -x and -w options to global.conf and issuing deprecation warning for -l and -s options.

pupmod-simp-simp_grafana

- Fixed bug in resource ordering of pki::copy and grafana::service
- Used simplib::passgen() in lieu of deprecated passgen()

pupmod-simp-simp_logstash

- Workaround for upstream bug where OEL6 logstash::service_provider must be set.

pupmod-simp-simp_rsyslog

- Made directory where logs are gathered configurable and make rules that organize them configurable.
- Updated simp_rsyslog::forward to allow configuration of the StreamDriverPermittedPeers directive in the forwarding rule actions for the remote rsyslog servers. This allows the user to set the correct StreamDriverPermittedPeers value when the default value is incorrect (e.g., when IP addresses are used in simp_rsyslog::log_servers or simp_rsyslog::failover_servers and one or more of those servers is not in the same domain as the client).
- Removed redundant rules for sudosh since the puppet module will correctly take care of adding those rules.
- Added support for tlog since it will be commonly replacing sudosh across the SIMP infrastructure.

pupmod-simp-simplib

- Fixed bug where uid_min would throw errors under operating systems without /etc/login.defs.
- Fixed bug where simplib_sysctl would throw an undefined method error on non-Linux OS's. (both those with sysctl (MacOS X) and without (Windows))
- Fixed bug with the *boot_dir_uuid* fact where it would throw an error if running on a system without a */boot* partition (like a container).
- Ensure that reboot_notify updates resources based on a modified 'reason'

pupmod-simp-ssh

- Hardened all ssh_host_* keys for security and compliance

pupmod-simp-sudo

- Enabled support for Default of *cmnd* type in sudoers file.

pupmod-simp-svckill

- Added 7.5 RHEL services to svckill::ignore_defaults list for EL7.

rubygem_simp_cli

- Updated 'simp config' to support environment-specific *Hiera* 5 configuration provided by SIMP-6.3.0.
  - Assumes a legacy Hiera 3 configuration, when the 'simp' environment only contains a 'hieradata' directory.
  - Assumes a Hiera 5 configuration configuration, when the 'simp' environment contains both a 'hiera.yaml' file and a 'data/'
- Fixed *simp bootstrap* errors in puppetserver 5+:
  - No longer overwrites *web-routes.conf* (fix fatal config errors)
  - No longer adds *-XX:MaxPermSize* for Java >= 8 (fix warnings)
- The *trusted_server_facts* was removed in Puppet 5.0.0. The presence of this setting will cause each puppet run to emit the warning:

  > Warning: Setting trusted_server_facts is deprecated.

  This patch causes *simp config* to quietly remove the setting if it is present and Puppet is version 5 or later.

New Features

pupmod-simp-x2go and pupmod-simp-mate

These modules are used to configure the x2go client and server to allow for remote access to desktops and servers. This is an alternative to VNC. An example configuration is documented in the *Graphical Remote Access* documentation.

pupmod-simp-tlog

This module configures *Tlog* for logging privileged user activities. Both *sudosh* and Tlog are currently available but sudosh is no longer being maintained and is expected to be deprecated in the future.

pupmod-simp-simp_pki_service

Warning: This is a technology preview and may break unexpectedly in the future

Traditionally, SIMP has used an internal "FakeCA" *openssl*-based CA. Over time, this has proven insufficient for our needs, particularly for capabilities in terms of Key Enrollment (SCEP and CMC), OCSP, and overall management of certificates.

Additionally, it was found that users wanted to adjust the certificate parameters for the Puppet subsystem itself outside of the defaults and/or use a "real", and more scalable CA system for all certificate management.

The pupmod-simp-simp_pki_service module can be used to configure a Certificate Authority (CA) using the Dogtag server. This CA can be configured either for the puppet server CA, the site CA in lieu of FakeCA, or both.

See the README in the module for details on how to configure it.

The Dogtag server was chosen because it is part of the FreeIPA suite and therefore likely to have any issues fixed and be well supported.

Known Bugs

Upgrading from previous SIMP 6.X versions

There are known issues when upgrading from Puppet 4 to Puppet 5. Make sure you read the *Upgrading SIMP* before attempting an upgrade.

Tlog

Tlog currently has a bug where session information may not be logged. The immediate mitigation to this is the fact that *pam_tty_audit* is the primary mode of auditing with `tlog` and/or `sudosh` being in place for a better overall tracking and behavior analysis experience.

Tlog has a second bug where the application fails if a user does not have a TTY. This has been mitigated by the SIMP wrapper script simply bypassing `tlog` if a TTY is not present.

4.2.7 SIMP Community Edition (CE) 6.2.0-0

Contents

This release is known to work with:

- RHEL 6.9 x86_64
- RHEL 7.4 x86_64
- CentOS 6.9 x86_64
- CentOS 7.0 1708 x86_64

Note: SIMP CE is expected to migrate to Puppet 5 on, or before, October 30 2018. We have not noticed any issues with the latest versions of Puppet 5 but it is taking time to get all of our tests updated to work with Puppet 5 for full coverage.

At this point, all vendor support for Puppet 4 will be discontinued as will SIMP CE support for Puppet prior to 4.10.4.

SIMP CE will no longer provide any support for Puppet 4 after June 30 2019.

Breaking Changes

Warning: This release of SIMP CE is **NOT** backwards compatible with the 4.X and 5.X releases. **Direct upgrades will not work!**

At this point, do not expect any of our code moving forward to work with Puppet 3.

If you find any issues, please file bugs!

Significant Updates

Warning: Due to various issues with earlier releases of Puppet, SIMP CE will now be shipping with, and supporting, puppet 4.10.4+.

It is strongly recommended that users upgrade their system as soon as they are able.

Note: SIMP will begin supporting Hiera v5 out of the box as of SIMP 6.3. This is mainly to facilitate compliance enforcement in the infrastructure since various versions of Puppet 4 do not work properly with Hiera v3 and enforcement.

No changes will be made to existing configurations but compliance enforcement from the `compliance_markup` module will not work until an upgrade to Hiera v5 is complete.

- UEFI systems should now be fully supported. Please note that you may need to adjust your `tftpboot` settings to handle your specific UEFI system since they are not as universal as the legacy BIOS entries.
- Many module updates simply added support for Puppet 5 and Oracle Enterprise Linux. These changes will not be listed individually below.

- Likewise, many modules were updated simply to improve tests. These improvements will also not be noted below.
- The simp_gitlab module no longer supports EL6. This is due to integration issues with GitLab that cannot be readily fixed by the module maintenance team, alone. The EL community had shown no interest in fixing minor issues with EL6 in the GitLab platform.

Security Announcements

RPM Updates

- Added the toml rubygem as an RPM for use with the elasticsearch modules.
- Updated to the latest 5.X release of Elasticsearch and Logstash
- Updated the ClamAV packages to 0.100.0-2
- Removed clamav-data-empty which is no longer used

Removed Modules

pupmod-simp-mcollective and pupmod-simp-activemq

- Puppetlabs no longer supports MCollective, so SIMP has removed the pupmod-simp-mcollective and pupmod-simp-activemq modules that support MCollective.

pupmod-simp-jenkins

- The jenkins module has not been updated in quite some time and it is unknown if it works with current versions of Jenkins since the team has moved to GitLab CI.

pupmod-simp-mcafee

- This module has not been updated and probably does not work with the latest McAfee products so it has been removed from the distribution.

pupmod-puppetlabs-java_ks

- This RPM has been removed, as it is no longer a dependency of any SIMP modules.

Security Updates

- The PKI certificates in /etc/pki/simp_apps are now purged by default so that unmanaged certificates are not available if the system is repurposed.

Fixed Bugs

pupmod-simp-aide

- Added /etc/logrotate.simp.d to default rules.
- Ensure that the `package` install comes before dependent `exec` statements.
- Allow the `cron` command to be customized.

pupmod-simp-compliance_markup

- Fixed several incorrectly typed parameters
- Consolidated several duplicate entries
- Added missing `IPT:` message start to `simp_rsyslog::default_logs`
- Synchronized CentOS and RHEL STIG settings

pupmod-simp-incron

- Fixed the permissions on the `incrond` service in `systemd` to remove logged errors.
- Matched RPM permissions based on STIG requirements.

pupmod-simp-iptables

- Updated to match the `ignore` parameter on input and output interfaces
- Fixed `scanblock` rule ordering to properly ban all hosts that are blocked by the rules.
- Fixed some issues in the chain retention and optimization code that would cause `iptables` to fail to reload in some cases.
- Fixed compilation failures if `proto` was specified in the `defaults` section of the options Hash.
- Fixed an issue where a `jump` target went to an empty ruleset and the chain was dropped.
- Retained all native IPTables `jump` points by default.
- Added a *deep rule* comparison on rulesets that are identical based on simple checks.
- Remediated potential memory leaks.
- Fixed ordering issues when used with `firewalld`.
- Matched RPM permissions based on STIG requirements.

pupmod-simp-libvirt

- Ensure idempotency by working around the fact that the modprobe changes - to _.

pupmod-simp-named

- Properly override the systemd service file for named-chroot instead of modifying the vendor provided service file.

pupmod-simp-ntpd

- Fixed a bug where ntpd::ntpd_options was not applied to ntpd::servers when ntpd::servers is an Array

pupmod-simp-pam

- Change the minimum allowed UID to the one defined in /etc/login.defs by default, or 1000 if nothing else is defined.
- Replace the removal of authconfig and authconfig-tui with the use of a authconfig no-op script, so that tools using authconfig do not break.

pupmod-simp-postfix

- Added changes to support the settings required by the STIGs.
- Match the RPM supplied file permissions are required by the STIG.

pupmod-simp-pupmod

- Allow modification of the allow and deny rules for supported keydist auth rules.
- Removed obsolete mcollective auth rules.
- Changed $pki_cacerts_all's auth rule from * to certname.
- Modified the default max_active_instances configuration to be safer by default.
- Make the Puppet Server service name dynamic to work properly with both PE and FOSS Puppet.
- Properly disable the puppet service if running in cron mode. This was not disabled before and could contribute to a "thundering herd" issue.
- Fixed the Java tmpdir path for the puppetserver which allows runs on systems that have been pre-hardened.

pupmod-simp-rsync

- Force `concat` ordering to be `numeric` due to a bug in `puppetlabs-concat` that reverses the order from the native type provided by the same module.

pupmod-simp-rsyslog

- Use double quotes to allow evaluation of line returns in strings.
- Added a `systemd` service override that fixes an ordering problem with older versions of `rsyslog`.
- Fixed bug that did not allow a TLS encrypted server to be configured to forward to a follow-on unencrypted rsyslog server.
- Fixed a bug where removing `rsyslog::rule` statements from the catalog would not cause the `rsyslog` service to restart.
- Clarified documentation around adding files to `/etc/rsyslog.d`.

pupmod-simp-selinux

- `$selinux::ensure` now defaults to `enforcing` and it used across the board instead of `$simp_options::selinux` which never behaved as designed.

pupmod-simp-simp

- Fixed a bug where if the `puppet_settings` fact did not exist, users in the `administrators` group could `rm -rf` any path.
- Fixed the certificate cleaning `sudo` rule to point to `$facts['puppet_settings']['main']['ssldir']`.
- Ensure that `prelink` is fully disabled when the system is in `FIPS` mode since the two are incompatible.
- Defined a `portreserve` service so that there would no longer be any service restart flapping.
- Fixed the permissions on the `ctrl-alt-del-capture` service file so that warnings would no longer be logged.
- Replace the deprecated `runpuppet` script with client Puppet bootstrap scripts which will not be inappropriately killed by `systemd`, when executed in highly-loaded environments. These scripts allow the `systemd` timeout to be specified and provide better error handling and logging.
- On systems with `systemd`, set the host name in client Puppet bootstrap scripts, to prevent issues that can arise when a `dhcp` lease expires. Not setting the hostname could cause the generated Puppet configuration for the client to use `localhost` as the client's hostname.
- Ensure that running on unsupported operating systems is completely safe.
- No longer deviate from vendor RPM default permissions per the STIG.
- Changed the permissions of `rc.local` to `750`.
- Removed the explicit setting of the `host_list` on all `sudo::user_specification` resources to let the updated module defaults handle setting `host_list` appropriately.

pupmod-simp-simp_apache

- Fix the ownership of the configuration files to use the `owner` variable instead of the `group` variable for user ownership.

pupmod-simp-simp_elasticsearch

- Add a missing `simp/pam` module dependency.

pupmod-simp-simp_gitlab

- Fixed the git `authorized_keys` lock problem.
- Dropped all support for CentOS 6 due to issues that kept cropping up during integration and the overall lack of support from EL upstream to fix minor bugs.
- Automatically opt-out of the GitLab data collection service in accordance with NIST 800-53r4 AC-20(1) and SC-38.

pupmod-simp-simp_nfs

- Ensure that users can fully disable `autofs` if they choose to.
- Fixed `systemd` dependencies.

pupmod-simp-simplib

- Fixed the `puppet_settings` fact so that the different sections are appropriately filled out. If not updated, this has been shown to cause the `puppetserver` process to be unable to restart on package update.
- Fixed `runlevel` enforcement so that it activates properly when called. Previously, no action would be taken on the running system.
- Added logic to prevent respawn of systemctl isolate if already in progress.
- Added a configurable timeout for changing runlevels based on issues discovered in the field with systemctl.
- Fixed bugs in the EL6 runlevel persistence where, in some cases, the runlevel line might not be added to /etc/inittab.

pupmod-simp-stunnel

- Fixed the `stunnel` startup scripts to ensure that they will always execute.
- Only display errors when errors occur during startup.
- Removed the `init.d` script on `systemd` systems.
- Ensure that the `stunnel` service name is set correctly in all instances, so that `tcpwrappers` functions properly.

pupmod-simp-svckill

- Add simp_client_bootstrap service to the ignore list; otherwise, svckill will kill the bootstrap process of SIMP clients.

pupmod-simp-vnc

- Fixed issues with the `xinetd` spawned `VNC` sessions where `'IPv4` needed to be set as a flag and the banner needed to be eliminated from the connection.

simp-cli

- Move to the updated OS facts for less fragility.
- Update several messages to be more clear to the user.
- Fix setting GRUB passwords on EL6.
- Fix ownership and permission issues on created files.
- Validate all puppet code present prior to bootstrapping.
- Fixed various logging issues.
- Improved validation and error handling.
- Fix `simp passgen` processing of all password files and improved password generation.
- Properly detect Puppet Enterprise on a system and avoid conflicting operations.
- Fixed some tests that were not safe to run on real operating systems.

simp-core

- Enabled GPG checking for the ISO-configured local filesystem repository by default
- Fixed errors in the `kickstart` scriptlets
- Improved detection of SSD devices using the `diskdetect.sh` script
- Removed obsolete `simp-big` and `simp-big-disk-crypt` kickstart options in EL7
- No longer install `prelink` at kickstart time
- Fixed EFI support on the ISO releases
- Removed EL7 references to function keys which no longer are honored
- Fixed the boot directory when `fips` is enabled on the ISO

simp-doc

- Remove OBE MCollective references
- Fixed issues in the sample `tftpboot` puppet code
- Fixed several broken links
- Made the installation guide more user friendly by rearranging the content

simp-environment

- Added the `dist` macro to the package name
- Pre-populate `/var/simp/environments/simp/site_files/pki_files` and set the permissions appropriately. This fixes the failure of `simp bootstrap` on systems where the `root` user's `umask` has already been set to `077`.
- FakeCA config files were marked as such in the RPM so that they will not be overwritten on RPM upgrade.
- Fixed a bug where the `cacertkey` file was not being generated in the correct location at install time.
- Removed `simp_options::selinux` from the scenario hieradata.
- Force a run of `fixfiles` in the `%post` section of `simp-environment`.

simp-rsync

- Fully support UEFI booting.

New Features

pupmod-simp-compliance_markup

- More closely aligned with the latest SSG STIG content.

pupmod-simp-dconf

- Added a module for managing `dconf` settings.

pupmod-simp-incron

- Allow users to define entries for `incron` system tables from Hiera.
- Added a native type `incron_system_table` to allow for client side path glob expansion.

pupmod-simp-libvirt

- Use kmod::load instead of a Ruby script to load the kernel module
- Added a libvirt_br_netfilter_loaded fact to determine if the br_netfilter kernel module is loaded

pupmod-simp-logrotate

- Moved SIMP-specific logrotate rules to a SIMP-managed configuration directory, /etc/logrotate.simp.d, and ensured logrotate processes that directory first. This ensures that SIMP rules take priority, when duplicate rules are specified (e.g., OS and SIMP rules for /var/log/boot.log).

pupmod-simp-nfs

- Change all stunnel connections to use stunnel::instance to that they are not interrupted due to issues with the global stunnel configuration.
- Added the ability to tweak stunnel parameters for all NFS connections.
- Ensure that all stunnel services used with NFS are now dependencies of the remote filesystem servers actually being active.
- Added the ability to set nfs::client::mount::autodetect_remote to override all autodetection of whether or not the remote system is the local NFS server.
- Added nfs::client::mount::stunnel to allow users to dictate the stunnel state for individual connections.

pupmod-simp-ntpd

- Added optional management of the /etc/ntp/step-tickers file.
- Added a $package_ensure parameter to control the ntp package version.
- Added management of /etc/sysconfig/ntpdate

pupmod-simp-openldap

- Ensure that concat resource ordering is set in numeric order.

pupmod-simp-openscap

- Added an oscap fact to collect the following: * OpenSCAP Version * OpenSCAP Supported Specifications * OpenSCAP Profiles from /usr/share/xml/scap/*/content/*-ds.xml

pupmod-simp-pam

- Added the ability to set unlock_time to never for pam_faillock.so.
- Set the default cracklib_maxclassrepeat to 3.
- Allow users to change the password hashing algorithm.
- Allow users to toggle password enforcement for the root user.

pupmod-simp-pki

- Purge /etc/pki/simp_apps by default to clean up old certificates and allow users to move this directory target.
- Added a new $pki::certname parameter that controls the name of the certificates in keydist that will be copied to the client. This is, by default, set to $trusted['certname'] but can be changed so that users can pull other certificates by default.
- Changed the CA certificate source to be a String so that NSS databases or https endpoints can be specified.

pupmod-simp-pupmod

- Added pupmod::master::generate_types which adds incron hooks that will automatically run puppet generate types on your server when environments or native types are updated in any environment.

pupmod-simp-resolv

- Prevent invalid resolv.conf files from being written.

pupmod-simp-simp

- Remove prelink if it is not enabled.
- Added support for connecting to IPA servers.
- Removed simp::mcollective class due to global deprecation.
- Removed group management for the root user based on feedback.
- Set the ownership and permissions of /etc/puppet/puppetdb.conf so that systems that already have the root umask set to 077 work properly.
- Added a simp::netconsole class to allow users to configure the netconsole kernel parameter for boot time logging.
- Split out the runpuppet logic into a bootstrap_simp_client script to be separate from the startup scripts and work around issues with systemd timeouts.
- Added an exponential backoff to the bootstrap_simp_client script to handle cases where a lot of servers are being built at the same time.
- Added Microsoft Windows support to the module that changes where the simp.version file is placed on that platform.

pupmod-simp-simp_docker

- Multiple minor updates mostly surrounding the updates to `simp/iptables` to make it better work with `docker`.

pupmod-simp-simp_gitlab

- Added support for the new GitLab 10+ LDAP options, specifically for TLS.

pupmod-simp-simp_grafana

- Added documentation regarding `rubygem-puppetserver-toml` for use with the `simp_grafana` module.

pupmod-simp-simp_ipa

- Initial release of a module for managing `IPA` connectivity settings.
- Does not currently manage `IPA` server installation.

pupmod-simp-simp_nfs

- Added the ability to force mounts to point to a remote host.

pupmod-simp-simp_openldap

- Allow users to set the `users` and `administrators` `GID` values in the `default.ldif` file.
- Use concat numeric ordering to allow placement of new modifications in a predictable and reliable order.

pupmod-simp-simp_options

- Added `simp_options::uid` and `simp_options::gid` since several modules required a consistent parameter set for enforcing these items globally.
- Removed `$simp_options::selinux` since it never worked as designed and was not required by more than one module. This is not considered a breaking change since it effectively never had any effect on the system anyway.

pupmod-simp-simplib

- Added a `Simplib::Domain` data type that validates DNS domains against the `TLD` restrictions from RFC 3968, Section 2.
- Added a `login_defs` custom fact that returns a structured fact for the entire contents of `/etc/login.defs`
- Added an `ipa` fact that returns information about connectivity to an `IPA` server.
- Added a `prelink` fact to determine whether or not `prelink` is installed on the system.
- Updated the `simplib::ldap::domain_to_dn` function to allow users to decide whether or not they want to upcase the returned LDAP attribute strings.
- Added a `simplib::reboot_notify` class to allow users to easily toggle global `reboot_notify` settings.

- Improved reboot_notify error handling.
- Allow users to set the log level on reboot_notify.
- Added a Simplib::PuppetLogLevel data type.
- Updated init_ulimit to allow it to work properly with puppet generate types.
- Added a simplib::hash_to_opts function which turns a Hash into a String that mirrors a usual shell command.
- Added a simplib::install defined type that allows package management based on a supplied Hash.
- Added a simplib::module_exist function to detect the existence of a module.
- Ensure that systemctl is never spawned more than once when attempting to change the system runlevel.
- Fixed an issue in EL6 runlevel persistence where the line may not be written to /etc/inittab.

pupmod-simp-ssh

- Ensure that GSSAPIAuthentication is disabled if the host is on an IPA domain.
- Moved all management of the /etc/ssh/ssh_config file to use the ssh_config augeasprovider. Management of all SSH configuration files is now done consistently.
- Removed the no longer required sshd.aug augeas lens.
- Added parameter management to the sshd_config to align with the STIG requirements.
- Default to not configure RhostsRSAAuthentication in sshd_config for versions of openssh that no longer allow that option.

pupmod-simp-sssd

- Updated to use the login_defs fact to determine the default uid_min and uid_max values.
- Added a defined type for connecting to an IPA server.
- Added tests for connecting to Active Directory and updated the configuration settings appropriately.
- Allow passing ldap_tls_cacert to the sssd::provider::ldap defined type.
- Align sssd permissions with the RPM defaults.

pupmod-simp-stunnel

- Isolated the instance logic away from the global connection logic completely.
- Added a native type that cleans up all instances that may have been abandoned by stunnel::instance.
- Added parameters to allow controlling systemd requirement chains.

pupmod-simp-sudo

- Added both the short `hostname` and long `fqdn` to the user access control by default.
- Update user_specification define to not accept an empty hostlist.

pupmod-simp-tftpboot

- Added support for UEFI PXEboot
- Moved the `tftpboot` root directory from `/tftpboot` to `/var/lib/tftpboot` to match the expectations of SELinux and the STIG.
- Added a `tftpboot::tftpboot_root_dir` parameter to all users to override the root directory location.

pupmod-simp-tpm

- Moved the policy `systemd` unit files to `/etc/systemd`
- Ensure that the `IMA` service only starts on reboot instead of during a puppet run.
- Disabled many `IMA` checks by default to make the impact lighter on a standard system.

pupmod-simp-useradd

- Set the min and max `UID` and `GID` based on what is in `login.defs`, and default to something sensible for the platform.

simp-core

- Added logic to auto.cfg to use OS-specific GPG keys in simp_filesystem.repo.
- Client kickstart files were updated to use the latest `simp::server::kickstart` API and to provide support for UEFI PXE boot
- EL6 kickstart files were updated to more closely match the EL7 kickstart files

simp-doc

- Added SIMP 6.1.0 to 6.2.0 upgrade guide
- Added SIMP on AWS documentation
- Added a HOWTO for IPA client enrollment
- Added a HOWTO for customizing settings for SSH
- Added documentation on how to disconnect from `puppetDB`
- Updated the documentation for UEFI PXE booting.
- Clarified certificate management
- Restructured pages for better navigation
- Updated contributors guide to description more details about the development workflow

simp-vendored-r10k

- Added a SIMP vendored version of `r10k` that lives at `/usr/share/simp/bin/r10k` to ensure that a known version of `r10k` is present on the system at all times. User `PATH` environment variables are **not** updated so that command must be called directly.

Known Bugs

- There is a bug in `Facter 3` that causes it to segfault when printing large unsigned integers - FACT-1732
  - This may cause your run to crash if you run `puppet agent -t --debug`
- The `krb5` module may have issues in some cases, validation pending
- The graphical `switch user` functionality appears to work randomly. We are working with the vendor to discover a solution

4.2.8 SIMP 6.1.0-0

Contents

This release is known to work with:

- RHEL 6.9 x86_64
- RHEL 7.4 x86_64
- CentOS 6.9 x86_64
- CentOS 7.0 1708 x86_64

Breaking Changes

> **Warning:** This release of SIMP is **NOT** backwards compatible with the 4.X and 5.X releases. **Direct upgrades will not work!**
>
> At this point, do not expect any of our code moving forward to work with Puppet 3.

If you find any issues, please file bugs!

Breaking Changes Since 6.0.0-0

Upgrade Issues

- You **MUST** read the *Upgrading from SIMP-6.0.0 to SIMP-6.1.0* section of the documentation for this upgrade. There were several RPM issues that require manual intervention for a clean upgrade.
  - The docs can be found at Read The Docs on the internet or under `/usr/share/doc` when the `simp-doc.noarch` RPM is installed.

Significant Updates

Puppetserver Log Issues

You may have noticed that you were not getting `puppetserver` logs recorded either on the file system or via `rsyslog`. We fixed the issue as identified in SIMP-4049 but we cannot safely upgrade existing systems to fix the issue.

To enable log collection via `rsyslog` (the default), you will need to add the following to your puppet server's hieradata:

- `rsyslog::udp_server:  true`
- `rsyslog::udp_listen_address:  '127.0.0.1'`

By default, this file will be located at `/etc/puppetlabs/code/environments/simp/hieradata/hosts/puppet.<your.domain>.yaml`

Puppetserver auth.conf

If you are upgrading from SIMP-6.0.0-0 to a later version:

- The legacy `auth.conf` (`/etc/puppetlabs/puppet/auth.conf`) has been deprecated
- `pupmod-simp-pupmod` will back up legacy puppet `auth.conf` after upgrade
- The puppetserver's `auth.conf` is now managed by Puppet
- You will need to re-produce any custom work done to legacy `auth.conf` in the new `auth.conf`, via the `puppet_authorization::rule` defined type
- The stock rules are managed in `pupmod::master::simp_auth`

No Longer Delivering ClamAV DAT Files

Given the wide spacing of SIMP releases, the team determined that it was ineffective for us to maintain the `simp-rsync-clamav` RPM with upstream ClamAV DAT file updates.

From this point forward, SIMP will not ship with updated ClamAV DAT files and we highly recommend updating your DAT files from the authoritative upstream sources.

SNMP Support Added

We have re-added SNMP support after a thorough re-assessment and update from our legacy `snmp` module. We now build upon a community module and wrap the SIMP-specific components on top of it.

Preparing for Puppet 5

We are in the process of updating all of our modules to include tests for Puppet 5 and, so far, things have gone quite well. Our expectation is that the update to Puppet 5 will be seamless for existing SIMP 6 installations.

Non-Breaking Version Updates

Many modules had dependencies that were updated in a manner that was breaking for the downstream module, but which did not affect the SIMP infrastructure. This caused quite a few of the SIMP modules to have version updates with no changes other than an update to the `metadata.json` file.

In general, this was due to dropping support for Puppet 3.

Long Puppet Compiles with AIDE Database Initialization

In order to expose `aide` database configuration errors during a Puppet compilation, the database initialization is no longer handled as a background process.

When the AIDE database must be initialized, this can extend the time for a Puppet compilation by **several minutes**. At the console the Puppet compilation will appear to pause at `(/Stage[main]/Aide/Exec[update_aide_db])`.

Security Announcements

- CVE-2017-2299
  - Versions of the puppetlabs-apache module prior to 1.11.1 and 2.1.0 make it very easy to accidentally misconfigure TLS trust.
  - SIMP brings in version puppetlabs-apache 2.1.0 to mitigate this issue.

RPM Updates

Package	Old Version	New Version
puppet-agent	1.8.3-1	1.10.6-1
puppet-client-tools	1.1.0-0	1.2.1-1
puppetdb	4.3.0-1	4.4.0-1
puppetdb-termini	4.3.0-1	4.4.0-1
puppetserver	2.7.2-1	2.8.0-1

Removed Modules

pupmod-herculesteam-augeasproviders

- This was a meta-module that simply required all other `augeasproviders_*` modules and was both not in use by the SIMP framework and was causing user confusion.

pupmod-herculesteam-augeasproviders_base

- Has internal bugs and was not in use by any SIMP components

Security Updates

pupmod-puppetlabs-apache

- Updated to 2.1.0 to fix CVE-2017-2299

Fixed Bugs

pupmod-simp-aide

- Fixed a bug where `aide` reports and errors were not being sent to syslog
- Now use FIPS-appropriate Hash algorithms when the system is in FIPS mode
- No longer hide AIDE initialization failures during Puppet runs
- Ensure that `aide` now properly retains the output database in accordance with the STIG checks

pupmod-simp-auditd

- Changed a typo in auditing `faillock` to the correct watch path

pupmod-simp-compliance_markup

- Fixed an issue where a crash would occur when `null` values were in the compliance markup data

pupmod-simp-libreswan

- Fixed issues when running `libreswan` on a FIPS-enabled system

pupmod-simp-logrotate

- Ensure that `nodateext` is set if the `dateext` parameter is set to `false`

pupmod-simp-simp_openldap

- Fixed an issue where `pki::copy` was not correctly hooked into the server service logic. This caused the OpenLDAP server to fail to restart if a new host certificate was placed on the system.
- Fixed an idempotency issue due to an `selinux` context not being set

pupmod-simp-simp_options

- Made some parameter fixes for a bug in Puppet 5 (PUP-8124)

pupmod-simp-pam

- Enable `pam_tty_audit` for `sudo` commands

pupmod-simp-simp

- Changed the `simp::sssd::client::min_id` parameter to `500` from `1000`
  - Having `min_id` at `1000` was causing intermittent retrieval errors for the `administrators` group (and potentially other supplementary groups) that users may be assigned to. This led to the potential of users below `1000` being left unable to log into their system and was reproduced using the stock `administrators` group.
  - The wording of the `sssd.conf` man page for `min_id` leads us to believe that the behavior of non-primary groups may not be well defined.

pupmod-simp-simp_rsyslog

- Ensure that aide and snmp logs are forwarded to remote syslog servers as part of the *security relevant* logs
- Persist aide logs on the remote syslog server in its own directory since the logs can get quite large

pupmod-simp-sssd

- Updated the Sssd::DebugLevel Data Type to handle all variants specified in the sssd.conf man page
- No longer add try_inotify by default since the auto-detection should suffice
- Ensure that an empty sssd::domains Array cannot be passed and set the maximum length to 255 characters

pupmod-simp-stunnel

- Improved the SysV init scripts to be more safe when killing stunnel services
- The stunnel PKI certificates are owned by the correct UID
- Fixed the init scripts for starting stunnel when SELinux was disabled
- Added a systemd unit for EL7+ systems
- Updated the systemd unit files to run stunnel in the foreground

pupmod-simp-svckill

- Fixed a bug in which svckill could fail on servers for which there are no aliased systemd services

simp-core

- Fixed several issues with the ISO build task: rake beaker:suites[rpm_docker]

simp-environment

- Fixed a bug where a relabel of the filesystem would incorrectly change **all** SELinux contexts on any environment files in /var/simp/environments with the exception of the default simp environment.
- Added the following items to the default puppet server hieradata file at /etc/puppetlabs/code/environments/simp/hieradata/hosts/puppet.your.domain.yaml to enable the UDP log server on 127.0.0.1 so that the puppetserver logs can be processed via rsyslog by default.
  - rsyslog::udp_server: true
  - rsyslog::udp_listen_address: '127.0.0.1'

simp-rsync

- Fixed a bug where a relabel of the filesystem would incorrectly change **all** SELinux contexts on any environment files in `/var/simp/environments` with the exception of the default `simp` environment.

New Features

pupmod-camptocamp-systemd

- Added as a SIMP core module

pupmod-vshn-gitlab

- Added as a SIMP extra

pupmod-simp-autofs

- Allow pinning of the `samba` and `autofs` packages to work around bugs in `autofs` that do not allow proper functionality when working with `stunnel`
  - autofs EL6 Beaker Bug Report
  - autofs EL7 Beaker Bug Report

pupmod-simp-clamav

- Added the option to not manage ClamAV data **at all**

pupmod-simp-compliance_markup

- Converted all of the module data to JSON for efficiency

pupmod-simp-krb5

- Allow users to modify the owner, group, and mode of various global kerberos-related files

pupmod-simp-logrotate

- Made the logrotate target directory configurable

pupmod-simp-pam

- Changed `pam_cracklib.so` to `pam_pwquality.so` in EL7 systems

pupmod-simp-pupmod

- Added a SHA256-based option to generate the minute parameter for a client's `puppet agent` cron entry based on its IP Address
  - This option is intended to mitigate the undesirable clustering of client `puppet agent` runs, when the number of IPs to be transformed is less than the minute range over which the randomization is requested (60) and/or the client IPs are not linearly assigned

pupmod-simp-simp_gitlab

- Added as a SIMP extra

pupmod-simp-selinux

- Added a reboot notification on appropriate SELinux state changes
- Ensure that a `/.autorelabel` file is created on appropriate SELinux state changes
  - This capability is *disabled* by default due to issues discovered with the autorelabel process in the operating system

pupmod-simp-simp_snmpd

- Added SNMP support back into SIMP!

pupmod-simp-simplib

- Updated `rand_cron` to allow the use of a SHA256-based algorithm specifically to improve randomization in systems that have non-linear IP address schemes
- Added a `simplib::assert_metadata_os` function that will read the `operatingsystem_support` field of a module's `metadata.json` and fail if the target OS is not in the supported list
  - This can be globally disabled by setting the variable `simplib::assert_metadata::options` to `{ 'enable' => false }`
- Began deprecation of legacy Puppet 3 functions by Puppet 4 counterparts. At this time, no deprecation warnings will be generated but this will change in a later release of SIMP 6.

pupmod-simp-timezone

- Forked `saz/timezone` since our Puppet 4 PR was not reviewed and no other Puppet 4 support seemed forthcoming

pupmod-simp-tpm

- Refactoring and updates to make using the TPM module easier and safer
- Addition of an `instances` feature to the TPM provider so that `puppet resource tpm_ownership` works as expected
- Changed the `owner_pass` to `well-known` by default in `tpm_ownership`
- Removed `ensure` in favor of `owned` in `tpm_ownership`

pupmod-simp-vsftpd

- Change `vsftpd` to use TLS 1.2 instead of TLS 1.0 by default

pupmod-voxpupuli-yum

- Added as a SIMP core module

simp-doc

- A large number of documentation changes and updates have been made
- It is **HIGHLY RECOMMENDED** that you review the new documentation

simp-rsync

- Removed the `simp-rsync-clamav` sub-package * SIMP will no longer ship with updated ClamAV DAT files

simp-utils

- Moved the default LDIF example files out of the `simp-doc` RPM and into `simp-utils` for wider accessibility

Known Bugs

- There is a bug in `Facter 3` that causes it to segfault when printing large unsigned integers - FACT-1732
  - This may cause your run to crash if you run `puppet agent -t --debug`
- The `krb5` module may have issues in some cases, validation pending
- The graphical `switch user` functionality does not work. We are working with the vendor to discover a solution
- The upgrade of the `simp-gpgkeys-3.0.1-0.noarch` RPM on a SIMP server fails to set up the keys in `/var/www/yum/SIMP/GPGKEYS`. This problem can be worked around by either uninstalling `simp-gpgkeys-3.0.1-0.noarch` prior to the SIMP 6.1.0 upgrade, or reinstalling the newer `simp-gpgkeys` RPM after the upgrade.

- An upgrade of the `pupmod-saz-timezone-3.3.0-2016.1.noarch` RPM to the `pupmod-simp-timezone-4.0.0-0.noarch` RPM fails to copy the installed files into `/etc/puppetlabs/code/environments/simp/modules`, when the `simp-adapter` is configured to execute the copy. This problem can be worked around by either uninstalling `pupmod-saz-timezone-3.3.0-2016.1.noarch` prior to the SIMP 6.1.0 upgrade, or reinstalling the `pupmod-simp-timezone-4.0.0-0.noarch` RPM after the upgrade.
- Setting selinux to disabled can cause stunnel daemon fail. Using the permissive mode of selinux does not cause these issues.

4.2.9 SIMP 6.0.0-0

Contents

This release is known to work with:

- RHEL 6.8 x86_64
- RHEL 7.3 x86_64
- CentOS 6.8 x86_64
- CentOS 7.0 1611 x86_64

Breaking Changes

Warning: This release of SIMP is **NOT** backwards compatible with previous releases. Direct updates will not work.

At this point, do not expect any of our code moving forward to work with Puppet 3.

If you find any issues, please file bugs!

Note: If you are working to integrate SIMP into Puppet Enterprise, these are the modules that you need to use since they are Puppet 4 compatible.

Breaking Changes Since RC1

Unfortunately, a few items were identified which necessitated additional breaking changes prior to the final release.

These are specifically enumerated here to make sure that they are not missed.

simp::yum Refactor

The `simp::yum` class was confusing and, as we attempted to install systems via `yum`, we found out just how bad it was.

Fundamentally, most installations of SIMP are going to have their own repos at some unknown location that they want to use. In ISO installations, which we can detect, there will be a local repo and we can set the parameters accordingly via `simp config`.

All of the old parameters have been removed, and to get back to old functionality, all that has to be done is add the following classes to nodes and adjust previous hiera settings to use the new classes:

```
---
classes:
- 'simp::yum::repo::local_os_updates'
- 'simp::yum::repo::local_simp'
---
```

RPM Installation

If installing from RPM, you will want to take a look at the latest documentation. The most important thing to be aware of is that there is now something called `simp-adapter` that must be installed with, or before, the `simp` RPM.

If you are using Puppet Enterprise, you'll want to use the `simp-adapter-pe` RPM instead.

Paths

Puppet AIO Paths

The system has been updated to use the Puppet AIO paths. Please see the Puppet Location Reference for full details.

SIMP Installation Paths

For better integration with r10k and Puppet Code Manager, SIMP now installs all materials in `/usr/share/simp` by default.

A script `simp_rpm_helper` has been added to copy the `environment` and `module` data into place at `/etc/puppetlabs/code` **if configured to do so**.

On the ISO, this configuration is done by default and will be set to auto-update for all future RPM updates. If you wish to disable this behavior, you should edit the options in `/etc/simp/adapter_config.yaml`.

Note: Anything that is in a Git or Subversion repository in the `simp` environment will **NOT** be overwritten by `simp_rpm_helper`.

SIMP Dynamic Content Paths

To ensure that SIMP dynamic content (ssh keys, generated passwords) are not mixed with Git-managed infrastructure, the SIMP dynamic content has been moved to `simp_autofiles` at the top level of the environment.

This will be moved down into `/var/simp/environments` for consistency in the final 6.0.0 release.

SIMP Rsync Paths

The SIMP Rsync subsystem now fully supports multiple environments. All environment-relevant materials have been moved to `/var/simp/environments/simp/rsync`. Please **copy** the contents of that directory if you create another environment.

SIMP Partitioning Scheme

SIMP no longer creates a `/srv` partition on EL 6 or 7. `/var` has assumed the role of `/srv`. The root partition size has been increased from **4GB** to **10GB**.

Significant Updates

Root Login via Console

Root is no longer allowed to log into clients or the SIMP server by default.

SIMP Scenarios and simp_config_settings.yaml

We have changed the way that SIMP includes classes. There is a new top-level variable, set in `manifests/site.pp` that controls the list of classes to be included. The goal of this change is to ease users with existing infrastructures into using full-bore SIMP.

Essentially, `simp_classes.yaml` has been replaced by class inclusions under the `simp::scenario` namespace and `simp_def.yaml` has been replaced by `simp_config_settings.yaml`. However, modifications should not be made to `simp_config_settings.yaml`. Settings from `simp_config_settings.yaml` should be changed by either running `simp config` again or be overwritten in `default.yaml`.

API Changes

Quite a few modules have had changes that make them incompatible with the Legacy SIMP stack.

We've attempted to capture those changes here at a high level so that you know where you are going to need to focus to validate your Hiera data, ENC hooks, and parameterized class calls.

Global catalysts and simp_options

SIMP Global catalysts now have a consistent naming scheme and are documented in code in the `simp_options` module. In particular, we have changed not only the value in hiera, but every module parameter that uses this value's name from `client_nets` to `simp_options::trusted_nets`. Other changes were less obtrusive, for example `enable_selinux` and other variations are now all called `simp_options::selinux`. Every Catalyst is strongly typed and documented in the module.

New catalysts are as follows:

- `simp_options::auditd`
- `simp_options::clamav`
- `simp_options::dns::search`
- `simp_options::dns::servers`
- `simp_options::firewall`
- `simp_options::haveged`
- `simp_options::ipsec`
- `simp_options::kerberos`
- `simp_options::ldap`
- `simp_options::ldap::base_dn`
- `simp_options::ldap::bind_dn`
- `simp_options::ldap::bind_hash`
- `simp_options::ldap::bind_pw`
- `simp_options::ldap::master`
- `simp_options::ldap::root_dn`
- `simp_options::ldap::sync_dn`
- `simp_options::ldap::sync_hash`
- `simp_options::ldap::sync_pw`
- `simp_options::ldap::uri`
- `simp_options::logrotate`
- `simp_options::ntpd::servers`
- `simp_options::openssl::cipher_suite`
- `simp_options::pam`
- `simp_options::pki`
- `simp_options::pki::source`
- `simp_options::puppet::ca`
- `simp_options::puppet::ca_port`
- `simp_options::puppet::server`
- `simp_options::puppet::server_distribution`
- `simp_options::selinux`

- simp_options::sssd
- simp_options::stunnel
- simp_options::syslog
- simp_options::syslog::failover_log_servers
- simp_options::syslog::log_servers
- simp_options::tcpwrappers
- simp_options::trusted_nets

Strong Parameter Typing

All SIMP provided modules should now be strong typed with Puppet Data Types.

De-Verbing of Defines

Many of the defined types have been renamed to no longer be 'verb-oriented'. The iptables module is probably the widest reaching change where the standard 'ease-of-use' aliases have been moved under a listen namespace.

For instance, iptables::tcp_stateful_listen is now iptables::listen::tcp_stateful

Additionally, any add_rule defines were changed to just rule. For example, auditd::add_rule was changed to just auditd::rule.

Centralized Management of Application x509 PKI Certs

In the past, application specific PKI certificates were copied into the application space. This varied per application and left certs strewn throughout the system. Now, certificates for all SIMP-managed applications are copied from /etc/pki/simp/x509, into a central location, /etc/pki/simp_apps/<application_name>/x509.

The extent to which SIMP manages PKI is governed by two new catalysts, pki and pki::source. Additionally, every SIMP module which uses pki has been modified to use a common set of pki class parameters. A high-level description is given below, using simp_elasticsearch as an example.

```
# @param pki
#   * If 'simp', include SIMP's pki module and use pki::copy to manage
#     application certs in /etc/pki/simp_apps/simp_elasticsearch/x509
#   * If true, do *not* include SIMP's pki module, but still use pki::copy
#     to manage certs in /etc/pki/simp_apps/simp_elasticsearch/x509
#   * If false, do not include SIMP's pki module and do not use pki::copy
#     to manage certs.  You will need to appropriately assign a subset of:
#     * app_pki_dir
#     * app_pki_key
#     * app_pki_cert
#     * app_pki_ca
#     * app_pki_ca_dir
#
# @param app_pki_external_source
#   * If pki = 'simp' or true, this is the directory from which certs will be
#     copied, via pki::copy.  Defaults to /etc/pki/simp/x509.
```

(continues on next page)

(continued from previous page)

```
#
#   * If pki = false, this variable has no effect.
```

Keydist

Keydist has been relocated to a second module path to facilitate working with r10k. The new modulepath is located at `/var/simp/environments/`, and the default location of keydist is now `/var/simp/environments/simp/site_files/pki_files/files/keydist/`

Forked modules

Most forked modules (modules that don't start with 'simp') have been updated to latest upstream.

Puppet AIO

The latest version of the Puppet AIO stack has been included, along with an updated Puppet Server and PuppetDB.

simp-extras

The main `simp` RPM has been split to move the lesser-used portions of the SIMP infrastructure into a `simp-extras` RPM. This RPM will grow as more of the non-essential portions are identified and isolated.

The goal of this RPM is to keep the SIMP core version churn to a minimum while allowing the ecosystem around the SIMP core to grow and flourish as time progresses.

Security Announcements

RPM Updates

Package	Old Version	New Version
puppet-agent	N/A	1.8.3-1
puppet-client-tools	N/A	1.1.1-1
puppetdb	2.3.8-1	4.3.0-1
puppetdb-termini	N/A	4.3.0-1
puppetdb-terminus	2.3.8-1	N/A
puppetserver	1.1.1-1	2.7.2-1

Removed Modules

pupmod-simp-foreman

- Removed until Foreman works consistently with Puppet 4

pupmod-simp-ganglia

- Not yet ported to Puppet 4

pupmod-simp-nscd

- Functionality replaced by `sssd`

pupmod-simp-openldap

- Renamed to `simp_openldap` to pave the way towards using a more up-to-date implementation of the core openldap component module from the community.

pupmod-simp-snmpd

- Not yet ported to Puppet 4.

pupmod-simp-windowmanager

- Rewritten and renamed module to `pupmod-simp-gnome`

pupmod-simp-xwindows

- Rewritten and renamed to `pupmod-simp-gdm`

Fixed Bugs

pupmod-simp-auditd

- Ensure that all rules are set to `always,exit` instead of `exit,always`
- Changed the default failure mode to `printk` since several required audit rules, such as `chmod` and `chown` would quickly overrun the auditd buffers on common scenarios, such as updating system packages
- Fixed an issue where the audisp `exec` was breaking idempotence. Also, now ensure proper restarting of auditd when audispd is updated

pupmod-simp-gdm

- Updated the managed service list

pupmod-simp-gnome

- Several minor bug fixes and package updates

pupmod-simp-pam

- Fixed the locations for the authconfig tools and made removal of the tools completely optional

pupmod-simp-pupmod

- Fixed the cron job unlock code so that it actually work as documented
- Made it more clear to the user how to disable the force-unlock

pupmod-simp-simp

- Fixed the removal of the auto-update cron job if disabled

pupmod-simp-simpcat

- To deconflict with the upstream `puppetlabs-concat` module, the `simpcat` **functions** were renamed to be prefaced by `simpcat` instead of `concat`.
- A simple find and replace of `concat_fragment` and `concat_build` in legacy code with `simpcat_fragment` and `simpcat_build` should suffice
  - Be sure to check for `Concat_fragment` and `Concat_build` resource dependencies!

pupmod-simp-simp_openldap

- Removed `acl` from the default log levels since it was causing `slapd` to hang on EL7 systems

pupmod-simp-ssh

- Fixed a bug in the `ssh::server::conf::subsystem` parameter where multiple word strings would be truncated to the first word only
- Updated the `UsePrivilegeSeparation` option on EL7 to be `sandbox`
- Defaulted `ssh::server::conf::pam` to `true`
- Changed default value of allowed remote hosts to `ALL` to prevent lockouts
  - If `simp_options::trusted_nets` is set, it will be used instead

New Features

pupmod-simp-at

- New module for controlling the `at` subsystem

pupmod-simp-logstash

- Added native TLS support and removed the requirement for Stunnel or IPTables redirects

pupmod-simp-iptables

- Added method to open ports through hiera.

pupmod-simp-pam

- Generic, custom content can be specified to replace templated content by using the `$use_templates` parameter.
- `pam::access:rule` resources can be added through hiera using the `$pam::access::users` hash.

pupmod-simp-pupmod

- Added explicit support for Puppet Enterprise systems
- Restrict auditing of puppet-related files to the Puppet Server

pupmod-simp-simp

- Moved the `runpuppet` code into its own class
- Added SIMP 'scenarios' which are common configurations for SIMP systems
  - simp -> Full SIMP, recommended
  - simp_lite -> SIMP without the scary stuff
  - poss -> Just connect Puppet on the client to the server
- Updated the GPG keys in the YUM repo lists

pupmod-simp-simplib

- Removed all manifests and Puppet code from this module. It now only contains functions and custom type aliases.
- List of modules that were created or forked after removing content from simplib:
  - pupmod-simp-at
  - pupmod-simp-chkrootkit
  - pupmod-simp-useradd
  - pupmod-simp-swap

  - pupmod-simp-cron
  - pupmod-simp-resolv
  - pupmod-simp-issue
  - pupmod-simp-fips
  - puppetlabs-motd
  - trlinkin-nsswitch
  - camptocamp-kmod
  - puppetlabs-motd
  - saz-timezone
- The rest of the content was added to our profile module, simp-simp

pupmod-simp-simp_rsyslog

- Added a warning if possible log looping is detected

pupmod-simp-sudo

- Added method to create `user_specification` resources through hiera

pupmod-simp-svckill

- The default service killing behavior has been set to 'warning'. However, `simp cli` will ask for the setting during config.

rubygem-simp_cli

- Completely updated `simp config` and `simp bootstrap`.
- `simp passgen` was updated to support environments.

Known Bugs

- A bug is still allowing root to log into client systems on a console even though `/etc/securetty` is present and empty
- The `krb5` module may have issues in some cases, validation pending
- The graphical `switch user` functionality does not work. We are working with the vendor to discover a solution

4.3 SIMP Getting Started Guide

Welcome to SIMP!

4.3.1 Introduction

What is SIMP?

The System Integrity Management Platform (*SIMP*) is an *Open Source* framework designed around the concept that individuals and organizations should not need to repeat the work of automating the basic components of their operating system infrastructure.

Expanding upon this philosophy, SIMP also aims to take care of routine policy compliance to include *NIST 800-53*, *FIPS* 140-2, the *DISA STIG*, and the *SCAP Security Guide*.

By using the *Puppet* automation stack, SIMP is working toward the concept of a self-healing infrastructure that, when used with a consistent configuration management process, will allow users to have confidence that their systems not only start in compliance but remain in compliance over time.

Finally, SIMP has a goal of remaining flexible enough to properly maintain your operational infrastructure. To this end, where possible, the SIMP components are written to allow all security-related capabilities to be easily adjusted to meet the needs of individual applications.

Conventions used in this documentation

The following typographic conventions are used in this documentation:

Purpose	Example
Code snippets	`ensure => true`
CLI commands	**`yum update -y`**
Files	`/etc/simp/version.`*`x.y`*
Programs	**`mcstransd`**
Packages (e.g., RPM, RubyGem)	simp-utils
Puppet Modules	simp/simplib
GitHub repos	simp/simp-doc
Jira issues	SIMP-8464
Literal text	`keyword`
Internal hyperlinks	*Changelogs*
Glossary terms	*SIMP*
External hyperlinks	SIMP website

4.3.2 System Requirements

The scalability of SIMP correlates to the scalability of Puppet. From the Puppet tuning guide, a number of factors contribute to scalability, including:

- Speed and quantity of available hardware
- Number of nodes, and frequency of check-in
- Number of modules in your module path
- Amount of hieradata

Official *hardware requirements* **for your SIMP server**:

- At least 3.4 GB of RAM + swap.

With less than 3.4 GB it is possible for either the puppet database or puppet server to not start.

We recommend the following hardware requirements **for your SIMP server**:

- **2** CPUs and **6 GB** of RAM, at a minimum
- **2 - 4** CPUs and **10 GB** of RAM to serve up to *1,000* nodes

The SIMP team recommends allocating the latter, in addition to a minimum of **50 GB** HDD space. Again, these are not hard requirements, but anything less may not leave adequate room for logs, applications, rsync data, etc.

Note: If you want to optimize the Puppet server, the Puppet tuning guide is a good place to start. Use the advanced memory debugging guide for further optimization.

4.3.3 Known OS Compatibility

- **SIMP 6.X**
- **CentOS 7.0**
  - **ISO #1:** CentOS-7-x86_64-DVD-2009.iso
  - **Checksum:** e33d7b1ea7a9e2f38c8f693215dd85254c3a4fe446f93f563279715b68d07987
- **CentOS 8.5**
  - **ISO #1:** CentOS-8.5.2111-x86_64-dvd1.iso
  - **Checksum:** 3b795863001461d4f670b0dedd02d25296b6d64683faceb8f2b60c53ac5ebb3e
- **CentOS 8.7**
  - **ISO #1:** CentOS-Stream-8-x86_64-20230125-dvd1.iso
  - **Checksum:** 91454029bf70382eff9032d384093747d588ce4c96d95028b1c21b6f8154c712
- **RedHat 7.8**
  - **ISO #1:** rhel-server-7.8-x86_64-dvd.iso
  - **Checksum:** 3ca2017f12392fc8acad435c05a941bb88620ece8eb5740278ffa61d8340792d
- **RedHat 8.5**
  - **ISO #1:** rhel-server-8.5-x86_64-dvd.iso
  - **Checksum:** 48f955712454c32718dcde858dea5aca574376a1d7a4b0ed6908ac0b85597811

4.3.4 Installation Options

Tip: Please take a look at the *Frequently Asked Questions* prior to installing SIMP. The most relevant questions for new users will always be at the top of the list.

If you are installing on a system with an existing operating system, you will want to *install from a SIMP Repository*. Otherwise, you will probably want to *install from the ISO*, which configures the disks and partitions to be compatible with most STIGs.

ISO Installation (Preferred)

Installing SIMP from an ISO

SIMP can be installed from a bootable ISO, which provides many advantages:

- Provides a ready-to-go OS, Puppet server, and SIMP installation
- Suitable for use in network-isolated enclaves/offline environments
- Ensures OS is configured with compliance-relevant install-time options:
  - The *Disk Partitioning* scheme is compatible with most security guides
  - The OS will boot in *FIPS* mode
  - Disks will be encrypted
    - (Please note the important caveats in the *Disk Encryption* section)

Obtaining a SIMP ISO file

You can obtain a SIMP installation ISO using one of the following methods:

1. Downloading an ISO image file
   - Official releases are available at https://download.simp-project.com/simp/ISO/
2. *[Advanced] Building a SIMP ISO* for yourself
   - The contents of the ISO can be customized to your preferences.
   - This is the only way to obtain a SIMP ISO that installs a licensed commercial OS, such as Red Hat Enterprise Linux (*RHEL*).

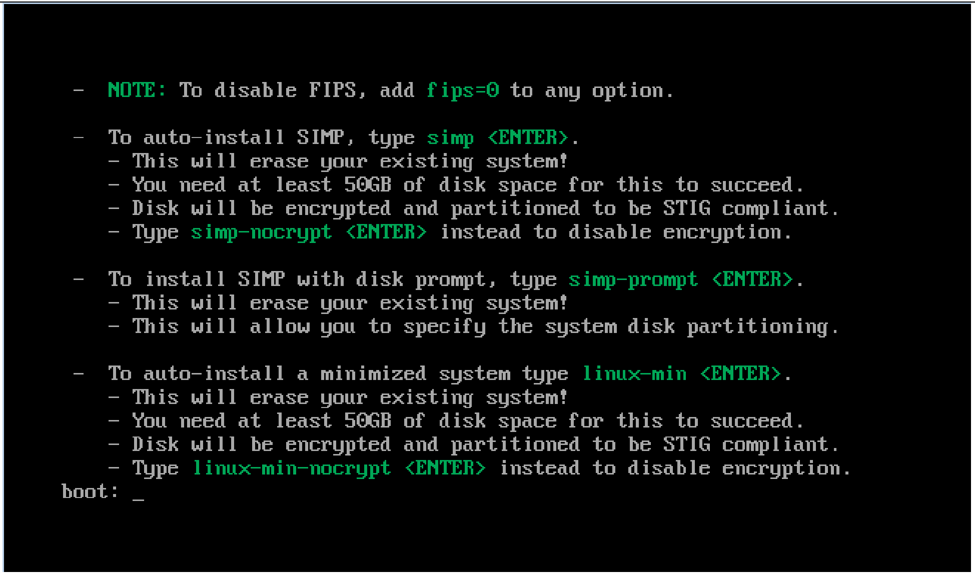
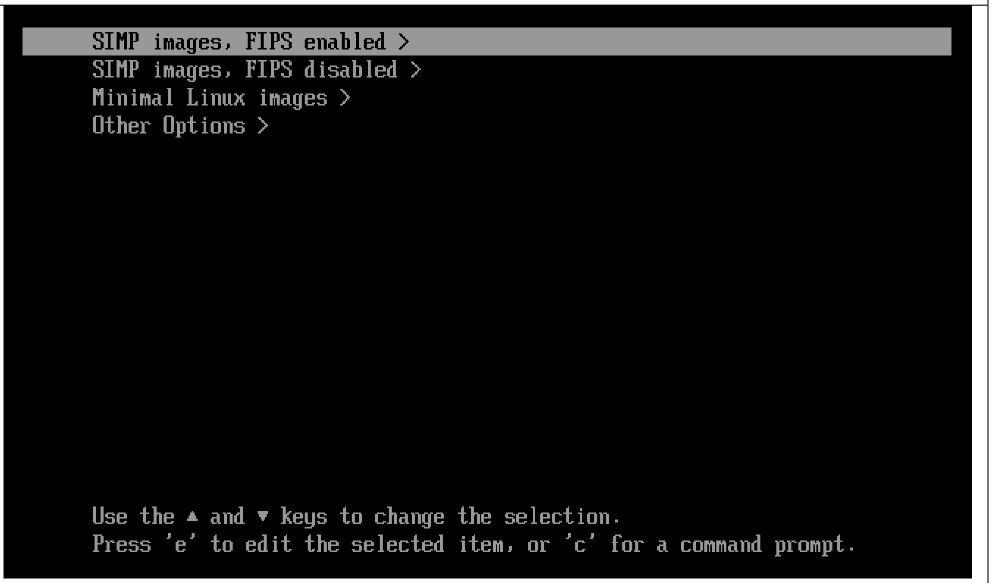
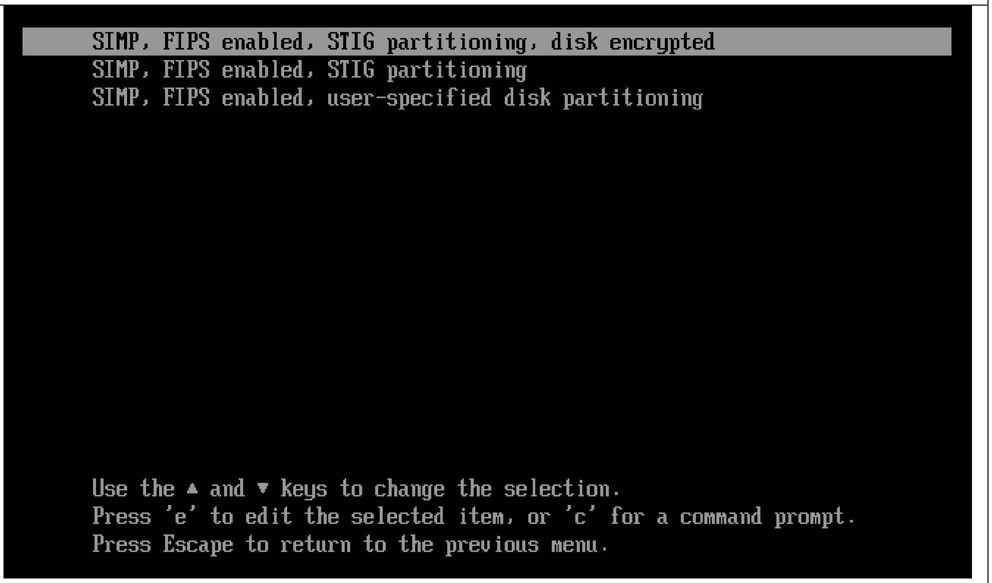
Installing the OS

A SIMP ISO will install its OS + SIMP on any host that supports the underlying operating system.

Install as follows:

1. Boot the system using the SIMP ISO.

   The ISO will load into a screen of boot options. The presentation will differ, depending on the boot firmware and ISO OS:

BIOS	
UEFI	
UEFI (sub-menu)	

```
 -  NOTE: To disable FIPS, add fips=0 to any option.

 -  To auto-install SIMP, type simp <ENTER>.
    - This will erase your existing system!
    - You need at least 50GB of disk space for this to succeed.
    - Disk will be encrypted and partitioned to be STIG compliant.
    - Type simp-nocrypt <ENTER> instead to disable encryption.

 -  To install SIMP with disk prompt, type simp-prompt <ENTER>.
    - This will erase your existing system!
    - This will allow you to specify the system disk partitioning.

 -  To auto-install a minimized system type linux-min <ENTER>.
    - This will erase your existing system!
    - You need at least 50GB of disk space for this to succeed.
    - Disk will be encrypted and partitioned to be STIG compliant.
    - Type linux-min-nocrypt <ENTER> instead to disable encryption.
boot: _
```

```
      SIMP images, FIPS enabled >
      SIMP images, FIPS disabled >
      Minimal Linux images >
      Other Options >

      Use the ▲ and ▼ keys to change the selection.
      Press 'e' to edit the selected item, or 'c' for a command prompt.
```

```
      SIMP, FIPS enabled, STIG partitioning, disk encrypted
      SIMP, FIPS enabled, STIG partitioning
      SIMP, FIPS enabled, user-specified disk partitioning

      Use the ▲ and ▼ keys to change the selection.
      Press 'e' to edit the selected item, or 'c' for a command prompt.
      Press Escape to return to the previous menu.
```

2. Press *Enter* to boot the standard SIMP installer, or customize the installation using the boot options.

   Note: For details about how SIMP implements disk encryption (enabled by default), see: *Disk Encryption*.

3. Once installation starts, you may see the graphical interface spawn.

   Warning: You should NOT interact with the GUI **unless** you have elected to manage your own disk partitions (e.g., `simp-prompt`).

   Note: If you have opted to manage your own disk partitions with (e.g., `simp-prompt`), follow the GUI instructions to enter your partition scheme.

   For example, using SIMP for CentOS 7:

   1. Click the `INSTALLATION DESTINATION` button
   2. Configure the desired partitioning
   3. Click the `DONE` button to finalize your disk selections
   4. Click the `Begin Installation` button on the main GUI page to continue.

   No further GUI interaction will be required.

   Tip: When applying disk encryption (enabled by default), the system may seem to pause and display messages about increasing entropy. You can speed up the installation by pressing random keys on the keyboard for a bit (this will generate additional entropy).

4. When the installation is complete, the system will restart automatically.

   Note: When the system boots, it may display: `error on start module sha1 not found could not insert sha_256 [...]`. This is expected and is a known issue.

5. Change the default passwords.

   Warning: There are default passwords present on the system that should be changed prior to deploying the system.

   Please make sure that you change these passwords!

   Note: See the *What is the Password Complexity for SIMP?* FAQ for tips on setting a functional password.

   a. Change the `root` user password.

   i. At the console, log on as `root` and type the default password shown in *SIMP Default Passwords*

   ii. Follow the prompts to complete the password change

   b. Change the `simp` user password.

i. At the console, log on as `simp` and type the default password shown in *SIMP Default Passwords*

ii. Follow the prompts to complete the password change

SIMP Default Passwords

Below is a table containing the default passwords found on a basic SIMP server upon install.

Important: All default passwords must be changed during the initial configuration process.

Utility	Password
Grub	GrubPassword
Root User	RootPassword
Simp User	UserPassword

Table: SIMP Default Passwords

Next Steps

Now that your system has been installed, you should proceed to *Initial SIMP Server Configuration* to complete the initial setup.

Advanced Installation Topics

Contents:

Disk Encryption

The default *ISO* and kickstart files in SIMP encrypt the first physical volume, if the `simp_disk_crypt` option is provided at the boot command line or, on an EFI system, one of the menu options that enable disk encryption is selected.

Warning: The system is set to **automatically** decrypt at boot! This means that the password is embedded in the *initrd* file.

Note: The `/boot` directory is **not** encrypted, since that would prevent the system from booting automatically.

Method

When enabled, SIMP implements disk encryption, with automatic decryption, so that users have the option to use their own keys in the future. Alternatively, users may remove the system local keys and require that a password be entered at each boot.

The primary goal of providing automatic decryption was to give users a clean and seamless experience when using the initial system. It is understood that this is not best practice since automatic decryption of the disks requires the system to embed the password files in the system *initrd*.

Disk encryption was not enabled by default for two reasons. The first is that it can take an unacceptable amount of time to build a system if enough entropy is not present. The second is that a lot of hardware contains the ability to encrypt the disk at that level. If this is present, the utility of a second layer of disk encryption is not necessarily warranted or a good idea.

Implementation

The system keys are referenced in `/etc/crypttab` and, by default, reside at `/etc/.cryptcreds`. At build time, these files are copied into all *initrd* files present on the system. This ensures that all kernels can successfully boot the system.

The `/etc/dracut.conf` file is also updated to ensure that any new kernel loads will be able to boot successfully.

Warning: The `/etc/.cryptcreds` file **is** encrypted when the system is off. However, a copy is in the unencrypted *initrd* files in `/boot` and should not be considered secure from physical access to the raw disk image.

Note: Please be aware that **all** characters in the `/etc/.cryptcreds` file are part of the password. The lack of a trailing newline is **very** important.

Replacing the Current Password

Note: The underlying system uses *LUKS*, so any usage outside of this document should refer to the *LUKS* implementation that matches your system version.

To change the password, you will need to perform the following steps.

1. Back up the original password file

- If something goes amiss, you're seriously going to need this

1. Get the *UUID* of your partition

- This will be in the `/etc/crypttab` file. You'll want the entire `UUID=<uuid>` string

1. Create the new password

- Remember that this needs to be **exactly** what you will use. If you ever expect to type this at the command line, don't forget to strip your trailing spaces.

```
#!/usr/bin/python

import sys
```

(continues on next page)

(continued from previous page)

```
import random
import string

# The length of the new password
length = 1024

# What the password should consist of
charset = string.lowercase+string.uppercase+string.digits

passfile = open('/etc/.cryptcreds.new','w')

passfile.write("".join(random.choice(charset) for i in range(length)))
```

4. Update the key

- There is a faster way to do this in *EL* 7, but this method works on both systems

```
cryptsetup luksAddKey --key-slot 1 --key-file /etc/.cryptcreds UUID=<uuid> /etc/.cryptcreds.new
cryptsetup luksKillSlot --key-file /etc/.cryptcreds 0

cryptsetup luksAddKey --key-slot 0 --key-file /etc/.cryptcreds.new UUID=<uuid> /etc/.cryptcreds.new
cryptsetup luksKillSlot --key-file /etc/.cryptcreds.new 1

# Only do this step if the previous steps succeeded!
mv /etc/.cryptcreds.new /etc/.cryptcreds
```

5. Update your *initrd* files

- You want to make sure to update **all** of your *initrd* files since you'll want to be able to boot from any kernel.

```
for x in `ls -d /lib/modules/*`; do
  installed_kernel=`basename $x`
  dracut -f "/boot/initramfs-${installed_kernel}.img" $installed_kernel
done
```

Removing the Password File

If you wish to remove the password file from your system, you will need to perform the following steps:

1. Back up the password file!

- If you lose this, you won't be able to get into your system after reboot

1. Using your favorite text editor, remove the *install_items* line in */etc/dracut.conf* that contains the reference to */etc/.cryptcreds*
2. Remove the */etc/.cryptcreds* file from the system
3. Update your *initrd* files

- You want to make sure to update **all** of your *initrd* files since you'll want to be able to boot from any kernel.

```
for x in `ls -d /lib/modules/*`; do
  installed_kernel=`basename $x`
  dracut -f "/boot/initramfs-${installed_kernel}.img" $installed_kernel
done
```

Disk Partitioning

The default SIMP installation has a disk partitioning scheme that is designed to meet multiple compliance and system hardening standards and requirements. If you choose to implement a custom partitioning scheme you will need to make sure to use one that meets the standards and requirements your organization implements.

Default Partition Scheme

By default SIMP has the following partitions:

- swap
- /
- /boot
- /boot/efi
- /home
- /tmp
- /var
- /var/log
- /var/log/audit

Required Partitions

SIMP will fail to install if the following partitions do not exist: * / * /boot

Installing SIMP From A Repository

Using the official SIMP YUM repositories is the simplest method for getting up and running with SIMP on an existing infrastructure. If you are using a virtual infrastructure, such as AWS, Microsoft Azure, Google Cloud, or your own internal VM stack, this is the method that you will want to use.

Note: This method does **not** modify your system's partitioning scheme or encryption scheme to meet any regulatory policies. If you want an example of what that should look like see the *Kickstart* files in the simp-core Git repository.

Enable EPEL

Note: RHEL systems will need to enable the *EPEL* Repositories manually.

```
sudo yum install epel-release -y
sudo yum install pygpgme yum-utils -y
```

Install The SIMP-Project Repositories

Please reference *HOWTO Use the SIMP Release RPM*.

Rebuild The Yum Cache

```
sudo yum makecache
```

Install the SIMP Server

Install the `puppetserver` package as follows:

```
sudo yum install -y puppetserver
```

Install the core SIMP packages as follows:

```
sudo yum install -y simp
```

The `simp` RPM installs the SIMP core Puppet modules and other critical SIMP assets such as its environment skeleton, custom SELinux policy, CLI, and utilities.

- The Puppet modules are installed into `/usr/share/simp` and do not affect any existing Puppet environment. Other steps in the SIMP server setup will deploy the modules into a Puppet environment.

SIMP also provides a large number of 'extra' Puppet module packages that you can install as needed (`pupmod-simp-gnome`, `pupmod-simp-nfs`, etc.). You can discover what extra modules are available by searching for `pupmod` via `yum`. Alternatively, you can install all of the extra Puppet modules into `/usr/share/simp` by simply running `sudo yum install -y simp-extras`.

Next Steps

Now that your system has been installed, you should proceed to *Initial SIMP Server Configuration* to complete the initial setup.

Once you're done setting up your environment, you should proceed to the *SIMP User Guide* for utilizing SIMP to its full potential.

Tip: If you run into issues or bugs, please drop us a line on the SIMP Development Mailing List.

4.4 SIMP User Guide

Contents:

4.4.1 Introduction

This chapter will walk a user through instructions on administering a *SIMP* system, including the processes for managing clients and users.

Level of Knowledge

SIMP targets users with a strong background in Linux systems management. The core technologies that require prerequisite knowledge are:

- *Puppet*
- *Domain Name System* (DNS) - *BIND*
- *Dynamic Host Configuration Protocol* (DHCP) - Internet Systems Consortium (ISC) DHCP
- *Lightweight Directory Access Protocol* (LDAP)
  - *389-DS* for *EL*8
  - *OpenLDAP* for *EL*7
- RedHat *Kickstart*, including all technologies involved: *Trivial File Transfer Protocol* (TFTP), *PXE*, PXELinux, etc.
- The Apache HTTP Server
- The *Yellowdog Updater, Modified* (YUM) package manager
- *Rsyslog* 8+
- *Firewalld* or *IPTables*
- *Auditd*, Basic knowledge of how the daemon works
- *Advanced Intrusion Detection Environment* (AIDE), basic knowledge of the rules
- Basic *X.509*-based *PKI* Key Management

SIMP handles as much of the initial setup and management of these tools as possible However, you will need at least some understanding of them in order to tailor a SIMP system to fit the desired environment. You will also need a general understanding of how to control and manipulate these tools from the *command line interface* (CLI); SIMP does not provide a *graphical user interface* (GUI).

Knowledge of scripting and *Ruby* programming will also help to further customize a SIMP install but is not required for routine use.

SIMP Defined

The System Integrity Management Platform (*SIMP*) is an *Open Source* framework designed around the concept that individuals and organizations should not need to repeat the work of automating the basic components of their operating system infrastructure.

Expanding upon this philosophy, SIMP also aims to take care of routine policy compliance to include *NIST 800-53*, *FIPS* 140-2, the *DISA STIG*, and the *SCAP Security Guide*.

By using the *Puppet* automation stack, SIMP is working toward the concept of a self-healing infrastructure that, when used with a consistent configuration management process, will allow users to have confidence that their systems not only start in compliance but remain in compliance over time.

Finally, SIMP has a goal of remaining flexible enough to properly maintain your operational infrastructure. To this end, where possible, the SIMP components are written to allow all security-related capabilities to be easily adjusted to meet the needs of individual applications.

4.4.2 Initial SIMP Server Configuration

Introduction to the SIMP Utility

The `simp` command provides a CLI intended to make the configuration of the *SIMP server* straightforward and repeatable. In these instructions, we will be using the `config` and `bootstrap` options of the `simp` command

For a list of the commands `simp` provides, type **`simp help`**. Type **`simp <Command> --help`** for more information on a specific command.

- **`simp config`** sets up configuration required to bootstrap the SIMP server with Puppet. It asks questions, generates configuration files, and applies preliminary server configuration based on the answers. It records the options chosen in a file, `/root/.simp/simp_conf.yaml` and generates a log file under `/root/.simp/`.
  - You can use the `--dry-run` option to step through the questions without changing anything and then run **`simp config -a /root/.simp/simp_conf.yaml`** to apply the changes.
  - **`simp config`** uses the `production` *Puppet Environment* by default. If you want to use a different initial environment, see *HOWTO Use an Alternate 'simp config' Environment*.
- **`simp bootstrap`** uses several targeted Puppet runs to configure the rest of the system and generates a log file under `/root/.simp/`.

For more details about initial configuration provided by **`simp config`**, see *Advanced Configuration*.

Configuring the SIMP Server

> **Warning:** Puppet has problems when hostnames contain capital letters (SERVER-1809) — do not use them!

1. Log on as a user that can gain `root` access and `sudo` to `root`.
   - **If you installed from the ISO**
     - Log in as `simp`.
     - Run **`sudo su - root`**.
   - **If you installed from RPM**
     - Create a local user that can escalate to `root` and use it to access the `root` account.

2. Run **simp config** and configure the system as prompted.

    - These settings will be used to set up files appropriate for bootstrapping the system.

        - For each setting:

            - Press *Enter* to keep the recommended value or enter your desired value.

    - For more details about **simp config**'s installation variables and actions, see *Advanced Configuration*.

Note: If you see a message about 'simp bootstrap' being 'locked', follow the steps in *Prevent Lockout from the SIMP Server during RPM Installation*:

3. Run **simp bootstrap**.

    If your SIMP server is on a virtual machine, or slow system, the default timeout for the Puppet server to start (5 minutes) may be too short. You will want to extend this time by using the -w option.

    For example, to extend the timeout to 10 minutes:

    ```
    simp bootstrap -w 10
    ```

Note: If the bootstrap progress bars of each Puppet run are of equal length, a problem has probably occurred due to an error in SIMP configuration. Refer to the previous step and make sure that all configuration options are correct.

You can debug issues by either looking at the log files in /root/.simp or by running **puppet agent -t --masterport=8150 --agent_disabled_lockfile /opt/puppetlabs/server/data/puppetserver/state/bootstrap.lock**.

4. Run **reboot** to restart your system and apply the necessary kernel configuration items.

After rebooting, SIMP-managed security settings have been applied and the SIMP server is ready for site-specific configuration.

To su to root from the simp user, you must now use **sudo su - root**.

Next Steps

- To continue configuring the system, move on to the next section in the *SIMP User Guide*, *Client Management*.

The following are links to other information in the user guide that are answers to questions sometimes asked at this time:

- To learn more details about how your system has just been configured see *Advanced Configuration*.
- To add additional simp modules or you own site modules to the environment *Updating an Existing Environment*
- To extract the Full OS to the yum server see *HOWTO Use unpack_dvd to Extract the Full OS RPM Package Set*

Prevent Lockout from the SIMP Server during RPM Installation

By default, SIMP:

- Disables remote logins for all users.
- Disables `root` logins at the console.

If either of the following scenarios applies, you must enable `sudo` and `ssh` access for a local user. If you do not do this, you may lose access to your system.

Scenario 1:

Console access is available, but not allowed. Only the `root` user (and no other user account) is available.

- This generally occurs when SIMP is installed from RPM and the user accepts `simp config`'s default value for `useradd:securetty` (an empty array).

Scenario 2:

Console access is not available, and the administrative user's `ssh` access has not yet been enabled permanently via Puppet.

- This generally occurs when SIMP is installed from RPM on cloud systems.

In either of these scenarios, `simp config` will issue the following warning and write a lock file to prevent `simp bootstrap` from running:

```
'simp bootstrap' has been locked due to potential login lockout.
* See /root/.simp/simp_bootstrap_start_lock for details
```

The remainder of this document provides instructions on ensuring that a local user has the appropriate level of system access.

After you address all issues identified in `/root/.simp/simp_bootstrap_start_lock` (see *Resolve Other Issues in the Bootstrap Lock File*), you should remove the file and *continue with the bootstrap process*.

Ensuring Local User Access

- **If you already have an unprivileged account:**
  - Replace `userx` with your current **non-root** username throughout the example code.
- **If you do NOT already have an unprivileged account:**
  - Create a local user account, using `useradd`.
    - This example assumes the local user is named `userx`.
    - Be sure to set the user's password if the user is logging in with a password!

1. Run `sudo su - root`
2. Run `cd /etc/puppetlabs/code/environments/production/data/hosts`
3. Add the following to `<puppetserver fqdn>.yaml`

```
# Add sudo user rules
sudo::user_specifications:
  # Any unique name
  userx_su:
    # The users to which to apply this sudo rule
    user_list:
```

(continues on next page)

(continued from previous page)

```
      - userx
    # The commands that the user is allowed to run
    cmnd:
      - ALL
    # Whether or not the user must use a password
    passwd: false
# Add a PAM remote access rule
pam::access::users:
  # The user to add
  userx:
    # Allow access from everywhere
    origins:
      - ALL
```

If Your Local User Uses an SSH Public Key

If the local user has an SSH public key available, copy the `authorized_keys` file for that user to the SIMP-managed location for authorized keys `/etc/ssh/local_keys`:

```
mkdir -p /etc/ssh/local_keys
chmod 755 /etc/ssh/local_keys
cp ~userx/.ssh/authorized_keys /etc/ssh/local_keys/userx
chmod 644 /etc/ssh/local_keys/userx
```

Resolve Other Issues in the Bootstrap Lock File

If any other issues are identified in `/root/.simp/simp_bootstrap_start_lock`, you must address them before removing the file.

- Currently, the only other issue `simp config` will identify is a possible misconfiguration of YUM repositories. `simp config` will lock out bootstrap if it cannot find a few of the key packages needed for bootstrapping. Fix your yum repository configuration and then verify the fix using the verification instructions in the lock message.

Resume Bootstrap Operation and Verify User Access

> **Warning:** **DO NOT REBOOT BEFORE VERIFYING USER ACCESS USING AN ALTERNATE TERMINAL OR SSH SESSION**

1. Remove the lock file and bootstrap the system

    - `rm /root/.simp/simp_bootstrap_start_lock`
    - `simp bootstrap`
    - `puppet agent -t`

    The following items are not failures and can be ignored. All other errors or warnings should be addressed prior to proceeding:

    - Reboot notifications.

  - Warning/errors related to modules that manage services you have not completely set up, such as `named`.
  - `svckill` warnings regarding services found that would be killed if `svckill::mode` was set to `enforcing`.
    - See *My Services Are Dying!*.

2. Verify user access
   - Using a **NEW SSH SESSION OR TERMINAL** (do NOT close your working session)
     - Log in as `userx`
     - `sudo su - root`

> **Warning:** If your new user cannot ssh into the server and sudo to `root`
>
> - DO NOT reboot the server until you resolve the problem!
> - DO NOT log out of your primary work terminal until you resolve the problem!

Finalization

Reboot your system to enact the kernel-level enforcement changes:

- `reboot`

Re-verify system access:

- Log back in as `userx`
- `sudo su - root`

Advanced Configuration

The goal of `simp config` is to allow the user to quickly configure the SIMP server with minimal user input/operations. To that end `simp config` sets installation variables based on information gathered from the user, existing system settings, and SIMP security requirements. It then applies the smallest subset of these system settings that is required to bootstrap the system with Puppet. Both the installation variables and their application via `simp config` are described in subsections that follow.

Installation Variables

This section describes the installation variables set by `simp config`. Although the table that follows lists **all** possible installation variables, the user will not be prompted for all of them, nor will all of them appear in the configuration files generated by `simp config`. Some of these variables will be automatically set based on other installation variables, system settings, or SIMP security requirements. Others will be omitted because either they are unnecessary for a particular site configuration, or their defaults are appropriate. Also, please note that variables beginning with `cli::` are only used internally by `simp config`, itself. The `cli::` variables are written to `simp_conf.yaml`, but not persisted to any Puppet hiera data files.

Important:

- Not all the settings listed below may be able to be preset in a configuration file input to `simp config`, via either `-a <Config File>` or `-A <Config File>`. Some settings for which you would not be prompted if you ran `simp config` interactively may be automatically determined by `simp config`.

- Passwords for which only hashed values are stored in the YAML output of `simp config` must be input as hashed values in an input configuration file.
- `simp config` behaves differently (asks different questions, automatically determines different settings) depending on the SIMP installation type. This is because it can safely assume certain server setup has been done, only if SIMP has been installed from the SIMP-provided ISO. For example, consider a `simp` local user. When SIMP is installed from ISO, `simp config` can safely assume that this user is the backup user installed by the ISO to prevent server lockout. As such, `su` and `ssh` privileges for the `simp` user should be allowed. For non-ISO installs, however, it would not be prudent for `simp config` to grant just any `simp` user both `su` and `ssh` privileges.
- `simp config` detects that SIMP has been installed from a SIMP-provided ISO by the presence of `/etc/yum.repos.d/simp_filesystem.repo`.

Tip: There are two `simp config` options that are particularly useful:

- `--dry-run` will run through all of the prompts without applying any changes to the system. This is useful to:
  - become familiar with the variables set by `simp config` without applying them
  - generate a configuration file to use as a template for subsequent `simp config` runs
- `-a <Config File>` will load and apply a previously-generated configuration (aka the 'answers' file) in lieu of prompting for settings.
  - This is useful to run on systems that will be rebuilt often.
  - Please note, however: if you edit the answers file, only configuration settings for which you would be prompted by `simp config` can be modified in that file—any changes made to settings that `simp config` automatically determines will be ignored.

If you want to understand what variables apply to your setup, run `simp config --dry-run` and examine the generated `simp_conf.yaml` file. That file will contain both the settings and their documentation.

Variable	Description
cli::is_simp_ldap_server	Whether the SIMP server will also be a SIMP-provided LDAP server
cli::network::dhcp	Whether to use DHCP for the network; *dhcp* to enable DHCP, *static* otherwise
cli::network::gateway	Default gateway
cli::network::hostname	FQDN of server
cli::network::interface	Network interface to use
cli::network::ipaddress	IP address of server
cli::network::netmask	Netmask of the system
cli::network::set_up_nic	Whether to set up the network interface; *true* or *false*
cli::set_grub_password	Whether to set a GRUB password on the server; *true* or *false*
cli::simp::scenario	SIMP scenario; *simp* = full SIMP system, *simp_lite* = SIMP system with some sec
cli::use_internet_simp_yum_repos	Whether to configure SIMP nodes to use internet SIMP and SIMP dependency YU
grub::password	GRUB password hash
puppetdb::master::config::puppetdb_port	Port used by the puppet database
puppetdb::master::config::puppetdb_server	DNS name or IP of puppet database server
simp_openldap::server::conf::rootpw	LDAP Root password hash
simp_options::dns::search	Search domain for DNS
simp_options::dns::servers	List of DNS servers for the managed hosts
simp_options::fips	Enable FIPS-140-2 compliance; *true* or *false*; value automatically set to detected s
simp_options::ldap	Whether to use LDAP; *true* or *false*
simp_options::ldap::base_dn	LDAP Server Base Distinguished Name

Variable	Description
simp_options::ldap::bind_dn	LDAP Bind Distinguished Name
simp_options::ldap::bind_hash	LDAP Bind password hash
simp_options::ldap::bind_pw	LDAP Bind password
simp_options::ldap::master	LDAP master URI
simp_options::ldap::sync_dn	LDAP Sync Distinguished Name
simp_options::ldap::sync_hash	LDAP Sync password hash
simp_options::ldap::sync_pw	LDAP Sync password
simp_options::ldap::uri	List of LDAP server URIs
simp_options::ntpd::servers	NTP servers
simp_options::puppet::ca	FQDN of Puppet Certificate Authority (CA)
simp_options::puppet::ca_port	Port Puppet CA will listen on
simp_options::puppet::server	FQDN of the puppet server
simp_options::sssd	Whether to use SSSD
simp_options::syslog::failover_log_servers	IP addresses of failover log servers
simp_options::syslog::log_servers	IP addresses of primary log servers
simp_options::trusted_nets	Subnet used for clients managed by the puppet server
simp::runlevel	Default system run level; 1-5
simp::server::allow_simp_user	Whether to allow local ‘simp’ user su and ssh privileges
simp::yum::repo::local_os_updates::enable_repo	Whether to enable the SIMP-managed, OS Update YUM repository that the SIMP
simp::yum::repo::local_os_updates::servers	YUM server(s) for SIMP-managed, OS Update packages
simp::yum::repo::local_simp::enable_repo	Whether to enable the SIMP-managed, SIMP and SIMP dependency YUM reposit
simp::yum::repo::local_simp::servers	YUM server(s) for SIMP-managed, SIMP and SIMP dependency packages
sssd::domains	List of SSSD domains
svckill::mode	Strategy svckill should use when it encounters undeclared services; *enforcing* = sh
useradd::securetty	A list of TTYs for which the root user can login

simp config Actions

In addition to creating the three configuration, YAML files, `simp config` performs a limited set of actions in order to prepare the system for bootstrapping. Although the table that follows lists all possible `simp config` actions, not all of these actions will apply for all site configurations.

Category	Actions Performed
Certificates	If no certificates for the host are found in `/var/simp/environments/production/site_files/pki_files/files/keydist`, `simp config` will use SIMP's FakeCA to generate interim host certificates. These certificates, which are independent of the certificates managed by Puppet, are required by SIMP and should be replaced by certificates from an official *Certificate Authority*, as soon as is practical.
GRUB	When the user selects to set the GRUB password `simp config` will set the password in the appropriate grub configuration file, `/etc/grub.conf` or `/etc/grub2.cfg`. After initial configuration, the GRUB password can be managed with the `simp-simp_grub` module. See *Managing GRUB Users* for more information.
LDAP	When the SIMP server is also a SIMP-provided LDAP server, `simp config` • Adds `simp::server::ldap` to the SIMP server host YAML file, which allows the SIMP server to act as a LDAP server • Adds the hash of the user-supplied LDAP root password to the SIMP server host YAML file as `simp_openldap::server::conf::rootpw` to the SIMP
Lockout Prevention	When the SIMP server is installed from ISO, the install creates a local `simp` user that the SIMP server configure to have both su and ssh privileges. (This user is provider to prevent server lockout, as, per security policy, SIMP by default disables logins via ssh for all users, including `root`.) So, when SIMP is **not** installed from ISO, `simp config` does the following: • Warns the operator of this problem • Writes a lock file containing details on how to rectify the problem. This lock file prevents `simp bootstrap` from running until the user manually fixes the problem. • Turns off the SIMP server configuration that allows su and ssh privileges for an inapplicable `simp` user.
Network	• When the user selects to configure the network interface, `simp config` uses Puppet to set the network interface parameters in system networking files and to bring up the interface. • `simp config` sets the hostname.
Puppet	• Creates a new `production` *SIMP Omni-Environment* unless `--force-config` is specified. • Backs up any existing 'production' environment prior to creating a new one. • Creates/updates `/etc/puppetlabs/puppet/autosign.conf`. • Updates the following Puppet settings: `digest_algorithm`, `keylength`, `server`, `ca_server`, `ca_port`, and `trusted_server_facts`.

simp config Output

At the end of `simp config`, when you answer `yes` to `Ready to apply?`, `simp config` generates three files:

1. **`/root/.simp/simp_conf.yaml`:**
   File containing all your `simp config` settings; can include additional settings related to ones you entered and other settings required for SIMP.
2. **`/etc/puppetlabs/code/environments/production/data/simp_config_settings.yaml`:**
   File containing global Hiera data relevant to SIMP clients and the SIMP server.
3. **`/etc/puppetlabs/code/environments/production/data/hosts/<server_fqdn>.yaml`:**
   The SIMP server's host-specific Hiera configuration.

If you terminate `simp config` before applying the configuration, it will generate a file `/root/.simp/.simp_conf.yaml`. This interim file (aka the *safety-save* file), contains the answers up until the point you terminated `simp config`.

4.4.3 Client Management

This chapter provides guidance to install and configure SIMP clients with the resources supplied by the SIMP installation.

This guide also assumes that your SIMP server is a *yum* package repository and that you are configuring the `production` *SIMP Omni-Environment*.

System Requirements

Client systems should meet the following minimum requirements:

- Hardware/*Virtual Machine* (VM): Capable of running RHEL 7 or 8 x86_64
- RAM: 2048 MB
- HDD: 22 GB

Configuring the Puppet Server

Perform the following actions as `root` on the *Puppet Server* **prior** to attempting to install a client.

Configure DNS

In SIMP, numerous and/or large configuration files are distributed via **`rsync`** by Puppet to minimize management cost. These managed files presently include DNS configuration files and can be found at `/var/simp/environments/production/rsync/OSTYPE/MAJORRELEASE/bind_dns/default`.

This section is not a complete manual for named. For more complete documentation on how to set up **`named`**, see `named(8)` and `named.conf(5)`.

The following configuration steps are for a SIMP-managed setup. However, you can use an existing DNS infrastructure.

1. Navigate to `/var/simp/environments/production/rsync/OSTYPE/MAJORRELEASE/bind_dns/default`
2. Modify the **`named`** files to correctly reflect the environment.
   - The relevant files under `bind_dns/default/` are as follows:

  - named/etc/named.conf
  - named/etc/zones/your.domain
  - named/var/named/forward/your.domain.db
  - named/var/named/reverse/0.0.10.db

- Review named/etc/named.conf and update the following:
  - Update the *IP* for allow-query and allow-recursion
  - Delete any unnecessary zone stanzas (i.e. forwarding) if not necessary
  - Substitute in the *FQDN* of your domain for all occurrences of your.domain
- Add clients to named/var/named/forward/your.domain.db and named/var/named/reverse/0.0.10.db and then rename these files to appropriately match your environment.

3. Run **puppet agent -t --tags named** on the Puppet Server to apply the changes.
4. Validate DNS and ensure the /etc/resolv.conf is updated appropriately.
5. If an rndc.key error appears when starting **named**, see the BIND Documentation. Once you have resolved the issue, re-run **puppet agent -t** on the Puppet Server to apply.

Note: You can adjust the list of clients in your named/var/named/forward/<your.domain>.db and named/var/named/reverse/<your reverse domain>.db files at any time; just remember to run **puppet agent -t --tags named** on the Puppet server to propagate the updates.

Configure DHCP

Note: The dhcpd.conf file was updated in SIMP 6.2 to include logic in the pxeclients class that determines the appropriate boot loader file on the TFTP server, based on whether the client is booting in *UEFI* or *BIOS* mode. If you have configured DHCP using an earlier version of SIMP and need to add UEFI support, make sure you update your dhcpd.conf in the rsync directory, appropriately.

MAC addresses in the following section need to be lower case letters.

Perform the following actions as root on the Puppet Server system prior to attempting to install a client.

Open /var/simp/environments/production/rsync/<OSTYPE>/Global/dhcpd/dhcpd.conf and edit it to suit the necessary environment. Make sure the following is done:

- The next-server setting in the pxeclients class block points to the IP Address of the *TFTP* server.
- Create a subnet block and edit the following:
  - Make sure the routers, subnet-mask, and netmask are correct for your environment.
  - Enter the hardware ethernet and fixed-address for each client that will be kickstarted. For increased security, it is suggested that SIMP environments not allow clients to pick random IP Address in a subnet. The MAC address must be associated with and IP Address here. (You can add additional ones as needed.)
  - Enter the domain name for option domain-name
  - Enter the IP Address of the DNS server for option domain-name-servers

  - If this DHCP server is used for PXE booting, make sure each `filename` parameter corresponds to the correct boot loader file on the TFTP server. If you are using SIMP's `simp::server::kickstart` class to manage the TFTP server, the default `filename` values listed in the `pxeclients` class of the sample `dhpcd.conf` will be correct.

Save and close the file.

Run **`puppet agent -t`** on the Puppet Server to apply the changes.

4.4.4 Configure PXE Boot

Sample kickstart templates have been provided in the `/var/www/ks` directory on the SIMP server and on the SIMP DVD under `/ks`. Pre-boot images are located in the DVD under `/images/pxeboot`. If you have an existing *Preboot Execution Environment* (PXE) setup you can use these to PXE a SIMP client. Follow your own sites procedures for this.

In this section we describe how to configure the Kickstart and *TFTP* servers to PXE boot a SIMP client. (The DNS and DHCP server setup, also required for PXE booting, are discussed in an earlier chapter.)

Note: This example sets up a PXE boot for a system that is the same OS as the SIMP Server. If you are setting up a PXE boot for a different OS then you must make sure that the OS packages are available for all systems you are trying to PXE boot through YUM. There are notes throughout the instructions to help in setting multiple OS but they are not comprehensive. You should understand DHCP, KS, YUM and TFTP relationships for PXE booting before attempting this.

Setting up Kickstart

This section describes how to configure the kickstart server.

1. Add the Kickstart Server Profile

   - In the Puppet server-specific *Hiera* file (by default located at `/etc/puppetlabs/code/environments/production/data/hosts/puppet.`*`your.domain`*`.yaml`), add the `simp::server::kickstart` class.

   ```
   ---
   simp::classes:
     - simp::server::kickstart
   ```

   - This profile class adds management of DHCP, DNS, the Kickstart service, as well as the example provisioning script.

   - After adding the above class, run puppet: **`puppet agent -t`**.

2. Locate the following files in the `/var/www/ks` directory

   - `pupclient_x86_64.cfg`: Example client kickstart configuration script.

   - `diskdetect.sh`: Example script to determine disks available on a system and then apply disk configuration. This script is used by `pupclient_x86_64.cfg`.

3. Open the `pupclient_x86_64.cfg` file and follow the instructions provided within it to replace the variables listed and to customize for *BIOS*/*UEFI* boot and/or FIPS/non-FIPS mode. If you have servers that require different boot mode or FIPS options, you will need to make customized copies of this file to provide those distinct configurations. You will also have to configure TFTP to point to the appropriate files.

   - Instructions are provided both at the top of the file and throughout the body of the file.

- You need to know the IP Addresses of the YUM, Kickstart, and TFTP servers. (They default to the SIMP server in **simp config**).
- Use the commands described in the comments at the top of the file to generate the root and GRUB passwords hashes. Be sure to replace `password` with your root password.
  - The GRUB password can also be managed later with the simp/simp_grub module. See *Managing GRUB Users* for more information.
- Follow the instructions throughout the file to customize for BIOS/UEFI boot.
- Follow the instructions throughout the file to customize for FIPS/non-FIPS mode.

4. Open the `diskdetect.sh` script and customize the disk device names and/or partitions as appropriate for your site. The sample `diskdetect.sh` script will work, as is, for most systems, as long as your disk device names are in the list. In addition, the sample script provides STIG-compliant partitioning.
5. Type **chown root.apache /var/www/ks/*** to ensure that all files are owned by `root` and in the `apache` group.
6. Type **chmod 640 /var/www/ks/*** to change the permissions so the owner can read and write the file and the `apache` group can only read.

Note: Two major changes were made to `pupclient_x86_64.cfg` in SIMP 6.2:

- UEFI PXE support was added.
- To address timeout issues that caused Puppet bootstrap failures, the use of the **runpuppet** script to bootstrap Puppet on the client was replaced with the use of two scripts, both provided by the `simp::server::kickstart` class:
  - A **systemd** unit file
  - A common bootstrap script (`bootstrap_simp_client`) used by both.

Note: The URLs and locations in the file are set up for a default SIMP install. That means the same OS and version as the SIMP server, all servers in one location (on the SIMP server) and in specific directories. If you have installed these servers in a different location than the defaults, you may need to edit URLs or directories.

Note: If you want to PXE boot more than this operating system, make a copy of these files, name them appropriately and update URLS and links inside and anything else you may need. (You must know what you are doing before attempting this.) If you are booting more than one OS you must also make sure your YUM server has the OS packages for the other OSs. By default, the YUM server on SIMP has the packages only for the version of OS installed on the SIMP server.

Setting up TFTP

This section describes the process of setting up static files and manifests for *TFTP*. The example in this section assumes we are configuring the `production` *SIMP Omni-Environment*.

Note: The tftp root directory was changed in SIMP 6.2 to conform to DISA STIG standards. In previous versions it was `/tftpboot`, and in 6.2 and later it is `/var/lib/tftpboot`. If you are upgrading to 6.2 from a prior release and wish the files to remain in the `/tftpboot` directory, set `tftpboot::tftpboot_root_dir` to `/tftpboot` in *Hiera*.

Static Files

Verify the static files are in the correct location:

1. cd into `/var/simp/environments/production/rsync/`*`OSTYPE`*`/Global/tftpboot/`

   (`{OSTYPE}` under rsync is the OS type of the SIMP **server**.)

2. Verify there is a `linux-install` directory and cd to this directory.

3. Under the linux-install directory you should find a directory named *`OSTYPE-MAJORRELEASE.MINORRELEASE-ARCH/`* and a **relative**-link to this directory named *`OSTYPE-MAJORRELEASE-ARCH/`*.

4. Under *`OSTYPE-MAJORRELEASE.MINORRELEASE-ARCH/`*, you should find the files:

   - `initrd.img`
   - `vmlinuz`

   If these files are not where they should be, then create the directories as needed and copy the files from `/var/www/yum/<OSTYPE>/<MAJORRELEASE>/<ARCH>/images/pxeboot/` or from the images directory on the SIMP DVD. The link name is what is used in the resources in the `tftpboot.pp` manifest examples.

   Note: The images in `tftpboot/` need to match the distribution. For example, if you upgrade your repo from CentOS 7.3 to 7.4 and will be using this repo to kickstart machines, you must also upgrade the images in `tftpboot/`. If they do not match you may encounter errors, such as `unknown file system type 'xfs'`.

5. Next, you need to set up the boot files for either BIOS boot mode, UEFI mode, or both.

   For more information, see the RedHat 7 Installation Source or RedHat 8 Installation Source Installation Guides.

   > **Note:** UEFI support has been automated since SIMP 6.2. If you are using an older version of SIMP please refer to that documentation for setting up UEFI manually.

Dynamic Linux Model Files

Create a *site profile* module for the TFTP server on the *Puppet Server* to set up the various files to model different systems.

1. Create the file `tftpboot.pp` in your *site profile*. This file will contain Linux models for different types of systems and a mapping of MAC addresses to each model.

   Use the source code example below. Linux model examples are given for CentOS 7 and 8 using both UEFI and BIOS boot mode.

   - Replace `KSSERVER` with the IP address of kickstart server (or the code to look up the IP Address using *Hiera*).
   - Replace `OSTYPE`, `MAJORRELEASE` and `ARCH` with the correct values for the systems you will be PXE booting.
   - `MODEL NAME` is usually of the form `OSTYPE-MAJORRELEASE-ARCH` for consistency.
   - You will need to know what kickstart file you are using. UEFI and BIOS mode require separate kickstart files. Other things that might require a different kickstart file to be configured are disk drive configurations and FIPS configuration. Create a different Linux model file for each different kickstart file needed.

   Note: If using the default cfg files, know that they do not have the `_el[8,7]` tags at the end of their names.

   Note: The `simp_disk_crypt` option shown below switches on transparent disk encryption as described in the *Disk Encryption* documentation and is recommended if you have a requirement for disk encryption and cannot enter a password at system boot time.

   Simply omit the option if you do not wish to use the capability.

```
class site::tftpboot {
  include '::tftpboot'

  #--------
  # BIOS MODE MODEL EXAMPLES

  # for CentOS/RedHat 7 Legacy/BIOS boot
  tftpboot::linux_model { 'el7_x86_64':
    kernel => 'OSTYPE-MAJORRELEASE-ARCH/vmlinuz',
    initrd => 'OSTYPE-MAJORRELEASE-ARCH/initrd.img',
    ks     => "https://KSSERVER/ks/pupclient_x86_64_el7.cfg",
    extra  => "inst.noverifyssl ksdevice=bootif simp_disk_crypt\nipappend 2"
  }

  # For CentOS/RedHat 8 Legacy/BIOS boot
  tftpboot::linux_model { 'el8_x86_64':
    kernel => 'OSTYPE-MAJORRELEASE-ARCH/vmlinuz',
    initrd => 'OSTYPE-MAJORRELEASE-ARCH/initrd.img',
    ks     => "https://KSSERVER/ks/pupclient_x86_64_el8.cfg",
    extra  => "inst.noverifyssl ksdevice=bootif simp_disk_crypt\nipappend 2"
  }

  #------
```

(continues on next page)

(continued from previous page)

```
  # UEFI MODE MODEL EXAMPLES

  # NOTE UEFI boot uses the linux_model_efi module and has different
  # `extra` arguments. You also would use a different kickstart file
  # because the bootloader command within the kickstart file is
  # different. Read the instructions in the default pupclient_x86_64.cfg
  # file and make sure you have the correct bootloader line.
  #
  # For CentOS/RedHat 7 UEFI boot
  tftpboot::linux_model_efi { 'el7_x86_64_efi':
    kernel => 'OSTYPE-MAJORRELEASE-ARCH/vmlinuz',
    initrd => 'OSTYPE-MAJORRELEASE-ARCH/initrd.img',
    ks     => "https://KSSERVER/ks/pupclient_x86_64_efi_el7.cfg",
    extra  => "inst.noverifyssl simp_disk_crypt"
  }

  # For CentOS/RedHat 8 UEFI boot
  tftpboot::linux_model_efi { 'el8_x86_64_efi':
    kernel      => 'OSTYPE-MAJORRELEASE-ARCH/vmlinuz',
    initrd      => 'OSTYPE-MAJORRELEASE-ARCH/initrd.img',
    ks          => "https://KSSERVER/ks/pupclient_x86_64_el8.cfg",
    extra       => "inst.noverifyssl simp_disk_crypt"
  }

  #------
  # DEFAULT HOST BOOT CONFIGURATION EXAMPLES

  # If desired, create defaults boot configuration for BIOS and UEFI.
  # Note that the name of the default UEFI configuration file needs
  # to be 'grub.cfg'.
  tftpboot::assign_host { 'default': model => 'el7_x86_64' }
  tftpboot::assign_host_efi { 'grub.cfg': model => 'el7_x86_64_efi' }


  #------
  # HOST BOOT CONFIGURATION ASSIGNMENT EXAMPLES

  # For each system define what module you want to use by pointing
  # its MAC address to the appropriate model. Note that the MAC
  # address is preceded by ``01-``.
  tftpboot::assign_host { '01-aa-ab-ac-1d-05-11': model => 'el7_x86_64' }
  tftpboot::assign_host_efi { '01-aa-bb-cc-dd-00-11': model => 'el7_x86_64_efi'␣
↪}
}
```

2. Add the `site::tftpboot` class on your puppet server node via *Hiera*. Create the file (or edit if it exists): `/etc/puppetlabs/code/environments/production/data/hosts/<tftp.server.fqdn>.yaml`. (By default, the TFTP server is the same as your puppet server so it should exist.) Add the following example code to that yaml file.

```
---
simp::classes:
```

(continues on next page)

(continued from previous page)

```
  - 'site::tftpboot'
```

3. After updating the above file, type **`puppet agent -t --tags tftpboot`** on the Puppet Server.

   Note: To provide PXE boot configuration for more OSs, create, in the `tftpboot.pp` file, a `tftpboot::linux_model` or `tftpboot::linux_model_efi` block for each OS type. Then, assign individual hosts to each model by adding `tftpboot::assign_host` or `tftpboot::assign_host_efi` resources.

4. Finally, make sure DHCP is set up correctly. In SIMP 6.2 the example `dhcpd.conf` was updated to determine the appropriate boot loader file to use, depending upon the boot mode of the PXE client. These changes are needed if you are booting UEFI systems.

For more information see the RedHat 8 PXE or RedHat 7 PXE Installation Guides.

4.4.5 Apply Certificates

All clients in a SIMP system should have *Public Key Infrastructure* (PKI) keypairs generated for the server. These are the referred to as the infrastructure or server keys. These certificates are used to encrypt communication and identify clients and are used by common applications such as LDAP and Apache.

Note: These keypairs are **not** the keys that the Puppet server uses for its operation. Do not get the two confused.

See *Certificate Management* for more information.

SIMP uses the simp/pki module to help distribute infrastructure keypairs. The global variable, `simp_options::pki` determines what parts of the module are included. It can be overridden in hiera data at several levels if different hosts or applications need to handle certificates differently.

`simp_options::pki` can have one of three settings:

1. `simp` - Keypairs are distributed from a central location on the Puppet server to the `/etc/pki/simp/x509` directory on the client. Any applications using them will then make a copy in `/etc/pki/simp_apps/<app name>/x509` with the correct permissions for an application to use.
2. `true` - Applications on the clients will copy the keypairs from a local directory on the client to `/etc/pki/simp_apps/<app name>/x509`. The default local directory to copy from is `/etc/pki/simp/x509` but this can be overridden by setting the `simp_options::pki::source` variable.
3. `false` - The user will have to manage keypairs themselves. You will need to look at each module that uses PKI on a client to determine what variables need to be set.

   Note: A setting of `false` does **not** disable the use of PKI in a module.

The following sections describe how to populate the central key distribution directory that :pupmod:'simp/pki` uses, when `simp_options::pki` is set to `simp`.

Installing Official Certificates

This section describes how to install infrastructure certificates from an official certificate authority on the *SIMP Server* for distribution to client servers. You need to have `simp_options::pki` set to `simp` on the client for this to work.

The key distribution directory on the SIMP Server is the `site_files/pki_files/files/keydist` sub-directory located under the *SIMP Secondary Environment*, `/var/simp/environments/environment`. Within the `keydist/` directory, the SIMP system expects there to be:

- A directory named `cacerts/` that contains the *CA* public certificates.
- Client-specific directories, each of which contains the public and private certificates for an individual client. The name of each client directory must be the `certname` of that client, which by default is the client's FQDN.

Here is an example key distribution directory for a `production` *SIMP Omni-Environment*:

```
/var/simp/environments/production/site_files/pki_files/files/keydist/cacerts/
/var/simp/environments/production/site_files/pki_files/files/keydist/cacerts/cacert_
↪a7a23f33.pem
/var/simp/environments/production/site_files/pki_files/files/keydist/cacerts/cca9a35.0
/var/simp/environments/production/site_files/pki_files/files/keydist/mycomputer.my.
↪domain/
/var/simp/environments/production/site_files/pki_files/files/keydist/mycomputer.my.
↪domain/mycomputer.my.domain.pem
/var/simp/environments/production/site_files/pki_files/files/keydist/mycomputer.my.
↪domain/mycomputer.my.domain.pub
/var/simp/environments/production/site_files/pki_files/files/keydist/yourcomputer.your.
↪domain/
/var/simp/environments/production/site_files/pki_files/files/keydist/yourcomputer.your.
↪domain/yourcomputer.your.domain.pem
/var/simp/environments/production/site_files/pki_files/files/keydist/yourcomputer.your.
↪domain/yourcomputer.your.domain.pub
```

To install official certificates on the SIMP Server, do the following:

1. Copy the certificates received from a proper *CA* to the SIMP server.
2. Add the certificates for the node to the key distribution directory in `site_files/`.

   a) Make the directory under the key distribution directory for the client's certificates using the client's `certname`.

   b) Copy the official public and private certificates to that directory.

   For example, to install certificates for a system named `mycomputer.my.domain` into the `production` environment:

   ```
   mkdir -p /var/simp/environments/production/site_files/pki_files/files/keydist/
   ↪mycomputer.my.domain
   mv /dir/where/the/certs/were/myprivatecert.pem \
      /var/simp/environments/production/site_files/pki_files/files/keydist/mycomputer.
   ↪my.domain/mycomputer.my.domain.pem
   mv /dir/where/the/certs/were/mypubliccert.pub \
      /var/simp/environments/production/site_files/pki_files/files/keydist/mycomputer.
   ↪my.domain/mycomputer.my.domain.pub
   ```

3. Create and populate the *CA* certificates directory.

   a) Make the CA directory, `cacerts/`.

b) Copy the root CA public certificates into `cacerts/` in *PEM* format, one per file.

```
cd /var/simp/environments/production/site_files/pki_files/files/keydist
mkdir cacerts
cd cacerts
for file in *.pem; do ln -s $file `openssl x509 -in $file -hash -noout`.0; done
```

4. Make sure the permissions are correct.

```
chown -R root.puppet /var/simp/environments/production/site_files/pki_files/files/
↪keydist
chmod -R u=rwX,g=rX,o-rwx /var/simp/environments/production/site_files/pki_files/
↪files/keydist
```

Note: The `site_files/` sub-directory of the SIMP Secondary Environment is configured as another module path in each *Puppet Environment's* `environment.conf`. For example, for the `production` environment, `/etc/puppetlabs/code/environments/production/environment.conf` would contain:

```
modulepath = modules:/var/simp/environments/production/site_files:$basemodulepath
```

Generating Infrastructure Certificates from the Fake CA

The "Fake" (self-signing) Certificate Authority (Fake CA) is provided by SIMP as a way to manage server certificates if official certificates could not be obtained at the time of client installation or the servers are operating in testing environments.

Note: This option should not be used for any operational system that can use proper enterprise PKI certificates.

Below are the steps to generate the certificates using the SIMP-provided, Fake CA. These steps assume the `production` environment.

1. cd to `/var/simp/environments/production/FakeCA/`
2. Run **`vi togen`**

   a) Remove old entries from the file and add the *Fully Qualified Domain Name* (FQDN) of the systems (one per line) for which certificates will be created.

   Note: To use alternate DNS names for the same system, separate the names with commas and omit any spaces.

   For example, `.name,alt.name1,alt.name2.`

3. Run **`wc cacertkey`**

   a) Verify that the `cacertkey` file is not empty.

   b) If it is empty: enter text into the file, then save and close the file.

4. Run **`./gencerts_nopass.sh`**

Warning: If the **clean.sh** command is run after the certificates have been generated, you will not be able to generate new host certificates under the old CA. To troubleshoot certificate problems, see the *Troubleshooting Certificate Issues* section.

If issues arise while generating keys, navigate to the `/var/simp/environments/production/FakeCA/` directory, then type **./clean.sh** to start over.

After running the **clean.sh** script, type **./gencerts_nopass.sh** to run the script again using the previous procedure table.

The certificates generated by the FakeCA in SIMP are set to expire annually. To change this, edit the following files with the number of days for the desired lifespan of the certificates:

- `/var/simp/environments/production/FakeCA/CA`
- `/var/simp/environments/production/FakeCA/ca.cnf`
- `/var/simp/environments/production/FakeCA/default_altnames.cnf`
- `/var/simp/environments/production/FakeCA/default.cnf`
- `/var/simp/environments/production/FakeCA/user.cnf`

In addition, any certificates that have already been created and signed will have a config file containing all of its details in `/var/simp/environments/production/FakeCA/output/conf/`.

Important: Editing any entries in the above mentioned config files will **not** affect existing certificates. Existing certificates must be regenerated if you need to make changes.

The following is an example of how to change the expiration time from one year (the default) to five years for any newly created certificate:

```
for file in $(grep -rl 365 /var/simp/environments/production/FakeCA/)
do
   sed -i 's/365/1825/' $file
done
```

4.4.6 Setting up the Client

Existing Clients

Cloud environments, such as AWS, Azure, OpenStack, and GCE do not need to follow the PXE model shown below. Likewise, pre-existing physical clients can be integrated into the SIMP environment using the method outlined in this section.

The SIMP system contains a bootstrap script that is able to be downloaded from the server. You should examine the actual client application to determine if it meets your needs as written but, in general, it should be well suited most applications.

The following invocation waits for the server to provide a signed PKI certificate prior to proceeding. This is the safest method but will hang if the client certificate is not signed.

```
curl -k -O https://<puppet.server.fqdn>/ks/bootstrap_simp_client

# Use the puppet provided ruby for a guaranteed compatible version
```

(continues on next page)

(continued from previous page)

```
/opt/puppetlabs/puppet/bin/ruby ./bootstrap_simp_client \
  --puppet-server <puppet.server.fqdn> \
  --puppet-ca <puppet.server.fqdn> \
  --puppet-wait-for-cert 0 \
  --debug
  --print-stats
```

PXE Booting

The following lists the steps to *PXE* boot the system and set up the client.

1. Set up your client's boot settings to boot off of the network.
2. Make sure the *MAC* address of the client is set up in *DHCP* (see *Configure DHCP* for more info.)
3. Restart the system.
4. Once the client installs, reboots, and begins to bootstrap, it will check in for the first time.
5. By default, Puppet will not autosign Puppet certificates, so the agent initially runs with `--waitforcert` enabled. This means the client will check in every 30 seconds for a signed certificate. Log on to the Puppet master and run `puppetserver ca sign --certname <puppet.agent.fqdn>`.

Upon successful deployment of a new client, it is highly recommended that *LDAP administrative accounts* be created.

Troubleshooting Puppet Issues

If the client has been kickstarted, but is not communicating with the Puppet master, try the following options:

- Check the forward and reverse *DNS* entries on the client and server; both must be correct. The **nslookup** command will help here.
- Check the time on the systems. More than an hour's difference will cause serious issues with certificates.
- Remove `/etc/puppetlabs/puppet/ssl` on the client system; run **puppetserver ca clean --certname <CLIENT.FQDN>** on the Puppet Server and try again.

If you are getting permission errors, make sure the SELinux context is correct on all files, as well as the owner and group permissions.

Troubleshooting Certificate Issues

If host certificates do not appear to be working, ensure that all certificates verify against the installed *CA* certificates.

The table below lists the steps to determine which certificates are working and which are not.

1. Navigate to `/var/simp/environments/production/site_files/pki_files/files/keydist/`
2. Run **find . -name "*<YOUR.DOMAIN>.pub" -exec openssl verify -CApath cacerts {} ;**

   The screen displays `./<Host Name>.<Your.Domain>/<Hostname>.<Your.Domain>.pub: OK` If anything other than OK appears for each host, analyze the error and ensure that the CA certificates are correct.

   If the `TXT_DB` error number **2** appears, revoke the certificate that is being regenerated. The table below lists the steps to revoke the certificate.
3. Navigate to the directory containing the CA certificates. For the FakeCA, it is `/var/simp/environments/production/FakeCA/`. The directory should contain the file `default.cnf`.

4. Run

```
OPENSSL_CONF=default.cnf openssl ca -revoke /var/simp/environments/production\
/site_files/pki_files/files/keydist/*<Host to Revoke>*/*<Host to Revoke>*.pub
```

4.4.7 SIMP Administration

This chapter provides guidance on general administration practices applicable to SIMP environments.

Important: It is important to remember that SIMP approaches system configuration from a least privilege, defense in depth, mindset where possible.

Classification and Data

Node Classification in SIMP

From the Puppet, Inc. website:

> Hiera is a key/value lookup tool for configuration data, built to set node-specific data without repeating yourself.

SIMP uses *Hiera* to attempt to make configuration of the overall system easier for our end users by providing a simple, centralized, method for setting class parameters using automatic parameter lookup.

It is **highly recommended** that you read the Hiera Documentation prior to jumping into using a SIMP system.

Hiera in SIMP

SIMP users are expected to make extensive use of Hiera to set parameters, particularly those that are deep within the code.

The default Hiera hierarchy used by SIMP defined at the environment level in the Hiera configuration file and looks like the following:

```
---
version: 5
defaults:
  datadir: data
  data_hash: yaml_data

hierarchy:

  - name: Per-node data
    paths:
    - "hosts/%{trusted.certname}.yaml"
    - "hosts/%{facts.fqdn}.yaml"
    - "hosts/%{facts.hostname}.yaml"

  - name: Per-domain data
    paths:
    - "domains/%{facts.domain}.yaml"
```

(continues on next page)

(continued from previous page)

```
- name: Per-OS data
  paths:
  - "%{facts.os.family}.yaml"
  - "%{facts.os.name}/%{facts.os.release.full}.yaml"
  - "%{facts.os.name}/%{facts.os.release.major}.yaml"
  - "%{facts.os.name}.yaml"

- name: Per-hostgroup data - See site.pp for more information
  paths:
  - "hostgroups/%{hostgroup}.yaml"

- name: General data
  paths:
  - "default.yaml"
  - "common.yaml"

- name: SIMP specific data - Please do not modify
  paths:
  - "simp_config_settings.yaml"
  - "scenarios/%{simp_scenario}.yaml"
```

Warning: This may not be accurate for your version of SIMP, please check your local Hiera settings!

The rest of this document will use this hierarchy as a reference.

Assigning Classes to Nodes

The SIMP profile module includes other classes needed for a secure baseline, which are discussed below in the *SIMP scenarios* section. The lists of included classed are controlled by the `simp::classes` and `simp::server::classes` class parameters:

- The `simp::classes` parameter is available for all nodes.
- The `simp::server::classes` parameter is available only for the *SIMP Server*.

To preserve various levels of ordering and overrides, it is highly recommended that you use the `simp::classes` and `simp::server::classes` class parameters in *Hiera* to manage standard class inclusions.

This allows you to use the `--` knockout prefix to exclude classes at any level of your hierarchy, which, in turn, allows you to use your own modules to manage specific applications that SIMP manages by default.

Listing 1: Example: Adding a Class to Standard List

```
---
simp::classes:
  - gnome
```

Listing 2: Example: Removing a Class from Client Nodes

```
---
simp::classes:
  - '--ntpd'
```

Listing 3: Example: Removing a Class from Client Nodes and the SIMP Server

```
---
simp::classes:
  - '--ntpd'

simp::server::classes:
  - '--ntpd'
```

Legacy Method

SIMP also includes class inclusion functionality via a top-level `classes` array in your Hiera hierarchy. Though this is no longer recommended, it is still supported.

In this case, instead of using the knockout prefix, there is also a `class_exclusions` array that will be used to remove classes from the include list. Note, however, this array cannot be applied to any classes included via `simp::classes` or `simp::server::classes`.

Assigning Defined Types to Nodes

Defined types do not have the ability to receive parameters via Hiera in the traditional sense. To include a defined type on a node, one could use `create_resources`, but this is messy and discouraged. Instead, create your own profile or a `site` module with site-specific manifests. For example, `/etc/puppetlabs/code/environments/production/modules/site/manifests/my_site.pp`.

Note: You can find a working example of this in the *Configure PXE Boot* section of the documentation

SIMP Scenarios

SIMP scenarios are groups of classes, settings, and simp_options that ensure the system is compliant and secure.

There are currently four SIMP scenarios:

- *simp*
- *simp_lite*
- *poss*
- *remote_access*

The *simp* scenario includes all security features enabled by default, including iptables and svckill. This scenario is what stock SIMP used to look like in previous releases.

The *simp_lite* scenario offers many security features, with a few explicitly turned off. This scenario was designed to make it easier to implement SIMP in an existing environment, because it might not be trivial to flip SELinux to Enforcing on all nodes.

The *poss* option is the bare-bones option. It only includes the `pupmod` class, to configure Puppet agent on clients. All of the simp_options default to false, so SIMP will not do a lot of modification to clients through Puppet when using this scenario.

The *remote_access* scenario includes the SSH module and the authentication stack, namely PAM and nsswitch. This scenario is useful for those who want to retain remote access to their machine while leaving virtually everything else untouched.

These scenarios are defined in the pupmod-simp-simp module. For more details refer to this module.

Note: The *SIMP Server* is exempt from most of these settings, and will be using most features from the *simp* scenario by default. The SIMP server should only have services on it related to Puppet and systems management, and SIMP modules all work with all security features enabled. See the `puppet.your.domain.yaml` in the `data/hosts` directory for details.

The SIMP Compliance Engine

The *SIMP Compliance Engine* allows users to enable validation and enforcement of inbuilt and/or custom compliance. *SIMP Community Edition* contains freely available profiles generally maintained by the community. *SIMP Enterprise Edition* contains additional profiles targeted more at commercial regulations. Enforcement is done at the Puppet *class parameter* level and validation may be done against both *classes* and *defined resource types*.

This capability is provided by the simp-compliance_markup Puppet module and will be renamed at some point in the future to `simp-compliance_engine` for clarity.

Important: Your system must be configured to use at least *Hiera* version 5 for the Compliance Engine to function properly.

SIMP 6.3+ supports this capability out of the box but earlier versions may need to perform a Hiera migration.

Enabling the SIMP Compliance Engine

In the Puppet implementation, the Compliance Engine is a *Hiera backend* that provides data from *SIMP compliance profile* data sets for use in configuring Puppet classes via *automatic parameter lookup*.

The following example demonstrates how to enable the SIMP Compliance Engine in an example `hiera.yaml` configuration.

```yaml
---
version: 5
defaults:
  datadir: "data"
  data_hash: "yaml_data"
hierarchy:
  # Anything defined at this level of the hierarchy will override settings
  # in the SIMP Compliance Engine
  - name: "Higher Priority Items"
```

(continues on next page)

(continued from previous page)

```
    paths:
      - "mandatory.yaml"

  - name: "SIMP Compliance Engine"
    lookup_key: "compliance_markup::enforcement"

  # Anything defined at this level of the hierarchy will be overridden by
  # the SIMP Compliance Engine.
  - name: "Lower Priority Items"
    paths:
      - "optional.yaml"
```

Note: Hierarchy lookups happen from top to bottom and stop at the first match. This means that if the SIMP Compliance Engine is in enforcing mode and is at the top of the hierarchy, then it cannot be overridden except by an *ENC*. See the Hiera config file syntax page for additional information.

After configuring Hiera, you then need to include the `compliance_markup` class.

```
include 'compliance_markup'
```

Discovering Available Profiles

Prior to SIMP 6.4, all profile data was embedded into the `simp-compliance_markup` module. As of 6.4, the data has been split across the modules with each SIMP module containing the module's specific profile data in the `SIMP` directory.

Data for modules that are not maintained by the SIMP organization are still housed in `simp-compliance_markup`.

At this time, SIMP CE comes with two publicly available profiles, `disa_stig` and `nist_800_53:rev4`.

A `simp` subcommand will eventually be provided to help discover available profiles in different environments but, for now, you can use the following code snippet:

```
ruby -r yaml -e "puts Dir. \
  glob('/etc/puppetlabs/code/environments/production/modules/*/{SIMP,simp}/compliance_
↪profiles/*.yaml'). \
  map{|x| begin YAML.load_file(x); rescue; end; }.compact. \
  map{|x| x['profiles'] && x['profiles'].keys}. \
  compact.flatten.sort.uniq"
```

Profile Enforcement

The selection of profiles to enforce is based on the `lookup_key` entry in the `hiera.yaml` configuration as shown in the previous section.

In this case, the `lookup_key`, is `compliance_markup::enforcement` and should be set in Hiera as an Array of compliance profiles that you would like to apply, in priority order.

If you wanted to conform with the requirements of the *DISA STIG*, you would apply the `disa_stig` profile as shown below:

```
---
compliance_markup::enforcement:
  - "disa_stig"
```

The next run of Puppet on the nodes to which this Hiera data applies would then have their parameters set to meet the STIG requirements as mapped by parameter in the associated compliance profile.

Additional profiles may be made available by various Puppet modules, such as those provided by *SIMP EE*.

Code-Level Compliance Validation

The SIMP Compliance Engine also provides a mechanism for generating reports based on the Puppet parameters that are set at compile time and where those differ from the parameters that are expected for a particular compliance profile.

Note: Enforcement and validation are two distinct capabilities and the settings of one do not affect the other. This allows you to validate against multiple profiles independently of any particular enforcement setting.

In the following example, we want to validate against the `nist_800_53_r4` profile:

```
---
compliance_markup::validate_profiles:
  - "nist_800_53_r4"
```

By default, reports will be generated on the Puppet server and written to the **server's** `vardir` which, by default, is `/opt/puppetlabs/server/data/puppetserver/simp/compliance_reports`. Only the latest compliance report for a host is stored. The report is a YAML format that may easily be offloaded for analysis and processing and fully documented in the simp-compliance_markup Puppet module.

SIMP File Structure

The default *Puppet environment* in SIMP is located at `/etc/puppetlabs/code/environments/production` and contains almost all necessary files for a Puppet infrastructure. It will look like this:

```
/etc/puppetlabs/code/environments/production/
├── environment.conf
├── data/
├── manifests/
└── modules/
```

- `environment.conf`
  - Sets the environment to include the *SIMP Secondary Environment* modulepath.
- `manifests/`
  - Contains SIMP's `site.pp`.
- `data/`
  - Default location of the YAML files used by Hiera; contains your node data.
- `modules/`
  - Default install location of Puppet modules. Populated automatically for you by `simp config`.

The skeleton for SIMP's Puppet environment is installed by the simp-environment-skeleton RPM into /usr/share/simp/environment-skeleton/puppet. This skeleton is copied into the production Puppet environment by simp config. However, at any time, you can regenerate this skeleton *SIMP Omni-Environment* by using simp environment new.

Second Modulepath

SIMP utilizes a second modulepath in the *SIMP Omni-Environment* to ensure that deployment tools like r10k do not squash keydist and some krb5 files. For the production environment, the path is /var/simp/environments/production/site_files/. *PKI Certificates* are stored there.

Hiera

There are three Hiera Configuration Layers in Hiera 5. SIMP configures Hiera at the environment level, via /etc/puppetlabs/code/environments/<environment name>/hiera.yaml. The global level configuration file at /etc/puppetlabs/puppet/hiera.yaml still exists but its hierarchy is empty in a fresh system and probably should remain that way for general usage.

The default hiera.yaml for a SIMP environment is shown in *Hiera in SIMP* and its syntax is explained in Hiera configuration file. The corresponding Hiera data directory layout in a fresh SIMP system (or a new environment created by simp environment new) is as follows:

```
/etc/puppetlabs/code/environments/production/data/
├── default.yaml
├── hostgroups/
├── hosts/
├── scenarios/
└── simp_config_settings.yaml
```

- data/simp_config_settings.yaml
  - Contains the variables needed to configure SIMP. Added by simp config.
- data/scenarios/
  - Directory containing settings for SIMP Scenarios. The scenario selected is set in manifests/site.pp. This is done automatically for you when you run simp config.
- data/hosts/
  - Settings for specific hosts. By populating this directory with host name files, you can assign parameters to specific hosts. Based on SIMP's hiera.yaml, Hiera looks for the name in following formats:
    - %{trusted.certname}.yaml
    - %{facts.fqdn}.yaml
    - %{facts.hostname}.yaml
- data/hostgroups/
  - Settings for specific groups of hosts. The hostgroup of a node is computed using a top-scope $hostgroup variable. By default, this is set via SIMP's top-level site.pp. Then, nodes assigned to $hostgroup will read Hiera from a file named <hostgroup>.yaml in this directory.
- data/default.yaml
  - Settings that should be applied to all systems in the environment.

Important: Do not change or update the scenarios YAML files or `simp_config_settings.yaml`. If you want to change any of the settings contained in these files for one or more hosts, override the settings in Hiera files earlier in the hierarchy, instead.

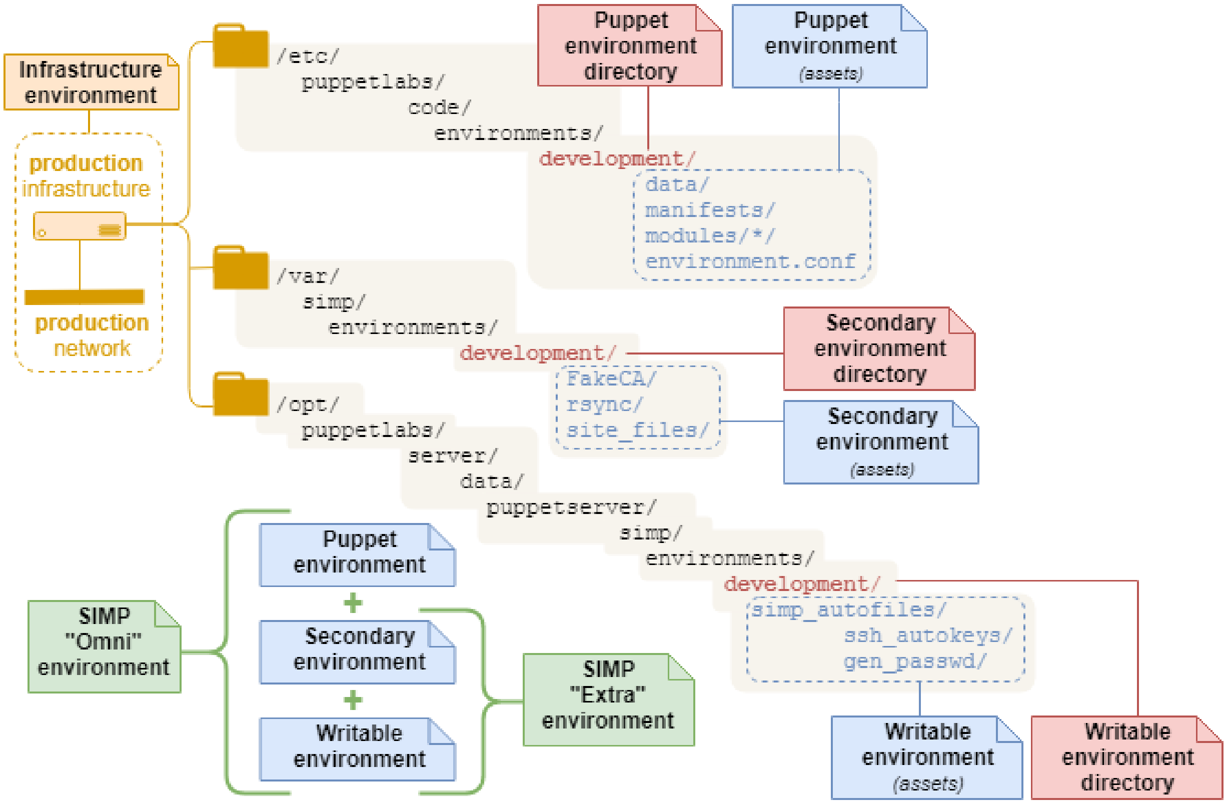
Deploying SIMP Environments

SIMP fully supports Puppet Environments, and extends the practice by providing parallel environment directories to manage data that does not belong in a control repository.

This section describes how SIMP interacts with Puppet environments, and how to integrate SIMP environments into your site's deployment strategy.

SIMP Environments

The term "environment" is heavily overloaded when discussing SIMP, Puppet, DevOps, and IT in general. We attempt to make the following distinctions:



A full SIMP "Omni" environment is composed from three corresponding environments:

Environment	Description
Puppet	- The standard concept of "Puppet environment" when Puppet/PE is operated independently from SIMP. - Acts as a namespace for managing nodes—with its own code, data, and lookup hierarchy. - Sourced (once deployed) from a Puppet environment directory
Secondary	- Provides SIMP with independently-maintained assets that support a specific Puppet environment - Useful for git-unfriendly files, such as secrets (e.g., `keydist/`) and large files distributed by `rsync`. - Sourced from a secondary environment directory
Writable	- Puppet environment-specific SIMP data (e.g., simplib::passgen() data), generated by the Puppet Server - Data is generated automatically by Puppet functions during catalog compilation - Written to/sourced from the writable environment directory

Each environment has an associated directory:

Puppet environment directory

- A single directory at `$codedir/environments/<environment_name>`
- Contains the Puppet code and Hiera data needed to manage a Puppet environment:
  - site manifest(s), Puppet modules, Hiera data (Hiera 4+), and Hiera hierarchy (Hiera 5+).
- Assets can deployed using `r10k` or Code Manager:
  - `r10k deploy` creates one Puppet environment directory per control repository branch.
  - `r10k puppetfile install` (using default settings) can be run at the top level of a Puppet environment directory to deploy its modules from the local `Puppetfile`.

Secondary environment directory

SIMP, by default, supplements each Puppet environment with a **Secondary environment directory**, located on the SIMP server at `/var/simp/environments/<environment_name>`.

This space holds all static, non-Puppet created files. It is generally used for large binary items that will be delivered via rsync, and for files that are too dangerous to add to a version control system. These include things like the SIMP rsync materials (`rsync/`), the *Infrastructure Certificates* (`site_files/pki_files/files/keydist/`), and the `krb5` Puppet module (`site_files/krb5_files/files/keytabs`).

As you add new Puppet environments, you will need to replicate the appropriate Secondary directory structure here.

Note: For more information on the SIMP rsync structure, please see *HOWTO Work with the SIMP Rsync Shares*

Writable environment directory

SIMP also maintains **Writable environment directories** in parallel with each Puppet environment at `/opt/puppetlabs/server/data/puppetserver/simp/environments/`. This space holds all non-static, Puppet **Server** created files. It is used for content that can be generated during catalog compilation, such as simplib::passgen() and `ssh::autokey()`,

Warning: You should never need to manually adjust anything in this directory space.

Building on the environments above, it is useful to consider the following "meta" environments:

Environment	Description
SIMP "Omni" environment	• The complete combination of Puppet + SIMP Extra (Secondary + Writable) environments of the same name
SIMP "Extra" environment	• A combined Secondary environment + Writable environment • (Just the SIMP environments w/o the Puppet environment)
Infrastructure Environment	• A local site's dedicated internal infrastructure tier • Exists independently from Puppet • Often named something along the lines of "development," "testing, acceptance," "staging," "production", etc., • In Puppet circles, this is sometimes referred to an infrastructure as an "application tier" or "permanent test environment"

Scaling considerations for Extra (Secondary and Writable) environment directories

Secondary and Writable environment directories pose a problem for Puppet scaling:

- Load-balanced compile servers *must* keep the contents of *both* secondary and writable environment directories in sync across all compile servers.
- Alternatively, disciplined use of Hiera Eyaml encryption must be used to replace all parameters that default to using Writable-environment functions like simplib::passgen().

Deployment Scenarios

SIMP supports deploying Puppet code using r10k or Code Manager, even when its modules are installed from RPMs. However, deployment strategies vary widely, based on a site's infrastructure, existing policies, and administrative considerations. SIMP broadly groups these strategies into high-level *deployment scenarios*, with some important variations:

Table 4: Deployment Scenarios and Variations

Deployment Scenario	Variation	r10k command	Control Repository?	Module Repos	Autodeploy?
Local	*Local module repos*	`r10k puppetfile install`	No	Local *(from RPMs)*	
Control Repo	*Manual r10k*	`r10k deploy environment`	Yes	Remote (Git hosting service)	
	Webhook + r10k	`r10k deploy environment`			✓
	PE Code Manager	`puppet code deploy`			✓

Regardless of the deployment scenario and variation employed, it is important to ensure that each Puppet environment is associated with an appropriate *SIMP Omni-environment*, or that the data provided by SIMP's "Extra" environments are handled by alternate means.

Local deployment scenario

The *Local deployment scenario* deploys Puppet **modules** from *Local Module Repositories* into an existing Puppet environment's `modules/` directory.

This scenario requires the least preparation: it doesn't require a control repository, a remote git-hosting service, or even a functioning network. It also works automatically on any monolithic SIMP server (6.4.0+) that was installed from ISO or RPM. In fact, the initial run of `simp config` uses this method to deploy the system's first modules into the production environment.

The local deployment scenario is recommended for:

- Systems that don't have access to a git-hosting service
- Users who prefer to treat the SIMP framework as a vendor-packaged appliance
- Users who prefer to avoid maintaining a control repository

It requires:

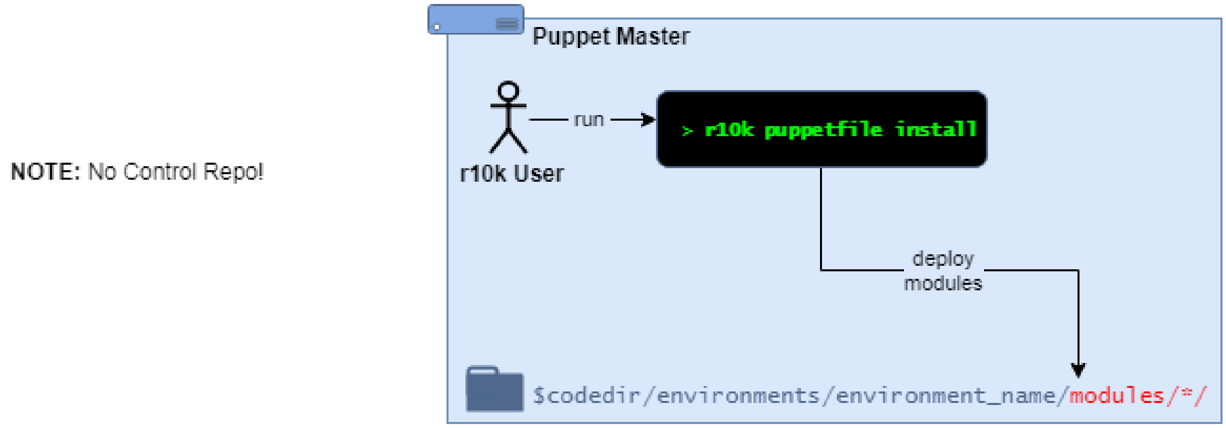
- SIMP RPM-delivered *Local Module Git Repositories*
- The `simp` command
- A pre-existing Puppet environment directory (`simp environment new`)
- A simp Puppetfile (`simp puppetfile generate`)
- No additional compile servers.

Important: The local deployment scenario is only suitable for a single monolithic SIMP server.

Note: If you need to scale beyond a single server, you should *deploy from a control repository*.

Deploying from Local module repos

This is the initial deployment scenario immediately after *Installing SIMP from an ISO*.


NOTE: No Control Repo!
Puppet Master
r10k User
run
> r10k puppetfile install
deploy modules
$codedir/environments/environment_name/modules/*/

Under this scenario, a *local* r10k user will:

- `cd` into an existing Puppet environment directory
- Use the command `r10k puppetfile install` to deploy the modules defined in the local `Puppetfile` (which may include a separate `Puppetfile.simp`)

Notes:

- The Puppet environment directory and `Puppetfile` must exist before `r10k puppetfile install` is run
- The modules in the `Puppetfile` are defined with `file://` URLs, and pull from Git repositories on the local file system

Control Repository deployment scenario

The *Control Repository deployment scenario* deploys complete Puppet **environments** from a remotely-hosted *Control Repository*.

SIMP administrators must ensure that each deployed Puppet environment is associated with an appropriate *SIMP Omni-environment* (or doesn't need to be). A good starting point is to review the article, *HOWTO Set up a SIMP Environment in a Control Repository*

The Control repository deployment scenario is recommended for:

- Environments that manage their infrastructure as code
- Infrastructures that need to scale beyond a single server (e.g., compile servers)
- Administrators who are comfortable with git and Puppet control repositories
- Puppet Enterprise users

It requires:

- A remote git-hosting service

- A (remotely-hosted) *Control Repository*
- *r10k* and/or *Code Manager*, configured to deploy from the Control Repository
- Keeping all remote Puppet module repositories (and the control repository's `Puppetfile`) up-to-date as SIMP RPMs deliver new module versions to the server's local filesystem

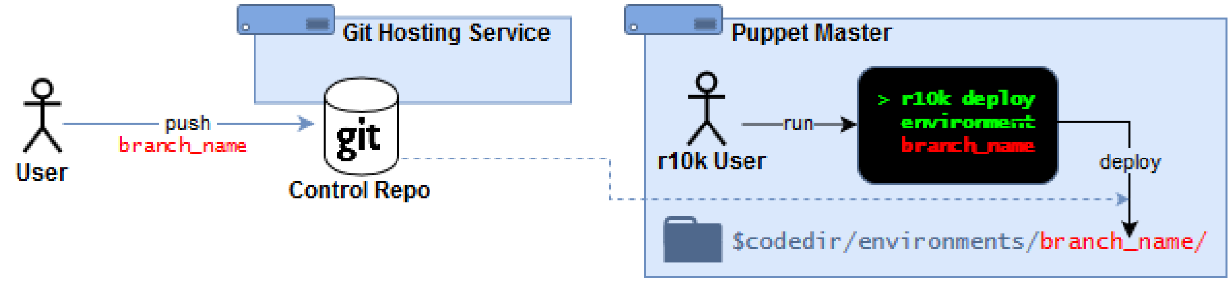
Depending on the variation, it may also require:

- (optional) a local webhook on the *Puppet Server*, configured to trigger the code deploy
- (optional) a webhook on the git-hosting service to trigger environment deployments on the Puppet Server after a control repository branch is updated

Important: Control repository deployment scenario administrators are expected to:

- Have sufficient expertise to understand and manage the requirements above
- Review and understand the documentation at *HOWTO Set up a SIMP Environment in a Control Repository*
- Understand the relationship between a deployed Puppet environment and the larger *SIMP Omni-environment*

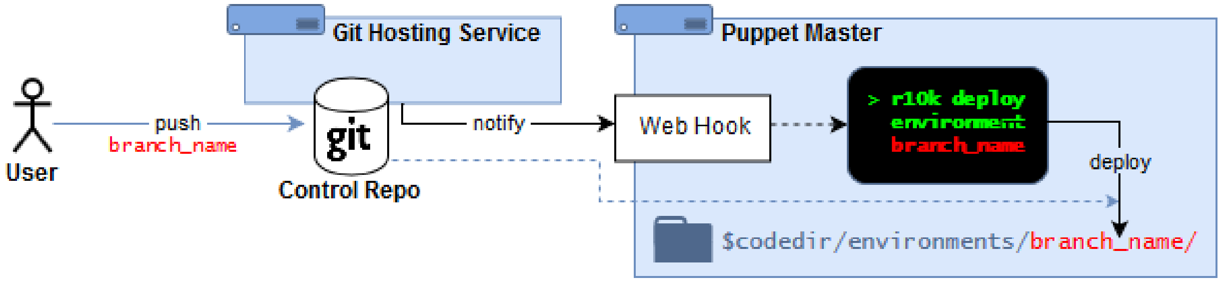
Manual `r10k deploy`



Under this scenario:

- Puppet admins maintain a Puppet control repository (and Puppet module repositories) in an external git-hosting service.
- An r10k user logs into the Puppet Server and manually deploys entire *Puppet environment* directories (one per branch) from the command line:
  - `r10k deploy environment [ENVIRONMENT]`
  - `r10k puppet code deploy [ENVIRONMENT]` (Puppet Enterprise)

Webhook-triggered `r10k deploy`



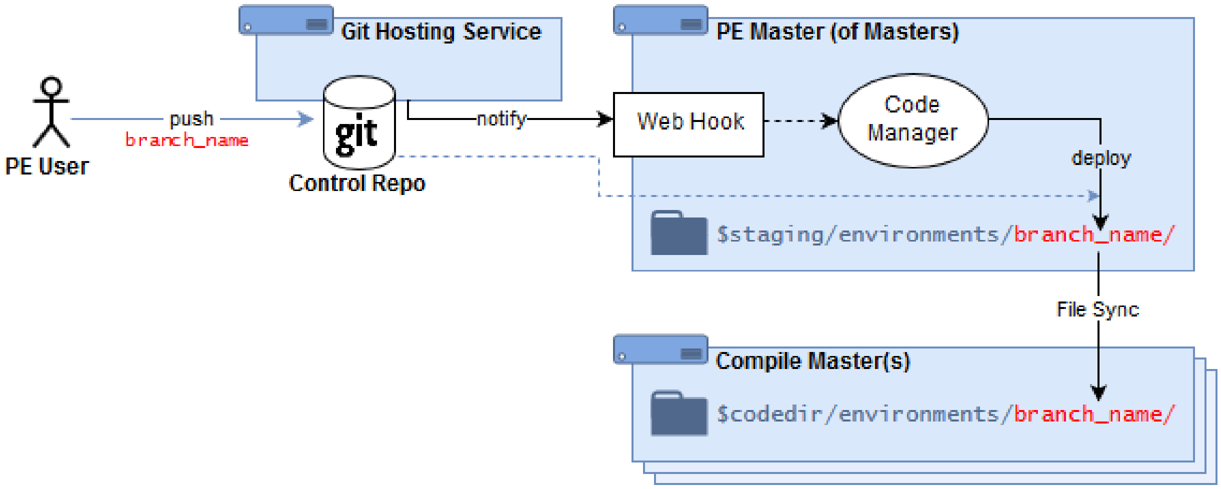
Under this scenario, Puppet admins:

- Maintain a Puppet control repository (and Puppet module repositories) in an external git-hosting service
- Deploy a Puppet environment by pushing updates to a control repository branch

When a control repository branch is updated, it automatically triggers an environment deployment on the Puppet Server:

- The Git-hosting service uses a webhook to notify the Puppet Server about the branch that was changed
- The webhook on the Puppet Server triggers r10k to deploy the branch as an environment

PE Code Manager



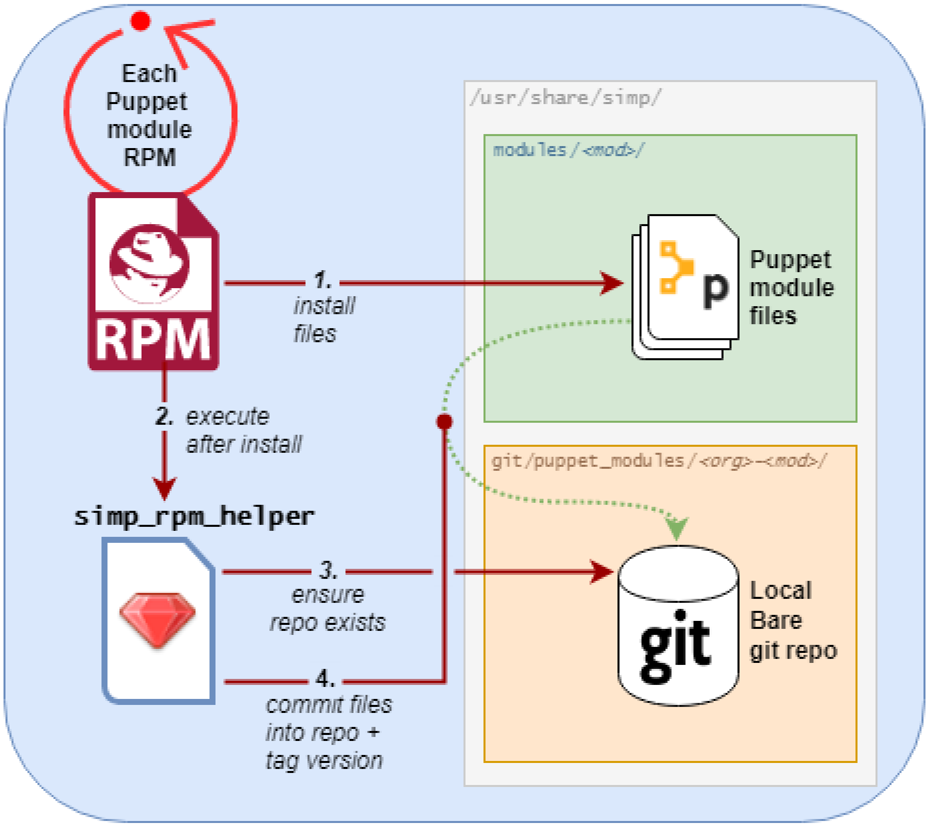
Under this scenario, Puppet admins:

- Maintain a Puppet control repository (and Puppet module repositories) in an external git-hosting service
- Deploy a Puppet environment by pushing updates to a control repository branch

When a control repository branch is updated, it automatically triggers an environment deployment on the Puppet Server:

- The Git-hosting service uses a webhook to notify the PE server/MoM about the branch that was changed
- The webhook on the Puppet Server triggers PE Code Manager to deploy the branch as a staged environment
- The PE File Sync service distributes the staged environment to any compile servers, and promotes the code from staging to live use.

Local Module Repositories



Each SIMP-packaged Puppet module RPM delivers it Puppet module content to two locations:

1. Under the directory `/usr/share/simp/modules/<module_name>/`

- These are reference files, and are recognized by package management tools like `rpm -ql` and `rpm -V`

2. A tagged commit to a local *Git* repository at `/usr/share/simp/git/puppet_modules/<org>-<module_name>/`

- This repository supports `r10k puppetfile install` and the `simp` commands, "*puppetfile generate*" and "*environment new*"
- The git repo and tags are created automatically during the RPM's %postrun

When a SIMP-packaged Puppet module RPM is:

- **Installed**, a new tag is created inside a new repository.
- **Upgraded**, a new tag is created and all previous tags are preserved.
- **Removed**, its local Git repository will remain on the filesystem (in case any Puppet environments still reference it).

The RPM's %postinstall git management is handled by the `simp_rpm_helper` script (from the simp-adapter package). It derives the repo's name (`<org>-<module_name>`) from module's `metadata.json` file. The repo name is guaranteed to be unique across all Puppet modules published to the *PuppetForge*.

Note: The `master` branch in a SIMP RPM-created Puppet module Git repository is a **local transaction history** of

RPM-generated module tags.

- It is **NOT** a copy of the module's Git development history from its public repository.
- It does **NOT** track the highest version available from the repository.
- The branch head will be whatever the most recent successful RPM %postinstall was—even if that was a *downgrade*.

Therefore: your Puppetfiles **should reference modules by tag** (and never `master`) when deploying from a SIMP RPM-created Puppet module Git repository.

SIMP CLI Commands for Environment Management

The SIMP CLI provides commands to aid you in many facets of Puppet environment management. This section will explain the basics of each command, many of which were used in *Creating and Deploying a Test Environment Using Local Module Repositories*.

Note: All SIMP CLI environment management commands need to be run as `root`.

simp environment new

`simp environment new` creates a new *SIMP Omni-Environment skeleton*. This allows sysadmins to create new, SIMP environments easily, when they need to create a new Puppet environment.

By default, `simp environment new` creates a SIMP Omni-Environment skeleton, which includes a `Puppetfile` that can be used to deploy Puppet modules from local, SIMP-managed Git repositories. Its behavior, however, is quite flexible. You can control which parts of the environment get created, (including the `Puppetfile`) with its command options. You can even have it deploy the Puppet modules using `r10k puppetfile install`, if you want it to create the full environment in one step.

`simp environment new` options are documented in its built-in SIMP CLI documentation. To see the options available, execute the following:

```
simp environment new -h
```

Default Behavior

When no options are specified, `simp environment new <env name>` takes the following actions:

1. Creates the Puppet environment `/etc/puppetlabs/code/environments/<env name>` from `/usr/share/simp/environment-skeleton/puppet`.
    - This includes creating the appropriate `environment.conf` file from the skeleton's `environment.conf.TEMPLATE`.
2. Generates the files `Puppetfile` and `Puppetfile.simp` under the new Puppet environment's directory:

    ```
    /etc/puppetlabs/code/environments/<env_name>/
        ├── Puppetfile
        └── Puppetfile.simp
    ```

    - This was equivalent to executing:

```
cd /etc/puppetlabs/code/environments/<env name>
simp puppetfile generate --skeleton > Puppetfile
simp puppetfile generate > Puppetfile.simp
```

- See "*simp puppetfile generate*" for an explanation about the benefits of having these two Puppetfiles.

3. Creates the secondary environment `/var/simp/environments/<env name>` from `/usr/share/simp/environment-skeleton/secondary`.
4. Copies files from `/usr/share/simp/environment-skeleton/rsync` into `/var/simp/environments/<env name>/rsync`.
5. If any `tftpboot` *PXE* image files can be found in directories matching `/var/www/yum/CentOS/**/x86_64/images/pxeboot`, copies those files into the appropriate sub-directory in `/var/simp/environments/<env name>/rsync/RedHat/Global/tftpboot/linux-install`.
6. Creates a new FakeCA cacertkey file at `/var/simp/environments/<env name>/FakeCA/cacertkey`. The file is populated with a random string.
7. Re-applies the *FACLs*, *SELinux* contexts, and permissions to the newly-created directory trees.
   - Equivalent to executing: `simp puppetfile fix <env name>`
8. Does not create the Writable environment, because in this scenario, that environment will be automatically generated, as needed, when `puppet` is run

simp environment fix

`simp environment fix` re-applies the *FACLs*, *SELinux* contexts, and permissions to a SIMP Omni-Environment. This can be especially useful, when you have added files to the environment and are unsure which settings are appropriate for them.

By default, `simp environment fix` operates on all directories in the SIMP Omni-Environment. However, its behavior is quite flexible. You can control which parts of the environment get fixed with its command options.

`simp environment fix` options are documented in its built-in SIMP CLI documentation. To see the options available, execute the following:

```
simp environment fix -h
```

Default Behavior

When no options are specified, `simp environment fix <env name>` , takes the following actions:

- Applies Puppet permissions (group) to `/var/simp/environments/<env name>`.
- Applies Puppet permissions (user + group) recursively under `/etc/puppetlabs/code/environments/<env name>`.
- Applies Puppet permissions (group) recursively under `/var/simp/environments/<env name>/site_files`.
- Restores SELinux contexts under `/var/simp/environments/<env name>`.
- Applies FACL rules to `/var/simp/environments/<env name>/rsync`

simp puppetfile generate

`simp puppetfile generate` creates one of two different types of *Puppetfiles*:

`Puppetfile.simp`

The *SIMP-only Puppetfile*, used to deploy SIMP's Puppet modules from local SIMP-managed Git repositories. These local repositories are created/updated when SIMP-packaged Puppet module RPMs are installed. This file can be regenerated to deploy the version of each module under `/usr/share/simp/modules` using the command:

```
simp puppetfile generate > Puppetfile.simp
```

`Puppetfile`

The *Parent Puppetfile*, which deploys all modules from *both* Puppetfiles. Administrators can customize this file to declare any additional (non-SIMP) modules they may need. It also includes the content from the SIMP-only Puppetfile from the same directory. A fresh parent Puppetfile can be generated by running:

```
simp puppetfile generate --skeleton > Puppetfile
```

The files are kept side-by-side at the top level of each Puppet environment directory:

```
<puppet_environment>/
├── Puppetfile.simp
└── Puppetfile
```

`simp puppetfile generate` generates these two files separately so administrators can manage any non-SIMP modules in the *Puppetfile*, while permitting simp to regenerate `Puppetfile.simp` as the SIMP-only modules are updated. *r10k* will deploy all the modules in both Puppetfiles.

If you want to update SIMP-only Puppetfile entries after installing/upgrading SIMP module RPMs, all you have to do is to execute:

```
simp puppetfile generate > <path to your environment>/Puppetfile.simp
```

Any Puppet modules defined in the parent Puppetfile will be unaffected.

For example, to update the SIMP-only Puppetfile created by `simp config` in a standard SIMP server install from SIMP ISO or SIMP RPMs, execute:

```
cd /etc/puppetlabs/code/environments/production
simp puppetfile generate > Puppetfile.simp
```

Default Behavior: Generating Latest SIMP-only Puppetfile Content

When no options are specified, `simp puppetfile generate`, prints to the console a Puppetfile that could be used to deploy the current set of SIMP Puppet modules installed via RPM in `/usr/share/simp/modules` from the corresponding local, SIMP-managed Git repositories in `/usr/share/simp/git/puppet_modules`.

Each Puppetfile entry will look like the following:

```
mod 'simp-simplib',
  :git => 'file:///usr/share/simp/git/puppet_modules/simp-simplib.git',
  :tag => '3.15.2'
```

The module name will be the name of the Puppet module in its `metadata.json` file (`<Puppet Forge org>-<module name>`), the Git URL will be its local, SIMP-managed Git repository, and the Git version will be the latest tag in that local repository.

SIMP chose the naming convention to match that of *PuppetForge*.

Note: The corresponding SIMP-packaged Puppet module RPM name is

`pupmod-<Puppet Forge org>-<module name>`

Generating a Parent Puppetfile Skeleton

When the `--skeleton` option is specified, `simp puppetfile generate --skeleton` prints to the console a parent Puppetfile that includes `Puppetfile.simp`, the assumed name of the Puppetfile containing only SIMP modules. This parent Puppetfile indicates where you would add site-specific modules, that would be deployed along with those in the included `Puppetfile.simp`. The `Puppetfile.simp` file is assumed to be in the same directory as the parent Puppetfile.

Tip: You can optionally have local modules in a specified environment automatically added to this parent Puppetfile using the `--local-modules SRC_ENV` option. Local modules are modules whose directories are not under Git source control and for which there is no local Git repository.

Other Options

The options `simp puppetfile generate` provides are documented in its built-in SIMP CLI documentation. To see the options available, execute the following:

```
simp puppetfile generate -h
```

Examples for Deploying SIMP Environments

The example below steps through one way to set up a test environment. You might also find the following HOWTO documents to be helpful:

- *How To Generate a SIMP Puppetfile*

  Gives detailed examples of how to generate a Puppetfile for use with r10k from a *Local deployment scenario* or using Internet Repos.

- *How To Create a Control Repository*

  Gives examples of how to create Control Repos for use with r10k or Code Manager both from a SIMP installation and from the Internet Repos.

- *How To Set up the SIMP Provided ENC*

  Explains how to set up the *ENC* provided by the simp-util RPM. Once this is set up, the environment a node is assigned to can be set in using a configuration file located on the puppetserver.

Updating an Existing Environment

When you are using the *Local deployment scenario* the following steps can be used to update and deploy Puppet modules from the local git repositories to an existing environment or to update any of your site specific modules in your environment.

1. **Update Site Specific Puppet Modules**

   Follow your site procedures to update any of your site specific Puppet modules.

   Make sure all your modules have an entry in the `Puppetfile`. See *Edit the Puppetfile to include non-simp site modules* for more details.

2. **Update SIMP Local Git Repositories**

   On the yum server (usually the SIMP server on smaller deployments) add any new or updated SIMP module RPMs to the yum repositories using your sites procedures.

   On your SIMP server, clear the yum cache

   ```
   yum clean all; yum makecache
   ```

   Then install or update any modules

   ```
   # For example if installing all the modules in simp-extras
   yum install simp-extras
   # or a simp Puppet module with a patch
   yum update pupmod-simp-aide
   ```

   This will update the local git repositories on the SIMP server.

3. **Deploy the Environment**

   - Generate the Environment's Puppetfile.simp

     Run **`simp`** to pull all the latest versions of the SIMP-packaged Puppet modules from the local git repositories:

     ```
     # The environment in this example is called `test`.  Replace `test` with the
     # name of your environment.
     cd /etc/puppetlabs/code/environments/test
     simp puppetfile generate > Puppetfile.simp
     ```

   - Verify the Environment's Puppetfile

     Make sure the `Puppetfile` you will be deploying from includes the following:

     - A line that includes the `Puppetfile.simp` which should look like:

       ```
       instance_eval(File.read(File.join(__dir__,"Puppetfile.simp")))
       ```

     - A line for each of your own modules. See *How To Generate a SIMP Puppetfile* for more information on how to generate and clean up the Puppetfile if needed.

     Warning: Any module not listed in the `Puppetfile` will be deleted from the target environment's `modules` directory, when you use *r10k* to deploy the modules.

   - Deploy the Modules

Use *r10k* to deploy the modules from the local Git repositories into the environment. Make sure the `umask` and `group` are set correctly so that the puppetserver has access to the files.

```
  # The environment in this example is called `test`.  Replace `test` with the
  # name of your environment.
  ( umask 0027 && sg puppet -c '/usr/share/simp/bin/r10k puppetfile install \
  --puppetfile /etc/puppetlabs/code/environments/test/Puppetfile \
  --moduledir /etc/puppetlabs/code/environments/test/modules' )

Use the :command:`--force` option if you get warnings that local changes will␣
↪get
overwritten and you are sure you do not have changes that need saving.
```

4. **Apply node configuration**

   Any new modules will now be available to use in configurations on your nodes in the updated environments. Any changes in updated modules will be applied on the next scheduled puppet run on any node in the environment. Run **`puppet agent -t`** to force puppet to run on any node in the environment if you do not want to wait for a scheduled run.

Creating and Deploying a Test Environment Using Local Module Repositories

This example uses the *Local deployment scenario* It steps through creating a new *SIMP Omni-Environment*, for which the SIMP *secondary* and *writable* environments are linked to the corresponding directories in the existing `production` environment. This will allow the two environments to use common passwords and PKI certificates. It is helpful to test out changes in SIMP or other Puppet modules.

It assumes:

- You have upgraded SIMP-packaged Puppet module RPMs via `yum`.
- You have updated some of your own Puppet modules.
- You want to create a `test` Puppet environment with both sets of the changes.
- You want to assign a subset of nodes to this new environment to test the changes.

1. **Create the new SIMP Omni-Environment**

   Run the `simp` command to create a *SIMP Omni-Environment* called `test` for which the SIMP *secondary* and *writable* environments are linked to the corresponding directories in the existing `production` environment.

   ```
   simp environment new test --link production
   ```

   This will:

   - Create the Puppet environment `test`
   - Copy over the entire `production` Puppet environment over to the `test` environment.
   - Link the `test` secondary environment to the `production` secondary environment.
   - Link the `test` writable environment to the `production` writable environment.

2. **Update the Puppetfile Pair**

   Edit /etc/puppetlabs/code/environment/test/Puppetfile and add/update any of your own Puppet modules that you are updating.

   Then update /etc/puppetlabs/code/environment/test/Puppetfile.simp using the following:

```
cd /etc/puppetlabs/code/environments/test
simp puppetfile generate > Puppetfile.simp
```

3. **Deploy the Modules**

   Use r10k to update the modules in the test environment:

   ```
   ( umask 0027 && sg puppet -c '/usr/share/simp/bin/r10k puppetfile install \
   --puppetfile /etc/puppetlabs/code/environments/test/Puppetfile \
   --moduledir /etc/puppetlabs/code/environments/test/modules' )
   ```

4. **Assign nodes to the new environment**

   Since the secondary and writable environments for the `production` and `test` SIMP Omni-Environments are linked in this example, you can move nodes between these two environments without generating new certificates.

   You can assign the node to the new environment by changing the `environment` setting in the node's `/etc/puppetlabs/puppet/puppet.conf` file, or, if you are using *the ENC supplied by the simp-util module*, adding a line like the following to `/etc/puppetlabs/puppet/environments.yaml`.

   ```
   node.my.domain: test
   ```

5. **Apply node configuration**

   To apply the configuration for a node in the `test` environment, simply run

   ```
   puppet agent -t
   ```

   You should see a message similar to the following at the beginning of the log for the Puppet run:

   `Info:  Using configured environment 'test'`

   > **Note:** Once you have deployed a Puppet environment, you can edit the modules in that Puppet environment directly to debug issues you may be having and run puppet agent -t to deploy those changes. However, the next time you use `r10k` to deploy the modules, it will **overwrite** any local changes, unless the module is defined as `:local => true` in the Puppetfile. It is recommended that you use `git` to manage your work flow.

Setting up a SIMP Environment in a Control Repository using Remote Git repositories

Although (with a detailed understanding of SIMP internals) you can have a *SIMP server* that will function with just a Puppet environment, by default, a functioning SIMP server **requires** its entire *SIMP Omni-Environment*.

This example creates a SIMP Omni-Environment skeleton whose Puppet environment you can check into a branch in your control repository.

This example assumes the new SIMP Omni-Environment will be using its own secondary and writable environments.

Before proceeding with this section, you should

- Have a basic understanding of control repositories.
- Have a basic understanding of how to use `git`.
- Have a Git repository that you will be using as your control repository.
- Have read *Deploying SIMP Environments*. It contains detailed descriptions of key topics:

  - the *SIMP Omni-Environment*
  - local, SIMP-managed Git repositories maintained by SIMP Puppet module RPMs
  - SIMP *CLI* commands to assist with SIMP Omni-Environment management.

- You may find it helpful to read the section that explains how a control repository works in Puppet, Inc.'s control repository documentation .

1. Install Prerequisite Packages

   The following packages are required:

   - `rubygem-simp-cli`
   - `simp-environment-skeleton`
   - `simp-rsync-skeleton`
   - `simp-selinux-policy`

   All of these packages will already be installed if you have installed the SIMP server from ISO or from RPMs.

   These RPMs can be obtained from an ISO or if the system has access to the internet the SIMP repositories can configured as described in *Installing SIMP From A Repository*.

   ```
   yum install rubygem-simp-cli simp-environment-skeleton simp-rsync-skeleton simp-
   ↪selinux-policy
   ```

2. Create the SIMP Omni-Environment skeleton without SIMP local repository Puppetfiles as `root`:

   ```
   simp environment new dev1 --no-puppetfile-gen
   ```

   This will create two skeleton directories:

   - a Puppet directory at `/etc/puppetlabs/code/environments/dev1`.
   - a Secondary directory at `/var/simp/environments/dev1`.

   A new Writable environment directory will be automatically generated, as needed, when `puppet` is run.

3. Generate the Puppetfile:

   Download the `Puppetfile` used to create a SIMP ISO for a specific release from the SIMP simp-core repository. In this example, we are going to use the SIMP `6.4.0-0` release.

   ```
   cd /etc/puppetlabs/code/environments/dev1
   simp puppetfile generate --skeleton > Puppetfile
   curl -o Puppetfile.simp  https://github.com/simp/simp-core/blob/6.4.0-0/Puppetfile.
   ↪pinned
   ```

   Manually edit the `Puppetfile.simp` to remove components that are not Puppet modules, by deleting all lines up to and including `moduledir 'src/puppet/modules'`.

   Manually edit the `Puppetfile` to add entries for any other non-SIMP modules your site requires.

   For more detailed information on editing these files see *How to Generate a Puppetfile*. This also explains how to create the Puppetfile if you are using Local Repositories instead of Internet Repositories.

4. Add/adjust any of the *Hiera* files in `/etc/puppetlabs/code/environments/dev1/data`

5. Create a New Branch in Your Control Repository

   Create an empty git repository in the newly created Puppet environment:

```
cd /etc/puppetlabs/code/environments/dev1
git init .
```

Add all the files to a branch named for the environment in this repository:

```
# create the branch
git checkout -b dev1

# add the directory tree
git add --all

# verify the directory tree doesn't have any temporary files you created
git status

git commit -m 'Initial dev1 environment'
```

Push the branch to your control repository:

```
# Add a remote for your control repository
git remote add control_repo <URL to the control repo>

# Push the branch
git push dev1 control_repo
```

6. To continue configuring the environment, move on to the *Client Management* section in the *SIMP User Guide*.

By default, this environment will be using a different Certificate Authority for *Infrastructure Certificates*. It therefore may not be able to communicate to a server in another environment when certificate authentication is required, for example LDAP. You will have to create a new LDAP server or add the new CA to the list of trusted CAs in the other environment.

This environment will also generate a new set of passwords when `simplib::passgen` function is used in a Puppet module. It therefore may not be able to talk to a server in another environment that requires password authentication.

Integrating Applications

This section describes how to integrate external applications into the SIMP managed infrastructure.

For most applications, there are only three SIMP control components that must be addressed for successful product integration.

IPTables

By default, the SIMP system drops all **incoming** connections to the server, save port `22`. Port `22` is allowed from **all** external sources since there is no safe way to restrict this that will not lock users out of freshly installed systems in many cases.

The default SIMP *IPTables* start-up sequence has been set to *fail safe*. This means that if the IPTables rules cannot cleanly apply, the system will only allow port `22` into the system for SSH troubleshooting and recovery.

There are many examples of how to use the `simp-iptables` module in the source code; the `simp-simp_apache` module is a particularly good example. This module can be found in your SIMP Puppet environment or, if SIMP is installed via ISO or RPM, at `/usr/share/simp/modules/simp_apache`.

You can also reference the *Defined Types* in the `simp-iptables` module, itself, to understand their purpose and choose the best option.

Local Access Controls

Following defense in depth best practice, SIMP does not trust a single system to determine the access that someone has to a system. All system accesses are, by default, restricted to users in the `administrators` group.

If you have an application that needs to use a login shell for configuration, or to run the service, you will need to follow the guidance in *PAM Access Restrictions* to ensure that your local user accounts have appropriate system access.

Note: This **does** affect `sudo` accounts! If your application is using a `sudo` account in a startup script, please consider switching to `runuser` since it is not affected by PAM controls.

Service Kill

To ensure that the system does not run unnecessary services, the SIMP team implemented a `svckill.rb` script to stop any service (not process) that is not properly defined in the Puppet catalog.

To prevent services from stopping, refer to the instructions in the *My Services Are Dying!* Troubleshooting section.

As of SIMP 6.0.0, the `svckill` Puppet Resource will now warn you that it would kill items by default and you will explicitly need to enable `svckill` enforcement.

General Administration

This section provides information on the standard administrative techniques used when managing SIMP systems.

Various philosophical decisions are also covered to help users understand why SIMP does some of the things that it does.

Warning: While working with the system, keep in mind that *Puppet* does not work well with capital letters in host names.

DO NOT USE CAPITAL LETTERS IN HOST NAMES

Package Management

The SIMP infrastructure has a consistent philosophy that managed packages should be at the latest version in the available repositories whenever the system is brought into alignment by *Puppet*.

All SIMP produced modules should, by default, have their versions set to `present` but `simp_options`, when set initially by `simp config`, should set that to `latest` across the environment.

The rationale behind this is that it is far easier to update a set of repositories than it is to precisely pin versions of all packages managed on a given system. Since repositories are generally common packages, the ability to create a set of symlinks that represent the latest tested state of a system should be far simpler than doing minutia management across your Puppet code.

Workflow

The general workflow to keep your system properly up to date would be as follows.

Note: We **highly** recommend using *Beaker* for testing these scenarios

1. Update the **Test** repository
2. Assign a test node to the repository via a `yumrepo` resource
3. Run Puppet and evaluate the results
4. Run a full system update and evaluate the results
   - This simulates the *Nightly Updates*, if enabled
5. If all goes well, migrate the changes to the **Production** repository
6. Let Puppet do the rest

See *Nightly Updates* for more information on setting up the repositories and providing packages to your clients.

Nightly Updates

All SIMP systems are configured, by default, to do a YUM update of the entire system on a nightly basis. When the update task runs, it will pull **ALL** updates that the system is aware of.

Note: Refer to *HOWTO Exclude YUM Repositories* for additional configuration information.

Note: See *HOWTO Modify the Nightly Update Schedule* for information on changing the nightly update schedule or disabling the nightly updates altogether.

To use this effectively, packages that all systems will receive should be placed into the `Updates` repository provided with SIMP. Any packages that will only go to specific system sets should then be placed into adjunct repositories under `/var/www/yum` and the user will point specific systems at those repositories using the `yumrepo` Puppet Type. Any common packages can be either symlinked or hard linked between repositories for efficiency.

Changing the Default Repositories

By default, SIMP stores *YUM* information in the following directories:

- `/var/www/yum`

The base SIMP repository is in `/var/www/yum/SIMP` and it is highly unlikely that you would want to modify anything in this directory.

By default, access to the YUM repository is restricted to the networks contained in the `simp_options::trusted_nets` parameter. For this section, we will assume that this is sufficient.

The Operating System Repos

The default location for the *Operating System* (OS) repositories, on the *SIMP Server*, is `/var/www/yum/<OSTYPE>/<MAJORRELEASE>/x86_64`.

An `Updates` repository has been configured in this space. All OS updates should be placed within this directory.

You should run the following in the `Updates` directory after **ANY** package addition or removal within that directory.

```
# createrepo .
# chown -R root.apache ./*
# find . -type f -exec chmod 640 {} \;
# find . -type d -exec chmod 750 {} \;
```

Adding a Custom Repository

For this section, we will assume that you have a repository named `foo` that you would like to expose to your systems. To do this, perform the following:

```
# cd /var/www/yum
# mkdir foo
# cd foo
# -- copy all RPMs into the folder
# createrepo .
# chown -R root.apache ./*
# find . -type f -exec chmod 640 {} \;
# find . -type d -exec chmod 750 {} \;
```

Note: For more information on managing YUM repos, please see the Red Hat local repository Documentation.

Configuring the Clients

Now that you have added this repository, you are going to want to add it to your clients.

The best way to do this is to make it part of your *site profile*. You **can** make it part of your module, but you will need to wrap it in a Defined Type so that the server parameter can be modified.

To add it to your clients, use the puppet `yumrepo` Type. You can find more information in the Puppet Type Reference.

The following is a basic `yumrepo` example:

```
yumrepo { example:
  baseurl         => "http://your.server.fqdn/yum/foo",
  enabled         => 1,
  enablegroups    => 0,
  gpgcheck        => 0,
  keepalive       => 0,
  metadata_expire => 3600
}
```

Session Auditing

Older versions of SIMP used *Sudosh* to enable logging of privileged user activities.

This has been replaced by *Tlog* for a more seamless user experience and all activities should be able to be done without the need for additional explicit `sudo` commands from this point on.

By default, the actions of the `root` user at a login shell are audited. Please see the *Tlog* documentation and the SIMP Tlog Puppet Module for additional details.

Note: The SIMP system does not allow the `root` user to execute `sudo` by default per common configuration guidance.

Note: If you built your system from an ISO, you will probably have a local `simp` user that has the ability to run `sudo su - root` directly.

The `simp` user is meant as an emergency 'break glass' user and should be removed or disabled once your environment is configured to your satisfaction.

User Accounts

The SIMP team tests both local and *LDAP* account access to systems. Other modes of access may function but are not tested by the SIMP test suite at this time.

We recommend that LDAP be used for adding all human users so that there is no conflict with multiple system updates and synchronization. For more information on managing LDAP users, refer to the *User Management* chapter.

If you need to create local system accounts, you can use the `user` and `group` *Puppet Resources*.

Certificate Management

This section describes the two different types of certificates used in a SIMP environment and provides links to further information.

Puppet Certificates

Puppet certificates are issued and maintained strictly within Puppet. Communication between the server and agents is granted and secured with HTTPS, which requires valid identifying SSL certificates. The *Puppet Server* acts as the certificate authority for managing these certificates.

The client will automatically send a certificate request to the server if it cannot find a valid certificate. The Puppet Server will automatically sign a certificate request if the client's name is in the `autosign.conf` file, otherwise an administrator must sign the request using the `puppetserver ca` tool.

Note: By default, Puppet certificates expire every five (5) years.

Infrastructure Certificates

Infrastructure certificates, commonly referred to as server certificates, are the standard *PKI* certificates assigned by an official *CA*. These are used by all other applications that require certificates.

SIMP offers capabilities to help manage these certificates including a FakeCA utility that will provide self-signed certificates for use on testing systems or until official certificates can be obtained.

The modules in SIMP have been designed so that infrastructure certificates can be managed from a central location and SIMP will distribute them to the applications that need them.

For more information on how to manage infrastructure certificates and how to use the FakeCA utility refer to the *Apply Certificates* section of *Client Management*.

The SIMP Utility

The SIMP server provides a command line utility called `simp` that is an interface into SIMP-specific settings and subsystems.

You can get information on the `simp` utility by running `simp help` on your SIMP server.

simp passgen

Throughout the SIMP codebase, you may find references to the `simplib::passgen()` function. This function auto-generates passwords and stores them in the *SIMP Server* in a sub-directory in the *SIMP Writable Environment*:

`/opt/puppetlabs/server/data/puppetserver/simp/environments/<environment>/simp_autofiles/gen_passwd`

For more information, see the simplib::passgen() documentation.

Graphical User Interfaces

SIMP was designed as a minimized system, but you may occasionally need a GUI. Refer to the *Graphical Desktop Setup* documentation for information on setting up GUIs for the systems.

Apply Certificates

All clients in a SIMP system should have *Public Key Infrastructure* (PKI) keypairs generated for the server. These are the referred to as the infrastructure or server keys. These certificates are used to encrypt communication and identify clients and are used by common applications such as LDAP and Apache.

Note: These keypairs are **not** the keys that the Puppet server uses for its operation. Do not get the two confused.

See *Certificate Management* for more information.

SIMP uses the simp/pki module to help distribute infrastructure keypairs. The global variable, `simp_options::pki` determines what parts of the module are included. It can be overridden in hiera data at several levels if different hosts or applications need to handle certificates differently.

`simp_options::pki` can have one of three settings:

1. `simp` - Keypairs are distributed from a central location on the Puppet server to the `/etc/pki/simp/x509` directory on the client. Any applications using them will then make a copy in `/etc/pki/simp_apps/<app name>/x509` with the correct permissions for an application to use.
2. `true` - Applications on the clients will copy the keypairs from a local directory on the client to `/etc/pki/simp_apps/<app name>/x509`. The default local directory to copy from is `/etc/pki/simp/x509` but this can be overridden by setting the `simp_options::pki::source` variable.
3. `false` - The user will have to manage keypairs themselves. You will need to look at each module that uses PKI on a client to determine what variables need to be set.

   Note: A setting of `false` does **not** disable the use of PKI in a module.

The following sections describe how to populate the central key distribution directory that :pupmod:'simp/pki` uses, when `simp_options::pki` is set to `simp`.

Installing Official Certificates

This section describes how to install infrastructure certificates from an official certificate authority on the *SIMP Server* for distribution to client servers. You need to have `simp_options::pki` set to `simp` on the client for this to work.

The key distribution directory on the SIMP Server is the `site_files/pki_files/files/keydist` sub-directory located under the *SIMP Secondary Environment*, `/var/simp/environments/`*`environment`*. Within the `keydist/` directory, the SIMP system expects there to be:

- A directory named `cacerts/` that contains the *CA* public certificates.
- Client-specific directories, each of which contains the public and private certificates for an individual client. The name of each client directory must be the `certname` of that client, which by default is the client's FQDN.

Here is an example key distribution directory for a `production` *SIMP Omni-Environment*:

```
/var/simp/environments/production/site_files/pki_files/files/keydist/cacerts/
/var/simp/environments/production/site_files/pki_files/files/keydist/cacerts/cacert_
↪a7a23f33.pem
/var/simp/environments/production/site_files/pki_files/files/keydist/cacerts/cca9a35.0
/var/simp/environments/production/site_files/pki_files/files/keydist/mycomputer.my.
↪domain/
/var/simp/environments/production/site_files/pki_files/files/keydist/mycomputer.my.
↪domain/mycomputer.my.domain.pem
/var/simp/environments/production/site_files/pki_files/files/keydist/mycomputer.my.
↪domain/mycomputer.my.domain.pub
/var/simp/environments/production/site_files/pki_files/files/keydist/yourcomputer.your.
↪domain/
/var/simp/environments/production/site_files/pki_files/files/keydist/yourcomputer.your.
↪domain/yourcomputer.your.domain.pem
/var/simp/environments/production/site_files/pki_files/files/keydist/yourcomputer.your.
↪domain/yourcomputer.your.domain.pub
```

To install official certificates on the SIMP Server, do the following:

1. Copy the certificates received from a proper *CA* to the SIMP server.
2. Add the certificates for the node to the key distribution directory in `site_files/`.
   a) Make the directory under the key distribution directory for the client's certificates using the client's `certname`.

   b) Copy the official public and private certificates to that directory.

   For example, to install certificates for a system named `mycomputer.my.domain` into the `production` environment:

```
mkdir -p /var/simp/environments/production/site_files/pki_files/files/keydist/
↪mycomputer.my.domain
mv /dir/where/the/certs/were/myprivatecert.pem \
   /var/simp/environments/production/site_files/pki_files/files/keydist/mycomputer.
↪my.domain/mycomputer.my.domain.pem
mv /dir/where/the/certs/were/mypubliccert.pub \
   /var/simp/environments/production/site_files/pki_files/files/keydist/mycomputer.
↪my.domain/mycomputer.my.domain.pub
```

3. Create and populate the *CA* certificates directory.

   a) Make the CA directory, `cacerts/`.

   b) Copy the root CA public certificates into `cacerts/` in *PEM* format, one per file.

```
cd /var/simp/environments/production/site_files/pki_files/files/keydist
mkdir cacerts
cd cacerts
for file in *.pem; do ln -s $file `openssl x509 -in $file -hash -noout`.0; done
```

4. Make sure the permissions are correct.

```
chown -R root.puppet /var/simp/environments/production/site_files/pki_files/files/
↪keydist
chmod -R u=rwX,g=rX,o-rwx /var/simp/environments/production/site_files/pki_files/
↪files/keydist
```

Note: The `site_files/` sub-directory of the SIMP Secondary Environment is configured as another module path in each *Puppet Environment's* `environment.conf`. For example, for the `production` environment, `/etc/puppetlabs/code/environments/production/environment.conf` would contain:

```
modulepath = modules:/var/simp/environments/production/site_files:$basemodulepath
```

Generating Infrastructure Certificates from the Fake CA

The "Fake" (self-signing) Certificate Authority (Fake CA) is provided by SIMP as a way to manage server certificates if official certificates could not be obtained at the time of client installation or the servers are operating in testing environments.

Note: This option should not be used for any operational system that can use proper enterprise PKI certificates.

Below are the steps to generate the certificates using the SIMP-provided, Fake CA. These steps assume the `production` environment.

1. cd to `/var/simp/environments/production/FakeCA/`
2. Run **`vi togen`**

a) Remove old entries from the file and add the *Fully Qualified Domain Name* (FQDN) of the systems (one per line) for which certificates will be created.

Note: To use alternate DNS names for the same system, separate the names with commas and omit any spaces.

For example, `.name,alt.name1,alt.name2.`

3. Run **`wc cacertkey`**

   a) Verify that the `cacertkey` file is not empty.

   b) If it is empty: enter text into the file, then save and close the file.

4. Run **`./gencerts_nopass.sh`**

Warning: If the **`clean.sh`** command is run after the certificates have been generated, you will not be able to generate new host certificates under the old CA. To troubleshoot certificate problems, see the *Troubleshooting Certificate Issues* section.

If issues arise while generating keys, navigate to the `/var/simp/environments/production/FakeCA/` directory, then type **`./clean.sh`** to start over.

After running the **`clean.sh`** script, type **`./gencerts_nopass.sh`** to run the script again using the previous procedure table.

The certificates generated by the FakeCA in SIMP are set to expire annually. To change this, edit the following files with the number of days for the desired lifespan of the certificates:

- `/var/simp/environments/production/FakeCA/CA`
- `/var/simp/environments/production/FakeCA/ca.cnf`
- `/var/simp/environments/production/FakeCA/default_altnames.cnf`
- `/var/simp/environments/production/FakeCA/default.cnf`
- `/var/simp/environments/production/FakeCA/user.cnf`

In addition, any certificates that have already been created and signed will have a config file containing all of its details in `/var/simp/environments/production/FakeCA/output/conf/`.

Important: Editing any entries in the above mentioned config files will **not** affect existing certificates. Existing certificates must be regenerated if you need to make changes.

The following is an example of how to change the expiration time from one year (the default) to five years for any newly created certificate:

```
for file in $(grep -rl 365 /var/simp/environments/production/FakeCA/)
do
   sed -i 's/365/1825/' $file
done
```

4.4.8 User Management

This chapter explains how to manage users in the default SIMP environment.

Managing Accounts in Lightweight Directory Access Protocol (LDAP)

By default, SIMP provides a *LDAP* server for centralized account management and to assist with meeting common policy requirements for account lockout and invalidation.

As of EL8, *OpenLDAP* has been removed from support but *389-DS* has become available and appears to be the path forward for general LDAP support.

As such, all SIMP systems running EL8+ will provide 389-DS as the default LDAP server. Client system configurations have been tested to support either system as a LDAP server.

Managing 389-DS Accounts

This section provides information on managing accounts in a SIMP-configured *389 DS* server.

If you are migrating from OpenLDAP, see *HOWTO Migrate OpenLDAP to 389 DS*.

The following provides base information for working with SIMP-managed *389-DS* systems.

For more information about getting started with 389-DS, see the 389-DS Quickstart

File Locations

- `/etc/dirsrv`
  - The default location for directory services
- `/usr/share/puppet_ds389_config`
  - Information used to bootstrap the 389-DS instances. May, or may not, be relevant once the system is fully operational.

ds* Commands

389-DS uses a collection of commands for managing instances.

Some of the more useful are listed below.

Note: It is important to know that the SIMP-managed 389-DS instances use an account-to-DN mapping for the `root` user that automatically maps `root` to the administrative DN for the instance if you are using LDAPI.

This makes the administration process much easier for daily activities and is recommended unless you need to manage the system remotely.

dsctl - Directory Server Control

- `dsctl -l`
  - List all instances on the system (ignore the `slapd-` prefix when referencing them in other commands).
- `dsctl [instance_name] <start|stop|restart|status>`
  - The easiest manner to manage the running state of your instances.
- `dsctl [instance_name] healthcheck`
  - Check the instance for common issues

dsconf - Directory Server Configuration

- `dsconf [instance_name] config get`
  - Print the main configuration of the specified instance.
- `dsconf [instance_name] security get`
  - Print the security configuration of the specified instance.
- `dsconf [instance_name] pwpolicy get`
  - Print the **global** password policy for the instance.
- `dsconf [instance_name] localpwp list`
  - Print all known local password policies in the instance.
- `dsconf [instance_name] localpwp get [DN]`
  - Print the details of the local password policy specified by `[DN]` (This is one of the items output by `localpwp list`).
  - Note that local password policies are overrides to individual global password policy entries.

dsidm - Directory Server Identity Management

The `dsidm` command provides account management capabilities and the usage is covered in detail in the account management sections.

To make using `dsidm` easier, you may want to add something like the following to `~/.dsrc`:

```
[<instance_name>]
uri = ldapi://%%2fvar%%2frun%%2fslapd-<instance_name>.socket
basedn = <base DN>
```

For a more concrete example, we will use the `accounts` instance provided by the `simp/simp_ds389` module.

To find your base DN, you can run the following:

```
dsidm accounts -b "<base DN>" account list | head -1
```

Assuming that our base DN is `dc=local,dc=com`, our configuration file would look like the following:

```
[accounts]
uri = ldapi://%%2fvar%%2frun%%2fslapd-accounts.socket
basedn = dc=local,dc=com
```

User Management in 389-DS

List 389-DS Users

You can list all users in the default SIMP *389-DS* instance by running:

```
dsidm accounts -b "<base DN>" user list
```

Add a User to 389-DS

To add a user to 389-DS, you can either run `dsidm user create` and it will prompt you for input or you can provide most parameters at the command line as follows:

```
dsidm accounts -b "<base DN>" user create --uid alice --cn "Alice User" --displayName
↪'Alice' \
  --uidNumber 1000 --gidNumber 1000 --homeDirectory /home/alice
```

Remove a User from 389-DS

To remove our *alice* user, run the following command:

```
dsidm accounts -b "<base DN>" user delete <DN>
```

It will prompt you to type `Yes I am sure` to confirm deletion.

To get the DN for the user run:

```
dsidm accounts -b "<base DN>" user get alice | head -1 | cut -f2- -d' '
```

Add a Password to a 389-DS User

You may notice that this user has been created without a password. The command line options do not provide this capability so a password will need to be added afterwards.

Note: No matter which of the following methods you choose, the user will be prompted to change their password at the next login by default.

Interactive Reset

To be prompted for the user credentials, you can run the following:

```
dsidm accounts -b "<base DN>" account reset_password "<DN>"
```

To obtain the DN run:

```
dsidm accounts -b "<base DN>" user get alice | head -1 | cut -f2- -d' '
```

Direct Reset

If you want to set the user's password directly, first generate the password using pwdhash:

```
pwdhash -D /etc/dirsrv/slapd-accounts "<plain_text_password>"
```

Then run the following, pasting the output of the previous command into <GENERATED_HASH>:

```
dsidm accounts -b "<base DN>" user modify alice add:userPassword:<GENERATED HASH>
```

Add a SSH Public Key to a 389-DS User

You can use the following command to add a SSH key to a 389-DS user:

```
dsidm accounts -b "<base DN>" user modify alice add:nsSshPublicKey:"<ssh-rsa AAA...>"
```

Remove a SSH Public Key from a 389-DS User

You can use the following command to remove a SSH key from a 389-DS user:

```
dsidm accounts -b "<base DN>" user modify alice delete:nsSshPublicKey:"<ssh-rsa AAA...>"
```

Group Management in 389-DS

List 389-DS Groups

You can list all groups in the default SIMP *389-DS* instance by running:

```
dsidm accounts -b "<base DN>" group list
```

If running a SIMP-generated default instance, you should see the usual users and administrators groups.

Add a Group to 389-DS

To add a group to 389-DS, you can either run `dsidm posixgroup create` and it will prompt you for input or you can provide most parameters at the command line as follows:

```
dsidm accounts -b "<base DN>" posixgroup create --cn alice --gidNumber 1000
```

Note: Note the use of `posixgroup` instead of `group` when adding groups.

- `posixgroup` => POSIX-style groups generally used for system accounts.
- `group` => Regular LDAP groups which may be useful for external services.

Remove a Group from 389-DS

To remove our *alice* group, run the following command:

```
dsidm accounts -b "<base DN>" group delete "<DN>"
```

It will prompt you to type `Yes I am sure` to confirm deletion.

To get the DN for the group run:

```
dsidm accounts -b "<base DN>" group get alice | head -1 | cut -f2- -d' '
```

Get Information about a 389-DS Group

Use the following command to get information about a specific group:

```
dsidm accounts -b "<base DN>" group get alice
```

Add a User to a 389-DS Group

Use the following command to add a user to a group:

```
dsidm accounts -b "<base DN>" group add_member "<DN>"
```

You can get the DN of a user by running:

```
dsidm accounts -b "<base DN>" user get <username> | head -1 | cut -f2- -d' '
```

It is important to note that, by default, referential integrity is **not** preserved between users and groups. This means that you will need to manually remove users from groups if you decide to delete a user.

If you want to change this behavior, you can enable the Referential Integrity Postoperation plug-in manually. However, this has ramifications in clustered environments so please read the related documentation before proceeding.

Managing OpenLDAP Accounts

This section provides information on managing accounts in a SIMP-configured *OpenLDAP* server.

If you need to migrate to *389 DS*, see *HOWTO Migrate OpenLDAP to 389 DS*.

Add a User to OpenLDAP

Note: Before getting started, take a look at how to prepare SIMP LDIFs to make sure you understand the basics of *LDIF* files.

Users can be added with or without a password. Follow the instructions in the following sections.

Note: Every user must belong to a unique, primary group, but can optionally belong to one or more, secondary groups.

Warning: This process should not be used to create users or groups for daemon processes unless the user has experience.

Add a User to OpenLDAP with a Password

To add a user with a password to the system, along with a unique group for that user:

1. Login to the LDAP server as `root`.
2. Use the `slappasswd` command to generate a password hash for a user.
3. Edit the `/root/ldifs/add_user_with_password.ldif` shown below.

```
dn: cn=<username>,ou=Group,dc=your,dc=domain
objectClass: posixGroup
objectClass: top
cn: <username>
gidNumber: <Unique GID Number>
description: "<Group Description>"

dn: uid=<username>,ou=People,dc=your,dc=domain
uid: <username>
cn: <username>
givenName: <First Name>
sn: <Last Name>
mail: <e-mail address>
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
```

(continues on next page)

(continued from previous page)

```
objectClass: shadowAccount
objectClass: ldapPublicKey
shadowMax: 180
shadowMin: 1
shadowWarning: 7
shadowLastChange: 10701
sshPublicKey: <some SSH public key>
loginShell: /bin/bash
uidNumber: <some UID number above 1000>
gidNumber: <GID number from above>
homeDirectory: /home/<username>
userPassword: <slappasswd generated SSHA hash>
pwdReset: TRUE
```

4. Type the following, substituting your DN information for `dc=your,dc=domain`:

```
ldapadd -Z -x -W -D "cn=LDAPAdmin,ou=People,dc=your,dc=domain" \
-f /root/ldifs/add_user_with_password.ldif
```

Ensure that an administrative account is created as soon as the SIMP system has been properly configured. Administrative accounts should belong to the `administrators` LDAP group (gidNumber 700). By default, Members of this group can directly access a privileged shell via `sudo su -`.

Note: The `pwdReset: TRUE` command causes the user to change the assigned password at the next login. This command is useful to pre-generate the password first and change it at a later time.

This command appears to be broken in some versions of `nss_ldap`. Therefore, to avoid future issues set `shadowLastChange` to a value around 10000.

Warning: The initial password set for a user must conform to the password policy or the user will not be able to login and change his/her password, even though the password reset has been enabled by `pwdReset: TRUE`.

Add a User to OpenLDAP without a Password

To add a user without a password to the system, along with a unique group for that user

1. Login to the LDAP server as `root`.
2. Edit the `/root/ldifs/add_user_no_password.ldif` shown below.

```
dn: cn=<username>,ou=Group,dc=your,dc=domain
objectClass: posixGroup
objectClass: top
cn: <username>
gidNumber: <Unique GID Number>
description: "<Group Description>"

dn: uid=<username>,ou=People,dc=your,dc=domain
uid: <username>
cn: <username>
```

(continues on next page)

(continued from previous page)

```
givenName: <First Name>
sn: <Last Name>
mail: <e-mail address>
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
objectClass: ldapPublicKey
sshPublicKey: <some SSH public key>
loginShell: /bin/bash
uidNumber: <some UID number above 1000>
gidNumber: <GID number from above>
homeDirectory: /home/<username>
```

3. Type the following, substituting your DN information for `dc=your,dc=domain`:

```
ldapadd -Z -x -W -D "cn=LDAPAdmin,ou=People,dc=your,dc=domain" \
-f /root/ldifs/add_user_no_password.ldif
```

Remove a User from OpenLDAP

To remove a user from the system, along with a unique group for that user:

1. Login to the LDAP server as `root`.
2. Edit the `/root/ldifs/del_user.ldif` shown below.

```
dn: cn=<User UID>,ou=Group,dc=example,dc=domain
changeType: delete

dn: uid=<User UID>,ou=People,dc=example,dc=domain
changeType: delete
```

3. Type the following, substituting your DN information for `dc=your,dc=domain`:

```
ldapmodify -Z -x -W -D "cn=LDAPAdmin,ou=People,dc=your,dc=domain" \
-f /root/ldifs/del_user.ldif
```

Update a User's SSH Public Key in OpenLDAP

To update an SSH public key:

1. Login to the LDAP server as `root`.
2. Edit the `/root/ldifs/mod_sshkey.ldif` shown below.

```
dn: uid=<User UID>,ou=People,dc=your,dc=domain
changetype: modify
replace: sshPublicKey
sshPublicKey: <User OpenSSH Public Key>
```

3. Type the following, substituting your DN information for `dc=your,dc=domain`:

```
ldapmodify -Z -x -W -D "cn=LDAPAdmin,ou=People,dc=your,dc=domain" \
-f /root/ldif/mod_sshkey.ldif
```

Force a Password Reset in OpenLDAP

To force a password reset for a user:

1. Login to the LDAP server as `root`.
2. Edit the `/root/ldifs/force_password_reset.ldif` shown below.

```
dn: uid=<username>,ou=People,dc=your,dc=domain
changetype: modify
replace: pwdReset
pwdReset: TRUE
-
replace: shadowLastChange
shadowLastChange: 10101
```

3. Type the following, substituting your DN information for `dc=your,dc=domain`:

```
ldapmodify -Z -x -W -D "cn=LDAPAdmin,ou=People,dc=your,dc=domain" \
-f /root/ldifs/force_password_reset.ldif
```

Note: The `ldapmodify` command is only effective when using the *ppolicy* overlay. In addition, the user's **shadowLastChange** must be changed to a value prior to the expiration date to force a *PAM* reset.

Lock an LDAP Account in OpenLDAP

To lock an LDAP account:

1. Login to the LDAP server as `root`.
2. Edit the `/root/ldifs/lock_user.ldif` shown below.

```
dn: uid=<username>,ou=People,dc=your,dc=domain
changetype: modify
replace: pwdAccountLockedTime
pwdAccountLockedTime: 000001010000Z
-
delete: sshPublicKey
-
replace: userPassword
userPassword: !!
```

3. Type the following, substituting your DN information for `dc=your,dc=domain`:

```
ldapmodify -Z -x -W -D "cn=LDAPAdmin,ou=People,dc=your,dc=domain" \
-f /root/ldifs/lock_user.ldif
```

Note: The `ldapmodify` command is only effective when using the *ppolicy* overlay.

Unlock an LDAP Account in OpenLDAP

To unlock an LDAP account:

1. Login to the LDAP server as `root`.
2. Edit the `/root/ldifs/unlock_account.ldif` shown below.

```
dn: uid=<User UID>,ou=People,dc=your,dc=domain
changetype: modify
delete: pwdAccountLockedTime
```

3. Type the following, substituting your DN information for `dc=your,dc=domain`:

```
ldapmodify -Z -x -W -D "cn=LDAPAdmin,ou=People,dc=your,dc=domain" \
-f /root/ldifs/unlock_account.ldif
```

Note: The `ldapmodify` command is only effective when using the *ppolicy* overlay.

Troubleshooting Issues in OpenLDAP

If a user's password is changed or the user changes it shortly after its initial set up, the "Password too young to change" error may appear. In this situation, apply the `pwdReset:TRUE` option to the user's account as described in *Add a User to OpenLDAP with a Password*.

Add a Group to OpenLDAP

Note: Before getting started, take a look at how to prepare SIMP LDIFs to make sure you understand the basics of *LDIF* files.

- *Remove a Group*
- *Add Users to a Group*
- *Remove Users from a Group*

SIMP systems are preconfigured with two groups:

- `administrators` (700): Group that has ssh and privilege escalation privileges
- `users` (100): Group that does not have ssh or privilege escalation privileges

To add another group:

1. Login to the LDAP server as `root`.
2. Edit the `/root/ldifs/add_group.ldif` shown below.

```
dn: cn=<groupname>,ou=Group,dc=your,dc=domain
objectClass: posixGroup
objectClass: top
cn: <groupname>
gidNumber: <Unique GID number>
description: "<Some useful group description>"
```

3. Type the following, substituting your DN information for `dc=your,dc=domain`:

```
ldapadd -Z -x -W -D "cn=LDAPAdmin,ou=People,dc=your,dc=domain" \
-f /root/ldifs/add_group.ldif
```

Remove a Group

To remove a group:

1. Login to the LDAP server as `root`.
2. Edit the `/root/ldifs/del_group.ldif` shown below.

```
dn: cn=<Group Name>,ou=Group,dc=your,dc=domain
changetype: delete
```

3. Type the following, substituting your DN information for `dc=your,dc=domain`:

```
ldapmodify -Z -x -W -D "cn=LDAPAdmin,ou=People,dc=your,dc=domain" \
-f /root/ldifs/del_group.ldif
```

Add Users to a Group

To add users to a group:

1. Login to the LDAP server as `root`.
2. Edit the `/root/ldifs/add_to_group.ldif` shown below.

```
dn: cn=<Group Name>,ou=Group,dc=your,dc=domain
changetype: modify
add: memberUid
memberUid: <UID1>
memberUid: <UID2>
...
memberUid: <UIDX>
```

3. Type the following, substituting your DN information for `dc=your,dc=domain`:

```
ldapmodify -Z -x -W -D "cn=LDAPAdmin,ou=People,dc=your,dc=domain" \
-f /root/ldifs/add_to_group.ldif
```

Remove Users from a Group

To remove users from a group:

1. Login to the LDAP server as `root`.
2. Edit the `/root/ldifs/del_to_group.ldif` shown below.

```
dn: cn=<Group Name>,ou=Group,dc=your,dc=domain
changetype: modify
delete: memberUid
memberUid: <UID1>
memberUid: <UID2>
...
memberUid: <UIDX>
```

3. Type the following, substituting your DN information for `dc=your,dc=domain`:

```
ldapmodify -Z -x -W -D "cn=LDAPAdmin,ou=People,dc=your,dc=domain" \
  -f /root/ldifs/del_from_group.ldif
```

Managing Local/Service Users

Though the SIMP team **highly recommends** using *LDAP* to centrally manage your users, you may occasionally need to set up a *service account* or specific local users on your systems.

This section walks you through doing this in a way that is compatible with SIMP.

The following examples assume that you are using a custom `site` module to manage site-specific puppet manifests and that you are configuring the `production` Puppet environment. The examples may easily be extrapolated into defined types if you wish, but are presented as classes for simplicity. Save the files below in `/etc/puppetlabs/code/environments/production/modules/site/manifests/` as `local_account.pp` and `service_account.pp`, ensuring the correct ownership, group, and permissions.

In `default.yaml`:

```
simp::classes:
  - 'site::local_account'
  - 'site::service_account'
```

Run `puppet`. The new accounts should be included in the /etc/passwd file.

If you are not familiar with setting up *SSH* keys, you may want to follow the relevant GitHub documentation.

Local User Account

```
class site::local_account {
  include 'ssh'

  $_local_account_user  = 'localuser'
  $_local_account_group = 'localgroup'
  $_local_account_id    = '1778'

  # You will probably want this in /home unless you are using NFS
```

(continues on next page)

(continued from previous page)

```
  $_local_account_homedir = "/home/${_local_account_user}"

  # You will need to get this from the user as it is their public key.
  $_local_account_ssh_public_key = 'AAA...=='

  group { $_local_account_group:
    gid       => $_local_account_id,
    allowdupe => false,
  }

  user { $_local_account_user:
    uid        => $_local_account_id,
    allowdupe  => false,
    gid        => $_local_account_group,
    home       => $_local_account_homedir,
    managehome => true,
    shell      => '/bin/bash'
  }

  # If you want your local user to have a password (no key),
  # omit this block and manually assign a password to the user
  # after creation (passwd <user>)
  file { "/etc/ssh/local_keys/${_local_account_user}":
    owner   => 'root',
    group   => $_local_account_group,
    mode    => '0644',
    content => $_local_account_ssh_public_key
  }

  sudo::user_specification { $_local_account_user:
    user_list => [$_local_account_user],
    host_list => [$::fqdn],
    runas     => 'root',
    cmnd      => ['/bin/cat /var/log/app.log'],
    passwd    => false
  }

  # Allow this account from everywhere
  pam::access::rule { "Allow ${_local_account_user}":
    users   => [$_local_account_user],
    origins => ['ALL']
  }
}
```

Service Account

```
class site::service_account {
  include 'ssh'

  $_svc_account_user    = 'svcuser'
  $_svc_account_group   = 'svcgroup'
  $_svc_account_id      = '1779'
  $_svc_account_homedir = "/var/local/${_svc_account_user}"

  # Since this is a service account, automatically generate an SSH key for
  # the user and store it on the Puppet Server for distribution.
  $_svc_account_ssh_private_key = ssh_autokey($_svc_account_user, '2048', true)
  $_svc_account_ssh_public_key  = ssh_autokey($_svc_account_user, '2048')

  group { $_svc_account_group:
    gid       => $_svc_account_id,
    allowdupe => false,
  }

  user { $_svc_account_user:
    uid        => $_svc_account_id,
    allowdupe  => false,
    gid        => $_svc_account_group,
    home       => $_svc_account_homedir,
    managehome => true,
    shell      => '/bin/bash'
  }

  file { "${_svc_account_homedir}/.ssh":
    ensure => directory,
    owner  => $_svc_account_user,
    group  => $_svc_account_group,
    mode   => '0600'
  }

  file { "${_svc_account_homedir}/.ssh/id_rsa":
    mode    => '0600',
    owner   => $_svc_account_user,
    group   => $_svc_account_group,
    content => $_svc_account_ssh_private_key
  }

   # In SIMP sshd is configured to use authorized_keys files in /etc/ssh/local_keys
  file { "/etc/ssh/local_keys/${_svc_account_user}":
    owner   => 'root',
    group   => $_svc_account_group,
    mode    => '0644',
    content => "ssh-rsa ${_svc_account_ssh_public_key}"
  }

  sudo::user_specification { $_svc_account_user:
```

(continues on next page)

(continued from previous page)

```
    user_list => [$_svc_account_user],
    host_list => [$facts['fqdn']],
    runas     => 'root',
    cmnd      => ['/bin/cat /var/log/app.log'],
    passwd    => false
  }

  # Allow this service account from everywhere
  pam::access::rule { "Allow ${_svc_account_user}":
    users   => [$_svc_account_user],
    origins => ['ALL']
  }
}
```

Testing

The table below lists the steps to test that the configuration was applied correctly.

1. Log on to a server that has the template code configuration applied.
2. Type `su - <USERNAME>`
3. Type `exec /usr/bin/ssh-agent /bin/bash` to ensure that ssh-agent has a shell running.
4. Type `/usr/bin/ssh-add` to attach the user's certificates.
5. **Optional**: Type `/usr/bin/ssh-add -l` to double check that the user's certificates were added successfully.
6. Type `ssh <HOST>` to SSH to a target machine that has the template code configuration applied.

If successful, the user should be authenticated and gain access to the target machine without entering a password.

If the user is prompted for a password, check to see if the permissions are set up properly and that the certificate keys are in the correct locations. In addition, check the `/etc/security/access.conf` file to ensure that it contains the user or user's group in an allow statement. See `access.conf(5)` for details.

Managing local users and groups with SSSD

Though the SIMP team **highly recommends** using *LDAP* to centrally manage your users, you may also wish to manage users via the local system.

Note: Prior to SSSD 1.16 (:*term: `EL 7+), there was a LOCAL provider. This has been fully deprecated as on :term: `EL* 8 and should no longer be used.

If you still need to use this capability, please see the documentation for `sssd::provider::local` directly.

This section walks you through setting up local user and group support using the SIMP `sssd` module.

The following examples assume that you are using the `site` module to set up your users. The examples are easily extrapolated into defined types but are presented as classes for simplicity.

The Simple Method

If you just want SSSD to pull from `/etc/passwd` and `/etc/group` then you just need to set the following in *Hiera*:

```
---
sssd::enable_files_domain: true
```

Using Alternate Files

If you want to use your own files, as documented in the **sssd-files(5)** man page, then you will need to set up an explicit domain with the correct settings.

To do this, use the following puppet code.

Important: The module will **not** manage the target files for you. You must ensure that the files have the correct content and exist prior to restarting SSSD.

```
class site::sssd_local {

  sssd::provider::files { 'local':
    passwd_files => ['/usr/local/etc/passwd'],
    group_files  => ['/usr/local/etc/group']
  }

  sssd::domain { 'local':
    description   => 'Default Local Domain',
    id_provider   => 'files',
  }
}
```

In `default.yaml`:

```
simp::classes:
  - 'site::sssd_local'
```

In *Hiera*, you will need to add the `local sssd` domain to `sssd::domains` if it does not already exist.

If you wish to include the domain in all of `$simp_options::trusted_nets`, add `sssd::domains` variable to `default.yaml`, copy existing domains from `simp_config_settings.yaml` and add `local` to the list of domain `id_providers`.

In `default.yaml`:

```
sssd::domains:
  - 'local'
  - <existing domains, ex. LDAP>
```

Run **puppet**.

A `local` domain should be created and referenced in `/etc/sssd/sssd.conf` and the **sssd** service should be running.

Additional Resources

If you have any issues logging in, you may want to see the *Troubleshooting Common Issues* section of the documentation.

Managing GRUB Users

In addition to being configured at initialization through the SIMP CLI or kickstarting via PXE boot, GRUB users can be managed by the `simp-simp_grub` module on both GRUB 2 and legacy GRUB systems.

Note: `simp-simp_grub` is not a core module and its corresponding package, `pupmod-simp-simp_grub`, may need to be installed prior to following this guide.

In *Hiera*, you will need to include the `simp_grub` class and set the appropriate parameters. For both GRUB 2 and legacy GRUB, this must include:

```
simp::classes:
  - 'simp_grub'

simp_grub::password: '<password hash or password>'
```

For GRUB 2, you will also have to set `simp_grub::admin` and can optionally specify whether unmanaged GRUB user accounts should be reported and/or purged and the number of rounds to use for hashing the password.

```
# required for GRUB 2
simp_grub::admin: '<username>'

# optional for GRUB 2
simp_grub::purge_unmanaged_users: <true or false>
simp_grub::report_unmanaged_users: <true or false>
simp_grub::hash_rounds: <integer>
```

After configuring Hiera, run `puppet`. The accounts and hashed passwords should be included in the `/etc/grub2.cfg` or `/etc/grub2-efi.cfg` files for GRUB 2 systems and in `/etc/grub.conf` on legacy GRUB systems.

4.4.9 Upgrading SIMP

This section describes both the general, recommended upgrade procedures as well as any version-specific upgrade procedures.

Important: To minimize upgrade problems in your production environment, we strongly recommend that you:

- Carefully read the CHANGELOG for the SIMP version to which you are upgrading, as well as the CHANGELOGs for any interim versions you are skipping over.
- Test your upgrades in a development environment before deploying to a production environment.
- Backup any critical server data/configurations prior to executing the upgrade to a production environment.
- On each managed server, ensure you have a local user with `su` and `ssh` privileges to prevent lockout.

General Upgrade Instructions

SIMP uses the Puppet modules' parameters as the system "API" (in terms of compatibility) and attempts to limit any API breaking changes to a minimum during a major release.

API breaking changes will have at least one minor release with deprecation warnings unless the change was to fix an actual bug in functionality.

A SIMP release version (e.g., "6.6.0-2") can be separated into three major numbers, in the format *X.Y.Z*:

- `X` is the MAJOR release number, and indicates severe API-breaking changes.
  - Updates to packages in the *simp-extras* RPM do not constitute a severe API-breaking change.
- `Y` is the MINOR release number, and indicates the addition of features or minor API-breaking changes either due to functionality bugs or after at least one MINOR release announcing the deprecation.
  - All API-breaking changes are kept to an absolute minimum and well documented in the release CHANGELOG.
- `Z` is the PATCH release number, and indicates full backwards-compatibility changes, such as bug fixes and improvements.

This section describes both the general, recommended upgrade procedures for `X`, `Y`, or `Z` releases.

Contents

Incremental Upgrades

For `Y` and `Z` SIMP changes, you should feel comfortable dropping the changes directly into your **test** systems. The promotion cycle from test to production should be short and painless if you reference the *version upgrade documentation*.

Beginning with SIMP 6.4.0, SIMP-packaged Puppet module RPMs no longer install updates directly into the `simp/` *Puppet environment* directory. You must upgrade your Puppet modules using the mechanism appropriate for your *environment deployment scenario*:

Important: Review any *Version-Specific Upgrade Instructions* prior to executing an Incremental Upgrade. There may be specific instructions regarding the upgrade process that you should follow.

Upgrading systems using the local deployment scenario

The following instructions are specific to the *Local deployment scenario*. They assume the Puppet environment you are updating is named `test`, and that you execute these steps as `root`:

1. Update the YUM Repositories

   - Update the repositories using a SIMP ISO:

     unpack_dvd can be used to extract the SIMP puppet module RPMs and the minimal OS RPMs from the SIMP ISO. **unpack_dvd** is installed from the simp-utils package.

     By default **unpack_dvd** uses information on the ISO to determine where to copy the RPMs to under `/var/www/yum` and then links the OS major version to the newly extracted OS directory. Since sometimes **unpack_dvd** can only determine the major version of the OS, you should supply a detailed version number for the OS using the -v option. The SIMP version release notes will tell you the version of the OS that is packaged with SIMP release.

     Use `unpack_dvd --help` for more information on the **unpack_dvd** and its options to modify any of the behavior described above.

     1. Copy the new SIMP ISO file to the yum server.
     2. From the yum server (as `root`):

        ```
        # Unpack the new SIMP ISO's RPMs into yum repositories
        unpack_dvd -v <OS version number> </path/to/ISO>
        ```

   - For RPM-based installation, follow your site's procedures to update your repositories.

2. Install the RPMs on your SIMP server:

   After updating the repositories log onto the SIMP server and **su** to `root` to perform the rest of the upgrade.

   ```
   # Make sure the puppet agent cron job does not run and pick up any
   # interim changes, including Puppet application RPM updates, until you
   # are ready for these changes.
   puppet agent --disable

   # Make sure yum picks up the new RPMs
   yum clean all; yum makecache

   # Apply updates to the local server
   yum update -y
   ```

   For SIMP 6.4 and later, this will also update the system-local, SIMP-managed Puppet module *Git* repositories.

3. If you are upgrading from a version prior to SIMP 6.4 you can skip to the step *Update the generated types for the environment*

   ** **The following steps only apply for upgrades from version 6.4 or later**

   - Generate the Environment's Puppetfile.simp

     Run **simp** to pull all the latest versions of the SIMP-packaged Puppet modules from the local git repositories:

     ```
     # The environment in this example is called `test`.  Replace `test` with the
     # name of your environment.
     cd /etc/puppetlabs/code/environments/test
     simp puppetfile generate > Puppetfile.simp
     ```

- Verify the Environment's Puppetfile

  Make sure the `Puppetfile` you will be deploying from includes the following:

  - A line that includes the `Puppetfile.simp` which should look like:

    ```
    instance_eval(File.read(File.join(__dir__,"Puppetfile.simp")))
    ```

  - A line for each of your own modules. See *How To Generate a SIMP Puppetfile* for more information on how to generate and clean up the Puppetfile if needed.

  > **Warning:** Any module not listed in the `Puppetfile` will be deleted from the target environment's `modules` directory, when you use *r10k* to deploy the modules.

- Deploy the Modules

  Use *r10k* to deploy the modules from the local Git repositories into the environment. Make sure the `umask` and `group` are set correctly so that the puppetserver has access to the files.

  ```
    # The environment in this example is called `test`.  Replace `test` with the
    # name of your environment.
    ( umask 0027 && sg puppet -c '/usr/share/simp/bin/r10k puppetfile install \
    --puppetfile /etc/puppetlabs/code/environments/test/Puppetfile \
    --moduledir /etc/puppetlabs/code/environments/test/modules' )

  Use the :command:`--force` option if you get warnings that local changes will get
  overwritten and you are sure you do not have changes that need saving.
  ```

** **This ends the steps that are only for 6.4 or later.** The next steps apply to all systems.

4. Update the generated types for the environment

   ```
   /usr/local/sbin/simp_generate_types -p /etc/puppetlabs/code/environments/test
   ```

5. Re-enable Puppet and apply the changes

   ```
   puppet agent --enable
   puppet agent -t
   ```

Upgrading systems that use control repositories

If you manage your SIMP server using *r10k* or *Code Manager* and are not using the server-local, SIMP-managed Git module repositories, you will need to work with the upstream Git repositories as appropriate for your workflow. This is the same for all versions of SIMP.

For SIMP 6.4 and later, the instructions in *HOWTO Set up a SIMP Environment in a Control Repository* may be helpful.

Breaking Changes

If the X version number has changed then you should expect **major** breaking changes to the way SIMP works. Please carefully read the *CHANGELOG* and the *SIMP User Guide* and do **not** deploy these changes directly on top of your production environment.

If the Y version number has changed then there may either be deprecation notices or **minor** breaking changes to the way SIMP works. Please carefully read the *CHANGELOG* and the associated *Version-Specific Upgrade Instructions*.

Important: Upgrading SIMP does **not** require re-kicking your clients, even if some core services move to the new Puppet node. All software configurations can be updated in Puppet, as needed.

With the release of 6.4, SIMP RPM upgrades now have a "hands-off" approach to upgrades that allow users to easily preserve different combinations of module sets as required by their environment. That being said, the SIMP team does not test all combinations of modules and may have difficulty providing support for untested combinations.

For releases moving from version of SIMP earlier than 6.3 to versions 6.4+, see *HOWTO Migrate to a New Puppet Server* for the simplest migration path. Also be sure to read the *Version-Specific Upgrade Instructions* for all of the intermediate versions.

Version-Specific Upgrade Instructions

Upgrading from SIMP-6.5.0 to SIMP-6.6.0

Before Upgrading to 6.6.0

SSSD Hiera Changes

This section describes two changes for systems that use *SSSD*.

Set simp::sssd::client::ldap_server_type

If *LDAP* and *SSSD* are both used, you must now set the type of LDAP server in *hiera* using the `simp::sssd::client::ldap_server_type` parameter.

If the system is using an OpenLDAP server, you would set this option to `plain`. For 389 DS-derived systems (389 DS, FreeIPA, RHDS, etc…) you would set this to `389ds`. Examples are provided below:

- When the upstream server is OpenLDAP

```
simp::sssd::client::ldap_server_type: 'plain'
```

- When the upstream server is 389-DS

```
simp::sssd::client::ldap_server_type: '389ds'
```

Remove SIMP-managed LOCAL domain configuration

In systems using **sssd** prior to 1.16.0, the daemon would fail to start if no domains were configured. Earlier versions of SIMP worked around this by creating a `LOCAL` domain that used the SSSD `files` provider.

For security and supportability reasons, SIMP no longer supports **sssd** prior to 1.16.0. Likewise, the `simp::sssd::client` class no longer configures a `LOCAL` domain. If you have configured a `LOCAL` domain you must do the following:

- Remove `LOCAL` from the list of domains in *hiera* from `sssd::domains`
- Disable the creation of the `LOCAL` domain by `simp::sssd::client`
  - This is only required until SIMP is upgraded to 6.6.0.

Below are examples:

- When LOCAL is the only `sssd::domains` entry

```
# BEFORE hieradata
sssd::domains:
- LOCAL
```

```
# AFTER hieradata
# sssd::domains has been removed!

# This can be removed once the upgrade is complete
simp::sssd::client::local_domain: false
```

- When LOCAL is not the only `sssd::domains` entry

```
# BEFORE hieradata
sssd::domains:
- LOCAL
- MY_DOMAIN
- SOMETHING_ELSE
```

```
# AFTER hieradata
# LOCAL has been removed from sssd::domains
sssd::domains:
- MY_DOMAIN
- SOMETHING_ELSE

# This can be removed once the upgrade is complete
simp::sssd::client::local_domain: false
```

NTP Hiera Changes

The `simp_options::ntpd::servers` parameter has been changed to `simp_options::ntp::servers` in order to support both **ntpd** and **chronyd** configuration, via simp/ntpd and puppet/chrony, respectively.

In *hiera*, users must:

- Rename `simp_options::ntpd::servers` to `simp_options::ntp::servers`
- Add an `alias` for `chrony::servers`
- Add a temporary `alias` for `simp_options::ntpd::servers` for backwards compatibility

Old *hiera*:

```
simp_options::ntpd::servers:
- server1.time.server.org
- server2.time.server.org
- server3.time.server.org
- 127.127.1.0
```

New *hiera*:

```
simp_options::ntp::servers:
- server1.time.server.org
- server2.time.server.org
- server3.time.server.org
- 127.127.1.0

chrony::servers:  "%{alias('simp_options::ntp::servers')}"

# This can be removed once the upgrade is complete
simp_options::ntpd::servers: "%{alias('simp_options::ntp::servers')}"
```

SIMP YUM Repository Changes

If you are using local SIMP *YUM* repositories, such as those installed on the SIMP server during a SIMP ISO install, you will need to migrate to the new SIMP repo format and ensure the latest Puppet GPG key is installed.

Migrate to new format and update SIMP repos

THIS REPLACES STEP 1, *Update the YUM Repositories*, **of the** *Incremental Upgrades* **instructions, 'Upgrading systems using the local deployment scenario' subsection.**

The SIMP *YUM* repository has been moved from `/var/www/yum/SIMP` to `/var/www/yum/SIMP/<os name>/<os version>` to make supporting multiple client operating systems easier. In addition, the `unpack_dvd` script in simp-utils was updated to extract the RPMs to the new location.

Do the following as *root* to properly migrate to the new format:

1. Manually migrate your existing /var/www/yum/SIMP repo to the new format

```
cd /var/www/yum/SIMP
mkdir -p CentOS/7.8.2003
chown -R root:apache CentOS
chmod -R g+rX CentOS
mv noarch CentOS/7.8.2003
mv x86_64/ CentOS/7.8.2003
```

2. Install the latest simp-utils.

```
# Mount the ISO
mkdir /mnt/iso
mount -o loop -t iso9660 <location of iso file> /mnt/iso

# Install the latest simp_utils RPM
cd /mnt/iso/SIMP/noarch
yum install -y ./simp-utils*.rpm || yum update -y ./simp-utils*.rpm

# Unmount the ISO
cd $HOME
umount /mnt/iso
```

3. Extract latest SIMP packages from SIMP ISO to the new format

```
# Extract the ISO.
unpack_dvd  -v 7.9.2009  <SIMP 6.6.0 ISO file path>
```

4. Create temporary links to the new repo directories that can be removed once the upgrade is complete

```
cd /var/www/yum/SIMP
ln -s ./CentOS/7/noarch ./noarch
ln -s ./CentOS/7/x86_64 ./x86_64
```

5. Update the `flocal` repo `baseurl` in `/etc/yum.repos.d/simp_filesystem.repo`, if your SIMP server was installed from ISO and that repository is enabled.

   - Original `baseurl`

```
baseurl=file:///var/www/yum/SIMP/x86_64
```

- Updated baseurl

```
baseurl=file:///var/www/yum/SIMP/CentOS/7/x86_64
```

Install the new Puppet GPG key

Puppet, Inc. has a new *GPG* key that must be used to install the new puppet-agent. Install this key as follows:

1. Install this key on the YUM server by updating simp-gpgkeys.

```
yum install -y simp-gpgkeys
```

2. To ensure that all the clients pick up the new key, add the following to the top level of *Hiera*:

```
# This can be removed once the upgrade is complete
simp::yum::repo::local_simp::extra_gpgkey_urls:
- https://<your yum server ip address>/yum/SIMP/GPGKEYS/RPM-GPG-KEY-puppet-20250406
```

3. To ensure the SIMP server picks up the new key, add the new key to the flocal repo gpgkey in /etc/yum.repos.d/simp_filesystem.repo, if your SIMP server was installed from ISO and that repository is enabled.

   - Original gpgkey

```
gpgkey=file:///var/www/yum/SIMP/GPGKEYS/RPM-GPG-KEY-puppet
    file:///var/www/yum/SIMP/GPGKEYS/RPM-GPG-KEY-puppetlabs
    file:///var/www/yum/SIMP/GPGKEYS/RPM-GPG-KEY-SIMP-6
    file:///var/www/yum/SIMP/GPGKEYS/RPM-GPG-KEY-SIMP
    file:///var/www/yum/SIMP/GPGKEYS/RPM-GPG-KEY-PGDG-96
```

   - Updated gpgkey

```
gpgkey=file:///var/www/yum/SIMP/GPGKEYS/RPM-GPG-KEY-puppet
    file:///var/www/yum/SIMP/GPGKEYS/RPM-GPG-KEY-puppetlabs
    file:///var/www/yum/SIMP/GPGKEYS/RPM-GPG-KEY-SIMP-6
    file:///var/www/yum/SIMP/GPGKEYS/RPM-GPG-KEY-SIMP
    file:///var/www/yum/SIMP/GPGKEYS/RPM-GPG-KEY-PGDG-96
    file:///var/www/yum/SIMP/GPGKEYS/RPM-GPG-KEY-puppet-20250406
```

Kickstart Prep

If you use kickstart, you will need to extract the PXEboot files before kickstarting new EL7 machines. See *Extract PXE files* for more information.

Complete the General Upgrade Steps

After resolving all applicable items in *Before Upgrading to 6.6.0*, resume *Incremental Upgrades* instructions **from Step 2**.

After Upgrading to 6.6.0

Cleanup Tasks

The following items can be removed from *hiera*:

```
simp::yum::repo::local_simp::extra_gpgkey_urls:
  - https://<your yum server ip address>/yum/SIMP/GPGKEYS/RPM-GPG-KEY-puppet-20250406

simp::sssd::client::local_domain: false

simp_options::ntpd::servers: "%{alias('simp_options::ntp::servers')}"
```

The following links can be removed on the *YUM* server:

```
unlink /var/www/yum/SIMP/noarch
unlink /var/www/yum/SIMP/x86_64
```

Upgrading to EL8

Main Server Migration

To upgrade your puppet server to EL8, you would follow the installation instructions and then the *HOWTO Migrate to a New Puppet Server*

Migrating from OpenLDAP to 389 DS

See: *HOWTO Migrate OpenLDAP to 389 DS*.

Upgrading from SIMP-6.4.0 to SIMP-6.5.0

> **Warning:** You **MUST** be on at least version 6.4.0 to upgrade directly to 6.5.0 due to changes to the deployment methods starting with 6.4.0.

Before Upgrading to 6.5.0

Survey EL6 Systems

Warning: Maintenance Updates for EL6 are EOL, and will stop on November 30 2020 for both *CentOS* and *RHEL*.

The SIMP 6.5.0 changelog contains a *list of (optional) Puppet modules that no longer support EL6*. Prior to upgrading, identify any EL6 nodes that still require these modules and either:

1. Update the affected EL6 nodes to EL7, or
2. Manage the affected EL6 nodes from a *SIMP 6.4.0 environment*.

Ensure Local Modules work with Puppet 6

This release updates **`puppet-agent`** to Puppet 6. Make sure all of your own Puppet modules work with Puppet 6 prior to upgrading.

Note: You may choose to stay on Puppet 5 by not upgrading the puppet-agent and puppetserver packages.

If you choose to stay on Puppet 5, please check the `metadata.json` for new modules to ensure that they still support Puppet 5 prior to using them.

Update NFS settings

The NFS Puppet modules have been updated to only accept IP addresses for the NFS server in order to support **`firewalld`** integration. If your site currently references NFS servers by hostname or FQDN, please change them to IP addresses before upgrading. This affects the parameters `simp_nfs::home_dir_server` and `nfs::client::mount::nfs_server`.

In Hiera:

```
simp_nfs::home_dir_server: '<change to server IP>'
```

In Puppet code:

```
nfs::client::mount { 'my directory':
  nfs_server =>  '<change to server IP>',
  # ...
}
```

Update DNS RNDC key

(The following only needs to be done if you are using pupmod-simp-named to configure your DNS server and it copies the files using rsync:)

On the SIMP Puppet server, remove the `rndc.key` from the rsync directories:

```
find /var/simp/environments/ -name rndc.key -exec rm -f {} \;
```

On your DNS servers, view the `/etc/rndc.key`. If the algorithm is `hmac-md5`, generate a new rndc key: Replace <`keyname`> with the name in the key field in your `/etc/rndc.key` files. (It should be the same as the name used in your `/etc/named.conf`.)

```
rndc-confgen -a -k <keyname> -b 256
```

Set `svckill::mode` to `warning`

Puppet recently made changes to how it internally enumerates services, with the consequence that, if `svckill::mode` is set to `enforcing`, it is possible that it could kill critical system services between updating to the new Puppet version and deploying the new Puppet modules.

As a safety precaution before upgrading, set `svckill::mode` from `enforcing` to `warning` until SIMP 6.5.0's Puppet modules have been deployed.

In Hiera:

```
svckill::mode: 'warning'
```

Uninstall out-of-date Puppet modules

This only affects user with a *Local deployment scenario*. The following list of optional Puppet modules required by the simp-extras RPM were not updated for SIMP 6.5.0.

If they are present, they will cause **`yum update`** to fail. If you installed them by installing simp-extras, then updating simp-extras will automatically resolve the dependencies. Otherwise you must do a **`yum remove`** of any of the following rpms installed on your system:

- pupmod-simp-journald
- pupmod-simp-simp_snmpd
- pupmod-simp-simp_gitlab
- pupmod-puppetlabs-docker

Note that this will **not** remove the local git repositories created for each Puppet module during previous SIMP RPM installations. The git repositories can still be referenced from SIMP 6.4 Puppetfiles and *deployed into a SIMP environment* that uses SIMP 6.4 modules.

Upgrading to 6.5.0

After resolving all applicable items in *Before Upgrading to 6.5.0*, follow the *Incremental Upgrades* instructions.

During/after the upgrade, you might see the following error during Puppet agent runs:

```
Error: /Stage[main]/Aide/Exec[update_aide_db]: Failed to call refresh: Command exceeded timeout
Error: /Stage[main]/Aide/Exec[update_aide_db]: Command exceeded timeout
```

This indicates that the rebuilding of the AIDE database has timed out. This is expected because many changes have been made to both the OS and Puppet modules. Follow whatever process you normally employ to update the AIDE database after the SIMP 6.5.0 updates have been made to the system.

After Upgrading to 6.5.0

Set `svckill::mode` back to `enforcing` (if applicable)

If you changed `svckill::mode` from `enforcing` to `warning` before the upgrade, you can set it back to `enforcing` after the upgrade is complete.

Mcstransd syslog errors

If the **`mcstransd`** daemon is enabled on a system, changes to how `/proc` was mounted to allow **`polkit`** to work can cause **`mcstransd`** to send a lot of errors to the system log.

simp/selinux has been updated with a fix for this, but it no longer manages **`mcstransd`** by default. To manage **`mcstransd`** with the simp/selinux module, set the following in Hiera:

```
selinux::manage_mcstrans_service: true
```

Upgrading from SIMP-6.3.3 to SIMP-6.4.0

Warning: You **MUST** be on at least SIMP 6.3.3 to upgrade to 6.4.0 due to issues in the RPMs prior to the 6.3.3 release.

If you need to upgrade from 6.2.0 to 6.3.X, please read *Upgrading from SIMP-6.2.0 to SIMP-6.3.0* with care.

As indicated by gradual updates in previous *Changelogs*, SIMP has been working on becoming more friendly to alternative modes of deployment and maintenance. The team has specifically been focusing on fully supporting *r10k* and *Code Manager* as methods of installation and maintenance over time.

SIMP 6.4.0 is the release where we make this happen!

We have taken care to make no technologically breaking changes and, in keeping with the version description in the *General Upgrade Instructions*, have not made changes that affect the API of modules provided by the supported module set.

That said, there are quite a few procedural items that need to be taken into account when upgrading to SIMP 6.4.0 so that things work generally like you are used to.

To get a full accounting of the major changes in behavior for 6.4.0, please see the *SIMP Community Edition (CE) 6.4.0-0* Changelog.

Important: This upgrade has a great potential for deleting files you need. Read the instructions carefully and back up your system prior to the upgrade.

RPM Name Changes and the Consequences

simp-rsync

simp-rsync has been replaced by simp-rsync-skeleton.

Note: Due to the nature of RPM uninstallation activities, to avoid ruining any active files in `/var/simp/environments/simp/rsync`, we had to create a final version of simp-rsync that is generally dependency free.

Nothing requires this RPM since it is not technically required any longer, but you **MUST** upgrade this RPM if you want to ensure that no files are accidentally removed from the `rsync` space on your SIMP server.

If you are not using a *Puppet Environment* named `simp` then you can safely remove the simp-rsync RPM after upgrading.

FakeCA

simp-environment has been renamed to simp-environment-skeleton.

Previously, the simp-environment RPM would place helpful file:*FakeCA* files into `/var/simp/environments/simp/FakeCA`. These files were for convenience and their presence, or absence, does not affect a running system in any way.

Upon upgrade, the simp-environment RPM will be removed and you will find that various `FakeCA` files are either missing or have `.rpmsave` appended to them if they were modified configuration files.

You may run the following script as `root` to restore the `FakeCA` functionality:

```
#!/bin/sh

cd /var/simp/environments/simp
\cp -r /usr/share/simp/environment-skeleton/secondary/FakeCA .

cd FakeCA
for x in *.rpmsave; do
  mv $x `basename $x .rpmsave`
done
```

Upgrade Steps

These steps for upgrading are for use in SIMP systems that are not currently using *r10k* or *Code Manager*.

Update OS and SIMP RPMs

Update the repos for the *OS* and SIMP and run **yum update**.

The following errors/warnings are expected:

```
warning: /var/simp/environments/simp/FakeCA/togen saved as /var/simp/environments/simp/
↪FakeCA/togen.rpmsave
warning: /var/simp/environments/simp/FakeCA/ca.cnf saved as /var/simp/environments/simp/
↪FakeCA/ca.cnf.rpmsave
libsemanage.semanage_direct_remove_key: Unable to remove module simp-environment at␣
↪priority 400. (No such file or directory).
/usr/sbin/semodule:  Failed!
```

Note: The old *SELinux* module, `simp-environment`, had to be removed before installing the new SELinux module, `simp` to avoid conflicts when the `simp` module from the simp-selinux-policy package is installed. Therefore, the error message above is emitted when the simp-environment RPM tries to remove the already-removed SELinux module.

If you are using the `FakeCA` in the `simp` environment, follow the steps in the `FakeCA` section above to restore the needed files.

Create the Puppetfile

At this point in the upgrade, the SIMP-packaged Puppet modules have been installed in `/usr/share/simp/modules` and imported into local Git repositories under `/usr/share/simp/git/puppet_modules`. Now you need to create the `Puppetfile` for *r10k* to use to deploy modules to the environment.

The environment name in this example is `myenv`. Run the following commands as `root` to create a default `Puppetfile`:

```
# Change to the top level puppet environment directory for your environment.
cd /etc/puppetlabs/code/environments/myenv

# Generate the main Puppetfile with local module entries
simp puppetfile generate --skeleton --local-modules myenv > Puppetfile

```

(continues on next page)

(continued from previous page)

```
# Generate the Puppetfile for the SIMP modules.
simp puppetfile generate > Puppetfile.simp

chown root:puppet Puppetfile Puppetfile.simp
chmod 640 Puppetfile Puppetfile.simp
```

Edit the main Puppetfile:

- Make sure all the local modules you created and have in the modules directory have `:local => true` entries. This includes the site module, which is no longer delivered as an RPM.
- Delete the `local => true` entries for any obsolete modules that you do not need (e.g. simpcat).

> **Warning:** Any module not listed in the `Puppetfile` will be deleted from the modules directory.

The following shows three examples of how to include your modules in the `Puppetfile`:

- site: a module that is not a Git repository and resides in the `modules/` directory
- mymodule: a Git repository that resides in a directory on the Puppet server
- mysrvmod: a Git repository on a GitLab/GitHub type server.

```
mod 'site',
  :local => true

mod 'mymodule',
  :git => 'file:///usr/share/mymodules/mymodule',
  :tag => '1.1.1'

mod 'mysrvmod',
  :git => 'https://gitserver.my.domain/mygitproject/mysrvmod.git',
  :tag => '1.0.1'
```

> **Important:** Make sure all the modules under your environments modules directory that you created and that are not SIMP modules are listed in the `Puppetfile`. This includes the site module, which SIMP no longer delivers.

> **Warning:** Any module not listed in the `Puppetfile` **will be deleted** from the modules directory.

Deploy the Puppet Modules to the Environment

Run the following commands to deploy the puppet modules to the environment: (The environment in this example is `myenv`. Replace this with the name of the environment you are updating.)

```
# Make sure the umask is set properly

# Set the umask and Run r10k as the puppet group to make sure
# the permissions and ownership are correct on the modules
( umask 0027 && sg puppet -c '/usr/share/simp/bin/r10k puppetfile install \
```

(continues on next page)

(continued from previous page)

```
--puppetfile /etc/puppetlabs/code/environments/myenv/Puppetfile \
--moduledir /etc/puppetlabs/code/environments/myenv/modules' )
```

Run Puppet

Run **`puppet agent -t`** on all your systems in the environment to apply the update.

Other Environments

If you have more than one environment, you need to create a Puppetfile and deploy the modules for each environment.

Adding SIMP Extra Modules After The Upgrade

If you had modules installed from the simp-extras package, they will have been updated with **`yum update`**, a local Git repository will have been created for them under `/usr/share/simp/modules`, and they will have been included in the `Puppet.simp` file when it was generated.

If, after the upgrade you decide to install a new SIMP module, you must update the `Puppetfile.simp` file and deploy the modules again.

For example if you want to install the snmp and snmpd modules:

```
yum install pupmod-puppet-snmp pupmod-simp-snmpd
# After running yum you will see the modules under
# /usr/share/simp/modules but not in your environment.

# Now update the Puppetfile.simp and deploy the environment again.
cd /etc/puppetlabs/code/environments/myenv
simp puppetfile generate > Puppetfile.simp
( umask 0027 && sg puppet -c '/usr/share/simp/bin/r10k puppetfile install \
--puppetfile /etc/puppetlabs/code/environments/myenv/Puppetfile \
--moduledir /etc/puppetlabs/code/environments/myenv/modules' )
```

If you add one of your own modules, manually edit the `Puppetfile` and deploy again. You do not need to run **`simp puppetfile generate`** with the skeleton option again.

Remember, if you do not add the module to the `Puppetfile` it will get erased the next time the `Puppetfile` is deployed!

Upgrading If R10K Is Already Being USED

UNDER CONSTRUCTION

Upgrading from SIMP-6.3.2 to SIMP-6.3.3

The 6.3.3 release of SIMP is a bugfix release focusing on a specific issue with the updated sssd module that may cause account lockouts on upgrade depending on how your LDAP accounts have been maintained.

You can follow *Incremental Upgrades* for upgrading.

> **Warning:** If you are upgrading from 6.2.0 to 6.3.X, please read *Upgrading from SIMP-6.2.0 to SIMP-6.3.0* with care.

Upgrading from SIMP-6.3.0 to SIMP-6.3.2

The 6.3.2 release of SIMP is a major bugfix release.

You can follow *Incremental Upgrades* for upgrading.

> **Warning:** If you are upgrading from 6.2.0 to 6.3.X, please read *Upgrading from SIMP-6.2.0 to SIMP-6.3.0* with care.

Upgrading from SIMP-6.2.0 to SIMP-6.3.0

The main purpose of the SIMP 6.3 release is to upgrade to Puppet 5/Hiera 5. You should familiarize yourself with the Puppet 5 release notes and the Hiera 5 upgrade documents.

You can follow *Incremental Upgrades* for upgrading with only the exception noted below in Upgrade the Puppetserver section.

After the system is upgraded you can enable the environment layer, by giving each environment its own `hiera.yaml` (recommended). General instructions are given in the Updating Hiera section below.

Upgrade the Puppetserver

After you have disabled the puppet agent on the puppetserver and before you have upgraded the puppetserver to puppet 5 or updated the simp modules, make the following two edits to the puppet configuration files. If this is skipped the puppet server will not restart after it is upgraded to Puppet 5.

File:

`/etc/puppetlabs/puppetserver/services.d/ca.cfg`

Add line:

`puppetlabs.trapperkeeper.services.watcher.filesystem-watch-service/filesystem-watch-service`

File:

`/etc/puppetlabs/puppetserver/conf.d/web-routes.conf`

Add line:
`"puppetlabs.trapperkeeper.services.metrics.metrics-service/metrics-webservice": "/metrics"`

Also note that you may see the following warnings multiple times during the upgrade:

```
Warning: Setting trusted_server_facts is deprecated.
   (location: /opt/puppetlabs/puppet/lib/ruby/vendor_ruby/puppet/settings.rb:1169:in␣
↪`issue_deprecation_warning')
Warning: Setting ca is deprecated.
    (location: /opt/puppetlabs/puppet/lib/ruby/vendor_ruby/puppet/settings.rb:1169:in␣
↪`issue_deprecation_warning')
```

This message is innocuous, but unavoidable unless you are comfortable with *Disabling all Puppet deprecation warnings*.

Restart PuppetDB

At this point, all packages should be updated and the **puppetserver** process has probably restarted. If you've been using your system with *puppetdb* for a while, then you will want to restart the **puppetdb** process to ensure a successful database migration.

```
/opt/puppetlabs/bin/puppet resource service puppetdb ensure=stopped
/opt/puppetlabs/bin/puppet resource service puppetdb ensure=running
```

Updating Hiera

One of the major features of *Hiera* 5 is the addition of the environment layer. You can enable the Hiera environment layer with the following steps.

The new `hiera.yaml` file is installed by simp-environment-5.3.0-0.<dist>.noarch.rpm into `/usr/share/simp/environments/simp`. This hiera.yaml file uses `data/` as the directory instead of `hieradata/`. (If you have made changes to the hiera hierarchy, make sure to incorporate these into the file.)

For every environment, copy the new `hiera.yaml` to the environment and then rename the `hieradata/` directory to `data/`.

```
cp /usr/share/simp/environments/simp/hiera.yaml /etc/puppetlabs/code/environments/
↪<environment>/
cd /etc/puppetlabs/code/environments/<environment>/
chmod 640 ./hiera.yaml
chown root:puppet ./hiera.yaml
mv hieradata data
```

When all the environments are converted, update the global `hiera.yaml` file:

```
#  Copy an empty hiera.yaml file to the global level.
cd /etc/puppetlabs/puppet
unlink hiera.yaml
rm hiera.yaml.simp
cp hiera.yaml.rpmnew hiera.yaml
chown root:puppet hiera.yaml
chmod 640 hiera.yaml
```

If the `.rpmnew` file does not exist the new global `hiera.yaml` file should look like:

```
version 5
# defaults:
# data_hash: yaml_data
# hierarchy:
# - name: Common
# data_hash: yaml_data
hierarchy: []
```

Upgrading from SIMP-6.1.0 to SIMP-6.2.0

Important: It is *highly recommended* that you read the information in this section in its entirety before upgrading.

Exclude puppet-agent from the `yum update`

A bug (SIMP-5021) was identified in SIMP 6.1.0 that will prevent the **`puppetserver`** service from starting after the puppet-agent package is upgraded. The issue is fixed in 6.2.0, but special handling is necessary during the upgrade.

Note:

- This problem affects ISO/RPM-based installations of SIMP when upgrading from 6.1.0, unless precautions are taken (detailed below).
- SIMP installations that are upgraded using *r10k* and *Code Manager* are unlikely to be affected, unless the SIMP server's puppet-agent package is updated independently.
- These instructions only apply to the SIMP server—no changes are required on clients.

1. Before upgrading anything, add the following line to `/etc/yum.conf`:

   ```
   exclude=puppet-agent
   ```

   Assuming you aren't already using `/etc/yum.conf` to exclude other packages, you can add this line automatically by running:

   ```
   puppet resource file_line workaround path=/etc/yum.conf line='exclude=puppet-agent'
   ```

2. Proceed with the upgrade as outlined in *Incremental Upgrades*, up through the **`yum update`**:

```
# (ISO Installations only:) Unpack the new ISO's RPMs into yum repositories
unpack_dvd </path/to/ISO>

# Make sure yum picks up the new RPMs
yum clean all; yum makecache

# Apply updates to the local server
yum update -y
```

3. After running **yum update**, remove the `exclude=puppet-agent` line from `/etc/yum.conf`.

   If you added the line using **puppet resource**, you can remove it the same way:

   ```
   puppet resource file_line workaround path=/etc/yum.conf line='exclude=puppet-agent' ensure=absent
   ```

4. Run **puppet agent -t** to upgrade the puppet-agent package
5. Run **puppet agent -t** again to ensure that everything runs cleanly.

Recovering if the puppetserver fails to restart

If, during the upgrade, the **puppetserver** service fails to start with the error:

```
Puppet::Error: Cannot determine basic system flavour
```

You should be able to recover by running the command:

```
puppet resource file /opt/puppetlabs/puppet/cache owner=puppet group=puppet

# The service should now restart
puppet resource service puppetserver ensure=running
```

After you have recovered in this manner, it is safe to proceed re-running the step you had been attempting when the puppetserver failed.

Update Kickstart Files

The example *kickstart* file (`pupclient_86_64.cfg`—see *Setting up Kickstart*) was updated. Existing kickstart files that are modeled on older versions of this file should backport these changes. New, OS-specific versions of this file are included in the SIMP-6.2.0 ISOs, but may also be found in the simp-core repository under `build/distributions/`*`OS`*`/`*`OS Major Version`*`/`*`Architecture`*`/DVD/ks/`.

Important: Careful examination of `pupclient_86_64.cfg` is recommended, if you have customized it for your site beyond token replacement. The latest version contains a few subtle bug fixes, such as fixing an incorrect path that would prevent UEFI systems from booting.

Two major changes were made:

1. `pupclient_86_64.cfg` was updated to include instructions regarding what to change in order to accommodate UEFI boot. The associated configuration lines are commented out by default.

2. `pupclient_86_64.cfg` was updated to download and use new bootstrap service files. These files, managed by `simp::server::kickstart`, are:

   - A **systemd** unit file for CentOS 7 (`simp_client_bootstrap.service`) or a **systemv** init script for CentOS 6 (`simp_client_bootstrap`).
   - A common bootstrap script (`bootstrap_simp_client`) used by both.

   This pair of files replaces the deprecated **runpuppet** script. They are required in order to solve two timeout problems on particularly loaded systems, both of which can cause client Puppet bootstrapping to fail and require subsequent manual intervention in order to fix:

   - On CentOS 7, **systemd** was killing **runpuppet**, when it ran longer than 5 minutes. This was solved by using an actual **systemd** unit file (`simp_client_bootstrap.service`) with a default timeout of 30 minutes.
   - On CentOS 7, if the DHCP lease expired in the middle of bootstrapping, the generated Puppet configuration would erroneously use `localhost` for the client hostname. This problem was solved by setting the static hostname of the client at the beginning of the client Puppet bootstrap process.

   In addition to solving these specific problems, the new bootstrap scripts use a configurable back-off algorithm in order prevent flooding a heavily-loaded *Puppet Server* with requests. See the `simp::server::kickstart::simp_client_bootstrap` class for details.

Update dhcpd.conf

Changes were added to the `dhcpd.conf` file that enable the DHCP server to determine what mode, BIOS or UEFI, a system is kickstarting in and then to set the appropriate boot loader file on the TFTP server.

On a SIMP server, the example `dhcpd.conf` file is installed in `/var/simp/environments/simp/RedHat/Global/dhcpd/dhcpd.conf` via the simp-rsync package. This file may also be found in the simp-rsync-skeleton repository under `environments/simp/rsync/RedHat/Global/dhcpd`.

Note: When the simp-rsync RPM is upgraded, a message may be displayed that indicates

```
warning: /var/simp/environments/simp/rsync/RedHat/Global/dhcpd/dhcpd.conf\
created as /var/simp/environments/simp/rsync/RedHat/Global/dhcpd/\
dhcpd.conf.rpmnew
```

Because the contents of the `/var/simp/environments/simp/rsync/` directory are pushed to remote systems, files ending with `.rpmnew` are deleted from the directory as part of the RPM deployment.

The commands below can be used to extract the new config file from the RPM, which can then be merged with the existing `dhcpd.conf` file.

1. Run **`rpm2cpio simp-rsync-6.2.1-0.el7.noarch.rpm |cpio -ir "*dhcpd.conf"`**

   - **rpm2cpio** converts the RPM file to cpio archive. In this example, the command is being run from the directory containing the RPM file; if the file is not in the current working directory, the full path to the file must be provided.
   - **cpio** extracts files from the cpio archive. In this case, the cpio options `-i`, `-r`, and "`\*dhcpd.conf`" direct cpio to extract one or more files from the archive, interactively rename the file, and the path and filename of files to be extracted should end with `dhcpd.conf`.

2. When prompted `rename ./var/simp/environments/simp/rsync/RedHat/Global/ dhcpd/dhcpd.conf ->`, enter the desired path and name of the file to be extracted, such as `/tmp/dhcpd.conf.rpmnew`. Paths can be either absolute or relative, depending whether they start with a `/` or `./`.

Update the TFTP Root Directory

The default *TFTP* root directory was changed to /var/lib/tftpboot to conform to DISA STIG standards. To continue using /tftpboot set tftpboot::tftpboot_root_dir in Hiera to /tftpboot. Alternately, to use the new directory, copy any files not managed by the rsync module (i.e., not stored in /var/simp/environments/*environment*/rsync/*OS*/Global/tftpboot) to the new directory. Make sure the permissions, including selinux context, are correct. TFTP boot will fail to find boot files that have the incorrect selinux context.

Optionally Remove OBE Logrotate Configuration Files

SIMP-managed **logrotate** rules are now in /etc/logrotate.simp.d instead of /etc/logrotate.d. The rules in /etc/logrotate.d are still applied, but **logrotate** is configured to read the rules in /etc/logrotate.simp.d first.

This change was made to ensure SIMP-managed rules take precedence over vendor-supplied rules, because, when there are multiple rules specified for the same file, only the first rule is applied. Any subsequent rules are discarded. In fact, for some versions of **logrotate**, a rule with a duplicate log file is discarded *in its entirety*, even if only one of the managed log files is a duplicate. This means the remaining log files specified in that discarded rule will *not* be rotated!

Because the location of the SIMP-managed **logrotate** rules has changed, existing, but now OBE, SIMP rules will still reside in /etc/logrotate.d. Although these rules cause no issues with **logrotate**, they may be confusing to system administrators. So, you may wish to manually remove these rules.

You can easily identify OBE SIMP rules in /etc/logrotate.d by their This file managed by puppet. comment lines.

Known Issues

Issues With Copying Repo Contents

A bug has been reported in some cases where the *simp-adapter* package does not appear to be properly updating existing repository spaces in all cases.

If you are getting unexpected errors, then you should run the following command and evaluate any differences. The materials in /usr/share/simp/modules are authoritative for the latest build but will not be copied over if the target files are checked into a version control system.

```
diff -qr /usr/share/simp/modules /etc/puppetlabs/code/environments/simp/modules
```

Once you have corrected any issues, you should run **/opt/puppetlabs/bin/puppetserver reload** and the system should function properly again.

Upgrading from SIMP-6.0.0 to SIMP-6.1.0

Important: It is *highly recommended* that you read the information in this section in its entirety.

Upgrade Script

There were several issues found during the SIMP 6.0.0 to 6.1.0 upgrade that necessitated the creation of an upgrade script that is to be run on your *SIMP Server* systems.

Note: No changes are required on your clients for the upgrade to succeed.

The upgrade script, `/usr/share/simp/upgrade_scripts/upgrade_simp_6.0.0_to_6.1.0`, will assist with the upgrade from `6.0.0` to `6.1.0`, taking into account all of the specific issues. This script is available in the simp-utils-6.1.0 package provided by SIMP 6.1.0 or the simp-utils repository.

As always, back up your system prior to upgrading!

Note: This script assumes that you are upgrading from the SIMP RPMs!

If you have chosen some other installation method, you will need to follow the general steps outlined in the script.

To perform the upgrade, with `root` permissions:

1. Upgrade the simp-utils package, *only*, by executing **`yum update -y simp-utils`**.
2. Run the script: **`/usr/share/simp/upgrade_scripts/upgrade_simp_6.0.0_to_6.1.0`**

Note: This script will:

1. Run a **`yum -y update`**.
2. Reinstall simp-gpgkeys and pupmod-simp-timezone due to RPM issues.
3. Stop and uninstall the PostgresSQL 9.4 server to prevent postgresql upgrade issues.
4. Restart the **`puppetserver`** process.
5. Run **`puppet agent -t`**.
    - Some systems have shown issues with the postgresql upgrade during this step.

Update auth.conf

The legacy `auth.conf`, `/etc/puppetlabs/puppet/auth.conf`, has been deprecated. pupmod-simp-pupmod will back up the legacy `auth.conf` after the upgrade.

The Puppet Server's `auth.conf` is now managed by Puppet. You will need to reproduce any custom work done to legacy `auth.conf` via the new `puppet_authorization::rule` define. The stock rules are managed in `pupmod::master::simp_auth`.

Set up ClamAV DAT Files Updates

Given the wide spacing of SIMP releases, the team determined that it was ineffective for us to maintain the simp-rsync-clamav RPM with upstream ClamAV DAT file updates.

From this point forward, SIMP will not ship with updated ClamAV DAT files and we highly recommend updating your DAT files from the authoritative upstream sources. See the ClamAV Virus Database FAQ for instructions on how to automatically update these files.

Prepare System for PostgreSQL Upgrade

During the Puppet-managed upgrade, from PostgreSQL 9.4 to PostgreSQL 9.6, the PostgreSQL 9.4 data is not automatically imported into the 9.6 database. If for any reason you need to retain this data, which normally is quite transitory, see Upgrading a PostgreSQL Cluster for detailed instructions.

4.4.10 Troubleshooting Common Issues

How to troubleshoot common problems that occur when installing and using SIMP.

My Services Are Dying!

The following section describes how to mitigate issues relating to destructive reasoning and avoiding destruction of the SIMP system.

Destructive Reasoning with *svckill*

Most security guides that have been published on the Internet strongly suggest disabling all services that are not necessary for system operation. However, to list every possible service that may be controlled by the `chkconfig` or `systemctl` on a given system in a manifest would not be useful and would bloat the memory space of the running Puppet process.

As an alternative solution, the SIMP Team implemented the svckill module that runs with every Puppet run.

The svckill module:

- Collects a list of all services on the system. These are the same services that the user sees using `systemctl list-unit-files --type=service --state=enabled`
- Ignores certain critical services, including those for Puppet, IPtables/firewalld, and the network.
- Collects a list of all services that are defined in the manifests and modules.
- Ensures that every service that is defined in the manifests and modules is excluded from the list of services to kill.
- Kills and disables everything else.

Avoiding Destruction

If certain services should not be killed, you have two options:

1. Add the service names to the `svckill::ignore` array in *Hiera*.

```
svckill::ignore:
- keepmealive1
- keepmealive2
```

2. Declare the services in the node manifest space:

```
# Preventing these services from being killed by svckill
service { "keepmealive1": }
service { "keepmealive2": }
```

Note: The key to declaring the services in manifests is to use the `service` resource without setting any other options.

Why Can't I Login?!

If you have reached this page, you are having issues logging into your system with a newly created account.

In almost all cases, this is because either your user has not been placed in a group allowed to access the system, your *DNS* is set up incorrectly, or your *PKI* certificates are invalid.

SSSD Password Checks

SSSD has been made the default name service caching service in SIMP. During this process, we discovered that SSSD will enforce password complexity restrictions **upon login**. This means that, if your password does not meet the system password complexity requirements, you will not be able to login until an administrator changes your password to something stronger.

For the default complexity rules, see the *What is the Password Complexity for SIMP?* FAQ.

PAM Access Restrictions

By default, SIMP uses the `pam_access.so` *PAM* module to restrict access on each individual host. While this may not seem as flexible as some methods, it is the most failsafe method for ensuring that you do not accidentally interrupt services due to network issues connecting to your *LDAP* server.

To allow a user to access a particular system, you need to use the pam::access::rule define as shown below.

```
pam::access::rule { 'Allow the security group into the system':
  users   => ['(security)'],
  origins => ['ALL'],
  comment => 'The core security team'
}

pam::access::rule { 'Allow bob into the system from the proxy only':
  users   => ['bob'],
```

(continues on next page)

(continued from previous page)

```
  origins => ["proxy.${facts['domain']}"],
  comment => 'Bob the proxied'
}
```

Faillock

If a user fails to authenticate properly in **5** consecutive tries (the default `pam::deny`), *PAM* will lock the account.

To see a list of user authentication attempts, run `faillock`.

If a user is marked as invalid (I) or reaches the max number of attempts, you will need to reset `faillock` before authentication can occur. To do so, run

```
# faillock --reset --user <user>
```

LDAP Lockout

If your account is in *LDAP*, you may have locked yourself out. Like *PAM*, LDAP has a maximum number of logins, **5** by default.

For more information on unlocking LDAP accounts, see the appropriate section for the server that you are using in *Managing Accounts in Lightweight Directory Access Protocol (LDAP)*.

Troubleshooting DNS

If *PAM* is not the issue, you may be having *DNS* issues. This can evidence itself in two ways.

First, per the ‘Bob’ example above, you may be using an *FQDN* to identify a host on your network. If *DNS* is not properly configured, then there is no way for the host to understand that you should have access from this remote system.

Second, the default *PKI* settings in SIMP ensure that all connections are validated against the *FQDN* of the client system. In the case of an *LDAP* connection, a misconfiguration in DNS may result in an inability to authenticate against the *LDAP* service.

In the following sections, we will assume that we have a host named `system.my.domain` with the IP address `1.2.3.4`.

Testing a Forward Lookup

The following should return the expected IP address for your system.

```
dig +short system.my.domain
```

Testing a Reverse Lookup

The following should return the expected hostname for your system. This hostname **must** be either the primary name in the *PKI* certificate or a valid alternate name.

```
dig +short -x 1.2.3.4
```

PKI Issues

If both PAM and DNS appear to be correct, you should next validate that your *PKI* certificates are both valid and functional.

See *Checking Your SIMP PKI Communication* for additional guidance.

Checking Your SIMP PKI Communication

SIMP comes with a fully functional *Public Key Infrastructure* in the guise of an aptly named Fake CA.

The Fake CA can be very useful for getting your environment running prior to obtaining proper certificates from an official CA.

> **Warning:** The Fake CA is **not** hardware backed by default and should not be used for sensitive cryptographic operations unless there is no other alternative

Each Puppet environment that is part of a *SIMP Omni-Environment* contains its own Fake CA. That Fake CA is located within the corresponding *SIMP Secondary Environment*.

Basic Server Setup Check

Just as with Puppet certificates, the time on your system must be correct and your DNS must be fully functional. Check that these are correct before proceeding.

Fake CA Setup Check

For the remainder of this section, we will assume the following:

- The active Puppet environment returned by `puppet config print environment` is part of a *SIMP Omni-Environment*.
- The *FQDN* of the system with issues is `system.my.domain`.
- The LDAP server to which `system.my.domain` is attempting to connect is `ldap.my.domain`.

1. Change directories to the *keydist* directory for the active Puppet environment's Fake CA.

   ```
   # cd /var/simp/environments/`puppet config print environment`/site_files/pki_files/
   ↪files/keydist
   ```

2. Validate the client system. When validating certificates, you want to make sure that there are no errors regarding your certificate or *CA*. Ideally, the command will simply return the string 'OK'.

```
# openssl verify -CApath cacerts system.my.domain/system.my.domain.pub
```

3. Validate the LDAP system.

```
# openssl verify -CApath cacerts ldap.my.domain/ldap.my.domain.pub
```

If there are any issues, you may need to follow the steps in *Apply Certificates* to generate new certificates for one or more of your hosts.

Puppet Issues

Puppet Deprecation Warnings

Puppet 5 has added deprecation warnings to several settings. If one of these settings is present in a host's `puppet.conf`, you may see warnings at the beginning of every `puppet` run like this:

```
Warning: Setting configprint is deprecated.
(location: /opt/puppetlabs/puppet/lib/ruby/vendor_ruby/puppet/settings.rb:1169:in 'issue_
↪deprecation_warning')
```

These messages are innocuous. The deprecated settings can still be used until Puppet 6.

SIMP's `pupmod` module usually manages its settings in `puppet.conf` without using deprecated settings. However, under some circumstances it has no other choice, resulting in unavoidable error messages.

`Warning:  Setting ca is deprecated.`

`[master] ca` is marked as deprecated from Puppet 5.5.6 onward. This results in a harmless—but unavoidable—deprecation warning whenever `puppet` is run on Puppet Servers configured to *not* act as the Puppet CA (`pupmod::master::enable_ca:  false`):

```
Warning: Setting ca is deprecated.
(location: /opt/puppetlabs/puppet/lib/ruby/vendor_ruby/puppet/settings.rb:1169:in 'issue_
↪deprecation_warning')
```

Disabling *all* Puppet deprecation warnings

If you are encountering an unavoidable deprecation warning and find it unacceptable, you can suppress such warnings during your `puppet` runs by disabling *all* deprecation warnings:

> **Warning: This will disable all deprecation warnings**. If new settings are deprecated in future releases, you will not see warnings about them. This is particularly important if you manage additional `puppet.conf` settings. Use with caution!

```
pupmod::conf { 'disable_all_deprecation_warnings':
  section => 'main',
  setting => 'disable_warnings',
  value   => 'deprecations',
  confdir => $facts['puppet_settings']['main']['confdir'],
```

(continues on next page)

(continued from previous page)

```
  notify  => Service[puppetserver],
  ensure  => present,
}
```

This will ensure the following setting in `puppet.conf`:

```
[main]
disable_warnings = deprecations
```

If you want to enable deprecation warnings again, change `ensure => present` to `ensure => absent`.

Puppet Certificate Issues

Puppet Client Certificate Issues

Most of the time, clients will have certificate issues due to the system clock not being properly set. Before taking any other measures, make sure that your system clock is correct on both the server and the clients!

If you need to fix client certificate issues outside of time, first make sure that you do not have a certificate already in place on your *Puppet Server*.

```
# puppetserver ca list --all
```

If you **do** have a certificate in place, and need to register a client with the same name, remove that client's certificate from the system.

```
# puppetserver ca clean --certname <fqdn.of.the.client>
```

Warning: If you delete the Puppet Server's certificate, you will need to re-deploy Puppet certificates to **all** of your nodes!

Puppet Client Re-Registration

If, for some reason, you need to re-register your client with a new server, simply run the following on your client once the server is ready.

```
# rm -rf `puppet config print ssldir`
# puppet agent -t
```

After running the puppet agent, sign off the new certificate request on the *Puppet Server*.

```
# puppetserver ca list
# puppetserver ca sign --certname <cert req name>
```

Puppet Server Certificate Issues

Partial CA Setup

If the `puppetserver` is interrupted during the initial setup, you may discover that the process appears to hang forever without ever listening on the expected TCP port.

If this happens, when the process is killed (give it at least 10 minutes), you know that the CA has not set up properly if you see something like the following in the `puppetserver` log:

```
` Cannot initialize CA with partial state; need all files or none. Found:  /etc/
puppetlabs/puppetserver/ca/serial Missing:  /etc/puppetlabs/puppetserver/ca/ca_crt.pem
/etc/puppetlabs/puppetserver/ca/ca_key.pem /etc/puppetlabs/puppetserver/ca/ca_crl.pem
/etc/puppetlabs/puppetserver/ca/inventory.txt `
```

To remedy this issue, run the following commands:

```
` bash puppet resource file /etc/puppetlabs/puppetserver/ca ensure=absent force=true
recurse=true puppetserver ca setup `
```

The Server Certificate was Removed

To fix the issue where the Puppet Server certificate was removed using `puppetserver ca clean`, run `puppetserver ca generate --certname <your puppetserver's cert name>` and restart the `puppetserver` service.

The SSL Directory was Removed

If the `/etc/puppetlabs/puppet/ssl` directory was removed on the Puppet Server (and you do not have a backup of it) or for some other reason you need to regenerate all the Puppet certificates and the Puppet CA do the following:

> **Warning:** This is destructive to your Puppet communications. This should only be used if you have no other options.

1. Stop the `puppetserver` and `puppetdb` services.
2. Remove the certificates and the CA on the Puppet Server and generate the new Puppet Server and CA certificates.

   ```
   puppet resource service puppet ensure=stopped
   puppet resource service puppetserver ensure=stopped
   puppet resource service puppetdb ensure=stopped

   puppet resource file /etc/puppetlabs/puppet/ssl ensure=absent force=true recurse=true
   puppet apply -e '#noop'

   puppet resource service puppetserver ensure=running
   ```

3. Remove the old certificates from each of the Puppet clients and re-register the client using the *Puppet Client Re-Registration* instructions.
4. Clean the old certificates out from the puppetdb directory and copy the new ones from the puppetserver using puppetdb's ssl setup script.

```
puppet resource file /etc/puppetlabs/puppetdb/ssl ensure=absent force=true␣
↪recurse=true
puppetdb ssl-setup
```

5. (Optional) Restart the `puppetdb` service

   Only do this if you wish to run puppetdb!

```
puppet resource service puppetdb ensure=running
```

6. Confirm that everything is working

```
puppet agent -t
```

Puppetserver and PuppetDB Certificate Mismatch

If the puppetserver CA has been updated and the puppetdb certificates are not cleaned a puppet agent run produces an error similar to this:

```
Error: Could not retrieve catalog from remote server: Error 500 on SERVER: Server Error:
Failed to execute '/pdb/cmd/v1?checksum=5584595ca917e6b8d5767f7ff0fd71863fdfc486&
↪version=5
&certname=puppet.your.domain&command=replace_facts&producer-timestamp=1521137360'
on at least 1 of the following 'server_urls': https://puppet.your.domain:8139
```

You will probably need to disconnect the *PuppetDB* process from the *Puppet Server*. Follow the *HOWTO Disconnect PuppetDB from the Puppet Server* Guide to remediate this issue.

4.4.11 Package Data

Base Packages

Information about the base SIMP packages is best gathered from the `simp` RPM and the RPM metadata on your system.

The dependencies for the `simp` RPM are those that are required for basic SIMP functionality and may be obtained as follows from an **installed system**:

```
for x in `rpm -q --requires simp | cut -f 1 -d' '`; do
  rpm -q --qf "%{NAME} %{VERSION}\n" $x | grep -v 'not installed';
done
```

The dependencies for the `simp-extras` RPM are those that are **not** required for basic SIMP functionality and may be obtained as follows from an **installed system**:

```
for x in `rpm -q --requires simp-extras | cut -f 1 -d' '`; do
  rpm -q --qf "%{NAME} %{VERSION}\n" $x | grep -v 'not installed';
done
```

External Packages

Quite a few external packages are available to, and used by, the SIMP infrastructure.

These are defined, with sources, per OS and architecture in the `packages.yaml` files under the `build` directory in the `simp-core` repository.

To find the particular package list for your version of SIMP, you can go to:

`https://github.com/simp/simp-core/blob/<version>/build/distributions/<os>/<os_version>/<arch>/yum_data/packages.yaml`

For example, if you are using SIMP 6.0.0-0 on CentOS 8 and x86_64 architecture, you would navigate to:

`https://github.com/simp/simp-core/blob/6.0.0-0/build/distributions/CentOS/8/x86_64/yum_data/packages.yaml`

4.5 SIMP HOWTO Guides

This section provides guidance on configuration of various common capabilities in the SIMP system.

Items have been grouped by topic for ease of use.

4.5.1 Account and Access Guides

Guides in this section relate to system accounts and access control capabilities.

HOWTO Customize Settings for SSH

By default, SIMP includes the `ssh` class from the simp/ssh module under most of the *SIMP scenarios*. If it is absent from the scenario you have selected, add the following to *Hiera* to include the class:

```
simp::classes:
  - 'ssh'
```

To exclude the SIMP `ssh` class so that you can manage SSH configuration by other means, refer to *Disable SSH Management*.

The SIMP `ssh` class is configured to automatically include the `ssh::server` and `ssh::client` classes. These classes, in turn, manage the SSH daemon settings for incoming connections and the SSH client settings for outgoing connections, respectively. They configure SSH with reasonable defaults for the OS and environment.

If you want to disable SIMP management of either the SSH server or client settings, you can do so through Hiera. For example, to disable SIMP management of both:

```
ssh::enable_client: false
ssh::enable_server: false
```

Managing Settings for the SSH Server

The `ssh::server` class is responsible for configuring `sshd`. It delegates configuration of specific `sshd` settings found in **`sshd_config`** to `ssh::server::conf`. Detailed descriptions of the these settings are provided in the `sshd_config(5)` man page.

The `ssh::server::conf` class uses a number of SIMP-specific parameters, such as whether the system is FIPS enabled, has a firewall, or utilizes LDAP. This allows seamless integration of SSH with other SIMP-managed applications in the larger SIMP environment. For example, `ssh::server::conf` ensures the appropriate fallback ciphers are used, ensures proper authentication is configured, and allows SSH traffic through the firewall. More detailed information is provided in the simp/ssh module README file.

Configuring `ssh::server::conf` from Hiera

`ssh::server::conf` provides default configuration for key `sshd_config` settings and a mechanism to set most other settings.

To customize the SSH server, edit the parameters of `ssh::server::conf` using Hiera or *ENC*.

Note: Unlike many SIMP modules, these customizations cannot be made directly with a resource-style class declaration — they *must* be made via automatic parameter lookup provided by Hiera or an ENC. Examples using Hiera are provided for illustrative purposes.

In Hiera:

```
ssh::server::conf::port: 2222
ssh::server::conf::passwordauthentication: false
```

Starting with version **6.8.0** of the simp/ssh module, multiple ports can be specified to listen for incoming SSH connections. So the `ssh::server::conf::port` parameter in the previous example could be set as follows in Hiera to listen on multiple ports:

```
ssh::server::conf::port: [22, 2222, 22222]
```

Managing Additional Settings with `sshd_config`

Starting with version **6.7.0** of the simp/ssh module, you can manage additional settings not explicitly mapped to `ssh::server::conf` parameters, using the `ssh::server::conf::custom_entries` parameter. For example, to specify configuration for `AllowAgentForwarding` and `AuthorizedPrincipalsCommand` **`sshd`** settings, you would include Hiera such as the following:

```
ssh::server::conf::custom_entries:
  AllowAgentForwarding: "yes"
  AuthorizedPrincipalsCommand: "/usr/local/bin/my_command"
```

There are a few limitations with `ssh::server::conf::custom_entries` that need to be noted:

- ***No setting validation*:**

  This parameter is **not validated**. Be careful to only specify settings that are allowed for your particular SSH daemon and avoid duplicate declaration of settings already specified. Invalid options may cause the **`sshd`** service to fail on restart. Duplicate settings will result in duplicate Puppet resources (i.e., manifest compilation failures).

- ***No direct MATCH entry support*:**
  Due to their complexity, `Match` entries are not supported. However, you can add them using the `sshd_config_match` resource from the puppet/augeasproviders_ssh module. Since simp/ssh uses this module internally, the `sshd_config_match` resource will be available to you on any node using simp/ssh.

Managing Settings for the SSH Client

The `ssh::client` class is responsible for configuring default client settings for outgoing SSH sessions to all hosts (`Host *`).

Managing Settings for the Default Host Entry (`Host *`)

If you want to customize the default settings, you must prevent `ssh::client` from declaring them automatically and then declare `Host *` settings manually. You do this by setting `ssh::client::add_default_entry` to `false` and using the defined type `ssh::client::host_config_entry`. For example:

In Hiera:

```
ssh::client::add_default_entry: false
```

In Puppet:

```
ssh::client::host_config_entry{ '*':
  gssapiauthentication      => true,
  gssapikeyexchange         => true,
  gssapidelegatecredentials => true,
}
```

Managing Client Settings for Specific Hosts

Different settings for particular hosts can be managed by using the defined type `ssh::client::host_config_entry`:

```
# `ancient.switch.fqdn` only understands old ciphers:
ssh::client::host_config_entry { 'ancient.switch.fqdn':
  ciphers => [ 'aes128-cbc', '3des-cbc' ],
}
```

Managing Additional Settings with `ssh_config`

Starting with version **6.4.0** of the simp/ssh module, you can use the ssh_config resource from the puppet/augeasproviders_ssh module to manage settings that the module does not cover.

For instance, to ensure that the default host entry's `RequestTTY` option is set to `auto`:

```
# RequestTTY is not managed by ssh::client::host_config_entry
ssh_config { 'Global RequestTTY':
  ensure => present,
  key    => 'RequestTTY',
  value  => 'auto',
}
```

HOWTO Disable SSH Management

If the ssh class from simp/ssh is included in your SIMP scenario and you wish remove it from the class list and stop managing SSH configuration, add to your *Hiera* configuration as follows:

- To remove from the client nodes only:

```
---
simp::classes:
  - '--ssh'
```

- To remove from the client nodes and SIMP server:

```
---
simp::classes:
  - '--ssh'

simp::server::classes:
  - '--ssh'
```

Removing SIMP's ssh class also removes the iptables rule that allows connections to **sshd**. However, if the svckill class (from the simp/svckill Puppet module) is also included in your SIMP scenario, it will **not** automatically kill **sshd** when you cease management of the SSH configuration. This is because **sshd** has been whitelisted by svckill::ignore_defaults. So, if you want svckill to kill running **sshd** services, you must add the following to your Hiera configuration:

```
---
svckill::ignore:
  - '--sshd'
```

Note: The '--ssh' knockout prefix above **cannot** prevent ssh from being classified if it is included directly from Puppet code (e.g., include 'ssh') or by an *ENC*. However, you can stop managing the SSH client and server configurations with the following Hiera configuration:

```
---
ssh::enable_client: false
ssh::enable_server: false
```

HOWTO Enroll Hosts into an IPA Domain

Hosts should be able to join an IPA domain with a few catches. SIMP already uses the login stack that IPA uses (PAM, SSSD), but it also optionally manages the same resources that IPA provides automation for. This includes:

Technology	Related SIMP Features	Related Tickets
sudoers	simp/simp and simp/sudo modules	SIMP-4898
autofs	optional simp/simp_nfs module	SIMP-4168
krb5	optional simp/krb5 module	SIMP-4167

The features in the above table may work in the future, but logins via **sssd** or LDAP should work without issue, now.

IPA should work in both the simp and simp_lite *scenarios*. There may be issues with logins if the simp/sssd module is not included.

Adding IPA Clients

Adding IPA clients requires two steps:

1. Add the hosts on the IPA server, setting a one time password.
2. Join each host to the IPA domain by running **`ipa-client-install`** on the host with the password generated in the previous step.

The second step is automated with the simp/simp_ipa module.

> **Note:** Using **`ipa-client-install`** on EL6 with FIPS mode is not currently supported and will result in the following error message: `Cannot install IPA client in FIPS mode`

Add Hosts to IPA

There are two ways to complete this step:

1. Use the IPA web interface, and take note of the one time password.
2. Run **`ipa host-add`** on the command line and pre-generate the password.

Only option 2 will be covered here.

To add hosts from the command line:

1. Log onto a machine that already has joined an IPA domain.
2. **`kinit`** into an account with the appropriate privileges, (e.g., **`kinit admin`**).
3. Use a script such as the example below, to add hosts in bulk:

```
#!/opt/puppetlabs/puppet/bin/ruby

# This scripts adds hosts to IPA using fqdn/IP address pairs listed in a
# 'hosts' input file and generated passwords of the form
#
#   <fqdn>-<random string>
#
# The <random string> portion will be the same for all host passwords
# in a specific run.
#
# The input file must contain lines formatted as follows:
#
#   <fqdn>,<IP address>
#
# such as
#
#   ws1.example.domain,192.168.1.3
#
require 'securerandom'

unless File.exist?('hosts')
  $stderr.puts "ERROR: Could not find 'hosts' file."
  exit 1
```

(continues on next page)

(continued from previous page)

```
end

password_suffix = SecureRandom.urlsafe_base64(8)
puts 'Using one-time passwords of the form of <fqdn>-' + password_suffix

File.readlines('hosts').each do |h|
  # skip comment lines and blank lines
  next if (h[0] == '#') || (h.strip.empty?)

  unless h.include?(',')
    $stderr.puts "WARN: Skipping malformed entry: '#{h.strip}'"
    next
  end

  fqdn, ip = h.split(',')
  fqdn.strip!
  ip.strip!
  unless !fqdn.empty? && ip && !ip.empty?
    $stderr.puts "WARN: Skipping malformed entry: '#{h.strip}'"
    next
  end

  cmd = "ipa host-add #{fqdn} --ip-address=#{ip} --password=#{fqdn}-#{password_
↪suffix}"
  puts cmd
  unless system(cmd)
    $stderr.puts "ERROR:  Command failed '#{cmd}'"
    $stderr.puts 'Exiting!'
    exit 2
  end
end
```

Join a Host to the IPA Domain

To join the host to the IPA domain, use `simp_ipa::client::install` from the simp/simp_ipa Puppet module, by setting the hieradata as shown in the examples below.

The following examples assume

- the IPA server is `ipa.example.domain` with an IP address of `192.153.1.2`
- the IPA domain is `example.domain`
- the IPA realm is `EXAMPLE.DOMAIN`.

```
# In the appropriate level hieradata file
---
simp::classes:
  # this will include this class in client node manifests
  - simp_ipa::client::install

simp_ipa::client::install::ensure: present
```

(continues on next page)

(continued from previous page)

```
# Set this to the one-time password generated when the host was added to IPA.
# This example assumes you used the example script.
simp_ipa::client::install::password: "%{trusted.certname}-<OTP suffix>"

# Set this to the IPA server FQDN
simp_ipa::client::install::server: ipa.example.domain

# Set these to match your IPA domain and realm
simp_ipa::client::install::domain: example.domain
simp_ipa::client::install::realm: EXAMPLE.DOMAIN
```

In addition to the above settings, other settings may be needed, depending on the configuration of the IPA server and the environment:

```
---
# IPA uses both of these technologies, so they need to be enabled.
# SSSD is already enabled in the 'simp' and 'simp_lite' scenarios.
simp_options::sssd: true
simp_options::ldap: true

# These 4 parameters have to be set, even though they may be unused because
# IPA does not, natively, set up a BIND DN or a SYNC DN.  If your IPA server
# has those DNs and you are using a SIMP module that uses them (e.g.,
# simp-simp_gitlab), be sure to set them to the real values.  It is likely
# you will also have to set the commented out parameters as well!
simp_options::ldap::bind_pw: "A-Unused-LDAP-Bind-Password"
simp_options::ldap::bind_hash: "{SSHA}this-is-not-a-real-password-hash"
simp_options::ldap::sync_pw: "A-Unused-LDAP-Sync-Password"
simp_options::ldap::sync_hash: "{SSHA}this-is-not-a-real-password-hash"
#simp_options::ldap::base_dn: FILL-ME-IN-AS-NEEDED
#simp_options::ldap::bind_dn: FILL-ME-IN-AS-NEEDED
#simp_options::ldap::sync_dn: FILL-ME-IN-AS-NEEDED
#simp_options::ldap::root_dn: FILL-ME-IN-AS-NEEDED
#simp_options::ldap::master:  FILL-ME-IN-AS-NEEDED
#simp_options::ldap::uri:     [ FILL-ME-IN-AS-NEEDED ]

# If the IPA server is a DNS server, this will allow you to use the DNS
# SRV records to discover other IPA provided services, like LDAP and krb5.
simp_options::dns::servers:
  # IP address of IPA server
  - 192.153.1.2

# Other DNS-related settings that may fix issues that pop up.
simp_options::dns::search:
   # IPA domain
   - example.domain
   resolv::named_autoconf: false
   resolv::caching: false

   # IPA domain
   resolv::resolv_domain: example.domain
```

Next time Puppet runs, your node will be part of the IPA domain and appropriate logins should work.

IPA User Accounts

Once a host has been joined to the IPA domain following the instructions above, users should be able to login with SSSD or LDAP. However, there are a few nuances about user accounts that are worth noting:

- Only users that are in an IPA group of type POSIX will be able to log into Linux systems. You may need to add such a group on the IPA server. For example, to add a POSIX group named posixusers via the command line:

  ```
  kinit admin
  # by default this will be a POSIX group
  ipa group-add posixusers --desc "A POSIX group for users"
  ```

- The default UID and GID ranges are very high in IPA (generated randomly by default and can be in the low billions), so they are a lot higher than both the SIMP and SSSD default max. You have a couple of options on how to avoid this issue:
  - Set the start user and group number when you install the IPA server by using the **--idstart** command line option (e.g., **ipa-server-install --idstart=5000**)
  - Change the UID/GID ranges in the IPA GUI.
  - Set simp_options::uid::max to match that of your existing IPA server.
- Users and groups still have to be added to PAM to be able to log in! You will need to allow access using the pam::access::rule define from the simp/pam Puppet module. For example, to allow access to the posixusers group created above:

  ```
  pam::access:rule { 'Allow IPA posixusers group into the system':
    users   => [ '(posixusers)' ],
    origins => [ $simp_options::trusted_nets ],
    comment => 'group for IPA users'
  }
  ```

HOWTO Enable Redundant LDAP

This section describes how to set up consumer *OpenLDAP* servers in SIMP.

389-DS instructions will be made available at a future date.

Note: The concept of master/slave has been deprecated by the OpenLDAP project in favor of the more flexible concept of provider/consumer. The SIMP team is in the process of updating both our code and documentation to reflect this. Please bear with us in this process, and feel free to pose any questions via any of the SIMP community resources

Set up the Primary Server

The easiest way to set up an LDAP primary server is to set it up on the Puppet server using **simp config** during the initial configuration of the Puppet server. This is done by answering "yes" when asked if you want to use LDAP during your initial **simp config** run and answering the basic questions it asks you. If it is not desirable to have the LDAP primary server on the Puppet server, the LDAP primary server can be set up on an alternate server by including the simp::server::ldap on the node of your choice.

Note: If you use another node, you may want to re-run `simp config` and answer the questions with this new LDAP primary server in mind.

If you do not want to run **simp config** again, you will need to configure the following settings in *Hiera*:

```
# === ldap ===
# Whether or not to use LDAP on this system.
# If you disable this, modules will not attempt to use LDAP where possible.
simp_options::ldap: true

# The Base DN of the LDAP server
simp_options::ldap::base_dn: "dc=your,dc=domain"

# LDAP Bind Distinguished Name
simp_options::ldap::bind_dn: "cn=hostAuth,ou=Hosts,%{hiera('ldap::base_dn')}"

# The LDAP bind password
simp_options::ldap::bind_pw: "MyRandomlyGeneratedLargePassword"

# The salted LDAP bind password hash
simp_options::ldap::bind_hash: "{SSHA}9nByVJSZFBe8FfMkar1ovpRxJLdB0Crr"

# The DN of the LDAP sync user
simp_options::ldap::sync_dn: "cn=LDAPSync,ou=Hosts,%{hiera('ldap::base_dn')}"

# The LDAP sync password
simp_options::ldap::sync_pw: "MyOtherRandomVeryLargePassword"

# The SSHA hash for ldap::sync_pw
simp_options::ldap::sync_hash: "{SSHA}VlgYUmRzyuuKZXM3L8RT28En/eqtuTUO"

# The LDAP root DN.
simp_options::ldap::root_dn: "cn=LDAPAdmin,ou=People,%{hiera('ldap::base_dn')}"

# The LDAP root password hash.
# If you set this with simp config, type the password and the hash will be
# generated for you.'
simp_openldap::server::conf::rootpw: "{SSHA}GSCDnNF6KMXBf1F8eIe5xvQxVJou3zGu"

# This is the LDAP server in URI form (ldap://server)
simp_options::ldap::master: ldap://ldap_server1.your.domain

# === ldap::uri ===
# List of OpenLDAP servers in URI form (ldap://server)
simp_options::ldap::uri:
  - ldap://ldap_server1.your.domain

# === sssd::domains ===
# A list of domains for SSSD to use.
# `simp config` will automatically populate this field with `FQDN` if
# `use_fqdn` is true, otherwise it will comment out the field.
#
```

(continues on next page)

(continued from previous page)

```
sssd::domains:
  - LDAP
```

Add the `simp::server::ldap` class into the yaml file for the LDAP server in Hiera, for example: `data/hosts/ldap_server1.your.domain.yaml`:

```
simp::classes:
  - 'simp::server::ldap'
```

Leave any other classes that are there if they are needed. Run the Puppet agent on the LDAP server until it runs cleanly. Run the agent on the Puppet server. Once all the other clients update against the Puppet server, they will be able to authenticate against the LDAP server. Adding users and groups is described in the *User Management* section.

Note: Information on how the create salted ({SSHA}) passwords can be found at the OpenLDAP site.

Set up the Redundant (Consumer) Servers

Default Settings

Once the LDAP primary server is ready, LDAP consumer nodes can be configured to replicate data from the primary server. These consumer servers are read-only, and modifications cannot be made to LDAP entries while the primary server is down.

Consumer nodes can be configured via Hiera by setting `simp::server::ldap::is_consumer` to `true`, setting the replication id (RID) , and adding the `simp::server::ldap` class. This will set up your redundant server using the defaults. To do these three things, add the following lines to the `data/hosts/ldap_server2.your.domain.yaml` file:

```
simp_openldap::server::conf::rootpw: "{SSHA}GSCDnNF6KMXBf1F8eIe5xvQxVJou3zGu"
simp::server::ldap::is_consumer: true
simp::server::ldap::rid: 888

simp::classes:
  - 'simp::server::ldap'
```

To make other clients aware of this server, add the redundant server's URI to lists of URIs in the `data/default.yaml` file:

```
# === ldap::uri ===
# List of OpenLDAP servers in URI form (ldap://server)
simp_options::ldap::uri:
  - ldap://ldap_server1.your.domain
  - ldap://ldap_server2.your.domain
```

Note: To see the defaults for LDAP replication in SIMP, review the parameters passed to the module `simp_openldap/manifests/server/syncrepl.pp`. These parameters are used to add the replication settings to the `syncrepl.conf` file. Definitions can be found in the syncrepl.conf (5) man page.

Custom Replication Settings

If settings other than the defaults are needed, create a custom manifest and use the `simp_openldap::server::syncrepl` class with the necessary parameters.

In this example, the *site profile* is called `site::ldap_consumer` and the RID of the server is `999` (these can be changed). One setting, `sizelimit`, is being overwritten but you can overwrite any number of them.

```
class site::ldap_consumer {

  include 'simp::server::ldap'

  # custom settings:
  simp_openldap::server::syncrepl { '999':
    sizelimit  => '5000',
  }
}
```

The name of the `simp_openldap::server::syncrepl` instance must be a unique replication id.

Place this file in the `site` module's `manifests/` directory using the name `ldap_consumer.pp`. Include this class from the ldap server's Hiera YAML file:

```
simp::classes:
  - 'site::ldap_consumer'
```

Lastly, add the server to the *URI* listing in `default.yaml` so all the clients know about it once they have updated from the Puppet server.

Promote a Consumer Node

A consumer node can be promoted to act as an LDAP primary server. To do this, change the node classifications of the relevant hosts. For a node with the default settings, just remove the `simp::server::ldap::is_consumer:  true` from the server's Hiera YAML file and change the setting for the LDAP primary server in Hiera. This setting is needed by all LDAP servers. (It defaults to the Puppet server if it is not set.)

```
# This is the LDAP primary server in URI form (ldap://server)
simp_options::ldap::master: ldap://ldap_server2.your.domain
```

For a redundant server setup using custom settings, remove the call to the custom class and replace it with the call to the `site::ldap_server` class in the servers yaml file and set the primary server setting in the Hiera as shown above.

In both cases, if the current primary server is not down, make sure it has completed replication before changing the settings. Once the settings are changed, run **`puppet agent -t`** on the LDAP primary server. After the next Puppet run on all the hosts the server will be promoted to primary server and all the consumers will point to it.

Remove a Node or Demote a Primary LDAP Server

To demote the primary server, simply configure it as consumer in either of the configurations above after the new server has been configured and put in place. Then run the Puppet agent. Lastly, manually remove the active database from the server. (Check the setting `simp_openldap::server::conf::directory` setting for the location of the files.)

To remove an LDAP server, first remove the server from the `simp_options::ldap::uri` settings in Hiera. Give the clients time to update from the Puppet server so they do not attempt to call it. Then remove relevant settings from its hiera.yaml file and run the Puppet agent.

Troubleshooting

If the system is not replicating, it is possible that another user has updated the `simp_options::ldap::sync_pw` and `simp_options::ldap::sync_hash` entries in Hiera file without also updating the value in LDAP itself; this is the most common issue reported by users. If simp config was used to set up the server these values are in the `simp_config_settings.yaml` file.

Currently, SIMP cannot self-modify the LDAP database directly; therefore, the LDAP Administrator needs to perform this action. Refer to the *User Management* chapter for more information on manipulating entries in LDAP.

The example below shows an example ldif used to update the sync user information in LDAP.

```
dn: cn=LDAPSync,ou=Hosts,dc=your,dc=domain
changetype: modify
replace: userPassword
userPassword: <Hash from simp_options::ldap::sync_hash>
```

Likewise if the bind password has changed in hiera, the `simp_options::ldap::bind_pw` and `simp_options::ldap::bind_hash` in the `simp_config_settings.yaml` file, the password must be updated in LDAP. If it is not, the clients will not be able to connect to the LDAP server. Use the following LDIF to update the bind entry in LDAP:

```
dn: cn=hostAuth,ou=Hosts,dc=simp,dc=test
changetype: modify
replace: userPassword
userPassword: <Hash from simp_options::ldap::bind_hash>
```

Further Information

The OpenLDAP site contains more information on configuring and maintaining OpenLDAP servers.

HOWTO Enable SFTP Restricted Accounts

This section describes the method for restricting an account to *SSH File Transfer Protocol* (SFTP) access only.

Add a User

Create a user account based on the following example.

```
user { "foo":
  uid   => <UID>,
  gid   => <GID>,
  shell => '/usr/libexec/openssh/sftp-server'
}
```

Modify `/etc/shells`

To allow your user to use the **sftp-server** application as a shell, you will need to add custom shell to `useradd::shells` in *Hiera* as shown below.

```
useradd::shells:
  - /usr/libexec/openssh/sftp-server
```

HOWTO Set up SSH Authorized Keys

This section provides guidance on managing SSH authorized keys within the SIMP environment.

LDAP Enabled

When enabled, SSH authorized keys are both stored and retrieved directly from LDAP.

See Also: *Managing Users with LDAP*

Without LDAP

If not using LDAP, or in addition to LDAP, SSH authorized keys can be placed in `/etc/ssh/local_keys/<USERNAME>`. This file location is controlled by the simp/ssh Puppet module and can be changed by setting the `ssh::server::conf::authorizedkeysfile` parameter in *Hiera* or your *ENC*.

See Also: *Managing Local/Service Users*

HOWTO Restrict Network Access to SSH

Like most SIMP modules, the SSH module utilizes a `trusted_nets` parameter to control access to the SSH service via both IPTables and TCPWrappers.

Since there is no way for the SIMP installation to successfully guess where you may be connecting from, or know about your particular network architecture, it defaults to allowing SSH connections from **any** host.

It is understandable that you may want to restrict this further. To do so, you simply need to set the `ssh::server::conf::trusted_nets` to an `Array` of networks or hosts from which you would like to connect.

Example: Set Trusted Nets to Alternate Networks via Hiera

```
---
ssh::server::conf::trusted_nets :
  - 1.2.3.4
  - 10.1.2.0/24
  - 192.168.0.0/16
```

You can find more information on `trusted_nets` in the *Installation Variables* in the *Advanced Configuration* section of the *SIMP Getting Started Guide*.

4.5.2 SIMP Guides

Guides in this section relate to configuring the SIMP framework itself.

HOWTO Use an Alternate 'simp config' Environment

Generally, when running the **`simp config`** and **`simp bootstrap`** commands, all items will be placed into the `production` *Puppet Environment* and *SIMP Secondary Environment*.

If you are installing SIMP onto a preexisting system, you may want to instead install into an environment other than `production`.

As of `simp-cli` version 6.0.0 (new with SIMP 6.5), you can configure the **`simp config`** and **`simp bootstrap`** commands to use an alternate initial environment by setting the shell environment variable, `$SIMP_ENVIRONMENT`.

For example, if you wanted to name your initial environment `simp`, then you would do the following:

```
export SIMP_ENVIRONMENT='simp'
simp config
...
simp bootstrap
```

See *Advanced Configuration* for more information on **`simp config`**.

HOWTO Configure SIMP for Compliance Enforcement

SIMP provides the capability to easily enforce specific compliance settings across all Puppet classes as long as a suitable *SIMP compliance profile* has been created.

See *The SIMP Compliance Engine* for full details on how to configure and use the compliance enforcement capabilities.

HOWTO Modify the Nightly Update Schedule

By default, SIMP applies *Nightly Updates* from all configured repositories.

This behavior is controlled by the simp::yum::schedule class from the `simp-simp` Puppet module and the parameters therein can be used to modify the schedule.

If you simply want to disable the nightly updates, you can set `simp::yum::schedule::enable` to `false` in *Hiera*.

HOWTO Fully Disable the System Firewall

Though we hope that you never actually want to do this, there may be situations where you want to use puppet to fully disable the system firewall.

When **`iptables`** was the only option, this was very straightforward. The introduction of **`firewalld`** has added a bit of complexity due to the preservation of backwards compatibility with calls into the `iptables::rules::*` *defined types*.

To fully disable **all** firewalls on the system (not just management of the firewalls) set the following via *Hiera*:

```
iptables::enable: false
firewalld::service_enable: false
firewalld::service_ensure: 'stopped'
```

As per usual, once this is set, Puppet will ensure that the firewall is fully disabled until the settings are reversed.

Important: Just setting `firewalld::service_enable: false` will likely cause your system to fall back to using **`iptables`**.

HOWTO Enable STIG Mode on a SIMP System

Enabling *STIG* mode in SIMP involves using *The SIMP Compliance Engine* to apply the STIG-specific *SIMP compliance profile*.

Like everything in *Puppet*, STIG-mode only applies to those nodes that actively include the correct settings.

Assumptions

1. You have a fully functional SIMP system up and running per the instructions in *Installation Options*.
2. The *FQDN* of your node is `stig.your.domain` and that you only want to enable STIG mode for this node. You can use *Hiera* to enable it for all nodes if you so choose.
3. You are using the `production` *Puppet Environment*
4. You have a regular user named `stiguser` that will be used for remote access to your system and for escalation of privileges to `root`. This is due to the fact that, by default, SIMP does not allow remote `root` access.

Setting up Hiera

First, we need to create the file `/etc/puppetlabs/code/environments/production/data/hosts/stig.your.domain.yaml`. The next few sections describe what should be added to this file to ensure that *STIG* mode will be activated and that your user can properly login to the system.

Allowing `stiguser` to login

```
pam::access::users:
  stiguser:
    origins:
      - ALL
```

Allow `stiguser` to escalate to `root`

```
sudo::user_specifications:
  stiguser_su:
    user_list:
      - stiguser
    cmnd:
      - ALL
    # You should change this to 'true' for full STIG compliance
    # Ensure that you have a working password set first!
    passwd: false
    options:
      # Required for seamless root sudo commands
      role: unconfined_r
```

Configure selinux to allow `stiguser` to run privileged commands

```
selinux::login_resources:
  stiguser:
    seuser: staff_u
    mls_range: "s0-s0:c0.c1023"
```

Place the system in STIG-enforcing mode

Now that you have ensured that `stiguser` can access your system as well as escalate to an administrative user without being blocked by SELinux, you are ready to enable STIG-enforcing mode.

First, include the SIMP Compliance Engine backend in the hierarchy defined in the environments hiera.yaml, `/etc/puppetlabs/code/environments/production/hiera.yaml`. Place it under the hierarchy tag just before default:

```
---
version: 5
defaults:
  datadir: data
  data_hash: yaml_data

hierarchy:

...

- name: SIMP Compliance Engine
```

(continues on next page)

(continued from previous page)

```
  lookup_key: compliance_markup::enforcement

- name: General data
  paths:
  - "default.yaml"
  - "common.yaml"

...
```

Then add the following to the `stig.your.domain.yaml` file that we have been editing:

```
compliance_markup::enforcement:
  - disa_stig
```

Next Steps

Applying the changes

At this point, your system is ready to apply the STIG enforcement settings. To begin enforcement, simply run **`puppet agent -t`** on the `stig.your.domain` node or wait for the next scheduled run of **`puppet`**.

Escalating privileges

Remote system access should work as you would expect but there is a new caveat to how you would normally run `sudo` to access the `root` account.

Since the STIG requires that all users be in an SELinux context, you will need to ensure that all administrative users are `staff_u` users as we did above.

Once this is complete, you must tell `sudo` what context you wish to transition into when running commands.

The simplest invocation is as follows:

```
[stiguser@localhost ~]$ sudo su - root
```

For additional information see the vendor documentation on confined and unconfined users and/or Dan Walsh's blog.

HOWTO Generate a Puppetfile

This section describes how to create a *Puppetfile* to be used by r10k or code manager to deploy SIMP and site modules. It assumes the user is on the puppetserver. It covers building from Local Repositories or using the Internet repositories

> **Warning:** Any module not listed in the `Puppetfile` will be deleted from the target environment's `modules` directory, when you use *r10k* to deploy the modules.

- *Generate the Puppetfile skeleton*
- *Edit the Puppetfile to include non-simp site modules*

Generate the Puppetfile skeleton

```
simp puppetfile generate -s > Puppetfile
```

Edit the Puppetfile to include non-simp site modules

Make sure the `Puppetfile` used to deploy from includes the following:

- A line that includes the `Puppetfile.simp` which should look like:

  ```
  instance_eval(File.read(File.join(__dir__,"Puppetfile.simp")))
  ```

- A line for each of your own modules.

  The SIMP *CLI* can help identify local modules in an environment that need to be listed in the `Puppetfile`, because they may not be under Git source control and, when removed by **`r10k`**, may not be able to be readily restored.

  To generate a list of local modules in an environment (`production` in this example) do the following:

  ```
  simp puppetfile generate --skeleton --local-modules production  > /tmp/
  ↪Puppetfile
  ```

  This will generate `/tmp/Puppetfile` which has

  - a directive to include the file `Puppetfile.simp`
  - a local entry for each module presently in the `production` environment, that does not have a local, SIMP-managed Git repository and does not appear to be managed by Git.

  The local entries will look like the following:

  ```
  mod 'module name', :local => true
  ```

  Verify that all modules with a local entry in `/tmp/Puppetfile` are in your environment's `Puppetfile` in one of the following forms:

  ```
  # a module that is not a Git repository and resides in the ``modules``␣
  ↪directory
  mod 'site',
    :local => true

  # a Git repository that resides in a directory on the Puppet server
  mod 'mymodule',
   :git => 'file:///usr/share/mymodules/mymodule',
   :tag => '1.1.1'

  #  a Git repository on a remote server
  ```

  (continues on next page)

(continued from previous page)

```
mod 'mysrvmod',
  :git => 'https://gitserver.my.domain/mygitproject/mysrvmod.git',
  :tag => '1.0.1'
```

Note: Any modules on the local system (ones that use the `:local => true` directive), should be converted to a Git repository. This will ensure that if **`r10k`** removes them because of an error in the `Puppetfile` they can easily be restored from the Git repository.

Generate the `Puppetfile.simp` file

Using local SIMP git repositories

Make sure all the SIMP module RPMs are installed using yum that you want to use and then run:

```
simp puppetfile generate > Puppetfile.simp
```

Using SIMP internet repositories

- Download the `Puppetfile` used to create a SIMP ISO for a specific release from the SIMP simp-core repository. This example uses the SIMP `6.4.0-0` release.

```
curl -o Puppetfile.simp https://github.com/simp/simp-core/blob/6.4.0-0/Puppetfile.
↪pinned
```

- Manually edit the `Puppetfile.simp` to remove components that are not Puppet modules, by deleting all lines up to and including `moduledir 'src/puppet/modules'`.
- Optionally, edit the `Puppetfile.simp` to remove any non-core SIMP modules that are packaged with the `simp-extras` RPM, that are not needed. The list of the SIMP extra modules can be obtained by examining the RPM requirements of the `simp-extras` RPM as follows:

```
yum deplist simp-extras
```

Copy the `Puppetfile` and `Puppetfile.simp` to top level of the environment directory or the top level of the control repo.

HOWTO Manage reboot_notify

`simplib::reboot_notify` was added to provide a capability for notifying users, across puppet runs, that their system needed to be rebooted.

It was originally created due to updates to kernel settings requiring a reboot but discovering that users would often forget to reboot unless a message continued to appear in front of them.

Disabling Reboot Notifications

Technically, reboot notifications cannot be excluded from the puppet catalog. However, you can silence them by setting `simplib::reboot_notify::log_level` to the level of your choosing.

For instance, setting the following via *Hiera* will ensure that you only see log messages when running in `debug` mode:

```
---
simplib::reboot_notify::log_level: debug
```

You may use any values that are valid for the puppet loglevel metaparameter.

HOWTO Manage Workstation Infrastructures

This chapter describes example code used to manage client workstations with a SIMP system including the *GUI*, repositories, virtualization, printing, and *Virtual Network Computing* (VNC).

The examples in this chapter assume the `production` *Puppet environment* is being configured and site-specific manifests are being added to a local `site` profile module at `/etc/puppetlabs/code/environments/production/modules/site/manifests`.

Install Extra Puppet Modules

The examples on this page use SIMP modules that are part of SIMP Extras and may not be installed on the Puppet server by default. The following is an example class, `site::workstation_packages`, that may be written to `site/manifests/workstation_packages.pp` and applied to the Puppet server to install the extra modules, when RPMs are being used to distribute the modules:

```
class site::workstation_packages {

  $package_list = [
    'pupmod-simp-dconf',
    'pupmod-simp-gdm',
    'pupmod-simp-gnome',
    'pupmod-simp-simp_nfs',
    'pupmod-simp-vnc',
    'pupmod-simp-libvirt',
  ]

  package { $package_list :
    ensure => installed,
  }
}
```

Create a Workstation Profile Class

Below is an example class, `site::workstation`, that may be written to `site/manifests/workstation.pp` and used to set up a user workstation. Each included `site::` class is described in the subsequent sections.

```
class site::workstation {
  include 'site::repos'
  include 'site::gui'
  include 'site::virt'
  include 'site::print::client'

  # make sure any repos are installed before they
  # are needed.  Include dependencies to
  # other classes if needed.
  Class[Site::Repos] -> Class[Site::Gui]

  # Make sure everyone can log into all nodes.
  # If you want to change this, simply remove this line and add
  # individual entries to your nodes as appropriate
  pam::access::rule { "Allow Users":
    comment => 'Allow all users in the "users" group to access the system from anywhere.',
    users   => ['(users)'],
    origins => ['ALL']
  }

  # Install additional packages on the workstations.
  # Example list of General Use Packages
  package { [
    'pidgin',
    'vim-enhanced',
    'tmux',
    'git'
  ]: ensure => installed,
     require => Class[Site::Repos]
  }
}
```

Workstation Repositories

The `site::repos` class below should be written to `site/manifests/repos.pp` and contain the `yumrepo` resources required to install the extra software needed:

```
class site::repos {
  yumrepo { 'myrepo':
    # whatever parameters you need
  }
}
```

Graphical Desktop Setup

The following example class, `site::gui`, may be written to `site/manifests/gui.pp` and used to set up a graphical desktop.

```
class site::gui (
  Boolean $libreoffice = true
) {

  include 'gdm'
  include 'gnome'
  include 'vnc::client'
  # Browser and e-mail client are not installed by default.
  include 'mozilla::firefox'

  Class['Gnome'] -> Class['Site::gui']

  # SIMP gnome package provides a basic interface.
  # Add gnome extensions for the users.
  package { [
    'gnome-color-manager',
    'gnome-shell-extension-windowsNavigator',
    'gnome-shell-extension-alternate-tab',
    ]:
     ensure => installed,
  }

  # GUI applications
  if $libreoffice {
    package { 'libreoffice': ensure => installed }
  }
}
```

Virtualization on User Workstations

The following example class, `site::virt`, may be written to `site/manifests/virt.pp` and used to allow users to run **`libvirt`** virtual machines.

Important: The `libvirt::polkit` class being called sets which users are allowed to use `libvirt` from the command line.

```
# If you want users to be able to run VMs on their workstations
# include a class like this.
# If this is installed, VM creation and management is still limited by PolicyKit

class site::virt {
  include 'libvirt::kvm'
  include 'libvirt::ksm'
  include 'swap'
  include 'network'
```

(continues on next page)

(continued from previous page)

```
  #set up a local bridge on the network
  network::eth { "em1":
    bridge => 'br0',
    hwaddr => $facts['macaddress_em1']
  }

  network::eth { "br0":
    net_type => 'Bridge',
    hwaddr   => $facts['macaddress_em1'],
    require  => Network::Eth['em1']
  }

  #add virt-manager package
  package { 'virt-manager': ensure => 'latest' }

  # Create polkit policy to allow users in virsh users group to use libvirt
  class { 'libvirt::polkit':
    ensure => present,
    group  => 'virshusers',
    local  => true,
    active => true
  }

  #Create group and add users.
  group{ 'virshusers':
    members => ['user1','user2']
  }

}
```

To set specific *swappiness* values use *Hiera* as follows:

```
# Settings for swap for creating/running virtual machines
swap::high_swappiness: 80
swap::max_swappiness: 100
```

Printer Setup

Below are example manifests for setting up a printing environment.

Setting up a Print Client

The following example class, `site::print::client` may be written to `site/manifests/print/client.pp` and sets up client-side printing:

```
class site::print::client inherits site::print::server {
  polkit::local_authority { 'print_support':
    identity           => ['unix_group:*'],
    action             => 'org.opensuse.cupskhelper.mechanism.*',
    section_name       => 'Allow all print management permissions',
    result_any         => 'yes',
    result_interactive => 'yes',
    result_active      => 'yes'
  }

  package { 'cups-pdf': ensure => 'latest' }
  package { 'cups-pk-helper': ensure => 'latest' }
  package { 'system-config-printer': ensure => 'present' }
}
```

Setting up a Print Server

The following example, `site::print::server`, may be written to `site/manifests/print/server.pp` and sets up server-side printing:

```
class site::print::server {

  # Note, this is *not* set up for being a central print server.
  # You will need to add the appropriate IPTables rules for that to work.
  package { 'cups': ensure => 'latest' }

  service { 'cups':
    enable     => 'true',
    ensure     => 'running',
    hasrestart => 'true',
    hasstatus  => 'true',
    require    => Package['cups']
  }
}
```

Using hostgroups to Apply the Settings

Once the workstation profile has been created and tested, one way of applying the profile to all workstations is to use the SIMP `hostgroups` *Hiera* configuration capability.

To use `hostgroups`, you will need to edit the `site.pp` in the target *Puppet environment site manifest*.

Adding the following to `/etc/puppetlabs/code/environments/production/manifests/site.pp` will make all nodes whose names start with `ws` followed by any number of digits use the `data/hostgroups/workstation.yaml`. All other nodes will fall back to the `default.yaml`.

```
case $facts['hostname'] {
  /^ws\d+.*/: { $hostgroup = 'workstation' }
  default:    { $hostgroup = 'default'     }
}
```

The workstation.yaml file will include settings for all the workstations.

The following example includes the GUI settings discussed here, in addition to settings for NFS mounted home directories. See *Exporting Home Directories for LDAP Users HOWTO Configure NFS* and for more information.

```
---

# Set the run level so it will bring up a graphical interface
simp::runlevel: 'graphical'
timezone::timezone: 'EST'

# Settings to mount home server
simp_nfs::home_dir_server: <home server's IP address>

# The site::workstation manifest will do most of the work.
simp::classes:
  - site::workstation
  - simp_nfs
```

Graphical Remote Access

This section describes how to install x2go and *VNC* software on nodes to access the desktop remotely.

x2go and MATE

Follow the instructions in *Install Extra Puppet Modules* to install the following Puppet modules on the Puppet server:

- pupmod-simp-x2go
- pupmod-simp-mate
- pupmod-simp-gnome
- pupmod-simp-dconf

The **x2go** *RPM* and its dependencies have been included on the SIMP ISO in version 6.3 and later. If you are not installing from the ISO you will need to enable the *EPEL* repo or download the RPMs manually.

To configure the **x2go** server on a system so it can be accessed remotely, add the following in the target node's *Hiera* data or corresponding workstation hostgroup:

```
x2go::client: false
x2go::server: true

# Optional settings
x2go::server::agent_options:
  '-clipboard': 'both'

simp::classes:
```

(continues on next page)

(continued from previous page)

```
  - 'x2go'
  - 'mate'
```

Note: MATE is used here for *EL* 7 systems since x2go cannot yet process GNOME 3 sessions natively with any kind of consistency.

MATE was removed from EL8 and a suitable replacement has yet to be integrated into SIMP.

For more details, see the x2go wiki

To install the client on a system, add the following in the client node's *Hiera* data:

```
x2go::client: true
x2go::server: false

simp::classes:
  - 'x2go'
```

The **x2go** client on the client node can then be used to access the server node with any user that has permission to log on via *SSH*.

The documentation for how to configure the x2go client can be found on the x2go wiki.

VNC Setup

Virtual Network Computing (VNC) can be enabled to provide remote GUI access to systems.

VNC Standard Setup

Follow the instructions in *Install Extra Puppet Modules* to install the following Puppet modules on the Puppet server:

- `pupmod-simp-vnc`

To enable remote access via VNC on the system, include `vnc::server` in Hiera for the node.

The default VNC setup that comes with SIMP can only be used over SSH and includes three default settings:

Setting Type	Setting Details
Standard	Port: 5901 Resolution: 1024x768@16
Low Resolution	Port: 5902 Resolution: 800x600@16
High Resolution	Port: 5903 Resolution: 1280x1024@16

Table: VNC Default Settings

To connect to any of these settings, SSH into the system running the VNC server and provide a tunnel to `127.0.0.1:<VNC Port>`. Refer to the SSH client's documentation for specific instructions.

To set up additional VNC port settings, refer to the code in `/etc/puppetlabs/code/environments/production/modules/vnc/manifests/server.pp` for examples.

Important: Multiple users can log on to the same system at the same time with no adverse effects; however, none of these sessions are persistent.

To maintain a persistent VNC session, use the **vncserver** application on the remote host. Type **man vncserver** to reference the manual for additional details.

VNC Through a Proxy

The section describes the process to VNC through a proxy. This setup provides the user with a persistent VNC session.

Important: In order for this setup to work, the system must have a VNC server (vserver.your.domain), a VNC client (vclnt.your.domain), and a proxy (proxy.your.domain). A vuser account must also be set up as the account being used for the VNC. The vuser is a common user that has access to the server, client, and proxy.

Modify Puppet

If definitions for the machines involved in the VNC do not already exist in Hiera, create an /etc/puppetlabs/code/environments/production/data/hosts/vserv.your.domain.yaml file. In the client host files, modify or create the entries shown in the examples below. These additional modules will allow the vserv system to act as a VNC server and the vclnt system to act as a client.

VNC Server node

```
# vserv.your.domain.yaml
simp::classes:
  - 'gnome'
  - 'mozilla::firefox'
  - 'vnc::server'
```

VNC client node

```
# vclnt.your.domain.yaml
simp::classes:
  - 'gnome'
  - 'mozilla::firefox'
  - 'vnc::client'
```

Run the Server

As vuser on vserv.your.domain, type **vncserver**.

The output should mirror the following:

> New ‘vserv.your.domain:<Port Number> (vuser)’ desktop is vserv.your.domain:<Port Number>

Starting applications specified in /home/vuser/.vnc/xstartup Log file is /home/vuser/.vnc/vserv.your.domain:*Port Number*.log

Remember the port number; it will be needed to set up an SSH tunnel!

Set up an SSH Tunnel

Set up a tunnel from the client (vclnt), through the proxy server (proxy), to the server (vserv). The table below lists the steps to set up the tunnel.

1. On the workstation, type `ssh -l vuser -L 590***<Port Number>*:localhost:590***<Port Number>***proxy.your.domain**`
   - This command takes the user to the proxy.
2. On the proxy, type `ssh -l vuser -L 590***<Port Number>*:localhost:590***<Port Number>***vserv.your.domain**`
   - This command takes the user to the VNC server.

Table: Set up SSH Tunnel Procedure

Note: The port number in 590<*Port Number*> is the same port number as previously described. For example, if the <*Port Number*> was 6, then all references below to 590<*Port Number*> become 5906.

Set up Clients

On `vclnt.your.domain`, type **`vncviewer localhost:590{Port Number}`** to open the Remote Desktop viewer.

Troubleshooting VNC Issues

If nothing appears in the terminal window, the *X Windows* may have crashed. To determine if this is the case, type **`ps -ef | grep XKeepsCrashing`**

If any matches result, stop the process associated with the command and try to restart **`vncviewer`** on `vclnt.your.domain`.

HOWTO Migrate OpenLDAP to 389 DS

The simp-utils package provides scripts to migrate from (SIMP-configured) user and group entries in *OpenLDAP* to *389 DS*.

They will be located on the SIMP server, under `/usr/share/simp/transition_scripts/openldap_to_389ds/`.

For full instructions, see the `README` in the `openldap_to_389ds` directory.

HOWTO Modify the Puppet Cron Schedule

SIMP deploys a cron-job, via `pupmod::agent::cron` from the simp/pupmod Puppet module, to run a non-daemonized puppet agent to ensure compliance, over time. By default, the cron-job is run twice every hour on a semi-random interval, to ensure all agents do not run puppet simultaneously. Additionally, the cron job **forcibly re-enables the puppet agent every 4.5 hours**.

Overriding Timing Parameters

In the example below, Puppet runs are scheduled during working hours, 0900-1700 M-F, twice every hour, in random intervals.

```
# Restrict puppet runs to execute only during working hours
pupmod::agent::cron::weekday: ['1-5']
pupmod::agent::cron::hour: ['9-17']
pupmod::agent::cron::minute: 'rand'
pupmod::agent::cron::run_timeframe: 60
pupmod::agent::cron::runs_per_timeframe: 2
```

For more information about timing parameters, refer to the `pupmod::agent::cron` class documentation.

HOWTO Upgrade SIMP

See the *Upgrading SIMP* documentation on guidance for upgrading SIMP systems.

HOWTO Use the SIMP Release RPM

Download Details

Location

The latest SIMP Community Release RPMs are always available at https://download.simp-project.com/. Install it by running:

```
# el7 Install
sudo yum install https://download.simp-project.com/simp-release-community.el7.rpm

# el8 Install
sudo yum install https://download.simp-project.com/simp-release-community.el8.rpm
```

RPM Contents and Actions

The RPM will do the following, which are detailed in sections below:

- Create two **yum** variable files to target specific SIMP releases
- Add the necessary repository files to the `/etc/yum.repos.d/` folder
- Add the necessary GPG keys needed to validate downloads to `/etc/simp/gpgkeys`

Yum Variables

The SIMP Release RPM utilizes Yum variables to determine which upstream repositories you access for SIMP files. These variables are created in two files located in the `/etc/yum/vars` folder.

YUM Variable File	Description
`simprelease`	Manages the target SIMP release version
`simpreleasetype`	Manages the SIMP release type (production, rolling, unstable)

SIMP Release Versions

The default setting for `/etc/yum/vars/simprelease` is the latest Major version of SIMP. The options for this file include:

Option (format)	Description	Example
Major	Downloads the latest Major release of SIMP, updating to all Minor, Patch, and Iteration releases that follow for that Major release, but not upgrading to the next Major release.	`6` will install the latest release of SIMP 6, upgrading everything that follows for SIMP 6, but not updating to SIMP 7.
Major.Minor	Downloads the latest Major.Minor release of SIMP, updating to all Patch and Iteration releases in the Minor release.	`6.4` will install the latest version of SIMP 6.4, 6.4, upgrading only the Patch releases, not upgrading to SIMP 6.5.
Major.Minor.Patch	Downloads the latest Major.Minor.Patch release of SIMP, only updating if an iteration release is available.	`6.4.0` will install SIMP 6.4.0, and update to any iteration updates, but not update to SIMP 6.4.1.
Major.Minor.Patch-Iteration	Downloads the specific version of SIMP, never updating to a newer version.	`6.4.0-0` will install SIMP 6.4.0-0 and never update.

SIMP Release Types

The `/etc/yum/vars/simpreleasetype` variable file controls the type of releases you want to download.

Option (exact contents of the file)	Description
releases	This setting will have the repos grab only fully tested SIMP releases.
rolling	This setting will have the repos grab updates to RPMs that have not yet made it into a SIMP Release, but have been tested and released individually with confidence. You can specify a Major or Major.Minor release for rolling updates.
unstable/simp6	This setting will have the repos grab all updates to RPMs in the unstable repo (This is extremely dangerous and not recommended for production environments). `unstable/simp7` will become available in the future when SIMP 7 is released.

These variables allow you to control the exact updates you receive for SIMP, and provide a dynamic system that won't need to be updated or re-installed for future versions of SIMP.

> **Warning:** Setting these Yum Var files to invalid contents will break the repo files and prevent successful downloads. The `simprelease` file should only include numbers, dots, and dashes, no words or other characters. The `simpreleasetype` file should only include words and potentially a slash for the unstable repos.

Avoid any quotes and other characters that would potentially break the repo URLs

Repository Files

The RPM will add the SIMP Community repo, as well as other necessary SIMP repo files, such as Postresql, Puppet, and EPEL. These files point to the same Release folder specified by the `/etc/yum/var/` files, but access the vendor specific repositories maintained there.

SIMP Enterprise Release RPM

The latest SIMP Enterprise Release RPMs are always available at https://download.simp-project.com/. You can install it as `root` by running

```
# el7 Install
yum install https://download.simp-project.com/simp-release-enterprise.el7.rpm

# el8 Install
yum install https://download.simp-project.com/simp-release-enterprise.el8.rpm
```

The Enterprise Release RPM includes the same files as the Community version, with added repo files for SIMP Enterprise, SIMP Console, and SIMP Scanner.

GPG Keys

The SIMP Release RPM will also add necessary GPG Keys to the `/etc/simp/gpgkeys` folder. These GPG Keys are placed in this folder to prevent the `simp-gpgkeys` package from conflicting with them, since the SIMP and Puppet keys are required to download the package via **yum**.

HOWTO Work with the SIMP Rsync Shares

When we added support for multiple environments, the SIMP rsync space in `/var/simp/environments/<environment>/rsync` became quite complex.

This will guide you through the new rsync layout as well as providing guidance on setting up new rsync shares for your various components.

This is very SIMP-specific and does not preclude you from using rsync however you like. However, if you want multi-environment support, you will need to replicate something like what we have done for your custom directories.

Warning: All symlinks in the **rsync** space must be relative links. Do **not** use absolute paths as this may result in misconfiguration of client nodes.

Why SIMP Uses Rsync

Rsync support was introduced in SIMP in the early days due to the fact that the Puppet native file synchronization mechanisms were relatively horrible at syncing large files (too much in memory) and large numbers of files in a directory tree (too many resources and system load).

Rsync neatly solves both of these issues and is present on all SIMP systems by default.

By default, SIMP wraps all **rsync** connections in a *Stunnel* connection to provide encrypted connections. Additionally, SIMP adds randomly generated passwords to sensitive shares to prevent unauthorized connections.

You can restrict this as far as necessary in your environment but the defaults should suit most needs.

Standard Capabilities

The standard SIMP rsync shares for the `production` *Puppet environment* exist at `/var/simp/environments/production/rsync`. This is an assumed path and changing this path will break some aspects of the system.

Within this directory, you will find a set of files with the name `.shares`. This file is used by the fact `simp_rsync_environments` to indicate that all directories at the given location should be added as rsync shared directories.

The data structure in the `simp_rsync_environments` is based on the **lower cased** name of the containing directory. This means that there should **NEVER** be two directories with the same name at the same level of the directory hierarchy. In this case, the last one present alphabetically will win.

As a concrete example, given the following directory structure:

```
var
└── simp
    └── environments
        └── production
            └── rsync
                ├── Global
                │   ├── .shares
                │   └── clamav
                │       ├── main.cvd
                │       ├── daily.cld
                │       └── bytecode.cld
                ├── .rsync.facl
                ├── README
                └── RedHat
                    ├── Global
                    │   ├── freeradius
                    │   ├── .shares
                    │   ├── tftpboot
                    │   │   └── linux-install
                    │   │       └── README
                    │   ├── dhcpd
                    │   │   ├── dhcpd.conf
                    │   │   └── LICENSE
                    │   ├── snmp
                    │   │   ├── mibs
                    │   │   └── dlmod
                    │   └── apache
                    │
```

(continues on next page)

(continued from previous page)

```
                    │       └── www
                    │           ├── cgi-bin
                    │           ├── error
                    │           │   └── include
                    │           ├── icons
                    │           │   └── small
                    │           └── html
                    ├── 6
                    │   ├── bind_dns
                    │   │   └── LICENSE
                    │   └── .shares
                    └── 7
                        ├── bind_dns
                        │   └── LICENSE
                        └── .shares
```

The following would be returned by the `simp_rsync_environments` fact:

```
{
  "production": {
    "id": "production",
    "rsync": {
      "id": "rsync",
      "global": {
        "id": "Global",
        "shares": [
          "clamav"
        ]
      },
      "redhat": {
        "id": "RedHat",
        "6": {
          "id": "6",
          "shares": [
            "bind_dns"
          ]
        },
        "7": {
          "id": "7",
          "shares": [
            "bind_dns"
          ]
        },
        "global": {
          "id": "Global",
          "shares": [
            "freeradius",
            "tftpboot",
            "dhcpd",
            "snmp",
            "apache"
          ]
```

(continues on next page)

(continued from previous page)

```
        }
      }
    }
  }
}
```

Breaking this down, the following data is shown:

```
{
  "downcased_directory_name": {
    "id": "Original_Directory_Name",
    "downcased_subdirectory_name": {
      "id": "Original_Subdirectory_Name",
      "shares": [
        "Directory One",
        "directory two"
      ]
    }
  }
}
```

Note: The presence of the `.shares` file in the directory tree tells the `simp_rsync_environments` fact that all directories at that level are to be exposed as shares in the returned data structure.

That said, it is up to your Puppet logic to actually expose them as such!

See the `simp::server::rsync_shares` class to see how we do this for the default **`rsync`** shares.

Supporting Additional Environments

Generally, in a SIMP environment, you are going to want to start with the directory structure that we have and simply copy the entire data structure to a directory with your custom name.

Warning: Be sure not to copy any sensitive information into the space!

For example, if you wanted to create the standard dev/test/prod structure, and the `production` environment already existed:

```
cd /var/simp/environments
cp -a production dev
cp -a production test
```

Alternatively, you can use **`simp environment new`** to affect the copy of all or some the *SIMP Omni-Environment*. (You can also use that command to affect links of the *SIMP Secondary Environment* or *SIMP Writable Environment*, which in some circumstances may be more appropriate.)

After this, you will now have an enhanced `simp_rsync_environments` data structure that holds all of the information for the `dev`, `test`, and `production` environments.

You can then manipulate the contents of the different environments to suit your needs.

Note: The contents of the various rsync directories are not under version control by default. While you may add them to a VCS of your choosing (SVN, Git, etc. . .), there may be some VERY large files present in these directories.

Make sure your system can handle the load before adding rsync content into a VCS!

Disabling Stunnel

If you decide to disable **stunnel**, you will need to specify your rsync server in *Hiera*, if it is not already specified.

Warning: If you disable stunnel, your data and any rsync access credentials will be passed in the clear!

```
---
simp_options::rsync::server: <rsync_server_fqdn>
```

Additionally, you will need to ensure your firewall is open on the rsync port. Include the following on the node acting as the rsync server.

```
class site::rsync_iptables (
  Simplib::Netlist $allow      = simplib::lookup('simp_options::trusted_nets'),
  Simplib::Port    $rsync_port = 873
){
  iptables::listen::tcp_stateful { "rsync_shares":
    trusted_nets => $allow,
    dports       => $rsync_port
  }
}
```

4.5.3 Puppet Guides

Guides in this section relate to the configuration and use of Puppet.

HOWTO Block or Cache Facts

As described in Configuring Facter with facter.conf, you can block or cache groups of facts on a node using settings in `facter.conf` on that node. SIMP has made this easier by adding management of `facter.conf` to simp/pupmod (version >= '8.0.0'). Using this module, you can manage `facter.conf` and specify its configuration in *Hiera*.

Here is example Hiera that blocks and caches select groups of facts:

```
---
pupmod::manage_facter_conf: true
pupmod::facter_options:
  facts:
    blocklist:
      - EC2
    ttls:
      - kernel: "8 hours"
      - memory: "8 hours"
```

(continues on next page)

(continued from previous page)

```
      - operating system: "8 hours"
      - processor: "8 hours"
```

HOWTO Move a Client to a new Puppet Server

The following provides details on how to move an client to a new *Puppet Server*.

Note: All commands in this section should be run as the `root` user.

On the Old Puppet Server

The following procedures will archive the agent's artifacts from all environments, copy them to the new Puppet Server, and clean out the agent's *Hiera* data.

Archive the agent's artifacts from all environments

1. Archive the agent's artifacts from all SIMP *Secondary Environments*:

```
find /var/simp/environments -name "*<agent-fqdn>*" -exec tar --selinux --xattrs -rpvf <agent-fqdn>_transfer.tar {} \;
```

2. Archive the agent's data from all SIMP *Writable Environments*:

```
find `puppet config --section server print vardir`/simp -name "*<agent-fqdn>*" -exec tar --selinux --xattrs -rpvf <agent-fqdn>_transfer.tar {} \;
```

3. Archive the agent's Hiera data from all *Puppet Environments*:

Warning: If you deploy your agents' Hiera data from a *Control Repository* on the new Puppet server, ensure the agent's Hiera data is in the relevant branches and **skip this step.**

```
find /etc/puppetlabs/code/environments/*/{data,hieradata} -name "<agent-hostname>.yaml" -exec tar --selinux --xattrs -rpvf <agent-hostname>_transfer.tar {} \;
find /etc/puppetlabs/code/environments/*/{data,hieradata} -name "<agent-fqdn>.yaml" -exec tar --selinux --xattrs -rpvf <agent-hostname>_transfer.tar {} \;
```

4. Copy `<agent-hostname>_transfer.tar` to the new Puppet server.

Remove agent-specific Hiera data from all environments

Warning: **Skip this section** if you deploy your agents' Hiera data from a *Control Repository*

1. Remove agent-specific Hiera data from all environments

   ```
   find /etc/puppetlabs/code/environments -name "<agent-fqdn>.yaml" --delete
   ```

   Note: You may have Hiera YAML files with the short name of the host still in place, but those are too dangerous to automatically delete since they may match multiple hosts.

2. Reload the **puppetserver** process after removing the agent's Hiera data:

   ```
   puppetserver reload
   ```

On the New Puppet Server

Warning: This assumes that the new Puppet server is set up identically to the old Puppet server. If it isn't, you will need to verify that the artifacts in the `tar` file are correctly placed.

1. Unpack the `<agent-hostname>_transfer.tar` archive onto the system:

   ```
   tar --selinux --xattrs -C / -xvf <agent-hostname>_transfer.tar
   ```

2. Reload the **puppetserver** process:

   ```
   puppetserver reload
   ```

On The Agent

Important: Make sure you are running these commands **on the agent**. If you run them on the server, there is a **very high risk** they will make your Puppet infrastructure inoperable.

Remove the Agent Puppet Certificates

To remove all legacy SSL files, run:

```
rm -rf $(puppet config print --section agent ssldir)
```

Update the Puppet Config

Update /etc/puppetlabs/puppet/puppet.conf with the following changes:

```
server = new.puppet.server.fqdn
ca_server = new.puppet.server.fqdn
ca_port = 8141
```

Run Puppet

Assuming the new Puppet server has been set up to properly accept the agent, execute a full Puppet run using **puppet agent --test**.

On the new puppet server you will need to sign off the certificate for the new client using **puppetserver ca cert sign <new client name**.

If everything was done properly, the agent will now be synchronized with the new Puppet server.

If you find issues, refer to the *Setting up the Client* and *Troubleshooting Puppet Issues* sections of the documentation, and ensure that the new Puppet CA is set up properly to trust the Puppet agent.

HOWTO Set up a SIMP Environment in a Control Repository

This HOWTO describes how to create Puppet *control repositories* for use with a *Control Repository deployment scenario*.

Contents

Requirements

To use any of the procedures in this section, you must:

1. Have access to a remotely-hosted *Git* repository where you will host your control repository.
2. Have a basic understanding of:
   - Puppet *Control Repositories*.
   - How to use the git command.
   - The topics covered in "*Deploying SIMP Environments*," particularly:
     - The composition of *SIMP Environments*
     - The *Local Deployment Scenario*
     - The simp commands needed to manage a *SIMP Omni-Environment*

You may find it helpful to read the section that explains how a control repository works in Puppet, Inc.'s control repository documentation .

Creating a Control Repository on a SIMP Server

This procedure creates an r10k control repository from a fresh SIMP ISO/RPM installation. It is currently limited to *just* the control repository—the RPM-provided Puppet module git repositories will remain on the SIMP server's local filesystem.

Important: This procedure does NOT describe how to migrate or host SIMP Puppet modules in remote git Repositories, or how to update the Puppetfile.simp file to deploy them.

Prerequisites

- You have installed SIMP (per the installation guide) from *ISO* or *RPM Repository*.
- This procedure needs to be done on the SIMP server.

Procedure

1. Create a git repository inside the `production` Puppet environment directory:

```
cd /etc/puppetlabs/code/environments/production
git init .
```

2. Create a new branch for the `production` Puppet environment:

```
git checkout -b production
```

3. Add files to the git repository. (Do **not** add the `modules/` directory or `.resource_types/` directory):

```
# Add the files
git add Puppetfile Puppetfile.simp hiera.yaml environment.conf

# Add directories
git add manifests/ data/
```

4. Commit the changes

```
git commit -m "Initial production environment"
```

5. Push the branch to your control repository:

```
# Add a remote for your control repository
git remote add control_repo <URL to the control repo>

# Push the branch
git push production control_repo
```

Creating a Control Repository without SIMP installed

This procedure creates a control repository with a branch for an environment named called `dev1`. The Puppet modules will be deployed from the SIMP project's public git repositories over the internet.

Important: If you intend to use this environment *to bootstrap a SIMP server without RPMs*, it **must** be named `production` (and not `dev1`).

Prerequisites

You will need the SIMP Puppet environment "skeleton" directory, which can be obtained from one of the following sources:

- `/usr/share/simp/environment-skeleton/puppet/` when the RPM package **simp-environment-skeleton** is installed[1].
- `environments/puppet/` under a checkout of the git repository https://github.com/simp/simp-environment-skeleton.

Procedure

1. Create an empty git repository:

```
mkdir $HOME/control-repo
cd $HOME/control-repo
git init .
```

2. Copy the puppet environment skeleton into your git repository:

```
cd $HOME/control-repo
cp -R /usr/share/simp/environment-skeleton/puppet/* .
```

3. Substitute your environment's name into `environment.conf`:

```
sed -e "s/%%SKELETON_ENVIRONMENT%%/dev1/g" ./environment.conf.TEMPLATE > ./
↪environment.conf
chmod 640 environment.conf
rm environment.conf.TEMPLATE
```

4. Download and edit the `Puppetfile.simp` file:

   a. Download the `Puppetfile` used to create a SIMP ISO for a specific release from the SIMP simp-core repository (in this example, it is `6.4.0-0`):

```
cd /etc/puppetlabs/code/environments/dev1
curl -o Puppetfile.simp https://github.com/simp/simp-core/blob/6.4.0-0/
↪Puppetfile.pinned
```

   b. Edit `Puppetfile.simp` to remove components that are not Puppet modules, deleting all lines up to and including `moduledir 'src/puppet/modules'`. You can do this from the command line by running:

[1] If your working host doesn't have the **simp-environment-skeleton** RPM and you'd like to install it, you can set up the SIMP yum repositories (see "*Installing SIMP From A Repository*)."

```
sed -i -e "0,/^moduledir 'src\/puppet\/modules'/d" Puppetfile.simp
```

   c. (Optionally,) edit Puppetfile.simp to remove any non-core SIMP modules (e.g., the ones packaged with simp-extras) that you don't need. You can discover the list of the SIMP extra modules by examining the dependencies of the simp-extras RPM:

```
yum deplist simp-extras | grep dependency:
```

5. Create the Puppetfile:

   a. Create the file $HOME/control-repo/Puppetfile, which should include the following line:

```
instance_eval(File.read(File.join(__dir__,"Puppetfile.simp")))
```

   b. (Optionally,) also add entries for any non-SIMP modules your site requires.

6. Add/adjust any of the *Hiera* files in the data/ directory.

7. Add all the files to a branch named for the environment in this repository:

```
# create the branch
git checkout -b dev1

# add the directory tree
git add --all

# verify the directory tree doesn't have any temporary files you created
git status

git commit -m 'Initial dev1 environment'
```

8. Push the branch to your control repository:

```
# Add a remote for your control repository
git remote add control_repo <URL to the control repo>

# Push the branch
git push dev1 control_repo
```

Advanced Topics

Bootstrapping A SIMP Server without SIMP Module RPMs

A full set of SIMP module RPMs is not required in order for the SIMP server to be initially configured. With a slight change to the procedures listed in *Initial SIMP Server Configuration*, a SIMP server can be bootstrapped with a production SIMP Omni-Environment skeleton, such as one created in this HOWTO.

Note: You may want to read through *Initial SIMP Server Configuration* before proceeding. It provides additional information that will not be repeated here.

In these procedures, we assume that you have created a production SIMP Omni-Environment skeleton that contains a Puppetfile with URLs to the core SIMP Puppet modules. For example, you followed the procedures to create a control repository for a production environment using internet module repositories.

Execute the following steps as `root`:

1. Deploy the modules in the `production` Puppet environment using **`r10K`** or `Code Manager`. Be sure the deployed modules are accessible to the `puppet` group.
2. Install the `puppetserver` package:

   ```
   yum install puppetserver
   ```

3. Run **`simp config`** with an option that tells it the SIMP Omni-Environment has already been created:

   ```
   simp config --force-config
   ```

4. Run **`simp bootstrap`**:

   ```
   simp bootstrap
   ```

5. After **`simp bootstrap`** completes, add the following generated Hiera files in the `production` Puppet environment to the `production` branch in your control repository:
   - `production/data/simp_config_settings.yaml`
   - `production/data/hosts/<SIMP server FQDN>.yaml`

To continue configuring the system, move on *Client Management* section in the *SIMP User Guide*.

HOWTO Disconnect PuppetDB from the Puppet Server

This section covers how to disconnect *PuppetDB* from the *Puppet Server* for troubleshooting, or to allow SIMP to repair the configuration.

Run the following script on the *Puppet Server* to stop **`puppetdb`** and restart the **`puppetserver`** process without the connection.

```
puppet resource service puppetdb ensure=stopped

# The following line assumes the puppet server configuration directory is
# /etc/puppetlabs/puppet (the default).
mv /etc/puppetlabs/puppet/routes.yaml /etc/puppetlabs/puppet/routes.yaml.backup

puppet config set --section server storeconfigs false
puppet config set --section server reports store
puppet config set --section main storeconfigs false

puppet resource service puppetserver ensure=stopped
puppet resource service puppetserver ensure=running
```

Note: When **`puppet`** is next run on the Puppet server, it will reconfigure the **`puppetserver`** and **`puppetdb`** to reconnect and restart the appropriate services. If you are debugging an issue, you may want to run **`puppet agent --disable`** to prevent Puppet from resetting the system while you are debugging.

HOWTO Set Up SIMP's External Node Classifier

An *External Node Classifier*, ENC, can be used to determine what Puppet environment is used by a node.

SIMP provides a simple, YAML-based ENC, `/usr/local/bin/set_environment`, available from the `simp-utils` RPM. That package will already be installed on your system, if you installed the SIMP server from a SIMP ISO or from RPM.

To use this script for your ENC, do the following as `root`:

1. Ensure the script can be executed by the **`puppetserver`**.

   ```
   chmod g+rX /usr/local/bin/set_environment
   chgrp puppet /usr/local/bin/set_environment
   ```

2. Configure Puppet to use this script as an ENC

   Set the following in the `[server]` section of `/etc/puppetlabs/puppet/puppet.conf`:

   a. Add or change the line `node_terminus` to `exec`.

   b. Set the `external_nodes` entry to `/usr/local/bin/set_environment`.

   The resulting lines should look something like this:

   ```
   ...
   [server]
   ...
   node_terminus = exec
   external_nodes = /usr/local/bin/set_environment
   ...
   ```

3. Add a file `/etc/puppetlabs/puppet/environments.yaml` to your system.

   This file defines the mapping of nodes to environments.

   - Each rule in this file should have the form

     `<regular expression or FQDN>:  'environment name'`.

   - The rules are processed from top down, and the first match wins.

   For example,

   ```
   # The puppet server will use the production environment
   'puppet.my.domain':            'production'

   # Any node in my.domain whose FQDN begins with test will use the test environment
   '/^test([0-9])+\.my\.domain/': 'test'

   # Default to the production environment in all other cases
   '/^.*$/':                      'production'
   ```

   To verify the file is properly formatted and yields the classification desired run the script manually with a node's FQDN. It should return

   ```
   environment:  <node's Puppet environment name>
   ```

   Using the above `environments.yaml` file, the following command should return

   ```
   environment:  test
   ```

```
/usr/local/bin/set_environment test11.my.domain
```

4. Ensure the **puppetserver** can access the ENC's configuration file

```
chmod g+rX /etc/puppetlabs/puppet/environments.yaml
chgrp puppet /etc/puppetlabs/puppet/environments.yaml
```

5. Restart the **puppetserver** service

```
# On EL7
systemctl restart puppetserver
```

For more information on ENCs, please see Puppet's ENC documentation.

HOWTO Enable Client Reporting

Puppet has the ability to send run status reports back to the server at the conclusion of each client run.

SIMP natively supports the *store* (default) and *puppetdb* report storage endpoints and can be configured as shown below.

Enable Client Reporting

Set the following in *Hiera* to enable client reporting:

```
---
pupmod::report: true
```

Once puppet applies this setting, clients will start sending reports to the server at the conclusion of each puppet run.

Filesystem Reports

By default, the puppet sever will enable the *store* reports target which will store the client reports on the local filesystem.

To view the raw reports, you can navigate to the directory output by `puppet config print reportdir`.

If the system has been set up for *puppetdb* reports and you need to change it back to *store*, set the following in *Hiera*:

```
---
puppetdb::master::config::manage_report_processor: true
puppetdb::master::config::enable_reports: false
```

PuppetDB Reports

Prior to proceeding, you should read *HOWTO Enable PuppetDB* if you have not already enabled **puppetdb**.

If you want to use a *GUI* application like Puppetboard to connect to **puppetdb** for reports, then you will need to ensure that the reports are being sent to **puppetdb** by the server.

To do this, set the following in *Hiera*:

```
---
puppetdb::master::config::manage_report_processor: true
puppetdb::master::config::enable_reports: true
```

CLI Reporting

You can use the PuppetDB API to get node reporting using **curl** or any other application that connect to a web endpoint and process *JSON* output.

SIMP provides a tool called **puppetlast** that can read from both the PuppetDB API as well as the locally stored *YAML* reports for a simple view of your environment.

The **puppetlast** command is provided by the simp-utils RPM.

HOWTO Enable PuppetDB

Important: Do not attempt to manage PuppetDB on Puppet Enterprise hosts!

As of SIMP 6.6.0, *puppetdb* is no longer enabled or managed by default.

Use the following YAML as a guide to the *hiera* settings required to enable the management of **puppetdb**.

```
---
# Include the puppetdb class
simp::server::classes:
  - simp::puppetdb

## SIMP Required Settings

# Let pupmod::master::base handle this.
puppetdb::master::config::restart_puppet: false

# Set the PuppetDB port
puppetdb::master::config::puppetdb_port: 8139

## Optional Settings

# The version of PuppetDB that should be installed
puppetdb::globals::version: 'latest'

# Set up node report storage
puppetdb::master::config::manage_report_processor: true
puppetdb::master::config::enable_reports: true
```

See the puppetlabs/puppetdb documentation for details on all of the settings that affect the configuration of **puppetdb**.

HOWTO Set up and Utilize hiera-eyaml

This section steps through configuring a *Puppet environment*'s hierarchy to use the "*eyaml*" Hiera backend, which enables you to keep encrypted data in otherwise plaintext *Hiera* files. The examples use `pkcs7`, which is the default and built-in encryption mechanism. Additional encryption mechanisms (like *GPG*) are available as plugins, distributed as RubyGems.

For additional information on configuring and using eyaml, see the Puppet documentation on configuring hiera-eyaml and VoxPupuli's hiera-eyaml documentation.

Installing the hiera-eyaml gem

A user must have the hiera-eyaml gem installed in their Ruby environment in order to manage secrets with the **eyaml** command.

```
$ gem install hiera-eyaml
```

If you are unable to install the gem from your system, you may be able to use the version that ships with the Puppet Agent at `/opt/puppetlabs/puppet/bin/eyaml`. You may need privileged access to run it, depending on the puppet-agent package and the system's umask when it was installed.

Configuring the Hiera hierarchy to use eyaml

At the top level of your *Puppet Environment*, edit the `hiera.yaml` file and add hierarchy tiers using the `eyaml` backend.

The following is an example of an eyaml tier in the Hiera hierarchy. Starting with SIMP 6.6.0, the simp-environment-skeleton RPM has been updated to provide eyaml entries like this one.

```
---
defaults:
  - datadir: data

hierarchy:
  - name: Eyaml Data
    lookup_key: eyaml_lookup_key
    paths:
    - "hosts/%{trusted.certname}.eyaml"
    - "hosts/%{facts.fqdn}.eyaml"
    - "hosts/%{facts.hostname}.eyaml"
    options:
      pkcs7_private_key: /var/lib/puppet/keys/private_key.pkcs7.pem
      pkcs7_public_key: /var/lib/puppet/keys/public_key.pkcs7.pem
...
```

Note the `lookup_key` and `options` keys, and that the data files have an `.eyaml` extension.

Note also that each hierarchy tier defines its own `options`, so in more advanced situations, you can configure Hiera to decrypt data from multiple key pairs using duplicate tiers with different keys.

Generating the keys

Use the **eyaml** command to generate the keys specified in each hierarchy tier's `options:`. Make sure the permissions on the keys are set securely, but that the Puppet Server has access to them.

```
# mkdir -p /var/lib/puppet/keys/
# /opt/puppetlabs/puppet/bin/eyaml createkeys \
   --pkcs7-private-key=/var/lib/puppet/keys/private_key.pkcs7.pem \
   --pkcs7-public-key=/var/lib/puppet/keys/public_key.pkcs7.pem
# cd /var/lib/
# chown -R puppet puppet
# chmod 500 puppet
# chmod 400 puppet/*.pem
```

Make sure the keys are backed up.

Encrypting Hiera values

Once the environment's hierarchy is set up, data can be encrypted and stored as values in the `.eyaml` files. The following is an example of how to encrypt the passwords used by the puppet module simp/simp_snmpd.

The simp_snmpd module accepts a Hash containing several credentials in the `simp_snmpd::v3_users_hash` parameter, which is used to configure the net-snmp users.

Unencrypted, this sensitive data would be exposed as plaintext in a file like `/etc/puppetlabs/code/environments/production/data/common.yaml`, looking something like this:

```
simp_snmpd::v3_users_hash:
  snmp_ro:
    authtype: 'SHA'
    privtype: 'AES'
    authpass: 'MyAuthPassw0rd'
    privpass: 'MyPrivPassw0rd'
  snmp_rw:
    authtype: 'SHA'
    privtype: 'AES'
    authpass: 'MyOtherAuthPassw0rd'
    privpass: 'MyOtherPrivPassw0rd'
```

This exposes the passwords to everyone with access to the production environment's files. If you deploy your environment from a *control repository*, they are also exposed to everyone with read access to the repository. To avoid this, we will safeguard the sensitive data by encrypting it.

To encrypt the passwords, you need access to the file defined as the eyaml backend's `pkcs7_public_key`, which can be safely distributed to users.

Use the **eyaml encrypt** command to use the public key and encrypt each password string:

```
# eyaml encrypt --pkcs7-public-key=/var/lib/puppet/keys/public_key.pkcs7.pem  -s
↪'MyAuthPassw0rd'
# eyaml encrypt --pkcs7-public-key=/var/lib/puppet/keys/public_key.pkcs7.pem  -s
↪'MyPrivPassw0rd'
# eyaml encrypt --pkcs7-public-key=/var/lib/puppet/keys/public_key.pkcs7.pem  -s
↪'MyOtherAuthPassw0rd'
```

(continues on next page)

(continued from previous page)

```
# eyaml encrypt --pkcs7-public-key=/var/lib/puppet/keys/public_key.pkcs7.pem  -s
↪'MyOtherPrivPassw0rd'
```

The output for each of the commands will look something like the following:

```
string: ENC[PKCS7,
↪MIIBeQYJKoZIhvcNAQcDoIIBajCCAWYCAQAxggEhMIIBHQIBADAFMAACAQEwDQYJKoZIhvcNAQEBBQAEggEAKNBCkXENUf6C0diKc
↪leJIYouylfPod9G/M1SC/Lw64fhzcC7dSOru+vJan3zT1Jp/
↪7nmsen263VBihOshbtkHKLSoJ7n96MlFqF0CrzOzxoz/p3y2591FoSXqjljCGG0PmV9FGONe1n5vUwWuy/
↪+YQlciZEtyjyUBCZyJgaWfFh6//
↪6vJT4G+5i0Ui1xzAtvYaDKW968Yx3ldQYy7btiRYct4Xvh6giFWDLXIE5Mnfe4fH6NwwXHDA8BgkqhkiG9w0BBwEwHQYJYIZIAWUD
```

or:

```
block: >
  ENC[PKCS7,MIIBeQYJKoZIhvcNAQcDoIIBajCCAWYCAQAxggEhMIIBHQIBAD
  AFMAACAQEwDQYJKoZIhvcNAQEBBQAEggEAKNBCkXENUf6C0diKcV1VPvB4r8
  q+AFzu9E4VsR9Ch50q0UJ5sO977VXWLkX1oYbEvqPZZrmH122gvrYp1xux+W
  +UuFZbCzMQ7AMNe8eiJ7FvYYs79/leJIYouylfPod9G/M1SC/Lw64fhzcC7d
  SOru+vJan3zT1Jp/7nmsen263VBihOshbtkHKLSoJ7n96MlFqF0CrzOzxoz/
  p3y2591FoSXqjljCGG0PmV9FGONe1n5vUwWuy/+YQlciZEtyjyUBCZyJgaWf
  Fh6//6vJT4G+5i0Ui1xzAtvYaDKW968Yx3ldQYy7btiRYct4Xvh6giFWDLXI
  E5Mnfe4fH6NwwXHDA8BgkqhkiG9w0BBwEwHQYJYIZIAWUDBAEqBBAXOTJRuX
  WXBSfxIlA9HqWfgBBhi06bLLsVsjQ2leNYg2N5]
```

Edit the `/etc/puppetlabs/code/environments/production/data/common.eyaml` and include the `simp_snmpd::v3_users_hash` with the passwords replaced by the encrypted values provided by the eyaml command.

```
---
simp_snmpd::v3_users_hash:
  snmp_ro:
    authtype: 'SHA'
    privtype: 'AES'
    authpass: ENC[PKCS7,
↪MIIBeQYJKoZIhvcNAQcDoIIBajCCAWYCAQAxggEhMIIBHQIBADAFMAACAQEwDQYJKoZIhvcNAQEBBQAEggEAKNBCkXENUf6C0diKc
↪leJIYouylfPod9G/M1SC/Lw64fhzcC7dSOru+vJan3zT1Jp/
↪7nmsen263VBihOshbtkHKLSoJ7n96MlFqF0CrzOzxoz/p3y2591FoSXqjljCGG0PmV9FGONe1n5vUwWuy/
↪+YQlciZEtyjyUBCZyJgaWfFh6//
↪6vJT4G+5i0Ui1xzAtvYaDKW968Yx3ldQYy7btiRYct4Xvh6giFWDLXIE5Mnfe4fH6NwwXHDA8BgkqhkiG9w0BBwEwHQYJYIZIAWUD
    privpass: ENC[PKCS7,
↪MIIBeQYJKoZIhvcNAQcDoIIBajCCAWYCAQAxggEhMIIBHQIBADAFMAACAQEwDQYJKoZIhvcNAQEBBQAEggEASZjHjGiEAaseJ3tOY
↪LxnJqrs/FeuhSrCSBkU9G2IsEcMD5k1KWvS5k81RwNO5DtXLnoEarxrAXkej6n4KAj1fU8o+6/
↪Fe9gBP2zUtJAv+xpZp22AzOlUvjRMZo3aXNUuTLYhKrFJ6bcZHotkVM4gFcGX+AEoN6e4rQfBK9AK1ZxRjJWT675/
↪oNW7Pa5naeucCKsFtFfWakLk5Hk7jZDqzlqlu3wpHLeRuWDqOxkM3i7WdNVKGlES3JMjTYIYNao21nACB4rj1FdqReXVsfiMYLkzA
  snmp_rw:
    authtype: 'SHA'
    privtype: 'AES'
    authpass: ENC[PKCS7,
↪MIIBiQYJKoZIhvcNAQcDoIIBejCCAXYCAQAxggEhMIIBHQIBADAFMAACAQEwDQYJKoZIhvcNAQEBBQAEggEAjxzvezqyGrgycNf26
↪8zvn+gx3rz8bHc+vVH3Fnej9Zj3FMoRbaYJAs2iNqii7ew398rumWy3TiqeV8isiAft7HnH1r/
↪zXDTfndgTsxyd0guu5ZFM7ecqVyS8Sqkc7nHYeR5u8o1NWTBMBgkqhkiG9w0BBwEwHQYJYIZIAWUDBAEqBBDFz6fVKaPmMshbKvs7
    privpass: ENC[PKCS7,
```

(continues on next page)

(continued from previous page)

```
↪MIIBiQYJKoZIhvcNAQcDoIIBejCCAXYCAQAxggEhMIIBHQIBADAFMAACAQEwDQYJKoZIhvcNAQEBBQAEggEAPROZvDIFre7M3+Bs2
↪hAl9J/zg13znafT+eLHsa6ds7YqlM208VVlxYWfl/
↪zhWEW8U3KYhzlHRo9TIXw5w5yAtpYVknF0UL5+MFhCrHKBES92PPq4hS+X0E/
↪o0Mk1zDu24ZgvT8+BRVH+7GmvLPQ+rrT89ou3ovy/
↪PRTu6jf2ppX9M1NFJAxB+bskEA9PMzgPshEGs85ns25mNknFrKG8R8YxejVm0l6JD5DTzWCEghnGkP799Kssem5PC8cD7BvaDJdmB
```

Save the file and make sure the Puppet Server has access to read it.

```
# chown root:puppet /etc/puppetlabs/code/environments/production/data/common.eyaml
# chmod 0640 /etc/puppetlabs/code/environments/production/data/common.eyaml
```

Remove the unencrypted `simp_snmpd::v3_users_hash` key from `/etc/puppetlabs/code/environments/production/data/common.yaml` so the passwords are no longer visible.

Run **`puppet agent -t`** on an agent node in the `production` environment where simp_snmpd is classified. The net-snmp users' credentials should be configured using the decrypted values.

Using `eyaml edit`

If a user editing the `.eyaml` file has access to both the private and public keys, they can use **`eyaml edit file.eyaml`** as a convenient alternative to the **`eyaml encrypt`** example in the previous section.

`eyaml edit` will automatically decrypt the file and bring up an editor to edit the values in plaintext. After exiting the editor, any edited values will be automatically re-encrypted in place.

It is also possible to encrypt blocks of data and entire files. See the hiera-eyaml documentation for more details on these and other features.

HOWTO Migrate to a New Puppet Server

Creating a new server and migrating clients

The recommended method for upgrading **major** breaking changes (`X` bump) is to create a new Puppet Server and migrate your data and clients to it. This process follows the path of least destruction; we will guide you through how to back up the existing Puppet server, create a new server, and transfer your clients.

1. Set up a new Puppet server that will house your new SIMP environment.

   Note: You must ensure that this node can be reached by any client that is to be migrated. The new system will not interfere with your existing Puppet system unless you specifically configure it to do so.

   Important: Do **NOT** destroy your old Puppet server until everything has been successfully migrated and is in production under the new server.

2. Consider vital services other than Puppet that are housed on your current Puppet server node (e.g., DNS, DHCP, LDAP, custom kickstart, YUM, NFS, etc.). You may choose to keep many of these services running on your old Puppet server node. Anything not preserved must be migrated to a new system.

Back Up the Existing Puppet Server

Prior to any modifications to your infrastructure, we **highly** recommend following *HOWTO Back up the Puppet Server*.

Create a New Server

Obtain an official SIMP ISO or point your server at the latest YUM Repositories: and follow the *ISO Installation (Preferred)* or *Installing SIMP From A Repository* as appropriate.

Follow the *Client Management* guide, and set up services as needed. Remember, you can opt-out of any core services (DNS, DHCP, etc.) you want your clients or old Puppet server to run! If you want the new Puppet server to run services the existing Puppet server ran, you may be able to use the backup of the **rsync** directories from the old system.

> **Warning:** Do not blindly drop **rsync** (or other) materials from the old Puppet server onto the new one. The required structures for these components may have changed.

When you *Apply Certificates* you may wish to transfer client certs to the new server. If you are using the FakeCA and still wish to preserve the certificates, follow the *Installing Official Certificates* guidance, and treat the existing Puppet server as your 'proper CA'.

Promote the New Puppet Server and Transfer Your Clients

Follow the *HOWTO Move a Client to a new Puppet Server* guide to begin integration of your new Puppet server into the existing environment.

> **Note:** You should *always* start migration with a small number of **least critical** clients!

Retire the Old Puppet Server

Once you have transferred the management of all your clients over to the new Puppet server, you may safely retire the old Puppet server.

HOWTO Back up the Puppet Server

This section details the steps required to back up the *Puppet Server*.

> **Note:** A default SIMP installation can use *Git* as a rudimentary method to back up the Puppet Server. If a different method is preferred, the user must install and configure it first.

1. Backup `/etc/puppetlabs/puppet/ssl`
2. Backup `/etc/puppetlabs/puppet`
3. Backup `/var/simp`
4. Backup ``` `puppet config --section server print vardir`/simp ```
5. *Optional:* Backup `/var/www`

Simple Full Backup Command

```
# tar --selinux --xattrs -czpvf simp_backup-$(date +%Y-%m-%d).tar.gz /etc/puppetlabs /
↪var/simp `puppet config --section server print vardir`/simp /var/www /var/simp
```

Simple Full Restore Command

```
# WARNING: This will overwrite your current system files!
tar --selinux --xattrs -C / -xzpvf simp_backup-<date>.tar.gz
```

Note: This only backs up data that is managed/owned by Puppet. It will not backup any application-centric data (such as the contents of *LDAP*).

4.5.4 Miscellaneous Guides

Guides in this section do not cleanly fit into any other section.

HOWTO Set Up Centralized Log Collection with Rsyslog

SIMP provides a pre-built set of classes in the `simp-simp_rsyslog` and `simp-rsyslog` Puppet modules for enabling centralized logging within the infrastructure. `simp-simp_rsyslog` is a profile module that uses `simp-rsyslog`.

There are no provisions here for setting up shared storage or deduplication. This is inherently not a use case that Rsyslog is well designed for and we suggest that you look at an alternative.

Note: For an overview of how to use Hiera to manage class parameters, please see *Classification and Data*.

Preparation

The `simp_rsyslog` Profile Module

SIMP's profile module, simp_rsyslog, helps configure systems for local and remote logging.

The `simp_rsyslog` class is automatically included on systems if the `simp` or `simp_lite` *scenarios* are used and, by default, configures local logging.

If scenarios are not being used, include the `simp_rsyslog` class on all systems including the log server. If you are using the default SIMP install, you can add it to the `simp::classes` array. Otherwise, you will need to use a standard Puppet `include` mechanism.

What is Logged

The `simp-simp_rsyslog` module uses the following parameters:

```
---
simp_rsyslog::default_logs     # A Hash of the default system logs to be collected
simp_rsyslog::log_collection   # Use this Hash to add logs to the default set
```

There are also Booleans available to enable collection of certain logs, such as those from OpenLDAP. See the `simp-simp_rsyslog` module for more details.

The Log Hash Format

The Hashes mentioned above are complex in nature but provide a clean interface to most aspects of log collection targeted to most users.

The *Puppet Data Type* representation of the Hashes is as follows:

```
Hash[
  Enum[
    'programs',
    'facilities',
    'msg_starts',
    'msg_regex'
  ],
  Array[String]
]
```

This means that you can have a `Hash`, with any of the keys `programs`, `facilities`, `msg_starts`, or `msg_regex` followed by an `Array` of `Strings`.

Using the following example `Hash`:

```
{
  'programs'   => [ 'sudo' ],
  'facilities' => [ 'cron.*' ],
  'msg_starts' => [ 'IMPORTANT:' ],
  'msg_regex'  => [ '*bad_guys*' ]
}
```

The `programs` line would match the following due to the highlighted section:

```
2017-03-14T15:26:53.589793+00:00 sample.host.name sudo: test_user : TTY=pts/0 ; PWD=/home/test_user ; USER=root ; COMMAND=/usr/sbin/visudo
```

The `facilities` line would match the following because the listed facility is `cron`:

```
2017-03-14T15:26:53.589793+00:00 sample.host.name CROND[31415]: (root) CMD (run-parts /etc/cron.hourly)
```

The `msg_starts` line would match the following due to the highlighted section:

```
2017-03-14T15:26:53.589793+00:00 sample.host.name kernel: IMPORTANT: This is an important message
```

The msg_regex line would match the following due to the highlighted section:

2017-03-14T15:26:53.589793+00:00 sample.host.name kernel: This system was prodded by **bad_guys** and should be watched

Set Log Servers

The list of log servers are usually set during simp config, and placed in the simp_config_settings.yaml *Hiera* file.

If this value needs to be changed, either simp config can be run again or the values below can be overridden in default.yaml or similar *Hiera* file to reach all nodes:

```
---
simp_options::syslog::log_servers:
  - 'logserver1.fullyqualified.domain'
  - 'logserver2.fullyqualified.domain'
simp_options::syslog::failover_log_servers:
  - 'failoverserver1.fullyqualified.domain'
  - 'failoverserver2.fullyqualified.domain'
```

If you list more than one primary log server your logs will be forwarded to **all** of the log servers in the array.

Failover log servers are optional.

Warning: If log forwarding is enabled on your log server, make sure you override the log server settings to NOT include itself. This will cause looping and will fill the disks on the system very quickly with repeated messages.

Note: It is common in big environments to use *DNS* aliases or to cluster servers so determining the name a server is using for logging is not straightforward. Because of this SIMP cannot reliably determine if a host is forwarding to itself.

TLS

If encryption is going to be used, make sure the certificates are in place. See the *Apply Certificates* documentation to understand how SIMP modules distribute certificates.

If SIMP is not being used to distribute certificates, see the documentation for the following PKI-related parameters in the rsyslog class of the simp-rsyslog module:

- rsyslog::pki
- rsyslog::app_pki_external_source_dir
- rsyslog::app_pki_dir

Enable Clients

To set up the clients enter the following settings in the `default.yaml` or similar *Hiera* file to reach all clients:

```
---
#If using TLS
simp_rsyslog::forward_logs: true
rsyslog::enable_tls_logging: true
```

or

```
---
#If not using TLS
simp_rsyslog::forward_logs: true
rsyslog::pki: false
rsyslog::enable_tls_logging: false
```

Enable the Server

To set up the server enter the following in the server's *Hiera* file:

```
---
# If using TLS
simp_rsyslog::is_server: true
simp_rsyslog::forward_logs: false
rsyslog::tls_tcp_server: true
```

or

```
---
# If NOT using TLS
simp_rsyslog::is_server: true
simp_rsyslog::forward_logs: false
rsyslog::tcp_server: true
rsyslog::tls_tcp_server: false
```

After `puppet` has run on all the systems, the logs from the clients will be stored in `/var/log/hosts/<client name>` directory on the log server.

`simp-simp_rsyslog` also sets up log rotation for these files by default using the `simp-logrotate` module.

Forwarding Log Files from a Log Server

If the log server needs to forward logs to another server, edit its *Hiera* file. Set `simp_rsyslog::forward_logs` to `true` and make sure that the `log_servers` array used on the relevant node does not include itself in the list. For example for a server using TLS:

```
---
simp_rsyslog::is_server: true
simp_rsyslog::forward_logs: true
rsyslog::tls_tcp_server: true
simp_options::syslog::log_servers:
```

(continues on next page)

(continued from previous page)

```
  - 'some-other-log-server.that.is.not.me'
simp_options::syslog::failover_log_servers:
  - 'some-other-failover-server.that.is.not.me'
```

This will forward the server's own logs, and all received client logs, to the specified servers.

HOWTO Discard Mail to Root

In many environments, you may have a central log collection facility for analyzing your log data. In this case, you may want to disable the default behavior of sending all e-mail to root.

The simplest method of discarding root's e-mail is to redirect it to `/dev/null` on the system using the following Puppet code.

> **Warning:** This is a **very** brute force approach and should only be used if you are **absolutely sure** that you want to discard all of root's e-mail on your systems.

```
postfix::alias { 'root':
  values => '/dev/null'
}
```

HOWTO Enable Core Dumps

SIMP disables system *core dump* capabilities by default for improved system security.

At times, users may need to re-enable core dumps for system troubleshooting.

Enabling Core Dumps at the System Level

Overall system core dumps can be enabled by setting the following in *Hiera*:

```
---
# Enable system core dumps
simp::sysctl::core_dumps: true

# Set the core dump output directory
simp::sysctl::core_dump_dir: /fully/qualified/path
```

This will also disable enforcement of core dump restrictions in *PAM*.

Preventing Core Dumps via PAM

If you decide to enable core dumps, you may want to still restrict them for users on your system.

To do this, you will need to add the following type of *puppet* code.

```
pam::limits::rule { 'prevent_core_dumps_all':
  # Add to all PAM domains
  domains => ['*'],
  # Set both hard and soft limits
  type    => 'hard',
  # Affect core dumps
  item    => 'core',
  # Set to '0'
  value   => 0,
  # Set at 99 in the order list (first match wins)
  order   => 99
}
```

Now, if you want to enable core dumps for the *root* user, you will want to add the following as well:

```
pam::limits::rule { 'allow_root_core_dump':
  domains => ['root'],
  type    => 'hard',
  item    => 'core',
  value   => 1,
  order   => 10
}
```

HOWTO Exclude YUM Repositories

By default, SIMP applies updates from all available repositories on a nightly basis. This ensures that bug fixes and security updates are applied to all systems without minute management in Puppet manifests. This section provides guidance on how to include or exclude specific repositories from nightly YUM updates.

Methodology

The `simp::yum::schedule::repos` and `simp::yum::schedule::disable` parameters in the `simp-simp` Puppet module control which repositories are enabled for nightly updating. Both variables must be specified in array format. `simp::yum::schedule::disable` is ignored if `simp::yum::schedule::repos` is not empty.

`simp::yum::schedule::repos` is used to specify an array of repositories from which updates are provided; no other repositories will be used.

`simp::yum::schedule::disable` is used to specify an array of repositories from which updates are not provided; all other repositories will be used.

HOWTO Enable Kerberos

For the latest documentation, see the documentation in the SIMP KRB5 Puppet Module.

The `simp-krb5` Puppet module helps administrators get a working *KDC* in place and clients configured to use the KDC.

The module, by default, sets up a fully functional KDC in your environment and generates keytabs for one admin user, and all of your hosts that it can discover via *keydist*.

Important: If you want to let SIMP automatically handle all of your hosts, you should follow the README included with the `simp-krb5` Puppet module and you should **NOT** proceed with this guide.

Note: The `keydist` discovery only works if the KDC is on the same system as your Puppet Server!

Warning: For distribution of keys to work properly, you **must** add `/var/simp/environments/<environment>/site_files` to your environment's `environment.conf` file and restart the `puppetserver` process.

The default in a `production` *SIMP Omni-Environment* is:

```
modulepath = modules:/var/simp/environments/**production**/site_files:$basemodulepath
```

Beginning with krb5

The following sections give a brief guide on how to get started with manual Kerberos configuration and distribution of keytabs, for more information, please see the MIT Kerberos documentation.

Creating Admin Principals

ACL Configuration

The following Puppet code snippet will create an *ACL* for your admin user that is **probably** appropriate for your organization.

```
krb5_acl { "${facts['domain']}_admin":
  principal       => "*/admin@${facts['domain']}",
  operation_mask  => '*'
}
```

Create Your Admin Principal

Your first principal will be an admin principal and will be allowed to manage the environment since it is in the admin group. This **must** be created on the KDC system.

Run the following command, as root, to create your principal:

```
# /usr/sbin/kadmin.local -r YOUR.DOMAIN -q "addprinc <username>/admin"
```

You can now do everything remotely using this principal. Load it using:

```
# /usr/bin/kinit <username>/admin
```

Creating Host Principals

Before you can really do anything with your hosts, you need to ensure that the host itself has a keytab.

SIMP uses the /var/simp/environments/<client_environment>/site_files/krb5_files/files/keytabs/<client_fqdn> directory for each host to securely distribute keytabs to the clients.

On the KDC, generate a principal for each host in your environment using the following command:

```
# /usr/sbin/kadmin.local -r YOUR.DOMAIN -q 'addprinc -randkey host/<fqdn>'
```

Create Your Keytabs

Then, create a separate keytab file for each of your created hosts using the following command:

```
# /usr/sbin/kadmin.local -r YOUR.DOMAIN -q 'ktadd -k <fqdn>.keytab host/<fqdn>'
```

Propagate the Keytabs

Move all of the resulting keytab files SECURELY to /var/simp/environments/<client_environment>/site_files/krb5_files/keytabs/<fqdn> on the *Puppet Server* as appropriate for each file.

Note: Make sure that all of your keytab directories are readable by the group **puppet** and not the entire world!

Then, update your node declarations to include 'krb5::keytab'.

Once the Puppet Agent runs on the clients, your keytabs will copied to /etc/krb5_keytabs. The keytab matching the system fqdn will be set in place as the default system keytab.

HOWTO Manage a TPM Device with SIMP

This document serves as a guide to enable and use TPM devices in SIMP. Currently, both *TPM* **1.2** and **2.0** are supported, but TPM 2.0 features are only available on EL7 systems.

TPM features in SIMP:

- TPM 1.2 and TPM 2.0 simulators
- Taking ownership
- Enabling a TPM-based PKCS#11 interface
- Intel TXT and Trusted Boot
- Enabling basic *IMA* measuring
  - Setting custom IMA policy (broken)
- Installation and configuration of the *HIRS* Provisioner

SIMP does not support clearing ownership or measured boot at this time.

Requirements

General Requirements:

- A host with a TPM 1.2 or 2.0 chip on the motherboard
- Tools to manage TPM hardware, namely tpm-tools or tpm2-tools
- A legacy, non-UEFI bootloader
- A BIOS password (one should be required to enable the TPM)
- Easy physical access to the machine to enter the BIOS password

Note: A simulated or software TPM may be used; however, it may not be able to perform all of the capabilities described.

Trusted Boot Hardware Requirements:

- A CPU with Intel Trusted Execution Technology (TXT)
- A chipset with Intel Trusted Execution Technology (TXT)

Starting with TPM

Ensure the system has a TPM. This can be done by looking for the character device `/dev/tpm0`, by checking the sys path manually, using tools provided to manage and utilize TPM hardware, or by checking the `tpm` or `tpm2` structured facts. The lack of a `/dev/tpm0` device does not necessarily preclude a TPM, as a software TPM may be present.

Note: The `tpm` and `tpm2` structured facts are included in the SIMP `tpm` and `tpm2` classes and rely upon functionality provided by the `tpm-tools` and `tpm2-tools` RPMs, respectively. The RPM packages are automatically installed if the classes are included and a TPM is detected by Puppet. To manually install these RPMs, enter the following commands with root permissions, either via sudo or running as root:

```
# yum install tpm-tools
# yum install tpm2-tools
```

The commands and sample outputs for successful tests verifying a TPM is present include:

```
# cat /sys/class/tpm/tpm0/device/active
1
# file /dev/tpm0
/dev/tpm0: character special (10/224)
# facter -p tpm.status
...
owned: 0,
enabled: 1,
active: 1,
...
# facter -p tpm2.tpm2_getcap.properties-variable.TPM_PT_PERSISTENT
...
ownerAuthSet => "clear",
endorsementAuthSet => "clear",
lockoutAuthSet => "clear",
reserved1 => "clear",
disableClear => "clear",
inLockout => "clear",
tpmGeneratedEPS => "set",
reserved2 => "clear"
...
# tpm_version
  TPM 1.2 Version Info:
  Chip Version:        1.2.18.160
  Spec Level:          2
...
# tpm2_getcap --capability="properties-fixed"
TPM_PT_FAMILY_INDICATOR:
  as UINT32:                0x08322e3000
  as string:                "2.0"
TPM_PT_LEVEL:               0
TPM_PT_REVISION:            1.49
...
```

TPM Simulators

The SIMP TPM 1.2 and TPM 2.0 simulator RPMs are packages containing compiled versions of IBM's Software TPM 1.2 and IBM's Software TPM 2.0. The RPMs can be regenerated by downloading the code from their respective project repositories, SIMP TPM 1.2 Simulator and SIMP TPM 2.0 Simulator. After installing the RPMs, the TPMs can be initialized.

Note: Root permission, either via sudo or running as root, is necessary to install and initialize the TPM simulators.

Installing and Initializing the TPM 1.2 Simulator

The TPM 1.2 simulator RPM is included with the SIMP repositories and can be installed via yum with the command:

```
# yum install simp-tpm12-simulator
```

To initialize the TPM 1.2 simulator on EL6, issue the following commands:

```
# service tpm12-simulator start
# service tpm12-tpmbios start
# service tpm12-simulator restart
# service tpm12-tpmbios start
# service tpm12-tpminit start
# service tpm12-tcsd start
```

To initialize the TPM 1.2 simulator on EL7, issue the following commands:

```
# systemctl start tpm12-simulator
# systemctl start tpm12-tpmbios
# systemctl restart tpm12-simulator
# systemctl start tpm12-tpmbios
# systemctl start tpm12-tpminit
# systemctl start tpm12-tcsd
```

The `tpm_version` command, provided by the `tpm-tools` RPM, can be used to verify the TPM 1.2 simulator has been initialized.

```
# tpm_version
```

Installing and Initializing the TPM 2.0 Simulator

The TPM 2.0 simulator RPM (EL7 only) is included with the SIMP repositories and can be installed via yum with the command:

```
# yum install simp-tpm2-simulator
```

To initialize the TPM 2.0 simulator, issue the following commands:

```
# runuser tpm2sim --shell /bin/sh -c "cd /tmp; nohup \
  /usr/local/bin/tpm2-simulator &> /tmp/tpm2-simulator.log &"
# mkdir -p /etc/systemd/system/tpm2-abrmd.service.d
# printf "[Service]\nExecStart=\nExecStart=/sbin/tpm2-abrmd -t socket" \
  > /etc/systemd/system/tpm2-abrmd.service.d/override.conf
# systemctl daemon-reload
# systemctl start tpm2-abrmd
```

The `tpm2_getcap` command, provided by tpm2-tools RPM, can be used to verify the TPM 2.0 simulator has been initialized:

```
# tpm2_getcap --capability="properties-fixed"
```

TPM 1.2 Management

Follow the steps below to enable and take ownership of the *TPM* **1.2**.

1. A BIOS password must be set to make sure no third parties can boot the host. Please set the admin password and the user password in the BIOS. If there is an option to require password at boot time, enable it. Do not enable Intel Platform Trust Technology (PTT) or Intel TXT at this time.
2. Before a TPM can be accessed by the operating system, it must first be enabled. This has to be done in the BIOS. Refer to the documentation provided with the hardware.
3. At this point, the SIMP TPM module can take over management of the device. Add `tpm` to the host's Hiera data according to the example below or use the `tpm_ownership` type directly.

```
simp::classes:
  - tpm

tpm::take_ownership: true
tpm::ownership::advanced_facts: true
```

Note: The `tpm_ownership` type does not support clearing the TPM. The process could possibly be destructive and has been left to be a manual process.

4. Run puppet

TPM 2.0 Management

Follow the steps below to enable and take ownership of the *TPM* **2.0**.

1. A BIOS password must be set to make sure no third parties can boot the host. Please set the admin password and the user password in the BIOS. If there is an option to require password at boot time, enable it. Do not enable Intel Platform Trust Technology (PTT) or Intel TXT at this time.
2. Before a TPM can be accessed by the operating system, it must first be enabled. This has to be done in the BIOS. Refer to the documentation provided with the hardware.
3. At this point, the SIMP TPM module can take over management of the device. Add `tpm2` to the host's Hiera data according to the example below or use the `tpm_ownership` type directly.

```
simp::classes:
  - tpm2

tpm2::take_ownership: true
tpm2::ownership::owner: set
tpm2::ownership::lockout:  clear
tpm2::ownership::endorsement: set
```

The passwords will default to automatically generated passwords using `simp::passgen()`. If you want to set them to specific passwords then set them in Hiera using the following settings (it expects a minimum password length of 14 characters):

```
tpm2::ownership::owner_auth: 'MyOwnerPassword'
tpm2::ownership::lock_auth: 'MyLockPassword'
tpm2::ownership::endorsement_auth: 'MyEndorsePassword'
```

> **Note:** The tpm_ownership type does not support clearing the TPM. The process could possibly be destructive and has been left to be a manual process.

4. Run puppet

Enable Basic IMA Measuring

This section assumes the previous section is complete, the TPM in the host is owned, and it is being managed with Puppet.

IMA is a neat tool that hashes the contents of a system, and stores that hash in the TPM. IMA is a kernel-level tool, and needs a few kernel parameters and reboots to be completely set up.

> **Note:** The default configuration of this module updates EFI boot parameters if they are present. If the system relies upon BIOS for boot, ensure there is not an EFI grub.cfg or grub2.cfg present or the BIOS grub config file will not be updated.

IMA Appraisal

IMA appraisal is the process that actually measures the state of the files and will stop changes to the filesystem if there is an issue detected.

1. Make sure / and /home are mounted with the i_version option. They are created by default with these options enabled.
2. Modify the Hiera data and add the following class:

```
simp::classes:
  - ima::appraise
```

3. Run Puppet to apply the policy changes to the system; the system will be configured to reboot into ima_appraise mode fix. Reboot the system.
4. The files on the system must now be measured and recorded. In order to do this, every file owned by root and included in the policy must be touched. This step will take some time. Puppet will provide notification not to reboot the system until the process is complete. Puppet will provide an ima_appraise_enforce_reboot notification when the process is complete.
5. Reboot the system again to set the ima_appraise to enforce mode.

If the IMA appraisal needs to be performed again to update files after the system is in enforce mode, the following steps may be taken:

1. Modify the Hiera data and add the following parameter:

```
ima::appraise::force_fixmode: true
```

2. Run Puppet to apply the policy to the system. The system will be configured to reboot into ima_appraise mode fix. Reboot the system.
3. Run the script ima_security_attr_update.sh. The files will be measured again and the values recorded; this will again take some time.

```
# /usr/local/bin/ima_security_attr_update.sh
```

4. When the appraisal is complete, Puppet will provide an `ima_appraise_enforce_reboot` notification. Set the `force_fixmode` attribute in the Hiera data back to `false`, then run Puppet again and reboot the system.

```
ima::appraise::force_fixmode: true
```

IMA Appraisal Debugging Tips and Warnings

If you reboot and are getting SELinux errors or you do not have permissions to access your files then you probably forgot to set `i_version` on your mounts in `/etc/fstab`.

If you reboot and it won't load the `initramfs`, then the `dracut` update didn't run. You can fix this by rebooting without the `ima` kernel settings, running `dracut -f` and then rebooting in `ima_appraise` mode `fix`.

Managing IMA policy

This module can also support modifying which files IMA watches by editing the `/sys/kernel/security/ima/policy`. Reference the module source file, located at `<environment path>/modules/ima/manifests/policy.pp` for further details on what can and cannot be measured.

Warning: The current RedHat implementation of IMA does not seem to work after inserting our default policy (generated example in `spec/files/default_ima_policy.conf`). It causes the system to become read-only, even though it is only using supported configuration elements. The module will be updated soon with more sane defaults to allow for at least the minimal amount of a system to be measured. A reboot will fix the issue, but with a TPM you will have to enter the password again.

1. Modify the Hiera data and add the following class:

```
simp::classes:
  - ima::policy
```

2. Run Puppet, then reboot.

Enabling Trusted Boot (tboot) (TPM 1.2 Only)

General Process

The steps in the section below provide guidance and automation to perform the following:

1. Set BIOS password
2. Activate and own the TPM
3. Install the `tboot` package and reboot into the `tboot no policy` kernel entry
4. Download SINIT and put it in `/boot`
5. Generate a policy and install it in the TPM NVRAM and `/boot`
6. Update GRUB
7. Reboot into a measured state

For more information about tboot in general, see:

- The tboot docs found in /usr/share/tboot-*/*
- https://fedoraproject.org/wiki/Tboot
- https://wiki.gentoo.org/wiki/Trusted_Boot
- https://software.intel.com/sites/default/files/managed/2f/7f/Config_Guide_for_Trusted_Compute_Pools_in_RHEL_OpenStack_Platform.pdf

Steps

1. Enable Intel TXT and VT-d in the BIOS.
2. Boot into the kernel you want to trust (do not worry, this kernel will be preserved!)
3. Follow the instructions in *Starting with TPM* and ensure:
   - The TPM is owned
   - You know the owner password
   - The SRK password is 'well-known' (-z)
4. Go to the Intel site and download the appropriate SINIT binary for your platform. Place this binary on a web server, on the host itself, or in a profile module. This cannot be distributed by SIMP for licensing reasons.
5. Add the following settings to your Hiera data for nodes that will be using Trusted Boot. It is recommended to use a *hostgroup* for this.
   - tpm::tboot::sinit_name - The name of the binary downloaded in the previous step
   - tpm::tboot::sinit_source - Where Puppet can find this binary
   - tpm::tboot::owner_password - The owner password

   Here is an example used for testing:

   ```
   tpm::tboot::sinit_name: 2nd_gen_i5_i7_SINIT_51.BIN
   tpm::tboot::sinit_source: 'file:///root/txt/2nd_gen_i5_i7-SINIT_51/2nd_gen_i5_i7_
   ↪SINIT_51.BIN'
   tpm::tboot::owner_password: "%{alias('tpm::ownership::owner_pass')}"
   ```

6. Add the tpm::tboot class to the simp::classes array with tpm.
   - The tpm::tboot class adds two boot entries to the GRUB configuration. One should read tboot, and there should be one above it called something along the lines of tboot, no policy.
   - The Trusted Boot process requires booting into the tboot kernel before creating the policy, so we have opted to create both entries. The intermediate, no policy boot option can later be removed by setting tpm::tboot::intermediate_grub_entry to false in Hiera.
7. Reboot into the tboot, no policy kernel entry.
8. Puppet should run at next boot, and create the policy. Log in, ensure /boot/list.data exists. If not, run puppet again.
9. Reboot into the tboot kernel entry.
10. Verify that the system has completed a measured launch by running txt-stat or checking the tboot fact.

```
# txt-stat
# facter -p tboot
```

Trusted Boot Debugging Tips and Warnings

- The parse_err command will show the error code, ready to lookup in the error table included in the zip.
- The tboot kernel option min_ram=0x2000000 (which is default) is **REQUIRED** on systems with more than 4GB of memory.
- Trusted Boot measures the file required to boot into a Linux environment, and updating those files will cause a system to boot into an untrusted state. Be careful updating the kernel packages and rebuilding the initramfs (or running dracut).

HIRS

The simp-hirs_provisioner module installs and configures the *HIRS* TPM provisioner on specified systems. An Attestation Certificate Authority (ACA) must be set up independently. Details of how to do this are provided on the *HIRS* website. Additionally, the acceptance tests in the SIMP hirs_provisioner module include an example of how to do so.

To install and configure the HIRS TPM Provisioner, add the following Hiera:

```
simp::classes:
  - hirs_provisioner

hirs_provisioner::config::aca_fqdn: 'aca.fullyqualified.domain'
```

HOWTO Configure NFS

This chapter describes multiple configurations of NFS. Topics include:

All implementations are based on the `simp-nfs`, `simp-simp_nfs`, and `simp-simp` modules.

Warning: `simp-nfs` version 7.0.0 and `simp-autofs` version 7.0.0 had major breaking changes in the pursuit of fixing long-standing bugs. General usage as noted in this document remains largely the same, but there are many changes to the API, some of which are nuanced. For example, parameters that used to accept hostnames now require IP addresses in support of firewalld. Please check your settings carefully on upgrade.

For ease of explanation, examples in this section use the concept of a *site profile* and are namespaced accordingly. The manifests are in a module called `site`. If using a different site profile, change the directory and the namespace in the examples.

Note: `simp-simp_nfs` and `simp-nfs` are not core modules, and their corresponding packages, `pupmod-simp-simp_nfs` and `pupmod-simp-nfs`, may need to be installed prior to following this guide.

Known Issues

Warning: A number of issues may render NFS inoperable. Please read through the known issues below before deploying into your environment.

Stunnel and Autofs

Warning: You must ensure that you are using autofs greater than `5.0.7-99`.

To use *NFS* over `stunnel` and `automount` directories with old CentOS 7 releases, you must use the appropriate `autofs` package.

To determine what version of `autofs` is installed, run `automount -V`.

Limited Kerberos Support

SIMP's NFS modules provide limited support for Kerberos and will not be discussed here.

- See the README for `simp-nfs` for information about the Kerberos support it provides and its integration with the `simp-krb5` module.
- The `simp-krb5` module is not fully integrated with NFS home directories at this time. See SIMP-1407 for details.

Exporting Directories

Goal:

- Export */var/nfs_share* on the server, for read-only access.
- Mount as */mnt/nfs* on the client.

Server

Create a manifest in your *site profile*. In this example the manifest in the `site` module is *nfs_server.pp*.

In *site/manifests/nfs_server.pp*:

```
class site::nfs_server (
  Stdlib::AbsolutePath                             $data_dir     = '/var/nfs_share',
  Simplib::Netlist                                 $trusted_nets = simplib::lookup('simp_
↪options::trusted_nets', { 'default_value' => ['127.0.0.1'] }),
  Array[Enum['none','sys','krb5','krb5i','krb5p']] $sec          = ['sys']
){
  include nfs::server

  file { $data_dir:
    ensure => 'directory',
    owner  => 'root',
    group  => 'root',
    mode   => '0755'
  }

  if !$nfs::stunnel {
    nfs::server::export { 'nfs_share':
      clients     => $trusted_nets,
      export_path => $data_dir,
      sec         => $sec,
      require     => File[$data_dir]
    }
  }
  else {
    nfs::server::export { 'nfs_share':
      # From the NFS server's perspective, the stunneled connections will
      # come from the local host
      clients     => [ '127.0.0.1' ],
      export_path => $data_dir,
```

(continues on next page)

(continued from previous page)

```
      insecure    => true,
      sec         => $sec,
      require     => File[$data_dir]
    }
  }
}
```

In *data/hosts/<your NFS server FQDN>.yaml*:

```
nfs::is_server: true

simp::classes:
  - 'site::nfs_server'
```

Client

The client can be configured to mount statically, automatically with a direct autofs mount, or automatically with an indirect autofs mount. Examples for all three types will be shown in this section. The indirect autofs mount example will configure autofs for an indirect mount that uses key substitution.

Note: If the NFS client is also a NFS server, there is autodetect logic in `nfs::client::mount` that attempts to detect that the client has both roles and, when this condition is detected, sets the NFS server's IP to `127.0.0.1` for efficiency. This detection logic has limitations. So, if you know the client is also the NFS server, it is better to disable the auto-detection by setting `nfs::client::mount::autodetect_remote` to `false`, and then set the `nfs::client::mount::nfs_server` to `127.0.0.1`.

Static Mount

Create a manifest in your *site profile*. In this example the manifest in the `site` modules is *nfs_client.pp*.

In *site/manifests/nfs_client.pp*:

```
 class site::nfs_client (
  Simplib::Ip                               $nfs_server,
  Enum['none','sys','krb5','krb5i','krb5p'] $sec = 'sys'
){

   $_mnt_point = '/mnt/nfs'

   file { $_mnt_point:
     ensure => 'directory',
     mode   => '755',
     owner  => 'root',
     group  => 'root'
   }

   nfs::client::mount { $_mnt_point:
     nfs_server  => $nfs_server,
     remote_path => '/var/nfs_share',
```

(continues on next page)

(continued from previous page)

```
     sec         => $sec,
     autofs      => false,
     # The mount point must already exist in a static mount
     require     => File[$_mnt_point]
   }
}
```

In *data/hosts/<your NFS client FQDN>.yaml*:

```
site::nfs_client::nfs_server: <your NFS server IP>

simp::classes:
  - 'site::nfs_client'
```

Direct autofs Mount

Create a manifest in your *site profile*. In this example the manifest in the `site` module is *nfs_client.pp*.

In *site/manifests/nfs_client.pp*:

```
 class site::nfs_client (
  Simplib::Ip                               $nfs_server,
  Enum['none','sys','krb5','krb5i','krb5p'] $sec = 'sys'
){

   nfs::client::mount { '/mnt/nfs':
     nfs_server  => $nfs_server,
     remote_path => '/var/nfs_share',
     sec         => $sec
   }
}
```

In *data/hosts/<your NFS client FQDN>.yaml*:

```
site::nfs_client::nfs_server: <your NFS server IP>

simp::classes:
  - 'site::nfs_client'
```

Indirect autofs Mount with Key Substitution

Create a manifest in your *site profile*. In this example the manifest in the `site` module is *nfs_client.pp*.

In *site/manifests/nfs_client.pp*:

```
 class site::nfs_client (
  Simplib::Ip                               $nfs_server,
  Enum['none','sys','krb5','krb5i','krb5p'] $sec = 'sys'
){

   nfs::client::mount { '/mnt/nfs':
```

(continues on next page)

(continued from previous page)

```
    nfs_server              => $nfs_server,
    remote_path             => '/var/nfs_share',
    sec                     => $sec,
    autofs_indirect_map_key => '*',
    autofs_add_key_subst    => true
  }
}
```

In *data/hosts/<your NFS client FQDN>.yaml*:

```
site::nfs_client::nfs_server: <your NFS server IP>

simp::classes:
  - 'site::nfs_client'
```

Exporting Home Directories for LDAP Users

Goal: Export home directories for LDAP users.

Utilize the SIMP profile module simp-simp_nfs:

1. simp-simp_nfs: Manages client and server configurations for managing NFS home directories.
2. simp_nfs::create_home_dirs: Enables an optional hourly cron job that binds to a *LDAP* server, simp_options::ldap::uri by default, and creates a NFS home directory for all users in the LDAP server. It also expires any home directories for users that no longer exist in LDAP.

Note: Any users logged onto a host at the time of module application will not have their home directories re-mounted until they log out and log back in.

Note: The simp-simp_nfs module utilizes an NFS root share which must be used to export any further directories from this server using NFSv4. This is because NFSv4 exports must exist in a single pseudo filesystem. See *Exporting Additional Directories on the NFS Home Server* for an example of how to do this.

Client

The following should be entered in the Hiera YAML files of all servers that need to mount home directories. Use *data/default.yaml* if you want to mount the home directories on all servers.

```
simp_nfs::home_dir_server: <your NFS server IP>

simp::classes:
  - simp_nfs
```

Server

The following should be entered in the Hiera YAML file of the NFS server.

In *data/hosts/<your NFS server FQDN>.yaml*:

```
nfs::is_server: true
simp_nfs::export::home::create_home_dirs: true

simp::classes:
  - simp_nfs::export::home
```

Exporting Additional Directories on the NFS Home Server

Goal:

- Export */var/nfs/share1* for read-write access to the 'administrators' group
  - This directory is located on the NFS server which is also sharing home directories for LDAP users.
  - The home directory share is set up by the `simp-simp_nfs` module.
- Statically mount the share to */share* on client systems.

The `simp-simp_nfs` module exports home directories under a root NFS share directory. Because NFSv4 exports exist in a single pseudo filesystem, each directory below that NFS share should be a bind mount to a directory on the NFS server.

The following example assumes you have set up the home server already following the instructions in the previous section and will be creating a bind mount under the root NFS share directory.

Server

Create a manifest in your *site profile*. In this example the manifest and the `site` module is *nfs_server.pp*.

In *site/manifest/nfs_server.pp*:

```
class site::nfs_server (
  #  Make sure the data_dir is the same as in simp_nfs.
  Stdlib::Absolutepath                              $data_dir     = '/var',
  Simplib::Netlist                                  $trusted_nets = simplib::lookup('simp_
↪options::trusted_nets', { 'default_value' => ['127.0.0.1'] }),
  Array[Enum['none','sys','krb5','krb5i','krb5p']] $sec = ['sys'],
) {

  #
  #  Exporting directories from the home directory server configured
  #  with the simp_nfs module.
  #
  include nfs::server

  # Create the directory where the data exists.
  file { '/var/nfs/share1':
    ensure => 'directory',
    mode   => '0775',
```

(continues on next page)

(continued from previous page)

```
    owner  => 'root',
    group  => 'administrators'
  }

  # Create a mount point under the NFS root share created in simp_nfs.
  file { "${data_dir}/nfs/exports/share1":
    ensure => 'directory',
    mode   => '0775',
    owner  => 'root',
    group  => 'administrators'
  }

  # Bind mount the share to the NFS root share created in simp_nfs.
  mount { "${data_dir}/nfs/exports/share1":
    ensure   => 'mounted',
    fstype   => 'none',
    device   => '/var/nfs/share1',
    remounts => true,
    options  => 'rw,bind',
    require  => [
      File["${data_dir}/nfs/exports/share1"],
      File['/var/nfs/share1']
    ]
  }

  # Export the directory
  if !$nfs::stunnel {
    nfs::server::export { 'share1':
      clients     => $trusted_nets,
      export_path => "${data_dir}/nfs/exports/share1",
      rw          => true,
      sec         => $sec
    }
  } else {
      nfs::server::export { 'share1':
      # From the NFS server's perspective, the stunneled connections will
      # come from the local host
      clients     => ['127.0.0.1'],
      export_path => "${data_dir}/nfs/exports/share1",
      rw          => true,
      sec         => $sec,
      insecure    => true
    }
  }
}
```

In *data/hosts/<your NFS server FQDN>.yaml*:

```
nfs::is_server: true

simp::classes:
  - site::nfs_server
```

(continues on next page)

(continued from previous page)

```
  - simp_nfs
```

Client

Create a manifest in your *site profile*. In this example the manifest in the `site` module is *nfs_client.pp*.

In *site/manifests/nfs_client.pp*:

```
class site::nfs_client (
  Simplib::Host                      $nfs_server,
  Enum['sys','krb5','krb5i','krb5p'] $sec = 'sys',
){

  include nfs

  $mount_point = '/share'

  # Only need the path relative to the root of the NFSv4 pseudo filesystem.
  $remote_path = '/share1'


  file { $mount_point:
    ensure => 'directory',
    mode   => '0775',
    owner  => 'root',
    group  => 'administrators'
  }

  nfs::client::mount { $mount_point:
    nfs_server  => $nfs_server,
    remote_path => $remote_path,
    sec         => $sec,
    autofs      => false,
    at_boot     => true,
    # The mount point must already exist in a static mount
    require     => File[$mount_point]
  }
}
```

Then include this manifest in Hiera for any system that should mount this share.

```
site::nfs_client::nfs_server: <your NFS server IP>

simp::classes:
  - site::nfs_client
```

Enabling/Disabling Stunnel

Stunnel is a means to encrypt your NFS data during transit.

Enable

If simp_options::stunnel is set to true, stunnel will be enabled.

If simp_options::stunnel is set to false and you do not wish to globally enable stunnel, you will need to set the following, in *data/default.yaml*:

```
nfs::stunnel: true
```

Disable

If simp_options::stunnel is set to true, but you do not want your NFS traffic to go through stunnel, set the following, in *data/default.yaml*:

```
nfs::stunnel: false
```

If simp_options::stunnel is set to false then stunnel is already disabled.

HOWTO Configure SNMPD

This document details how to use the simp-simp_snmpd Puppet profile module to configure the *SNMP* daemon.

Simple instructions to configure the snmpd daemon using the simp-simp_snmpd profile module are described in its README file.

Note: simp-simp_snmpd and its dependency puppet-snmp are not core modules and may need to be installed prior to following this guide.

SNMPD Configuration

There are two primary configuration directories:

- /etc/snmp/simp_snmpd.d
  - Files managed by puppet
- /etc/snmp/user_snmpd.d
  - Files not managed by puppet
  - Extended configurations should be placed here
  - Settings in this directory will override settings in the simp_snmpd.d directory

snmptrapd is disabled by default. The daemon can be enabled, but pupmod-simp-simp_snmpd will not configure it. If you need to run snmptrapd, set simp_snmpd::trap_service_ensure and simp_snmpd::trap_service_startatboot appropriately, and place any configuration files in /etc/snmp/user_trapd.d, with a .conf extension.

Agent Addresses and Firewall

By default, simp-simp_snmpd configures snmpd to listen on the local interface. Use simp_snmpd::agentaddress to toggle what interfaces snmpd will listen on.

Note: simp_snmpd::agentaddress is an array of strings that should follow the format defined in the man page for snmpd, under the LISTENING ADDRESS section.

The following is an example agent address:

```
---
simp_snmpd::agentaddress:
  - upd:161
  - tcp:%{facts.fqdn}:161
```

If simp_options::firewall is turned on, simp-simp_snmpd will parse the array of listening addresses to determine what ports should be opened. It does not, at this time, do anything for ipx or pvc. simp_snmpd::trusted_nets is used to determine what networks can access the ports.

Note: If the agent address is set in a conf file in the user directory, but not in Hiera or in the simp_snmpd resource call, simp-simp_snmpd will not open the ports in the firewall.

Access

simp-simp_snmpd configures SNMP v3, with

- User-based Security Model (USM)
- View-based Access Control Model (VACM).

The profile module, by default, installs two users:

- snmp_ro is configured for read only access to system view
- snmp_rw is configured for read/write access to everything

User passwords are auto-generated using simplib::passgen() and stored on the *SIMP Server* in a sub-directory in the *SIMP Writable Environment*:

/opt/puppetlabs/server/data/puppetserver/simp/environments/production/simp_autofile/gen_passwd.

Access is configured by /etc/snmp/simp_snmpd.d/access.conf

- To create the access.conf file, the profile modules uses a set of hashes.
- The default hashes are in the data/common.yaml file.
- These hashes are merged with any hash you defined in the *Hiera* files on the *Puppet Server*. Merging is described in Puppet docs
- To remove something from the default hash add the name of object with no keys

Note: To remove a user, or modify their password, the snmpusm utility must be used, or remove /var/lib/net-simp and run puppet. Changing the password in the hash or removing the keys will not change the password of an existing user.

Example hashes used to create users, views, group and give access:

User Hash

```
simp_snmpd::v3_users_hash
  username:
    authtype: MD5|SHA
    privtype: DES|AES
    privpass: 'your priv password'
    authpass: 'your auth password'
```

- If authtype or privtype is missing, it will use the modules `$defauthtype` and `$defprivtype`
- If either of the passwords are missing, it will be automatically generated using `simplib::passgen()`

View Hash

```
simp_snmpd::view_hash:
  viewname:
    included: [array of OIDs to include]
    excluded: [array of OIDs to exclude]
```

- One or both of included, excluded needs to be specified. Any number of OIDs can be listed
- It will create one view line for each OID in the list with exclude or include

Group Hash

```
simp_snmpd::group_hash:
  groupname:
    model: The security model to use (default to defsecuritymodel)
    secname: [array of user names to include in this group]
```

- It does not verify the user exists

Access Hash

```
simp_snmpd::access_hash:
  accessname:
    vread: view to use for reading access (default none)
    vwrite: view to use for write access (default none)
    vnotify: view to use for notify (default none)
    level:  priv|auth|noauth (default is defsecuritylevel)
    model: the model to use (default is defsecuritymodel)
    context: context to use (default "")
    prefix:  prefix for the context exact| prefix (default exact)
    groups: [array of groups to create this access for]
```

- It does not verify the group exists
- The access name is just a place holder
- For all hashes, anything with a default does not need to be included in the hash

Note: Any views, groups, or access lines set up in user conf files will be in addition to anything configured in the hash.

Remove Values from Default Hash

If you do not want the default user, or any of the views, groups, or access created, you can pass and empty hash and it will ignore that setting:

```yaml
---
simp_snmpd::v3_user_hash:
  snmp_ro:
  myuser:
    authpass: 'HardToBreakPassword'
    privpass: 'OtherPassword'
simp_snmpd::group:
  readonly_group:
    secname: myuser
```

- The above example will not create the snmp_ro user and add myuser. If the snmp_ro user is already created it will not delete it.
- It will override the default definition of readonly_group
- The quickest way to delete users or change the password is to configure the hashes and the remove the /var/lib/net-simp directory, stop the snmpd daemon, and run `puppet`.

Client

By default, the `net-snmp-utils` package is not installed, so snmpd utilities such as `snmpget`, `snmpset`, and `snmpwalk` will not be available. Set `simp_snmpd::manage_client` to `true` to install the package.

```yaml
simp_snmpd::manage_client: true
```

Note: After installation, the default security model, level, authentication, and privacy types will be configured. No default passwords will be configured.

Rsync MIBS and DLMODS

Rsync can be used to push out custom `MIBS` and dynamically loaded shared objects, or `dlmod`.

By default, rsync will copy `MIBS` into the directory used by `net-snmp`. To copy them elsewhere, set `simp_snmpd::rsync_mibs_dir` to the fully qualified path.

Note: The module will rsync the files to a `MIBS` directory under that path and add the directory to the `MIBS` path.

`DLMODS` are copied the same way as `MIBS`, using the `rsync_dlmod_dir` as the destination, creating a dlmod directory. In order to load dlmods, you must add the name of the module to the `simp_snmpd::dlmods` list. This will create a `dlmod.conf` file in `simp_snmpd`. The `.so` extension will be added. See the `Dynamically Loadable Modules` section in man page of `snmpd.conf` for more information.

Below is an example showing how to activate rsync of `MIBS` and `dlmods`:

```
---
simp_snmpd::rsync_dlmod: true
simp_snmpd::rsync_mibs: true
simp_snmpd::dlmods:
  - mymodulename
```

HOWTO Use unpack_dvd to Extract the Full OS RPM Package Set

The SIMP ISO provides a minimal set of packages.

If you require additional OS packages, you can extract them from vendor ISOs using **`unpack_dvd`**. Additionally you can extract the PXE files from the ISO to the rsync directories.

Extract the OS Packages

`Unpack_dvd` will, by default

- extract OS metadata from the ISO
- create a directory named `<os-family>/<os-version>` under `/var/www/yum`, the directory used for the yum server in simp/simp.
- extract the files on the ISO that directory.
- create a link `/var/www/yum/<os-family>/<os-major-version>` to above directory.
- create `Updates` directory if it does not exist
- create a link for all RPMs extracted from the ISO under the `Updates` directory.
- run **`createrepo`** in the `Updates` directory.
- If run as root, change permissions recursively on `Updates` and any directory with RPMs to be owned and readable by group `apache`

Use `unpack_dvd --help` to see all options available.

Note: If unpack_dvd can only determine the OS major version from the ISO metadata it will fail and ask you to supply a more detail version number using the `-v` option.

The following example will extract to the default location:

1. Log on and run `sudo su - root`.
2. Copy the appropriate vendor OS ISO(s) to the server.
3. If the server where you are unpacking the vendor ISO was **NOT** built using the SIMP ISO , you must create `/var/www/yum`.
4. unpack using **unpack_dvd**

```
# unpack_dvd -v <os version> <full path to iso>
unpack_dvd -v 7.8.2003 /myisodir/CentOS-7-x86_64-DVD-2003.iso
```

5. Ensure that subsequent *yum* operations are aware of the new RPM packages by refreshing the system's yum cache:

```
yum clean all && yum makecache
```

> **Warning:** At this time **unpack_dvd** does not work entirely with EL8 ISOs. EL8 introduced modules to repositories and **unpack_dvd** can not handle these. It will extract some but not all of RPMs on an EL8 ISO. The files extracted are enough to kickstart a basic EL8 system but some of the application stream files are not extracted and added to the repository correctly.

Extract PXE files

Extracting the PXE files was added to **unpack_dvd** in simp-utils-6.4.0. Use the `-X` option to tell **unpack_dvd** to extract the PXE files and the `--no-unpack-yum` option if you do not want to extract the yum files.

When PXE file extraction is enabled with the `-X` option, **unpack_dvd** will, by default,

- extract OS version information from the ISO metadata
- create a directory for the PXE files in the rsync directory used by simp/tftpboot in the production environment. * named <os-family>-<version>-<arch> * The default rsync directory is `/var/simp/environments/production/rsync/<os-family>/Global/tftpboot/linux-install/`.
- extract the files to the created OS directory.
- create a link <os-family>-<major-version>-<arch> to the above directory.
- if run as root, change the permissions recursively on created directory and its contents to those of the rsync directory.

Use `unpack_dvd --help` for options to change some of the defaults.

The following example will extract only the PXE files to the rsync directory:

```
# Place the -X options after the ISO name.
sudo su - root
# copy the iso to the system
unpack_dvd -v 8.0.1905 --no-unpack-yum /myisodir/CentOS-8-x86_64-1905-dvd1.iso -X
```

The following example will extract both the RPMs and PXE files to alternate directories:

```
sudo su - root
# Make the directories.  Set the permissions as needed.
mkdir -p /my/repodir
mkdir -p /my/tftpboot
# The PXE directory must follow the -X option.
# The -d options changes the directory to extract OS files
unpack_dvd -v 8.0.1905 -d /my/repodir /myisodir/CentOS-8-x86_64-1905-dvd1.iso -X /my/
↪tftpboot
yum clean all && yum makecache
```

4.6 Frequently Asked Questions

This section addresses some of the frequently asked questions (FAQ) about SIMP.

4.6.1 SIMP Version Guide

The SIMP versioning system has caused some confusion over time and this document serves as the authoritative reference for clarification.

Top-Level SIMP for 6.X+

Note: This is the version number that you get when you run **rpm -q simp**

The top level SIMP version for SIMP releases from 6.0.0 onward will be following Semantic Versioning 2.0.0.

In short, this means (from the reference):

Given a version number `MAJOR.MINOR.PATCH`, increment the:

1. `MAJOR` version when you make incompatible API changes
2. `MINOR` version when you add functionality in a backwards-compatible manner
3. `PATCH` version when you make backwards-compatible bug fixes

Sub-Component Versioning

For all versions of SIMP, sub-components generally follow Semantic Versioning 2.0.0.

4.6.2 What is the Password Complexity for SIMP?

The following are the default password requirements for a standard SIMP system. This is based off of an amalgam of various password policies and may vary based on individual policies that are set for your installation.

The default complexity is enforced in both *PAM* and *LDAP*, which are managed by SIMP's modules:

- simp/pam for pam
- simp/simp_openldap for *OpenLDAP* servers
- simp/simp_ds389 for *389-DS* servers

Warning: If you are using the *SIMP Compliance Engine* to enforce a specific compliance profile, the enforced rules may be different than those shown below. Be sure to examine the profile to understand the rules applicable to your system.

Complexity Rules

- 15 Characters or greater
- 1 Upper case letter
- 1 Lower case letter
- 1 Number
- 1 Special character
- No more than 2 repetitions of the same character
  - OK: `aab`
  - BAD: `aaa`
- No more than 3 repetitions of a character from the same character class
  - OK: `abcD`
  - BAD: `abcd`
- No more than 4 characters in a monotonic character sequence
  - OK: `1b2c3d`
  - BAD: `aBcDe`
  - BAD: `EdCbA`
- Cannot contain your username in straight or reversed form
- Cannot contain items from your GECOS field (usually your full name)
- Must have more than 4 character changes from the old password
- Must not be one of the last 24 passwords that you have used

You can check whether or not a password will meet the system requirements as follows:

- On systems that use **`pam_pwquality`**, use the **`pwscore`** utility.
- On systems that use **`pam_cracklib`**, use the **`cracklib-check`** utility.

Note: Locked out accounts **will** unlock automatically after 15 minutes for non-root users and one minute for the root user.

Important: Systems that use **`pam_cracklib`** may differ slightly in behavior from systems that use **`pam_pwquality`**. If issues are found, please file a bug with the *OS* vendor noting the issue.

4.6.3 Enabling root Logins

Keeping in line with general best practice, SIMP does not allow root to login to the system remotely or at local terminals by default.

However, there may be cases where you need to login as root for perfectly valid reasons.

Enabling Terminal Logins

To allow root to login at the terminal, you will need to set the useradd::securetty Array to include all tty devices from which you wish to allow root access.

For example, to allow the root user to login at the first three virtual consoles and the first serial device, you would place the following in *hiera*:

```
useradd::securetty:
  - tty0
  - tty1
  - tty2
  - ttyS0
```

Enabling Remote SSH Logins

If you need to allow remote root logins over SSH (we **highly** advise against this), you can add the following to *hiera*:

```
ssh::server::conf::permitrootlogin: true
```

4.6.4 Meltdown and Spectre

SIMP does not force users to update to kernels, or other system components, that work around either *Meltdown* or *Spectre*. This is done to not adversely affect production environments.

Please follow your vendor's instructions on mitigating these attacks on your systems.

You can find additional information on the exploits at the Meltdown Attack website.

You find additional information on how automatic system updates work in SIMP by referring to the *Nightly Updates* section of the *SIMP User Guide*.

4.6.5 Why aren't audit logs being forwarded to syslog?

Audit logs can be sent to syslog in addition to being persisted locally in /var/log/audit. By default, SIMP disables forwarding of audit logs due to the excessive size of the collected logs.

When audit logs are sent to remote syslog servers, the logs can quickly overwhelm the underlying network.

If forwarding of audit logs via syslog is appropriate for your site, you can enable that forwarding by setting the following in *hiera*:

```
auditd::config::audisp::syslog::drop_audit_logs: false
```

4.6.6 Why Does Logrotate Complain About Repeated Configuration Settings

As of SIMP 6.2.0, SIMP-managed **logrotate** rules are now in `/etc/logrotate.simp.d` instead of `/etc/logrotate.d`. The rules in `/etc/logrotate.d` are still applied, but **logrotate** is configured to read the rules in `/etc/logrotate.simp.d` first.

This change was made to ensure SIMP-managed rules take precedence over vendor-supplied rules. When multiple rules are specified for the same file, only the first rule is applied. Any subsequent rules are discarded.

Note: For some versions of **logrotate**, a rule with a duplicate is discarded *in its entirety*, even if only one of the managed log files is a duplicate. This means the remaining log files specified in that discarded rule will *not* be rotated!

Because the location of the SIMP-managed **logrotate** rules has changed, any previously existing (but now obsolete) SIMP rules will still reside in `/etc/logrotate.d`. Although these rules cause no issues with **logrotate**, they may be confusing to system administrators. So, you may wish to manually remove these rules.

The following script can be used to identify obsolete SIMP **logrotate** rules and they can be removed manually as necessary.

```
grep -rl 'managed by puppet' /etc/logrotate.d
```

4.6.7 Omni-Environment Already Exists Error

This FAQ covers what to do when running **simp config** outputs an "`Unable to configure:  Invalid SIMP omni-environment for 'production' exists`" error

If installing SIMP on a system that has a pre-existing `production` *Puppet Environment*, you may encounter the following error when running **simp config**:

```
Unable to configure: Invalid SIMP omni-environment for 'production' exists:
  >> Puppet environment 'production' exists
  >> Secondary environment 'production' does not exist
```

or (depending on the version):

```
Unabled to configure: Invalid SIMP omni-environment for 'production' exists:
  >> Puppet environment 'production' exists
  >> Secondary environment 'production' does not exist
```

Resolution Options

A New Start

If you want to start clean, then just rename the `production/` directory (`/var/simp/environments/production`) to `production.bak` and re-run **simp config** and **simp bootstrap**.

A Different Environment

If you want to keep your existing `production` environment, you can use an alternate environment during your initial configuration of the system.

To do this, see *HOWTO Use an Alternate 'simp config' Environment*.

Remember to assign nodes to this environment using your *ENC* of choice.

For a simple built-in solution, see *HOWTO Set Up SIMP's External Node Classifier*.

Puppet Enterprise Users

If you are using Puppet Enterprise, we recommend you use a control repo. See the section on *HOWTO Set up a SIMP Environment in a Control Repository*.

4.6.8 Puppet-Related Issues

What do I do when puppet gets certificate verification errors?

If you are experiencing an error when running puppet such as `certificate verify failed` then there are a few things that you can try in an attempt to troubleshoot the issue.

1. Make sure that your system clocks are within one hour of each other.
2. Ensure that the forward and reverse lookup for the FQDN of your systems is correct and matches the hostnames listed in the output of **`openssl x509 -text -noout -in $(puppet config print hostcert) | less`**

   HINT: Look at the `Subject` and `X509v3 Subject Alternative Name` sections.
3. Check that the connection from the client system to the server can successfully connect:

```
openssl s_client -host $(puppet config print server) \
-port $(puppet config print masterport) \
-cert $(puppet config print hostcert) \
-key $(puppet config print hostprivkey) \
-CAfile $(puppet config print localcacert)
```

If none of these items provides useful information, you may need to check permissions on your server and/or dig more closely into the **`puppetserver`** or client logs.

Why is my Puppet Agent crashing when run with `--debug`?

The bug FACT-1732 can cause Facter to crash while attempting to print a Bignum-sized number. On 64-bit systems, this is any integer greater than **4611686018427387903**[1].

Note: Facts provided by SIMP's modules are **not affected** by FACT-1732.

- This issue only affects facts introduced from *non-SIMP* sources.
- It will cause the commands **`puppet agent -t --debug`** and **`facter -p`** to fail with errors when they encounter Bignum-sized *numeric* fact values.
- You can fix your own facts to avoid FACT-1732 by returning any potentially large numeric value as a String.

Older versions of SIMP and FACT-1732

SIMP modules' facts haven't been susceptible to FACT-1732 since SIMP 6.1.0-0. Before that, the `shmall` and `shmax` facts from simp/simplib would crash on systems with a lot of memory.

When should I run `puppet generate types`?

The **`puppet generate types`** command addresses the problem of Puppet Environment isolation (SERVER-94) by generating *custom type* metadata definitions for each environment. The command must therefore be re-run in response to changes in Puppet environments and compilers.

By default, SIMP automates some of these cases using **`incron`** triggers. However, there are still some situations where you will have to make sure that **`puppet generate types`** is run.

Situations `incron` handles automatically

By default, SIMP configures the **`incron`** daemon to automatically run **`puppet generate types`** under either of the following circumstances:

- The **`puppet`** or **`puppetserver`** binaries have been updated.
- A new *Puppet environment* directory is added to the system.

This behavior is managed by the Puppet class `pupmod::master::generate_types`.

Differences from Previous versions of SIMP

Earlier versions of **`simp-pupmod`** (7.6.0 through 7.7.1, shipped with SIMP 6.2.0-0 through 6.3.1-0) attempted to automatically trigger **`puppet generate types`** under every relevant circumstance. However, some of the triggers could add too much load on the system and were removed from the **`incron`**'s watchlist.

These situations must be addressed by other means (see below).

[1] 4611686018427387904 == 2 ^{62}

Situations incron doesn't handle

incron does not handle all cases, so you will need to ensure that **puppet generate types** is after the following events:

- A new *module* that includes custom types is added to an existing environment.
- An existing custom type's internal code is updated.

Generating types manually

You can run the **puppet generate types** command as **root** on the Puppet Server. However, in order to ensure that the Puppet Server process can read the generated files, you must also ensure they have the correct ownership and permissions. One way to do this is by running the following command:

```
(umask 0027 && sg puppet -c 'puppet generate types --environment ENVIRONMENT')
```

This creates all files with the correct group ownership.

Automatically generating types after r10k deploy environment

If you are using *r10k* to deploy *Control Repository* branches using **r10k deploy environment**, you can set the "generate_types" option in the r10k.yaml file to automatically run **puppet generate types** for each environment after it is deployed:

Listing 4: Inside r10k.yaml:

```
# Important: this option *must* be defined under a top-level `deploy:`
deploy:
  generate_types: true
```

If you use **r10k** to deploy modules as **root** on the Puppet Server, you must ensure that the generated files have the correct ownership and permissions for the Puppet Server process to read them. One way to do this is by running the following command:

```
( umask 0027 && sg puppet -c '/usr/share/simp/bin/r10k deploy environment production' )
```

This will deploy the environment with the correct permissions and group ownership. If deploy/generate_types is set to true in r10k.yaml, it will also generate environment-safe type metadata files with the same permissions and ownership.

4.6.9 Why does SIMP use rsync?

SIMP uses **rsync** to manage both large files and large numbers of small files. This is to reduce the number of resources in the catalog and take advantage of rsync's syncing engine to reduce network load and Puppet run times.

The common SIMP use cases for rsync include:

- **clamav**
- **tftpboot**
- **named**
- **dhcpd**

Large Files

Both the system kickstart images, and the clamav virus definitions are fairly large (100MB+). This isn't itself an issue. However, as the file changes over time, Puppet would have to transfer the entire file every time it changes.

To access the accuracy of a file defined in the catalog, Puppet checksums the file and compares it to the checksum of the expected content. This process could take a long time, depending on the size of the file. If the sums don't match, Puppet replaces and transfers the entire file. Rsync is smarter than that, and only replaces the parts of the file that need replacing. In this case, rsync saves bandwidth, Puppet run time, and a few CPU cycles.

Large Numbers of Files

named and **dhcpd** are the opposite situation. In both of these cases, they may manage large numbers of files. Typically, like above, Puppet would have to checksum every file and see if it needed changing, with each file setting up a new connection to the Puppet server transferring each file individually. A small number of file resources wouldn't be the end of the world when managing something with Puppet, but rsync limits every one of these files to one transaction and one resource. If you have a highly complex site, without rsync, this could grow your catalog to the point where Puppet would have a difficult time processing the entries in a timely manner. Syncing directories in this fashion also allows for configuration to be managed outside of the Puppet space.

Where are the rsync files?

SIMP packages the rsync materials in the `simp-rsync-skeleton` RPM, which installs a file tree `/usr/share/simp/environment-skeleton/rsync`. This directory is automatically installed in the *SIMP Secondary Environment* for the `production` *SIMP Omni-Environment* created by **`simp config`** (`/var/simp/environments/production`) or the corresponding directory for a new environment created by **`simp environment new`**.

The rsync directories in the SIMP Secondary Environment are shared by the `simp::server::rsync_shares` class, which is included on the SIMP server if the `simp_options::rsync` catalyst is enabled.

4.6.10 Recovering from SELINUX policy failures

If you experience a failed boot after running **`simp bootstrap`** with an error that says something along the lines of `Failed to load SELINUX policy, freezing`, follow these instructions:

1. Reboot into a rescue shell (instructions on EL8 and EL7). You may need your GRUB password that was set during **`simp config`** or set using the simp/simp_grub module.
2. Reinstall the selinux policy: **`yum reinstall -y selinux-policy-targeted`**
3. Tell the kernel to relabel all files during next boot: **`touch /.autorelabel`**
4. Reboot

4.6.11 YUM Repo Issues

This FAQ covers various issues that relate to YUM repositories and SIMP systems.

Global repo_gpgcheck=1

> **Warning:** Disabling `repo_gpgcheck` should only be done against repositories that you ultimately trust. Doing otherwise could allow untrusted repository maintainers to compromise your system.
>
> More information can be found on this SCAP Security Guide Mailing List Thread.

The *DISA STIG* requires that the `repo_gpgcheck` setting be set to `1` globally on *EL* systems.

When SIMP is set into STIG enforcing mode using the *SIMP Compliance Engine*, it will automatically flip the global `repo_gpgcheck` setting to `1` in accordance with the STIG.

Unfortunately, this will break repositories such as *EPEL* and the commercial *RHEL* repositories.

To mitigate this, you can modify the global settings by changing the appropriate value in the `yum::config_options` Hash. However, doing this will show as a finding during STIG compliance scans.

Alternatively, you can update each repository that is having issues and disable GPG checking for just that repository using the yumrepo puppet resource.

Didn't find the information you need? Please see *SIMP HOWTO Guides*

4.7 Contributing to SIMP

4.7.1 Introduction

Thank you for taking interest in contributing to the SIMP project!

We firmly believe that this type of project can't be accomplished by a single team and that everything matters from bug reports to documentation patches.

4.7.2 Contribution Procedure

Contributing updates to SIMP

We use the standard GitHub workflow for SIMP development with the exception that we use a Squash and Merge merge method for pulling in changes. This is done to maintain a more legible commit history on *master*.

1. Search the SIMP JIRA for an open ticket that is relevant to the issue or open a new one.
2. Use the GitHub GUI to fork and clone the repository (we'll use `pupmod-simp-iptables` for the rest of this walkthrough)
3. Clone the repo you want to work on:
   - `git clone git@github.com:<YOUR_GITHUB_NAME>/pupmod-simp-iptables iptables`
4. Enter the directory and create a feature branch: `git checkout -b SIMP-XXXX`
5. Do your work! *(Including tests, of course)*
   - See *gsg-contributors_guide-contribution_procedure-testing_your_submissions* for detailed guidance on test procedures.
6. Commit your work. We will squash your pull request into one commit when we merge it, so you can use as many commits as you'd like.

   Important: The **first** commit (and/or Pull Request) message should follow our *Commit Message Conventions*

7. Push your changes to GitHub on your feature branch:
   - `git push origin SIMP-XXXX`
8. Using the GitHub GUI, create a pull request from your feature branch to the branch of the original repo that you want to contribute to. Leave the 'Allow edits from maintainers' checkbox checked to let a team member add commits to your pull request.
9. GitHub Actions will run the spec tests for the branch and a member of the SIMP team will review your submission. You should receive emails from GitHub as code reviews progress.

Commit Message Conventions

An example commit message that following the SIMP conventions:

```
(SIMP-999) Fix the broken thing [50 chars max]

Discussion about the fix (if needed) [each line: 72 chars max]

SIMP-998 #comment Comment on a related issue [72 chars max]
SIMP-999 #close
```

The first commit message should be the following format:

- First line:
  - Start with the Issue name in parentheses [e.g., `(SIMP-999)`], followed by a summary of the change
  - No longer than **50** characters
  - Followed by a line of white space
- Subsequent lines:
  - Each line should be no longer than **72** characters

  - Describe the previous behavior, why it was changed, and the changes in detail
- Issue references:
  - JIRA issues can be referenced at the end of the commit message
  - It is recommended to only use JIRA Smart Commit Tags `#comment` and `#close`
  - Avoid `#resolve` and `#time` as it will not update JIRA until after the issue is merged

Testing Your Submissions

First off, thank you again for your contribution! Things don't get better without your help!

This section contains two sets of guidelines. First, ones that are recommended for external contributions from the community. Second, ones that are expected to be adhered to by the core development team.

External Contributors

We will happily accept all levels of contributions, small, medium, or large without any tests.

However, for us to quickly and effectively assess your contribution you should either add unit (`rspec-puppet`) and/or acceptance (`beaker`) tests.

As the size of the contribution increases, this becomes increasingly important, because, depending upon the complexity of the changes, it may simply be too difficult to do a timely assessment of such a contribution without corresponding tests. In these cases, it would be best if you split your contribution into smaller pull requests that are easier to assess.

Core Developer Contributions

The core development team is expected to follow these guidelines when adding code to the project.

In all cases, a cursory `grep` through the simp-doc project should be done and a ticket should be entered if the overall project documentation may be affected by your change.

This should also be done by the core development team for any external contributions, since it is unreasonable to expect external contributors to take the effort to dig through the simp-doc project.

Trivial Contributions

Trivial contributions are those that constitute a small documentation update, code correction, or bug fix consisting of only a few lines. These contributions must not negatively impact the behavior of the user experience or code.

Trivial contributions do not require an associated ticket and may be covered under a `SIMP-MAINT` branch.

Trivial contributions require **one maintenance team member** review and may optionally add additional unit or acceptance testing.

Minor Contributions

Minor contributions are those that add a feature or fix a larger bug in components that are more than five or ten lines and/or are not only documentation updates.

Minor contributions must have unit tests and should have acceptance tests. Acceptance tests may be deferred but a ticket must be filed with an explanation and a link in the PR if the acceptance test addition is deferred.

Minor contributions require **one maintenance team member** review. The reviewing team member may decide that acceptance tests are required based on the understandability of the contribution.

Major Contributions

Major contributions are any changes that affect multiple parts of the system, any contribution of moderate or higher cyclomatic complexity, or anything that adds a breaking change to the system.

Major contributions must have unit tests that cover all major code paths and pay particular attention to edge cases.

Acceptance tests must also be provided that cover the primary usage of the code that, at a minimum, test the code in a way that end users would use it.

User facing changes should also contain documentation updates that cover the expected use cases.

Major contributions require **two maintenance team member** reviews.

Emergency Contributions

On occasion, a fix or patch will need to be made with a very short turn around time. These may include up to *Minor Contributions* and may be added after **two code reviews** without the addition of tests. However, a ticket must be added that notes a requirement for tests to be added to the specified capability. This ticket should link directly to the PR that added the code for later reference.

Experimental Contributions

Experimental contributions are changes that may not be ready for the end user, but that need reviews and/or attention.

For items that are not end-user facing, such as the testing components or frameworks, there may be a need to try out different techniques prior to releasing a full update. These may be added to the unstable `master` branch without testing but tests should be added if the changes will be released in the future.

4.7.3 Maintenance Procedure

If you're a SIMP maintainer, you're in the right spot! Otherwise, you'll want to head over to the *Contribution Procedure*.

This section exists to document the correct procedure for SIMP Maintainers to update and release code. These procedures are above and beyond the *Contribution Procedure*.

Amending Changes to Submitted Pull Requests

Note: It is recommended that all SIMP Maintainers use the hub Git extensions and all examples in this section will expect that hub is installed and ready for use.

1. Clone the source repo:

   - `git clone https://github.com/simp/simp-doc doc`

   Important: We use `git clone` instead of `hub clone` so that we can't accidentally push to the main SIMP repositories. While we have protected branches for the critical components, one wrong command and life can get unpleasant.

2. Pull down the pull request (PR) as found on the GitHub GUI. The local branch should match the branch in the PR (for example, branch SIMP-XXXX):

   - `hub checkout https://github.com/simp/simp-doc/pull/9999 SIMP-XXXX`

3. Review the code or make your additional changes

   - `HACK HACK HACK`

4. Add a new commit with your changes:

   - `git commit -a -m "I made the docs better"`

5. Set up the target repo for a push:

   - `hub remote set-url -p jeefberkey`

6. Push your new commit to the feature branch of the **owner** of the pull request. In this example, the owner is *jeefberkey*, and the feature branch name is *SIMP-XXXX*: `hub push jeefberkey HEAD:SIMP-XXXX`

7. The pull request has been updated, and participants have received an email

Tagging and Releasing Components

Warning: The intent of this section is to list the current state of the SIMP Team's release processes. Since these processes are constantly being improved and automated, you can expect this section content to evolve as well and may be best served by reading the version from the `master` branch of the `simp-doc` repository.

This section describes the release procedures for SIMP. The SIMP Team releases:

- Individual Puppet modules as tar files to PuppetForge
- Individual Puppet modules as signed RPMs to the appropriate path at https://download.simp-project.com/simp/yum/.
- Ruby gems for building and testing to RubyGems.org
- SIMP system dependencies as signed RPMs to the appropriate path at https://download.simp-project.com/simp/yum/.
- SIMP-system ISOs to https://download.simp-project.com/simp/ISO/.

SIMP component releases listed above are based off of an official GitHub release the SIMP Team has made to a corresponding SIMP GitHub project. In the case of a SIMP ISO, the component release tag is for the `simp-core` project, which compiles existing, released component RPMs and dependencies into an ISO.

Note: The SIMP ISO includes RPMs for Puppet modules that are not maintained by SIMP. When a suitable signed RPM does not already exist for such a module (e.g., `kmod` Puppet module maintained by `camptocamp`), SIMP builds a signed RPM for that project, using one of that project's GitHub release tags.

All modules provided by the SIMP Project, are directly sourced from SIMP-controlled repository forks. We do not pull directly from upstream sources.

Component Versioning

Version Philosophy

SIMP follows Semantic Versioning 2.0.0 and has the following versioning structure: `X.Y.Z`, where

- `X` indicates breaking changes
- `Y` indicates new features
- `Z` indicates bug fixes.

When can a component be released?

A component can be released when

- `X`, `Y`, or `Z` changes have been made.
- All dependencies of the component has been released.
- If a SIMP-owned component, all unit, acceptance, and integration tests pass.
- If a SIMP-owned component, the version number has been appropriately bumped and the corresponding changelog has been updated.

The SIMP project version/changelog files are as follows:

Component Type	Version Files	Changelog Files
SIMP-owned Puppet module	`metadata.json` and `CHANGELOG`	`CHANGELOG`
Ruby gem	`lib/simp/\*\*/version.rb` and either `build/<name>.spec` or `CHANGELOG.md`	`build/<name>.spec` or `CHANGELOG.md`
Other ISO-related project	`build/<name>.spec`	`build/<name>.spec`
`simp-doc`	auto-generated	`CHANGELOG`
SIMP ISO (`simp-core`)	`Changelog.rst` and `src/assets/simp/build/simp.spec`	`Changelog.rst` and `src/assets/simp/build/simp.spec`

What file changes require a version change?

Any changes to mission impacting (significant) files require a new release. In general, this includes the `metadata.json`, `CHANGELOG` and `hiera.yaml` files for Puppet modules, as well as files in the following directories:

- `build/`
- `data/`
- `files/`
- `functions/`
- `lib/`
- `manifests/`
- `scripts/`
- `share/`
- `src/`
- `templates/`
- `types/`

Changes to the following do not typically warrant a new release of a component:

- Any hidden file/directory (entry that begins with a `.` such as `.rspec`, `.gitignore`, `.gitlab-ci.yml`)
- `Gemfile`
- `Gemfile.lock`
- `Rakefile`
- `spec/`
- `doc/`

What version/changelog linters are available?

In the `simp-rake-helpers` Ruby gem, we have the following version/changelog-related linters for SIMP Puppet modules:

- `pkg:create_tag_changelog`: Generates an appropriate annotated tag entry from a `CHANGELOG`. The results should be carefully examined to ensure the output is correct, because processing stops at the first invalid changelog entry.
- `pkg:compare_latest_tag`: Compares mission-impacting files with the latest tag and identifies the relevant files that have changed. When mission-impacting files have changed, fails if:
  1. Latest version cannot be extracted from the top-most `CHANGELOG` entry.
  2. The latest version in the `CHANGELOG` (minus the release qualifier) does not match the version in the `metadata.json` file.
  3. A version bump is required but not recorded in both the `CHANGELOG` and `metadata.json` files.
  4. The latest version is smaller than the latest tag (version regression).
- `pkg::check_version`: Compares all files with the closest tag and logs an error if any files have changed, but the version has not been updated, or the versions in the `metadata.json` and `CHANGELOG` files do not match.

Note: Moving forward, these linters will be enhanced to handle the version/changelog nuances of the other projects SIMP releases and will be included as tests in all TravisCI builds.

SIMP-Owned Puppet Module Tag And Release Procedures

This section will describe the partially-automated, release procedures we use for SIMP-owned Puppet modules.

For demonstration purposes, we will be using the `pupmod-simp-iptables` project, which uses the `master` branch as its development branch.

Note: You can identify whether a Puppet module is owned by SIMP, by examining the outer-most `name` entry in the module's `metadata.json` file. The value for the `name` key will be of the form <*owner*>-<*module name*>.

Pre-Release Checklist

The bulk of the work to release a component is to verify that the component is ready for release. Below is the list of verifications that must be executed before proceeding with the release. If any of these checks fail, the problem identified must be fixed before you can proceed with the tag and release steps.

Checklist Steps

- *Verify a release is warranted*
- *Verify the CHANGELOG*
- *Verify the component's dependencies*
- *Verify RPMs can be created*
- *Verify unit tests pass*
- *Verify acceptance tests pass*
- *Verify interoperability with last SIMP release*
- *Verify the component RPM upgrade succeeds*
- *Verify the component yields valid SIMP ISOs*
- *Verify the component works in an actual SIMP system*

Verify a release is warranted

This check verifies a new release is warranted and the version has been properly update:

1. Clone the component repository and checkout the development branch to be tagged

```
git clone https://github.com/simp/pupmod-simp-iptables.git
cd pupmod-simp-iptables
git checkout master # this step isn't needed for master branch
```

2. Run the `pkg:compare_latest_tag` rake task

```
bundle update
bundle exec rake pkg:compare_latest_tag
```

Important: If this check indicates no new tag is required, there is no reason to continue with the release procedures.

Verify the CHANGELOG

This check verifies that the CHANGELOG information can be properly extracted:

1. Run the `pkg:create_tag_changelog` rake task

   ```
   bundle exec rake pkg:create_tag_changelog
   ```

2. Manually verify the changelog information is emitted and complete.

   - It should begin with `Release of x.y.z` and then be followed by one or more comment blocks. For example,

     ```
     Release of 6.0.3

     * Thu Aug 10 2017 Nick Markowski <nmarkowski@keywcorp.com> - 6.0.3-0
       - Updated iptables::listen::tcp_stateful example to pass valid
         Iptables::DestPort types to dports
     ```

   - It should be understandable.
   - It should be free from typos.
   - Any parsing error messages emitted should *only* be for changelog entries for earlier versions.

Important: The changelog information emitted will be used as the content of the GitHub release notes.

Verify the component's dependencies

This check verifies the component's dependencies are correct in the `metadata.json`:

- Verify that the dependencies in the `metadata.json` file are complete. This means that the sources of all external functions/classes used within the module are listed in the `metadata.json`.
- Verify that the version constraints for each dependency are correct.

Important: Beginning with `simp-rake-helpers-4.1.0`, the RPM dependencies for a component will determined from its `metadata.json` file, and if present, the component's entry in the `simp-core/build/rpm/dependencies.yaml`.

Verify RPMs can be created

This check verifies that an RPM can be generated for this module from `simp-core`:

1. Clone `simp-core`

```
git clone https://github.com/simp/simp-core.git
```

2. Update the URL for the component under test `Puppetfile.tracking`, if needed

```
cd simp-core
vi Puppetfile.tracking
```

3. Build RPM

```
bundle update
bundle exec rake deps:checkout
bundle exec rake pkg:single[iptables]
```

Note: This command will build the RPM for the OS of the server on which it was executed.

Verify unit tests pass

This check verifies that the component's unit tests have succeeded in GitHub Actions.

- Navigate to the project's GitHub Actions results page and verify the tests for the development branch to be tagged and released have passed. For our project, this page is https://github.com/simp/pupmod-simp-iptables/actions

Important: If the tests in GitHub Actions fail, you **must** fix them before proceeding. The automated release procedures will only succeed, if the unit tests succeed.

Verify acceptance tests pass

This check verifies that the component's acceptance tests have succeeded:

- Run the `beaker:suites` rake task with and without FIPS enabled

```
BEAKER_fips=yes bundle exec rake beaker:suites
bundle exec rake beaker:suites
```

Note:

- For older projects that have not been updated to use test suites, you may have to run the `acceptance` rake task, instead.
- If the GitLab instance for the project is current (it is sync'd every 3 hours), you can look at the latest acceptance test results run by GitLab. For our project, the results will be at https://gitlab.com/simp/pupmod-simp-iptables/-/pipelines.

Verify interoperability with last SIMP release

This check verifies that this version of the component interoperates with the last full SIMP release. For many components, the best automated way of doing this is by running the simp-core and pupmod-simp-simp acceptance tests, as these tests provide extensive, multi-component, integration tests.

1. Checkout the simp-core project for the last SIMP release. For this discussion, we will assume it is 6.0.0-1.

   ```
   git clone https://github.com/simp/simp-core.git
   cd simp-core
   git fetch -t origin
   git checkout tags/6.0.0-1  # can use a ref spec in lieu of a tag
   ```

2. Create a Puppetfile.tracking file that is a copy of the Puppetfile.stable file for which this component version and any newer dependencies this version itself requires have been updated.

3. Run the default simp-core acceptance tests

   ```
   bundle update
   bundle exec rake beaker:suites
   ```

4. Checkout the version of pupmod-simp-simp corresponding to the last simp-core release

   ```
   bundle exec rake deps:checkout
   cd src/puppet/modules/pupmod-simp-simp
   ```

5. Create a .fixtures.yml file that overlays the contents of the Puppetfile.stable file 3 directories above, with this component version and any newer dependencies this version itself requires.

   Note: Currently, there are prototype utilities to generate the .fixtures.yml file for you. When these utilities are released, this documentation will be (thankfully) updated.

6. Run the acceptance tests with and without FIPS mode enabled

   ```
   bundle update

   BEAKER_fips=yes bundle exec rake beaker:suites
   bundle exec rake beaker:suites

   BEAKER_fips=yes bundle exec rake beaker:suites[base_apps]
   bundle exec rake beaker:suites[base_apps]

   BEAKER_fips=yes bundle exec rake beaker:suites[no_simp_server]
   bundle exec rake beaker:suites[no_simp_server]

   BEAKER_fips=yes bundle exec rake beaker:suites[scenario_one_shot]
   bundle exec rake beaker:suites[scenario_one_shot]

   BEAKER_fips=yes bundle exec rake beaker:suites[scenario_poss]
   bundle exec rake beaker:suites[scenario_poss]

   BEAKER_fips=yes bundle exec rake beaker:suites[scenario_remote_access]
   bundle exec rake beaker:suites[scenario_remote_access]
   ```

Verify the component RPM upgrade succeeds

This check verifies that the RPM for this component can be used to upgrade the last full SIMP release. For CentOS 7, do the following:

1. Bring up a CentOS server that was booted from the last SIMP ISO release and for which `simp config` and `simp bootstrap` has been run.

   Note: If the VirtualBox for the last SIMP ISO was created by the simp-packer project, you can simply setup the appropriate VirtualBox network for that box and then bring up that bootstrapped image with `vagrant up`.

2. Copy the component RPM generated from the above RPM verification check to the server and install with yum. For example,

   ```
   sudo yum install pupmod-simp-iptables-6.0.3-1.noarch.rpm
   ```

   Note:

   - If the component requires updated dependencies, those RPMs will have to be built and installed at the same time.
   - Puppet agent runs will be tested in *Verify the component works in an actual SIMP system*

Verify the component yields valid SIMP ISOs

This check verifies that with this component, valid SIMP ISOs for CentOS 7 can be built. An ISO is considered to be valid when a SIMP server can be booted from it, configured via `simp config`, and then bootstrapped via `simp bootstrap`.

For CentOS 7:

1. Login to a machine that has Docker installed and the `docker` service running.

   Important: In our development environment, the version of Docker that is available with CentOS works best.

2. Checkout the `simp-core` project for the last SIMP release. For this example, we will use `6.5.0-1`.

   ```
   git clone https://github.com/simp/simp-core.git
   cd simp-core
   git fetch -t origin
   git checkout tags/6.5.0-1
   ```

3. Create a `Puppetfile.tracking` file that contains the contents of `Puppetfile.stable` in which the URLs for the component and any of its updated dependencies have been updated to reference the versions under test.

4. Populate `simp-core/ISO` directory with CentOS distribution ISOs

   ```
   mkdir ISO
   <curl the distribution ISOs into the ISO directory>
   ```

5. Build ISO(s). For example,

```
bundle update
SIMP_BUILD_docs=no \
SIMP_BUILD_verbose=yes \
SIMP_PKG_verbose=yes \
bundle exec rake beaker:suites[rpm_docker]
```

Important:

1. The most reliable way to build each ISO is from a clean checkout of `simp-core`.

6. Use simp-packer to verify that the SIMP ISO can be bootstrapped, when booted with the default options.

Verify the component works in an actual SIMP system

This is the *Eat Our Own Dogfood* soak test. It verifies that the component operates as expected on a typical SIMP system. For this verification, we install the component via R10K in the SIMP development environment:

1. Create a branch in the control repo for the version under test.
2. Use the module-portion of the `Puppetfile.tracking` from the ISO-build-verification step as the Puppetfile for the environment.
3. Deploy the environment using r10k. In this example our environment will be `simp_6_1_0_test`

   ```
   /opt/puppetlabs/puppet/bin/r10k deploy environment simp_6_1_0_test -p
   ```

4. Assign nodes to the test environment using the installed ENC
5. Verify `puppet agent -t` successfully runs for each node assigned to the test environment.

Release to GitHub and Deploy to PuppetForge

Each SIMP component is configured to automatically create a GitHub release and push the release to PuppetForge, when an annotated tag is created for the GitHub project **and** the CI tests for the annotated tag push succeed.

To create the releases from an annotated tag:

1. Clone the component repository and checkout the development branch to be tagged

   ```
   git clone git@github.com:simp/pupmod-simp-iptables.git
   cd pupmod-simp-iptables
   git checkout master # this step isn't needed for master branch
   ```

2. Generate the changelog content

   ```
   bundle update
   bundle exec rake pkg:create_tag_changelog > foo
   ```

3. Create the annotated tag. In this example the content of 'foo' is:

   ```
   Release of 6.0.2

   * Wed May 24 2017 Brandon Riden <brandon.riden@onyxpoint.com> - 6.0.2-0
   ```

(continues on next page)

(continued from previous page)

```
  - Added a workaround for Puppet 4.10 type issues
    - There was a bug in Puppet where all lookup() Hash keys were being converted
      into Strings even if they were another data type
    - This is fixed in Puppet > 4.10.2 but this patch will remain for backwards
      compatibility
  - Update puppet dependency in metadata.json
  - Remove OBE pe dependency in metadata.json
```

```
git tag -a 6.0.2 -F foo
git push origin 6.0.2
```

Note: For markdown-style changelogs, you will need to specify `--cleanup=whitespace` so comment headers are not stripped.

4. Verify GitHub Actions completes successfully

Important: If any of the required builds for the project fail, for example due to intermittent connectivity problems, you can complete the release process by manually restarting the failed build on the GitHub Actions page for that build.

5. Verify release exists on GitHub. This release will have been created by `simp-auto`.

Other Puppet Module Release Procedures

This section will describe the release procedures for Puppet module projects for which SIMP is not the owner. In these procedures, the SIMP Team will release RPMs of these projects, using SIMP forks to which **no SIMP modifications have been made**. The purpose of these forks is simply to retain a backup copy of the official repositories in the case that the upstream repositories are compromised or taken down unexpectedly.

Note: We **highly** recommend that you keep copies of all external repositories as a clone in your internal systems if you are deploying via `r10k` or Code Manager.

Important: If the owner has made unreleased modifications to the project that are essential to SIMP *OR* the SIMP Team has an outstanding pull request for the project with essential changes, the SIMP Team must take ownership of this version of the Puppet module to release it. This is the only way for SIMP to release the modified version to PuppetForge.

Note: You can identify whether a Puppet module is owned by SIMP, by examining the outer-most `name` entry in the module's `metadata.json` file. The value for the `name` key will be of the form <*owner*>-<*module name*>.

Pre-Release Checklist

For each project, the verification required is to ensure the version desired has already been released to GitHub and PuppetForge by the project owner and has been used for testing SIMP components in unit (rspec), acceptance (beaker), and SIMP ISO validation (packer) tests:

1. Verify the version required has an official GitHub release.
2. Verify the version required has been released to PuppetForge.
3. Verify the `.fixtures.yml` and `metadata.json` for SIMP components that depend upon the component match the version being released.
4. Verify the `Puppetfile.tracking` file of the `simp-core` project match the version being released.

Build Signed RPM and Deploy to SIMP Download Server

If a New RPM Needs to be Built

1. Build the RPM for the component that you wish to publish

```
git clone simp-core
git checkout master # or an appropriate branch
bundle update
bundle exec rake pkg:single[MODULE_NAME or PATH]
```

Note: If, for some reason, the above does not work, you can go into the target component and run `rake pkg:rpm`

The output will be in the `dist` directory of the targeted artifact

1. Pass the RPM over to an authorized signing team member who will sign it using `rpm --resign`

Push to SIMP Download Server

Currently the authorized signing team member that signs the RPM will use `scp` to push the file to the proper folder on the SIMP Download Server. The file will then be accessible by following the appropriate path at https://download.simp-project.com.

Ruby Gem Release Procedures

This section will describe the release procedures for the SIMP Ruby gems used to build and test SIMP components. The relevant components include

- `rubygem-simp-beaker-helpers`
- `rubygem-simp-build-helpers`
- `rubygem-simp-rake-helpers`
- `rubygem-simp-rspec-puppet-facts`

Note: `rubygem-simp-cli` is covered in *Other ISO-Related Project Release Procedures*.

For demonstration purposes, we will be using the `simp-rake-helpers` project, which uses the `master` branch as its development branch.

Pre-Release Checklist

The bulk of the work to release a component is to verify that the component is ready for release. Below is the list of verifications that must be executed before proceeding with the release. If any of these steps fail, the problem identified must be fixed before you can proceed with the tag and release steps.

- *Verify a release is warranted*
- *Verify the CHANGELOG*
- *Verify the component's dependencies*
- *Verify a Ruby gem can be created*
- *Verify unit tests pass*
- *Verify acceptance tests pass*
- *Verify gem works for SIMP projects*

Verify a release is warranted

This check verifies a new release is warranted and the version has been properly updated:

1. Clone the component repository and checkout the development branch to be tagged

```
git clone https://github.com/simp/rubygem-simp-rake-helpers.git
cd rubygem-simp-rake-helpers
git checkout master # this step isn't needed for master branch
```

2. Manually compare the development branch with the last release tag. (The existing rake task `pkg:compare_latest_tag` won't necessarily work here.)

```
git fetch -t origin

# manually figure out which is last tag

git diff tags/<last release tag> --name-only

# manually verify mission-impacting changes have been
# made (i.e., changes that warrant a release) and the
# version has been updated in the CHANGELOG.md, version.rb
# and/or TBD file.
```

Verify the changelog

This check verifies that the changelog information is available and can be extracted

- Manually inspect the appropriate file (e.g., CHANGELOG.md) (The existing rake task `pkg:create_tag_changelog` won't necessarily work here.)

Verify the component's dependencies

This check verifies that the component's dependencies are correct in the `Gemfile` and `<component>.gemspec`

- Manually inspect the `Gemfile` and `<component>.gemspec` to look for inconsistencies or missing runtime dependencies.

Verify a Ruby gem can be created

This check verifies that a Ruby gem can be created for this component:

```
bundle update
bundle exec rake pkg:gem
```

Verify unit tests pass

This check verifies that the component's unit tests have succeeded in GitHub Actions.

- Navigate to the project's GitHub Actions results page and verify the tests for the development branch to be tagged and released have passed. For our project, this page is https://github.com/simp/rubygem-simp-rake-helpers/actions

  Important: If the tests in GitHub Actions fail, you **must** fix them before proceeding. The automated release procedures will only succeed, if the unit tests succeed.

Verify acceptance tests pass

This check verifies that the component's acceptance tests have succeeded:

- Run the appropriate acceptance test rake task, if it exists. For this project, `rake -T` shows that `rake acceptance` is the appropriate task

  ```
  bundle exec rake acceptance
  ```

  Note: If the GitLab instance for the project is configured and current (it is sync'd every 3 hours), you can look at the latest acceptance test results run by GitLab. For our project, the results would be at https://gitlab.com/simp/rubygem-simp-rake-helpers/-/pipelines.

Verify gem works for SIMP projects

This check verifies that SIMP components can use this gem for build and test tasks.

1. Install the gem you just built, locally.

```
rvm all do gem install dist/simp-rake-helpers-4.0.1.gem
```

2. Download the latest versions of most of the SIMP components using the `simp-core` project.

```
git clone https://github.com/simp/simp-core.git
cd simp-core
bundle update
bundle exec rake deps:checkout
```

3. If the major version number for the gem has increased, for the following projects, update their Gemfiles to permit the newer version

   - All projects in `src/assets/`
   - The `simp-doc` project in `src/doc`
   - All SIMP-owned projects in `src/puppet/modules/`

4. In each project listed above, execute the rake tasks affected by the changes. In this case, we assume the `spec` task was affected.

```
bundle update
bundle exec rake spec
```

Release To GitHub and Deploy to RubyGems.org

At this time, most but not all of the SIMP Ruby build and test gems are configured to automatically release from an annotated tag. So, this section will describe both the automated steps and the manual steps required to release SIMP Ruby gems to GitHub and RubyGems.org.

Common Release Steps

Most of the SIMP Ruby gems are configured to automatically create a GitHub release and push the release to RubyGems.org, when an annotated tag is created for the GitHub project **and** the GitHub Actions tests for the annotated tag push succeed.

To create the releases from an annotated tag:

1. Clone the component repository and checkout the development branch to be tagged

```
git clone git@github.com:simp/rubygem-simp-rake-helpers.git
cd rubygem-simp-rake-helpers
git checkout BRANCH # this step isn't needed for master branch
```

2. Manually generate the changelog content in a file.

   - The first line should be blank.
   - The second line should be 'Release of x.y.z'
   - The third line should be blank

   - The remaining lines should contain the list of changes.

3. Create the annotated tag. In this example the content of `foo` is:

   > Release of 4.0.1
   >
   > - Reverted the bundler pinning since it was causing too many issues on CI systems

   ```
   git tag -a 4.0.1 -F foo
   git push origin 4.0.1
   ```

   Note: For markdown-style changelogs, you will need to specify `--cleanup=whitespace` so comment headers are not stripped.

4. Verify GitHub Actions completes successfully

   Important: If any of the required GitHub Actions builds for the project fail, for example due to intermittent connectivity problems, you can complete the release process by manually restarting the failed build on the GitHub Actions page for that build.

Automated Release Steps

This section applies to gems that have a `release` GitHub Action configured.

1. Verify release exists on GitHub. This release will have been created by `simp-auto`.
2. Verify release exists on RubyGems.org.

Manual Release Steps

For any gem that has not been configured to automatically release from an annotated tag, you must manually release the gem.

To create the releases from an annotated tag:

1. Create a release of the annotated tag on GitHub.
   - Select the `Draft a new release` button.
   - Click in the `Tag version` box and then select the annotated release version from the drop-down menu.
   - Select the `Publish release` button. The changelog information for the annotated tag will automatically appear as the release notes.
2. Publish to RubyGems.org

   Note: This requires that you have a GPG key in place that allows you to publish to RubyGems.org and is valid for the Gem that you are attempting to push.

   - Run `gem build simp-rake-helpers.gemspec`
   - Run `gem push simp-rake-helpers-4.0.1.gem`

Other ISO-Related Project Release Procedures

This section will describe the release procedures we use for the miscellaneous, non-Puppet-module components required to build a SIMP ISO. The relevant components include

- `rubygem-simp-cli`
- `simp-adapter`
- `simp-doc`
- `simp-environment`
- `simp-gpgkeys`
- `simp-rsync`
- `simp-utils`

For demonstration purposes, we will be using the `simp-adapter` project, which uses the `master` branch as its development branch.

Pre-Release Checklist

The bulk of the work to release each component is to verify that the component is ready for release. Below is the list of verifications that must be executed before proceeding with the release. If any of these steps fail, the problem identified must be fixed before you can proceed with the tag and release steps.

- *Verify a release is warranted*
- *Verify the changelog*
- *Verify RPMs can be created*
- *Verify unit tests pass*
- *Verify acceptance tests pass*
- *Verify the component RPM upgrade succeeds*
- *Verify the component yields valid SIMP ISOs*

Verify a release is warranted

The check verifies a new release is warranted and the version has been properly updated.

1. Clone the component repository and checkout the development branch to be tagged

```
git clone https://github.com/simp/simp-adapter.git
cd simp-adapter
git checkout master # this step isn't needed for master branch
```

2. Run the `pkg:compare_latest_tag` rake task

```
bundle update
bundle exec rake pkg:compare_latest_tag
```

Important: If this check indicates no new tag is required, there is no reason to continue with the release procedures.

Verify the changelog

This check verifies that the CHANGELOG information can be properly extracted:

1. Run the `pkg:create_tag_changelog` rake task

```
bundle exec rake pkg:create_tag_changelog
```

2. Manually verify the changelog information is emitted and complete.

   - It should begin with `Release of x.y.z` and then be followed by one or more comment blocks. For example,

   ```
   Release of 6.0.3

   * Thu Aug 10 2017 Nick Markowski <nmarkowski@keywcorp.com> - 6.0.3-0
     - Updated iptables::listen::tcp_stateful example to pass valid
       Iptables::DestPort types to dports
   ```

   - It should be understandable.
   - It should be free from typos.
   - Any parsing error messages emitted should *only* be for changelog entries for earlier versions.

Verify RPMs can be created

This check verifies that an RPM can be generated for this module from `simp-core`:

1. Clone `simp-core`

```
git clone https://github.com/simp/simp-core.git``
```

2. Update the URL for the component under test `Puppetfile.tracking`, if needed

```
cd simp-core
vi Puppetfile.tracking
```

3. Build RPM

```
bundle update
bundle exec rake deps:checkout
bundle exec rake pkg:single[adapter]
```

Note: This command will build the RPM for the OS of the server on which it was executed.

Verify unit tests pass

This check verifies that the component's unit tests have succeeded.

- Navigate to the project's GitHub Actions results page and verify the tests for the development branch to be tagged and released have passed. For our project, this page is https://github.com/simp/simp-adapter/actions

Important: If the tests are failing, you **must** fix them before proceeding. The automated release procedures will only succeed if the unit tests succeed.

Verify acceptance tests pass

This check verifies that the component's acceptance tests have succeeded:

- Run the appropriate acceptance test rake task, if it exists. For this project, `rake beaker:suites` is the appropriate task

  ```
  bundle exec rake beaker:suites
  ```

Note: If the GitLab instance for the project is configured and current (it is sync'd every 3 hours), you can look at the latest acceptance test results run by GitLab. For our project, the results would be at https://gitlab.com/simp/simp-adapter/-/pipelines.

Verify the component RPM upgrade succeeds

This check verifies that the RPM for this component can be used to upgrade the last full SIMP release. For CentOS 7, do the following:

1. Bring up a CentOS server that was booted from the appropriate SIMP ISO and for which `simp config` and `simp bootstrap` has been run.

   Note: If the VirtualBox for the last SIMP ISO was created by the simp-packer project, you can simply setup the appropriate VirtualBox network for that box and then bring up that bootstrapped image with `vagrant up`.

2. Copy the component RPM generated from the above RPM verification step to the server and install with yum. For example,

   ```
   sudo yum install simp-adapter-0.0.3-0.el7.noarch.rpm
   ```

   Note: If the component requires updated dependencies, those RPMs will have to be built and installed at the same time.

3. Verify the `puppet agent` runs succeed on the *Puppet Server*
   - login as root
   - execute `puppet agent -t`
4. Execute any other verifications unique to the component

Verify the component yields valid SIMP ISOs

This check verifies that with this component, valid SIMP ISOs for CentOS 7 can be built. An ISO is considered to be valid when a SIMP server can be booted from it, configured via `simp config`, and then bootstrapped via `simp bootstrap`. For CentOS 7:

1. Login to a machine that has Docker installed and the `docker` service running.

   Important: In our development environment, the version of Docker that is available with CentOS works best.

2. Checkout the `simp-core` project for the last SIMP release. For this discussion, we will assume it is `6.0.0-1`.

   ```
   git clone https://github.com/simp/simp-core.git``
   cd simp-core
   git fetch -t origin
   git checkout tags/6.0.0-1
   ```

3. Create a `Puppetfile.tracking` file that contains the contents of `Puppetfile.stable` in which the URLs for the component and any of its updated dependencies have been updated to reference the versions under test.

4. Populate `simp-core/ISO` directory with CentOS 7 distribution ISOs

   ```
   mkdir ISO
   cp /net/ISO/Distribution_ISOs/CentOS-7-x86_64-1708.iso ISO/
   ```

5. Build each ISO for CentOS. For example,

   ```
   bundle update
   SIMP_BUILD_docs=no \
   SIMP_BUILD_verbose=yes \
   SIMP_PKG_verbose=yes \
   bundle exec rake beaker:suites[rpm_docker]
   ```

   Important:

   1. The most reliable way to build each ISO is from a clean checkout of `simp-core`.

6. Use simp-packer to verify the SIMP ISO can be bootstrapped, when booted with the default options.

Release to GitHub

At this time all of the ISO-related SIMP projects are configured to automatically release to GitHub, but as we transition to the new deploy strategy, there may be issues with old deploy keys. This section will describe both the automated steps and the manual corrective steps required to release the other ISO-related projects to GitHub.

Automated Release Steps

All SIMP ISO-related project are configured to automatically create a GitHub release, when an annotated tag is created for the GitHub project.

To create the a release from an annotated tag:

1. Clone the component repository and checkout the development branch to be tagged

```
git clone git@github.com:simp/rubygem-simp-cli.git
cd rubygem-simp-cli
git checkout master # this step isn't needed for master branch
```

2. Generate the changelog content

```
bundle update
bundle exec rake pkg:create_tag_changelog > foo
```

3. Create the annotated tag. In this example the content of 'foo' is:

```
Release of 4.0.4

* Mon Oct 16 2017 Trevor Vaughan <tvaughan@onyxpoint.com> - 4.0.4
  - Fix intermittent failure in RPM builds due to missing rubygems
```

```
git tag -a 4.0.4 -F foo
git push origin 4.0.4
```

4. Verify release exists on GitHub. This release will have been created by `simp-auto`.

Fixing a Failed Deploy to GitHub

If the deploy stage for a project fails to release to GitHub because of deploy key issues, the following manual steps can be followed to manually correct the issue:

1. Create a release of the annotated tag on GitHub.
   - Select the `Draft a new release` button.
   - Click in the `Tag version` box and then select the annotated release version from the drop-down menu.
   - Select the `Publish release` button. The changelog information for the annotated tag will automatically appear as the release notes.

Build Signed RPM and Deploy to Download Server

Build Signed RPM and Deploy to SIMP Download Server

If a New RPM Needs to be Built

1. Build the RPM for the component that you wish to publish

```
git clone simp-core
git checkout master # or an appropriate branch
bundle update
bundle exec rake pkg:single[MODULE_NAME or PATH]
```

Note: If, for some reason, the above does not work, you can go into the target component and run `rake pkg:rpm`

The output will be in the `dist` directory of the targeted artifact

1. Pass the RPM over to an authorized signing team member who will sign it using `rpm --resign`

Push to SIMP Download Server

Currently the authorized signing team member that signs the RPM will use `scp` to push the file to the proper folder on the SIMP Download Server. The file will then be accessible by following the appropriate path at https://download.simp-project.com.

ISO Release Procedures

This section will describe the partially-automated, release procedures we use for SIMP ISOs.

Pre-Release Checklist

The bulk of the work to release both *EL* 7 and *EL* 8 versions of a SIMP ISO is to verify that each ISO is ready for release. Below is the list of verifications that must be executed **for each ISO**, before proceeding with the release of that ISO. If any of these steps fail, the problem identified must be fixed before you can proceed with the tag and release steps.

- *Verify SIMP server R10K install*

Update Policy Evaluation Response Reports

Since one of the main goals of SIMP is to assist with compliance of various standards, we should add a response to the latest security scans that we use.

Given that most scanners are only one view on the world and often are not flexible enough to meet all possible solutions to a given policy, it is expected that there will be explanations of both false positives as well as helpful material on why the SIMP framework is compliant for the benefit of our users.

These scans should be added, as applicable, to the *Evaluation Artifacts* section of the documentation.

Verify RPMs are available on the Download Server

This check is to verify that all artifacts used to create the ISO exist as signed RPMs in the appropriate location under https://download.simp-project.com. This will include:

- SIMP-owned Puppet modules
- Other Puppet modules
- SIMP utility RPMs (`rubygem-simp-cli`, `simp-adapter`, `simp-utils`, etc.)
- `simp-doc`
- SIMP application RPMs
- External vendor application RPMs
- OS RPMs

For nearly all the projects listed in `Puppetfile.tracking`, you can verify that the RPMs for those projects exist by executing the `pkg:check_published` Rake command:

1. Checkout the `simp-core` project.

   ```
   git clone https://github.com/simp/simp-core.git
   cd simp-core
   ```

2. Verify the `Puppetfile.tracking` file contains the component tags for the release.
3. Execute the `pkg:check_published` Rake command

   ```
   bundle exec rake pkg:check_published > check_published.out
   ```

4. Examine the `check_published.out` content to verify that, except for the `simp-doc` project, no projects lists `RPM Publish Required:` or `Git Release Tag Required:`. What you should see are lines such as:

   ```
   ...
   Found Existing Remote RPM: pupmod-simp-stunnel-6.1.0-0.noarch.rpm
   Found Existing Remote RPM: pupmod-simp-sudo-5.0.3-0.noarch.rpm
   ...
   ```

   Important: If you see a message like `Warning:  Unable to generate build-specific YUM cache`, your results are invalid, as connection to the specified repo under https://download.simp-project.com failed.

#. Manually verify the appropriate `simp-doc` RPM exists in the appropriate location under https://download.simp-project.com.

For the external vendor RPMs

- Upload all vendor RPMs to the proper vendor repository in the relevant version repository under https://download.simp-project.com. Any existing RPMs will not be overwritten.
  - `package_cloud push simp-project/VERSION_Dependencies/el/OS_MAJOR_VERSION /path/to/packages`

> **Warning: DO NOT** push any Core Operating System RPMs up to the Download Server, those should be retrieved from official vendor sources.

Verify a valid Puppetfile exists

This check is to verify that that `Puppetfile.tracking` file for the `simp-core` project is complete and accurate:

- It includes all the SIMP-owned Puppet modules, other Puppet modules that are dependencies of SIMP-owned Puppet modules, and utilities to configure the SIMP system when installed from ISO.
- The URL for each artifact corresponds to the tag for its signed, published RPM.

Verify the Changelog.rst

This check is to verify that the `simp-core` Changelog.rst has been updated:

- Manually inspect

Verify the dependencies.yaml

This check is to verify that `simp-core/build/rpm/dependencies.yaml` contains the correct adjustments to the RPM dependencies, obsoletes, requires, and/or release fields for any of the components listed in the `Puppetfile.tracking` file.

Manually inspect the file to verify there are entries for

- All non-SIMP Puppet modules that have more dependencies listed in their `metadata.json` files than are actually required on a SIMP system. Each entry must list all the relevant dependencies in a `:requires` element.
- Any component that has changed name (e.g. `pupmod-saz-timezone` changing to `pupmod-simp-timezone`). Each entry must list the package and version obsoleted in an `:obsoletes` element.
- Any component for which for which the RPM release field must be specified (e.g. a component with a RPM-packaging-only change). Each entry must list a `:requires` element.

Verify the simp-core RPMs can be created

This check verifies that an RPM can be generated for `simp-core`:

```
git clone https://github.com/simp/simp-core.git
cd simp-core/src/assets/simp
bundle update
bundle exec rake pkg:rpm
```

Note: This command will build the RPM for the OS of the server on which it was executed.

Verify simp-core tests pass

This check verifies that the `simp-core` unit and acceptance test have succeeded.

To verify that the `simp-core` unit tests have succeeded, examine the test results in GitHub Actions.

- Navigate to the project's GitHub Actions results page and verify the tests for the development branch to be tagged and released have passed. For our project, this page is https://github.com/simp/simp-core/actions

  Important: If the tests in GitHub Actions fail, you **must** fix them before proceeding. The automated release procedures will only succeed, if the tests succeed.

To verify that the `simp-core` acceptance tests have succeeded

1. Checkout the `simp-core` project for the last SIMP release.

   ```
   git clone https://github.com/simp/simp-core.git
   cd simp-core
   ```

2. Run the default `simp-core` acceptance tests

   ```
   bundle update
   bundle exec rake beaker:suites
   ```

Note: If the GitLab instance for `simp-core` is current (it is sync'd every 3 hours), you can look at the latest acceptance test results run by GitLab, instead. The results will be at https://gitlab.com/simp/simp-core/-/pipelines.

Verify ISOs can be created

This check verifies that SIMP ISOs for CentOS 7 can be built from the local `simp-core` clone and RPMs pushed to the SIMP Download Server.

For CentOS 7:

1. Login to a machine that has Docker installed and the `docker` service running.

   Important: In our development environment, the version of Docker that is available with CentOS works best.

2. Checkout the `simp-core` project for the last SIMP release.

```
git clone https://github.com/simp/simp-core.git
cd simp-core
```

3. Populate `simp-core/ISO` directory with CentOS distribution ISOs

```
mkdir ISO
<curl the latest EL8 and EL7 ISOs>
```

4. Build each ISO for supported CentOS versions. For example,

```
bundle update
SIMP_BUILD_docs=no \
SIMP_BUILD_verbose=yes \
SIMP_PKG_verbose=yes \
bundle exec rake beaker:suites[rpm_docker]
```

Important:

1. The most reliable way to build each ISO is from a clean checkout of `simp-core`.

5. Verify none of the RPMs in the ISO that SIMP would have generated are signed by the SIMP development GPG key. For example, for a CentOS 7 build:

```
cd build/distributions/CentOS/7/x86_64/SIMP/RPMS/noarch

# The 7da6f216 key ID may change as the SIMP signing keys get updated over time
# The output of this command should be *EMPTY*
rpm -q --qf '%{NAME}-%{VERSION}-%{RELEASE} %{SIGPGP:pgpsig} %{SIGGPG:pgpsig}\n' -p * | grep -v 7da6f216
```

Verify SIMP ISO boot options work

This hefty check verifies that a server booted from the SIMP ISO can be bootstrapped for the 'simp' scenario and following boot options:

- Using default boot option
- Using disk encryption boot option
- Using FIPS disabled boot option
- Using disk encryption and FIPS disabled boot options
- Using simp-prompt option
- Using simp-prompt and disk encryption boot options
- Using simp-prompt and FIPS disabled boot options
- Using simp-prompt, disk encryption, and FIPS disabled boot options
- Using linux-min boot option
- Using linux-min and disk encryption boot options
- Using linux-min and FIPS disabled boot options

- Using linux-min, disk encryption, and FIPS disabled boot options

For the default boot options with/without encryption and the FIPS disabled boot option with/without encryption test cases, the simp-packer project is the easiest way to verify a SIMP VM can be booted from the ISO and bootstrapped. Otherwise, the check has to be done manually:

- Boot a VM with the SIMP ISO
- Select the appropriate boot options
- Once the server boots, login to the server as root
- Bootstrap the system

```
simp config
simp bootstrap
reboot
```

- Login to the server as root and run `puppet agent -t` until the results are stable
- Verify the server is/is not in FIPS mode by inspecting */proc/sys/crypto/fips_enabled*
- Verify the appropriate disk is/is not encrypted by executing

```
blkid
```

- Verify the appropriate disk partitioning

```
lsblk
```

Important: For the `linux-min` test cases, the only verification required is verification that the server boots up.

Verify component interoperability

This check verifies, with `simp-core` and `pupmod-simp-simp` acceptance tests, that this aggregation of module versions interoperate. (These tests provide extensive, cross-component, integration tests.)

Note: If `simp-core` and `pupmod-simp-simp` acceptance tests have effectively already passed on one of our continuous integration platforms (e.g., in GitLab), you can skip this painful step. However, you must be sure that the tests were run with the correct component versions.

1. Checkout the `simp-core` project.

```
git clone https://github.com/simp/simp-core.git
cd simp-core
```

2. Verify the `Puppetfile.tracking` file contains the component tags for the release.
3. Run the default `simp-core` acceptance tests

```
bundle update
bundle exec rake beaker:suites
```

4. Checkout the version of `pupmod-simp-simp` corresponding to this `simp-core` version

```
bundle exec rake deps:checkout
cd src/puppet/modules/pupmod-simp-simp
```

5. Create a `.fixtures.yml` file that sets the version of each dependency to the version contained in the `Puppetfile.tracking` file for this ISO release.

6. Run **all** the functioning acceptance tests with and without FIPS mode enabled

```
bundle update

BEAKER_fips=yes bundle exec rake beaker:suites
bundle exec rake beaker:suites

BEAKER_fips=yes bundle exec rake beaker:suites[base_apps]
bundle exec rake beaker:suites[base_apps]

BEAKER_fips=yes bundle exec rake beaker:suites[no_simp_server]
bundle exec rake beaker:suites[no_simp_server]

BEAKER_fips=yes bundle exec rake beaker:suites[scenario_one_shot]
bundle exec rake beaker:suites[scenario_one_shot]

BEAKER_fips=yes bundle exec rake beaker:suites[scenario_poss]
bundle exec rake beaker:suites[scenario_poss]

BEAKER_fips=yes bundle exec rake beaker:suites[scenario_remote_access]
bundle exec rake beaker:suites[scenario_remote_access]
```

Verify otherwise untested capabilities

This check verifies that all other major capabilities (not otherwise tested in acceptance/simp-packer tests) do function as advertised:

Note: In order to speed time to market, the goal is to automate as many of these manual tests as possible!

- A SIMP client can be PXE booted using the kickstart files from the SIMP ISO
- A SIMP client can use the SIMP server for DNS
- A 'simp_lite' client operates with a SIMP server
  - login operations (PAM, LDAP, local user)
  - NFS operations (home directory)
  - logging operations (rsyslog)
  - auditing operations
- A 'simp_poss' client operates with a SIMP server
- The SIMP server can be converted from FIPS enabled to FIPS disabled mode.
- The SIMP server can be converted from Selinux enforcing to Selinux permissive.
- The SIMP server can be converted from Selinux permissive to Selinux enforcing.

- A local user with sudo privileges can be created and login to both the SIMP server and client on CentOS 7.
- An LDAP user in the `administrators` group can login to both the SIMP server and client on CentOS 7.
- Local and LDAP users can change their passwords on both the SIMP server and client on CentOS 7.
- The Rsyslog rules from `simp_rsyslog`, `syslog` and SIMP application modules (`aide`, `tlog`, etc.) result in application log messages being written to the correct local and remote log files.

  Note: Although the `simp_rsyslog` and `syslog` modules have excellent acceptance tests, neither has a full-system test to verify integration with actual log producers. The tests for these modules use `logger` as a mock message sender.

- The compliance map reports for a full SIMP system are accurate.
  - No reports list non-compliant configuration that is really a parameter mismatches. (Parameter tested differs from parameter that should have been tested; value tested differs from actual values allowed, etc.)
  - SIMP server and SIMP client reports are generated.
- `simp-utils` executables that are not tested otherwise work as advertised
  - `unpack_dvd`
  - `gen_ldap_update`
  - `updaterepos`
- The *SIMP HOWTO Guides* are still correct.

Verify SIMP server RPM install

This check verifies that CentOS 7 SIMP servers can be installed using the set of RPMs contained in the SIMP ISOs The verification steps largely follow the details in *Installing SIMP From A Repository*. All RPMs except the `simp-core` RPM should be able to be pulled from the appropriate location under https://download.simp-project.com.

Verify SIMP server RPM upgrade

This check verifies that the set of RPMs in the SIMP ISO can upgrade the last full SIMP release.

1. Bring up a CentOS server that was booted from the appropriate SIMP ISO and for which `simp config` and `simp bootstrap` has been run.

   Note: If the VirtualBox for the last SIMP ISO was created by the simp-packer project, you can simply setup the appropriate VirtualBox network for that box and then bring up that bootstrapped image with `vagrant up`.

2. Copy the SIMP and system RPMs packaged in the SIMP ISO to the server and install with yum.
   - FIXME Should put RPMs into appropriate updates repos, run something like the following

```
cd <updates dir>
createrepo .
chown -R root.apache ./*
find . -type f -exec chmod 640 {} \;
find . -type d -exec chmod 750 {} \;
```

(continues on next page)

(continued from previous page)

```
yum clean all;
yum make cache
yum update
```

3. Verify `puppet agent -t` runs cleanly
4. Verify no custom content is removed by the upgrade (e.g., `environments/production/modules/site/manifests`, content in `environments/production/data`)

Verify SIMP server R10K install

This check verifies that CentOS 7 SIMP servers can be installed via *r10k*. Since this capability is already automatically tested in a `simp-core` acceptance test, all verification is handled by *Verify simp-core tests pass*.

Release `simp-core` to GitHub and PuppetForge

`simp-core` is configured to automatically create a GitHub release and push the (meta-module) release to PuppetForge, when an annotated tag is created for the GitHub project **and** the GitHub Actions tests for the annotated tag push succeed.

To create the releases from an annotated tag:

1. Clone the component repository and checkout the development branch to be tagged

```
git clone git@github.com:simp/simp-core.git
cd simp-core
git checkout master # this step isn't needed for master branch
```

2. Create the annotated tag for the release. In this example, we are assuming the version is `6.1.0` and we are using the full Changelog.rst content.

```
git tag -a 6.0.2 -F Changelog.rst --cleanup--whitespace
git push origin 6.0.2
```

3. Verify GitHub Actions tests complete successfully

Important: If any of the required GitHub Actions builds for the project fail, for example due to intermittent connectivity problems, you can complete the release process by manually restarting the failed build on the GitHub Actions page for that build.

4. Verify release exists on GitHub. This release will have been created by `simp-auto`.

Build Signed simp-core RPM and Deploy to Download Server

Note: For `simp-core`, the `pkg:single` path will need to be used. Specifically, `src/assets/simp`

Build Signed RPM and Deploy to SIMP Download Server

If a New RPM Needs to be Built

1. Build the RPM for the component that you wish to publish

```
git clone simp-core
git checkout master # or an appropriate branch
bundle update
bundle exec rake pkg:single[MODULE_NAME or PATH]
```

Note: If, for some reason, the above does not work, you can go into the target component and run `rake pkg:rpm`

The output will be in the `dist` directory of the targeted artifact

1. Pass the RPM over to an authorized signing team member who will sign it using `rpm --resign`

Push to SIMP Download Server

Currently the authorized signing team member that signs the RPM will use `scp` to push the file to the proper folder on the SIMP Download Server. The file will then be accessible by following the appropriate path at https://download.simp-project.com.

Build Final ISO and Deploy to simp-project

Building the Final ISO

The `build::auto` Rake task will pull all upstream published RPMs for any repositories that are listed as part of the target distribution `yum_repos` metadata.

The final ISO should be built from published RPMs by running `SIMP_BUILD_docs=yes rake build:auto[<path to ISOs>]` on the **same operating system version for which you are building**.

The `rpm_docker` acceptance test has good working examples of this process.

Important: Validate that no RPMs that were included into the ISO were signed by the generated development GPG key. If they were, then there is a disconnect between the published RPMs and the local component repository versions.

Publishing to simp-project.com

The final ISO should be provided to personnel with upload access to the `simp/ISO` directory of `https://download.simp-project.com` for final delivery.

Notify Mailing List

Upon release of a new version of SIMP, an e-mail should be sent to the following mailing lists:

- simp-announce@googlegroups.com
- simp-users@googlegroups.com
- simp-dev@googlegroups.com

The standard text should be something like the following (feel free to edit with any pertinent information and update all of the links).

The standard subject is: **SIMP {VERSION} Has Been Released!**

Warning: Be 100% sure that all links work prior to sending the message!

Release Candidate/Alpha/Beta

> All,
>
> We've just dropped the {first|second|etc…} Release Candidate for SIMP {VERSION} and would appreciate any feedback that you can provide.
>
> IMPORTANT: Please read the {old version} to {new version} upgrade guide in detail!
>
> The SIMP Download Server have been updated with a new release repository.
>
> You can also download an ISO if you want to experiment with the new features from a fresh system.
>
> Thanks,
>
> {Your Name}

Final Release

> All,
>
> We're pleased to announce the general availability of SIMP {VERSION}!
>
> IMPORTANT: Please read the {old version} to {new version} upgrade guide in detail!
>
> The SIMP Download Server have been updated with a new release repository.
>
> You can also download an ISO if you need a clean installation.
>
> Thanks,
>
> {Your Name}

4.7.4 Documentation

Style Guides

SIMP documentation (simp-doc)

SIMP documentation uses ReStructuredText roles to keep formatting consistent, automatically cross-reference glossary terms, and generate external hyperlinks to web resources.

When available, prefer using a relevant role from the tables below over inline formatting such as bold *text* or double-backticks (``):

Table 5: Roles already built into Sphinx

Purpose	Example	Code	Notes
Code snippets	ensure => true	:code:`ensure => true`	
CLI commands	**yum update -y**	:command:`yum update -y`	
Files	/etc/simp/version.*x*.*y*	:file:`/etc/simp/version.{x}.{y}`	Use curly braces {foo} to denote variable path elements
Programs	**mcstransd**	:program:`mcstransd`	
Glossary terms	*SIMP*	:term:`SIMP`	Automatically creates internal hyperlink to term in Glossary

Table 6: Custom roles, created for simp-doc

Purpose	Example	Code	Notes
Packages (e.g., RPM, RubyGem)	simp-utils	:package:`simp-utils`	
Puppet Modules	simp/simplib	:pupmod:`simp/simplib`	When text is in org-name or org/name format, will auto-link to Puppet Forge URL
GitHub repos	simp/simp-doc	:github:`simp/simp-doc`	Auto-links to GitHub repo URL
Jira issues	SIMP-8464	:jira:`SIMP-8464`	Auto-links to (SIMP project) Jira issue URL

Table 7: Inline formatting

Purpose	Example	Code	Notes
Literal text	`keyword`	``` ``keyword`` ```	ReST's basic markup for fixed-width text. Only use this if there isn't an appropriate role.
Internal hyperlinks	*Changelogs*	``:ref:`changelogs` ``	`:ref:` crosslinks are actually built-in roles, but we'll group them here
External hyperlinks	SIMP website	`` `SIMP website <https://www.simp-project.com>`_ ``	External hyperlinks are decorated with a special icon

SIMP Puppet modules

Documentation for Puppet modules should follow the Puppet Strings Style Guide.

SIMP git repositories

SIMP project git repositories should contain a `README.md` file at their top level.

4.7.5 Building a SIMP ISO

The following guides provide information on building a SIMP ISO for yourself.

You may want to do this for various reasons including:

- Building a *RHEL* ISO
- Customizing the kickstart
- Adding your own packages
- Updating packages from the defaults

> **Warning:** Prior to starting any build method, you will need to ensure that you follow the instructions in *Environment Preparation*.

Environment Preparation

Getting Started

> **Warning:** Use a **non-root** user for building SIMP!

Ensure Sufficient Entropy

The SIMP build generates various keys and does quite a bit of package signing. As such, your system must be able to keep its entropy pool full at all times. If you check `/proc/sys/kernel/random/entropy_avail` and it shows a number below **1024**, then you should either make sure that `rngd` is running and pointed to a hardware source (preferred) or install and use **haveged**.

```
sudo yum install haveged
sudo systemctl start haveged
sudo systemctl enable haveged
```

Setup a Container Management System

There are a lot of different methods for running containers (we like **podman**) so you will need to find one that works for your operating system.

While most of our development occurs on SIMP-managed systems, we have had luck using WSL in Windows and Docker in OS X.

Examples in the documentation will use **podman** and **buildah**.

Ensure Sufficient Space

Building the ISO takes quite a bit of space so make sure you have at least 30G of free space available.

If you use a container, you need to make sure that wherever you are hosting your container has enough space available.

If your default location does not have enough space, you may need to change your `graphroot` in `$HOME/.config/containers/storage.conf`.

Install Puppet Bolt

The latest build system uses puppetlabs/bolt to spin up **pulp** containers for repository mirroring.

This was done to work with modular repositories in EL8+.

Setup Your Build Container

SIMP needs to be built using the **exact same distribution** that the ISO targets.

Use an Existing Image

For the fastest start, use a published image if one is available that matches your system.

```
podman pull simpproject/simp_build_centos8
```

Build a Fresh Container

If a matching image is not available (or updates are available that have yet to be published) then use the build-related Dockerfiles in the simp/simp-core project under the `build/Dockerfiles` subdirectory.

The following provides an example of how to build and start the image:

```
git clone https://github.com/simp/simp-core
cd simp-core/build/Dockerfiles
buildah build --layers -f SIMP_EL8_Build.dockerfile -t simp_build_centos8
podman run -id --name simp_build_centos8 simp_build_centos8
podman exec -it simp_build_centos8 bash
su - build_user
cd simp-core
git tag
git checkout <tag of simp-core that you wish to build>
```

Mirroring Packages

Getting Started

Follow the instructions in *Environment Preparation* if you have not done that yet.

Note: The following steps must be done on the container hosting system unless you have nested container capabilities in your build container.

Install Bolt

See the documentation at puppetlabs/bolt to install **bolt** on your system.

Create the Pulp Cloning System

Using puppetlabs/bolt and simp/bolt-pulp3, set up the pulp container.

```
git clone https://github.com/simp/bolt-pulp3
cd bolt-pulp3
rvm use system # You can skip this if not using RVM
/opt/puppetlabs/bolt/bin/bolt module install --force
/opt/puppetlabs/bolt/bin/gem install --user -g gem.deps.rb
/opt/puppetlabs/bolt/bin/bolt plan run pulp3::in_one_container
# Wait a few minutes for the service to get fully started
```

Clone the Packages

This uses a configuration file from the target version of SIMP that you are going to build.

You will either need to download the file from simp/simp-core or clone the repository and checkout the git tag that you wish to build.

For example, if you are going to build CentOS 8 for SIMP 6.6.0:

```
curl -O https://raw.githubusercontent.com/simp/simp-core/6.6.0-1/build/distributions/
↪CentOS/8/x86_64/bolt_pulp3_config.yaml
./slim-pulp-repo-copy.rb -f bolt_pulp3_config.yaml
```

or:

Note: You can change the `url` keys in the packages YAML file if you wish to use alternate servers.

Troubleshooting

Repo Clone Hangs

Occasionally, there are issues with the upstream repos that require the pulp container to be restarted.

The most common one is when the clone process hangs and continually prints that it is in status `Waiting`.

If this occurs:

```
podman stop pulp
podman start pulp
# Wait a few minutes for the service to get fully started
./slim-pulp-repo-copy.rb -f build/6.6.0/CentOS/8/x86_64/repo_packages.yaml
```

Updated YAML file

If you have updated your YAML file, you will need to flush and re-clone the repositories so that you don't pull stale packages.

To do this:

```
rm -rf output
rm -rf _download_data # Only if the reposync has been run
/opt/puppetlabs/bolt/bin/bolt plan run pulp3::in_one_container::destroy
/opt/puppetlabs/bolt/bin/bolt plan run pulp3::in_one_container
# Wait a few minutes for the service to get fully started
./slim-pulp-repo-copy.rb -f build/6.6.0/CentOS/8/x86_64/repo_packages.yaml
```

If the slim-pulp-repo-copy.rb Fails

If the slim-pulp-repo-copy.rb fails for any reason, the pulp container could be in an unknown state. To ensure that nothing from the failed run will cause issues with future runs, do the following:

This will ensure that the container is removed and recreated in a fresh state.

Copy the Repo Contents

Once the clone process has completed, you need to copy the cloned packages out of the repository for use in the rest of the build process.

To do so:

```
cd output
./_slim_repos.*reposync.sh
```

Copy the Packages Into the Build Container

You now need to copy the files into the build container for use during the build process.

```
podman cp _download_path/ simp_build_centos8:/tmp
podman exec -it simp_build_centos8 chown -R build_user:build_user /tmp/_download_path
```

(Optional) Cleanup

When you are done with the pulp container, you may want to reclaim the space on your system.

To do this, run the following:

```
cd bolt-pulp3
/opt/puppetlabs/bolt/bin/bolt plan run pulp3::in_one_container::destroy
```

Building SIMP From Source

Getting Started

Please have your environment prepared as specified by *Environment Preparation* and have the necessary packages mirrored in accordance with *Mirroring Packages* before continuing.

All of the following items assume that you are running as `build_user` in the `simp_build_centos8` container.

Download the CentOS/RedHat installation media:

- Refer to `release_mappings.yaml` to determine the distribution ISO compatible with the version of SIMP you want to build. `release_mappings.yaml` is maintained the simp/simp-core repository in the `build/distributions/<distribution>/<release>/<arch>` directory.

Updating release_mappings.yaml

It is very possible the versions outlined in the *release_mappings.yaml* are out of date, as such you may need to download a newer version of the installation media and create a new section inside of the file. You can either modify one of the existing release versions or create an entirely new one to house the information for the new iso. We recommend simply copying one of the existing releases, renaming it, and modifying the iso section as necessary.

The size of the iso can be obtained with:

The checksum can be obtained with:

After obtaining the information necessary, ensure that the iso name is correct, modify the build command to have the correct iso name and release version, and ensure that the OS version is correct.

Note: The build process is handled by simp/rubygem-simp-rake-helpers. If you discover issues, that is where you will want to look for answers.

Generating The ISO

Clone simp-core:

```
git clone https://github.com/simp/simp-core
cd simp-core
```

Check out your desired branch of SIMP:

- To check out a stable SIMP release, check out a tag (*Recommended*):

```
git checkout tags/6.6.0-1
```

- To check out an unstable SIMP release, check out the latest `master`:

```
git checkout master
```

Run `bundle` to make sure that all of the build tools and dependencies are installed and up to date:

```
bundle install
```

Make an `ISO` directory, and copy in the CentOS/RHEL installation media:

```
mkdir ISO
cp </path/to/dvd*.iso> ISO
```

Next, you need to prepare your local packages for inclusion in the final ISO. This example builds off of CentOS 8, but all distributions work the same way.

Change to the target distribution directory:

```
cd build/distributions/CentOS/8/x86_64
```

You will see two directories `DVD` and `yum_data`.

Boot and Kickstart Customization (Optional)

The `DVD` directory holds information that is used to provide ISO boot options in both *UEFI* and *BIOS* boot modes. It also contains a `ks` directory that is used as the automated *kickstart* for hands-off provisioning of the initial server.

Package Customization

The `yum_data` directory is where you will modify the settings to include your own packages in the ISO.

Change to the `yum_data` directory:

```
cd yum_data
```

You will now see a `reposync` directory. Any YUM repository placed into this directory will be copied onto the final ISO using the following rules:

1. If the `reposync` directory has the exact same name as a directory already on the ISO
    - Remove the ISO directory and copy in the `reposync` directory
2. Otherwise

- Add the `reposync` directory as a subdirectory of `SimpRepos`

For example, if the original ISO has the following directory structure:

```
/BaseOS
/AppStream
```

And the `reposync` directory contains the following repositories:

```
/BaseOS
/appstream
/SimpRepos/puppet
```

The resulting ISO will contain the following:

```
/BaseOS (the reposync version)
/AppStream (the ISO version)
/SimpRepos/appstream (case matters)
/SimpRepos/puppet
```

Add the Repositories from Pulp

You can now add the repositories that you mirrored in *Mirroring Packages* to the `reposync` directory.

Note: Check the kickstart files in the `DVD` directory to see what repositories will be used by default.

At this point, the `reposync/SimpRepos` directory may contain both a `puppet` and `puppet6` directory. If you wish to use `puppet` version 6 by default, move the `puppet` directory to `puppet7` and rename `puppet6` to `puppet`.

```
mv reposync/puppet reposync/puppet7
mv reposync/puppet6 reposync/puppet
```

Modifying the Repositories (Optional)

At this point, you may perform the following actions:

1. Add your own additional repositories
2. Update repositories that do **not** have groups or modules present
   - Repos with groups will contain a `*-comps.xml` file in the `repodata` directory
   - Repos with modules will contain a `*-modules.yaml` file in the `repodata` directory

Verify the Upstream Vendor Repositories

The SIMP component build process first attempts to use an upstream YUM repository to pull down a matching build artifact. Failing that, a local copy is built for packaging. This copy is **authoritative** and will override anything from `reposync`.

The optional `repos/` directory under `yum_data/` holds a selection of repositories that will be used for fetching upstream SIMP RPMs from the official sources. These files should be regular *YUM* repository configuration files.

Once downloaded, non-SIMP files will be housed in a `packages` directory and the `packages.yaml` file will be updated to reflect the download source for auditing purposes. If you need to re-download the files, simply erase the `packages` and `packages.yaml` files.

Build the ISO

You are now ready to build the ISO!

To do so, run the following, substituting `6.6` with the expected build version from `release_mappings.yaml`:

```
cd </path/to>/simp-core
SIMP_BUILD_prompt=yes \
SIMP_BUILD_reposync_only=yes \
SIMP_BUILD_distro=CentOS,8Stream,x86_64 \
bundle exec rake build:auto[$PWD/ISO,<release_version>] #The release_version is the desired release number defined in release_mappings.yaml
```

Once the process completes, you should have a bootable SIMP ISO, in: `build/distributions/<OS>/<rel>/<arch>/SIMP_ISO/`

You can download it as follows (using CentOS 8 as an example):

```
podman cp simp_build_centos8:/home/build_user/simp-core/build/distributions/CentOS/8/x86_64/SIMP_ISO .
```

Rebuilding the ISO

If the ISO build fails for any reason or you simply want to re-build the ISO because you've made a change you need to run the following commands to clean up the previous run:

After running these commands the ISO build can be safely re-run.

Other Build Directories of Note

The following directories exist at the same level as `SIMP_ISO/` and may be of use:

- `DVD_Overlay`
  - The SIMP product RPMs and artifacts as a `tar` file. This is extracted into the ISO after all other modifications have occurred.
- `SIMP_ISO_STAGING`
  - The 'staging' directory for the ISO. This is essentially the final ISO in 'unpacked' form and a useful place to look if you think something is missing or incorrect.

After You Build

You may have noticed that a development GPG key has been generated for the build.

This key is only valid for one week from generation and has been specifically generated for packages compiled specifically for your ISO build. If all of your packages were downloaded via Pulp, then there should be no packages on your ISO that need the development GPG key.

Doing this allows you to have a validly signed set of RPMs while reducing the risk that you will have invalid RPMs distributed around your infrastructure.

Note: If you need to build and sign your RPMs with your own key, you can certainly do so using the `rpm --resign` command.

The new development key will be placed at the root of your ISO and will be called `RPM-GPG-KEY-SIMP_dev`. This key can be added to your clients, or served via a web server, if you need to install from a centralized *yum* repository.

Please see the Red Hat Guide to Configuring Yum and Yum Repositories for additional information.

4.7.6 Testing on FIPS Systems

If you're running a system that requires compliance with *NIST 800-53* or *NIST 800-171*, you may find that having your system *FIPS*-enabled is causing your workflow to simply fall apart.

Since we try to eat our own dog food, we try to develop on SIMP as much is as practical and have the following advice that works at the time of writing this document.

Many of the tools that we use are getting better, and we have been diligent about filing bugs with projects that fail to meet the requirements set out by FIPS or which simply crash due to being run on a FIPS enabled system. We do understand that not all operations require FIPS security but, unfortunately, the underlying software simply can't tell whether an algorithm is being used for security or convenience.

Bundler

Bundler is probably the first hurdle that you will encounter.

There is an original bug that we filed that has a fix released in Bundler 1.14.X. While this has worked for us (and is what we recommend), apparently there were some issues with the patch and it was reverted. Likewise, a new bug has been filed that is tracking current progress and we have faith that the team will get it fully fixed in the near future.

To pin your runs to a FIPS-compatible Bundler, you will need to both install a non-crashing version, as well as ensure that you always use that version during your runs.

A simple method for doing this would be to do the following:

```
gem install bundler -v 1.14.6
alias bundle='bundle _1.14.6_'
```

RSpec-Puppet

There is one change that you need to make to your `spec/spec_helper.rb` file to ensure that `rspec` does not attempt to use MD5 checksums.

You simply need to add something like the following to your `RSpec.configure` section:

```
RSpec.configure do |c|
  c.before(:each) do
    Puppet[:digest_algorithm] = 'sha256'
  end
end
```

4.7.7 Useful Resources

- GitHub Documentation
- SIMP Project Status Links
  - Open Changes
  - Changes that need attention
  - Pending Changes
  - Merged Changes

4.8 SIMP Security Concepts

Contents:

4.8.1 Introduction

This manual describes the security concepts of the SIMP system. The system was originally designed to meet a specific set of technical security controls using industry best practices and has been modified recently to meet as many of the security controls provided by the National Institute of Standards and Technology's (*NIST*) special publication 800-53 as possible.

This manual outlines three categories of security:

- **Technical Architecture**: discusses the technical approaches to securing the system
- **Operational Security**: discusses the security of SIMP in an operational setting
- **Information System Management**: discusses how SIMP helps achieve security in terms of system management

A brief discussion of how the SIMP system helps achieve categories of controls is provided; additional technical details regarding each control can be found in the *SIMP Security Control Mapping*.

When possible, the NIST security control identifier will be found at the end of a concept to provide the reader with a reference to the specific control that is being discussed. The identifier is written as [AB-X(Y)], where A is the control family, X is the control section, and Y is the control enhancement.

Note: At present, this document will **not** be mapped to any additional standards since there are available mappings of the 800-53 to various other security frameworks.

If you believe that we are missing anything in particular, please file a bug!

4.8.2 Technical Security

This chapter contains SIMP security concepts that are related to the technical security controls described in *NIST 800-53*.

Identification and Authentication

This section addresses the identification and authentication of users and devices.

User Identification and Authentication

Identification and authentication of system and service users can occur at either the *Operating System* level or globally in the SIMP architecture. While local accounts and groups can be created manually, the SIMP team suggests adding users using the native Puppet user and group types. System users can authenticate their access using Secure Shell (SSH) keys or passwords. For more centralized control, identify and authenticate users by using the Lightweight Directory Access Protocol (*LDAP*). [*IA-2 : IDENTIFICATION AND AUTHENTICATION (ORGANIZATIONAL USERS)*]

The SIMP team recommends using *LDAP* as the primary source for user management and provides a functional default OpenLDAP configuration for this purpose. *LDAP* and Pluggable Authentication Modules (*PAM*) work together closely and, with the default SIMP configuration, the PAM settings are enforced on top of the LDAP settings for two layers of control. Due to this partnership, items such as account lockouts may need to be reset on both the local system and the LDAP server. If the suggested settings in the SIMP-provided default LDAP Directory Interchange Formats (*LDIF*) are not used, implementations must ensure that security is maintained through manual procedures. Use of group accounts for users is strongly discouraged. System services may need to have accounts, but all of these should be managed by Puppet using the user and group native types. [*IA-2 (5) : GROUP AUTHENTICATION*].

Device Identification and Authentication

Devices are identified by a Media Access Control (*MAC*) address prior to receiving an *IP* address via the Dynamic Host Configuration Protocol (*DHCP*). In the default SIMP architecture, *IP* addresses are fixed mappings to their associated *MAC* address (i.e., not assigned dynamically). There is no authentication for the binding of *MAC* addresses to *IP* addresses due to the nature of the *DHCP* protocol.

Device authentication occurs through the mapping of the MAC to the IP through the internally controlled DHCP and the mapping of the IP to the host name through the internally controlled Domain Name System (DNS) service for each individual Puppet client. After kickstart, each client system generates an internal cryptographic identifier and communicates that information with the Puppet server to be approved by an administrator at a later time. All further communication between the Puppet server and the clients over the Puppet protocol is encrypted subsequently and authenticated with this identifier. Automatic approval can be set up in tightly controlled environments; however, this option is not suggested for open environments. [*IA-3 : DEVICE IDENTIFICATION AND AUTHENTICATION*, *IA-3 (3) : DYNAMIC ADDRESS ALLOCATION*]

Identifier Management

Managing user identifiers (also known as user names) involves administrative procedures that are unique for each implementation. Disabling unused local accounts is the only control that SIMP can enforce technologically. In this case, if an account has an expired password that has not been changed 35 days after expiration, the account will be disabled. If a user does not have a password (e.g., he or she only authenticates with SSH keys), then there is no inherent technological mechanism for enforcement due to the nature of the software. [*IA-4e.*]

Authenticator Management

Authenticators for users are passwords and/or *SSH* keys; the management of each is implementation specific. SSH keys do not expire; therefore, implementations must provide a procedure for removing invalid keys. Removing public keys from LDAP is one practical solution.

When using passwords, local and LDAP passwords provided for users should be set to change at first login. This is the default in the SIMP-provided LDIFs. Once a user attempts to change a password, the settings in PAM and LDAP enforce complexity requirements.

For the default password complexity rules see the *What is the Password Complexity for SIMP?* FAQ.

[*IA-5 : AUTHENTICATOR MANAGEMENT*, *IA-5 (1) : PASSWORD-BASED AUTHENTICATION*, *IA-5 (4) : AUTOMATED SUPPORT FOR PASSWORD STRENGTH DETERMINATION*]

Password aging and history is enforced through a combination of *PAM* and *LDAP*. By default, the previous **24** passwords cannot be reused.

[*IA-5 (1)(e)*]

There are a number of default passwords in SIMP that are required for installation. Each implementation requires the user to change the default passwords and protect the new passwords. In addition, there are embedded passwords within the SIMP system that are used due to a lack of software-supported alternatives.

Please see the *SIMP User Guide* for additional information.

Access Control

This section describes the various levels of access control, including account management, access enforcement, information flow enforcement, separation of duties, least privilege, session controls, permitted actions without identification and authentication, security attributes, and remote access.

Account Management

Account management procedures should be created and maintained for each implementation of SIMP. The procedures should include the information listed in *NIST 800-53* control *AC-2 : ACCOUNT MANAGEMENT*. SIMP has the mechanisms in place to enforce most account management policies. The mechanisms for account management have several default settings including:

- Central account management using OpenLDAP. [*AC-2 (1) : AUTOMATED SYSTEM ACCOUNT MANAGEMENT*]
- Password expiration.
  - Local accounts expire 35 days after password expiration. [*AC-2 (3) : DISABLE INACTIVE ACCOUNTS*]
  - *LDAP* accounts do not expire automatically due to inactivity; implementations should audit LDAP accounts regularly.

- Auditing of administrative actions to capture local account creation and modifications to *LDAP* accounts is done via the `/var/log/slapd_audit.log` file and `/var/log/audit/audit.log` for local accounts. [*AC-2 (4) : AUTOMATED AUDIT ACTIONS*]
- Shell sessions timeout after **15 minutes** of inactivity. [*AC-2 (5) : INACTIVITY LOGOUT*]
  - This can be circumvented by running a command that opens an endless pipe such as `/bin/cat`. However, this command cannot be enforced more heavily due to the high likelihood of breaking system applications. If the optional gnome module is used, the GNOME screen saver will lock the screen after **15 minutes** of inactivity.
- Assignment of users into groups locally or centrally via LDAP. [*AC-2 (7) : ROLE-BASED SCHEMES*]
  - By default, SIMP will have an administrators groups that has the ability to run `sudo su - root`. Implementations should further define administrators or user groups and limit them with the Puppet `sudo` class.

Access Enforcement

SIMP uses the implementation of Discretionary Access Control (*DAC*) that is native to Linux. Specific file permissions have been assigned based on published security guidance for Red Hat, CentOS, and UNIX.

Default permissions on files created by users are enforced with user file access mask settings (using the `umask` command) that allow only the owner to read and write to the file. Implementations may further extend the access control in UNIX by restricting access to application files or using the file Access Control List (*ACL*) commands `getfacl` and `setacl`. Users of SIMP should not change file permissions on operating system files as it may decrease the overall security of the system. If a group needs access to a particular file or directory, use the `setfacl` command to allow the necessary access without lessening the permissions on the system. [*AC-3 : ACCESS ENFORCEMENT*]

Information Flow Enforcement

IPTables on each SIMP system is controlled by the IPTables Puppet module. When developing a new module, the IPTables rules needed for an application should be included with the module by calling the appropriate methods from the IPTables module. The end result should be a running IPTables rule set that includes the default SIMP rules and any rules needed for applications. The default communications allowed are included in *Default Server Ports* and *Default Client Ports*. [*AC-4 : INFORMATION FLOW ENFORCEMENT*]

Default Server Ports

Application	Direction	Protocol	Transport	Ports	Comment
Puppet	Localhost	HTTP	TCP	8140	The port upon which the *Puppet Server* listens for client connections via Apache
Puppet CA	In	HTTPS	TCP	8141	This is used to ensure that Apache can verify all certificates from external systems properly prior to allowing access to Puppet.
Apache/YUM	In	HTTP	TCP	443	This is used for YUM and is encrypted using https.
DHCPD	In	DHCP/BOOTP	TCP/UDP	546,547	DHCP pooling is disabled by default and should only be used if the implementation requires the use of this protocol.
TFTP	In	TFTP	TCP/UDP	69	This is used for kickstart. It could also be used to update network devices. TFTP does not support encryption.
rsyslog	Out	syslog	TCP/UDP	6514	This is encrypted when communicating with a SIMP syslog server (not installed by default).
named	In/Out	DNS	TCP/UDP	53	Inbound connections happen to the locally managed hosts. Outbound connections happen to other domains per the normal operations of DNS.
NTPD	Out	NTP	TCP/UDP	123	Only connects to an external time source by default.
SSHD	In	SSH	TCP	22	SSH is always allowed from any source IP by default.
stunnel	In	TLS	TCP	8730	Stunnel is a protected connection for rsyncing configuration files to Puppet clients.
rsync	Localhost	RSYNC	TCP	873	This accepts connections to the localhost and forwards through Stunnel.
LDAP	In	LDAP	TCP	389	Connections are protected by bi-directional, authenticated encryption.
LDAPS	In	LDAPS	TCP	636	Used for LDAP over SSL.

Default Client Ports

Application	Direction	Protocol	Transport	Ports	Comment
Puppet	Out	HTTPS	TCP	8140	Communications to the Puppet server.
rsyslog	Out	syslog	TCP/UDP	6514	This is encrypted when communicating with a SIMP syslog server.
DNS Client	Out	DNS	TCP/UDP	53	Normal name resolution.
NTPD	Out	NTP	TCP/UDP	123	Only connects to an external time source by default.
SSHD	In	SSH	TCP	22	SSH is allowed from any source IP by default.
LDAP	Out	LDAP	TCP	389	Connections are protected by bi-directional authenticated encryption.

Separation of Duties

SIMP enforces separation of duties using account groups. Groups are created with each implementation to separate roles or duties properly. The SIMP team recommends that this management be done using the **posixGroup** object in *LDAP* for full *OS* support. [*AC-5 : SEPARATION OF DUTIES*]

Least Privilege

SIMP does not allow `root` to directly *SSH* into a system. Direct access to the `root` user must occur via a console (or at a virtual instance of the physical console) to log on. Otherwise, users must log on as themselves and perform privileged commands using `sudo`. [*AC-6 : LEAST PRIVILEGE*]

NIST 800-53 least privilege security controls give people access to objects only as needed. SIMP provides only the needed software, services, and ports to allow the system to be functional and scalable. The system then relies on a given implementation to perform proper account management and user role assignments. [*AC-6 : LEAST PRIVILEGE*]

Session Controls

SIMP provides a number of security features for sessions. These features include:

- Accounts are locked after **five** invalid log on attempts over a **15 minute** period. The account is then locked for **15 minutes**. No administrator action is required to unlock an account. [*AC-7 : UNSUCCESSFUL LOGON ATTEMPTS*]
- System banners are presented to a user both before and after logging on. The default banner should be customized for each implementation. [*AC-8 : SYSTEM USE NOTIFICATION*]
- After a successful log on, the date, time, and source of the last log on is presented to the user. The number of failed log on attempts since the last log on is also provided. [*AC-9 : PREVIOUS LOGON (ACCESS) NOTIFICATION* and *AC-9 (1) : UNSUCCESSFUL LOGONS*]
- A limit of **10** concurrent SSH sessions are allowed per user. This can be further limited if an implementation decides it is set too high. Given the way SSH is used in most operational settings, this default value is reasonable. [*AC-10 : CONCURRENT SESSION CONTROL*]
- Session lock only applies if the `windowmanager::gnome` module is used. Sessions lock automatically after **15 minutes** of inactivity. Users must authenticate their access with valid credentials to reestablish a session. [*AC-11 : SESSION LOCK*]

Permitted Actions Without Identification and Authentication

SIMP has a number of applications that do not require both identification and authentication. These services are listed below along with an explanation of why these aspects are not required. Implementations should include any additional services that do require identification and/or authentication. [*AC-14 : PERMITTED ACTIONS WITHOUT IDENTIFICATION OR AUTHENTICATION*]

Service/Application	Rationale
TFTP	TFTP is a simple file transfer application that, in the SIMP environment, does not allow for writing to the files being accessed. This application is primarily used to support the Preboot Execution Environment (PXE) booting of hosts and the updating of network devices. There is no option to authenticate systems at this level by protocol design. TFTP is limited to a user's local subnet using IPtables and is enforced additionally with TCPWrappers.
DHCP	By default, system IP addresses are not pooled, but are rather statically assigned to a client, which is identified by the MAC address. DHCP is limited to the local subnet.
Apache/YUM	RPMs are stored in a directory for systems to use for both kickstart and package updating. Sensitive information should never be stored here. Apache/YUM is limited to the local subnet.
DNS	The DNS protocol does not require identification nor authentication. DNS is limited to the local subnet.

Table: Actions Without Identification and Authentication

Security Attributes

SELinux is fully enforcing, in targeted mode, in SIMP. SELinux is an implementation of *Mandatory Access Control*. It can be set to enforcing mode during the SIMP configuration or turned on at a later time. All of the SIMP packaged modules have been designed to work with SELinux set to enforcing. [*AC-16 : SECURITY ATTRIBUTES*]

Remote Access

Remote access in SIMP is performed over *SSH*, specifically using the OpenSSH software. OpenSSH provides both confidentiality and integrity of remote access sessions. The SSH *IPTables* rules allow connections from any host. SSH relies on other Linux mechanisms to provide identification and authentication of a user. As discussed in the auditing section, user actions are audited with the audit daemon (`auditd`) and *Tlog*. [*AC-17 : REMOTE ACCESS*]

Systems and Communications Protection

The following sections provide information regarding application partitioning, shared resources, and various levels of protection for systems and communications.

User and Administration Application Separation (Application Partitioning)

SIMP can be used in a variety of ways. The most common is a platform for hosting other services or applications. In that case, there are only administrative users present. Users with accounts will be considered as a type of privileged user.

SIMP can also be used as a platform for workstations or general users performing non-administrative activities. In both cases, general users with accounts on an individual host are allowed access to the host using the `pam::access` module, so long as they have an account on the target host. No user may perform or have access to administrative functions unless given `sudo` privileges via Puppet.

Shared Resources

There are several layers of access control that prevent the unauthorized sharing of resources in SIMP. Account access, operating system *DAC* settings, and the use of *PKI* collectively prevent resources from being shared in ways that were not intended. [*SC-4 : INFORMATION IN SHARED RESOURCES*]

Denial of Service Protection

SIMP has limited ability to prevent or limit the effects of Denial of Service (*DoS*) attacks. The primary measures in place are to drop improperly formatted packets using *IPTables* and Kernel configurations such as *SYN cookies*. [*SC-5 : DENIAL OF SERVICE PROTECTION*]

Boundary Protection

SIMP does not provide boundary protection. [SC-7]

Transmission Security

SIMP traffic is protected with protocols that provide confidentiality and integrity of data while in transit. The tables in *Information Flow Enforcement* describe the protocols used to encrypt traffic and explain the protocols that cannot be protected at the transmission layer. *SSH*, and *TLS* all provide data transmission integrity and confidentiality. The software that controls them on on the operating system are OpenSSH, OpenSSL, and GnuTLS. The SIMP team takes industry guidance into consideration when configuring these services. For example, the list the cryptographic ciphers available is limited to the highest ciphers that SIMP needs. All others are disabled.

[*SC-8 : TRANSMISSION CONFIDENTIALITY AND INTEGRITY*, *SC-9 : TRANSMISSION CONFIDENTIALITY*, *SC-23 : SESSION AUTHENTICITY*, *SC-7 : BOUNDARY PROTECTION*]

Single User Mode

SIMP systems have a password requirement for single user mode. In the event maintenance needs to be performed at a system console, users must be in possession of the `root` password before they can be authenticated. Bootloader passwords are also set to prevent unauthorized modifications to boot parameters. [*SC-24 : FAIL IN KNOWN STATE*]

PKI and Cryptography

SIMP has two native certificate authorities. The first is known as *Fake CA*. A local certificate authority is used to create properly formed server certificates if an implementation does not have other means of obtaining them. Many SIMP services require certificates; therefore, SIMP provides this tool for testing or for situations where other certificates are not available. The second certificate authority, *Puppet CA*, is built into Puppet. Puppet creates, distributes, and manages certificates that are specifically for Puppet.

The *Fake CA* certificates should be replaced with your own hardware-generated certificates if at all possible. The *Puppet CA* may be replaced but please understand all ramifications to the infrastructure before doing so.

More information on the Puppet CA can be found in the product security documentation. [*SC-17 : PUBLIC KEY INFRASTRUCTURE CERTIFICATES*, *SC-13 : CRYPTOGRAPHIC PROTECTION*]

Warning: Fake CA certificates should not be used in an operational setting unless no better options are available.

Mobile Code

SIMP does not use mobile code; however, there are not any particular tools that will prevent its use. [*SC-18 : MOBILE CODE*]

Protection of Information at Rest

SIMP provides the capability to enable Full Disk Encryption (FDE) by default. However, in the interest of automated reboots, the initial **randomly generated** key is baked into the `initrd`. Please see the *Disk Encryption* section of the Installation Guide for details. [*SC-28 : PROTECTION OF INFORMATION AT REST*]

Audit and Accountability

This section discusses the content, storage, and protection of auditable events.

Auditable Events

`Auditd` and `Rsyslog` provide the foundation for SIMP auditing. `Auditd` performs the majority of the security-related events; however, other Linux logs also have security information in them and are captured using `rsyslog`.

The default auditable events for SIMP were developed based on several industry best practices including those from the SCAP Security Guide and several government configuration guides. The suggested rules by those guides were fine-tuned so the audit daemon would not fill logs with useless records or reduce performance. These guides should be referenced for a detailed explanation of why rules are applied. Additional justification can be found in the comments of the SIMP audit rules found in the appendix of this guide. [*AU-2 : AUDIT EVENTS*]

The SIMP development team reviews every release of the major security guides for updated auditable events suggestions. Each of those suggestions is reviewed and applied if deemed applicable. [*AU-2 (3) : REVIEWS AND UPDATES*]

Privileged commands are audited as part of the SIMP auditing configuration. This is accomplished by monitoring `sudo` commands with `auditd`. The output of session interaction for administrative login shells is also collected using *Tlog*. *Tlog* session recordings are sent to *Syslog* for further processing. [*AU-2 (4) : PRIVILEGED FUNCTIONS*]

Content of Audit Records

Audit records capture the following information [*AU-3 : CONTENT OF AUDIT RECORDS*]:

- Date and Time
- UID and GID of the user performing the action
- Command
- Event ID
- Key
- Node Hostname/IP Address
- Login Session ID
- Executable

Audit Storage

Audit logs are stored locally on a separate partition in the /var/log directory. The size of this partition is configurable. Other default audit storage configurations include:

- A syslog log is written when the audit partition has **75MB** free. (This can be changed to e-mail, if an e-mail infrastructure is in place.) [*AU-5a.*, *AU-5 (1) : AUDIT STORAGE CAPACITY*]
- The log file rotates once it reaches **30MB**.

Audit Reduction and Response

SIMP provides a means to capture the proper information for audit records and stores them centrally. Each implementation must decide and document how it reduces, analyzes, and responds to audit events. [*AU-5 : RESPONSE TO AUDIT PROCESSING FAILURES*]

Auditd, like all services in SIMP, is controlled by Puppet. Stopping the service without disabling Puppet means the service will always be started automatically during a Puppet run. The files that control the audit configuration will also revert to their original state if changed manually on a client node. In the event auditd fails, the system will continue to operate. Several security guides have suggested that the system should shut down if auditd fails for any reason. To prevent operational issues, SIMP will not shut down, but will provide an alert via syslog when this happens. [*AU-5 (1) : AUDIT STORAGE CAPACITY*]

Protection of Audit Information

The primary means of protecting the audit logs is through the use of file permissions. Audit records are stored in the /var/log directory and can only be accessed by root. Audit logs are rotated off daily if the implementation has not developed a way of offloading the logs to another location where they can be backed up. Lastly, if the rsyslog::stock::log_server module is implemented, logs are transmitted to the log server over a TLS protected link.

Time Synchronization

Each SIMP client (including the Puppet Server) has ntpd enabled by default. Part of the installation directs the clients to a time server. If no servers are available, the SIMP clients can use the Puppet Server as the central time source. Audit logs receive their time stamp from the local server's system clock; therefore, the SIMP client must be connected to a central time source for timestamps in audit logs to be accurate.

4.8.3 Operational Security

This chapter contains SIMP security concepts that are related to the operational security controls in *NIST 800-53*.

Configuration Management

This section describes the management of various configurations within SIMP.

Baseline Configurations

SIMP baselines include configuration settings and Puppet modules. Currently supported baselines can be found in *SIMP Community Edition (CE) 6.6.0*.

Each configuration item that is managed by a Puppet module has an RPM installed on the *Puppet Server* in the form of `pupmod-name-x.x.x-x`. This process allows for one main SIMP baseline to be maintained and modules to be upgraded easily. An overall SIMP RPM is also installed on the Puppet Server, which denotes the version number of SIMP that is installed. [*CM-2 : BASELINE CONFIGURATION*, *CM-2 (2) : AUTOMATION SUPPORT FOR ACCURACY / CURRENCY*, *CM-2 (3) : RETENTION OF PREVIOUS CONFIGURATIONS*, *CM-6 : CONFIGURATION SETTINGS*]

SIMP installs a minimal set of *RPM* packages, which can be found in the kickstart files on the ISO. RPMs, services, and IPTables rules all use a whitelist stance for allowing access or installation. [*CM-2 (5) : AUTHORIZED SOFTWARE*]

- Additional RPMs must be installed by each implementation.
- Services must be declared explicitly or they will be disabled by Puppet
- IPTables rules must allow a service explicitly.

Managing Configuration Changes

Configuration change approvals are managed by each implementation; SIMP only provides the mechanisms to apply changes on clients. A combination of Puppet, rsync, and *YUM* is used to apply those changes across any number of target Puppet clients. All changes made are audited with `auditd` or are logged to via `syslog`. [*CM-3a.*, *CM-3 (3) : AUTOMATED CHANGE IMPLEMENTATION*]

Linux systems are made up of hundreds of configuration files that can contain numerous settings. SIMP does not make an attempt to manage all of the settings in every file. Instead, critical operating system files or files that need to be controlled centrally are managed. Implementations can manage additional files if they are deemed necessary. [*CM-6 : CONFIGURATION SETTINGS*]

Security Verification and Flaw Remediation

SIMP cannot detect flaws automatically; each implementation is responsible for tracking flaws. However, SIMP provides a way for flaws to be fixed across all clients. One or all of the following can help automate flaw remediation [*CM-6 : CONFIGURATION SETTINGS*, *SI-2 : FLAW REMEDIATION*, *SI-2 (1) : CENTRAL MANAGEMENT*, *SI-2 (4) : AUTOMATED PATCH MANAGEMENT TOOLS*]:

- **Puppet:**
  - Apply a configuration change to files that are managed by Puppet.
- **rsync:**
  - Use this mechanism to deliver a file to a client. This can be used with or without Puppet to synchronize files.
- **YUM:**
  - Update packages nightly with YUM. Placing an updated package in YUM and running a YUM update manually, or allowing time for the cron job to run, will ensure packages on all clients are updated. Otherwise, a cron job will perform a daily update of packages with YUM.

The extent of security verification that is performed currently is based on changes to files that Puppet or the Advanced Intrusion Detection Environment (AIDE) provides. There are also Security Content Automation Protocol (SCAP) profiles available from the SCAP-Security-Guide project that check security configuration settings. [*SI-6 : SECURITY FUNCTION VERIFICATION*]

Malicious Code Protection

For most environments, SIMP will use ClamAV to protect against malicious code. Rsync is used to push out new definitions, which should be updated by the local administrator regularly. SIMP also comes with a `mcafee::uvscan` module that manages an installation of uvscan, if it is preferred. The module can configure `.dat` file updates to occur over `rsync`.

Both the ClamAV and McAfee modules provide a method to run a scan via cron on a customer scheduled basis. [*SI-3 : MALICIOUS CODE PROTECTION*]

SIMP also comes with the `chkrootkit` tool to check for *rootkits*. The tool runs as a cron job and places its output into syslog. [*SI-3 : MALICIOUS CODE PROTECTION*]

Software and Information Integrity

Unauthorized changes to a local client can be detected by Puppet or AIDE (for any file managed by Puppet). In the event that a managed file is changed locally, Puppet will revert the file back to its original state. It is important to note that this is a function of Puppet and is intended to be more of a configuration management feature rather than a security feature. If a Puppet client has been compromised, the Puppet Server may not have the ability to retake control over that client. However, the Puppet Server can configure all other nodes to deny traffic from the compromised node if they are configured by the administrator to do so. There are additional configuration files that are checked by AIDE, which is triggered by a cron job. AIDE logs any detected file changes in syslog. Each implementation may add additional files that are managed by Puppet or watched by AIDE. The AIDE baseline database is updated periodically to handle the installation and updating of system RPMs and reduce false positives. [*SI-7 : SOFTWARE, FIRMWARE, AND INFORMATION INTEGRITY*, *SI-7 (1) : INTEGRITY CHECKS*, *SI-7 (2) : AUTOMATED NOTIFICATIONS OF INTEGRITY VIOLATIONS*, *SI-7 (3) : CENTRALLY-MANAGED INTEGRITY TOOLS*]

Remote Maintenance

Remote maintenance can be performed on SIMP using *SSH*. Local maintenance can be performed at the console or via serial port (if available). SSH sessions are tracked and logged using the security features built into SIMP. Console access requires someone to have access to the physical (or virtual) console along with the `root` password. Auditing of those actions also occurs in accordance with the configured audit policy. It is up to the implementer to decide how to distribute authentication information for remote maintenance. [*MA-4 : NONLOCAL MAINTENANCE*, *MA-4 (1) : AUDITING AND REVIEW*, *MA-6 : TIMELY MAINTENANCE*]

Incident Response

While Puppet is not intended to be a security product primarily, its features help provide security functionality such as dynamic reconfigurations and wide-scale consistent mitigation application. If an implementation chooses, they can leverage Puppet's ability to reconfigure systems as part of incident response. [*IR-1 : INCIDENT RESPONSE POLICY AND PROCEDURES*]

Contingency Planning

SIMP does not provide any direct support for contingency planning. Some of the mechanisms provided by SIMP might be used to support an implementation's contingency plan.

System Backup and Recovery

SIMP does not directly support any particular backup and recovery product. Solutions vary widely, and should be determined as part of an implementation's broader contingency plan. SIMP provides mechanisms that might be used to support backup and recovery procedures.

Administrators seeking FOSS software to implement backup and recovery solutions may be interested in products such as Bacula, BackupPC, duplicity, and scat.

4.8.4 Information System Management

This chapter contains SIMP security concepts that are related to the management security controls in *NIST 800-53*.

Risk Assessment

This section describes the process of identifying risks within a system.

SIMP Self Risk Assessment

Risk can be found in any system. The SIMP team is constantly evaluating the system and the settings to minimize inherit risk. Most risks can be mitigated by processes and procedures at the implementation level. The following table describes the known areas in SIMP. [*RA-1 : RISK ASSESSMENT POLICY AND PROCEDURES*]

Risk	Possible Mitigations
Disabling Puppet: This can cause the clients to be out of sync with the *Puppet Server*.	SIMP attempts to force a break on any locks and restart Puppet on all clients after a time of 4*runinterval (30 minutes by default). Implementations should ensure that further steps have not been taken to disable Puppet and should monitor their logs. Administrators can use the puppetlast command on the Puppet Server to detect servers that have not checked in within a reasonable time period.
Out of Date Patches: SIMP can be built with the RPMs from CentOS or Red Hat. Those RPMs should be assumed out of date at the time a system is initially installed (if using the SIMP DVD).	Implementations should obtain the latest RPMs and apply them in a reasonable manner. All SIMP systems will, by default, attempt to update all packages using YUM nightly. Therefore, having an updated repository will ensure that the systems are updated on a regular basis.
Poor Account Management: SIMP security access control is based on users being created and managed over time. Giving shell access to unnecessary users allows them the opportunity to escalate privileges.	Use the default *LDIF* files and local user modules to ensure that account settings remain restrictive. Ensure the system has policies and procedures in place to manage accounts. Finally, ensure that users are in appropriate groups with limited privileges.

Table: SIMP Risk

Vulnerability Scanning

The SIMP development and security team performs regular vulnerability scanning of the product using commercial and open source tools. Results and mitigations for findings from those tools can be provided upon request. [*CA-2 : SECURITY ASSESSMENTS*, *RA-5 : VULNERABILITY SCANNING*]

Security Assessment and Authorization

Assessment and authorization varies by implementation. Implementations are encouraged to use documentation artifacts provided by the SIMP team to assist with assessment and authorization. [*CA-2 : SECURITY ASSESSMENTS*]

Note: Should users find issues with internal assessments, the SIMP team highly encourages them to submit a bug report using our Bug Tracker

4.8.5 Evaluation Artifacts

This section presents various artifacts that are the result of either public review or internal team evaluation using various compliance scanners.

If you have data that you feel should be added here, please contact us using our *Public Resources* or enter a PR against the simp-doc project.

SCAP Scan Results

Components have different *SCAP* scans that apply to their systems.

The SCAP Security Guide is the general metric by which SIMP systems are measured.

The associated *SCAP* profile should be referenced in each associated document.

The following scan results are available for the various subsystems:

How to Run a SCAN

1. Download the latest SSG Release OVAL ZIP file onto the target system
2. Unzip the downloaded file and `cd` into the directory
3. Make sure that you have the `openscap-scanner` package installed
4. Run `oscap xccdf eval --profile <profile_name> --results ~/scan-output.xml \ --report ~/scan-output.html ssg-<OS>-ds.xml`
   - You can get the list of available profiles by running `oscap info ssg-<OS>-ds.xml`
   - For example, to run the `STIG` profile on `CentOS 7`, you would run the following command:

   ```
   oscap xccdf eval --profile xccdf_org.ssgproject.content_profile_stig-rhel7-disa \
   --results ~/scan-output.xml --report ~/scan-output.html ssg-centos7-ds.xml
   ```

Note: This is an **example report template** to be used when responding to SCAP Scan results.

The following is a short example from a CentOS 7 scan. You will need to adjust all content as appropriate.

TEMPLATE - SSG Scan - EL 7 STIG

- Scan Date: 1/1/1970
- SIMP Version: `6.1.0-RC1`
- SSG Version: `0.1.36`
- Data Stream: `ssh-centos7-ds.xml`
- SIMP Enforcement Profile: `disa_stig`

Terminology:

Finding
Valid issues found by the scanner

Alternate Implementation
Valid implementations per policy that do not match the scan

Exception
Items that will need to be declared as a policy exception for the stated reason

False Positive
Bugs in the scanner that should be reported

Ensure gpgcheck Enabled for Repository Metadata

- Rule ID: `xccdf_org.ssgproject.content_rule_ensure_gpgcheck_repo_metadata`
- Type: **Exception**
- **Recommend SSG Feedback**
  - This should not be a `high` severity if using TLS
  - This opens potential vulnerabilities to all client systems
  - Discussion ongoing on SSG mailing list and ComplianceAsCode/content#2455

Justification

The way that YUM works means that all GPG keys become **trusted** by the entire system. Enabling repository metadata signatures globally means that RPMs will be trusted that come from any system with a trusted GPG key and may allow software to be installed on systems that does not come from the vendor.

Configure Periodic Execution of AIDE

- Rule ID: `xccdf_org.ssgproject.content_rule_aide_periodic_cron_checking`
- Type: **Alternate Implementation**

Notes

We use the Puppet `cron` resource to add the AIDE rule to the `root` user's `crontab`.

System Result

```
# crontab -l

# Puppet Name: aide_schedule                        5 4 *
* 0 /bin/nice -n 19 /usr/sbin/aide -C
```

Build and Test AIDE Database

- Rule ID: `xccdf_org.ssgproject.content_rule_aide_build_database`
- Type: **False Positive**

System Result

```
# ls /var/lib/aide/
aide.db.gz
```

Record Attempts to Alter Logon and Logout Events - faillock

- Rule ID: `xccdf_org.ssgproject.content_rule_audit_rules_login_events_faillock`
- Type: **Finding**

Notes

Marked as a valid finding and tracked as SIMP-4047

Verify and Correct File Permissions with RPM

- Rule ID: xccdf_org.ssgproject.content_rule_rpm_verify_permissions
- Type: Mixed - Mostly False Positives
- Recommend SSG Feedback: Permissions that are obviously more restrictive should not be flagged
- Identifier: V-71849

Most files have more restrictive permissions than provided by the RPMs. Some services, like `openldap`, run as a service specific system user.

The following exceptions are a combination of running the above command on an EL 7 SIMP system:

	File	Puppet log
`SM5....T.c`	/etc/audit/auditd.conf	mode changed '0640' to '0600'
`.M.......c`	/etc/default/nss	mode changed '0644' to '0640'
`SM5....T.c`	/etc/default/useradd	mode changed '0644' to '0600'
`SM5....T.c`	/etc/hosts.allow	mode changed '0644' to '0444'
`SM5....T.`	/etc/init/control-alt-delete.conf	mode changed '0644' to '0640'
`SM5....T.c`	/etc/login.defs	mode changed '0644' to '0640'
`.M....G..c`	/etc/ntp.conf	group changed 'root' to 'ntp' ; mode changed '0644' to '0600'
`.M....G..c`	/etc/openldap/schema/dyngroup.ldif	group changed 'root' to 'ldap' ; mode changed '0444' to '0644'
`.M....G..c`	/etc/openldap/schema/dyngroup.schema	group changed 'root' to 'ldap' ; mode changed '0444' to '0644'
`.M....G..c`	/etc/openldap/schema/inetorgperson.ldif	group changed 'root' to 'ldap' ; mode changed '0444' to '0644'
`.M....G..c`	/etc/openldap/schema/inetorgperson.schema	group changed 'root' to 'ldap' ; mode changed '0444' to '0644'
`.M....G..c`	/etc/openldap/schema/java.ldif	group changed 'root' to 'ldap' ; mode changed '0444' to '0644'
`.M....G..c`	/etc/openldap/schema/java.schema	group changed 'root' to 'ldap' ; mode changed '0444' to '0644'
`.M....G..c`	/etc/openldap/schema/misc.ldif	group changed 'root' to 'ldap' ; mode changed '0444' to '0644'
`.M....G..c`	/etc/openldap/schema/misc.schema	group changed 'root' to 'ldap' ; mode changed '0444' to '0644'
`.M....G..c`	/etc/openldap/schema/nis.ldif	group changed 'root' to 'ldap' ; mode changed '0444' to '0644'
`.M....G..c`	/etc/openldap/schema/nis.schema	group changed 'root' to 'ldap' ; mode changed '0444' to '0644'
`.M....G..c`	/etc/openldap/schema/openldap.ldif	group changed 'root' to 'ldap' ; mode changed '0444' to '0644'
`.M....G..c`	/etc/openldap/schema/openldap.schema	group changed 'root' to 'ldap' ; mode changed '0444' to '0644'

continues on next page

Table 8 – continued from previous page

	File	Puppet log
.M.... G..c	/etc/openldap/schema/pmi.ldif	group changed 'root' to 'ldap' ; mode changed '0444' to '0644'
.M.... G..c	/etc/openldap/schema/pmi.schema	group changed 'root' to 'ldap' ; mode changed '0444' to '0644'
.M.... G..c	/etc/openldap/schema/ppolicy.ldif	group changed 'root' to 'ldap' ; mode changed '0444' to '0644'
.M.... G..c	/etc/openldap/schema/ppolicy.schema	group changed 'root' to 'ldap' ; mode changed '0444' to '0644'
.M.... G..	/etc/puppetlabs/code	group changed 'root' to 'puppet' ; mode changed '0755' to '0750'
.M.... G..	/etc/puppetlabs/code/environments	group changed 'root' to 'puppet' ; mode changed '0755' to '0770' (u=rwx,g=rwx,o-rwx)
.M.... G..	/etc/puppetlabs/code/environments/production	group changed 'root' to 'puppet' ; mode changed '0755' to '0770' (u=rwx,g=rwx,o-rwx)
.M.... G..	/etc/puppetlabs/puppet	group changed 'root' to 'puppet' ; mode changed '0755' to '0750'
SM5... GT.c	/etc/puppetlabs/puppet/puppet.conf	group changed 'root' to 'puppet' ; mode changed '0644' to '0640'
.M.... G..n	/etc/puppetlabs/puppetserver/conf.d	group changed 'root' to 'puppet' ; mode changed '0755' to '0750'
SM5... GT.c	/etc/puppetlabs/puppetserver/conf.d/puppetserver.conf	group changed 'root' to 'puppet' ; mode changed '0644' to '0640'
SM5... GT.c	/etc/puppetlabs/puppetserver/conf.d/web-routes.conf	group changed 'root' to 'puppet' ; mode changed '0644' to '0640'
SM5... GT.c	/etc/puppetlabs/puppetserver/conf.d/webserver.conf	group changed 'root' to 'puppet' ; mode changed '0644' to '0640'
SM5... GT.c	/etc/puppetlabs/puppetserver/logback.xml	group changed 'root' to 'puppet' ; mode changed '0644' to '0640'
SM5... GT.c	/etc/puppetlabs/puppetserver/services.d/ca.cfg	group changed 'root' to 'puppet' ; mode changed '0644' to '0640'
SM5... .T.c	/etc/rsyncd.conf	mode changed '0644' to '0400'
SM5... .T.c	/etc/rsyslog.conf	mode changed '0644' to '0600'
SM5... .T.c	/etc/securetty	mode changed '0600' to '0400'
SM5... .T.c	/etc/security/limits.conf	mode changed '0644' to '0640'
SM5... .T.c	/etc/sysconfig/ktune	mode changed '0777' to '0640'
SM5... .T.c	/etc/sysconfig/ntpd	mode changed '0644' to '0640'
SM5... .T.c	/etc/sysconfig/ntpdate	mode changed '0644' to '0640'
SM5... GT.c	/etc/sysconfig/puppetserver	group changed 'root' to 'puppet' ; mode changed '0644' to '0640'
SM5... .T.c	/etc/sysconfig/rsyslog	mode changed '0644' to '0640'
SM5... .T.c	/etc/sysconfig/slapd	mode changed '0644' to '0640'

continues on next page

Table 8 – continued from previous page

	File	Puppet log
SM5... .T.c	/etc/tuned.conf	mode changed '0777' to '0640'
.M.... ...	/var/lib/ntp	mode changed '0755' to '0750'

InSpec Scan Results

SIMP components may have different *InSpec* scans that apply to their systems.

The associated *InSpec* profile should be referenced in each associated document.

The following scan results are available:

InSpec Subsystem Place Holder

Links will be added here to the various subsystems as they are made available

Verify and Correct File Permissions with RPM

- Rule ID: xccdf_org.ssgproject.content_rule_rpm_verify_permissions
- Type: Mixed - Mostly False Positives
- Recommend SSG Feedback: Permissions that are obviously more restrictive should not be flagged
- Identifier: V-71849

Most files have more restrictive permissions than provided by the RPMs. Some services, like `openldap`, run as a service specific system user.

The following exceptions are a combination of running the above command on an EL 7 SIMP system:

	File	Puppet log
SM5... .T.c	/etc/audit/auditd.conf	mode changed '0640' to '0600'
.M.... ...c	/etc/default/nss	mode changed '0644' to '0640'
SM5... .T.c	/etc/default/useradd	mode changed '0644' to '0600'
SM5... .T.c	/etc/hosts.allow	mode changed '0644' to '0444'
SM5... .T.	/etc/init/control-alt-delete.conf	mode changed '0644' to '0640'
SM5... .T.c	/etc/login.defs	mode changed '0644' to '0640'
.M.... G..c	/etc/ntp.conf	group changed 'root' to 'ntp' ; mode changed '0644' to '0600'
.M.... G..c	/etc/openldap/schema/dyngroup.ldif	group changed 'root' to 'ldap' ; mode changed '0444' to '0644'
.M.... G..c	/etc/openldap/schema/dyngroup.schema	group changed 'root' to 'ldap' ; mode changed '0444' to '0644'

continues on next page

Table 9 – continued from previous page

	File	Puppet log
.M.... G..c	/etc/openldap/schema/inetorgperson.ldif	group changed ‘root’ to ‘ldap’ ; mode changed ‘0444’ to ‘0644’
.M.... G..c	/etc/openldap/schema/inetorgperson.schema	group changed ‘root’ to ‘ldap’ ; mode changed ‘0444’ to ‘0644’
.M.... G..c	/etc/openldap/schema/java.ldif	group changed ‘root’ to ‘ldap’ ; mode changed ‘0444’ to ‘0644’
.M.... G..c	/etc/openldap/schema/java.schema	group changed ‘root’ to ‘ldap’ ; mode changed ‘0444’ to ‘0644’
.M.... G..c	/etc/openldap/schema/misc.ldif	group changed ‘root’ to ‘ldap’ ; mode changed ‘0444’ to ‘0644’
.M.... G..c	/etc/openldap/schema/misc.schema	group changed ‘root’ to ‘ldap’ ; mode changed ‘0444’ to ‘0644’
.M.... G..c	/etc/openldap/schema/nis.ldif	group changed ‘root’ to ‘ldap’ ; mode changed ‘0444’ to ‘0644’
.M.... G..c	/etc/openldap/schema/nis.schema	group changed ‘root’ to ‘ldap’ ; mode changed ‘0444’ to ‘0644’
.M.... G..c	/etc/openldap/schema/openldap.ldif	group changed ‘root’ to ‘ldap’ ; mode changed ‘0444’ to ‘0644’
.M.... G..c	/etc/openldap/schema/openldap.schema	group changed ‘root’ to ‘ldap’ ; mode changed ‘0444’ to ‘0644’
.M.... G..c	/etc/openldap/schema/pmi.ldif	group changed ‘root’ to ‘ldap’ ; mode changed ‘0444’ to ‘0644’
.M.... G..c	/etc/openldap/schema/pmi.schema	group changed ‘root’ to ‘ldap’ ; mode changed ‘0444’ to ‘0644’
.M.... G..c	/etc/openldap/schema/ppolicy.ldif	group changed ‘root’ to ‘ldap’ ; mode changed ‘0444’ to ‘0644’
.M.... G..c	/etc/openldap/schema/ppolicy.schema	group changed ‘root’ to ‘ldap’ ; mode changed ‘0444’ to ‘0644’
.M.... G..	/etc/puppetlabs/code	group changed ‘root’ to ‘puppet’ ; mode changed ‘0755’ to ‘0750’
.M.... G..	/etc/puppetlabs/code/environments	group changed ‘root’ to ‘puppet’ ; mode changed ‘0755’ to ‘0770’ (u=rwx,g=rwx,o-rwx)
.M.... G..	/etc/puppetlabs/code/environments/production	group changed ‘root’ to ‘puppet’ ; mode changed ‘0755’ to ‘0770’ (u=rwx,g=rwx,o-rwx)
.M.... G..	/etc/puppetlabs/puppet	group changed ‘root’ to ‘puppet’ ; mode changed ‘0755’ to ‘0750’
SM5... GT.c	/etc/puppetlabs/puppet/puppet.conf	group changed ‘root’ to ‘puppet’ ; mode changed ‘0644’ to ‘0640’
.M.... G..n	/etc/puppetlabs/puppetserver/conf.d	group changed ‘root’ to ‘puppet’ ; mode changed ‘0755’ to ‘0750’
SM5... GT.c	/etc/puppetlabs/puppetserver/conf.d/puppetserver.conf	group changed ‘root’ to ‘puppet’ ; mode changed ‘0644’ to ‘0640’
SM5... GT.c	/etc/puppetlabs/puppetserver/conf.d/web-routes.conf	group changed ‘root’ to ‘puppet’ ; mode changed ‘0644’ to ‘0640’
SM5... GT.c	/etc/puppetlabs/puppetserver/conf.d/webserver.conf	group changed ‘root’ to ‘puppet’ ; mode changed ‘0644’ to ‘0640’
SM5... GT.c	/etc/puppetlabs/puppetserver/logback.xml	group changed ‘root’ to ‘puppet’ ; mode changed ‘0644’ to ‘0640’
SM5... GT.c	/etc/puppetlabs/puppetserver/services.d/ca.cfg	group changed ‘root’ to ‘puppet’ ; mode changed ‘0644’ to ‘0640’

continues on next page

Table 9 – continued from previous page

	File	Puppet log
SM5... .T.c	/etc/rsyncd.conf	mode changed '0644' to '0400'
SM5... .T.c	/etc/rsyslog.conf	mode changed '0644' to '0600'
SM5... .T.c	/etc/securetty	mode changed '0600' to '0400'
SM5... .T.c	/etc/security/limits.conf	mode changed '0644' to '0640'
SM5... .T.c	/etc/sysconfig/ktune	mode changed '0777' to '0640'
SM5... .T.c	/etc/sysconfig/ntpd	mode changed '0644' to '0640'
SM5... .T.c	/etc/sysconfig/ntpdate	mode changed '0644' to '0640'
SM5... GT.c	/etc/sysconfig/puppetserver	group changed 'root' to 'puppet' ; mode changed '0644' to '0640'
SM5... .T.c	/etc/sysconfig/rsyslog	mode changed '0644' to '0640'
SM5... .T.c	/etc/sysconfig/slapd	mode changed '0644' to '0640'
SM5... .T.c	/etc/tuned.conf	mode changed '0777' to '0640'
.M.... ...	/var/lib/ntp	mode changed '0755' to '0750'

4.9 SIMP Security Control Mapping

This is the security control mapping for SIMP. The document has two main sections. The first section contains the security components, which in our case are SIMP modules. Each component is documented and mapped to the security control sources. The second contains security control sources which are references or reference documents that contain security guidance.

Contents:

4.9.1 SIMP Components

This section contains the security documentation for SIMP modules (represented as components).

SIMP

Module Name: `pupmod-simp-simp`

This puppet module provides a set of default classes that will be useful to most users and which form the foundation of the core SIMP installation.

Access Enforcement

SIMP uses a combination of discretionary and mandatory access control configurations to protect the operating system and the applications installed. Both forms of access control are built upon a model where a subject's (user or process) access to an object is controlled by the underlying operating system.

System Access

The `simp::admin` class provides a default set of accesses to SIMP systems.

By default, the `administrators` group may access the system and may gain access to the `root` account via passwordless `sudo`. Passwordless `sudo` was chosen since many systems run without passwords and only key access. This may be changed by setting the appropriate `Boolean` in the class.

These users may access the system from any location via `ssh`.

Users in the `security` group may access all systems from the `simp_options::trusted_nets` setting and are restricted to privileged use of a specific set of auditing-related commands which have been selected to disallow escalation of privileges.

Mountpoint Control

The `simp::mountpoints` class puts some specific access control configurations in place. The `/tmp`, `/var/tmp`, and `/dev/shm` directories have `nodev`, `noexec`, and `nosuid` set to prevent users from misusing the systems global read/write directories.

Additionally, `/sys` is mounted with `nodev` and `noexec`.

References: *AC-3 : ACCESS ENFORCEMENT*

Account Management

SIMP makes several account management decisions that are part of the overall account management strategy. One of those cases is the use of passwordless sudo for any user in the `administrators` or `auditors` groups. This is on by default due to the expected use of SSH keys and lack of local passwords.

References: *AC-2 : ACCOUNT MANAGEMENT*, *AC-6 (1) : AUTHORIZE ACCESS TO SECURITY FUNCTIONS*

Audit Events

SIMP logs successful and unsuccessful logins. Logins from unknown users are also logged. These settings are found in the `/etc/login.defs` file and are activated by default.

References: *AU-2 : AUDIT EVENTS*

Audit Storage Capacity

When a SIMP client serves as syslog server, logrotate is used to help manage storage capacity. The following log rotate rules are applied:

- Logs are rotated weekly
- A maximum of 12 rotated logs are stored

References: *AU-4 : AUDIT STORAGE CAPACITY*

Authenticator Management

The operating system protects locally stored passwords by hashing them. SIMP maximizes that protection by using the SHA512 algorithm.

Passwords expire every 180 days, must be at least 1 day old to be changed, and users are warned 14 days before the password will expire.

References: *IA-5c.*, *IA-5f.*, *IA-5h.*

Authorize Access to Security Functions

One of the main mechanisms to control access to security functions is the use of sudo. SIMP installs the following *sudo* rules:

Account	Sudo Commands	Run As Account	Password Required
administrators	/bin/su - root -l	root	no
administrators	/usr/sbin/puppetd	root	no
administrators	/usr/sbin/puppetca	root	no
administrators	/bin/rm -rf /var/lib/puppet/ssl	root	no
auditors	/bin/cat, /bin/ls, /usr/bin/lsattr, /sbin/aureport, /sbin/ausearch, /sbin/lspci, /sbin/lsusb, /sbin/lsmod, /usr/sbin/lsof, /bin/netstat, /sbin/ifconfig -a, /sbin/route, /sbin/route -[venC], /usr/bin/getent, /usr/bin/tail	root	no

References: *AC-6 (1) : AUTHORIZE ACCESS TO SECURITY FUNCTIONS*

Authorized Software

SIMP builds and configures centralized YUM repositories which are hosted on the SIMP server. These repositories host all of the packages that are needed to install a SIMP server and client. Additionally, all of the packages available on a CentOS/RedHat ISO are also placed in a repository.

All of the repositories installed by the SIMP module require packages to the signed with a known GPG key.

References: *CM-7 (5) : AUTHORIZED SOFTWARE / WHITELISTING*

Baseline Configuration

SIMP uses `crond` to schedule a number of jobs that help keep systems in a consistent and known baseline. The `simp` module ensures that the `cron` daemon is installed and running on all systems.

Specifically, the *puppet* agent is run via `cron` as is `aide` and a small number of maintenance tasks.

References: *CM-2 (1) : REVIEWS AND UPDATES*

Boundary Protection

The `simp::sysctl` class uses the kernel's sysctl `rp_filter` (reverse path) setting to drop spoofed IPv4 packets.

It also enables the use of `tcp_syncookies` to resist SYN flood attacks.

Finally, several classes in the `simp` module enable *IPTables* in a deny-by-default mode.

References: *SC-7 : BOUNDARY PROTECTION*

Centralized Management of Planned Audit Record Content

SIMP centrally controls what audit events are recorded on the clients. The SIMP module controls which of the those events are sent to local *syslog* daemon so that they may be optionally forwarded to a central syslog server. The following orthogonal list contains the conditions to be met for the SIMP logs to be sent to syslog.

- $programname == 'tlog-rec-session'
- $programname == 'tlog'
- $programname =='yum'
- $syslogfacility-text == 'cron'
- $syslogfacility-text == 'authpriv'
- $syslogfacility-text == 'local5'
- $syslogfacility-text == 'local6
- $syslogfacility-text == 'local7'
- $syslogpriority-text == 'emerg'
- $syslogfacility-text == 'kern' and $msg startswith 'IPT:'

SIMP also has a stock *rsyslog* module which is able to configure an `rsyslog` server for centralized collection. The stock `rsyslog` server configures the `rsyslog` daemon to accept logs from SIMP clients and places them in `/var/log/hosts/`. The following files are created for each host in that directory:

- tlog.log

- httpd.log
- dhcpd.log
- puppet-agent-err.log
- puppet-agent.log
- puppet-master.log
- audit.log
- slapd.log
- iptables.log
- secure.log
- messages.log
- maillog.log
- cron.log
- spooler.log
- boot.log

References: *AU-3 (2) : CENTRALIZED MANAGEMENT OF PLANNED AUDIT RECORD CONTENT*, *AU-13 (2) : REVIEW OF MONITORED SITES*, *AU-6 (4) : CENTRAL REVIEW AND ANALYSIS*

Concurrent Session Controls

A limit of 10 concurrent sessions are allowed per user. This value is controlled by the operating system's Pluggable Authentication Modules(PAM) module pam_limits.so setting.

References: *AC-10 : CONCURRENT SESSION CONTROL*

Configuration Management Policy and Procedures

All software developed and delivered under SIMP has a version associated with it. The aggregation of those components come together to make up a SIMP version. The current installed version of SIMP is written to a local file `/etc/simp/simp/version`.

References: *CM-1 : CONFIGURATION MANAGEMENT POLICY AND PROCEDURES*

Cryptographic Protection

SIMP enables Federal Information Processing Standard(*FIPS*) mode. FIPS Publication 140-2, is a computer security standard, developed by a U.S. Government and industry working group to validate the quality of cryptographic modules.

Enabling FIPS mode installs an integrity checking package and modifies ciphers available for applications to use.

References: *SC-13 : CRYPTOGRAPHIC PROTECTION*

Denial of Service Protection

SIMP takes several measures to reduce the chances of Denial of Service (DoS) attacks. The primary measures in place are to limit traffic with IPTables and set several kernel parameters. The kernel parameters set include limiting ICMP redirects, logging martian packets, ignoring ICMP broadcast traffic, ignoring bogus ICMP errors, and enabling protection against SYN cookies.

References: *SC-5 : DENIAL OF SERVICE PROTECTION*

Disable Inactive Accounts

Local accounts are disabled 35 days after their password expires. This is enforced using the 'inactive' value in the `/etc/default/useradd` file.

References: *AC-2 (3) : DISABLE INACTIVE ACCOUNTS*

Discretionary Access Control

SIMP uses the implementation of Discretionary Access Control (DAC) that is native to Linux. Specific file permissions have been assigned based on published security guidance for the supported operating in *SIMP Community Edition (CE) 6.6.0*.

To ensure default permissions are as restrictive as possible, the user's *umask* is set to 0077 while the daemon umask is set to 0027.

References: *AC-3 (4) : DISCRETIONARY ACCESS CONTROL*

Error Handling

Core dumps are disabled in SIMP. Core dump files may contain sensitive information and therefore are not written to disk.

References: *SI-11 : ERROR HANDLING*, *CP-12 : SAFE MODE*

Flaw Remediation

Continuous Remediation

Additionally, `puppet` runs on a regular basis to pull the system back into a known good state against a controlled configuration baseline.

System Updates

The *YUM* client is configured to point to all SIMP repositories. Each night, a `cron` job runs `yum update` to install updated packages on each SIMP client. Therefore any packages in a repository are delivered within a 24 hour time period.

References: *SI-2 : FLAW REMEDIATION*

Identification and Authentication

By default, The `root` user may not log into the system from any console, remote or local.

References: *IA-2 : IDENTIFICATION AND AUTHENTICATION (ORGANIZATIONAL USERS)*

Identification and Authentication

SIMP uses the SSSD client to authenticate with the SIMP LDAP server. The SSSD client is configured to:

- Use LDAP
- Use autofs
- Use sudo
- Use SSH
- Enforce a minimum user ID of 500

References: *IA-2 : IDENTIFICATION AND AUTHENTICATION (ORGANIZATIONAL USERS)*

Least Functionality

Whenever possible, SIMP prevents kernel modules that could cause harm or are unnecessary from loading. The operating system's modprobe blacklist feature is used to stop the following kernel modules from loading:

- bluetooth
- cramfs
- dccp
- dccp_ipv4
- dccp_ipv6
- freevxfs
- hfs
- hfsplus
- ieee1394
- jffs2
- net-pf-31
- rds
- sctp
- squashfs
- tipc
- udf
- usb-storage

Certain applications or application features are also explicitly disabled. The``hosts.equiv`` (part of the r-series of commands) is disabled. Prelinking, which changes binaries to increase startup time, is also disabled.

References: *CM-7 : LEAST FUNCTIONALITY*

Least Privilege

SIMP utilizes the cron daemon's access control by implementing the cron.allow feature. Only users in the cron.allow file are allowed to schedule cron jobs. Only the `root` user is in that file. The cron.deny file is forced to be absent, therefore all other users are denied the ability to schedule jobs.

The AT and incron services have the same access control configuration setup. Only the `root` user can schedule jobs and all other users are denied.

References: *AC-6 : LEAST PRIVILEGE*

Malicious Code Protection

SIMP installs the chkrootkit tool. Chkrootkit scans systems for the presence of rootkits. A cron job runs chkrootkit once per day.

References: *SI-3 : MALICIOUS CODE PROTECTION*, *SI-3a.*

Predictable Failure Prevention

SIMP uses TCP to transmit syslog messages. TCP has built in transmission retries and reliability of packet delivery.

References: *SI-13 : PREDICTABLE FAILURE PREVENTION*

Previous Login Notification

SIMP implements PAM's lastlog module to display previous login information. When a user logs into a host, the previous session's login time, source host, and terminal is displayed.

References: *AC-9 : PREVIOUS LOGON (ACCESS) NOTIFICATION*, *AC-9 (1) : UNSUCCESSFUL LOGONS*, *AC-9 (2) : SUCCESSFUL / UNSUCCESSFUL LOGONS*

Privileged Accounts

SIMP systems require a password when the system enters single user mode. In the event system maintenance needs to be performed at the system console, users must have the `root` password to be authenticated. Grub passwords are also set to prevent unauthorized modifications to boot parameters.

References: *AC-6 (5) : PRIVILEGED ACCOUNTS*

Role Based Access Control

SIMP creates a group called `administrators`. The `administrators` group is for privileged users and is configured to have `root` level access to the system.

References: *AC-2 (7) : ROLE-BASED SCHEMES*

Secure Name / Address Resolution Service

To protect against DNS spoofing, the `/etc/host` file is configured to log the potentially spoofed name lookups.

References: *SC-20 : SECURE NAME / ADDRESS RESOLUTION SERVICE (AUTHORITATIVE SOURCE)*

Session Lock

Sessions do not "lock". Instead, when there is a shell open and idle for 15 minutes, the session will timeout. This applies only when the shell is not running a command/process. Once the session is terminated, the user must reestablish the shell via console or SSH.

References: *AC-11a.*, *AC-11b.*

Session Termination

Sessions are terminated after three failed logins. Users must start a new session to make additional attempts to authenticate. Sessions will also timeout after 60 seconds if not attempt is made to authenticate. Lastly, when prompted to change a password, a user has 3 attempts to successfully change it before the session is terminated.

References: *AC-12 : SESSION TERMINATION*

System Use Notification

A default SIMP warning banner is presented to the user prior to login. The content of that banner is:

```
----------------------------------- ATTENTION -----------------------------------
                    THIS IS A RESTRICTED COMPUTER SYSTEM

This computer system, and all related equipment, networks, and network devices
are provided for authorised use only.  All systems controlled by this
organisation will be monitored for all lawful purposes.  Monitoring includes
the totality of the operating system and connected networks.  No events on this
system are excluded from record and there are no exclusions from this policy.

Use of this system constitutes consent to full monitoring of your activities
for use by the authorised monitoring organisation.  Unauthorised use of this
system, including uninvited connections, may subject you to criminal
prosecution.

The data collected from this system may be used for any purpose by the
collecting organisation.  If you do not agree to this monitoring, discontinue
use of the system IMMEDIATELY.
```

References: *AC-8a.*, *AC-8a.2.*, *AC-8a.3.*, *AC-8c.1.*

AIDE

Module Name: `pupmod-simp-aide`

This module installs AIDE and creates a baseline set of rules and files that should be monitored.

Automated Notifications of Integrity Violations

When an integrity event is detected by AIDE, the event is written both to a local AIDE log and to syslog. When combined with a central logging capability, all AIDE events can be stored and searched from a central location.

References *SC-7 (2) : PUBLIC ACCESS*

Software, Firmware, and Information Integrity

AIDE is installed and configured. SIMP configures a default set of files to be monitored. When a change is made to one of those files, AIDE will log that event.

The default list of files include:

```
/boot   NORMAL
/bin    NORMAL
/sbin   NORMAL
/lib    NORMAL
/opt    NORMAL
/usr    NORMAL
/root   NORMAL
!/usr/src
!/usr/tmp
/etc    PERMS
!/etc/mtab
!/etc/.*~
/etc/exports  NORMAL
/etc/fstab    NORMAL
/etc/passwd   NORMAL
/etc/group    NORMAL
/etc/gshadow  NORMAL
/etc/shadow   NORMAL
/etc/security/opasswd   NORMAL
/etc/hosts.allow   NORMAL
/etc/hosts.deny    NORMAL
/etc/sudoers NORMAL
/etc/skel NORMAL
/etc/logrotate.d NORMAL
/etc/resolv.conf DATAONLY
/etc/nscd.conf NORMAL
/etc/securetty NORMAL
/etc/profile NORMAL
/etc/bashrc NORMAL
/etc/bash_completion.d/ NORMAL
/etc/login.defs NORMAL
/etc/zprofile NORMAL
/etc/zshrc NORMAL
```

(continues on next page)

(continued from previous page)

```
/etc/zlogin NORMAL
/etc/zlogout NORMAL
/etc/profile.d/ NORMAL
/etc/X11/ NORMAL
/etc/yum.conf NORMAL
/etc/yumex.conf NORMAL
/etc/yumex.profiles.conf NORMAL
/etc/yum/ NORMAL
/etc/yum.repos.d/ NORMAL
/var/log   LOG
!/var/log/sa
!/var/log/aide/aide.log
!/var/log/aide/aide.report
/etc/audit/ LSPP
/etc/libaudit.conf LSPP
/usr/sbin/stunnel LSPP
/var/spool/at LSPP
/etc/at.allow LSPP
/etc/at.deny LSPP
/etc/cron.allow LSPP
/etc/cron.deny LSPP
/etc/cron.d/ LSPP
/etc/cron.daily/ LSPP
/etc/cron.hourly/ LSPP
/etc/cron.monthly/ LSPP
/etc/cron.weekly/ LSPP
/etc/crontab LSPP
/var/spool/cron/root LSPP
/etc/login.defs LSPP
/etc/securetty LSPP
/var/log/faillog LSPP
/var/log/lastlog LSPP
/etc/hosts LSPP
/etc/sysconfig LSPP
/etc/inittab LSPP
/etc/grub LSPP
/etc/rc.d LSPP
/etc/ld.so.conf LSPP
/etc/localtime LSPP
/etc/sysctl.conf LSPP
/etc/modprobe.d/00_simp_blacklist.conf LSPP
/etc/pam.d LSPP
/etc/security LSPP
/etc/aliases LSPP
/etc/postfix LSPP
/etc/ssh/sshd_config LSPP
/etc/ssh/ssh_config LSPP
/etc/stunnel LSPP
/etc/vsftpd.ftpusers LSPP
/etc/vsftpd LSPP
/etc/issue LSPP
/etc/issue.net LSPP
```

(continues on next page)

(continued from previous page)

```
/etc/cups LSPP
!/var/log/and-httpd
```

References: *SC-7 : BOUNDARY PROTECTION*

Transfer to Alternate Storage

The AIDE logs are configured to be sent to syslog. In a default SIMP install, this does not send them to an external host until one is defined.

References: *AU-4 (1) : TRANSFER TO ALTERNATE STORAGE*

Apache

Module Name: `pupmod-simp-apache`

This Puppet module provides the capability to configure Apache and component sites.

Audit Storage and Capacity

The Apache logs are written to the `/var/log` partition. This puts them on the same logical volume as the audit logs. That volume is mounted on a separate partition so that log space does not interfere with operations.

References: *AU-4 : AUDIT STORAGE CAPACITY*

Automated Central Management / Application / Verification

SIMP uses *rsync* (over *stunnel*) to keep files in `/var/www` synchronized between all web servers. Any files that need to be the same on all web servers are then managed from the *Puppet Server*.

References: *CM-7 (1) : PERIODIC REVIEW*

Content of Audit Records

The SIMP Apache configuration uses the following string to populate the Apache logs: `%h %l %u %t "%r" %>s %b "%{Referer}i" "%{User-Agent}i"`

That will capture the remote hostname, the request log ID, the remote username, the time of the request, the first line of the request, the request status, the size of the response, the referrer, and the user agent used for the request.

There is an additional log file written for SSL logs. The following string is used for that log: `%t %h %{SSL_CLIENT_S_DN_CN}x %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b %s`

That will capture the time stamp, hostname, the distinguished name of the client certification, SSL protocol used, first line of the request, size of the response, and the request status.

References: *AU-3 : CONTENT OF AUDIT RECORDS*

Information Flow Enforcement

The Apache module explicitly opens up ports 80 and 443 for the root web servers by using IPTables rules. The connecting source IPs are limited to the value of `$simp_options::trusted_nets`, which for most installs is the local network.

References: *AC-4 : INFORMATION FLOW ENFORCEMENT*

Least Privilege

The Apache service runs under the `apache` user and `apache` group. This is allows directory permissions to limit the service's access to files/directories not owned by the `apache` user/group. The apache user does not have a valid login shell.

References: *AC-6 : LEAST PRIVILEGE*

Mandatory Access Control

When SELinux is enabled in SIMP, Apache is configured to run within a context. Booleans specific to apache are also set.

References: *AC-3 : ACCESS ENFORCEMENT*, *AC-3 (4) : DISCRETIONARY ACCESS CONTROL*

Transfer to Alternate Storage

The Apache logs are configured to be sent to syslog. In a default SIMP install, this does not send them to an external host until one is defined.

References: *AU-4 (1) : TRANSFER TO ALTERNATE STORAGE*

Transmission Confidentiality and Integrity

The *SIMP Server* has a *TLS* enabled Apache web server running on port 443. The protocols are limited to TLSv1.2. If the web client does not support those protocols, the connection will be rejected. The certificates are in the `/etc/pki/simp_apps/simp_apache/x509` directory.

References: *SC-8 : TRANSMISSION CONFIDENTIALITY AND INTEGRITY*

Auditd

Module Name: `pupmod-simp-auditd`

This Puppet module provides the capability to configure auditd and rules affecting your system.

Audit Events

SIMP audit rules were built by using industry best practices gathered over the years. The heaviest reliance has been on the SCAP-Security Guide (SSG). SIMP aims for a balance between performance and operational needs so the settings are rarely an exact match from these guides.

The following audit rules are applied to SIMP systems:

```
## For audit 1.6.5 and higher
##

# Ignore errors
# This may sound counterintuitive, but we'd rather skip bad rules and load the
# rest than miss half the file.  Warnings are still logged in the daemon
# restart output.
-i

## Remove any existing rules
-D

## Continue loading rules on failure.
# Particularly with the automatically generated nature of these rules in
# Puppet, it is possible that one or more may fail to load. We want to continue
# in that case so that we audit as much as possible.
-c

## Increase buffer size to handle the increased number of messages.
## Feel free to increase this if the machine panic's
# Default: 8192
-b 32768

## Set failure mode to panic
# Default: 2
-f 1

## Rate limit messages
# Default: 0
# If you set this to non-zero, you almost definitely want to set -f to 1 above.
-r 0

## Get rid of all anonymous and daemon junk.  It clogs up the logs and doesn't
# do anyone # any good.
-a exit,never -F auid=-1

# Ignore system services. In most guides this is tagged onto every rule but
# that just makes for more processing time.
-a exit,never -F auid!=0 -F auid<500

## unsuccessful file operations
-a always,exit -F arch=b64 -S creat -S mkdir -S mknod -S link -S symlink -S
mkdirat -S mknodat -S linkat -S symlinkat -S openat -S open -S close -S rename
-S truncate -S ftruncate -S rmdir -S unlink -S unlinkat -F exit=-EPERM -k access
-a always,exit -F arch=b32 -S creat -S mkdir -S mknod -S link -S symlink -S
```

(continues on next page)

(continued from previous page)

```
mkdirat -S mknodat -S linkat -S symlinkat -S openat -S open -S close -S rename
-S truncate -S ftruncate -S rmdir -S unlink -S unlinkat -F exit=-EPERM -k access

-a always,exit -F perm=a -F exit=-EPERM -k access

# Permissions auditing
-a always,exit -F arch=b64 -S chown -S fchmod -S fchmodat -S fchown -S fchownat
-S lchown -S setxattr -S lsetxattr -S fsetxattr -S removexattr -S lremovexattr
-S fremovexattr -k perm_mod
-a always,exit -F arch=b32 -S chown -S fchmod -S fchmodat -S fchown -S fchownat
-S lchown -S setxattr -S lsetxattr -S fsetxattr -S removexattr -S lremovexattr
-S fremovexattr -k perm_mod

# Audit useful items that someone does when su'ing to root.
# Had to add an entry at the top for getting rid of anonymous records.  They
# are only moderately useful and contain *way* too much noise since this covers
# things like cron as well.

-a always,exit -F arch=b64 -F auid!=0 -F uid=0 -S capset -S mknod -S pivot_root
-S quotactl -S setsid -S settimeofday -S setuid -S swapoff -S swapon -k
su-root-activity
-a always,exit -F arch=b32 -F auid!=0 -F uid=0 -S capset -S mknod -S pivot_root
-S quotactl -S setsid -S settimeofday -S setuid -S swapoff -S swapon -k
su-root-activity

# Audit the execution of suid and sgid binaries.
-a always,exit -F arch=b64 -F euid=0 -F uid!=0 -S execve -k suid-root-exec
-a always,exit -F arch=b32 -F euid=0 -F uid!=0 -S execve -k suid-root-exec

## Audit the loading and unloading of kernel modules.
-w /sbin/insmod -p x -k modules
-w /sbin/rmmod -p x -k modules
-w /sbin/modprobe -p x -k modules
-a always,exit -F arch=b64 -S init_module -S delete_module -k modules
-a always,exit -F arch=b32 -S init_module -S delete_module -k modules

## Things that could affect time
-a exit,always -F arch=b32 -S adjtimex -S stime -S clock_settime -S settimeofday
-k audit_time_rules
-a exit,always -F arch=b64 -S adjtimex -S clock_settime -S settimeofday -k
audit_time_rules

-w /etc/localtime -p wa -k audit_time_rules

## Things that could affect system locale
-a always,exit -F arch=b32 -S sethostname -S setdomainname -k
audit_network_modifications
-a always,exit -F arch=b64 -S sethostname -S setdomainname -k
audit_network_modifications
-w /etc/issue -p wa -k audit_network_modifications
-w /etc/issue.net -p wa -k audit_network_modifications
-w /etc/hosts -p wa -k audit_network_modifications
```

(continues on next page)

(continued from previous page)

```
-w /etc/sysconfig/network -p wa -k audit_network_modifications

# Mount options.
-a always,exit -F arch=b32 -S mount -S umount -S umount2 -k mount
-a always,exit -F arch=b64 -S mount -S umount2 -k mount

# audit umask changes.
# This is uselessly noisy.
# -a exit,always -S umask -k umask

-w /etc/group -p wa -k audit_account_changes
-w /etc/group- -p wa -k audit_account_changes
-w /etc/passwd -p wa -k audit_account_changes
-w /etc/passwd- -p wa -k audit_account_changes
-w /etc/gshadow -p wa -k audit_account_changes
-w /etc/shadow -p wa -k audit_account_changes
-w /etc/shadow- -p wa -k audit_account_changes
-w /etc/security/opasswd -p wa -k audit_account_changes

-w /etc/selinux/ -p wa -k MAC-policy

-w /var/log/faillog -p wa -k logins
-w /var/log/lastlog -p wa -k logins

-w /var/run/utmp -p wa -k session
-w /var/run/btmp -p wa -k session
-w /var/run/wtmp -p wa -k session

-w /etc/sudoers -p wa -k CFG_sys

# Generally good things to audit.
-w /boot/grub/grub.conf -p wa -k CFG_grub
-w /etc/aliases -p wa -k CFG_sys
-w /etc/anacrontab -p wa -k CFG_cron
-w /etc/at.deny -p wa -k CFG_sys
-w /etc/bashrc -p wa -k CFG_shell
-w /etc/cron.d -p wa -k CFG_cron
-w /etc/cron.daily -p wa -k CFG_cron
-w /etc/cron.deny -p wa -k CFG_cron
-w /etc/cron.hourly -p wa -k CFG_cron
-w /etc/cron.monthly -p wa -k CFG_cron
-w /etc/cron.weekly -p wa -k CFG_cron
-w /etc/crontab -p wa -k CFG_cron
-w /etc/csh.cshrc -p wa -k CFG_shell
-w /etc/csh.login -p wa -k CFG_shell
-w /etc/default -p wa -k CFG_sys
-w /etc/exports -p wa -k CFG_sys
-w /etc/fstab -p wa -k CFG_sys
-w /etc/host.conf -p wa -k CFG_sys
-w /etc/hosts.allow -p wa -k CFG_sys
-w /etc/hosts.deny -p wa -k CFG_sys
-w /etc/initlog.conf -p wa -k CFG_sys
```

(continues on next page)

(continued from previous page)

```
-w /etc/inittab -p wa -k CFG_sys
-w /etc/issue -p wa -k CFG_sys
-w /etc/issue.net -p wa -k CFG_sys
-w /etc/krb5.conf -p wa -k CFG_sys
-w /etc/ld.so.conf -p wa -k CFG_sys
-w /etc/ld.so.conf.d -p wa -k CFG_sys
-w /etc/login.defs -p wa -k CFG_sys
-w /etc/modprobe.conf.d -p wa -k CFG_sys
-w /etc/modprobe.d/00_simp_blacklist.conf -p wa -k CFG_sys
-w /etc/nsswitch.conf -p wa -k CFG_sys
-w /etc/pam.d -p wa -k CFG_pam
-w /etc/pam_smb.conf -p wa -k CFG_pam
-w /etc/profile -p wa -k CFG_shell
-w /etc/rc.d/init.d -p wa -k CFG_sys
-w /etc/rc.local -p wa -k CFG_sys
-w /etc/rc.sysinit -p wa -k CFG_sys
-w /etc/resolv.conf -p wa -k CFG_sys
-w /etc/securetty -p wa -k CFG_sys
-w /etc/security -p wa -k CFG_security
-w /etc/services -p wa -k CFG_services
-w /etc/shells -p wa -k CFG_shell
-w /etc/snmp/snmpd.conf -p wa -k CFG_sys
-w /etc/ssh/sshd_config -p wa -k CFG_sys
-w /etc/sysconfig -p wa -k CFG_sys
-w /etc/sysctl.conf -p wa -k CFG_sys
-w /etc/xinetd.conf -p wa -k CFG_xinetd
-w /etc/xinetd.d -p wa -k CFG_sys
-w /etc/yum.conf -p wa -k yum-config
-w /etc/yum.repos.d -p wa -k yum-config
-w /lib/firmware/microcode.dat -p wa -k CFG_sys
-w /var/spool/at -p wa -k CFG_sys
-a exit,always -F arch=b32 -S ptrace -k paranoid
-a exit,always -F arch=b64 -S ptrace -k paranoid
-a always,exit -F arch=b32 -S personality -k paranoid
-a always,exit -F arch=b64 -S personality -k paranoid
-w /etc/aide.conf -p wa -k CFG_aide
-w /etc/aide.conf.d/default.aide -p wa -k CFG_aide
-w /etc/rc.d/init.d/auditd -p wa -k auditd
-w /var/log/audit.log -p wa -k audit-logs
-w /etc/pam_ldap.conf -p a -k CFG_etc_ldap
-w /etc/pki/private -p wa -k PKI
-w /etc/pki/public -p wa -k PKI
-w /etc/pki/cacerts -p wa -k PKI
-w /etc/pki/private/blade01.my.domain.pem -p wa -k PKI
-w /etc/pki/public/blade01.my.domain.pub -p wa -k PKI
-a always,exit -F dir=/etc/puppet -F uid!=puppet -p wa -k Puppet_Config
-a always,exit -F dir=/var/log/puppet -F uid!=puppet -p wa -k Puppet_Log
-a always,exit -F dir=/var/run/puppet -F uid!=puppet -p wa -k Puppet_Run
-a always,exit -F dir=$vardir/ssl -F uid!=puppet -p wa -k Puppet_SSL
-w /var/log/audit.log.1 -p rwa -k audit-logs
-w /var/log/audit.log.2 -p rwa -k audit-logs
-w /var/log/audit.log.3 -p rwa -k audit-logs
```

(continues on next page)

(continued from previous page)

```
-w /var/log/audit.log.4 -p rwa -k audit-logs
-w /var/log/audit.log.5 -p rwa -k audit-logs
-w /etc/init/ -p wa -k CFG_upstart
```

References: *AU-2 : AUDIT EVENTS*

Audit Generation

SIMP enables auditd on all systems. Auditd is the userspace component to the Linux Auditing System. It's responsible for writing audit records to the disk. Viewing the logs is done with the ausearch or aureport utilities. Configuring the audit rules is done with the auditctl utility. During startup, the rules in /etc/audit/audit.rules are read by auditctl.

The audit daemon is configured to initiate auditing at boot time.

References: *AU-12 : AUDIT GENERATION*, *AU-12a.*, *AU-12c.*

Audit Reviews and Updates

SIMP developers are constantly reviewing the audit rules for accuracy, relevance, and performance. Rules are added and in some cases removed as security requirements change or as we find ways to improve the performance of auditd.

References: *AU-2 (3) : REVIEWS AND UPDATES*

Audit Storage Capacity

To help manage the amount of local storage occupied by audit logs, the following rules are applied:

- A maximum of 5 log files are retained. The oldest is removed when the logs are rotated.
- Files can reach a maximum of 24 Mb before being rotated.

References: *AU-4 : AUDIT STORAGE CAPACITY*

Content of Audit Records

The Linux audit daemon contains event type, date/time, host, and outcome of events by default. Each event type has a varying level of detail depending on the audited action. The auditd documentation should be referred to for the event types.

References: *AU-3 : CONTENT OF AUDIT RECORDS*

Response to Audit Processing Failures

The auditing dispatcher is system that allows external applications to access and make use of the auditd daemon in real time. When the internal queue of the audit dispatcher is full, a message is sent to syslog.

References: *AU-5 : RESPONSE TO AUDIT PROCESSING FAILURES*

Response To Audit Processing Failures - Audit Storage Capacity

Auditd has been configured to handle audit failures or potential failures due to storage capacity. Those settings include:

- Send a warning to syslog when there is less than 75Mb of space on the audit partition (space_left).
- Suspend the audit daemon when there is less than 50Mb of space left on the audit partition (admin_space_left).

References: *AU-5 (1) : AUDIT STORAGE CAPACITY*

Transfer to Alternate Storage

The audit logs are configured to be sent to syslog. In a default SIMP install, this does not send them to an external host until one is defined.

References: *AU-4 (1) : TRANSFER TO ALTERNATE STORAGE*

ClamAV

Module Name: `pupmod-simp-clamav`

This module installs and configures ClamAV. ClamAV is a command line malicious code detection tool.

Malicious Code Protection

SIMP installs and configures ClamAV. ClamAV is a command line malicious code detection tool.

ClamAV is scheduled to run once per day and scans `/tmp`, `/var/tmp`, and `/dev/shm`.

References: *SI-3 : MALICIOUS CODE PROTECTION*, *SI-3a.*

Transfer to Alternate Storage

The ClamAV logs are configured to be sent to syslog. In a default SIMP install, this does not send them to an external host until one is defined.

References: *AU-4 (1) : TRANSFER TO ALTERNATE STORAGE*

Compliance

Module Name: `pupmod-simp-compliance_markup`

This module adds a function to enable compliance annotations in Puppet code.

Automated Central Management / Application / Verification

SIMP has a custom function that is embedded within the module code to validate each variable. Those variables are then verified against SIMP default configuration settings using hiera. Each time puppet runs on a client, the hiera variables are validated against SIMP defaults.

References: *CM-7 (1) : PERIODIC REVIEW*

IPTables

Module Name: `pupmod-simp-iptables`

The *iptables* module manages all IPTables and IP6Tables rules in an atomic fashion. All rules are applied only once per puppet agent run during the application of the last executed *iptables* resource.

Boundary Protection

The SIMP IPTables module adds an IPtables rule that will prevent external IP addresses from being able to send spoofed packets to your system. This applies to IPv6 traffic. IPv4 spoofing is prevented using the rp_filter sysctl setting.

References: *SC-7 : BOUNDARY PROTECTION*

Enable / Disable Security Policy Filters

Only the root user or a user who has escalated to root can modify the IPtables filters.

References: *AC-4 (10) : ENABLE / DISABLE SECURITY POLICY FILTERS*

Information Flow Enforcement

IPTables is installed and running on all SIMP clients. IPtables controls the flow of inbound traffic by limiting IP addresses, protocols, and port numbers.

The default IPTables rules:

- Allow all outbound traffic
- Allow ping
- Allow traffic from established connections
- Drop broadcast traffic
- Drop multicast traffic
- Drop all other traffic

References: *AC-4 : INFORMATION FLOW ENFORCEMENT*, *CM-7b.*

Named/Bind

Module Name: pupmod-simp-named

This Puppet module provides the capability to configure either a chrooted named process or a caching nameserver.

Automated Central Management / Application / Verification

Named configuration files are synchronized between the *Puppet Server* and the *DNS* servers using *rsync*.

References: *CM-7 (1) : PERIODIC REVIEW*

Information Flow Enforcement

The named module explicitly opens TCP and UDP ports 53 for the DNS by using IPTables rules. The connecting source IPs are limited to the value of `$simp_options::trusted_nets` which for most installs is the local network.

References: *AC-4 : INFORMATION FLOW ENFORCEMENT*

Least Functionality

The SIMP named service is configured to run within a chroot jail. This ensures that the service cannot see or access files outside of named directory. Should the named service become remotely compromised, the attack cannot be escalated to other parts of the file system.

References: *CM-7 : LEAST FUNCTIONALITY*

Least Privilege

The named service runs under the `named` user and `named` group. This is allows directory permissions to limit the service's access to files/directories not owned by the `apache` user/group. The `named` user does not have a valid login shell.

References: *AC-6 : LEAST PRIVILEGE*

OpenLDAP

Module Name: pupmod-simp-openldap

This Puppet module provides the capability to configure OpenLDAP servers and clients.

Access Enforcement

User password history (shadowLastChange) is written to the LDAP server. For this to happen, the user is given write access to their own shadowLastChange entry in LDAP.

References: *AC-3 : ACCESS ENFORCEMENT*

Audit Storage Capacity

Logrotate is used to help manage log storage capacity. The following log rotate rules are applied to OpenLDAP:

- Logs are rotated daily
- A maximum of 7 rotated logs are stored

References: *AU-4 : AUDIT STORAGE CAPACITY*

Authenticator Management

Authenticator strength is enforced using slapo-ppolicy overlay for LDAP. The ppolicy overlay is then configured to use PAM cracklib to enforce complexity.

For the default password complexity rules see the *What is the Password Complexity for SIMP?* FAQ.

The integration point between the remote LDAP server and PAM is the pam_ldap pam module. SIMP configures pam_ldap to point to the SIMP LDAP server and communicates using TLS.

References: *IA-5 (1)(a)*, *IA-5 (1)(e)*

Content of Audit Records

All LDAP transactions to the LDAP database are audited and written to `/var/log/slapd.audit/` in LDIF format.

References: *AU-3 : CONTENT OF AUDIT RECORDS*

Device Identification and Authentication

There is an account and password setup to authenticate devices needing to synchronize with the LDAP server. The username for that account is `LDAPSync` and the account and associated password are stored in LDAP.

There is also an account for a device to authenticate prior to being allowed to do anything else with the LDAP server. The username for that account is hostAuth and the account is stored in LDAP.

References: *IA-3 : DEVICE IDENTIFICATION AND AUTHENTICATION*

Identification and Authentication (Organizational Users)

The pam_ldap module ensures that the username is mapped to the uid portion of the DN in LDAP.

The pam_ldap module is configured to tell the clients to ignore the following user names, forcing them to be authenticated locally:

- root
- bin
- daemon
- adm
- lp
- mail
- operator

- nobody
- dbus
- ntp
- saslauth
- postfix
- sshd
- puppet
- stunnel
- nscd
- haldaemon
- clamav
- rpcuser
- rpc
- clam
- nfsnobody
- rpm
- nslcd
- avahi
- gdm
- rtkit
- pulse
- hsqldb
- radvd
- apache
- tomcat

There as an ldap account created for LDAP administration. The username for that account is `LDAPAdmin`.

References: *IA-2 : IDENTIFICATION AND AUTHENTICATION (ORGANIZATIONAL USERS)*

Information Flow Enforcement

Since TCPWrappers has a default deny policy in place, a specific entry is added to allow all hosts to connect to the slapd service.

The OpenLDAP module explicitly opens up ports 389 (LDAP) and 636 (LDAPS) using IPTables rules. The connecting source IPs are limited to the value of `$simp_options::trusted_nets` which for most installs is the local network.

References: *AC-4 : INFORMATION FLOW ENFORCEMENT*

Least Privilege

The OpenLDAP service runs under the `ldap` user and `ldap` group. This is allows directory permissions to limit the service's access to files/directories not owned by the `ldap` user/group. The ldap user does not have a valid login shell.

The default LDAP server policy denies all users access to everything (default deny). Access to LDAP entries are explicitly added.

References: *AC-6 : LEAST PRIVILEGE*

Transfer to Alternate Storage

The LDAP logs are configured to be sent to syslog. In a default SIMP install, this does not send them to an external host until one is defined.

References: *AU-4 (1) : TRANSFER TO ALTERNATE STORAGE*

Transmission Confidentiality and Integrity

The pam_ldap OpenLDAP module is configured to use TLS to communicate with the LDAP server. It currently only supports TLSv1, TLSv2, and SSLv3. Supporting SSLv3 is a limitation of OpenLDAP.

References: *SC-8 : TRANSMISSION CONFIDENTIALITY AND INTEGRITY*

PAM

Module Name: `pupmod-simp-pam`

This Puppet module provides the capability to configure various PAM settings on the system.

Included are capabilities to manage:

- system-auth
- Group-based access to the system
- access.conf

The system-auth settings are a bit draconian, but simple enough to work within.

Authenticator Management

Authenticator strength is enforced using pam_cracklib.so. The SIMP settings ensure that passwords:

- Have at least four characters that are different from the previous password
- Do not repeat a character more than two times in a row
- Do not have the username (forward or reversed) in the password
- Have at least one character from three of the four classes: upper, lower, number, special character
- Have at least 14 characters
- Are not the same as any of the previous 24 passwords

Passwords are hashed using the SHA512 algorithm. Each password is hashed using 1000 rounds.

References: *IA-5 (1)(a)*, *IA-5 (1)(e)*

Discretionary Access Control

When creating a home directory for the first time, PAM creates that directory using the umask of `0077`.

References: *AC-2 : ACCOUNT MANAGEMENT*

Group Authentication

SIMP does not use group accounts for authenticators. Instead, users are added to a group. In the case of the `administrators` group, a user first authenticates to their account, and then escalates to `root` using sudo.

References: *IA-2 (5) : GROUP AUTHENTICATION*

Least Privilege

SIMP uses the access conf file to identify which accounts can login to a system. After all other identification and authentication checks have passed, the pam access.conf file is checked to ensure the user is allowed to login. SIMP allows `root` and the `administrators` group to login to all systems and the `simp` user to login to the *Puppet Server*. All other users must be explicitly added to the `access.conf` file using the SIMP *pam* module.

References: *AC-6 : LEAST PRIVILEGE*

Privileged Accounts

Linux historically uses the wheel group to as an administrators group. SIMP makes use of the sudoers file with more granular group permissions. The PAM module enforces that only the root user is in the wheel group.

References: *AC-6 (5) : PRIVILEGED ACCOUNTS*

Unsuccessful Login Attempts

A user is allowed three failed logins per session. After the third unsuccessful login attempt, the user is disconnected and must initiate a new session in order to make additional attempts.

After 5 failed login attempts in a time 15 minute span, the account is locked for a period of 15 minutes.

The root user account will be locked for one hour after 5 failed login attempts.

References: *AC-7 : UNSUCCESSFUL LOGON ATTEMPTS*, *AC-7(b)*, *IA-11*

Pupmod

Module Name: `pupmod-simp-pupmod`

This Puppet module provides the capability to configure both puppet servers and puppet clients.

The ability to switch puppetd from a system service to a cron job is also supported.

Access Enforcement

The *Puppet Server* uses a whitelist to determine which puppet clients can connect via the network. The certificate of the connecting client must match the fully qualified domain name of the system as resolved via *DNS*. If it does not then the connection is denied.

References: *AC-3 : ACCESS ENFORCEMENT*

Audit Events

The following puppet files are added to the audit rules so that modifications to them are audited by auditd.

- `-a always,exit -F dir=${confdir} -F uid!=puppet -p wa -k Puppet_Config`
- `-a always,exit -F dir=${logdir} -F uid!=puppet -p wa -k Puppet_Log`
- `-a always,exit -F dir=${rundir} -F uid!=puppet -p wa -k Puppet_Run`
- `-a always,exit -F dir=${ssldir} -F uid!=puppet -p wa -k Puppet_SSL`

References: *AU-2 : AUDIT EVENTS*

Audit Storage and Capacity

The Puppet logs are written to the `/var/log` partition. This puts them on the same logical volume as the audit logs. That volume is mounted on a separate partition so that log space does not interfere with operations.

The *Puppet Server* logs reports from client puppet runs in `/var/lib/puppet/reports`. The SIMP *pupmod puppet module* purges reports older than 7 days.

References: *AU-4 : AUDIT STORAGE CAPACITY*

Automated Change Implementation

The most prominent tool in the SIMP architecture is Puppet. Puppet is a client/server tool where managed nodes run the Puppet agent application. One or more servers run the *Puppet Server* application in the form of a web-based service using *TLS* encrypted connections.

The Puppet agent sends facts to the Puppet Server and request a catalog. The master compiles and returns that node's catalog, using several sources of information it has access to.

Once it receives a catalog, Puppet agent applies it by checking each resource the catalog describes. If it finds any resources that are not in their desired state, it makes any changes necessary to correct them. After applying the catalog, the agent submits a report to the Puppet Server.

Puppet clients have a scheduled job configured to run the puppet agent every 30 minutes by default.

References: *CM-3 (3) : AUTOMATED CHANGE IMPLEMENTATION*

Content of Audit Records

The *Puppet Server* log level is set to `WARN`. Any changes that are made during a run of the puppet agent, are logged to the client's log file and captured by *syslog*.

References: *AU-3 : CONTENT OF AUDIT RECORDS*

Information Flow Enforcement

The *pupmod* *puppet module* listens on ports 8140 and 8141 by default and makes these ports available via the system firewall.

Port 8140 is the *Puppet Server* port and 8141 is the certificate authority port. The connecting source IPs are limited to the value of `$trusted_nets`, which for most installs, is the local network.

References: *AC-4 : INFORMATION FLOW ENFORCEMENT*

Public Key Infrastructure

Puppet has its own public key infrastructure (*PKI*) that is used exclusively for the puppet application. The PKI is used to provide access control and protect communications between the *Puppet Server* and the clients.

Additional information on Puppet and PKI can be found in the product documentation.

SIMP installs a scheduled job that will download a copy of the certificate revocation list (*CRL*) two times per day. If there is a client certificate that needs to be revoked, they can be added to the CRL and will no longer be able to connect to the Puppet Server.

References: *SC-17 : PUBLIC KEY INFRASTRUCTURE CERTIFICATES*

Transfer to Alternate Storage

The puppet logs are configured to be sent to syslog facility local6. In a default SIMP install, this does not send them to an external host until one is defined.

References: *AU-4 (1) : TRANSFER TO ALTERNATE STORAGE*

Transmission Confidentiality and Integrity

The *Puppet Server* uses *TLS* for communications between the puppet master and clients. The protocols for that communications are limited to TLSv1.2.

References: *SC-8 : TRANSMISSION CONFIDENTIALITY AND INTEGRITY*

Rsync

Module Name: `pupmod-simp-rsync`

This Puppet module provides the capability to configure an rsync server. The intent is for this server to be run encrypted via a stunnel channel.

Client rsync rules have not been integrated into this module at this time.

Access Enforcement

SIMP rsync is limited to read only so that files cannot be remotely modified.

References: *AC-3 : ACCESS ENFORCEMENT*

Information Flow Enforcement

The rsync server port (over stunnel) is open to the IP addresses defined by the value of `$simp_options::trusted_nets`, which for most installs is the local network.

References: *AC-4 : INFORMATION FLOW ENFORCEMENT*

Transmission Confidentiality and Integrity

Rsync is not encrypted. To mitigate this, SIMP only allows rsync to listen on the local host. The server to client communications is then protected using the SIMP stunnel module.

References: *SC-8 : TRANSMISSION CONFIDENTIALITY AND INTEGRITY*

SSH

Module Name: `pupmod-simp-ssh`

This Puppet module manages the configuration of the system-wide SSH server and client.

Authenticator Management

The SSH daemon disallows the use of empty passwords. Additionally, the SSH daemon uses PAM to support authenticator security.

References: *IA-5c.*

Cryptographic Key Establishment and Management

The SSH server is configured to use the system's existing system certificates. Those certificates are stored in `/etc/pki` and are used to generated the SSH server certificates stored in `/etc/ssh`.

References: *SC-12 : CRYPTOGRAPHIC KEY ESTABLISHMENT AND MANAGEMENT*

Cryptographic Protection

In the default FIPS mode, the SSH daemon limits the key exchange algorithms to:

- ecdh-sha2-nistp521
- ecdh-sha2-nistp384
- ecdh-sha2-nistp256
- diffie-hellman-group-exchange-sha256

In the default FIPS mode, the SSH daemon limits the message authentication code (MAC) algorithms to:

- hmac-sha2-256
- hmac-sha1'

In the default FIPS mode, the SSH client limits the key exchange algorithms to:

- aes256-gcm@openssh.com
- aes128-gcm@openssh.com

In the default FIPS mode, the SSH client limits the MAC algorithms to:

- hmac-sha2-256
- hmac-sha1'

References: *SC-13 : CRYPTOGRAPHIC PROTECTION*

Information Flow Enforcement

The SSH module explicitly opens up port 22 for the SSH server by using IPTables rules.

Since TCPWrappers has a default deny policy in place, a specific entry is added to allow all hosts to connect to the SSH service.

References: *AC-4 : INFORMATION FLOW ENFORCEMENT*

Least Privilege

The SSH service runs under the `ssh` user and `ssh` group. This is allows directory permissions to limit the service's access to files/directories not owned by the `ssh` user/group. The ssh user does not have a valid login shell.

X11 forwarding over SSH is explicitly disallowed. This limits the exposure of the SSH server to networks outside of the control of SIMP.

References: *AC-6 : LEAST PRIVILEGE*

Privileged Accounts

The SSH daemon disables root login. The root user is only allowed to login locally.

References: *AC-6 (5) : PRIVILEGED ACCOUNTS*, *AC-6 (2) : NON-PRIVILEGED ACCESS FOR NONSECURITY FUNCTIONS*

System Use Notification

The SSH daemon is configured to use the `/etc/issue.net` file to present a banner prior to login.

References: *AC-8a.*

Transfer to Alternate Storage

The SSH logs are configured to be sent to the syslog facility `AUTHPRIV`. In a default SIMP install, this does not send them to an external host until one is defined.

References: *AU-4 (1) : TRANSFER TO ALTERNATE STORAGE*

Stunnel

Module Name: `pupmod-simp-stunnel`

This Puppet module provides the capability to configure stunnel channels on your system.

Access Enforcement

Stunnel verifies the client certificate as a form of access control. It only checks that the client certificate is valid.

References: *AC-3 : ACCESS ENFORCEMENT*

Least Functionality

The SIMP stunnel service is configured to run within a chroot jail. This ensures that the service cannot see or access files outside of stunnel directory. Should the stunnel service become remotely compromised, the attack cannot be escalated to other parts of the file system.

References: *CM-7 : LEAST FUNCTIONALITY*

Least Privilege

The stunnel service runs under the `stunnel` user and `stunnel` group. This allows directory permissions to limit the service's access to files/directories not owned by the `stunnel` user/group. The stunnel user does not have a valid login shell.

References: *AC-6 : LEAST PRIVILEGE*

Transfer to Alternate Storage

The stunnel logs are configured to be sent to syslog. In a default SIMP install, this does not send them to an external host until one is defined.

References: *AU-4 (1) : TRANSFER TO ALTERNATE STORAGE*

Transmission Confidentiality and Integrity

The stunnel module is a framework used by other modules to encrypt communications for applications that might not natively support it.

The cipher negotiation is determined by the OpenSSL ciphers. In a default SIMP system, this will be TLSv1.1 or higher.

The certificates used for stunnel are in the `/etc/pki` directory.

References: *SC-8 : TRANSMISSION CONFIDENTIALITY AND INTEGRITY*

Sudo

Module Name: `pupmod-simp-sudo`

This Puppet module manages the sudoers infrastructure.

Authorize Access to Security Functions

The SIMP Sudoers module make use of the operating system's sudo capability to grant access to privileged functions. Specific rules are written to grant each user/group access to privileged command(s).

References: *AC-6 (1) : AUTHORIZE ACCESS TO SECURITY FUNCTIONS*

Tlog

Module Name: `pupmod-simp-tlog`

This Puppet module provides the capability to use *Tlog* to log designated login sessions to a *syslog* server.

Session Audit

The *Tlog* application is installed on each SIMP node. It is set, by default, to log interactive shell sessions to privileged user accounts via a login shell hook.

The `tlog-rec-session` application may optionally be set as the user's default shell to log all sessions without the optional hook.

A `tlog-play` application is also provided to replay captured sessions.

In addition to *Tlog*, the *PAM* module `pam_tty_audit` is used to record keystrokes during a `root` user's session. Additional accounts can be audited by adding them to the parameter `pam::tty_audit_users`.

Note: As a safeguard against recording sensitive credentials (such as passwords), both `tlog` and `pam_tty_audit` do NOT record when `echo` is turned off.

Warning: The audit logs **WILL RECORD SENSITIVE DETAILS** (such as passwords) for any scripts or applications that:

- Do _not_ protect terminal output while entering or echoing sensitive data
- AND are run by an audited user (e.g., `root`)

It is therefore HIGHLY RECOMMENDED to update any such scripts or applications to turn of echo during these sensitive operations.

References: *AU-14 : SESSION AUDIT*

TCP Wrappers

Module Name: `pupmod-simp-tcpwrappers`

This Puppet module allows you to manage `/etc/hosts.allow`, `/etc/hosts.deny` is set to ALL:ALL by default.

Information Flow Enforcement

TCP Wrappers is enabled on SIMP systems. TCP Wrappers is a host-based networking ACL system, used to filter access to IP addresses. It allows host or subnetwork IP addresses, names and/or ident query replies, to be used as tokens on which to filter for access control purposes.

TCP Wrappers uses the `/etc/hosts.allow` and the `/etc/hosts.deny` files to configure the access control.

References: *AC-4 : INFORMATION FLOW ENFORCEMENT*

Least Privilege

SIMP configures tcpwrappers to deny all, meaning that a service will be denied access to the TCP stack unless it is explicitly allowed. Each SIMP module that needs access to the TCP stack has an entry added to the host.allow file using this tcpwrappers module.

References: *AC-6 : LEAST PRIVILEGE*

4.9.2 Security Control Sources

NIST 800-53 Rev4

Control Family: ACCESS CONTROL

AC-1 : ACCESS CONTROL POLICY AND PROCEDURES

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: This control addresses the establishment of policy and procedures for the effective implementation of selected security controls and control enhancements in the AC family. Policy and procedures reflect applicable federal laws, Executive Orders, directives, regulations, policies, standards, and guidance. Security program policies and procedures at the organization level may make the need for system-specific policies and procedures unnecessary. The policy can be included as part of the general information security policy for organizations or conversely, can be represented by multiple policies reflecting the complex nature of certain organizations. The procedures can be established for the security program in general and for particular information systems, if needed. The organizational risk management strategy is a key factor in establishing policy and procedures.

Related Controls: *PM-9*

AC-1a.

Develops, documents, and disseminates to [Assignment: organization-defined personnel or roles]:

AC-1a.1.

An access control policy that addresses purpose, scope, roles, responsibilities, management commitment, coordination among organizational entities, and compliance; and

AC-1a.2.

Procedures to facilitate the implementation of the access control policy and associated access controls; and

AC-1b.

Reviews and updates the current:

AC-1b.1.

Access control policy [Assignment: organization-defined frequency]; and

AC-1b.2.

Access control procedures [Assignment: organization-defined frequency].

Control Family: ACCESS CONTROL

AC-2 : ACCOUNT MANAGEMENT

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: Information system account types include, for example, individual, shared, group, system, guest/anonymous, emergency, developer/manufacturer/vendor, temporary, and service. Some of the account management requirements listed above can be implemented by organizational information systems. The identification of authorized users of the information system and the specification of access privileges reflects the requirements in other security controls in the security plan. Users requiring administrative privileges on information system accounts receive additional scrutiny by appropriate organizational personnel (e.g., system owner, mission/business owner, or chief information security officer) responsible for approving such accounts and privileged access. Organizations may choose to define access privileges or other attributes by account, by type of account, or a combination of both. Other attributes required for authorizing access include, for example, restrictions on time-of-day, day-of-week, and point-of-origin. In defining other account attributes, organizations consider system-related requirements (e.g., scheduled maintenance, system upgrades) and mission/business requirements, (e.g., time zone differences, customer requirements, remote access to support travel requirements). Failure to consider these factors could affect information system availability. Temporary and emergency accounts are accounts intended for short-term use. Organizations establish temporary accounts as a part of normal account activation procedures when there is a need for short-term accounts without the demand for immediacy in account activation. Organizations establish emergency accounts in response to crisis situations and with the need for rapid account activation. Therefore, emergency account activation may bypass normal account authorization processes. Emergency and temporary accounts are not to be confused with infrequently used accounts (e.g., local logon accounts used for special tasks defined by organizations or when network resources are unavailable). Such accounts remain available and are not subject to automatic disabling or removal dates. Conditions for disabling or deactivating accounts include, for example: (i) when shared/group, emergency, or temporary accounts are no longer required; or (ii) when individuals are transferred or terminated. Some types of information system accounts may require specialized training.

Related Controls: *AC-3*, *AC-4*, *AC-5*, *AC-6*, *AC-10*, *AC-17*, *AC-19*, *AC-20*, *AU-9*, *IA-2*, *IA-4*, *IA-5*, *IA-8*, *CM-5*, *CM-6*, *CM-11*, *MA-3*, *MA-4*, *MA-5*, *PL-4*, *SC-13*

AC-2a.

Identifies and selects the following types of information system accounts to support organizational missions/business functions: [Assignment: organization-defined information system account types];

AC-2b.

Assigns account managers for information system accounts;

AC-2c.

Establishes conditions for group and role membership;

AC-2d.

Specifies authorized users of the information system, group and role membership, and access authorizations (i.e., privileges) and other attributes (as required) for each account;

AC-2e.

Requires approvals by [Assignment: organization-defined personnel or roles] for requests to create information system accounts;

AC-2f.

Creates, enables, modifies, disables, and removes information system accounts in accordance with [Assignment: organization-defined procedures or conditions];

AC-2g.

Monitors the use of information system accounts;

AC-2h.

Notifies account managers:

AC-2h.1.

When accounts are no longer required;

AC-2h.2.

When users are terminated or transferred; and

AC-2h.3.

When individual information system usage or need-to-know changes;

AC-2i.

Authorizes access to the information system based on:

AC-2i.1.

A valid access authorization;

AC-2i.2.

Intended system usage; and

AC-2i.3.

Other attributes as required by the organization or associated missions/business functions;

AC-2j.

Reviews accounts for compliance with account management requirements [Assignment: organization-defined frequency]; and

AC-2k.

Establishes a process for reissuing shared/group account credentials (if deployed) when individuals are removed from the group.

AC-2 (1) : AUTOMATED SYSTEM ACCOUNT MANAGEMENT

Baseline-Impact: *MODERATE*, **HIGH**

The organization employs automated mechanisms to support the management of information system accounts.

Note: The use of automated mechanisms can include, for example: using email or text messaging to automatically notify account managers when users are terminated or transferred; using the information system to monitor account usage; and using telephonic notification to report atypical system account usage.

AC-2 (2) : REMOVAL OF TEMPORARY / EMERGENCY ACCOUNTS

Baseline-Impact: *MODERATE*, **HIGH**

The information system automatically [Selection: removes; disables] temporary and emergency accounts after [Assignment: organization-defined time period for each type of account].

Note: This control enhancement requires the removal of both temporary and emergency accounts automatically after a predefined period of time has elapsed, rather than at the convenience of the systems administrator.

AC-2 (3) : DISABLE INACTIVE ACCOUNTS

Baseline-Impact: *MODERATE*, **HIGH**

> The information system automatically disables inactive accounts after [Assignment: organization-defined time period].

AC-2 (4) : AUTOMATED AUDIT ACTIONS

Baseline-Impact: *MODERATE*, **HIGH**

> The information system automatically audits account creation, modification, enabling, disabling, and removal actions, and notifies [Assignment: organization-defined personnel or roles].

Related Controls: *AU-2*, *AU-12*

AC-2 (5) : INACTIVITY LOGOUT

Baseline-Impact: **HIGH**

> The organization requires that users log out when [Assignment: organization-defined time-period of expected inactivity or description of when to log out].

Related Controls: *SC-23*

AC-2 (6) : DYNAMIC PRIVILEGE MANAGEMENT

> The information system implements the following dynamic privilege management capabilities: [Assignment: organization-defined list of dynamic privilege management capabilities].

Note: In contrast to conventional access control approaches which employ static information system accounts and predefined sets of user privileges, dynamic access control approaches (e.g., service-oriented architectures) rely on run time access control decisions facilitated by dynamic privilege management. While user identities may remain relatively constant over time, user privileges may change more frequently based on ongoing mission/business requirements and operational needs of organizations. Dynamic privilege management can include, for example, the immediate revocation of privileges from users, as opposed to requiring that users terminate and restart their sessions to reflect any changes in privileges. Dynamic privilege management can also refer to mechanisms that change the privileges of users based on dynamic rules as opposed to editing specific user profiles. This type of privilege management includes, for example, automatic adjustments of privileges if users are operating out of their normal work times, or if information systems are under duress or in emergency maintenance situations. This control enhancement also includes the ancillary effects of privilege changes, for example, the potential changes to encryption keys used for communications. Dynamic privilege management can support requirements for information system resiliency.

Related Controls: *AC-16*

AC-2 (7) : ROLE-BASED SCHEMES

The organization:

Note: Privileged roles are organization-defined roles assigned to individuals that allow those individuals to perform certain security-relevant functions that ordinary users are not authorized to perform. These privileged roles include, for example, key management, account management, network and system administration, database administration, and web administration.

AC-2 (7)(a)

Establishes and administers privileged user accounts in accordance with a role-based access scheme that organizes allowed information system access and privileges into roles;

AC-2 (7)(b)

Monitors privileged role assignments; and

AC-2 (7)(c)

Takes [Assignment: organization-defined actions] when privileged role assignments are no longer appropriate.

AC-2 (8) : DYNAMIC ACCOUNT CREATION

The information system creates [Assignment: organization-defined information system accounts] dynamically.

Note: Dynamic approaches for creating information system accounts (e.g., as implemented within service-oriented architectures) rely on establishing accounts (identities) at run time for entities that were previously unknown. Organizations plan for dynamic creation of information system accounts by establishing trust relationships and mechanisms with the appropriate authorities to validate related authorizations and privileges.

Related Controls: *AC-16*

AC-2 (9) : RESTRICTIONS ON USE OF SHARED / GROUP ACCOUNTS

The organization only permits the use of shared/group accounts that meet [Assignment: organization-defined conditions for establishing shared/group accounts].

AC-2 (10) : SHARED / GROUP ACCOUNT CREDENTIAL TERMINATION

The information system terminates shared/group account credentials when members leave the group.

AC-2 (11) : USAGE CONDITIONS

Baseline-Impact: HIGH

The information system enforces [Assignment: organization-defined circumstances and/or usage conditions] for [Assignment: organization-defined information system accounts].

Note: Organizations can describe the specific conditions or circumstances under which information system accounts can be used, for example, by restricting usage to certain days of the week, time of day, or specific durations of time.

AC-2 (12) : ACCOUNT MONITORING / ATYPICAL USAGE

Baseline-Impact: HIGH

The organization:

Note: Atypical usage includes, for example, accessing information systems at certain times of the day and from locations that are not consistent with the normal usage patterns of individuals working in organizations.

Related Controls: *CA-7*

AC-2 (12)(a)

Monitors information system accounts for [Assignment: organization-defined atypical usage]; and

AC-2 (12)(b)

Reports atypical usage of information system accounts to [Assignment: organization-defined personnel or roles].

AC-2 (13) : DISABLE ACCOUNTS FOR HIGH-RISK INDIVIDUALS

Baseline-Impact: HIGH

The organization disables accounts of users posing a significant risk within [Assignment: organization-defined time period] of discovery of the risk.

Note: Users posing a significant risk to organizations include individuals for whom reliable evidence or intelligence indicates either the intention to use authorized access to information systems to cause harm or through whom adversaries will cause harm. Harm includes potential adverse impacts to organizational operations and assets, individuals, other organizations, or the Nation. Close coordination between authorizing officials, information system administrators, and human resource managers is essential in order for timely execution of this control enhancement.

Related Controls: *PS-4*

Control Family: ACCESS CONTROL

AC-3 : ACCESS ENFORCEMENT

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The information system enforces approved authorizations for logical access to information and system resources in accordance with applicable access control policies.

Note: Access control policies (e.g., identity-based policies, role-based policies, control matrices, cryptography) control access between active entities or subjects (i.e., users or processes acting on behalf of users) and passive entities or objects (e.g., devices, files, records, domains) in information systems. In addition to enforcing authorized access at the information system level and recognizing that information systems can host many applications and services in support of organizational missions and business operations, access enforcement mechanisms can also be employed at the application and service level to provide increased information security.

Related Controls: *AC-2*, *AC-4*, *AC-5*, *AC-6*, *AC-16*, *AC-17*, *AC-18*, *AC-19*, *AC-20*, *AC-21*, *AC-22*, *AU-9*, *CM-5*, *CM-6*, *CM-11*, *MA-3*, *MA-4*, *MA-5*, *PE-3*

AC-3 (1) : RESTRICTED ACCESS TO PRIVILEGED FUNCTIONS

> [Withdrawn: Incorporated into AC-6].

AC-3 (2) : DUAL AUTHORIZATION

> The information system enforces dual authorization for [Assignment: organization-defined privileged commands and/or other organization-defined actions].

Note: Dual authorization mechanisms require the approval of two authorized individuals in order to execute. Organizations do not require dual authorization mechanisms when immediate responses are necessary to ensure public and environmental safety. Dual authorization may also be known as two-person control.

Related Controls: *CP-9*, *MP-6*

AC-3 (3) : MANDATORY ACCESS CONTROL

> The information system enforces [Assignment: organization-defined mandatory access control policy] over all subjects and objects where the policy:

Note: Mandatory access control as defined in this control enhancement is synonymous with nondiscretionary access control, and is not constrained only to certain historical uses (e.g., implementations using the Bell-LaPadula Model). The above class of mandatory access control policies constrains what actions subjects can take with information obtained from data objects for which they have already been granted access, thus preventing the subjects from passing the information to unauthorized subjects and objects. This class of mandatory access control policies also constrains what actions subjects can take with respect to the propagation of access control privileges; that is, a subject with a privilege cannot pass that privilege to other subjects. The policy is uniformly enforced over all subjects and objects to which the information system has control. Otherwise, the access control policy can be circumvented. This enforcement typically is provided via an implementation that meets the reference monitor concept (see AC-25). The policy is bounded by the information system boundary (i.e., once the information is passed outside of the control of the system, additional means may be required to ensure that the constraints on the information remain in effect). The trusted subjects described above are granted privileges consistent with the concept of least privilege (see AC-6). Trusted subjects are only given the minimum privileges relative to the above policy necessary for satisfying organizational mission/business needs. The control is most applicable when there is some policy mandate (e.g., law, Executive Order, directive, or regulation) that establishes a policy regarding access to sensitive/classified information and some users of the information system are not authorized access to all sensitive/classified information resident in the information system. This control can operate in conjunction with AC-3 (4). A subject that is constrained in its operation by policies governed by this control is still able to operate under the less rigorous constraints of AC-3 (4), but policies governed by this control take precedence over the less rigorous constraints of AC-3 (4). For example, while a mandatory access control policy imposes a constraint preventing a subject from passing information to another subject operating at a different sensitivity label, AC-3 (4) permits the subject to pass the information to any subject with the same sensitivity label as the subject.

Related Controls: *AC-25*, *SC-11*

AC-3 (3)(a)

Is uniformly enforced across all subjects and objects within the boundary of the information system;

AC-3 (3)(b)

Specifies that a subject that has been granted access to information is constrained from doing any of the following;

AC-3 (3)(b)(1)

Passing the information to unauthorized subjects or objects;

AC-3 (3)(b)(2)

Granting its privileges to other subjects;

AC-3 (3)(b)(3)

Changing one or more security attributes on subjects, objects, the information system, or information system components;

AC-3 (3)(b)(4)

Choosing the security attributes and attribute values to be associated with newly created or modified objects; or

AC-3 (3)(b)(5)

Changing the rules governing access control; and

AC-3 (3)(c)

Specifies that [Assignment: organization-defined subjects] may explicitly be granted [Assignment: organization-defined privileges (i.e., they are trusted subjects)] such that they are not limited by some or all of the above constraints.

AC-3 (4) : DISCRETIONARY ACCESS CONTROL

The information system enforces [Assignment: organization-defined discretionary access control policy] over defined subjects and objects where the policy specifies that a subject that has been granted access to information can do one or more of the following:

Note: When discretionary access control policies are implemented, subjects are not constrained with regard to what actions they can take with information for which they have already been granted access. Thus, subjects that have been granted access to information are not prevented from passing (i.e., the subjects have the discretion to pass) the information to other subjects or objects. This control enhancement can operate in conjunction with AC-3 (3). A subject that is constrained in its operation by policies governed by AC-3 (3) is still able to operate under the less rigorous constraints of this control enhancement. Thus, while AC-3 (3) imposes constraints preventing a subject from passing information to another subject operating at a different sensitivity level, AC-3 (4) permits the subject to pass the information to any subject at the same sensitivity level. The policy is bounded by the information system boundary. Once the information is passed outside of the control of the information system, additional means may be required to ensure that the constraints remain in effect. While the older, more traditional definitions of discretionary access control require identity-based access control, that limitation is not required for this use of discretionary access control.

AC-3 (4)(a)

Pass the information to any other subjects or objects;

AC-3 (4)(b)

Grant its privileges to other subjects;

AC-3 (4)(c)

Change security attributes on subjects, objects, the information system, or the information systems components;

AC-3 (4)(d)

Choose the security attributes to be associated with newly created or revised objects; or

AC-3 (4)(e)

Change the rules governing access control.

AC-3 (5) : SECURITY-RELEVANT INFORMATION

The information system prevents access to [Assignment: organization-defined security-relevant information] except during secure, non-operable system states.

Note: Security-relevant information is any information within information systems that can potentially impact the operation of security functions or the provision of security services in a manner that could result in failure to enforce system security policies or maintain the isolation of code and data. Security-relevant information includes, for example, filtering rules for routers/firewalls, cryptographic key management information, configuration parameters for security services, and access control lists. Secure, non-operable system states include the times in which information systems are not performing mission/business-related processing (e.g., the system is off-line for maintenance, troubleshooting, boot-up, shut down).

Related Controls: *CM-3*

AC-3 (6) : PROTECTION OF USER AND SYSTEM INFORMATION

[Withdrawn: Incorporated into MP-4 and SC-28].

AC-3 (7) : ROLE-BASED ACCESS CONTROL

The information system enforces a role-based access control policy over defined subjects and objects and controls access based upon [Assignment: organization-defined roles and users authorized to assume such roles].

Note: Role-based access control (RBAC) is an access control policy that restricts information system access to authorized users. Organizations can create specific roles based on job functions and the authorizations (i.e., privileges) to perform needed operations on organizational information systems associated with the organization-defined roles. When users are assigned to the organizational roles, they inherit the authorizations or privileges defined for those roles. RBAC simplifies privilege administration for organizations because privileges are not assigned directly to every user (which can be a significant number of individuals for mid- to large-size organizations) but are instead acquired through role assignments. RBAC can be implemented either as a mandatory or discretionary form of access control. For organizations implementing RBAC with mandatory access controls, the requirements in AC-3 (3) define the scope of the subjects and objects covered by the policy.

AC-3 (8) : REVOCATION OF ACCESS AUTHORIZATIONS

> The information system enforces the revocation of access authorizations resulting from changes to the security attributes of subjects and objects based on [Assignment: organization-defined rules governing the timing of revocations of access authorizations].

Note: Revocation of access rules may differ based on the types of access revoked. For example, if a subject (i.e., user or process) is removed from a group, access may not be revoked until the next time the object (e.g., file) is opened or until the next time the subject attempts a new access to the object. Revocation based on changes to security labels may take effect immediately. Organizations can provide alternative approaches on how to make revocations immediate if information systems cannot provide such capability and immediate revocation is necessary.

AC-3 (9) : CONTROLLED RELEASE

> The information system does not release information outside of the established system boundary unless:

Note: Information systems can only protect organizational information within the confines of established system boundaries. Additional security safeguards may be needed to ensure that such information is adequately protected once it is passed beyond the established information system boundaries. Examples of information leaving the system boundary include transmitting information to an external information system or printing the information on one of its printers. In cases where the information system is unable to make a determination of the adequacy of the protections provided by entities outside its boundary, as a mitigating control, organizations determine procedurally whether the external information systems are providing adequate security. The means used to determine the adequacy of the security provided by external information systems include, for example, conducting inspections or periodic testing, establishing agreements between the organization and its counterpart organizations, or some other process. The means used by external entities to protect the information received need not be the same as those used by the organization, but the means employed are sufficient to provide consistent adjudication of the security policy to protect the information. This control enhancement requires information systems to employ technical or procedural means to validate the information prior to releasing it to external systems. For example, if the information system passes information to another system controlled by another organization, technical means are employed to validate that the security attributes associated with the exported information are appropriate for the receiving system. Alternatively, if the information system passes information to a printer in organization-controlled space, procedural means can be employed to ensure that only appropriately authorized individuals gain access to the printer. This control enhancement is most applicable when there is some policy mandate (e.g., law, Executive Order, directive, or regulation) that establishes policy regarding access to the information, and that policy applies beyond the realm of a particular information system or organization.

AC-3 (9)(a)

The receiving [Assignment: organization-defined information system or system component] provides [Assignment: organization-defined security safeguards]; and

AC-3 (9)(b)

[Assignment: organization-defined security safeguards] are used to validate the appropriateness of the information designated for release.

AC-3 (10) : AUDITED OVERRIDE OF ACCESS CONTROL MECHANISMS

The organization employs an audited override of automated access control mechanisms under [Assignment: organization-defined conditions].

Related Controls: *AU-2*, *AU-6*

Control Family: ACCESS CONTROL

AC-4 : INFORMATION FLOW ENFORCEMENT

Priority: P1

Baseline-Impact: *MODERATE*, **HIGH**

The information system enforces approved authorizations for controlling the flow of information within the system and between interconnected systems based on [Assignment: organization-defined information flow control policies].

Note:

Information flow control regulates where information is allowed to travel within an information system and between information systems (as opposed to who is allowed to access the information) and without explicit regard to subsequent accesses to that information. Flow control restrictions include, for example, keeping export-controlled information from being transmitted in the clear to the Internet, blocking outside traffic that claims to be from within the organization, restricting web requests to the Internet that are not from the internal web proxy server, and limiting information transfers between organizations based on data structures and content. Transferring information between information systems representing different security domains with different security policies introduces risk that such transfers violate one or more domain security policies. In such situations, information owners/stewards provide guidance at designated policy enforcement points between interconnected systems. Organizations consider mandating specific architectural solutions when required to enforce specific security policies. Enforcement includes, for example: (i) prohibiting information transfers between interconnected systems (i.e., allowing access only); (ii) employing hardware mechanisms to enforce one-way information flows; and (iii) implementing trustworthy regrading mechanisms to reassign security attributes and security labels.

Organizations commonly employ information flow control policies and enforcement mechanisms to control the flow of information between designated sources and destinations (e.g., networks, individuals, and devices) within information systems and between interconnected systems. Flow control is based on the characteristics of the information and/or the information path. Enforcement occurs, for example, in boundary protection devices (e.g., gateways, routers, guards, encrypted tunnels, firewalls) that employ rule sets or establish configuration settings that restrict information system services, provide a packet-filtering capability based on header information, or message-filtering capability based on message content (e.g., implementing key word searches or using document characteristics). Organizations also consider the trustworthiness of filtering/inspection mechanisms (i.e., hardware, firmware, and software components) that are critical to information flow enforcement. Control enhancements 3 through 22 primarily address cross-domain solution needs which focus on more advanced filtering techniques, in-depth analysis, and stronger flow enforcement mechanisms implemented in cross-domain products, for example, high-assurance guards. Such capabilities are generally not available in commercial off-the-shelf information technology products.

Related Controls: *AC-3*, *AC-17*, *AC-19*, *AC-21*, *CM-6*, *CM-7*, *SA-8*, *SC-2*, *SC-5*, *SC-7*, *SC-18*

AC-4 (1) : OBJECT SECURITY ATTRIBUTES

The information system uses [Assignment: organization-defined security attributes] associated with [Assignment: organization-defined information, source, and destination objects] to enforce [Assignment: organization-defined information flow control policies] as a basis for flow control decisions.

Note: Information flow enforcement mechanisms compare security attributes associated with information (data content and data structure) and source/destination objects, and respond appropriately (e.g., block, quarantine, alert administrator) when the mechanisms encounter information flows not explicitly allowed by information flow policies. For example, an information object labeled Secret would be allowed to flow to a destination object labeled Secret, but an information object labeled Top Secret would not be allowed to flow to a destination object labeled Secret. Security attributes can also include, for example, source and destination addresses employed in traffic filter firewalls. Flow enforcement using explicit security attributes can be used, for example, to control the release of certain types of information.

Related Controls: *AC-16*

AC-4 (2) : PROCESSING DOMAINS

The information system uses protected processing domains to enforce [Assignment: organization-defined information flow control policies] as a basis for flow control decisions.

Note: Within information systems, protected processing domains are processing spaces that have controlled interactions with other processing spaces, thus enabling control of information flows between these spaces and to/from data/information objects. A protected processing domain can be provided, for example, by implementing domain and type enforcement. In domain and type enforcement, information system processes are assigned to domains; information is identified by types; and information flows are controlled based on allowed information accesses (determined by domain and type), allowed signaling among domains, and allowed process transitions to other domains.

AC-4 (3) : DYNAMIC INFORMATION FLOW CONTROL

> The information system enforces dynamic information flow control based on [Assignment: organization-defined policies].

Note: Organizational policies regarding dynamic information flow control include, for example, allowing or disallowing information flows based on changing conditions or mission/operational considerations. Changing conditions include, for example, changes in organizational risk tolerance due to changes in the immediacy of mission/business needs, changes in the threat environment, and detection of potentially harmful or adverse events.

Related Controls: *SI-4*

AC-4 (4) : CONTENT CHECK ENCRYPTED INFORMATION

> The information system prevents encrypted information from bypassing content-checking mechanisms by [Selection (one or more): decrypting the information; blocking the flow of the encrypted information; terminating communications sessions attempting to pass encrypted information; [Assignment: organization-defined procedure or method]].

Related Controls: *SI-4*

AC-4 (5) : EMBEDDED DATA TYPES

> The information system enforces [Assignment: organization-defined limitations] on embedding data types within other data types.

Note: Embedding data types within other data types may result in reduced flow control effectiveness. Data type embedding includes, for example, inserting executable files as objects within word processing files, inserting references or descriptive information into a media file, and compressed or archived data types that may include multiple embedded data types. Limitations on data type embedding consider the levels of embedding and prohibit levels of data type embedding that are beyond the capability of the inspection tools.

AC-4 (6) : METADATA

> The information system enforces information flow control based on [Assignment: organization-defined metadata].

Note: Metadata is information used to describe the characteristics of data. Metadata can include structural metadata describing data structures (e.g., data format, syntax, and semantics) or descriptive metadata describing data contents (e.g., age, location, telephone number). Enforcing allowed information flows based on metadata enables simpler and more effective flow control. Organizations consider the trustworthiness of metadata with regard to data accuracy (i.e., knowledge that the metadata values are correct with respect to the data), data integrity (i.e., protecting against unauthorized changes to metadata tags), and the binding of metadata to the data payload (i.e., ensuring sufficiently strong binding techniques with appropriate levels of assurance).

Related Controls: *AC-16*, *SI-7*

AC-4 (7) : ONE-WAY FLOW MECHANISMS

The information system enforces [Assignment: organization-defined one-way information flows] using hardware mechanisms.

AC-4 (8) : SECURITY POLICY FILTERS

The information system enforces information flow control using [Assignment: organization-defined security policy filters] as a basis for flow control decisions for [Assignment: organization-defined information flows].

Note: Organization-defined security policy filters can address data structures and content. For example, security policy filters for data structures can check for maximum file lengths, maximum field sizes, and data/file types (for structured and unstructured data). Security policy filters for data content can check for specific words (e.g., dirty/clean word filters), enumerated values or data value ranges, and hidden content. Structured data permits the interpretation of data content by applications. Unstructured data typically refers to digital information without a particular data structure or with a data structure that does not facilitate the development of rule sets to address the particular sensitivity of the information conveyed by the data or the associated flow enforcement decisions. Unstructured data consists of: (i) bitmap objects that are inherently non language-based (i.e., image, video, or audio files); and (ii) textual objects that are based on written or printed languages (e.g., commercial off-the-shelf word processing documents, spreadsheets, or emails). Organizations can implement more than one security policy filter to meet information flow control objectives (e.g., employing clean word lists in conjunction with dirty word lists may help to reduce false positives).

AC-4 (9) : HUMAN REVIEWS

The information system enforces the use of human reviews for [Assignment: organization-defined information flows] under the following conditions: [Assignment: organization-defined conditions].

Note: Organizations define security policy filters for all situations where automated flow control decisions are possible. When a fully automated flow control decision is not possible, then a human review may be employed in lieu of, or as a complement to, automated security policy filtering. Human reviews may also be employed as deemed necessary by organizations.

AC-4 (10) : ENABLE / DISABLE SECURITY POLICY FILTERS

The information system provides the capability for privileged administrators to enable/disable [Assignment: organization-defined security policy filters] under the following conditions: [Assignment: organization-defined conditions].

Note: For example, as allowed by the information system authorization, administrators can enable security policy filters to accommodate approved data types.

AC-4 (11) : CONFIGURATION OF SECURITY POLICY FILTERS

> The information system provides the capability for privileged administrators to configure [Assignment: organization-defined security policy filters] to support different security policies.

Note: For example, to reflect changes in security policies, administrators can change the list of dirty words that security policy mechanisms check in accordance with the definitions provided by organizations.

AC-4 (12) : DATA TYPE IDENTIFIERS

> The information system, when transferring information between different security domains, uses [Assignment: organization-defined data type identifiers] to validate data essential for information flow decisions.

Note: Data type identifiers include, for example, filenames, file types, file signatures/tokens, and multiple internal file signatures/tokens. Information systems may allow transfer of data only if compliant with data type format specifications.

AC-4 (13) : DECOMPOSITION INTO POLICY-RELEVANT SUBCOMPONENTS

> The information system, when transferring information between different security domains, decomposes information into [Assignment: organization-defined policy-relevant subcomponents] for submission to policy enforcement mechanisms.

Note: Policy enforcement mechanisms apply filtering, inspection, and/or sanitization rules to the policy-relevant subcomponents of information to facilitate flow enforcement prior to transferring such information to different security domains. Parsing transfer files facilitates policy decisions on source, destination, certificates, classification, attachments, and other security-related component differentiators.

AC-4 (14) : SECURITY POLICY FILTER CONSTRAINTS

> The information system, when transferring information between different security domains, implements [Assignment: organization-defined security policy filters] requiring fully enumerated formats that restrict data structure and content.

Note: Data structure and content restrictions reduce the range of potential malicious and/or unsanctioned content in cross-domain transactions. Security policy filters that restrict data structures include, for example, restricting file sizes and field lengths. Data content policy filters include, for example: (i) encoding formats for character sets (e.g., Universal Character Set Transformation Formats, American Standard Code for Information Interchange); (ii) restricting character data fields to only contain alpha-numeric characters; (iii) prohibiting special characters; and (iv) validating schema structures.

AC-4 (15) : DETECTION OF UNSANCTIONED INFORMATION

> The information system, when transferring information between different security domains, examines the information for the presence of [Assignment: organized-defined unsanctioned information] and prohibits the transfer of such information in accordance with the [Assignment: organization-defined security policy].

Note: Detection of unsanctioned information includes, for example, checking all information to be transferred for malicious code and dirty words.

Related Controls: *SI-3*

AC-4 (16) : INFORMATION TRANSFERS ON INTERCONNECTED SYSTEMS

> [Withdrawn: Incorporated into AC-4].

AC-4 (17) : DOMAIN AUTHENTICATION

> The information system uniquely identifies and authenticates source and destination points by [Selection (one or more): organization, system, application, individual] for information transfer.

Note: Attribution is a critical component of a security concept of operations. The ability to identify source and destination points for information flowing in information systems, allows the forensic reconstruction of events when required, and encourages policy compliance by attributing policy violations to specific organizations/individuals. Successful domain authentication requires that information system labels distinguish among systems, organizations, and individuals involved in preparing, sending, receiving, or disseminating information.

Related Controls: *IA-2*, *IA-3*, *IA-4*, *IA-5*

AC-4 (18) : SECURITY ATTRIBUTE BINDING

> The information system binds security attributes to information using [Assignment: organization-defined binding techniques] to facilitate information flow policy enforcement.

Note: Binding techniques implemented by information systems affect the strength of security attribute binding to information. Binding strength and the assurance associated with binding techniques play an important part in the trust organizations have in the information flow enforcement process. The binding techniques affect the number and degree of additional reviews required by organizations.

Related Controls: *AC-16*, *SC-16*

AC-4 (19) : VALIDATION OF METADATA

> The information system, when transferring information between different security domains, applies the same security policy filtering to metadata as it applies to data payloads.

Note: This control enhancement requires the validation of metadata and the data to which the metadata applies. Some organizations distinguish between metadata and data payloads (i.e., only the data to which the metadata is bound). Other organizations do not make such distinctions, considering metadata and the data to which the metadata applies as part of the payload. All information (including metadata and the data to which the metadata applies) is subject to filtering and inspection.

AC-4 (20) : APPROVED SOLUTIONS

> The organization employs [Assignment: organization-defined solutions in approved configurations] to control the flow of [Assignment: organization-defined information] across security domains.

Note: Organizations define approved solutions and configurations in cross-domain policies and guidance in accordance with the types of information flows across classification boundaries. The Unified Cross Domain Management Office (UCDMO) provides a baseline listing of approved cross-domain solutions.

AC-4 (21) : PHYSICAL / LOGICAL SEPARATION OF INFORMATION FLOWS

> The information system separates information flows logically or physically using [Assignment: organization-defined mechanisms and/or techniques] to accomplish [Assignment: organization-defined required separations by types of information].

Note: Enforcing the separation of information flows by type can enhance protection by ensuring that information is not commingled while in transit and by enabling flow control by transmission paths perhaps not otherwise achievable. Types of separable information include, for example, inbound and outbound communications traffic, service requests and responses, and information of differing security categories.

AC-4 (22) : ACCESS ONLY

> The information system provides access from a single device to computing platforms, applications, or data residing on multiple different security domains, while preventing any information flow between the different security domains.

Note: The information system, for example, provides a desktop for users to access each connected security domain without providing any mechanisms to allow transfer of information between the different security domains.

Control Family: ACCESS CONTROL

AC-5 : SEPARATION OF DUTIES

Priority: P1

Baseline-Impact: *MODERATE*, **HIGH**

The organization:

Note: Separation of duties addresses the potential for abuse of authorized privileges and helps to reduce the risk of malevolent activity without collusion. Separation of duties includes, for example: (i) dividing mission functions and information system support functions among different individuals and/or roles; (ii) conducting information system support functions with different individuals (e.g., system management, programming, configuration management, quality assurance and testing, and network security); and (iii) ensuring security personnel administering access control functions do not also administer audit functions.

Related Controls: *AC-3*, *AC-6*, *PE-3*, *PE-4*, *PS-2*

AC-5a.

Separates [Assignment: organization-defined duties of individuals];

AC-5b.

Documents separation of duties of individuals; and

AC-5c.

Defines information system access authorizations to support separation of duties.

Control Family: ACCESS CONTROL

AC-6 : LEAST PRIVILEGE

Priority: P1

Baseline-Impact: *MODERATE*, **HIGH**

The organization employs the principle of least privilege, allowing only authorized accesses for users (or processes acting on behalf of users) which are necessary to accomplish assigned tasks in accordance with organizational missions and business functions.

Note: Organizations employ least privilege for specific duties and information systems. The principle of least privilege is also applied to information system processes, ensuring that the processes operate at privilege levels no higher than necessary to accomplish required organizational missions/business functions. Organizations consider the creation of additional processes, roles, and information system accounts as necessary, to achieve least privilege. Organizations also apply least privilege to the development, implementation, and operation of organizational information systems.

Related Controls: *AC-2*, *AC-3*, *AC-5*, *CM-6*, *CM-7*, *PL-2*

AC-6 (1) : AUTHORIZE ACCESS TO SECURITY FUNCTIONS

Baseline-Impact: *MODERATE*, **HIGH**

> The organization explicitly authorizes access to [Assignment: organization-defined security functions (deployed in hardware, software, and firmware) and security-relevant information].

Note: Security functions include, for example, establishing system accounts, configuring access authorizations (i.e., permissions, privileges), setting events to be audited, and setting intrusion detection parameters. Security-relevant information includes, for example, filtering rules for routers/firewalls, cryptographic key management information, configuration parameters for security services, and access control lists. Explicitly authorized personnel include, for example, security administrators, system and network administrators, system security officers, system maintenance personnel, system programmers, and other privileged users.

Related Controls: *AC-17*, *AC-18*, *AC-19*

AC-6 (2) : NON-PRIVILEGED ACCESS FOR NONSECURITY FUNCTIONS

Baseline-Impact: *MODERATE*, **HIGH**

> The organization requires that users of information system accounts, or roles, with access to [Assignment: organization-defined security functions or security-relevant information], use non-privileged accounts or roles, when accessing nonsecurity functions.

Note: This control enhancement limits exposure when operating from within privileged accounts or roles. The inclusion of roles addresses situations where organizations implement access control policies such as role-based access control and where a change of role provides the same degree of assurance in the change of access authorizations for both the user and all processes acting on behalf of the user as would be provided by a change between a privileged and non-privileged account.

Related Controls: *PL-4*

AC-6 (3) : NETWORK ACCESS TO PRIVILEGED COMMANDS

Baseline-Impact: **HIGH**

> The organization authorizes network access to [Assignment: organization-defined privileged commands] only for [Assignment: organization-defined compelling operational needs] and documents the rationale for such access in the security plan for the information system.

Note: Network access is any access across a network connection in lieu of local access (i.e., user being physically present at the device).

Related Controls: *AC-17*

AC-6 (4) : SEPARATE PROCESSING DOMAINS

> The information system provides separate processing domains to enable finer-grained allocation of user privileges.

Note: Providing separate processing domains for finer-grained allocation of user privileges includes, for example: (i) using virtualization techniques to allow additional privileges within a virtual machine while restricting privileges to other virtual machines or to the underlying actual machine; (ii) employing hardware and/or software domain separation mechanisms; and (iii) implementing separate physical domains.

Related Controls: *AC-4*, *SC-3*, *SC-30*, *SC-32*

AC-6 (5) : PRIVILEGED ACCOUNTS

Baseline-Impact: *MODERATE*, **HIGH**

> The organization restricts privileged accounts on the information system to [Assignment: organization-defined personnel or roles].

Note: Privileged accounts, including super user accounts, are typically described as system administrator for various types of commercial off-the-shelf operating systems. Restricting privileged accounts to specific personnel or roles prevents day-to-day users from having access to privileged information/functions. Organizations may differentiate in the application of this control enhancement between allowed privileges for local accounts and for domain accounts provided organizations retain the ability to control information system configurations for key security parameters and as otherwise necessary to sufficiently mitigate risk.

Related Controls: *CM-6*

AC-6 (6) : PRIVILEGED ACCESS BY NON-ORGANIZATIONAL USERS

> The organization prohibits privileged access to the information system by non-organizational users.

Related Controls: *IA-8*

AC-6 (7) : REVIEW OF USER PRIVILEGES

> The organization:

Note: The need for certain assigned user privileges may change over time reflecting changes in organizational missions/business function, environments of operation, technologies, or threat. Periodic review of assigned user privileges is necessary to determine if the rationale for assigning such privileges remains valid. If the need cannot be revalidated, organizations take appropriate corrective actions.

Related Controls: *CA-7*

AC-6 (7)(a)

Reviews [Assignment: organization-defined frequency] the privileges assigned to [Assignment: organization-defined roles or classes of users] to validate the need for such privileges; and

AC-6 (7)(b)

Reassigns or removes privileges, if necessary, to correctly reflect organizational mission/business needs.

AC-6 (8) : PRIVILEGE LEVELS FOR CODE EXECUTION

The information system prevents [Assignment: organization-defined software] from executing at higher privilege levels than users executing the software.

Note: In certain situations, software applications/programs need to execute with elevated privileges to perform required functions. However, if the privileges required for execution are at a higher level than the privileges assigned to organizational users invoking such applications/programs, those users are indirectly provided with greater privileges than assigned by organizations.

AC-6 (9) : AUDITING USE OF PRIVILEGED FUNCTIONS

Baseline-Impact: *MODERATE*, **HIGH**

The information system audits the execution of privileged functions.

Note: Misuse of privileged functions, either intentionally or unintentionally by authorized users, or by unauthorized external entities that have compromised information system accounts, is a serious and ongoing concern and can have significant adverse impacts on organizations. Auditing the use of privileged functions is one way to detect such misuse, and in doing so, help mitigate the risk from insider threats and the advanced persistent threat (APT).

Related Controls: *AU-2*

AC-6 (10) : PROHIBIT NON-PRIVILEGED USERS FROM EXECUTING PRIVILEGED FUNCTIONS

Baseline-Impact: *MODERATE*, **HIGH**

The information system prevents non-privileged users from executing privileged functions to include disabling, circumventing, or altering implemented security safeguards/countermeasures.

Note: Privileged functions include, for example, establishing information system accounts, performing system integrity checks, or administering cryptographic key management activities. Non-privileged users are individuals that do not possess appropriate authorizations. Circumventing intrusion detection and prevention mechanisms or malicious code protection mechanisms are examples of privileged functions that require protection from non-privileged users.

Control Family: ACCESS CONTROL

AC-7 : UNSUCCESSFUL LOGON ATTEMPTS

Priority: P2

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The information system:

Note: This control applies regardless of whether the logon occurs via a local or network connection. Due to the potential for denial of service, automatic lockouts initiated by information systems are usually temporary and automatically release after a predetermined time period established by organizations. If a delay algorithm is selected, organizations may choose to employ different algorithms for different information system components based on the capabilities of those components. Responses to unsuccessful logon attempts may be implemented at both the operating system and the application levels.

Related Controls: *AC-2*, *AC-9*, *AC-14*, *IA-5*

AC-7a.

Enforces a limit of [Assignment: organization-defined number] consecutive invalid logon attempts by a user during a [Assignment: organization-defined time period]; and

AC-7b.

Automatically [Selection: locks the account/node for an [Assignment: organization-defined time period]; locks the account/node until released by an administrator; delays next logon prompt according to [Assignment: organization-defined delay algorithm]] when the maximum number of unsuccessful attempts is exceeded.

AC-7 (1) : AUTOMATIC ACCOUNT LOCK

[Withdrawn: Incorporated into AC-7].

AC-7 (2) : PURGE / WIPE MOBILE DEVICE

The information system purges/wipes information from [Assignment: organization-defined mobile devices] based on [Assignment: organization-defined purging/wiping requirements/techniques] after [Assignment: organization-defined number] consecutive, unsuccessful device logon attempts.

Note: This control enhancement applies only to mobile devices for which a logon occurs (e.g., personal digital assistants, smart phones, tablets). The logon is to the mobile device, not to any one account on the device. Therefore, successful logons to any accounts on mobile devices reset the unsuccessful logon count to zero. Organizations define information to be purged/wiped carefully in order to avoid over purging/wiping which may result in devices becoming unusable. Purging/wiping may be unnecessary if the information on the device is protected with sufficiently strong encryption mechanisms.

Related Controls: *AC-19*, *MP-5*, *MP-6*, *SC-13*

Control Family: ACCESS CONTROL

AC-8 : SYSTEM USE NOTIFICATION

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The information system:

Note: System use notifications can be implemented using messages or warning banners displayed before individuals log in to information systems. System use notifications are used only for access via logon interfaces with human users and are not required when such human interfaces do not exist. Organizations consider system use notification messages/banners displayed in multiple languages based on specific organizational needs and the demographics of information system users. Organizations also consult with the Office of the General Counsel for legal review and approval of warning banner content.

AC-8a.

Displays to users [Assignment: organization-defined system use notification message or banner] before granting access to the system that provides privacy and security notices consistent with applicable federal laws, Executive Orders, directives, policies, regulations, standards, and guidance and states that:

AC-8a.1.

Users are accessing a U.S. Government information system;

AC-8a.2.

Information system usage may be monitored, recorded, and subject to audit;

AC-8a.3.

Unauthorized use of the information system is prohibited and subject to criminal and civil penalties; and

AC-8a.4.

Use of the information system indicates consent to monitoring and recording;

AC-8b.

Retains the notification message or banner on the screen until users acknowledge the usage conditions and take explicit actions to log on to or further access the information system; and

AC-8c.

For publicly accessible systems:

AC-8c.1.

Displays system use information [Assignment: organization-defined conditions], before granting further access;

AC-8c.2.

Displays references, if any, to monitoring, recording, or auditing that are consistent with privacy accommodations for such systems that generally prohibit those activities; and

AC-8c.3.

Includes a description of the authorized uses of the system.

Control Family: ACCESS CONTROL

AC-9 : PREVIOUS LOGON (ACCESS) NOTIFICATION

Priority: P0

The information system notifies the user, upon successful logon (access) to the system, of the date and time of the last logon (access).

Note: This control is applicable to logons to information systems via human user interfaces and logons to systems that occur in other types of architectures (e.g., service-oriented architectures).

Related Controls: *AC-7*, *PL-4*

AC-9 (1) : UNSUCCESSFUL LOGONS

The information system notifies the user, upon successful logon/access, of the number of unsuccessful logon/access attempts since the last successful logon/access.

AC-9 (2) : SUCCESSFUL / UNSUCCESSFUL LOGONS

> The information system notifies the user of the number of [Selection: successful logons/accesses; unsuccessful logon/access attempts; both] during [Assignment: organization-defined time period].

AC-9 (3) : NOTIFICATION OF ACCOUNT CHANGES

> The information system notifies the user of changes to [Assignment: organization-defined security-related characteristics/parameters of the users account] during [Assignment: organization-defined time period].

AC-9 (4) : ADDITIONAL LOGON INFORMATION

> The information system notifies the user, upon successful logon (access), of the following additional information: [Assignment: organization-defined information to be included in addition to the date and time of the last logon (access)].

Note: This control enhancement permits organizations to specify additional information to be provided to users upon logon including, for example, the location of last logon. User location is defined as that information which can be determined by information systems, for example, IP addresses from which network logons occurred, device identifiers, or notifications of local logons.

Control Family: ACCESS CONTROL

AC-10 : CONCURRENT SESSION CONTROL

Priority: P3

Baseline-Impact: HIGH

> The information system limits the number of concurrent sessions for each [Assignment: organization-defined account and/or account type] to [Assignment: organization-defined number].

Note: Organizations may define the maximum number of concurrent sessions for information system accounts globally, by account type (e.g., privileged user, non-privileged user, domain, specific application), by account, or a combination. For example, organizations may limit the number of concurrent sessions for system administrators or individuals working in particularly sensitive domains or mission-critical applications. This control addresses concurrent sessions for information system accounts and does not address concurrent sessions by single users via multiple system accounts.

Control Family: ACCESS CONTROL

AC-11 : SESSION LOCK

Priority: P3

Baseline-Impact: *MODERATE*, **HIGH**

The information system:

Note: Session locks are temporary actions taken when users stop work and move away from the immediate vicinity of information systems but do not want to log out because of the temporary nature of their absences. Session locks are implemented where session activities can be determined. This is typically at the operating system level, but can also be at the application level. Session locks are not an acceptable substitute for logging out of information systems, for example, if organizations require users to log out at the end of workdays.

Related Controls: *AC-7*

AC-11a.

Prevents further access to the system by initiating a session lock after [Assignment: organization-defined time period] of inactivity or upon receiving a request from a user; and

AC-11b.

Retains the session lock until the user reestablishes access using established identification and authentication procedures.

AC-11 (1) : PATTERN-HIDING DISPLAYS

Baseline-Impact: *MODERATE*, **HIGH**

The information system conceals, via the session lock, information previously visible on the display with a publicly viewable image.

Note: Publicly viewable images can include static or dynamic images, for example, patterns used with screen savers, photographic images, solid colors, clock, battery life indicator, or a blank screen, with the additional caveat that none of the images convey sensitive information.

Control Family: ACCESS CONTROL

AC-12 : SESSION TERMINATION

Priority: P2

Baseline-Impact: *MODERATE*, **HIGH**

> The information system automatically terminates a user session after [Assignment: organization-defined conditions or trigger events requiring session disconnect].

Note: This control addresses the termination of user-initiated logical sessions in contrast to SC-10 which addresses the termination of network connections that are associated with communications sessions (i.e., network disconnect). A logical session (for local, network, and remote access) is initiated whenever a user (or process acting on behalf of a user) accesses an organizational information system. Such user sessions can be terminated (and thus terminate user access) without terminating network sessions. Session termination terminates all processes associated with a users logical session except those processes that are specifically created by the user (i.e., session owner) to continue after the session is terminated. Conditions or trigger events requiring automatic session termination can include, for example, organization-defined periods of user inactivity, targeted responses to certain types of incidents, time-of-day restrictions on information system use.

Related Controls: *SC-10*, *SC-23*

AC-12 (1) : USER-INITIATED LOGOUTS / MESSAGE DISPLAYS

> The information system:

Note: Information resources to which users gain access via authentication include, for example, local workstations, databases, and password-protected websites/web-based services. Logout messages for web page access, for example, can be displayed after authenticated sessions have been terminated. However, for some types of interactive sessions including, for example, file transfer protocol (FTP) sessions, information systems typically send logout messages as final messages prior to terminating sessions.

AC-12 (1)(a)

> Provides a logout capability for user-initiated communications sessions whenever authentication is used to gain access to [Assignment: organization-defined information resources]; and

AC-12 (1)(b)

> Displays an explicit logout message to users indicating the reliable termination of authenticated communications sessions.

Control Family: ACCESS CONTROL

AC-13 : SUPERVISION AND REVIEW - ACCESS CONTROL

[Withdrawn: Incorporated into AC-2 and AU-6].

Control Family: ACCESS CONTROL

AC-14 : PERMITTED ACTIONS WITHOUT IDENTIFICATION OR AUTHENTICATION

Priority: P3

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: This control addresses situations in which organizations determine that no identification or authentication is required in organizational information systems. Organizations may allow a limited number of user actions without identification or authentication including, for example, when individuals access public websites or other publicly accessible federal information systems, when individuals use mobile phones to receive calls, or when facsimiles are received. Organizations also identify actions that normally require identification or authentication but may under certain circumstances (e.g., emergencies), allow identification or authentication mechanisms to be bypassed. Such bypasses may occur, for example, via a software-readable physical switch that commands bypass of the logon functionality and is protected from accidental or unmonitored use. This control does not apply to situations where identification and authentication have already occurred and are not repeated, but rather to situations where identification and authentication have not yet occurred. Organizations may decide that there are no user actions that can be performed on organizational information systems without identification and authentication and thus, the values for assignment statements can be none.

Related Controls: *CP-2*, *IA-2*

AC-14a.

Identifies [Assignment: organization-defined user actions] that can be performed on the information system without identification or authentication consistent with organizational missions/business functions; and

AC-14b.

Documents and provides supporting rationale in the security plan for the information system, user actions not requiring identification or authentication.

AC-14 (1) : NECESSARY USES

[Withdrawn: Incorporated into AC-14].

Control Family: ACCESS CONTROL

AC-15 : AUTOMATED MARKING

[Withdrawn: Incorporated into MP-3].

Control Family: ACCESS CONTROL

AC-16 : SECURITY ATTRIBUTES

Priority: P0

The organization:

Note:

Information is represented internally within information systems using abstractions known as data structures. Internal data structures can represent different types of entities, both active and passive. Active entities, also known as subjects, are typically associated with individuals, devices, or processes acting on behalf of individuals. Passive entities, also known as objects, are typically associated with data structures such as records, buffers, tables, files, inter-process pipes, and communications ports. Security attributes, a form of metadata, are abstractions representing the basic properties or characteristics of active and passive entities with respect to safeguarding information. These attributes may be associated with active entities (i.e., subjects) that have the potential to send or receive information, to cause information to flow among objects, or to change the information system state. These attributes may also be associated with passive entities (i.e., objects) that contain or receive information. The association of security attributes to subjects and objects is referred to as binding and is typically inclusive of setting the attribute value and the attribute type. Security attributes when bound to data/information, enables the enforcement of information security policies for access control and information flow control, either through organizational processes or information system functions or mechanisms. The content or assigned values of security attributes can directly affect the ability of individuals to access organizational information.

Organizations can define the types of attributes needed for selected information systems to support missions/business functions. There is potentially a wide range of values that can be assigned to any given security attribute. Release markings could include, for example, US only, NATO, or NOFORN (not releasable to foreign nationals). By specifying permitted attribute ranges and values, organizations can ensure that the security attribute values are meaningful and relevant. The term security labeling refers to the association of security attributes with subjects and objects represented by internal data structures within organizational information systems, to enable information system-based enforcement of information security policies. Security labels include, for example, access authorizations, data life cycle protection (i.e., encryption and data expiration), nationality, affiliation as contractor, and classification of information in accordance with legal and compliance requirements. The term security marking refers to the association of security attributes with objects in a human-readable form, to enable organizational process-based enforcement of information security policies. The AC-16 base control represents the requirement for user-based attribute association (marking). The enhancements to AC-16 represent additional requirements including information system-based attribute association (labeling). Types of attributes include, for example, classification level for objects and clearance (access authorization) level for subjects. An example of a value for both of these attribute types is Top Secret.

Related Controls: *AC-3*, *AC-4*, *AC-6*, *AC-21*, *AU-2*, *AU-10*, *SC-16*, *MP-3*

AC-16a.

Provides the means to associate [Assignment: organization-defined types of security attributes] having [Assignment: organization-defined security attribute values] with information in storage, in process, and/or in transmission;

AC-16b.

Ensures that the security attribute associations are made and retained with the information;

AC-16c.

Establishes the permitted [Assignment: organization-defined security attributes] for [Assignment: organization-defined information systems]; and

AC-16d.

Determines the permitted [Assignment: organization-defined values or ranges] for each of the established security attributes.

AC-16 (1) : DYNAMIC ATTRIBUTE ASSOCIATION

The information system dynamically associates security attributes with [Assignment: organization-defined subjects and objects] in accordance with [Assignment: organization-defined security policies] as information is created and combined.

Note: Dynamic association of security attributes is appropriate whenever the security characteristics of information changes over time. Security attributes may change, for example, due to information aggregation issues (i.e., the security characteristics of individual information elements are different from the combined elements), changes in individual access authorizations (i.e., privileges), and changes in the security category of information.

Related Controls: *AC-4*

AC-16 (2) : ATTRIBUTE VALUE CHANGES BY AUTHORIZED INDIVIDUALS

The information system provides authorized individuals (or processes acting on behalf of individuals) the capability to define or change the value of associated security attributes.

Note: The content or assigned values of security attributes can directly affect the ability of individuals to access organizational information. Therefore, it is important for information systems to be able to limit the ability to create or modify security attributes to authorized individuals.

Related Controls: *AC-6*, *AU-2*

AC-16 (3) : MAINTENANCE OF ATTRIBUTE ASSOCIATIONS BY INFORMATION SYSTEM

> The information system maintains the association and integrity of [Assignment: organization-defined security attributes] to [Assignment: organization-defined subjects and objects].

Note: Maintaining the association and integrity of security attributes to subjects and objects with sufficient assurance helps to ensure that the attribute associations can be used as the basis of automated policy actions. Automated policy actions include, for example, access control decisions or information flow control decisions.

AC-16 (4) : ASSOCIATION OF ATTRIBUTES BY AUTHORIZED INDIVIDUALS

> The information system supports the association of [Assignment: organization-defined security attributes] with [Assignment: organization-defined subjects and objects] by authorized individuals (or processes acting on behalf of individuals).

Note: The support provided by information systems can vary to include: (i) prompting users to select specific security attributes to be associated with specific information objects; (ii) employing automated mechanisms for categorizing information with appropriate attributes based on defined policies; or (iii) ensuring that the combination of selected security attributes selected is valid. Organizations consider the creation, deletion, or modification of security attributes when defining auditable events.

AC-16 (5) : ATTRIBUTE DISPLAYS FOR OUTPUT DEVICES

> The information system displays security attributes in human-readable form on each object that the system transmits to output devices to identify [Assignment: organization-identified special dissemination, handling, or distribution instructions] using [Assignment: organization-identified human-readable, standard naming conventions].

Note: Information system outputs include, for example, pages, screens, or equivalent. Information system output devices include, for example, printers and video displays on computer workstations, notebook computers, and personal digital assistants.

AC-16 (6) : MAINTENANCE OF ATTRIBUTE ASSOCIATION BY ORGANIZATION

> The organization allows personnel to associate, and maintain the association of [Assignment: organization-defined security attributes] with [Assignment: organization-defined subjects and objects] in accordance with [Assignment: organization-defined security policies].

Note: This control enhancement requires individual users (as opposed to the information system) to maintain associations of security attributes with subjects and objects.

AC-16 (7) : CONSISTENT ATTRIBUTE INTERPRETATION

> The organization provides a consistent interpretation of security attributes transmitted between distributed information system components.

Note: In order to enforce security policies across multiple components in distributed information systems (e.g., distributed database management systems, cloud-based systems, and service-oriented architectures), organizations provide a consistent interpretation of security attributes that are used in access enforcement and flow enforcement decisions. Organizations establish agreements and processes to ensure that all distributed information system components implement security attributes with consistent interpretations in automated access/flow enforcement actions.

AC-16 (8) : ASSOCIATION TECHNIQUES / TECHNOLOGIES

> The information system implements [Assignment: organization-defined techniques or technologies] with [Assignment: organization-defined level of assurance] in associating security attributes to information.

Note: The association (i.e., binding) of security attributes to information within information systems is of significant importance with regard to conducting automated access enforcement and flow enforcement actions. The association of such security attributes can be accomplished with technologies/techniques providing different levels of assurance. For example, information systems can cryptographically bind security attributes to information using digital signatures with the supporting cryptographic keys protected by hardware devices (sometimes known as hardware roots of trust).

AC-16 (9) : ATTRIBUTE REASSIGNMENT

> The organization ensures that security attributes associated with information are reassigned only via re-grading mechanisms validated using [Assignment: organization-defined techniques or procedures].

Note: Validated re-grading mechanisms are employed by organizations to provide the requisite levels of assurance for security attribute reassignment activities. The validation is facilitated by ensuring that re-grading mechanisms are single purpose and of limited function. Since security attribute reassignments can affect security policy enforcement actions (e.g., access/flow enforcement decisions), using trustworthy re-grading mechanisms is necessary to ensure that such mechanisms perform in a consistent/correct mode of operation.

AC-16 (10) : ATTRIBUTE CONFIGURATION BY AUTHORIZED INDIVIDUALS

> The information system provides authorized individuals the capability to define or change the type and value of security attributes available for association with subjects and objects.

Note: The content or assigned values of security attributes can directly affect the ability of individuals to access organizational information. Therefore, it is important for information systems to be able to limit the ability to create or modify security attributes to authorized individuals only.

Control Family: ACCESS CONTROL

AC-17 : REMOTE ACCESS

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: Remote access is access to organizational information systems by users (or processes acting on behalf of users) communicating through external networks (e.g., the Internet). Remote access methods include, for example, dial-up, broadband, and wireless. Organizations often employ encrypted virtual private networks (VPNs) to enhance confidentiality and integrity over remote connections. The use of encrypted VPNs does not make the access non-remote; however, the use of VPNs, when adequately provisioned with appropriate security controls (e.g., employing appropriate encryption techniques for confidentiality and integrity protection) may provide sufficient assurance to the organization that it can effectively treat such connections as internal networks. Still, VPN connections traverse external networks, and the encrypted VPN does not enhance the availability of remote connections. Also, VPNs with encrypted tunnels can affect the organizational capability to adequately monitor network communications traffic for malicious code. Remote access controls apply to information systems other than public web servers or systems designed for public access. This control addresses authorization prior to allowing remote access without specifying the formats for such authorization. While organizations may use interconnection security agreements to authorize remote access connections, such agreements are not required by this control. Enforcing access restrictions for remote connections is addressed in AC-3.

Related Controls: *AC-2*, *AC-3*, *AC-18*, *AC-19*, *AC-20*, *CA-3*, *CA-7*, *CM-8*, *IA-2*, *IA-3*, *IA-8*, *MA-4*, *PE-17*, *PL-4*, *SC-10*, *SI-4*

AC-17a.

Establishes and documents usage restrictions, configuration/connection requirements, and implementation guidance for each type of remote access allowed; and

AC-17b.

Authorizes remote access to the information system prior to allowing such connections.

AC-17 (1) : AUTOMATED MONITORING / CONTROL

Baseline-Impact: *MODERATE*, **HIGH**

The information system monitors and controls remote access methods.

Note: Automated monitoring and control of remote access sessions allows organizations to detect cyber attacks and also ensure ongoing compliance with remote access policies by auditing connection activities of remote users on a variety of information system components (e.g., servers, workstations, notebook computers, smart phones, and tablets).

Related Controls: *AU-2*, *AU-12*

AC-17 (2) : PROTECTION OF CONFIDENTIALITY / INTEGRITY USING ENCRYPTION

Baseline-Impact: *MODERATE*, **HIGH**

The information system implements cryptographic mechanisms to protect the confidentiality and integrity of remote access sessions.

Note: The encryption strength of mechanism is selected based on the security categorization of the information.

Related Controls: *SC-8*, *SC-12*, *SC-13*

AC-17 (3) : MANAGED ACCESS CONTROL POINTS

Baseline-Impact: *MODERATE*, **HIGH**

The information system routes all remote accesses through [Assignment: organization-defined number] managed network access control points.

Note: Limiting the number of access control points for remote accesses reduces the attack surface for organizations. Organizations consider the Trusted Internet Connections (TIC) initiative requirements for external network connections.

Related Controls: *SC-7*

AC-17 (4) : PRIVILEGED COMMANDS / ACCESS

Baseline-Impact: *MODERATE*, **HIGH**

The organization:

Related Controls: *AC-6*

AC-17 (4)(a)

Authorizes the execution of privileged commands and access to security-relevant information via remote access only for [Assignment: organization-defined needs]; and

AC-17 (4)(b)

Documents the rationale for such access in the security plan for the information system.

AC-17 (5) : MONITORING FOR UNAUTHORIZED CONNECTIONS

[Withdrawn: Incorporated into SI-4].

AC-17 (6) : PROTECTION OF INFORMATION

The organization ensures that users protect information about remote access mechanisms from unauthorized use and disclosure.

Related Controls: *AT-2*, *AT-3*, *PS-6*

AC-17 (7) : ADDITIONAL PROTECTION FOR SECURITY FUNCTION ACCESS

[Withdrawn: Incorporated into AC-3 (10)].

AC-17 (8) : DISABLE NONSECURE NETWORK PROTOCOLS

[Withdrawn: Incorporated into CM-7].

AC-17 (9) : DISCONNECT / DISABLE ACCESS

The organization provides the capability to expeditiously disconnect or disable remote access to the information system within [Assignment: organization-defined time period].

Note: This control enhancement requires organizations to have the capability to rapidly disconnect current users remotely accessing the information system and/or disable further remote access. The speed of disconnect or disablement varies based on the criticality of missions/business functions and the need to eliminate immediate or future remote access to organizational information systems.

Control Family: ACCESS CONTROL

AC-18 : WIRELESS ACCESS

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: Wireless technologies include, for example, microwave, packet radio (UHF/VHF), 802.11x, and Bluetooth. Wireless networks use authentication protocols (e.g., EAP/TLS, PEAP), which provide credential protection and mutual authentication.

Related Controls: *AC-2*, *AC-3*, *AC-17*, *AC-19*, *CA-3*, *CA-7*, *CM-8*, *IA-2*, *IA-3*, *IA-8*, *PL-4*, *SI-4*

AC-18a.

Establishes usage restrictions, configuration/connection requirements, and implementation guidance for wireless access; and

AC-18b.

Authorizes wireless access to the information system prior to allowing such connections.

AC-18 (1) : AUTHENTICATION AND ENCRYPTION

Baseline-Impact: *MODERATE*, **HIGH**

The information system protects wireless access to the system using authentication of [Selection (one or more): users; devices] and encryption.

Related Controls: *SC-8*, *SC-13*

AC-18 (2) : MONITORING UNAUTHORIZED CONNECTIONS

[Withdrawn: Incorporated into SI-4].

AC-18 (3) : DISABLE WIRELESS NETWORKING

The organization disables, when not intended for use, wireless networking capabilities internally embedded within information system components prior to issuance and deployment.

Related Controls: *AC-19*

AC-18 (4) : RESTRICT CONFIGURATIONS BY USERS

Baseline-Impact: **HIGH**

The organization identifies and explicitly authorizes users allowed to independently configure wireless networking capabilities.

Note: Organizational authorizations to allow selected users to configure wireless networking capability are enforced in part, by the access enforcement mechanisms employed within organizational information systems.

Related Controls: *AC-3*, *SC-15*

AC-18 (5) : ANTENNAS / TRANSMISSION POWER LEVELS

Baseline-Impact: **HIGH**

The organization selects radio antennas and calibrates transmission power levels to reduce the probability that usable signals can be received outside of organization-controlled boundaries.

Note: Actions that may be taken by organizations to limit unauthorized use of wireless communications outside of organization-controlled boundaries include, for example: (i) reducing the power of wireless transmissions so that the transmissions are less likely to emit a signal that can be used by adversaries outside of the physical perimeters of organizations; (ii) employing measures such as TEMPEST to control wireless emanations; and (iii) using directional/beam forming antennas that reduce the likelihood that unintended receivers will be able to intercept signals. Prior to taking such actions, organizations can conduct periodic wireless surveys to understand the radio frequency profile of organizational information systems as well as other systems that may be operating in the area.

Related Controls: *PE-19*

Control Family: ACCESS CONTROL

AC-19 : ACCESS CONTROL FOR MOBILE DEVICES

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: A mobile device is a computing device that: (i) has a small form factor such that it can easily be carried by a single individual; (ii) is designed to operate without a physical connection (e.g., wirelessly transmit or receive information); (iii) possesses local, non-removable or removable data storage; and (iv) includes a self-contained power source. Mobile devices may also include voice communication capabilities, on-board sensors that allow the device to capture information, and/or built-in features for synchronizing local data with remote locations. Examples include smart phones, E-readers, and tablets. Mobile devices are typically associated with a single individual and the device is usually in close proximity to the individual; however, the degree of proximity can vary depending upon on the form factor and size of the device. The processing, storage, and transmission capability of the mobile device may be comparable to or merely a subset of desktop systems, depending upon the nature and intended purpose of the device. Due to the large variety of mobile devices with different technical characteristics and capabilities, organizational restrictions may vary for the different classes/types of such devices. Usage restrictions and specific implementation guidance for mobile devices include, for example, configuration management, device identification and authentication, implementation of mandatory protective software (e.g., malicious code detection, firewall), scanning devices for malicious code, updating virus protection software, scanning for critical software updates and patches, conducting primary operating system (and possibly other resident software) integrity checks, and disabling unnecessary hardware (e.g., wireless, infrared). Organizations are cautioned that the need to provide adequate security for mobile devices goes beyond the requirements in this control. Many safeguards and countermeasures for mobile devices are reflected in other security controls in the catalog allocated in the initial control baselines as starting points for the development of security plans and overlays using the tailoring process. There may also be some degree of overlap in the requirements articulated by the security controls within the different families of controls. AC-20 addresses mobile devices that are not organization-controlled.

Related Controls: *AC-3*, *AC-7*, *AC-18*, *AC-20*, *CA-9*, *CM-2*, *IA-2*, *IA-3*, *MP-2*, *MP-4*, *MP-5*, *PL-4*, *SC-7*, *SC-43*, *SI-3*, *SI-4*

AC-19a.

Establishes usage restrictions, configuration requirements, connection requirements, and implementation guidance for organization-controlled mobile devices; and

AC-19b.

Authorizes the connection of mobile devices to organizational information systems.

AC-19 (1) : USE OF WRITABLE / PORTABLE STORAGE DEVICES

[Withdrawn: Incorporated into MP-7].

AC-19 (2) : USE OF PERSONALLY OWNED PORTABLE STORAGE DEVICES

[Withdrawn: Incorporated into MP-7].

AC-19 (3) : USE OF PORTABLE STORAGE DEVICES WITH NO IDENTIFIABLE OWNER

[Withdrawn: Incorporated into MP-7].

AC-19 (4) : RESTRICTIONS FOR CLASSIFIED INFORMATION

The organization:

Related Controls: *CA-6*, *IR-4*

AC-19 (4)(a)

Prohibits the use of unclassified mobile devices in facilities containing information systems processing, storing, or transmitting classified information unless specifically permitted by the authorizing official; and

AC-19 (4)(b)

Enforces the following restrictions on individuals permitted by the authorizing official to use unclassified mobile devices in facilities containing information systems processing, storing, or transmitting classified information:

AC-19 (4)(b)(1)

Connection of unclassified mobile devices to classified information systems is prohibited;

AC-19 (4)(b)(2)

Connection of unclassified mobile devices to unclassified information systems requires approval from the authorizing official;

AC-19 (4)(b)(3)

Use of internal or external modems or wireless interfaces within the unclassified mobile devices is prohibited; and

AC-19 (4)(b)(4)

Unclassified mobile devices and the information stored on those devices are subject to random reviews and inspections by [Assignment: organization-defined security officials], and if classified information is found, the incident handling policy is followed.

AC-19 (4)(c)

Restricts the connection of classified mobile devices to classified information systems in accordance with [Assignment: organization-defined security policies].

AC-19 (5) : FULL DEVICE / CONTAINER-BASED ENCRYPTION

Baseline-Impact: *MODERATE*, **HIGH**

The organization employs [Selection: full-device encryption; container encryption] to protect the confidentiality and integrity of information on [Assignment: organization-defined mobile devices].

Note: Container-based encryption provides a more fine-grained approach to the encryption of data/information on mobile devices, including for example, encrypting selected data structures such as files, records, or fields.

Related Controls: *MP-5*, *SC-13*, *SC-28*

Control Family: ACCESS CONTROL

AC-20 : USE OF EXTERNAL INFORMATION SYSTEMS

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization establishes terms and conditions, consistent with any trust relationships established with other organizations owning, operating, and/or maintaining external information systems, allowing authorized individuals to:

Note:

External information systems are information systems or components of information systems that are outside of the authorization boundary established by organizations and for which organizations typically have no direct supervision and authority over the application of required security controls or the assessment of control effectiveness. External information systems include, for example: (i) personally owned information systems/devices (e.g., notebook computers, smart phones, tablets, personal digital assistants); (ii) privately owned computing and communications devices resident in commercial or public facilities (e.g., hotels, train stations, convention centers, shopping malls, or airports); (iii) information systems owned or controlled by nonfederal governmental organizations; and (iv) federal information systems that are not owned by, operated by, or under the direct supervision and authority of organizations. This control also addresses the use of external information systems for the processing, storage, or transmission of organizational information, including, for example, accessing cloud services (e.g., infrastructure as a service, platform as a service, or software as a service) from organizational information systems.

For some external information systems (i.e., information systems operated by other federal agencies, including organizations subordinate to those agencies), the trust relationships that have been established between those organizations and the originating organization may be such, that no explicit terms and conditions are required. Information systems within these organizations would not be considered external. These situations occur when, for example, there are pre-existing sharing/trust agreements (either implicit or explicit) established between federal agencies or organizations subordinate to those agencies, or when such trust agreements are specified by applicable laws, Executive Orders, directives, or policies. Authorized individuals include, for example, organizational personnel, contractors, or other individuals with authorized access to organizational information systems and over which organizations have the authority to impose rules of behavior with regard to system access. Restrictions that organizations impose on authorized individuals need not be uniform, as those restrictions may vary depending upon the trust relationships between organizations. Therefore, organizations may choose to impose different security restrictions on contractors than on state, local, or tribal governments. This control does not apply to the use of external information systems to access public interfaces to organizational information systems (e.g., individuals accessing federal information through www.usa.gov). Organizations establish terms and conditions for the use of external information systems in accordance with organizational security policies and procedures. Terms and conditions address as a minimum: types of applications that can be accessed on organizational information systems from external information systems; and the highest security category of information that can be processed, stored, or transmitted on external information systems. If terms and conditions with the owners of external information systems cannot be established, organizations may impose restrictions on organizational personnel using those external systems.

Related Controls: *AC-3*, *AC-17*, *AC-19*, *CA-3*, *PL-4*, *SA-9*

AC-20a.

Access the information system from external information systems; and

AC-20b.

Process, store, or transmit organization-controlled information using external information systems.

AC-20 (1) : LIMITS ON AUTHORIZED USE

Baseline-Impact: *MODERATE*, **HIGH**

The organization permits authorized individuals to use an external information system to access the information system or to process, store, or transmit organization-controlled information only when the organization:

Note: This control enhancement recognizes that there are circumstances where individuals using external information systems (e.g., contractors, coalition partners) need to access organizational information systems. In those situations, organizations need confidence that the external information systems contain the necessary security safeguards (i.e., security controls), so as not to compromise, damage, or otherwise harm organizational information systems. Verification that the required security controls have been implemented can be achieved, for example, by third-party, independent assessments, attestations, or other means, depending on the confidence level required by organizations.

Related Controls: *CA-2*

AC-20 (1)(a)

Verifies the implementation of required security controls on the external system as specified in the organizations information security policy and security plan; or

AC-20 (1)(b)

Retains approved information system connection or processing agreements with the organizational entity hosting the external information system.

AC-20 (2) : PORTABLE STORAGE DEVICES

Baseline-Impact: *MODERATE*, **HIGH**

The organization [Selection: restricts; prohibits] the use of organization-controlled portable storage devices by authorized individuals on external information systems.

Note: Limits on the use of organization-controlled portable storage devices in external information systems include, for example, complete prohibition of the use of such devices or restrictions on how the devices may be used and under what conditions the devices may be used.

AC-20 (3) : NON-ORGANIZATIONALLY OWNED SYSTEMS / COMPONENTS / DEVICES

The organization [Selection: restricts; prohibits] the use of non-organizationally owned information systems, system components, or devices to process, store, or transmit organizational information.

Note: Non-organizationally owned devices include devices owned by other organizations (e.g., federal/state agencies, contractors) and personally owned devices. There are risks to using non-organizationally owned devices. In some cases, the risk is sufficiently high as to prohibit such use. In other cases, it may be such that the use of non-organizationally owned devices is allowed but restricted in some way. Restrictions include, for example: (i) requiring the implementation of organization-approved security controls prior to authorizing such connections; (ii) limiting access to certain types of information, services, or applications; (iii) using virtualization techniques to limit processing and storage activities to servers or other system components provisioned by the organization; and (iv) agreeing to terms and conditions for usage. For personally owned devices, organizations consult with the Office of the General Counsel regarding legal issues associated with using such devices in operational environments, including, for example, requirements for conducting forensic analyses during investigations after an incident.

AC-20 (4) : NETWORK ACCESSIBLE STORAGE DEVICES

> The organization prohibits the use of [Assignment: organization-defined network accessible storage devices] in external information systems.

Note: Network accessible storage devices in external information systems include, for example, online storage devices in public, hybrid, or community cloud-based systems.

Control Family: ACCESS CONTROL

AC-21 : INFORMATION SHARING

Priority: P2

Baseline-Impact: *MODERATE*, **HIGH**

> The organization:

Note: This control applies to information that may be restricted in some manner (e.g., privileged medical information, contract-sensitive information, proprietary information, personally identifiable information, classified information related to special access programs or compartments) based on some formal or administrative determination. Depending on the particular information-sharing circumstances, sharing partners may be defined at the individual, group, or organizational level. Information may be defined by content, type, security category, or special access program/compartment.

Related Controls: *AC-3*

AC-21a.

> Facilitates information sharing by enabling authorized users to determine whether access authorizations assigned to the sharing partner match the access restrictions on the information for [Assignment: organization-defined information sharing circumstances where user discretion is required]; and

AC-21b.

Employs [Assignment: organization-defined automated mechanisms or manual processes] to assist users in making information sharing/collaboration decisions.

AC-21 (1) : AUTOMATED DECISION SUPPORT

The information system enforces information-sharing decisions by authorized users based on access authorizations of sharing partners and access restrictions on information to be shared.

AC-21 (2) : INFORMATION SEARCH AND RETRIEVAL

The information system implements information search and retrieval services that enforce [Assignment: organization-defined information sharing restrictions].

Control Family: ACCESS CONTROL

AC-22 : PUBLICLY ACCESSIBLE CONTENT

Priority: P3

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: In accordance with federal laws, Executive Orders, directives, policies, regulations, standards, and/or guidance, the general public is not authorized access to nonpublic information (e.g., information protected under the Privacy Act and proprietary information). This control addresses information systems that are controlled by the organization and accessible to the general public, typically without identification or authentication. The posting of information on non-organization information systems is covered by organizational policy.

Related Controls: *AC-3*, *AC-4*, *AT-2*, *AT-3*, *AU-13*

AC-22a.

Designates individuals authorized to post information onto a publicly accessible information system;

AC-22b.

Trains authorized individuals to ensure that publicly accessible information does not contain nonpublic information;

AC-22c.

Reviews the proposed content of information prior to posting onto the publicly accessible information system to ensure that nonpublic information is not included; and

AC-22d.

Reviews the content on the publicly accessible information system for nonpublic information [Assignment: organization-defined frequency] and removes such information, if discovered.

Control Family: ACCESS CONTROL

AC-23 : DATA MINING PROTECTION

Priority: P0

The organization employs [Assignment: organization-defined data mining prevention and detection techniques] for [Assignment: organization-defined data storage objects] to adequately detect and protect against data mining.

Note: Data storage objects include, for example, databases, database records, and database fields. Data mining prevention and detection techniques include, for example: (i) limiting the types of responses provided to database queries; (ii) limiting the number/frequency of database queries to increase the work factor needed to determine the contents of such databases; and (iii) notifying organizational personnel when atypical database queries or accesses occur. This control focuses on the protection of organizational information from data mining while such information resides in organizational data stores. In contrast, AU-13 focuses on monitoring for organizational information that may have been mined or otherwise obtained from data stores and is now available as open source information residing on external sites, for example, through social networking or social media websites.

Control Family: ACCESS CONTROL

AC-24 : ACCESS CONTROL DECISIONS

Priority: P0

The organization establishes procedures to ensure [Assignment: organization-defined access control decisions] are applied to each access request prior to access enforcement.

Note: Access control decisions (also known as authorization decisions) occur when authorization information is applied to specific accesses. In contrast, access enforcement occurs when information systems enforce access control decisions. While it is very common to have access control decisions and access enforcement implemented by the same entity, it is not required and it is not always an optimal implementation choice. For some architectures and distributed information systems, different entities may perform access control decisions and access enforcement.

AC-24 (1) : TRANSMIT ACCESS AUTHORIZATION INFORMATION

> The information system transmits [Assignment: organization-defined access authorization information] using [Assignment: organization-defined security safeguards] to [Assignment: organization-defined information systems] that enforce access control decisions.

Note: In distributed information systems, authorization processes and access control decisions may occur in separate parts of the systems. In such instances, authorization information is transmitted securely so timely access control decisions can be enforced at the appropriate locations. To support the access control decisions, it may be necessary to transmit as part of the access authorization information, supporting security attributes. This is due to the fact that in distributed information systems, there are various access control decisions that need to be made and different entities (e.g., services) make these decisions in a serial fashion, each requiring some security attributes to make the decisions. Protecting access authorization information (i.e., access control decisions) ensures that such information cannot be altered, spoofed, or otherwise compromised during transmission.

AC-24 (2) : NO USER OR PROCESS IDENTITY

> The information system enforces access control decisions based on [Assignment: organization-defined security attributes] that do not include the identity of the user or process acting on behalf of the user.

Note: In certain situations, it is important that access control decisions can be made without information regarding the identity of the users issuing the requests. These are generally instances where preserving individual privacy is of paramount importance. In other situations, user identification information is simply not needed for access control decisions and, especially in the case of distributed information systems, transmitting such information with the needed degree of assurance may be very expensive or difficult to accomplish.

Control Family: ACCESS CONTROL

AC-25 : REFERENCE MONITOR

Priority: P0

> The information system implements a reference monitor for [Assignment: organization-defined access control policies] that is tamperproof, always invoked, and small enough to be subject to analysis and testing, the completeness of which can be assured.

Note: Information is represented internally within information systems using abstractions known as data structures. Internal data structures can represent different types of entities, both active and passive. Active entities, also known as subjects, are typically associated with individuals, devices, or processes acting on behalf of individuals. Passive entities, also known as objects, are typically associated with data structures such as records, buffers, tables, files, interprocess pipes, and communications ports. Reference monitors typically enforce mandatory access control policiesa type of access control that restricts access to objects based on the identity of subjects or groups to which the subjects belong. The access controls are mandatory because subjects with certain privileges (i.e., access permissions) are restricted from passing those privileges on to any other subjects, either directly or indirectlythat is, the information system strictly enforces the access control policy based on the rule set established by the policy. The tamperproof property of the reference monitor prevents adversaries from compromising the functioning of the mechanism. The always invoked property prevents adversaries from bypassing the mechanism and hence violating the security policy. The smallness property helps to ensure the completeness in the analysis and testing of the mechanism to detect weaknesses or deficiencies (i.e., latent flaws) that would prevent the enforcement of the security policy.

Related Controls: *AC-3*, *AC-16*, *SC-3*, *SC-39*

Control Family: AWARENESS AND TRAINING

AT-1 : SECURITY AWARENESS AND TRAINING POLICY AND PROCEDURES

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: This control addresses the establishment of policy and procedures for the effective implementation of selected security controls and control enhancements in the AT family. Policy and procedures reflect applicable federal laws, Executive Orders, directives, regulations, policies, standards, and guidance. Security program policies and procedures at the organization level may make the need for system-specific policies and procedures unnecessary. The policy can be included as part of the general information security policy for organizations or conversely, can be represented by multiple policies reflecting the complex nature of certain organizations. The procedures can be established for the security program in general and for particular information systems, if needed. The organizational risk management strategy is a key factor in establishing policy and procedures.

Related Controls: *PM-9*

AT-1a.

Develops, documents, and disseminates to [Assignment: organization-defined personnel or roles]:

AT-1a.1.

A security awareness and training policy that addresses purpose, scope, roles, responsibilities, management commitment, coordination among organizational entities, and compliance; and

AT-1a.2.

Procedures to facilitate the implementation of the security awareness and training policy and associated security awareness and training controls; and

AT-1b.

Reviews and updates the current:

AT-1b.1.

Security awareness and training policy [Assignment: organization-defined frequency]; and

AT-1b.2.

Security awareness and training procedures [Assignment: organization-defined frequency].

Control Family: AWARENESS AND TRAINING

AT-2 : SECURITY AWARENESS TRAINING

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization provides basic security awareness training to information system users (including managers, senior executives, and contractors):

Note: Organizations determine the appropriate content of security awareness training and security awareness techniques based on the specific organizational requirements and the information systems to which personnel have authorized access. The content includes a basic understanding of the need for information security and user actions to maintain security and to respond to suspected security incidents. The content also addresses awareness of the need for operations security. Security awareness techniques can include, for example, displaying posters, offering supplies inscribed with security reminders, generating email advisories/notices from senior organizational officials, displaying logon screen messages, and conducting information security awareness events.

Related Controls: *AT-3*, *AT-4*, *PL-4*

AT-2a.

As part of initial training for new users;

AT-2b.

When required by information system changes; and

AT-2c.

[Assignment: organization-defined frequency] thereafter.

AT-2 (1) : PRACTICAL EXERCISES

The organization includes practical exercises in security awareness training that simulate actual cyber attacks.

Note: Practical exercises may include, for example, no-notice social engineering attempts to collect information, gain unauthorized access, or simulate the adverse impact of opening malicious email attachments or invoking, via spear phishing attacks, malicious web links.

Related Controls: *CA-2*, *CA-7*, *CP-4*, *IR-3*

AT-2 (2) : INSIDER THREAT

Baseline-Impact: *MODERATE*, **HIGH**

The organization includes security awareness training on recognizing and reporting potential indicators of insider threat.

Note: Potential indicators and possible precursors of insider threat can include behaviors such as inordinate, long-term job dissatisfaction, attempts to gain access to information not required for job performance, unexplained access to financial resources, bullying or sexual harassment of fellow employees, workplace violence, and other serious violations of organizational policies, procedures, directives, rules, or practices. Security awareness training includes how to communicate employee and management concerns regarding potential indicators of insider threat through appropriate organizational channels in accordance with established organizational policies and procedures.

Related Controls: *PL-4*, *PM-12*, *PS-3*, *PS-6*

Control Family: AWARENESS AND TRAINING

AT-3 : ROLE-BASED SECURITY TRAINING

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization provides role-based security training to personnel with assigned security roles and responsibilities:

Note: Organizations determine the appropriate content of security training based on the assigned roles and responsibilities of individuals and the specific security requirements of organizations and the information systems to which personnel have authorized access. In addition, organizations provide enterprise architects, information system developers, software developers, acquisition/procurement officials, information system managers, system/network administrators, personnel conducting configuration management and auditing activities, personnel performing independent verification and validation activities, security control assessors, and other personnel having access to system-level software, adequate security-related technical training specifically tailored for their assigned duties. Comprehensive role-based training addresses management, operational, and technical roles and responsibilities covering physical, personnel, and technical safeguards and countermeasures. Such training can include for example, policies, procedures, tools, and artifacts for the organizational security roles defined. Organizations also provide the training necessary for individuals to carry out their responsibilities related to operations and supply chain security within the context of organizational information security programs. Role-based security training also applies to contractors providing services to federal agencies.

Related Controls: *AT-2*, *AT-4*, *PL-4*, *PS-7*, *SA-3*, *SA-12*, *SA-16*

AT-3a.

Before authorizing access to the information system or performing assigned duties;

AT-3b.

When required by information system changes; and

AT-3c.

[Assignment: organization-defined frequency] thereafter.

AT-3 (1) : ENVIRONMENTAL CONTROLS

The organization provides [Assignment: organization-defined personnel or roles] with initial and [Assignment: organization-defined frequency] training in the employment and operation of environmental controls.

Note: Environmental controls include, for example, fire suppression and detection devices/systems, sprinkler systems, handheld fire extinguishers, fixed fire hoses, smoke detectors, temperature/humidity, HVAC, and power within the facility. Organizations identify personnel with specific roles and responsibilities associated with environmental controls requiring specialized training.

Related Controls: *PE-1*, *PE-13*, *PE-14*, *PE-15*

AT-3 (2) : PHYSICAL SECURITY CONTROLS

The organization provides [Assignment: organization-defined personnel or roles] with initial and [Assignment: organization-defined frequency] training in the employment and operation of physical security controls.

Note: Physical security controls include, for example, physical access control devices, physical intrusion alarms, monitoring/surveillance equipment, and security guards (deployment and operating procedures). Organizations identify personnel with specific roles and responsibilities associated with physical security controls requiring specialized training.

Related Controls: *PE-2*, *PE-3*, *PE-4*, *PE-5*

AT-3 (3) : PRACTICAL EXERCISES

> The organization includes practical exercises in security training that reinforce training objectives.

Note: Practical exercises may include, for example, security training for software developers that includes simulated cyber attacks exploiting common software vulnerabilities (e.g., buffer overflows), or spear/whale phishing attacks targeted at senior leaders/executives. These types of practical exercises help developers better understand the effects of such vulnerabilities and appreciate the need for security coding standards and processes.

AT-3 (4) : SUSPICIOUS COMMUNICATIONS AND ANOMALOUS SYSTEM BEHAVIOR

> The organization provides training to its personnel on [Assignment: organization-defined indicators of malicious code] to recognize suspicious communications and anomalous behavior in organizational information systems.

Note: A well-trained workforce provides another organizational safeguard that can be employed as part of a defense-in-depth strategy to protect organizations against malicious code coming in to organizations via email or the web applications. Personnel are trained to look for indications of potentially suspicious email (e.g., receiving an unexpected email, receiving an email containing strange or poor grammar, or receiving an email from an unfamiliar sender but who appears to be from a known sponsor or contractor). Personnel are also trained on how to respond to such suspicious email or web communications (e.g., not opening attachments, not clicking on embedded web links, and checking the source of email addresses). For this process to work effectively, all organizational personnel are trained and made aware of what constitutes suspicious communications. Training personnel on how to recognize anomalous behaviors in organizational information systems can potentially provide early warning for the presence of malicious code. Recognition of such anomalous behavior by organizational personnel can supplement automated malicious code detection and protection tools and systems employed by organizations.

Control Family: AWARENESS AND TRAINING

AT-4 : SECURITY TRAINING RECORDS

Priority: P3

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The organization:

Note: Documentation for specialized training may be maintained by individual supervisors at the option of the organization.

Related Controls: *AT-2*, *AT-3*, *PM-14*

AT-4a.

Documents and monitors individual information system security training activities including basic security awareness training and specific information system security training; and

AT-4b.

Retains individual training records for [Assignment: organization-defined time period].

Control Family: AWARENESS AND TRAINING

AT-5 : CONTACTS WITH SECURITY GROUPS AND ASSOCIATIONS

[Withdrawn: Incorporated into PM-15].

Control Family: AUDIT AND ACCOUNTABILITY

AU-1 : AUDIT AND ACCOUNTABILITY POLICY AND PROCEDURES

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: This control addresses the establishment of policy and procedures for the effective implementation of selected security controls and control enhancements in the AU family. Policy and procedures reflect applicable federal laws, Executive Orders, directives, regulations, policies, standards, and guidance. Security program policies and procedures at the organization level may make the need for system-specific policies and procedures unnecessary. The policy can be included as part of the general information security policy for organizations or conversely, can be represented by multiple policies reflecting the complex nature of certain organizations. The procedures can be established for the security program in general and for particular information systems, if needed. The organizational risk management strategy is a key factor in establishing policy and procedures.

Related Controls: *PM-9*

AU-1a.

Develops, documents, and disseminates to [Assignment: organization-defined personnel or roles]:

AU-1a.1.

An audit and accountability policy that addresses purpose, scope, roles, responsibilities, management commitment, coordination among organizational entities, and compliance; and

AU-1a.2.

Procedures to facilitate the implementation of the audit and accountability policy and associated audit and accountability controls; and

AU-1b.

Reviews and updates the current:

AU-1b.1.

Audit and accountability policy [Assignment: organization-defined frequency]; and

AU-1b.2.

Audit and accountability procedures [Assignment: organization-defined frequency].

Control Family: AUDIT AND ACCOUNTABILITY

AU-2 : AUDIT EVENTS

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: An event is any observable occurrence in an organizational information system. Organizations identify audit events as those events which are significant and relevant to the security of information systems and the environments in which those systems operate in order to meet specific and ongoing audit needs. Audit events can include, for example, password changes, failed logons, or failed accesses related to information systems, administrative privilege usage, PIV credential usage, or third-party credential usage. In determining the set of auditable events, organizations consider the auditing appropriate for each of the security controls to be implemented. To balance auditing requirements with other information system needs, this control also requires identifying that subset of auditable events that are audited at a given point in time. For example, organizations may determine that information systems must have the capability to log every file access both successful and unsuccessful, but not activate that capability except for specific circumstances due to the potential burden on system performance. Auditing requirements, including the need for auditable events, may be referenced in other security controls and control enhancements. Organizations also include auditable events that are required by applicable federal laws, Executive Orders, directives, policies, regulations, and standards. Audit records can be generated at various levels of abstraction, including at the packet level as information traverses the network. Selecting the appropriate level of abstraction is a critical aspect of an audit capability and can facilitate the identification of root causes to problems. Organizations consider in the definition of auditable events, the auditing necessary to cover related events such as the steps in distributed, transaction-based processes (e.g., processes that are distributed across multiple organizations) and actions that occur in service-oriented architectures.

Related Controls: *AC-6*, *AC-17*, *AU-3*, *AU-12*, *MA-4*, *MP-2*, *MP-4*, *SI-4*

AU-2a.

Determines that the information system is capable of auditing the following events: [Assignment: organization-defined auditable events];

AU-2b.

Coordinates the security audit function with other organizational entities requiring audit-related information to enhance mutual support and to help guide the selection of auditable events;

AU-2c.

Provides a rationale for why the auditable events are deemed to be adequate to support after-the-fact investigations of security incidents; and

AU-2d.

Determines that the following events are to be audited within the information system: [Assignment: organization-defined audited events (the subset of the auditable events defined in AU-2 a.) along with the frequency of (or situation requiring) auditing for each identified event].

AU-2 (1) : COMPILATION OF AUDIT RECORDS FROM MULTIPLE SOURCES

[Withdrawn: Incorporated into AU-12].

AU-2 (2) : SELECTION OF AUDIT EVENTS BY COMPONENT

[Withdrawn: Incorporated into AU-12].

AU-2 (3) : REVIEWS AND UPDATES

Baseline-Impact: *MODERATE*, **HIGH**

The organization reviews and updates the audited events [Assignment: organization-defined frequency].

Note: Over time, the events that organizations believe should be audited may change. Reviewing and updating the set of audited events periodically is necessary to ensure that the current set is still necessary and sufficient.

AU-2 (4) : PRIVILEGED FUNCTIONS

[Withdrawn: Incorporated into AC-6 (9)].

Control Family: AUDIT AND ACCOUNTABILITY

AU-3 : CONTENT OF AUDIT RECORDS

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The information system generates audit records containing information that establishes what type of event occurred, when the event occurred, where the event occurred, the source of the event, the outcome of the event, and the identity of any individuals or subjects associated with the event.

Note: Audit record content that may be necessary to satisfy the requirement of this control, includes, for example, time stamps, source and destination addresses, user/process identifiers, event descriptions, success/fail indications, filenames involved, and access control or flow control rules invoked. Event outcomes can include indicators of event success or failure and event-specific results (e.g., the security state of the information system after the event occurred).

Related Controls: *AU-2*, *AU-8*, *AU-12*, *SI-11*

AU-3 (1) : ADDITIONAL AUDIT INFORMATION

Baseline-Impact: *MODERATE*, **HIGH**

> The information system generates audit records containing the following additional information: [Assignment: organization-defined additional, more detailed information].

Note: Detailed information that organizations may consider in audit records includes, for example, full text recording of privileged commands or the individual identities of group account users. Organizations consider limiting the additional audit information to only that information explicitly needed for specific audit requirements. This facilitates the use of audit trails and audit logs by not including information that could potentially be misleading or could make it more difficult to locate information of interest.

AU-3 (2) : CENTRALIZED MANAGEMENT OF PLANNED AUDIT RECORD CONTENT

Baseline-Impact: HIGH

> The information system provides centralized management and configuration of the content to be captured in audit records generated by [Assignment: organization-defined information system components].

Note: This control enhancement requires that the content to be captured in audit records be configured from a central location (necessitating automation). Organizations coordinate the selection of required audit content to support the centralized management and configuration capability provided by the information system.

Related Controls: *AU-6*, *AU-7*

Control Family: AUDIT AND ACCOUNTABILITY

AU-4 : AUDIT STORAGE CAPACITY

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The organization allocates audit record storage capacity in accordance with [Assignment: organization-defined audit record storage requirements].

Note: Organizations consider the types of auditing to be performed and the audit processing requirements when allocating audit storage capacity. Allocating sufficient audit storage capacity reduces the likelihood of such capacity being exceeded and resulting in the potential loss or reduction of auditing capability.

Related Controls: *AU-2*, *AU-5*, *AU-6*, *AU-7*, *AU-11*, *SI-4*

AU-4 (1) : TRANSFER TO ALTERNATE STORAGE

> The information system off-loads audit records [Assignment: organization-defined frequency] onto a different system or media than the system being audited.

Note: Off-loading is a process designed to preserve the confidentiality and integrity of audit records by moving the records from the primary information system to a secondary or alternate system. It is a common process in information systems with limited audit storage capacity; the audit storage is used only in a transitory fashion until the system can communicate with the secondary or alternate system designated for storing the audit records, at which point the information is transferred.

Control Family: AUDIT AND ACCOUNTABILITY

AU-5 : RESPONSE TO AUDIT PROCESSING FAILURES

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The information system:

Note: Audit processing failures include, for example, software/hardware errors, failures in the audit capturing mechanisms, and audit storage capacity being reached or exceeded. Organizations may choose to define additional actions for different audit processing failures (e.g., by type, by location, by severity, or a combination of such factors). This control applies to each audit data storage repository (i.e., distinct information system component where audit records are stored), the total audit storage capacity of organizations (i.e., all audit data storage repositories combined), or both.

Related Controls: *AU-4*, *SI-12*

AU-5a.

Alerts [Assignment: organization-defined personnel or roles] in the event of an audit processing failure; and

AU-5b.

Takes the following additional actions: [Assignment: organization-defined actions to be taken (e.g., shut down information system, overwrite oldest audit records, stop generating audit records)].

AU-5 (1) : AUDIT STORAGE CAPACITY

Baseline-Impact: HIGH

The information system provides a warning to [Assignment: organization-defined personnel, roles, and/or locations] within [Assignment: organization-defined time period] when allocated audit record storage volume reaches [Assignment: organization-defined percentage] of repository maximum audit record storage capacity.

Note: Organizations may have multiple audit data storage repositories distributed across multiple information system components, with each repository having different storage volume capacities.

AU-5 (2) : REAL-TIME ALERTS

Baseline-Impact: HIGH

The information system provides an alert in [Assignment: organization-defined real-time period] to [Assignment: organization-defined personnel, roles, and/or locations] when the following audit failure events occur: [Assignment: organization-defined audit failure events requiring real-time alerts].

Note: Alerts provide organizations with urgent messages. Real-time alerts provide these messages at information technology speed (i.e., the time from event detection to alert occurs in seconds or less).

AU-5 (3) : CONFIGURABLE TRAFFIC VOLUME THRESHOLDS

The information system enforces configurable network communications traffic volume thresholds reflecting limits on auditing capacity and [Selection: rejects; delays] network traffic above those thresholds.

Note: Organizations have the capability to reject or delay the processing of network communications traffic if auditing such traffic is determined to exceed the storage capacity of the information system audit function. The rejection or delay response is triggered by the established organizational traffic volume thresholds which can be adjusted based on changes to audit storage capacity.

AU-5 (4) : SHUTDOWN ON FAILURE

The information system invokes a [Selection: full system shutdown; partial system shutdown; degraded operational mode with limited mission/business functionality available] in the event of [Assignment: organization-defined audit failures], unless an alternate audit capability exists.

Note: Organizations determine the types of audit failures that can trigger automatic information system shutdowns or degraded operations. Because of the importance of ensuring mission/business continuity, organizations may determine that the nature of the audit failure is not so severe that it warrants a complete shutdown of the information system supporting the core organizational missions/business operations. In those instances, partial information system shutdowns or operating in a degraded mode with reduced capability may be viable alternatives.

Related Controls: *AU-15*

Control Family: AUDIT AND ACCOUNTABILITY

AU-6 : AUDIT REVIEW, ANALYSIS, AND REPORTING

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: Audit review, analysis, and reporting covers information security-related auditing performed by organizations including, for example, auditing that results from monitoring of account usage, remote access, wireless connectivity, mobile device connection, configuration settings, system component inventory, use of maintenance tools and nonlocal maintenance, physical access, temperature and humidity, equipment delivery and removal, communications at the information system boundaries, use of mobile code, and use of VoIP. Findings can be reported to organizational entities that include, for example, incident response team, help desk, information security group/department. If organizations are prohibited from reviewing and analyzing audit information or unable to conduct such activities (e.g., in certain national security applications or systems), the review/analysis may be carried out by other organizations granted such authority.

Related Controls: *AC-2*, *AC-3*, *AC-6*, *AC-17*, *AT-3*, *AU-7*, *AU-16*, *CA-7*, *CM-5*, *CM-10*, *CM-11*, *IA-3*, *IA-5*, *IR-5*, *IR-6*, *MA-4*, *MP-4*, *PE-3*, *PE-6*, *PE-14*, *PE-16*, *RA-5*, *SC-7*, *SC-18*, *SC-19*, *SI-3*, *SI-4*, *SI-7*

AU-6a.

Reviews and analyzes information system audit records [Assignment: organization-defined frequency] for indications of [Assignment: organization-defined inappropriate or unusual activity]; and

AU-6b.

Reports findings to [Assignment: organization-defined personnel or roles].

AU-6 (1) : PROCESS INTEGRATION

Baseline-Impact: *MODERATE*, **HIGH**

The organization employs automated mechanisms to integrate audit review, analysis, and reporting processes to support organizational processes for investigation and response to suspicious activities.

Note: Organizational processes benefiting from integrated audit review, analysis, and reporting include, for example, incident response, continuous monitoring, contingency planning, and Inspector General audits.

Related Controls: *AU-12*, *PM-7*

AU-6 (2) : AUTOMATED SECURITY ALERTS

[Withdrawn: Incorporated into SI-4].

AU-6 (3) : CORRELATE AUDIT REPOSITORIES

Baseline-Impact: *MODERATE*, **HIGH**

The organization analyzes and correlates audit records across different repositories to gain organization-wide situational awareness.

Note: Organization-wide situational awareness includes awareness across all three tiers of risk management (i.e., organizational, mission/business process, and information system) and supports cross-organization awareness.

Related Controls: *AU-12*, *IR-4*

AU-6 (4) : CENTRAL REVIEW AND ANALYSIS

The information system provides the capability to centrally review and analyze audit records from multiple components within the system.

Note: Automated mechanisms for centralized reviews and analyses include, for example, Security Information Management products.

Related Controls: *AU-2*, *AU-12*

AU-6 (5) : INTEGRATION / SCANNING AND MONITORING CAPABILITIES

Baseline-Impact: HIGH

> The organization integrates analysis of audit records with analysis of [Selection (one or more): vulnerability scanning information; performance data; information system monitoring information; [Assignment: organization-defined data/information collected from other sources]] to further enhance the ability to identify inappropriate or unusual activity.

Note: This control enhancement does not require vulnerability scanning, the generation of performance data, or information system monitoring. Rather, the enhancement requires that the analysis of information being otherwise produced in these areas is integrated with the analysis of audit information. Security Event and Information Management System tools can facilitate audit record aggregation/consolidation from multiple information system components as well as audit record correlation and analysis. The use of standardized audit record analysis scripts developed by organizations (with localized script adjustments, as necessary) provides more cost-effective approaches for analyzing audit record information collected. The correlation of audit record information with vulnerability scanning information is important in determining the veracity of vulnerability scans and correlating attack detection events with scanning results. Correlation with performance data can help uncover denial of service attacks or cyber attacks resulting in unauthorized use of resources. Correlation with system monitoring information can assist in uncovering attacks and in better relating audit information to operational situations.

Related Controls: *AU-12*, *IR-4*, *RA-5*

AU-6 (6) : CORRELATION WITH PHYSICAL MONITORING

Baseline-Impact: HIGH

> The organization correlates information from audit records with information obtained from monitoring physical access to further enhance the ability to identify suspicious, inappropriate, unusual, or malevolent activity.

Note: The correlation of physical audit information and audit logs from information systems may assist organizations in identifying examples of suspicious behavior or supporting evidence of such behavior. For example, the correlation of an individuals identity for logical access to certain information systems with the additional physical security information that the individual was actually present at the facility when the logical access occurred, may prove to be useful in investigations.

AU-6 (7) : PERMITTED ACTIONS

> The organization specifies the permitted actions for each [Selection (one or more): information system process; role; user] associated with the review, analysis, and reporting of audit information.

Note: Organizations specify permitted actions for information system processes, roles, and/or users associated with the review, analysis, and reporting of audit records through account management techniques. Specifying permitted actions on audit information is a way to enforce the principle of least privilege. Permitted actions are enforced by the information system and include, for example, read, write, execute, append, and delete.

AU-6 (8) : FULL TEXT ANALYSIS OF PRIVILEGED COMMANDS

> The organization performs a full text analysis of audited privileged commands in a physically distinct component or subsystem of the information system, or other information system that is dedicated to that analysis.

Note: This control enhancement requires a distinct environment for the dedicated analysis of audit information related to privileged users without compromising such information on the information system where the users have elevated privileges including the capability to execute privileged commands. Full text analysis refers to analysis that considers the full text of privileged commands (i.e., commands and all parameters) as opposed to analysis that considers only the name of the command. Full text analysis includes, for example, the use of pattern matching and heuristics.

Related Controls: *AU-3*, *AU-9*, *AU-11*, *AU-12*

AU-6 (9) : CORRELATION WITH INFORMATION FROM NONTECHNICAL SOURCES

> The organization correlates information from nontechnical sources with audit information to enhance organization-wide situational awareness.

Note: Nontechnical sources include, for example, human resources records documenting organizational policy violations (e.g., sexual harassment incidents, improper use of organizational information assets). Such information can lead organizations to a more directed analytical effort to detect potential malicious insider activity. Due to the sensitive nature of the information available from nontechnical sources, organizations limit access to such information to minimize the potential for the inadvertent release of privacy-related information to individuals that do not have a need to know. Thus, correlation of information from nontechnical sources with audit information generally occurs only when individuals are suspected of being involved in a security incident. Organizations obtain legal advice prior to initiating such actions.

Related Controls: *AT-2*

AU-6 (10) : AUDIT LEVEL ADJUSTMENT

> The organization adjusts the level of audit review, analysis, and reporting within the information system when there is a change in risk based on law enforcement information, intelligence information, or other credible sources of information.

Note: The frequency, scope, and/or depth of the audit review, analysis, and reporting may be adjusted to meet organizational needs based on new information received.

Control Family: AUDIT AND ACCOUNTABILITY

AU-7 : AUDIT REDUCTION AND REPORT GENERATION

Priority: P2

Baseline-Impact: *MODERATE*, **HIGH**

The information system provides an audit reduction and report generation capability that:

Note: Audit reduction is a process that manipulates collected audit information and organizes such information in a summary format that is more meaningful to analysts. Audit reduction and report generation capabilities do not always emanate from the same information system or from the same organizational entities conducting auditing activities. Audit reduction capability can include, for example, modern data mining techniques with advanced data filters to identify anomalous behavior in audit records. The report generation capability provided by the information system can generate customizable reports. Time ordering of audit records can be a significant issue if the granularity of the timestamp in the record is insufficient.

Related Controls: *AU-6*

AU-7a.

Supports on-demand audit review, analysis, and reporting requirements and after-the-fact investigations of security incidents; and

AU-7b.

Does not alter the original content or time ordering of audit records.

AU-7 (1) : AUTOMATIC PROCESSING

Baseline-Impact: *MODERATE*, **HIGH**

The information system provides the capability to process audit records for events of interest based on [Assignment: organization-defined audit fields within audit records].

Note: Events of interest can be identified by the content of specific audit record fields including, for example, identities of individuals, event types, event locations, event times, event dates, system resources involved, IP addresses involved, or information objects accessed. Organizations may define audit event criteria to any degree of granularity required, for example, locations selectable by general networking location (e.g., by network or subnetwork) or selectable by specific information system component.

Related Controls: *AU-2*, *AU-12*

AU-7 (2) : AUTOMATIC SORT AND SEARCH

The information system provides the capability to sort and search audit records for events of interest based on the content of [Assignment: organization-defined audit fields within audit records].

Note: Sorting and searching of audit records may be based upon the contents of audit record fields, for example: (i) date/time of events; (ii) user identifiers; (iii) Internet Protocol (IP) addresses involved in the event; (iv) type of event; or (v) event success/failure.

Control Family: AUDIT AND ACCOUNTABILITY

AU-8 : TIME STAMPS

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The information system:

Note: Time stamps generated by the information system include date and time. Time is commonly expressed in Coordinated Universal Time (UTC), a modern continuation of Greenwich Mean Time (GMT), or local time with an offset from UTC. Granularity of time measurements refers to the degree of synchronization between information system clocks and reference clocks, for example, clocks synchronizing within hundreds of milliseconds or within tens of milliseconds. Organizations may define different time granularities for different system components. Time service can also be critical to other security capabilities such as access control and identification and authentication, depending on the nature of the mechanisms used to support those capabilities.

Related Controls: *AU-3*, *AU-12*

AU-8a.

Uses internal system clocks to generate time stamps for audit records; and

AU-8b.

Records time stamps for audit records that can be mapped to Coordinated Universal Time (UTC) or Greenwich Mean Time (GMT) and meets [Assignment: organization-defined granularity of time measurement].

AU-8 (1) : SYNCHRONIZATION WITH AUTHORITATIVE TIME SOURCE

Baseline-Impact: *MODERATE*, **HIGH**

The information system:

Note: This control enhancement provides uniformity of time stamps for information systems with multiple system clocks and systems connected over a network.

AU-8 (1)(a)

Compares the internal information system clocks [Assignment: organization-defined frequency] with [Assignment: organization-defined authoritative time source]; and

AU-8 (1)(b)

Synchronizes the internal system clocks to the authoritative time source when the time difference is greater than [Assignment: organization-defined time period].

AU-8 (2) : SECONDARY AUTHORITATIVE TIME SOURCE

The information system identifies a secondary authoritative time source that is located in a different geographic region than the primary authoritative time source.

Control Family: AUDIT AND ACCOUNTABILITY

AU-9 : PROTECTION OF AUDIT INFORMATION

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The information system protects audit information and audit tools from unauthorized access, modification, and deletion.

Note: Audit information includes all information (e.g., audit records, audit settings, and audit reports) needed to successfully audit information system activity. This control focuses on technical protection of audit information. Physical protection of audit information is addressed by media protection controls and physical and environmental protection controls.

Related Controls: *AC-3*, *AC-6*, *MP-2*, *MP-4*, *PE-2*, *PE-3*, *PE-6*

AU-9 (1) : HARDWARE WRITE-ONCE MEDIA

The information system writes audit trails to hardware-enforced, write-once media.

Note: This control enhancement applies to the initial generation of audit trails (i.e., the collection of audit records that represents the audit information to be used for detection, analysis, and reporting purposes) and to the backup of those audit trails. The enhancement does not apply to the initial generation of audit records prior to being written to an audit trail. Write-once, read-many (WORM) media includes, for example, Compact Disk-Recordable (CD-R) and Digital Video Disk-Recordable (DVD-R). In contrast, the use of switchable write-protection media such as on tape cartridges or Universal Serial Bus (USB) drives results in write-protected, but not write-once, media.

Related Controls: *AU-4*, *AU-5*

AU-9 (2) : AUDIT BACKUP ON SEPARATE PHYSICAL SYSTEMS / COMPONENTS

Baseline-Impact: HIGH

> The information system backs up audit records [Assignment: organization-defined frequency] onto a physically different system or system component than the system or component being audited.

Note: This control enhancement helps to ensure that a compromise of the information system being audited does not also result in a compromise of the audit records.

Related Controls: *AU-4*, *AU-5*, *AU-11*

AU-9 (3) : CRYPTOGRAPHIC PROTECTION

Baseline-Impact: HIGH

> The information system implements cryptographic mechanisms to protect the integrity of audit information and audit tools.

Note: Cryptographic mechanisms used for protecting the integrity of audit information include, for example, signed hash functions using asymmetric cryptography enabling distribution of the public key to verify the hash information while maintaining the confidentiality of the secret key used to generate the hash.

Related Controls: *AU-10*, *SC-12*, *SC-13*

AU-9 (4) : ACCESS BY SUBSET OF PRIVILEGED USERS

Baseline-Impact: *MODERATE*, **HIGH**

> The organization authorizes access to management of audit functionality to only [Assignment: organization-defined subset of privileged users].

Note: Individuals with privileged access to an information system and who are also the subject of an audit by that system, may affect the reliability of audit information by inhibiting audit activities or modifying audit records. This control enhancement requires that privileged access be further defined between audit-related privileges and other privileges, thus limiting the users with audit-related privileges.

Related Controls: *AC-5*

AU-9 (5) : DUAL AUTHORIZATION

> The organization enforces dual authorization for [Selection (one or more): movement; deletion] of [Assignment: organization-defined audit information].

Note: Organizations may choose different selection options for different types of audit information. Dual authorization mechanisms require the approval of two authorized individuals in order to execute. Dual authorization may also be known as two-person control.

Related Controls: *AC-3*, *MP-2*

AU-9 (6) : READ ONLY ACCESS

> The organization authorizes read-only access to audit information to [Assignment: organization-defined subset of privileged users].

Note: Restricting privileged user authorizations to read-only helps to limit the potential damage to organizations that could be initiated by such users (e.g., deleting audit records to cover up malicious activity).

Control Family: AUDIT AND ACCOUNTABILITY

AU-10 : NON-REPUDIATION

Priority: P2

Baseline-Impact: HIGH

> The information system protects against an individual (or process acting on behalf of an individual) falsely denying having performed [Assignment: organization-defined actions to be covered by non-repudiation].

Note: Types of individual actions covered by non-repudiation include, for example, creating information, sending and receiving messages, approving information (e.g., indicating concurrence or signing a contract). Non-repudiation protects individuals against later claims by: (i) authors of not having authored particular documents; (ii) senders of not having transmitted messages; (iii) receivers of not having received messages; or (iv) signatories of not having signed documents. Non-repudiation services can be used to determine if information originated from a particular individual, or if an individual took specific actions (e.g., sending an email, signing a contract, approving a procurement request) or received specific information. Organizations obtain non-repudiation services by employing various techniques or mechanisms (e.g., digital signatures, digital message receipts).

Related Controls: *SC-12*, *SC-8*, *SC-13*, *SC-16*, *SC-17*, *SC-23*

AU-10 (1) : ASSOCIATION OF IDENTITIES

> The information system:

Note: This control enhancement supports audit requirements that provide organizational personnel with the means to identify who produced specific information in the event of an information transfer. Organizations determine and approve the strength of the binding between the information producer and the information based on the security category of the information and relevant risk factors.

Related Controls: *AC-4*, *AC-16*

AU-10 (1)(a)

Binds the identity of the information producer with the information to [Assignment: organization-defined strength of binding]; and

AU-10 (1)(b)

Provides the means for authorized individuals to determine the identity of the producer of the information.

AU-10 (2) : VALIDATE BINDING OF INFORMATION PRODUCER IDENTITY

The information system:

Note: This control enhancement prevents the modification of information between production and review. The validation of bindings can be achieved, for example, by the use of cryptographic checksums. Organizations determine if validations are in response to user requests or generated automatically.

Related Controls: *AC-3*, *AC-4*, *AC-16*

AU-10 (2)(a)

Validates the binding of the information producer identity to the information at [Assignment: organization-defined frequency]; and

AU-10 (2)(b)

Performs [Assignment: organization-defined actions] in the event of a validation error.

AU-10 (3) : CHAIN OF CUSTODY

The information system maintains reviewer/releaser identity and credentials within the established chain of custody for all information reviewed or released.

Note: Chain of custody is a process that tracks the movement of evidence through its collection, safeguarding, and analysis life cycle by documenting each person who handled the evidence, the date and time it was collected or transferred, and the purpose for the transfer. If the reviewer is a human or if the review function is automated but separate from the release/transfer function, the information system associates the identity of the reviewer of the information to be released with the information and the information label. In the case of human reviews, this control enhancement provides organizational officials the means to identify who reviewed and released the information. In the case of automated reviews, this control enhancement ensures that only approved review functions are employed.

Related Controls: *AC-4*, *AC-16*

AU-10 (4) : VALIDATE BINDING OF INFORMATION REVIEWER IDENTITY

The information system:

Note: This control enhancement prevents the modification of information between review and transfer/release. The validation of bindings can be achieved, for example, by the use of cryptographic checksums. Organizations determine validations are in response to user requests or generated automatically.

Related Controls: *AC-4*, *AC-16*

AU-10 (4)(a)

Validates the binding of the information reviewer identity to the information at the transfer or release points prior to release/transfer between [Assignment: organization-defined security domains]; and

AU-10 (4)(b)

Performs [Assignment: organization-defined actions] in the event of a validation error.

AU-10 (5) : DIGITAL SIGNATURES

[Withdrawn: Incorporated into SI-7].

Control Family: AUDIT AND ACCOUNTABILITY

AU-11 : AUDIT RECORD RETENTION

Priority: P3

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization retains audit records for [Assignment: organization-defined time period consistent with records retention policy] to provide support for after-the-fact investigations of security incidents and to meet regulatory and organizational information retention requirements.

Note: Organizations retain audit records until it is determined that they are no longer needed for administrative, legal, audit, or other operational purposes. This includes, for example, retention and availability of audit records relative to Freedom of Information Act (FOIA) requests, subpoenas, and law enforcement actions. Organizations develop standard categories of audit records relative to such types of actions and standard response processes for each type of action. The National Archives and Records Administration (NARA) General Records Schedules provide federal policy on record retention.

Related Controls: *AU-4*, *AU-5*, *AU-9*, *MP-6*

AU-11 (1) : LONG-TERM RETRIEVAL CAPABILITY

> The organization employs [Assignment: organization-defined measures] to ensure that long-term audit records generated by the information system can be retrieved.

Note: Measures employed by organizations to help facilitate the retrieval of audit records include, for example, converting records to newer formats, retaining equipment capable of reading the records, and retaining necessary documentation to help organizational personnel understand how to interpret the records.

Control Family: AUDIT AND ACCOUNTABILITY

AU-12 : AUDIT GENERATION

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The information system:

Note: Audit records can be generated from many different information system components. The list of audited events is the set of events for which audits are to be generated. These events are typically a subset of all events for which the information system is capable of generating audit records.

Related Controls: *AC-3*, *AU-2*, *AU-3*, *AU-6*, *AU-7*

AU-12a.

> Provides audit record generation capability for the auditable events defined in AU-2 a. at [Assignment: organization-defined information system components];

AU-12b.

> Allows [Assignment: organization-defined personnel or roles] to select which auditable events are to be audited by specific components of the information system; and

AU-12c.

> Generates audit records for the events defined in AU-2 d. with the content defined in AU-3.

AU-12 (1) : SYSTEM-WIDE / TIME-CORRELATED AUDIT TRAIL

Baseline-Impact: HIGH

> The information system compiles audit records from [Assignment: organization-defined information system components] into a system-wide (logical or physical) audit trail that is time-correlated to within [Assignment: organization-defined level of tolerance for the relationship between time stamps of individual records in the audit trail].

Note: Audit trails are time-correlated if the time stamps in the individual audit records can be reliably related to the time stamps in other audit records to achieve a time ordering of the records within organizational tolerances.

Related Controls: *AU-8*, *AU-12*

AU-12 (2) : STANDARDIZED FORMATS

> The information system produces a system-wide (logical or physical) audit trail composed of audit records in a standardized format.

Note: Audit information that is normalized to common standards promotes interoperability and exchange of such information between dissimilar devices and information systems. This facilitates production of event information that can be more readily analyzed and correlated. Standard formats for audit records include, for example, system log records and audit records compliant with Common Event Expressions (CEE). If logging mechanisms within information systems do not conform to standardized formats, systems may convert individual audit records into standardized formats when compiling system-wide audit trails.

AU-12 (3) : CHANGES BY AUTHORIZED INDIVIDUALS

Baseline-Impact: HIGH

> The information system provides the capability for [Assignment: organization-defined individuals or roles] to change the auditing to be performed on [Assignment: organization-defined information system components] based on [Assignment: organization-defined selectable event criteria] within [Assignment: organization-defined time thresholds].

Note: This control enhancement enables organizations to extend or limit auditing as necessary to meet organizational requirements. Auditing that is limited to conserve information system resources may be extended to address certain threat situations. In addition, auditing may be limited to a specific set of events to facilitate audit reduction, analysis, and reporting. Organizations can establish time thresholds in which audit actions are changed, for example, near real-time, within minutes, or within hours.

Related Controls: *AU-7*

Control Family: AUDIT AND ACCOUNTABILITY

AU-13 : MONITORING FOR INFORMATION DISCLOSURE

Priority: P0

> The organization monitors [Assignment: organization-defined open source information and/or information sites] [Assignment: organization-defined frequency] for evidence of unauthorized disclosure of organizational information.

Note: Open source information includes, for example, social networking sites.

Related Controls: *PE-3*, *SC-7*

AU-13 (1) : USE OF AUTOMATED TOOLS

> The organization employs automated mechanisms to determine if organizational information has been disclosed in an unauthorized manner.

Note: Automated mechanisms can include, for example, automated scripts to monitor new posts on selected websites, and commercial services providing notifications and alerts to organizations.

AU-13 (2) : REVIEW OF MONITORED SITES

> The organization reviews the open source information sites being monitored [Assignment: organization-defined frequency].

Control Family: AUDIT AND ACCOUNTABILITY

AU-14 : SESSION AUDIT

Priority: P0

> The information system provides the capability for authorized users to select a user session to capture/record or view/hear.

Note: Session audits include, for example, monitoring keystrokes, tracking websites visited, and recording information and/or file transfers. Session auditing activities are developed, integrated, and used in consultation with legal counsel in accordance with applicable federal laws, Executive Orders, directives, policies, regulations, or standards.

Related Controls: *AC-3*, *AU-4*, *AU-5*, *AU-9*, *AU-11*

AU-14 (1) : SYSTEM START-UP

The information system initiates session audits at system start-up.

AU-14 (2) : CAPTURE/RECORD AND LOG CONTENT

The information system provides the capability for authorized users to capture/record and log content related to a user session.

AU-14 (3) : REMOTE VIEWING / LISTENING

The information system provides the capability for authorized users to remotely view/hear all content related to an established user session in real time.

Control Family: AUDIT AND ACCOUNTABILITY

AU-15 : ALTERNATE AUDIT CAPABILITY

Priority: P0

The organization provides an alternate audit capability in the event of a failure in primary audit capability that provides [Assignment: organization-defined alternate audit functionality].

Note: Since an alternate audit capability may be a short-term protection employed until the failure in the primary auditing capability is corrected, organizations may determine that the alternate audit capability need only provide a subset of the primary audit functionality that is impacted by the failure.

Related Controls: *AU-5*

Control Family: AUDIT AND ACCOUNTABILITY

AU-16 : CROSS-ORGANIZATIONAL AUDITING

Priority: P0

The organization employs [Assignment: organization-defined methods] for coordinating [Assignment: organization-defined audit information] among external organizations when audit information is transmitted across organizational boundaries.

Note: When organizations use information systems and/or services of external organizations, the auditing capability necessitates a coordinated approach across organizations. For example, maintaining the identity of individuals that requested particular services across organizational boundaries may often be very difficult, and doing so may prove to have significant performance ramifications. Therefore, it is often the case that cross-organizational auditing (e.g., the type of auditing capability provided by service-oriented architectures) simply captures the identity of individuals issuing requests at the initial information system, and subsequent systems record that the requests emanated from authorized individuals.

Related Controls: *AU-6*

AU-16 (1) : IDENTITY PRESERVATION

> The organization requires that the identity of individuals be preserved in cross-organizational audit trails.

Note: This control enhancement applies when there is a need to be able to trace actions that are performed across organizational boundaries to a specific individual.

AU-16 (2) : SHARING OF AUDIT INFORMATION

> The organization provides cross-organizational audit information to [Assignment: organization-defined organizations] based on [Assignment: organization-defined cross-organizational sharing agreements].

Note: Because of the distributed nature of the audit information, cross-organization sharing of audit information may be essential for effective analysis of the auditing being performed. For example, the audit records of one organization may not provide sufficient information to determine the appropriate or inappropriate use of organizational information resources by individuals in other organizations. In some instances, only the home organizations of individuals have the appropriate knowledge to make such determinations, thus requiring the sharing of audit information among organizations.

Control Family: SECURITY ASSESSMENT AND AUTHORIZATION

CA-1 : SECURITY ASSESSMENT AND AUTHORIZATION POLICY AND PROCEDURES

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The organization:

Note: This control addresses the establishment of policy and procedures for the effective implementation of selected security controls and control enhancements in the CA family. Policy and procedures reflect applicable federal laws, Executive Orders, directives, regulations, policies, standards, and guidance. Security program policies and procedures at the organization level may make the need for system-specific policies and procedures unnecessary. The policy can be included as part of the general information security policy for organizations or conversely, can be represented by multiple policies reflecting the complex nature of certain organizations. The procedures can be established for the security program in general and for particular information systems, if needed. The organizational risk management strategy is a key factor in establishing policy and procedures.

Related Controls: *PM-9*

CA-1a.

Develops, documents, and disseminates to [Assignment: organization-defined personnel or roles]:

CA-1a.1.

A security assessment and authorization policy that addresses purpose, scope, roles, responsibilities, management commitment, coordination among organizational entities, and compliance; and

CA-1a.2.

Procedures to facilitate the implementation of the security assessment and authorization policy and associated security assessment and authorization controls; and

CA-1b.

Reviews and updates the current:

CA-1b.1.

Security assessment and authorization policy [Assignment: organization-defined frequency]; and

CA-1b.2.

Security assessment and authorization procedures [Assignment: organization-defined frequency].

Control Family: SECURITY ASSESSMENT AND AUTHORIZATION

CA-2 : SECURITY ASSESSMENTS

Priority: P2

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note:

Organizations assess security controls in organizational information systems and the environments in which those systems operate as part of: (i) initial and ongoing security authorizations; (ii) FISMA annual assessments; (iii) continuous monitoring; and (iv) system development life cycle activities. Security assessments: (i) ensure that information security is built into organizational information systems; (ii) identify weaknesses and deficiencies early in the development process; (iii) provide essential information needed to make risk-based decisions as part of security authorization processes; and (iv) ensure compliance to vulnerability mitigation procedures. Assessments are conducted on the implemented security controls from Appendix F (main catalog) and Appendix G (Program Management controls) as documented in System Security Plans and Information Security Program Plans. Organizations can use other types of assessment activities such as vulnerability scanning and system monitoring to maintain the security posture of information systems during the entire life cycle. Security assessment reports document assessment results in sufficient detail as deemed necessary by organizations, to determine the accuracy and completeness of the reports and whether the security controls are implemented correctly, operating as intended, and producing the desired outcome with respect to meeting security requirements. The FISMA requirement for assessing security controls at least annually does not require additional assessment activities to those activities already in place in organizational security authorization processes. Security assessment results are provided to the individuals or roles appropriate for the types of assessments being conducted. For example, assessments conducted in support of security authorization decisions are provided to authorizing officials or authorizing official designated representatives.

To satisfy annual assessment requirements, organizations can use assessment results from the following sources: (i) initial or ongoing information system authorizations; (ii) continuous monitoring; or (iii) system development life cycle activities. Organizations ensure that security assessment results are current, relevant to the determination of security control effectiveness, and obtained with the appropriate level of assessor independence. Existing security control assessment results can be reused to the extent that the results are still valid and can also be supplemented with additional assessments as needed. Subsequent to initial authorizations and in accordance with OMB policy, organizations assess security controls during continuous monitoring. Organizations establish the frequency for ongoing security control assessments in accordance with organizational continuous monitoring strategies. Information Assurance Vulnerability Alerts provide useful examples of vulnerability mitigation procedures. External audits (e.g., audits by external entities such as regulatory agencies) are outside the scope of this control.

Related Controls: *CA-5*, *CA-6*, *CA-7*, *PM-9*, *RA-5*, *SA-11*, *SA-12*, *SI-4*

CA-2a.

Develops a security assessment plan that describes the scope of the assessment including:

CA-2a.1.

Security controls and control enhancements under assessment;

CA-2a.2.

Assessment procedures to be used to determine security control effectiveness; and

CA-2a.3.

Assessment environment, assessment team, and assessment roles and responsibilities;

CA-2b.

Assesses the security controls in the information system and its environment of operation [Assignment: organization-defined frequency] to determine the extent to which the controls are implemented correctly, operating as intended, and producing the desired outcome with respect to meeting established security requirements;

CA-2c.

Produces a security assessment report that documents the results of the assessment; and

CA-2d.

Provides the results of the security control assessment to [Assignment: organization-defined individuals or roles].

CA-2 (1) : INDEPENDENT ASSESSORS

Baseline-Impact: *MODERATE*, **HIGH**

The organization employs assessors or assessment teams with [Assignment: organization-defined level of independence] to conduct security control assessments.

Note: Independent assessors or assessment teams are individuals or groups who conduct impartial assessments of organizational information systems. Impartiality implies that assessors are free from any perceived or actual conflicts of interest with regard to the development, operation, or management of the organizational information systems under assessment or to the determination of security control effectiveness. To achieve impartiality, assessors should not: (i) create a mutual or conflicting interest with the organizations where the assessments are being conducted; (ii) assess their own work; (iii) act as management or employees of the organizations they are serving; or (iv) place themselves in positions of advocacy for the organizations acquiring their services. Independent assessments can be obtained from elements within organizations or can be contracted to public or private sector entities outside of organizations. Authorizing officials determine the required level of independence based on the security categories of information systems and/or the ultimate risk to organizational operations, organizational assets, or individuals. Authorizing officials also determine if the level of assessor independence provides sufficient assurance that the results are sound and can be used to make credible, risk-based decisions. This includes determining whether contracted security assessment services have sufficient independence, for example, when information system owners are not directly involved in contracting processes or cannot unduly influence the impartiality of assessors conducting assessments. In special situations, for example, when organizations that own the information systems are small or organizational structures require that assessments are conducted by individuals that are in the developmental, operational, or management chain of system owners, independence in assessment processes can be achieved by ensuring that assessment results are carefully reviewed and analyzed by independent teams of experts to validate the completeness, accuracy, integrity, and reliability of the results. Organizations recognize that assessments performed for purposes other than direct support to authorization decisions are, when performed by assessors with sufficient independence, more likely to be usable for such decisions, thereby reducing the need to repeat assessments.

CA-2 (2) : SPECIALIZED ASSESSMENTS

Baseline-Impact: HIGH

> The organization includes as part of security control assessments, [Assignment: organization-defined frequency], [Selection: announced; unannounced], [Selection (one or more): in-depth monitoring; vulnerability scanning; malicious user testing; insider threat assessment; performance/load testing; [Assignment: organization-defined other forms of security assessment]].

Note: Organizations can employ information system monitoring, insider threat assessments, malicious user testing, and other forms of testing (e.g., verification and validation) to improve readiness by exercising organizational capabilities and indicating current performance levels as a means of focusing actions to improve security. Organizations conduct assessment activities in accordance with applicable federal laws, Executive Orders, directives, policies, regulations, and standards. Authorizing officials approve the assessment methods in coordination with the organizational risk executive function. Organizations can incorporate vulnerabilities uncovered during assessments into vulnerability remediation processes.

Related Controls: *PE-3*, *SI-2*

CA-2 (3) : EXTERNAL ORGANIZATIONS

> The organization accepts the results of an assessment of [Assignment: organization-defined information system] performed by [Assignment: organization-defined external organization] when the assessment meets [Assignment: organization-defined requirements].

Note: Organizations may often rely on assessments of specific information systems by other (external) organizations. Utilizing such existing assessments (i.e., reusing existing assessment evidence) can significantly decrease the time and resources required for organizational assessments by limiting the amount of independent assessment activities that organizations need to perform. The factors that organizations may consider in determining whether to accept assessment results from external organizations can vary. Determinations for accepting assessment results can be based on, for example, past assessment experiences one organization has had with another organization, the reputation that organizations have with regard to assessments, the level of detail of supporting assessment documentation provided, or mandates imposed upon organizations by federal legislation, policies, or directives.

Control Family: SECURITY ASSESSMENT AND AUTHORIZATION

CA-3 : SYSTEM INTERCONNECTIONS

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The organization:

Note: This control applies to dedicated connections between information systems (i.e., system interconnections) and does not apply to transitory, user-controlled connections such as email and website browsing. Organizations carefully consider the risks that may be introduced when information systems are connected to other systems with different security requirements and security controls, both within organizations and external to organizations. Authorizing officials determine the risk associated with information system connections and the appropriate controls employed. If interconnecting systems have the same authorizing official, organizations do not need to develop Interconnection Security

Agreements. Instead, organizations can describe the interface characteristics between those interconnecting systems in their respective security plans. If interconnecting systems have different authorizing officials within the same organization, organizations can either develop Interconnection Security Agreements or describe the interface characteristics between systems in the security plans for the respective systems. Organizations may also incorporate Interconnection Security Agreement information into formal contracts, especially for interconnections established between federal agencies and nonfederal (i.e., private sector) organizations. Risk considerations also include information systems sharing the same networks. For certain technologies (e.g., space, unmanned aerial vehicles, and medical devices), there may be specialized connections in place during preoperational testing. Such connections may require Interconnection Security Agreements and be subject to additional security controls.

Related Controls: *AC-3*, *AC-4*, *AC-20*, *AU-2*, *AU-12*, *AU-16*, *CA-7*, *IA-3*, *SA-9*, *SC-7*, *SI-4*

CA-3a.

Authorizes connections from the information system to other information systems through the use of Interconnection Security Agreements;

CA-3b.

Documents, for each interconnection, the interface characteristics, security requirements, and the nature of the information communicated; and

CA-3c.

Reviews and updates Interconnection Security Agreements [Assignment: organization-defined frequency].

CA-3 (1) : UNCLASSIFIED NATIONAL SECURITY SYSTEM CONNECTIONS

The organization prohibits the direct connection of an [Assignment: organization-defined unclassified, national security system] to an external network without the use of [Assignment: organization-defined boundary protection device].

Note: Organizations typically do not have control over external networks (e.g., the Internet). Approved boundary protection devices (e.g., routers, firewalls) mediate communications (i.e., information flows) between unclassified national security systems and external networks. This control enhancement is required for organizations processing, storing, or transmitting Controlled Unclassified Information (CUI).

CA-3 (2) : CLASSIFIED NATIONAL SECURITY SYSTEM CONNECTIONS

The organization prohibits the direct connection of a classified, national security system to an external network without the use of [Assignment: organization-defined boundary protection device].

Note: Organizations typically do not have control over external networks (e.g., the Internet). Approved boundary protection devices (e.g., routers, firewalls) mediate communications (i.e., information flows) between classified national security systems and external networks. In addition, approved boundary protection devices (typically managed interface/cross-domain systems) provide information flow enforcement from information systems to external networks.

CA-3 (3) : UNCLASSIFIED NON-NATIONAL SECURITY SYSTEM CONNECTIONS

> The organization prohibits the direct connection of an [Assignment: organization-defined unclassified, non-national security system] to an external network without the use of [Assignment; organization-defined boundary protection device].

Note: Organizations typically do not have control over external networks (e.g., the Internet). Approved boundary protection devices (e.g., routers, firewalls) mediate communications (i.e., information flows) between unclassified non-national security systems and external networks. This control enhancement is required for organizations processing, storing, or transmitting Controlled Unclassified Information (CUI).

CA-3 (4) : CONNECTIONS TO PUBLIC NETWORKS

> The organization prohibits the direct connection of an [Assignment: organization-defined information system] to a public network.

Note: A public network is any network accessible to the general public including, for example, the Internet and organizational extranets with public access.

CA-3 (5) : RESTRICTIONS ON EXTERNAL SYSTEM CONNECTIONS

Baseline-Impact: *MODERATE*, **HIGH**

> The organization employs [Selection: allow-all, deny-by-exception; deny-all, permit-by-exception] policy for allowing [Assignment: organization-defined information systems] to connect to external information systems.

Note: Organizations can constrain information system connectivity to external domains (e.g., websites) by employing one of two policies with regard to such connectivity: (i) allow-all, deny by exception, also known as blacklisting (the weaker of the two policies); or (ii) deny-all, allow by exception, also known as whitelisting (the stronger of the two policies). For either policy, organizations determine what exceptions, if any, are acceptable.

Related Controls: *CM-7*

Control Family: SECURITY ASSESSMENT AND AUTHORIZATION

CA-4 : SECURITY CERTIFICATION

> [Withdrawn: Incorporated into CA-2].

Control Family: SECURITY ASSESSMENT AND AUTHORIZATION

CA-5 : PLAN OF ACTION AND MILESTONES

Priority: P3

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: Plans of action and milestones are key documents in security authorization packages and are subject to federal reporting requirements established by OMB.

Related Controls: *CA-2*, *CA-7*, *CM-4*, *PM-4*

CA-5a.

Develops a plan of action and milestones for the information system to document the organizations planned remedial actions to correct weaknesses or deficiencies noted during the assessment of the security controls and to reduce or eliminate known vulnerabilities in the system; and

CA-5b.

Updates existing plan of action and milestones [Assignment: organization-defined frequency] based on the findings from security controls assessments, security impact analyses, and continuous monitoring activities.

CA-5 (1) : AUTOMATION SUPPORT FOR ACCURACY / CURRENCY

The organization employs automated mechanisms to help ensure that the plan of action and milestones for the information system is accurate, up to date, and readily available.

Control Family: SECURITY ASSESSMENT AND AUTHORIZATION

CA-6 : SECURITY AUTHORIZATION

Priority: P2

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: Security authorizations are official management decisions, conveyed through authorization decision documents, by senior organizational officials or executives (i.e., authorizing officials) to authorize operation of information systems and to explicitly accept the risk to organizational operations and assets, individuals, other organizations, and the Nation based on the implementation of agreed-upon security controls. Authorizing officials provide budgetary oversight for organizational information systems or assume responsibility for the mission/business operations supported by those systems. The security authorization process is an inherently federal responsibility and therefore, authorizing officials must be federal employees. Through the security authorization process, authorizing officials assume responsibility and are accountable for security risks associated with the operation and use of organizational information systems. Accordingly, authorizing officials are in positions with levels of authority commensurate with understanding and accepting such information security-related risks. OMB policy requires that organizations conduct ongoing authorizations of information systems by implementing continuous monitoring programs. Continuous monitoring programs can satisfy three-year reauthorization requirements, so separate reauthorization processes are not necessary. Through the employment of comprehensive continuous monitoring processes, critical information contained in authorization packages (i.e., security plans, security assessment reports, and plans of action and milestones) is updated on an ongoing basis, providing authorizing officials and information system owners with an up-to-date status of the security state of organizational information systems and environments of operation. To reduce the administrative cost of security reauthorization, authorizing officials use the results of continuous monitoring processes to the maximum extent possible as the basis for rendering reauthorization decisions.

Related Controls: *CA-2*, *CA-7*, *PM-9*, *PM-10*

CA-6a.

Assigns a senior-level executive or manager as the authorizing official for the information system;

CA-6b.

Ensures that the authorizing official authorizes the information system for processing before commencing operations; and

CA-6c.

Updates the security authorization [Assignment: organization-defined frequency].

Control Family: SECURITY ASSESSMENT AND AUTHORIZATION

CA-7 : CONTINUOUS MONITORING

Priority: P2

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization develops a continuous monitoring strategy and implements a continuous monitoring program that includes:

Note: Continuous monitoring programs facilitate ongoing awareness of threats, vulnerabilities, and information security to support organizational risk management decisions. The terms continuous and ongoing imply that organizations assess/analyze security controls and information security-related risks at a frequency sufficient to support organizational risk-based decisions. The results of continuous monitoring programs generate appropriate risk response actions by organizations. Continuous monitoring programs also allow organizations to maintain the security authorizations of information systems and common controls over time in highly dynamic environments of operation with changing mission/business needs, threats, vulnerabilities, and technologies. Having access to security-related information on a continuing basis through reports/dashboards gives organizational officials the capability to make more effective and timely risk management decisions, including ongoing security authorization decisions. Automation supports more frequent updates to security authorization packages, hardware/software/firmware inventories, and other system information. Effectiveness is further enhanced when continuous monitoring outputs are formatted to provide information that is specific, measurable, actionable, relevant, and timely. Continuous monitoring activities are scaled in accordance with the security categories of information systems.

Related Controls: *CA-2*, *CA-5*, *CA-6*, *CM-3*, *CM-4*, *PM-6*, *PM-9*, *RA-5*, *SA-11*, *SA-12*, *SI-2*, *SI-4*

CA-7a.

Establishment of [Assignment: organization-defined metrics] to be monitored;

CA-7b.

Establishment of [Assignment: organization-defined frequencies] for monitoring and [Assignment: organization-defined frequencies] for assessments supporting such monitoring;

CA-7c.

Ongoing security control assessments in accordance with the organizational continuous monitoring strategy;

CA-7d.

Ongoing security status monitoring of organization-defined metrics in accordance with the organizational continuous monitoring strategy;

CA-7e.

Correlation and analysis of security-related information generated by assessments and monitoring;

CA-7f.

Response actions to address results of the analysis of security-related information; and

CA-7g.

Reporting the security status of organization and the information system to [Assignment: organization-defined personnel or roles] [Assignment: organization-defined frequency].

CA-7 (1) : INDEPENDENT ASSESSMENT

Baseline-Impact: *MODERATE*, **HIGH**

> The organization employs assessors or assessment teams with [Assignment: organization-defined level of independence] to monitor the security controls in the information system on an ongoing basis.

Note: Organizations can maximize the value of assessments of security controls during the continuous monitoring process by requiring that such assessments be conducted by assessors or assessment teams with appropriate levels of independence based on continuous monitoring strategies. Assessor independence provides a degree of impartiality to the monitoring process. To achieve such impartiality, assessors should not: (i) create a mutual or conflicting interest with the organizations where the assessments are being conducted; (ii) assess their own work; (iii) act as management or employees of the organizations they are serving; or (iv) place themselves in advocacy positions for the organizations acquiring their services.

CA-7 (2) : TYPES OF ASSESSMENTS

> [Withdrawn: Incorporated into CA-2].

CA-7 (3) : TREND ANALYSES

> The organization employs trend analyses to determine if security control implementations, the frequency of continuous monitoring activities, and/or the types of activities used in the continuous monitoring process need to be modified based on empirical data.

Note: Trend analyses can include, for example, examining recent threat information regarding the types of threat events that have occurred within the organization or across the federal government, success rates of certain types of cyber attacks, emerging vulnerabilities in information technologies, evolving social engineering techniques, results from multiple security control assessments, the effectiveness of configuration settings, and findings from Inspectors General or auditors.

Control Family: SECURITY ASSESSMENT AND AUTHORIZATION

CA-8 : PENETRATION TESTING

Priority: P2

Baseline-Impact: **HIGH**

> The organization conducts penetration testing [Assignment: organization-defined frequency] on [Assignment: organization-defined information systems or system components].

Note: Penetration testing is a specialized type of assessment conducted on information systems or individual system components to identify vulnerabilities that could be exploited by adversaries. Such testing can be used to either validate vulnerabilities or determine the degree of resistance organizational information systems have to adversaries within a set of specified constraints (e.g., time, resources, and/or skills). Penetration testing attempts to duplicate the actions of adversaries in carrying out hostile cyber attacks against organizations and provides a more in-depth analysis of security-related weaknesses/deficiencies. Organizations can also use the results of vulnerability analyses to support penetration testing activities. Penetration testing can be conducted on the hardware, software, or firmware components of an information system and can exercise both physical and technical security controls. A standard method for penetration testing includes, for example: (i) pretest analysis based on full knowledge of the target system; (ii) pretest identification of potential vulnerabilities based on pretest analysis; and (iii) testing designed to determine exploitability of identified vulnerabilities. All parties agree to the rules of engagement before the commencement of penetration testing scenarios. Organizations correlate the penetration testing rules of engagement with the tools, techniques, and procedures that are anticipated to be employed by adversaries carrying out attacks. Organizational risk assessments guide decisions on the level of independence required for personnel conducting penetration testing.

Related Controls: *SA-12*

CA-8 (1) : INDEPENDENT PENETRATION AGENT OR TEAM

> The organization employs an independent penetration agent or penetration team to perform penetration testing on the information system or system components.

Note: Independent penetration agents or teams are individuals or groups who conduct impartial penetration testing of organizational information systems. Impartiality implies that penetration agents or teams are free from any perceived or actual conflicts of interest with regard to the development, operation, or management of the information systems that are the targets of the penetration testing. Supplemental guidance for CA-2 (1) provides additional information regarding independent assessments that can be applied to penetration testing.

Related Controls: *CA-2*

CA-8 (2) : RED TEAM EXERCISES

> The organization employs [Assignment: organization-defined red team exercises] to simulate attempts by adversaries to compromise organizational information systems in accordance with [Assignment: organization-defined rules of engagement].

Note: Red team exercises extend the objectives of penetration testing by examining the security posture of organizations and their ability to implement effective cyber defenses. As such, red team exercises reflect simulated adversarial attempts to compromise organizational mission/business functions and provide a comprehensive assessment of the security state of information systems and organizations. Simulated adversarial attempts to compromise organizational missions/business functions and the information systems that support those missions/functions may include technology-focused attacks (e.g., interactions with hardware, software, or firmware components and/or mission/business processes) and social engineering-based attacks (e.g., interactions via email, telephone, shoulder surfing, or personal conversations). While penetration testing may be largely laboratory-based testing, organizations use red team exercises to provide more comprehensive assessments that reflect real-world conditions. Red team exercises can be used to improve security awareness and training and to assess levels of security control effectiveness.

Control Family: SECURITY ASSESSMENT AND AUTHORIZATION

CA-9 : INTERNAL SYSTEM CONNECTIONS

Priority: P2

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: This control applies to connections between organizational information systems and (separate) constituent system components (i.e., intra-system connections) including, for example, system connections with mobile devices, notebook/desktop computers, printers, copiers, facsimile machines, scanners, sensors, and servers. Instead of authorizing each individual internal connection, organizations can authorize internal connections for a class of components with common characteristics and/or configurations, for example, all digital printers, scanners, and copiers with a specified processing, storage, and transmission capability or all smart phones with a specific baseline configuration.

Related Controls: *AC-3*, *AC-4*, *AC-18*, *AC-19*, *AU-2*, *AU-12*, *CA-7*, *CM-2*, *IA-3*, *SC-7*, *SI-4*

CA-9a.

Authorizes internal connections of [Assignment: organization-defined information system components or classes of components] to the information system; and

CA-9b.

Documents, for each internal connection, the interface characteristics, security requirements, and the nature of the information communicated.

CA-9 (1) : SECURITY COMPLIANCE CHECKS

The information system performs security compliance checks on constituent system components prior to the establishment of the internal connection.

Note: Security compliance checks may include, for example, verification of the relevant baseline configuration.

Related Controls: *CM-6*

Control Family: CONFIGURATION MANAGEMENT

CM-1 : CONFIGURATION MANAGEMENT POLICY AND PROCEDURES

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: This control addresses the establishment of policy and procedures for the effective implementation of selected security controls and control enhancements in the CM family. Policy and procedures reflect applicable federal laws, Executive Orders, directives, regulations, policies, standards, and guidance. Security program policies and procedures at the organization level may make the need for system-specific policies and procedures unnecessary. The policy can be included as part of the general information security policy for organizations or conversely, can be represented by multiple policies reflecting the complex nature of certain organizations. The procedures can be established for the security program in general and for particular information systems, if needed. The organizational risk management strategy is a key factor in establishing policy and procedures.

Related Controls: *PM-9*

CM-1a.

Develops, documents, and disseminates to [Assignment: organization-defined personnel or roles]:

CM-1a.1.

A configuration management policy that addresses purpose, scope, roles, responsibilities, management commitment, coordination among organizational entities, and compliance; and

CM-1a.2.

Procedures to facilitate the implementation of the configuration management policy and associated configuration management controls; and

CM-1b.

Reviews and updates the current:

CM-1b.1.

Configuration management policy [Assignment: organization-defined frequency]; and

CM-1b.2.

Configuration management procedures [Assignment: organization-defined frequency].

Control Family: CONFIGURATION MANAGEMENT

CM-2 : BASELINE CONFIGURATION

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The organization develops, documents, and maintains under configuration control, a current baseline configuration of the information system.

Note: This control establishes baseline configurations for information systems and system components including communications and connectivity-related aspects of systems. Baseline configurations are documented, formally reviewed and agreed-upon sets of specifications for information systems or configuration items within those systems. Baseline configurations serve as a basis for future builds, releases, and/or changes to information systems. Baseline configurations include information about information system components (e.g., standard software packages installed on workstations, notebook computers, servers, network components, or mobile devices; current version numbers and patch information on operating systems and applications; and configuration settings/parameters), network topology, and the logical placement of those components within the system architecture. Maintaining baseline configurations requires creating new baselines as organizational information systems change over time. Baseline configurations of information systems reflect the current enterprise architecture.

Related Controls: *CM-3*, *CM-6*, *CM-8*, *CM-9*, *SA-10*, *PM-5*, *PM-7*

CM-2 (1) : REVIEWS AND UPDATES

Baseline-Impact: *MODERATE*, **HIGH**

> The organization reviews and updates the baseline configuration of the information system:

Related Controls: *CM-5*

CM-2 (1)(a)

> [Assignment: organization-defined frequency];

CM-2 (1)(b)

> When required due to [Assignment organization-defined circumstances]; and

CM-2 (1)(c)

> As an integral part of information system component installations and upgrades.

CM-2 (2) : AUTOMATION SUPPORT FOR ACCURACY / CURRENCY

Baseline-Impact: HIGH

> The organization employs automated mechanisms to maintain an up-to-date, complete, accurate, and readily available baseline configuration of the information system.

Note: Automated mechanisms that help organizations maintain consistent baseline configurations for information systems include, for example, hardware and software inventory tools, configuration management tools, and network management tools. Such tools can be deployed and/or allocated as common controls, at the information system level, or at the operating system or component level (e.g., on workstations, servers, notebook computers, network components, or mobile devices). Tools can be used, for example, to track version numbers on operating system applications, types of software installed, and current patch levels. This control enhancement can be satisfied by the implementation of CM-8 (2) for organizations that choose to combine information system component inventory and baseline configuration activities.

Related Controls: *CM-7*, *RA-5*

CM-2 (3) : RETENTION OF PREVIOUS CONFIGURATIONS

Baseline-Impact: *MODERATE*, **HIGH**

> The organization retains [Assignment: organization-defined previous versions of baseline configurations of the information system] to support rollback.

Note: Retaining previous versions of baseline configurations to support rollback may include, for example, hardware, software, firmware, configuration files, and configuration records.

CM-2 (4) : UNAUTHORIZED SOFTWARE

> [Withdrawn: Incorporated into CM-7].

CM-2 (5) : AUTHORIZED SOFTWARE

> [Withdrawn: Incorporated into CM-7].

CM-2 (6) : DEVELOPMENT AND TEST ENVIRONMENTS

> The organization maintains a baseline configuration for information system development and test environments that is managed separately from the operational baseline configuration.

Note: Establishing separate baseline configurations for development, testing, and operational environments helps protect information systems from unplanned/unexpected events related to development and testing activities. Separate baseline configurations allow organizations to apply the configuration management that is most appropriate for each type of configuration. For example, management of operational configurations typically emphasizes the need for stability, while management of development/test configurations requires greater flexibility. Configurations in the test environment mirror the configurations in the operational environment to the extent practicable so that the results of the testing are representative of the proposed changes to the operational systems. This control enhancement requires separate configurations but not necessarily separate physical environments.

Related Controls: *CM-4*, *SC-3*, *SC-7*

CM-2 (7) : CONFIGURE SYSTEMS, COMPONENTS, OR DEVICES FOR HIGH-RISK AREAS

Baseline-Impact: *MODERATE*, **HIGH**

The organization:

Note: When it is known that information systems, system components, or devices (e.g., notebook computers, mobile devices) will be located in high-risk areas, additional security controls may be implemented to counter the greater threat in such areas coupled with the lack of physical security relative to organizational-controlled areas. For example, organizational policies and procedures for notebook computers used by individuals departing on and returning from travel include, for example, determining which locations are of concern, defining required configurations for the devices, ensuring that the devices are configured as intended before travel is initiated, and applying specific safeguards to the device after travel is completed. Specially configured notebook computers include, for example, computers with sanitized hard drives, limited applications, and additional hardening (e.g., more stringent configuration settings). Specified safeguards applied to mobile devices upon return from travel include, for example, examining the device for signs of physical tampering and purging/reimaging the hard disk drive. Protecting information residing on mobile devices is covered in the media protection family.

CM-2 (7)(a)

Issues [Assignment: organization-defined information systems, system components, or devices] with [Assignment: organization-defined configurations] to individuals traveling to locations that the organization deems to be of significant risk; and

CM-2 (7)(b)

Applies [Assignment: organization-defined security safeguards] to the devices when the individuals return.

Control Family: CONFIGURATION MANAGEMENT

CM-3 : CONFIGURATION CHANGE CONTROL

Priority: P1

Baseline-Impact: *MODERATE*, **HIGH**

The organization:

Note: Configuration change controls for organizational information systems involve the systematic proposal, justification, implementation, testing, review, and disposition of changes to the systems, including system upgrades and modifications. Configuration change control includes changes to baseline configurations for components and configuration items of information systems, changes to configuration settings for information technology products (e.g., operating systems, applications, firewalls, routers, and mobile devices), unscheduled/unauthorized changes, and changes to remediate vulnerabilities. Typical processes for managing configuration changes to information systems include, for example, Configuration Control Boards that approve proposed changes to systems. For new development information systems or systems undergoing major upgrades, organizations consider including representatives from development organizations on the Configuration Control Boards. Auditing of changes includes activities before and after changes are made to organizational information systems and the auditing activities required to implement such changes.

Related Controls: *CA-7*, *CM-2*, *CM-4*, *CM-5*, *CM-6*, *CM-9*, *SA-10*, *SI-2*, *SI-12*

CM-3a.

Determines the types of changes to the information system that are configuration-controlled;

CM-3b.

Reviews proposed configuration-controlled changes to the information system and approves or disapproves such changes with explicit consideration for security impact analyses;

CM-3c.

Documents configuration change decisions associated with the information system;

CM-3d.

Implements approved configuration-controlled changes to the information system;

CM-3e.

Retains records of configuration-controlled changes to the information system for [Assignment: organization-defined time period];

CM-3f.

Audits and reviews activities associated with configuration-controlled changes to the information system; and

CM-3g.

Coordinates and provides oversight for configuration change control activities through [Assignment: organization-defined configuration change control element (e.g., committee, board)] that convenes [Selection (one or more): [Assignment: organization-defined frequency]; [Assignment: organization-defined configuration change conditions]].

CM-3 (1) : AUTOMATED DOCUMENT / NOTIFICATION / PROHIBITION OF CHANGES

Baseline-Impact: HIGH

The organization employs automated mechanisms to:

CM-3 (1)(a)

Document proposed changes to the information system;

CM-3 (1)(b)

Notify [Assignment: organized-defined approval authorities] of proposed changes to the information system and request change approval;

CM-3 (1)(c)

Highlight proposed changes to the information system that have not been approved or disapproved by [Assignment: organization-defined time period];

CM-3 (1)(d)

Prohibit changes to the information system until designated approvals are received;

CM-3 (1)(e)

Document all changes to the information system; and

CM-3 (1)(f)

Notify [Assignment: organization-defined personnel] when approved changes to the information system are completed.

CM-3 (2) : TEST / VALIDATE / DOCUMENT CHANGES

Baseline-Impact: *MODERATE*, **HIGH**

The organization tests, validates, and documents changes to the information system before implementing the changes on the operational system.

Note: Changes to information systems include modifications to hardware, software, or firmware components and configuration settings defined in CM-6. Organizations ensure that testing does not interfere with information system operations. Individuals/groups conducting tests understand organizational security policies and procedures, information system security policies and procedures, and the specific health, safety, and environmental risks associated with particular facilities/processes. Operational systems may need to be taken off-line, or replicated to the extent feasible, before testing can be conducted. If information systems must be taken off-line for testing, the tests are scheduled to occur during planned system outages whenever possible. If testing cannot be conducted on operational systems, organizations employ compensating controls (e.g., testing on replicated systems).

CM-3 (3) : AUTOMATED CHANGE IMPLEMENTATION

> The organization employs automated mechanisms to implement changes to the current information system baseline and deploys the updated baseline across the installed base.

CM-3 (4) : SECURITY REPRESENTATIVE

> The organization requires an information security representative to be a member of the [Assignment: organization-defined configuration change control element].

Note: Information security representatives can include, for example, senior agency information security officers, information system security officers, or information system security managers. Representation by personnel with information security expertise is important because changes to information system configurations can have unintended side effects, some of which may be security-relevant. Detecting such changes early in the process can help avoid unintended, negative consequences that could ultimately affect the security state of organizational information systems. The configuration change control element in this control enhancement reflects the change control elements defined by organizations in CM-3.

CM-3 (5) : AUTOMATED SECURITY RESPONSE

> The information system implements [Assignment: organization-defined security responses] automatically if baseline configurations are changed in an unauthorized manner.

Note: Security responses include, for example, halting information system processing, halting selected system functions, or issuing alerts/notifications to organizational personnel when there is an unauthorized modification of a configuration item.

CM-3 (6) : CRYPTOGRAPHY MANAGEMENT

> The organization ensures that cryptographic mechanisms used to provide [Assignment: organization-defined security safeguards] are under configuration management.

Note: Regardless of the cryptographic means employed (e.g., public key, private key, shared secrets), organizations ensure that there are processes and procedures in place to effectively manage those means. For example, if devices use certificates as a basis for identification and authentication, there needs to be a process in place to address the expiration of those certificates.

Related Controls: *SC-13*

Control Family: CONFIGURATION MANAGEMENT

CM-4 : SECURITY IMPACT ANALYSIS

Priority: P2

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The organization analyzes changes to the information system to determine potential security impacts prior to change implementation.

Note: Organizational personnel with information security responsibilities (e.g., Information System Administrators, Information System Security Officers, Information System Security Managers, and Information System Security Engineers) conduct security impact analyses. Individuals conducting security impact analyses possess the necessary skills/technical expertise to analyze the changes to information systems and the associated security ramifications. Security impact analysis may include, for example, reviewing security plans to understand security control requirements and reviewing system design documentation to understand control implementation and how specific changes might affect the controls. Security impact analyses may also include assessments of risk to better understand the impact of the changes and to determine if additional security controls are required. Security impact analyses are scaled in accordance with the security categories of the information systems.

Related Controls: *CA-2*, *CA-7*, *CM-3*, *CM-9*, *SA-4*, *SA-5*, *SA-10*, *SI-2*

CM-4 (1) : SEPARATE TEST ENVIRONMENTS

Baseline-Impact: HIGH

> The organization analyzes changes to the information system in a separate test environment before implementation in an operational environment, looking for security impacts due to flaws, weaknesses, incompatibility, or intentional malice.

Note: Separate test environment in this context means an environment that is physically or logically isolated and distinct from the operational environment. The separation is sufficient to ensure that activities in the test environment do not impact activities in the operational environment, and information in the operational environment is not inadvertently transmitted to the test environment. Separate environments can be achieved by physical or logical means. If physically separate test environments are not used, organizations determine the strength of mechanism required when implementing logical separation (e.g., separation achieved through virtual machines).

Related Controls: *SA-11*, *SC-3*, *SC-7*

CM-4 (2) : VERIFICATION OF SECURITY FUNCTIONS

> The organization, after the information system is changed, checks the security functions to verify that the functions are implemented correctly, operating as intended, and producing the desired outcome with regard to meeting the security requirements for the system.

Note: Implementation is this context refers to installing changed code in the operational information system.

Related Controls: *SA-11*

Control Family: CONFIGURATION MANAGEMENT

CM-5 : ACCESS RESTRICTIONS FOR CHANGE

Priority: P1

Baseline-Impact: *MODERATE*, **HIGH**

> The organization defines, documents, approves, and enforces physical and logical access restrictions associated with changes to the information system.

Note: Any changes to the hardware, software, and/or firmware components of information systems can potentially have significant effects on the overall security of the systems. Therefore, organizations permit only qualified and authorized individuals to access information systems for purposes of initiating changes, including upgrades and modifications. Organizations maintain records of access to ensure that configuration change control is implemented and to support after-the-fact actions should organizations discover any unauthorized changes. Access restrictions for change also include software libraries. Access restrictions include, for example, physical and logical access controls (see AC-3 and PE-3), workflow automation, media libraries, abstract layers (e.g., changes implemented into third-party interfaces rather than directly into information systems), and change windows (e.g., changes occur only during specified times, making unauthorized changes easy to discover).

Related Controls: *AC-3*, *AC-6*, *PE-3*

CM-5 (1) : AUTOMATED ACCESS ENFORCEMENT / AUDITING

Baseline-Impact: HIGH

> The information system enforces access restrictions and supports auditing of the enforcement actions.

Related Controls: *AU-2*, *AU-12*, *AU-6*, *CM-3*, *CM-6*

CM-5 (2) : REVIEW SYSTEM CHANGES

Baseline-Impact: HIGH

> The organization reviews information system changes [Assignment: organization-defined frequency] and [Assignment: organization-defined circumstances] to determine whether unauthorized changes have occurred.

Note: Indications that warrant review of information system changes and the specific circumstances justifying such reviews may be obtained from activities carried out by organizations during the configuration change process.

Related Controls: *AU-6*, *AU-7*, *CM-3*, *CM-5*, *PE-6*, *PE-8*

CM-5 (3) : SIGNED COMPONENTS

Baseline-Impact: HIGH

> The information system prevents the installation of [Assignment: organization-defined software and firmware components] without verification that the component has been digitally signed using a certificate that is recognized and approved by the organization.

Note: Software and firmware components prevented from installation unless signed with recognized and approved certificates include, for example, software and firmware version updates, patches, service packs, device drivers, and basic input output system (BIOS) updates. Organizations can identify applicable software and firmware components by type, by specific items, or a combination of both. Digital signatures and organizational verification of such signatures, is a method of code authentication.

Related Controls: *CM-7*, *SC-13*, *SI-7*

CM-5 (4) : DUAL AUTHORIZATION

> The organization enforces dual authorization for implementing changes to [Assignment: organization-defined information system components and system-level information].

Note: Organizations employ dual authorization to ensure that any changes to selected information system components and information cannot occur unless two qualified individuals implement such changes. The two individuals possess sufficient skills/expertise to determine if the proposed changes are correct implementations of approved changes. Dual authorization may also be known as two-person control.

Related Controls: *AC-5*, *CM-3*

CM-5 (5) : LIMIT PRODUCTION / OPERATIONAL PRIVILEGES

> The organization:

Note: In many organizations, information systems support multiple core missions/business functions. Limiting privileges to change information system components with respect to operational systems is necessary because changes to a particular information system component may have far-reaching effects on mission/business processes supported by the system where the component resides. The complex, many-to-many relationships between systems and mission/business processes are in some cases, unknown to developers.

Related Controls: *AC-2*

CM-5 (5)(a)

Limits privileges to change information system components and system-related information within a production or operational environment; and

CM-5 (5)(b)

Reviews and reevaluates privileges [Assignment: organization-defined frequency].

CM-5 (6) : LIMIT LIBRARY PRIVILEGES

The organization limits privileges to change software resident within software libraries.

Note: Software libraries include privileged programs.

Related Controls: *AC-2*

CM-5 (7) : AUTOMATIC IMPLEMENTATION OF SECURITY SAFEGUARDS

[Withdrawn: Incorporated into SI-7].

Control Family: CONFIGURATION MANAGEMENT

CM-6 : CONFIGURATION SETTINGS

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note:

Configuration settings are the set of parameters that can be changed in hardware, software, or firmware components of the information system that affect the security posture and/or functionality of the system. Information technology products for which security-related configuration settings can be defined include, for example, mainframe computers, servers (e.g., database, electronic mail, authentication, web, proxy, file, domain name), workstations, input/output devices (e.g., scanners, copiers, and printers), network components (e.g., firewalls, routers, gateways, voice and data switches, wireless access points, network appliances, sensors), operating systems, middleware, and applications. Security-related parameters are those parameters impacting the security state of information systems including the parameters required to satisfy other security control requirements. Security-related parameters include, for example: (i) registry settings; (ii) account, file, directory permission settings; and (iii) settings for functions, ports, protocols, services, and remote connections. Organizations establish organization-wide configuration settings and subsequently derive specific settings for information systems. The established settings become part of the systems configuration baseline.

Common secure configurations (also referred to as security configuration checklists, lockdown and hardening guides, security reference guides, security technical implementation guides) provide recognized, standardized, and established benchmarks that stipulate secure configuration settings for specific information technology platforms/products and instructions for configuring those information system components to meet operational requirements. Common secure configurations can be developed by a variety of organizations including, for example, information technology product developers, manufacturers, vendors, consortia, academia, industry, federal agencies, and other organizations in the public and private sectors. Common secure configurations include the United States Government Configuration Baseline (USGCB) which affects the implementation of CM-6 and other controls such as AC-19 and CM-7. The Security Content Automation Protocol (SCAP) and the defined standards within the protocol (e.g., Common Configuration Enumeration) provide an effective method to uniquely identify, track, and control configuration settings. OMB establishes federal policy on configuration requirements for federal information systems.

Related Controls: *AC-19*, *CM-2*, *CM-3*, *CM-7*, *SI-4*

CM-6a.

Establishes and documents configuration settings for information technology products employed within the information system using [Assignment: organization-defined security configuration checklists] that reflect the most restrictive mode consistent with operational requirements;

CM-6b.

Implements the configuration settings;

CM-6c.

Identifies, documents, and approves any deviations from established configuration settings for [Assignment: organization-defined information system components] based on [Assignment: organization-defined operational requirements]; and

CM-6d.

Monitors and controls changes to the configuration settings in accordance with organizational policies and procedures.

CM-6 (1) : AUTOMATED CENTRAL MANAGEMENT / APPLICATION / VERIFICATION

Baseline-Impact: HIGH

The organization employs automated mechanisms to centrally manage, apply, and verify configuration settings for [Assignment: organization-defined information system components].

Related Controls: *CA-7*, *CM-4*

CM-6 (2) : RESPOND TO UNAUTHORIZED CHANGES

Baseline-Impact: **HIGH**

The organization employs [Assignment: organization-defined security safeguards] to respond to unauthorized changes to [Assignment: organization-defined configuration settings].

Note: Responses to unauthorized changes to configuration settings can include, for example, alerting designated organizational personnel, restoring established configuration settings, or in extreme cases, halting affected information system processing.

Related Controls: *IR-4*, *SI-7*

CM-6 (3) : UNAUTHORIZED CHANGE DETECTION

[Withdrawn: Incorporated into SI-7].

CM-6 (4) : CONFORMANCE DEMONSTRATION

[Withdrawn: Incorporated into CM-4].

Control Family: CONFIGURATION MANAGEMENT

CM-7 : LEAST FUNCTIONALITY

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: Information systems can provide a wide variety of functions and services. Some of the functions and services, provided by default, may not be necessary to support essential organizational operations (e.g., key missions, functions). Additionally, it is sometimes convenient to provide multiple services from single information system components, but doing so increases risk over limiting the services provided by any one component. Where feasible, organizations limit component functionality to a single function per device (e.g., email servers or web servers, but not both). Organizations review functions and services provided by information systems or individual components of information systems, to determine which functions and services are candidates for elimination (e.g., Voice Over Internet Protocol, Instant Messaging, auto-execute, and file sharing). Organizations consider disabling unused or unnecessary physical and logical ports/protocols (e.g., Universal Serial Bus, File Transfer Protocol, and Hyper Text Transfer Protocol) on information systems to prevent unauthorized connection of devices, unauthorized transfer of information, or unauthorized tunneling. Organizations can utilize network scanning tools, intrusion detection and prevention systems, and end-point protections such as firewalls and host-based intrusion detection systems to identify and prevent the use of prohibited functions, ports, protocols, and services.

Related Controls: *AC-6*, *CM-2*, *RA-5*, *SA-5*, *SC-7*

CM-7a.

Configures the information system to provide only essential capabilities; and

CM-7b.

Prohibits or restricts the use of the following functions, ports, protocols, and/or services: [Assignment: organization-defined prohibited or restricted functions, ports, protocols, and/or services].

CM-7 (1) : PERIODIC REVIEW

Baseline-Impact: *MODERATE*, **HIGH**

The organization:

Note: The organization can either make a determination of the relative security of the function, port, protocol, and/or service or base the security decision on the assessment of other entities. Bluetooth, FTP, and peer-to-peer networking are examples of less than secure protocols.

Related Controls: *AC-18*, *CM-7*, *IA-2*

CM-7 (1)(a)

Reviews the information system [Assignment: organization-defined frequency] to identify unnecessary and/or nonsecure functions, ports, protocols, and services; and

CM-7 (1)(b)

Disables [Assignment: organization-defined functions, ports, protocols, and services within the information system deemed to be unnecessary and/or nonsecure].

CM-7 (2) : PREVENT PROGRAM EXECUTION

Baseline-Impact: *MODERATE*, **HIGH**

The information system prevents program execution in accordance with [Selection (one or more): [Assignment: organization-defined policies regarding software program usage and restrictions]; rules authorizing the terms and conditions of software program usage].

Related Controls: *CM-8*, *PM-5*

CM-7 (3) : REGISTRATION COMPLIANCE

The organization ensures compliance with [Assignment: organization-defined registration requirements for functions, ports, protocols, and services].

Note: Organizations use the registration process to manage, track, and provide oversight for information systems and implemented functions, ports, protocols, and services.

CM-7 (4) : UNAUTHORIZED SOFTWARE / BLACKLISTING

Baseline-Impact: *MODERATE*,

The organization:

Note: The process used to identify software programs that are not authorized to execute on organizational information systems is commonly referred to as blacklisting. Organizations can implement CM-7 (5) instead of this control enhancement if whitelisting (the stronger of the two policies) is the preferred approach for restricting software program execution.

Related Controls: *CM-6*, *CM-8*, *PM-5*

CM-7 (4)(a)

Identifies [Assignment: organization-defined software programs not authorized to execute on the information system];

CM-7 (4)(b)

Employs an allow-all, deny-by-exception policy to prohibit the execution of unauthorized software programs on the information system; and

CM-7 (4)(c)

Reviews and updates the list of unauthorized software programs [Assignment: organization-defined frequency].

CM-7 (5) : AUTHORIZED SOFTWARE / WHITELISTING

Baseline-Impact: HIGH

The organization:

Note: The process used to identify software programs that are authorized to execute on organizational information systems is commonly referred to as whitelisting. In addition to whitelisting, organizations consider verifying the integrity of white-listed software programs using, for example, cryptographic checksums, digital signatures, or hash functions. Verification of white-listed software can occur either prior to execution or at system startup.

Related Controls: *CM-2*, *CM-6*, *CM-8*, *PM-5*, *SA-10*, *SC-34*, *SI-7*

CM-7 (5)(a)

Identifies [Assignment: organization-defined software programs authorized to execute on the information system];

CM-7 (5)(b)

Employs a deny-all, permit-by-exception policy to allow the execution of authorized software programs on the information system; and

CM-7 (5)(c)

Reviews and updates the list of authorized software programs [Assignment: organization-defined frequency].

Control Family: CONFIGURATION MANAGEMENT

CM-8 : INFORMATION SYSTEM COMPONENT INVENTORY

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: Organizations may choose to implement centralized information system component inventories that include components from all organizational information systems. In such situations, organizations ensure that the resulting inventories include system-specific information required for proper component accountability (e.g., information system association, information system owner). Information deemed necessary for effective accountability of information system components includes, for example, hardware inventory specifications, software license information, software version numbers, component owners, and for networked components or devices, machine names and network addresses. Inventory specifications include, for example, manufacturer, device type, model, serial number, and physical location.

Related Controls: *CM-2*, *CM-6*, *PM-5*

CM-8a.

Develops and documents an inventory of information system components that:

CM-8a.1.

Accurately reflects the current information system;

CM-8a.2.

Includes all components within the authorization boundary of the information system;

CM-8a.3.

Is at the level of granularity deemed necessary for tracking and reporting; and

CM-8a.4.

Includes [Assignment: organization-defined information deemed necessary to achieve effective information system component accountability]; and

CM-8b.

Reviews and updates the information system component inventory [Assignment: organization-defined frequency].

CM-8 (1) : UPDATES DURING INSTALLATIONS / REMOVALS

Baseline-Impact: *MODERATE*, **HIGH**

The organization updates the inventory of information system components as an integral part of component installations, removals, and information system updates.

CM-8 (2) : AUTOMATED MAINTENANCE

Baseline-Impact: **HIGH**

The organization employs automated mechanisms to help maintain an up-to-date, complete, accurate, and readily available inventory of information system components.

Note: Organizations maintain information system inventories to the extent feasible. Virtual machines, for example, can be difficult to monitor because such machines are not visible to the network when not in use. In such cases, organizations maintain as up-to-date, complete, and accurate an inventory as is deemed reasonable. This control enhancement can be satisfied by the implementation of CM-2 (2) for organizations that choose to combine information system component inventory and baseline configuration activities.

Related Controls: *SI-7*

CM-8 (3) : AUTOMATED UNAUTHORIZED COMPONENT DETECTION

Baseline-Impact: *MODERATE*, **HIGH**

The organization:

Note: This control enhancement is applied in addition to the monitoring for unauthorized remote connections and mobile devices. Monitoring for unauthorized system components may be accomplished on an ongoing basis or by the periodic scanning of systems for that purpose. Automated mechanisms can be implemented within information systems or in other separate devices. Isolation can be achieved, for example, by placing unauthorized information system components in separate domains or subnets or otherwise quarantining such components. This type of component isolation is commonly referred to as sandboxing.

Related Controls: *AC-17*, *AC-18*, *AC-19*, *CA-7*, *SI-3*, *SI-4*, *SI-7*, *RA-5*

CM-8 (3)(a)

Employs automated mechanisms [Assignment: organization-defined frequency] to detect the presence of unauthorized hardware, software, and firmware components within the information system; and

CM-8 (3)(b)

Takes the following actions when unauthorized components are detected: [Selection (one or more): disables network access by such components; isolates the components; notifies [Assignment: organization-defined personnel or roles]].

CM-8 (4) : ACCOUNTABILITY INFORMATION

Baseline-Impact: HIGH

The organization includes in the information system component inventory information, a means for identifying by [Selection (one or more): name; position; role], individuals responsible/accountable for administering those components.

Note: Identifying individuals who are both responsible and accountable for administering information system components helps to ensure that the assigned components are properly administered and organizations can contact those individuals if some action is required (e.g., component is determined to be the source of a breach/compromise, component needs to be recalled/replaced, or component needs to be relocated).

CM-8 (5) : NO DUPLICATE ACCOUNTING OF COMPONENTS

Baseline-Impact: *MODERATE*, **HIGH**

> The organization verifies that all components within the authorization boundary of the information system are not duplicated in other information system component inventories.

Note: This control enhancement addresses the potential problem of duplicate accounting of information system components in large or complex interconnected systems.

CM-8 (6) : ASSESSED CONFIGURATIONS / APPROVED DEVIATIONS

> The organization includes assessed component configurations and any approved deviations to current deployed configurations in the information system component inventory.

Note: This control enhancement focuses on configuration settings established by organizations for information system components, the specific components that have been assessed to determine compliance with the required configuration settings, and any approved deviations from established configuration settings.

Related Controls: *CM-2*, *CM-6*

CM-8 (7) : CENTRALIZED REPOSITORY

> The organization provides a centralized repository for the inventory of information system components.

Note: Organizations may choose to implement centralized information system component inventories that include components from all organizational information systems. Centralized repositories of information system component inventories provide opportunities for efficiencies in accounting for organizational hardware, software, and firmware assets. Such repositories may also help organizations rapidly identify the location and responsible individuals of system components that have been compromised, breached, or are otherwise in need of mitigation actions. Organizations ensure that the resulting centralized inventories include system-specific information required for proper component accountability (e.g., information system association, information system owner).

CM-8 (8) : AUTOMATED LOCATION TRACKING

> The organization employs automated mechanisms to support tracking of information system components by geographic location.

Note: The use of automated mechanisms to track the location of information system components can increase the accuracy of component inventories. Such capability may also help organizations rapidly identify the location and responsible individuals of system components that have been compromised, breached, or are otherwise in need of mitigation actions.

CM-8 (9) : ASSIGNMENT OF COMPONENTS TO SYSTEMS

The organization:

Note: Organizations determine the criteria for or types of information system components (e.g., microprocessors, motherboards, software, programmable logic controllers, and network devices) that are subject to this control enhancement.

Related Controls: *SA-4*

CM-8 (9)(a)

Assigns [Assignment: organization-defined acquired information system components] to an information system; and

CM-8 (9)(b)

Receives an acknowledgment from the information system owner of this assignment.

Control Family: CONFIGURATION MANAGEMENT

CM-9 : CONFIGURATION MANAGEMENT PLAN

Priority: P1

Baseline-Impact: *MODERATE*, **HIGH**

The organization develops, documents, and implements a configuration management plan for the information system that:

Note: Configuration management plans satisfy the requirements in configuration management policies while being tailored to individual information systems. Such plans define detailed processes and procedures for how configuration management is used to support system development life cycle activities at the information system level. Configuration management plans are typically developed during the development/acquisition phase of the system development life cycle. The plans describe how to move changes through change management processes, how to update configuration settings and baselines, how to maintain information system component inventories, how to control development, test, and operational environments, and how to develop, release, and update key documents. Organizations can employ templates to help ensure consistent and timely development and implementation of configuration management plans. Such templates can represent a master configuration management plan for the organization at large with subsets of the plan implemented on a system by system basis. Configuration management approval processes include designation of key management stakeholders responsible for reviewing and approving proposed changes to information systems, and personnel that conduct security impact analyses prior to the implementation of changes to the systems. Configuration items are the information system items (hardware, software, firmware, and documentation) to be configuration-managed. As information systems continue through the system development life cycle, new configuration items may be identified and some existing configuration items may no longer need to be under configuration control.

Related Controls: *CM-2*, *CM-3*, *CM-4*, *CM-5*, *CM-8*, *SA-10*

CM-9a.

Addresses roles, responsibilities, and configuration management processes and procedures;

CM-9b.

Establishes a process for identifying configuration items throughout the system development life cycle and for managing the configuration of the configuration items;

CM-9c.

Defines the configuration items for the information system and places the configuration items under configuration management; and

CM-9d.

Protects the configuration management plan from unauthorized disclosure and modification.

CM-9 (1) : ASSIGNMENT OF RESPONSIBILITY

The organization assigns responsibility for developing the configuration management process to organizational personnel that are not directly involved in information system development.

Note: In the absence of dedicated configuration management teams assigned within organizations, system developers may be tasked to develop configuration management processes using personnel who are not directly involved in system development or integration. This separation of duties ensures that organizations establish and maintain a sufficient degree of independence between the information system development and integration processes and configuration management processes to facilitate quality control and more effective oversight.

Control Family: CONFIGURATION MANAGEMENT

CM-10 : SOFTWARE USAGE RESTRICTIONS

Priority: P2

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: Software license tracking can be accomplished by manual methods (e.g., simple spreadsheets) or automated methods (e.g., specialized tracking applications) depending on organizational needs.

Related Controls: *AC-17*, *CM-8*, *SC-7*

CM-10a.

Uses software and associated documentation in accordance with contract agreements and copyright laws;

CM-10b.

Tracks the use of software and associated documentation protected by quantity licenses to control copying and distribution; and

CM-10c.

Controls and documents the use of peer-to-peer file sharing technology to ensure that this capability is not used for the unauthorized distribution, display, performance, or reproduction of copyrighted work.

CM-10 (1) : OPEN SOURCE SOFTWARE

The organization establishes the following restrictions on the use of open source software: [Assignment: organization-defined restrictions].

Note: Open source software refers to software that is available in source code form. Certain software rights normally reserved for copyright holders are routinely provided under software license agreements that permit individuals to study, change, and improve the software. From a security perspective, the major advantage of open source software is that it provides organizations with the ability to examine the source code. However, there are also various licensing issues associated with open source software including, for example, the constraints on derivative use of such software.

Control Family: CONFIGURATION MANAGEMENT

CM-11 : USER-INSTALLED SOFTWARE

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: If provided the necessary privileges, users have the ability to install software in organizational information systems. To maintain control over the types of software installed, organizations identify permitted and prohibited actions regarding software installation. Permitted software installations may include, for example, updates and security patches to existing software and downloading applications from organization-approved app stores. Prohibited software installations may include, for example, software with unknown or suspect pedigrees or software that organizations consider potentially malicious. The policies organizations select governing user-installed software may be organization-developed or provided by some external entity. Policy enforcement methods include procedural methods (e.g., periodic examination of user accounts), automated methods (e.g., configuration settings implemented on organizational information systems), or both.

Related Controls: *AC-3*, *CM-2*, *CM-3*, *CM-5*, *CM-6*, *CM-7*, *PL-4*

CM-11a.

Establishes [Assignment: organization-defined policies] governing the installation of software by users;

CM-11b.

Enforces software installation policies through [Assignment: organization-defined methods]; and

CM-11c.

Monitors policy compliance at [Assignment: organization-defined frequency].

CM-11 (1) : ALERTS FOR UNAUTHORIZED INSTALLATIONS

The information system alerts [Assignment: organization-defined personnel or roles] when the unauthorized installation of software is detected.

Related Controls: *CA-7*, *SI-4*

CM-11 (2) : PROHIBIT INSTALLATION WITHOUT PRIVILEGED STATUS

The information system prohibits user installation of software without explicit privileged status.

Note: Privileged status can be obtained, for example, by serving in the role of system administrator.

Related Controls: *AC-6*

Control Family: CONTINGENCY PLANNING

CP-1 : CONTINGENCY PLANNING POLICY AND PROCEDURES

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: This control addresses the establishment of policy and procedures for the effective implementation of selected security controls and control enhancements in the CP family. Policy and procedures reflect applicable federal laws, Executive Orders, directives, regulations, policies, standards, and guidance. Security program policies and procedures at the organization level may make the need for system-specific policies and procedures unnecessary. The policy can be included as part of the general information security policy for organizations or conversely, can be represented by multiple policies reflecting the complex nature of certain organizations. The procedures can be established for the security program in general and for particular information systems, if needed. The organizational risk management strategy is a key factor in establishing policy and procedures.

Related Controls: *PM-9*

CP-1a.

Develops, documents, and disseminates to [Assignment: organization-defined personnel or roles]:

CP-1a.1.

A contingency planning policy that addresses purpose, scope, roles, responsibilities, management commitment, coordination among organizational entities, and compliance; and

CP-1a.2.

Procedures to facilitate the implementation of the contingency planning policy and associated contingency planning controls; and

CP-1b.

Reviews and updates the current:

CP-1b.1.

Contingency planning policy [Assignment: organization-defined frequency]; and

CP-1b.2.

Contingency planning procedures [Assignment: organization-defined frequency].

Control Family: CONTINGENCY PLANNING

CP-2 : CONTINGENCY PLAN

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: Contingency planning for information systems is part of an overall organizational program for achieving continuity of operations for mission/business functions. Contingency planning addresses both information system restoration and implementation of alternative mission/business processes when systems are compromised. The effectiveness of contingency planning is maximized by considering such planning throughout the phases of the system development life cycle. Performing contingency planning on hardware, software, and firmware development can be an effective means of achieving information system resiliency. Contingency plans reflect the degree of restoration required for organizational information systems since not all systems may need to fully recover to achieve the level of continuity of operations desired. Information system recovery objectives reflect applicable laws, Executive Orders, directives, policies, standards, regulations, and guidelines. In addition to information system availability, contingency plans also address other security-related events resulting in a reduction in mission and/or business effectiveness, such as malicious attacks compromising the confidentiality or integrity of information systems. Actions addressed in contingency plans include, for example, orderly/graceful degradation, information system shutdown, fallback to a manual mode, alternate information flows, and operating in modes reserved for when systems are under attack. By closely coordinating contingency planning with incident handling activities, organizations can ensure that the necessary contingency planning activities are in place and activated in the event of a security incident.

Related Controls: *AC-14*, *CP-6*, *CP-7*, *CP-8*, *CP-9*, *CP-10*, *IR-4*, *IR-8*, *MP-2*, *MP-4*, *MP-5*, *PM-8*, *PM-11*

CP-2a.

Develops a contingency plan for the information system that:

CP-2a.1.

Identifies essential missions and business functions and associated contingency requirements;

CP-2a.2.

Provides recovery objectives, restoration priorities, and metrics;

CP-2a.3.

Addresses contingency roles, responsibilities, assigned individuals with contact information;

CP-2a.4.

Addresses maintaining essential missions and business functions despite an information system disruption, compromise, or failure;

CP-2a.5.

Addresses eventual, full information system restoration without deterioration of the security safeguards originally planned and implemented; and

CP-2a.6.

Is reviewed and approved by [Assignment: organization-defined personnel or roles];

CP-2b.

Distributes copies of the contingency plan to [Assignment: organization-defined key contingency personnel (identified by name and/or by role) and organizational elements];

CP-2c.

Coordinates contingency planning activities with incident handling activities;

CP-2d.

Reviews the contingency plan for the information system [Assignment: organization-defined frequency];

CP-2e.

Updates the contingency plan to address changes to the organization, information system, or environment of operation and problems encountered during contingency plan implementation, execution, or testing;

CP-2f.

Communicates contingency plan changes to [Assignment: organization-defined key contingency personnel (identified by name and/or by role) and organizational elements]; and

CP-2g.

Protects the contingency plan from unauthorized disclosure and modification.

CP-2 (1) : COORDINATE WITH RELATED PLANS

Baseline-Impact: *MODERATE*, **HIGH**

The organization coordinates contingency plan development with organizational elements responsible for related plans.

Note: Plans related to contingency plans for organizational information systems include, for example, Business Continuity Plans, Disaster Recovery Plans, Continuity of Operations Plans, Crisis Communications Plans, Critical Infrastructure Plans, Cyber Incident Response Plans, Insider Threat Implementation Plan, and Occupant Emergency Plans.

CP-2 (2) : CAPACITY PLANNING

Baseline-Impact: **HIGH**

The organization conducts capacity planning so that necessary capacity for information processing, telecommunications, and environmental support exists during contingency operations.

Note: Capacity planning is needed because different types of threats (e.g., natural disasters, targeted cyber attacks) can result in a reduction of the available processing, telecommunications, and support services originally intended to support the organizational missions/business functions. Organizations may need to anticipate degraded operations during contingency operations and factor such degradation into capacity planning.

CP-2 (3) : RESUME ESSENTIAL MISSIONS / BUSINESS FUNCTIONS

Baseline-Impact: *MODERATE*, **HIGH**

> The organization plans for the resumption of essential missions and business functions within [Assignment: organization-defined time period] of contingency plan activation.

Note: Organizations may choose to carry out the contingency planning activities in this control enhancement as part of organizational business continuity planning including, for example, as part of business impact analyses. The time period for resumption of essential missions/business functions may be dependent on the severity/extent of disruptions to the information system and its supporting infrastructure.

Related Controls: *PE-12*

CP-2 (4) : RESUME ALL MISSIONS / BUSINESS FUNCTIONS

Baseline-Impact: HIGH

> The organization plans for the resumption of all missions and business functions within [Assignment: organization-defined time period] of contingency plan activation.

Note: Organizations may choose to carry out the contingency planning activities in this control enhancement as part of organizational business continuity planning including, for example, as part of business impact analyses. The time period for resumption of all missions/business functions may be dependent on the severity/extent of disruptions to the information system and its supporting infrastructure.

Related Controls: *PE-12*

CP-2 (5) : CONTINUE ESSENTIAL MISSIONS / BUSINESS FUNCTIONS

Baseline-Impact: HIGH

> The organization plans for the continuance of essential missions and business functions with little or no loss of operational continuity and sustains that continuity until full information system restoration at primary processing and/or storage sites.

Note: Organizations may choose to carry out the contingency planning activities in this control enhancement as part of organizational business continuity planning including, for example, as part of business impact analyses. Primary processing and/or storage sites defined by organizations as part of contingency planning may change depending on the circumstances associated with the contingency (e.g., backup sites may become primary sites).

Related Controls: *PE-12*

CP-2 (6) : ALTERNATE PROCESSING / STORAGE SITE

> The organization plans for the transfer of essential missions and business functions to alternate processing and/or storage sites with little or no loss of operational continuity and sustains that continuity through information system restoration to primary processing and/or storage sites.

Note: Organizations may choose to carry out the contingency planning activities in this control enhancement as part of organizational business continuity planning including, for example, as part of business impact analyses. Primary processing and/or storage sites defined by organizations as part of contingency planning may change depending on the circumstances associated with the contingency (e.g., backup sites may become primary sites).

Related Controls: *PE-12*

CP-2 (7) : COORDINATE WITH EXTERNAL SERVICE PROVIDERS

> The organization coordinates its contingency plan with the contingency plans of external service providers to ensure that contingency requirements can be satisfied.

Note: When the capability of an organization to successfully carry out its core missions/business functions is dependent on external service providers, developing a timely and comprehensive contingency plan may become more challenging. In this situation, organizations coordinate contingency planning activities with the external entities to ensure that the individual plans reflect the overall contingency needs of the organization.

Related Controls: *SA-9*

CP-2 (8) : IDENTIFY CRITICAL ASSETS

Baseline-Impact: *MODERATE*, **HIGH**

> The organization identifies critical information system assets supporting essential missions and business functions.

Note: Organizations may choose to carry out the contingency planning activities in this control enhancement as part of organizational business continuity planning including, for example, as part of business impact analyses. Organizations identify critical information system assets so that additional safeguards and countermeasures can be employed (above and beyond those safeguards and countermeasures routinely implemented) to help ensure that organizational missions/business functions can continue to be conducted during contingency operations. In addition, the identification of critical information assets facilitates the prioritization of organizational resources. Critical information system assets include technical and operational aspects. Technical aspects include, for example, information technology services, information system components, information technology products, and mechanisms. Operational aspects include, for example, procedures (manually executed operations) and personnel (individuals operating technical safeguards and/or executing manual procedures). Organizational program protection plans can provide assistance in identifying critical assets.

Related Controls: *SA-14*, *SA-15*

Control Family: CONTINGENCY PLANNING

CP-3 : CONTINGENCY TRAINING

Priority: P2

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The organization provides contingency training to information system users consistent with assigned roles and responsibilities:

Note: Contingency training provided by organizations is linked to the assigned roles and responsibilities of organizational personnel to ensure that the appropriate content and level of detail is included in such training. For example, regular users may only need to know when and where to report for duty during contingency operations and if normal duties are affected; system administrators may require additional training on how to set up information systems at alternate processing and storage sites; and managers/senior leaders may receive more specific training on how to conduct mission-essential functions in designated off-site locations and how to establish communications with other governmental entities for purposes of coordination on contingency-related activities. Training for contingency roles/responsibilities reflects the specific continuity requirements in the contingency plan.

Related Controls: *AT-2*, *AT-3*, *CP-2*, *IR-2*

CP-3a.

> Within [Assignment: organization-defined time period] of assuming a contingency role or responsibility;

CP-3b.

> When required by information system changes; and

CP-3c.

> [Assignment: organization-defined frequency] thereafter.

CP-3 (1) : SIMULATED EVENTS

Baseline-Impact: **HIGH**

> The organization incorporates simulated events into contingency training to facilitate effective response by personnel in crisis situations.

CP-3 (2) : AUTOMATED TRAINING ENVIRONMENTS

The organization employs automated mechanisms to provide a more thorough and realistic contingency training environment.

Control Family: CONTINGENCY PLANNING

CP-4 : CONTINGENCY PLAN TESTING

Priority: P2

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: Methods for testing contingency plans to determine the effectiveness of the plans and to identify potential weaknesses in the plans include, for example, walk-through and tabletop exercises, checklists, simulations (parallel, full interrupt), and comprehensive exercises. Organizations conduct testing based on the continuity requirements in contingency plans and include a determination of the effects on organizational operations, assets, and individuals arising due to contingency operations. Organizations have flexibility and discretion in the breadth, depth, and timelines of corrective actions.

Related Controls: *CP-2*, *CP-3*, *IR-3*

CP-4a.

Tests the contingency plan for the information system [Assignment: organization-defined frequency] using [Assignment: organization-defined tests] to determine the effectiveness of the plan and the organizational readiness to execute the plan;

CP-4b.

Reviews the contingency plan test results; and

CP-4c.

Initiates corrective actions, if needed.

CP-4 (1) : COORDINATE WITH RELATED PLANS

Baseline-Impact: *MODERATE*, **HIGH**

The organization coordinates contingency plan testing with organizational elements responsible for related plans.

Note: Plans related to contingency plans for organizational information systems include, for example, Business Continuity Plans, Disaster Recovery Plans, Continuity of Operations Plans, Crisis Communications Plans, Critical Infrastructure Plans, Cyber Incident Response Plans, and Occupant Emergency Plans. This control enhancement does not require organizations to create organizational elements to handle related plans or to align such elements with specific plans. It does require, however, that if such organizational elements are responsible for related plans, organizations should coordinate with those elements.

Related Controls: *IR-8*, *PM-8*

CP-4 (2) : ALTERNATE PROCESSING SITE

Baseline-Impact: HIGH

The organization tests the contingency plan at the alternate processing site:

Related Controls: *CP-7*

CP-4 (2)(a)

To familiarize contingency personnel with the facility and available resources; and

CP-4 (2)(b)

To evaluate the capabilities of the alternate processing site to support contingency operations.

CP-4 (3) : AUTOMATED TESTING

The organization employs automated mechanisms to more thoroughly and effectively test the contingency plan.

Note: Automated mechanisms provide more thorough and effective testing of contingency plans, for example: (i) by providing more complete coverage of contingency issues; (ii) by selecting more realistic test scenarios and environments; and (iii) by effectively stressing the information system and supported missions.

CP-4 (4) : FULL RECOVERY / RECONSTITUTION

The organization includes a full recovery and reconstitution of the information system to a known state as part of contingency plan testing.

Related Controls: *CP-10*, *SC-24*

Control Family: CONTINGENCY PLANNING

CP-5 : CONTINGENCY PLAN UPDATE

[Withdrawn: Incorporated into CP-2].

Control Family: CONTINGENCY PLANNING

CP-6 : ALTERNATE STORAGE SITE

Priority: P1

Baseline-Impact: *MODERATE*, **HIGH**

The organization:

Note: Alternate storage sites are sites that are geographically distinct from primary storage sites. An alternate storage site maintains duplicate copies of information and data in the event that the primary storage site is not available. Items covered by alternate storage site agreements include, for example, environmental conditions at alternate sites, access rules, physical and environmental protection requirements, and coordination of delivery/retrieval of backup media. Alternate storage sites reflect the requirements in contingency plans so that organizations can maintain essential missions/business functions despite disruption, compromise, or failure in organizational information systems.

Related Controls: *CP-2*, *CP-7*, *CP-9*, *CP-10*, *MP-4*

CP-6a.

Establishes an alternate storage site including necessary agreements to permit the storage and retrieval of information system backup information; and

CP-6b.

Ensures that the alternate storage site provides information security safeguards equivalent to that of the primary site.

CP-6 (1) : SEPARATION FROM PRIMARY SITE

Baseline-Impact: *MODERATE*, **HIGH**

The organization identifies an alternate storage site that is separated from the primary storage site to reduce susceptibility to the same threats.

Note: Threats that affect alternate storage sites are typically defined in organizational assessments of risk and include, for example, natural disasters, structural failures, hostile cyber attacks, and errors of omission/commission. Organizations determine what is considered a sufficient degree of separation between primary and alternate storage sites based on the types of threats that are of concern. For one particular type of threat (i.e., hostile cyber attack), the degree of separation between sites is less relevant.

Related Controls: *RA-3*

CP-6 (2) : RECOVERY TIME / POINT OBJECTIVES

Baseline-Impact: **HIGH**

The organization configures the alternate storage site to facilitate recovery operations in accordance with recovery time and recovery point objectives.

CP-6 (3) : ACCESSIBILITY

Baseline-Impact: *MODERATE*, **HIGH**

The organization identifies potential accessibility problems to the alternate storage site in the event of an area-wide disruption or disaster and outlines explicit mitigation actions.

Note: Area-wide disruptions refer to those types of disruptions that are broad in geographic scope (e.g., hurricane, regional power outage) with such determinations made by organizations based on organizational assessments of risk. Explicit mitigation actions include, for example: (i) duplicating backup information at other alternate storage sites if access problems occur at originally designated alternate sites; or (ii) planning for physical access to retrieve backup information if electronic accessibility to the alternate site is disrupted.

Related Controls: *RA-3*

Control Family: CONTINGENCY PLANNING

CP-7 : ALTERNATE PROCESSING SITE

Priority: P1

Baseline-Impact: *MODERATE*, **HIGH**

The organization:

Note: Alternate processing sites are sites that are geographically distinct from primary processing sites. An alternate processing site provides processing capability in the event that the primary processing site is not available. Items covered by alternate processing site agreements include, for example, environmental conditions at alternate sites, access rules, physical and environmental protection requirements, and coordination for the transfer/assignment of personnel. Requirements are specifically allocated to alternate processing sites that reflect the requirements in contingency plans to maintain essential missions/business functions despite disruption, compromise, or failure in organizational information systems.

Related Controls: *CP-2*, *CP-6*, *CP-8*, *CP-9*, *CP-10*, *MA-6*

CP-7a.

Establishes an alternate processing site including necessary agreements to permit the transfer and resumption of [Assignment: organization-defined information system operations] for essential missions/business functions within [Assignment: organization-defined time period consistent with recovery time and recovery point objectives] when the primary processing capabilities are unavailable;

CP-7b.

Ensures that equipment and supplies required to transfer and resume operations are available at the alternate processing site or contracts are in place to support delivery to the site within the organization-defined time period for transfer/resumption; and

CP-7c.

Ensures that the alternate processing site provides information security safeguards equivalent to those of the primary site.

CP-7 (1) : SEPARATION FROM PRIMARY SITE

Baseline-Impact: *MODERATE*, **HIGH**

The organization identifies an alternate processing site that is separated from the primary processing site to reduce susceptibility to the same threats.

Note: Threats that affect alternate processing sites are typically defined in organizational assessments of risk and include, for example, natural disasters, structural failures, hostile cyber attacks, and errors of omission/commission. Organizations determine what is considered a sufficient degree of separation between primary and alternate processing sites based on the types of threats that are of concern. For one particular type of threat (i.e., hostile cyber attack), the degree of separation between sites is less relevant.

Related Controls: *RA-3*

CP-7 (2) : ACCESSIBILITY

Baseline-Impact: *MODERATE*, **HIGH**

The organization identifies potential accessibility problems to the alternate processing site in the event of an area-wide disruption or disaster and outlines explicit mitigation actions.

Note: Area-wide disruptions refer to those types of disruptions that are broad in geographic scope (e.g., hurricane, regional power outage) with such determinations made by organizations based on organizational assessments of risk.

Related Controls: *RA-3*

CP-7 (3) : PRIORITY OF SERVICE

Baseline-Impact: *MODERATE*, **HIGH**

> The organization develops alternate processing site agreements that contain priority-of-service provisions in accordance with organizational availability requirements (including recovery time objectives).

Note: Priority-of-service agreements refer to negotiated agreements with service providers that ensure that organizations receive priority treatment consistent with their availability requirements and the availability of information resources at the alternate processing site.

CP-7 (4) : PREPARATION FOR USE

Baseline-Impact: **HIGH**

> The organization prepares the alternate processing site so that the site is ready to be used as the operational site supporting essential missions and business functions.

Note: Site preparation includes, for example, establishing configuration settings for information system components at the alternate processing site consistent with the requirements for such settings at the primary site and ensuring that essential supplies and other logistical considerations are in place.

Related Controls: *CM-2*, *CM-6*

CP-7 (5) : EQUIVALENT INFORMATION SECURITY SAFEGUARDS

> [Withdrawn: Incorporated into CP-7].

CP-7 (6) : INABILITY TO RETURN TO PRIMARY SITE

> The organization plans and prepares for circumstances that preclude returning to the primary processing site.

Control Family: CONTINGENCY PLANNING

CP-8 : TELECOMMUNICATIONS SERVICES

Priority: P1

Baseline-Impact: *MODERATE*, **HIGH**

> The organization establishes alternate telecommunications services including necessary agreements to permit the resumption of [Assignment: organization-defined information system operations] for essential missions and business functions within [Assignment: organization-defined time period] when the primary telecommunications capabilities are unavailable at either the primary or alternate processing or storage sites.

Note: This control applies to telecommunications services (data and voice) for primary and alternate processing and storage sites. Alternate telecommunications services reflect the continuity requirements in contingency plans to maintain essential missions/business functions despite the loss of primary telecommunications services. Organizations may specify different time periods for primary/alternate sites. Alternate telecommunications services include, for example, additional organizational or commercial ground-based circuits/lines or satellites in lieu of ground-based communications. Organizations consider factors such as availability, quality of service, and access when entering into alternate telecommunications agreements.

Related Controls: *CP-2*, *CP-6*, *CP-7*

CP-8 (1) : PRIORITY OF SERVICE PROVISIONS

Baseline-Impact: *MODERATE*, **HIGH**

The organization:

Note: Organizations consider the potential mission/business impact in situations where telecommunications service providers are servicing other organizations with similar priority-of-service provisions.

CP-8 (1)(a)

Develops primary and alternate telecommunications service agreements that contain priority-of-service provisions in accordance with organizational availability requirements (including recovery time objectives); and

CP-8 (1)(b)

Requests Telecommunications Service Priority for all telecommunications services used for national security emergency preparedness in the event that the primary and/or alternate telecommunications services are provided by a common carrier.

CP-8 (2) : SINGLE POINTS OF FAILURE

Baseline-Impact: *MODERATE*, **HIGH**

The organization obtains alternate telecommunications services to reduce the likelihood of sharing a single point of failure with primary telecommunications services.

CP-8 (3) : SEPARATION OF PRIMARY / ALTERNATE PROVIDERS

Baseline-Impact: HIGH

The organization obtains alternate telecommunications services from providers that are separated from primary service providers to reduce susceptibility to the same threats.

Note: Threats that affect telecommunications services are typically defined in organizational assessments of risk and include, for example, natural disasters, structural failures, hostile cyber/physical attacks, and errors of omission/commission. Organizations seek to reduce common susceptibilities by, for example, minimizing shared infrastructure among telecommunications service providers and achieving sufficient geographic separation between services. Organizations may consider using a single service provider in situations where the service provider can provide alternate telecommunications services meeting the separation needs addressed in the risk assessment.

CP-8 (4) : PROVIDER CONTINGENCY PLAN

Baseline-Impact: HIGH

The organization:

Note: Reviews of provider contingency plans consider the proprietary nature of such plans. In some situations, a summary of provider contingency plans may be sufficient evidence for organizations to satisfy the review requirement. Telecommunications service providers may also participate in ongoing disaster recovery exercises in coordination with the Department of Homeland Security, state, and local governments. Organizations may use these types of activities to satisfy evidentiary requirements related to service provider contingency plan reviews, testing, and training.

CP-8 (4)(a)

Requires primary and alternate telecommunications service providers to have contingency plans;

CP-8 (4)(b)

Reviews provider contingency plans to ensure that the plans meet organizational contingency requirements; and

CP-8 (4)(c)

Obtains evidence of contingency testing/training by providers [Assignment: organization-defined frequency].

CP-8 (5) : ALTERNATE TELECOMMUNICATION SERVICE TESTING

The organization tests alternate telecommunication services [Assignment: organization-defined frequency].

Control Family: CONTINGENCY PLANNING

CP-9 : INFORMATION SYSTEM BACKUP

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: System-level information includes, for example, system-state information, operating system and application software, and licenses. User-level information includes any information other than system-level information. Mechanisms employed by organizations to protect the integrity of information system backups include, for example, digital signatures and cryptographic hashes. Protection of system backup information while in transit is beyond the scope of this control. Information system backups reflect the requirements in contingency plans as well as other organizational requirements for backing up information.

Related Controls: *CP-2*, *CP-6*, *MP-4*, *MP-5*, *SC-13*

CP-9a.

Conducts backups of user-level information contained in the information system [Assignment: organization-defined frequency consistent with recovery time and recovery point objectives];

CP-9b.

Conducts backups of system-level information contained in the information system [Assignment: organization-defined frequency consistent with recovery time and recovery point objectives];

CP-9c.

Conducts backups of information system documentation including security-related documentation [Assignment: organization-defined frequency consistent with recovery time and recovery point objectives]; and

CP-9d.

Protects the confidentiality, integrity, and availability of backup information at storage locations.

CP-9 (1) : TESTING FOR RELIABILITY / INTEGRITY

Baseline-Impact: *MODERATE*, **HIGH**

The organization tests backup information [Assignment: organization-defined frequency] to verify media reliability and information integrity.

Related Controls: *CP-4*

CP-9 (2) : TEST RESTORATION USING SAMPLING

Baseline-Impact: **HIGH**

The organization uses a sample of backup information in the restoration of selected information system functions as part of contingency plan testing.

Related Controls: *CP-4*

CP-9 (3) : SEPARATE STORAGE FOR CRITICAL INFORMATION

Baseline-Impact: **HIGH**

The organization stores backup copies of [Assignment: organization-defined critical information system software and other security-related information] in a separate facility or in a fire-rated container that is not collocated with the operational system.

Note: Critical information system software includes, for example, operating systems, cryptographic key management systems, and intrusion detection/prevention systems. Security-related information includes, for example, organizational inventories of hardware, software, and firmware components. Alternate storage sites typically serve as separate storage facilities for organizations.

Related Controls: *CM-2*, *CM-8*

CP-9 (4) : PROTECTION FROM UNAUTHORIZED MODIFICATION

[Withdrawn: Incorporated into CP-9].

CP-9 (5) : TRANSFER TO ALTERNATE STORAGE SITE

Baseline-Impact: **HIGH**

> The organization transfers information system backup information to the alternate storage site [Assignment: organization-defined time period and transfer rate consistent with the recovery time and recovery point objectives].

Note: Information system backup information can be transferred to alternate storage sites either electronically or by physical shipment of storage media.

CP-9 (6) : REDUNDANT SECONDARY SYSTEM

> The organization accomplishes information system backup by maintaining a redundant secondary system that is not collocated with the primary system and that can be activated without loss of information or disruption to operations.

Related Controls: *CP-7*, *CP-10*

CP-9 (7) : DUAL AUTHORIZATION

> The organization enforces dual authorization for the deletion or destruction of [Assignment: organization-defined backup information].

Note: Dual authorization ensures that the deletion or destruction of backup information cannot occur unless two qualified individuals carry out the task. Individuals deleting/destroying backup information possess sufficient skills/expertise to determine if the proposed deletion/destruction of backup information reflects organizational policies and procedures. Dual authorization may also be known as two-person control.

Related Controls: *AC-3*, *MP-2*

Control Family: CONTINGENCY PLANNING

CP-10 : INFORMATION SYSTEM RECOVERY AND RECONSTITUTION

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The organization provides for the recovery and reconstitution of the information system to a known state after a disruption, compromise, or failure.

Note: Recovery is executing information system contingency plan activities to restore organizational missions/business functions. Reconstitution takes place following recovery and includes activities for returning organizational information systems to fully operational states. Recovery and reconstitution operations reflect mission and business priorities, recovery point/time and reconstitution objectives, and established organizational metrics consistent with contingency plan requirements. Reconstitution includes the deactivation of any interim information system capabilities that may have been needed during recovery operations. Reconstitution also includes assessments of fully restored information system capabilities, reestablishment of continuous monitoring activities, potential information system reauthorizations, and activities to prepare the systems against future disruptions, compromises, or failures. Recovery/reconstitution capabilities employed by organizations can include both automated mechanisms and manual procedures.

Related Controls: *CA-2*, *CA-6*, *CA-7*, *CP-2*, *CP-6*, *CP-7*, *CP-9*, *SC-24*

CP-10 (1) : CONTINGENCY PLAN TESTING

[Withdrawn: Incorporated into CP-4].

CP-10 (2) : TRANSACTION RECOVERY

Baseline-Impact: *MODERATE*, **HIGH**

The information system implements transaction recovery for systems that are transaction-based.

Note: Transaction-based information systems include, for example, database management systems and transaction processing systems. Mechanisms supporting transaction recovery include, for example, transaction rollback and transaction journaling.

CP-10 (3) : COMPENSATING SECURITY CONTROLS

[Withdrawn: Addressed through tailoring procedures].

CP-10 (4) : RESTORE WITHIN TIME PERIOD

Baseline-Impact: **HIGH**

The organization provides the capability to restore information system components within [Assignment: organization-defined restoration time-periods] from configuration-controlled and integrity-protected information representing a known, operational state for the components.

Note: Restoration of information system components includes, for example, reimaging which restores components to known, operational states.

Related Controls: *CM-2*

CP-10 (5) : FAILOVER CAPABILITY

[Withdrawn: Incorporated into SI-13].

CP-10 (6) : COMPONENT PROTECTION

The organization protects backup and restoration hardware, firmware, and software.

Note: Protection of backup and restoration hardware, firmware, and software components includes both physical and technical safeguards. Backup and restoration software includes, for example, router tables, compilers, and other security-relevant system software.

Related Controls: *AC-3*, *AC-6*, *PE-3*

Control Family: CONTINGENCY PLANNING

CP-11 : ALTERNATE COMMUNICATIONS PROTOCOLS

Priority: P0

The information system provides the capability to employ [Assignment: organization-defined alternative communications protocols] in support of maintaining continuity of operations.

Note: Contingency plans and the associated training and testing for those plans, incorporate an alternate communications protocol capability as part of increasing the resilience of organizational information systems. Alternate communications protocols include, for example, switching from Transmission Control Protocol/Internet Protocol (TCP/IP) Version 4 to TCP/IP Version 6. Switching communications protocols may affect software applications and therefore, the potential side effects of introducing alternate communications protocols are analyzed prior to implementation.

Control Family: CONTINGENCY PLANNING

CP-12 : SAFE MODE

Priority: P0

The information system, when [Assignment: organization-defined conditions] are detected, enters a safe mode of operation with [Assignment: organization-defined restrictions of safe mode of operation].

Note: For information systems supporting critical missions/business functions including, for example, military operations and weapons systems, civilian space operations, nuclear power plant operations, and air traffic control operations (especially real-time operational environments), organizations may choose to identify certain conditions under which those systems revert to a predefined safe mode of operation. The safe mode of operation, which can be activated automatically or manually, restricts the types of activities or operations information systems could execute when those conditions are encountered. Restriction includes, for example, allowing only certain functions that could be carried out under limited power or with reduced communications bandwidth.

Control Family: CONTINGENCY PLANNING

CP-13 : ALTERNATIVE SECURITY MECHANISMS

Priority: P0

> The organization employs [Assignment: organization-defined alternative or supplemental security mechanisms] for satisfying [Assignment: organization-defined security functions] when the primary means of implementing the security function is unavailable or compromised.

Note: This control supports information system resiliency and contingency planning/continuity of operations. To ensure mission/business continuity, organizations can implement alternative or supplemental security mechanisms. These mechanisms may be less effective than the primary mechanisms (e.g., not as easy to use, not as scalable, or not as secure). However, having the capability to readily employ these alternative/supplemental mechanisms enhances overall mission/business continuity that might otherwise be adversely impacted if organizational operations had to be curtailed until the primary means of implementing the functions was restored. Given the cost and level of effort required to provide such alternative capabilities, this control would typically be applied only to critical security capabilities provided by information systems, system components, or information system services. For example, an organization may issue to senior executives and system administrators one-time pads in case multifactor tokens, the organizations standard means for secure remote authentication, is compromised.

Related Controls: *CP-2*

Control Family: IDENTIFICATION AND AUTHENTICATION

IA-1 : IDENTIFICATION AND AUTHENTICATION POLICY AND PROCEDURES

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The organization:

Note: This control addresses the establishment of policy and procedures for the effective implementation of selected security controls and control enhancements in the IA family. Policy and procedures reflect applicable federal laws, Executive Orders, directives, regulations, policies, standards, and guidance. Security program policies and procedures at the organization level may make the need for system-specific policies and procedures unnecessary. The policy can be included as part of the general information security policy for organizations or conversely, can be represented by multiple policies reflecting the complex nature of certain organizations. The procedures can be established for the security program in general and for particular information systems, if needed. The organizational risk management strategy is a key factor in establishing policy and procedures.

Related Controls: *PM-9*

IA-1a.

Develops, documents, and disseminates to [Assignment: organization-defined personnel or roles]:

IA-1a.1.

An identification and authentication policy that addresses purpose, scope, roles, responsibilities, management commitment, coordination among organizational entities, and compliance; and

IA-1a.2.

Procedures to facilitate the implementation of the identification and authentication policy and associated identification and authentication controls; and

IA-1b.

Reviews and updates the current:

IA-1b.1.

Identification and authentication policy [Assignment: organization-defined frequency]; and

IA-1b.2.

Identification and authentication procedures [Assignment: organization-defined frequency].

Control Family: IDENTIFICATION AND AUTHENTICATION

IA-2 : IDENTIFICATION AND AUTHENTICATION (ORGANIZATIONAL USERS)

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The information system uniquely identifies and authenticates organizational users (or processes acting on behalf of organizational users).

Note:

Organizational users include employees or individuals that organizations deem to have equivalent status of employees (e.g., contractors, guest researchers). This control applies to all accesses other than: (i) accesses that are explicitly identified and documented in AC-14; and (ii) accesses that occur through authorized use of group authenticators without individual authentication. Organizations may require unique identification of individuals in group accounts (e.g., shared privilege accounts) or for detailed accountability of individual activity. Organizations employ passwords, tokens, or biometrics to authenticate user identities, or in the case multifactor authentication, or some combination thereof. Access to organizational information systems is defined as either local access or network access. Local access is any access to organizational information systems by users (or processes acting on behalf of users) where such access is obtained by direct connections without the use of networks. Network access is access to organizational information systems by users (or processes acting on behalf of users) where such access is obtained through network connections (i.e., nonlocal accesses). Remote access is a type of network access that involves communication through external networks (e.g., the Internet). Internal networks include local area networks and wide area networks. In addition, the use of encrypted virtual private networks (VPNs) for network connections between organization-controlled endpoints and non-organization controlled endpoints may be treated as internal networks from the perspective of protecting the confidentiality and integrity of information traversing the network.

Organizations can satisfy the identification and authentication requirements in this control by complying with the requirements in Homeland Security Presidential Directive 12 consistent with the specific organizational implementation plans. Multifactor authentication requires the use of two or more different factors to achieve authentication. The factors are defined as: (i) something you know (e.g., password, personal identification number [PIN]); (ii) something you have (e.g., cryptographic identification device, token); or (iii) something you are (e.g., biometric). Multifactor solutions that require devices separate from information systems gaining access include, for example, hardware tokens providing time-based or challenge-response authenticators and smart cards such as the U.S. Government Personal Identity Verification card and the DoD common access card. In addition to identifying and authenticating users at the information system level (i.e., at logon), organizations also employ identification and authentication mechanisms at the application level, when necessary, to provide increased information security. Identification and authentication requirements for other than organizational users are described in IA-8.

Related Controls: *AC-2*, *AC-3*, *AC-14*, *AC-17*, *AC-18*, *IA-4*, *IA-5*, *IA-8*

IA-2 (1) : NETWORK ACCESS TO PRIVILEGED ACCOUNTS

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The information system implements multifactor authentication for network access to privileged accounts.

Related Controls: *AC-6*

IA-2 (2) : NETWORK ACCESS TO NON-PRIVILEGED ACCOUNTS

Baseline-Impact: *MODERATE*, **HIGH**

The information system implements multifactor authentication for network access to non-privileged accounts.

IA-2 (3) : LOCAL ACCESS TO PRIVILEGED ACCOUNTS

Baseline-Impact: *MODERATE*, **HIGH**

The information system implements multifactor authentication for local access to privileged accounts.

Related Controls: *AC-6*

IA-2 (4) : LOCAL ACCESS TO NON-PRIVILEGED ACCOUNTS

Baseline-Impact: **HIGH**

The information system implements multifactor authentication for local access to non-privileged accounts.

IA-2 (5) : GROUP AUTHENTICATION

The organization requires individuals to be authenticated with an individual authenticator when a group authenticator is employed.

Note: Requiring individuals to use individual authenticators as a second level of authentication helps organizations to mitigate the risk of using group authenticators.

IA-2 (6) : NETWORK ACCESS TO PRIVILEGED ACCOUNTS - SEPARATE DEVICE

The information system implements multifactor authentication for network access to privileged accounts such that one of the factors is provided by a device separate from the system gaining access and the device meets [Assignment: organization-defined strength of mechanism requirements].

Related Controls: *AC-6*

IA-2 (7) : NETWORK ACCESS TO NON-PRIVILEGED ACCOUNTS - SEPARATE DEVICE

The information system implements multifactor authentication for network access to non-privileged accounts such that one of the factors is provided by a device separate from the system gaining access and the device meets [Assignment: organization-defined strength of mechanism requirements].

IA-2 (8) : NETWORK ACCESS TO PRIVILEGED ACCOUNTS - REPLAY RESISTANT

Baseline-Impact: *MODERATE*, **HIGH**

The information system implements replay-resistant authentication mechanisms for network access to privileged accounts.

Note: Authentication processes resist replay attacks if it is impractical to achieve successful authentications by replaying previous authentication messages. Replay-resistant techniques include, for example, protocols that use nonces or challenges such as Transport Layer Security (TLS) and time synchronous or challenge-response one-time authenticators.

IA-2 (9) : NETWORK ACCESS TO NON-PRIVILEGED ACCOUNTS - REPLAY RESISTANT

Baseline-Impact: **HIGH**

> The information system implements replay-resistant authentication mechanisms for network access to non-privileged accounts.

Note: Authentication processes resist replay attacks if it is impractical to achieve successful authentications by recording/replaying previous authentication messages. Replay-resistant techniques include, for example, protocols that use nonces or challenges such as Transport Layer Security (TLS) and time synchronous or challenge-response one-time authenticators.

IA-2 (10) : SINGLE SIGN-ON

> The information system provides a single sign-on capability for [Assignment: organization-defined information system accounts and services].

Note: Single sign-on enables users to log in once and gain access to multiple information system resources. Organizations consider the operational efficiencies provided by single sign-on capabilities with the increased risk from disclosures of single authenticators providing access to multiple system resources.

IA-2 (11) : REMOTE ACCESS - SEPARATE DEVICE

Baseline-Impact: *MODERATE*, **HIGH**

> The information system implements multifactor authentication for remote access to privileged and non-privileged accounts such that one of the factors is provided by a device separate from the system gaining access and the device meets [Assignment: organization-defined strength of mechanism requirements].

Note: For remote access to privileged/non-privileged accounts, the purpose of requiring a device that is separate from the information system gaining access for one of the factors during multifactor authentication is to reduce the likelihood of compromising authentication credentials stored on the system. For example, adversaries deploying malicious code on organizational information systems can potentially compromise such credentials resident on the system and subsequently impersonate authorized users.

Related Controls: *AC-6*

IA-2 (12) : ACCEPTANCE OF PIV CREDENTIALS

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The information system accepts and electronically verifies Personal Identity Verification (PIV) credentials.

Note: This control enhancement applies to organizations implementing logical access control systems (LACS) and physical access control systems (PACS). Personal Identity Verification (PIV) credentials are those credentials issued by federal agencies that conform to FIPS Publication 201 and supporting guidance documents. OMB Memorandum 11-11 requires federal agencies to continue implementing the requirements specified in HSPD-12 to enable agency-wide use of PIV credentials.

Related Controls: *AU-2*, *PE-3*, *SA-4*

IA-2 (13) : OUT-OF-BAND AUTHENTICATION

> The information system implements [Assignment: organization-defined out-of-band authentication] under [Assignment: organization-defined conditions].

Note: Out-of-band authentication (OOBA) refers to the use of two separate communication paths to identify and authenticate users or devices to an information system. The first path (i.e., the in-band path), is used to identify and authenticate users or devices, and generally is the path through which information flows. The second path (i.e., the out-of-band path) is used to independently verify the authentication and/or requested action. For example, a user authenticates via a notebook computer to a remote server to which the user desires access, and requests some action of the server via that communication path. Subsequently, the server contacts the user via the users cell phone to verify that the requested action originated from the user. The user may either confirm the intended action to an individual on the telephone or provide an authentication code via the telephone. This type of authentication can be employed by organizations to mitigate actual or suspected man-in the-middle attacks. The conditions for activation can include, for example, suspicious activities, new threat indicators or elevated threat levels, or the impact level or classification level of information in requested transactions.

Related Controls: *IA-10*, *IA-11*, *SC-37*

Control Family: IDENTIFICATION AND AUTHENTICATION

IA-3 : DEVICE IDENTIFICATION AND AUTHENTICATION

Priority: P1

Baseline-Impact: *MODERATE*, **HIGH**

> The information system uniquely identifies and authenticates [Assignment: organization-defined specific and/or types of devices] before establishing a [Selection (one or more): local; remote; network] connection.

Note: Organizational devices requiring unique device-to-device identification and authentication may be defined by type, by device, or by a combination of type/device. Information systems typically use either shared known information (e.g., Media Access Control [MAC] or Transmission Control Protocol/Internet Protocol [TCP/IP] addresses) for device identification or organizational authentication solutions (e.g., IEEE 802.1x and Extensible Authentication Protocol [EAP], Radius server with EAP-Transport Layer Security [TLS] authentication, Kerberos) to identify/authenticate devices on local and/or wide area networks. Organizations determine the required strength of authentication mechanisms by the security categories of information systems. Because of the challenges of applying this control on large scale, organizations are encouraged to only apply the control to those limited number (and type) of devices that truly need to support this capability.

Related Controls: *AC-17*, *AC-18*, *AC-19*, *CA-3*, *IA-4*, *IA-5*

IA-3 (1) : CRYPTOGRAPHIC BIDIRECTIONAL AUTHENTICATION

The information system authenticates [Assignment: organization-defined specific devices and/or types of devices] before establishing [Selection (one or more): local; remote; network] connection using bidirectional authentication that is cryptographically based.

Note: A local connection is any connection with a device communicating without the use of a network. A network connection is any connection with a device that communicates through a network (e.g., local area or wide area network, Internet). A remote connection is any connection with a device communicating through an external network (e.g., the Internet). Bidirectional authentication provides stronger safeguards to validate the identity of other devices for connections that are of greater risk (e.g., remote connections).

Related Controls: *SC-8*, *SC-12*, *SC-13*

IA-3 (2) : CRYPTOGRAPHIC BIDIRECTIONAL NETWORK AUTHENTICATION

[Withdrawn: Incorporated into IA-3 (1)].

IA-3 (3) : DYNAMIC ADDRESS ALLOCATION

The organization:

Note: DHCP-enabled clients obtaining leases for IP addresses from DHCP servers, is a typical example of dynamic address allocation for devices.

Related Controls: *AU-2*, *AU-3*, *AU-6*, *AU-12*

IA-3 (3)(a)

Standardizes dynamic address allocation lease information and the lease duration assigned to devices in accordance with [Assignment: organization-defined lease information and lease duration]; and

IA-3 (3)(b)

Audits lease information when assigned to a device.

IA-3 (4) : DEVICE ATTESTATION

The organization ensures that device identification and authentication based on attestation is handled by [Assignment: organization-defined configuration management process].

Note: Device attestation refers to the identification and authentication of a device based on its configuration and known operating state. This might be determined via some cryptographic hash of the device. If device attestation is the means of identification and authentication, then it is important that patches and updates to the device are handled via a configuration management process such that the those patches/updates are done securely and at the same time do not disrupt the identification and authentication to other devices.

Control Family: IDENTIFICATION AND AUTHENTICATION

IA-4 : IDENTIFIER MANAGEMENT

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization manages information system identifiers by:

Note: Common device identifiers include, for example, media access control (MAC), Internet protocol (IP) addresses, or device-unique token identifiers. Management of individual identifiers is not applicable to shared information system accounts (e.g., guest and anonymous accounts). Typically, individual identifiers are the user names of the information system accounts assigned to those individuals. In such instances, the account management activities of AC-2 use account names provided by IA-4. This control also addresses individual identifiers not necessarily associated with information system accounts (e.g., identifiers used in physical security control databases accessed by badge reader systems for access to information systems). Preventing reuse of identifiers implies preventing the assignment of previously used individual, group, role, or device identifiers to different individuals, groups, roles, or devices.

Related Controls: *AC-2*, *IA-2*, *IA-3*, *IA-5*, *IA-8*, *SC-37*

IA-4a.

Receiving authorization from [Assignment: organization-defined personnel or roles] to assign an individual, group, role, or device identifier;

IA-4b.

Selecting an identifier that identifies an individual, group, role, or device;

IA-4c.

Assigning the identifier to the intended individual, group, role, or device;

IA-4d.

Preventing reuse of identifiers for [Assignment: organization-defined time period]; and

IA-4e.

Disabling the identifier after [Assignment: organization-defined time period of inactivity].

IA-4 (1) : PROHIBIT ACCOUNT IDENTIFIERS AS PUBLIC IDENTIFIERS

The organization prohibits the use of information system account identifiers that are the same as public identifiers for individual electronic mail accounts.

Note: Prohibiting the use of information systems account identifiers that are the same as some public identifier such as the individual identifier section of an electronic mail address, makes it more difficult for adversaries to guess user identifiers on organizational information systems.

Related Controls: *AT-2*

IA-4 (2) : SUPERVISOR AUTHORIZATION

The organization requires that the registration process to receive an individual identifier includes supervisor authorization.

IA-4 (3) : MULTIPLE FORMS OF CERTIFICATION

The organization requires multiple forms of certification of individual identification be presented to the registration authority.

Note: Requiring multiple forms of identification, such as documentary evidence or a combination of documents and biometrics, reduces the likelihood of individuals using fraudulent identification to establish an identity, or at least increases the work factor of potential adversaries.

IA-4 (4) : IDENTIFY USER STATUS

The organization manages individual identifiers by uniquely identifying each individual as [Assignment: organization-defined characteristic identifying individual status].

Note: Characteristics identifying the status of individuals include, for example, contractors and foreign nationals. Identifying the status of individuals by specific characteristics provides additional information about the people with whom organizational personnel are communicating. For example, it might be useful for a government employee to know that one of the individuals on an email message is a contractor.

Related Controls: *AT-2*

IA-4 (5) : DYNAMIC MANAGEMENT

The information system dynamically manages identifiers.

Note: In contrast to conventional approaches to identification which presume static accounts for preregistered users, many distributed information systems including, for example, service-oriented architectures, rely on establishing identifiers at run time for entities that were previously unknown. In these situations, organizations anticipate and provision for the dynamic establishment of identifiers. Preestablished trust relationships and mechanisms with appropriate authorities to validate identities and related credentials are essential.

Related Controls: *AC-16*

IA-4 (6) : CROSS-ORGANIZATION MANAGEMENT

The organization coordinates with [Assignment: organization-defined external organizations] for cross-organization management of identifiers.

Note: Cross-organization identifier management provides the capability for organizations to appropriately identify individuals, groups, roles, or devices when conducting cross-organization activities involving the processing, storage, or transmission of information.

IA-4 (7) : IN-PERSON REGISTRATION

The organization requires that the registration process to receive an individual identifier be conducted in person before a designated registration authority.

Note: In-person registration reduces the likelihood of fraudulent identifiers being issued because it requires the physical presence of individuals and actual face-to-face interactions with designated registration authorities.

Control Family: IDENTIFICATION AND AUTHENTICATION

IA-5 : AUTHENTICATOR MANAGEMENT

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization manages information system authenticators by:

Note: Individual authenticators include, for example, passwords, tokens, biometrics, PKI certificates, and key cards. Initial authenticator content is the actual content (e.g., the initial password) as opposed to requirements about authenticator content (e.g., minimum password length). In many cases, developers ship information system components with factory default authentication credentials to allow for initial installation and configuration. Default authentication credentials are often well known, easily discoverable, and present a significant security risk. The requirement to protect individual authenticators may be implemented via control PL-4 or PS-6 for authenticators in the possession of individuals and by controls AC-3, AC-6, and SC-28 for authenticators stored within organizational information systems (e.g., passwords stored in hashed or encrypted formats, files containing encrypted or hashed passwords accessible with administrator privileges). Information systems support individual authenticator management by organization-defined settings and restrictions for various authenticator characteristics including, for example, minimum password length, password composition, validation time window for time synchronous one-time tokens, and number of allowed rejections during the verification stage of biometric authentication. Specific actions that can be taken to safeguard authenticators include, for example, maintaining possession of individual authenticators, not loaning or sharing individual authenticators with others, and reporting lost, stolen, or compromised authenticators immediately. Authenticator management includes issuing and revoking, when no longer needed, authenticators for temporary access such as that required for remote maintenance. Device authenticators include, for example, certificates and passwords.

Related Controls: *AC-2*, *AC-3*, *AC-6*, *CM-6*, *IA-2*, *IA-4*, *IA-8*, *PL-4*, *PS-5*, *PS-6*, *SC-12*, *SC-13*, *SC-17*, *SC-28*

IA-5a.

Verifying, as part of the initial authenticator distribution, the identity of the individual, group, role, or device receiving the authenticator;

IA-5b.

Establishing initial authenticator content for authenticators defined by the organization;

IA-5c.

Ensuring that authenticators have sufficient strength of mechanism for their intended use;

IA-5d.

Establishing and implementing administrative procedures for initial authenticator distribution, for lost/compromised or damaged authenticators, and for revoking authenticators;

IA-5e.

Changing default content of authenticators prior to information system installation;

IA-5f.

Establishing minimum and maximum lifetime restrictions and reuse conditions for authenticators;

IA-5g.

Changing/refreshing authenticators [Assignment: organization-defined time period by authenticator type];

IA-5h.

Protecting authenticator content from unauthorized disclosure and modification;

IA-5i.

Requiring individuals to take, and having devices implement, specific security safeguards to protect authenticators; and

IA-5j.

Changing authenticators for group/role accounts when membership to those accounts changes.

IA-5 (1) : PASSWORD-BASED AUTHENTICATION

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The information system, for password-based authentication:

Note: This control enhancement applies to single-factor authentication of individuals using passwords as individual or group authenticators, and in a similar manner, when passwords are part of multifactor authenticators. This control enhancement does not apply when passwords are used to unlock hardware authenticators (e.g., Personal Identity Verification cards). The implementation of such password mechanisms may not meet all of the requirements in the enhancement. Cryptographically-protected passwords include, for example, encrypted versions of passwords and one-way cryptographic hashes of passwords. The number of changed characters refers to the number of changes required with respect to the total number of positions in the current password. Password lifetime restrictions do not apply to temporary passwords. To mitigate certain brute force attacks against passwords, organizations may also consider salting passwords.

Related Controls: *IA-6*

IA-5 (1)(a)

Enforces minimum password complexity of [Assignment: organization-defined requirements for case sensitivity, number of characters, mix of upper-case letters, lower-case letters, numbers, and special characters, including minimum requirements for each type];

IA-5 (1)(b)

Enforces at least the following number of changed characters when new passwords are created: [Assignment: organization-defined number];

IA-5 (1)(c)

Stores and transmits only cryptographically-protected passwords;

IA-5 (1)(d)

Enforces password minimum and maximum lifetime restrictions of [Assignment: organization-defined numbers for lifetime minimum, lifetime maximum];

IA-5 (1)(e)

Prohibits password reuse for [Assignment: organization-defined number] generations; and

IA-5 (1)(f)

Allows the use of a temporary password for system logons with an immediate change to a permanent password.

IA-5 (2) : PKI-BASED AUTHENTICATION

Baseline-Impact: *MODERATE*, **HIGH**

The information system, for PKI-based authentication:

Note: Status information for certification paths includes, for example, certificate revocation lists or certificate status protocol responses. For PIV cards, validation of certifications involves the construction and verification of a certification path to the Common Policy Root trust anchor including certificate policy processing.

Related Controls: *IA-6*

IA-5 (2)(a)

Validates certifications by constructing and verifying a certification path to an accepted trust anchor including checking certificate status information;

IA-5 (2)(b)

Enforces authorized access to the corresponding private key;

IA-5 (2)(c)

Maps the authenticated identity to the account of the individual or group; and

IA-5 (2)(d)

Implements a local cache of revocation data to support path discovery and validation in case of inability to access revocation information via the network.

IA-5 (3) : IN-PERSON OR TRUSTED THIRD-PARTY REGISTRATION

Baseline-Impact: *MODERATE*, **HIGH**

The organization requires that the registration process to receive [Assignment: organization-defined types of and/or specific authenticators] be conducted [Selection: in person; by a trusted third party] before [Assignment: organization-defined registration authority] with authorization by [Assignment: organization-defined personnel or roles].

IA-5 (4) : AUTOMATED SUPPORT FOR PASSWORD STRENGTH DETERMINATION

The organization employs automated tools to determine if password authenticators are sufficiently strong to satisfy [Assignment: organization-defined requirements].

Note: This control enhancement focuses on the creation of strong passwords and the characteristics of such passwords (e.g., complexity) prior to use, the enforcement of which is carried out by organizational information systems in IA-5 (1).

Related Controls: *CA-2*, *CA-7*, *RA-5*

IA-5 (5) : CHANGE AUTHENTICATORS PRIOR TO DELIVERY

The organization requires developers/installers of information system components to provide unique authenticators or change default authenticators prior to delivery/installation.

Note: This control enhancement extends the requirement for organizations to change default authenticators upon information system installation, by requiring developers and/or installers to provide unique authenticators or change default authenticators for system components prior to delivery and/or installation. However, it typically does not apply to the developers of commercial off-the-shelve information technology products. Requirements for unique authenticators can be included in acquisition documents prepared by organizations when procuring information systems or system components.

IA-5 (6) : PROTECTION OF AUTHENTICATORS

> The organization protects authenticators commensurate with the security category of the information to which use of the authenticator permits access.

Note: For information systems containing multiple security categories of information without reliable physical or logical separation between categories, authenticators used to grant access to the systems are protected commensurate with the highest security category of information on the systems.

IA-5 (7) : NO EMBEDDED UNENCRYPTED STATIC AUTHENTICATORS

> The organization ensures that unencrypted static authenticators are not embedded in applications or access scripts or stored on function keys.

Note: Organizations exercise caution in determining whether embedded or stored authenticators are in encrypted or unencrypted form. If authenticators are used in the manner stored, then those representations are considered unencrypted authenticators. This is irrespective of whether that representation is perhaps an encrypted version of something else (e.g., a password).

IA-5 (8) : MULTIPLE INFORMATION SYSTEM ACCOUNTS

> The organization implements [Assignment: organization-defined security safeguards] to manage the risk of compromise due to individuals having accounts on multiple information systems.

Note: When individuals have accounts on multiple information systems, there is the risk that the compromise of one account may lead to the compromise of other accounts if individuals use the same authenticators. Possible alternatives include, for example: (i) having different authenticators on all systems; (ii) employing some form of single sign-on mechanism; or (iii) including some form of one-time passwords on all systems.

IA-5 (9) : CROSS-ORGANIZATION CREDENTIAL MANAGEMENT

> The organization coordinates with [Assignment: organization-defined external organizations] for cross-organization management of credentials.

Note: Cross-organization management of credentials provides the capability for organizations to appropriately authenticate individuals, groups, roles, or devices when conducting cross-organization activities involving the processing, storage, or transmission of information.

IA-5 (10) : DYNAMIC CREDENTIAL ASSOCIATION

> The information system dynamically provisions identities.

Note: Authentication requires some form of binding between an identity and the authenticator used to confirm the identity. In conventional approaches, this binding is established by pre-provisioning both the identity and the authenticator to the information system. For example, the binding between a username (i.e., identity) and a password (i.e., authenticator) is accomplished by provisioning the identity and authenticator as a pair in the information system. New authentication techniques allow the binding between the identity and the authenticator to be implemented outside an information system. For example, with smartcard credentials, the identity and the authenticator are bound together on the card. Using these credentials, information systems can authenticate identities that have not been pre-provisioned, dynamically provisioning the identity after authentication. In these situations, organizations can anticipate the dynamic provisioning of identities. Preestablished trust relationships and mechanisms with appropriate authorities to validate identities and related credentials are essential.

IA-5 (11) : HARDWARE TOKEN-BASED AUTHENTICATION

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The information system, for hardware token-based authentication, employs mechanisms that satisfy [Assignment: organization-defined token quality requirements].

Note: Hardware token-based authentication typically refers to the use of PKI-based tokens, such as the U.S. Government Personal Identity Verification (PIV) card. Organizations define specific requirements for tokens, such as working with a particular PKI.

IA-5 (12) : BIOMETRIC-BASED AUTHENTICATION

> The information system, for biometric-based authentication, employs mechanisms that satisfy [Assignment: organization-defined biometric quality requirements].

Note: Unlike password-based authentication which provides exact matches of user-input passwords to stored passwords, biometric authentication does not provide such exact matches. Depending upon the type of biometric and the type of collection mechanism, there is likely to be some divergence from the presented biometric and stored biometric which serves as the basis of comparison. There will likely be both false positives and false negatives when making such comparisons. The rate at which the false accept and false reject rates are equal is known as the crossover rate. Biometric quality requirements include, for example, acceptable crossover rates, as that essentially reflects the accuracy of the biometric.

IA-5 (13) : EXPIRATION OF CACHED AUTHENTICATORS

The information system prohibits the use of cached authenticators after [Assignment: organization-defined time period].

IA-5 (14) : MANAGING CONTENT OF PKI TRUST STORES

The organization, for PKI-based authentication, employs a deliberate organization-wide methodology for managing the content of PKI trust stores installed across all platforms including networks, operating systems, browsers, and applications.

IA-5 (15) : FICAM-APPROVED PRODUCTS AND SERVICES

The organization uses only FICAM-approved path discovery and validation products and services.

Note: Federal Identity, Credential, and Access Management (FICAM)-approved path discovery and validation products and services are those products and services that have been approved through the FICAM conformance program, where applicable.

Control Family: IDENTIFICATION AND AUTHENTICATION

IA-6 : AUTHENTICATOR FEEDBACK

Priority: P2

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The information system obscures feedback of authentication information during the authentication process to protect the information from possible exploitation/use by unauthorized individuals.

Note: The feedback from information systems does not provide information that would allow unauthorized individuals to compromise authentication mechanisms. For some types of information systems or system components, for example, desktops/notebooks with relatively large monitors, the threat (often referred to as shoulder surfing) may be significant. For other types of systems or components, for example, mobile devices with 2-4 inch screens, this threat may be less significant, and may need to be balanced against the increased likelihood of typographic input errors due to the small keyboards. Therefore, the means for obscuring the authenticator feedback is selected accordingly. Obscuring the feedback of authentication information includes, for example, displaying asterisks when users type passwords into input devices, or displaying feedback for a very limited time before fully obscuring it.

Related Controls: *PE-18*

Control Family: IDENTIFICATION AND AUTHENTICATION

IA-7 : CRYPTOGRAPHIC MODULE AUTHENTICATION

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The information system implements mechanisms for authentication to a cryptographic module that meet the requirements of applicable federal laws, Executive Orders, directives, policies, regulations, standards, and guidance for such authentication.

Note: Authentication mechanisms may be required within a cryptographic module to authenticate an operator accessing the module and to verify that the operator is authorized to assume the requested role and perform services within that role.

Related Controls: *SC-12*, *SC-13*

Control Family: IDENTIFICATION AND AUTHENTICATION

IA-8 : IDENTIFICATION AND AUTHENTICATION (NON-ORGANIZATIONAL USERS)

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The information system uniquely identifies and authenticates non-organizational users (or processes acting on behalf of non-organizational users).

Note: Non-organizational users include information system users other than organizational users explicitly covered by IA-2. These individuals are uniquely identified and authenticated for accesses other than those accesses explicitly identified and documented in AC-14. In accordance with the E-Authentication E-Government initiative, authentication of non-organizational users accessing federal information systems may be required to protect federal, proprietary, or privacy-related information (with exceptions noted for national security systems). Organizations use risk assessments to determine authentication needs and consider scalability, practicality, and security in balancing the need to ensure ease of use for access to federal information and information systems with the need to protect and adequately mitigate risk. IA-2 addresses identification and authentication requirements for access to information systems by organizational users.

Related Controls: *AC-2*, *AC-14*, *AC-17*, *AC-18*, *IA-2*, *IA-4*, *IA-5*, *MA-4*, *RA-3*, *SA-12*, *SC-8*

IA-8 (1) : ACCEPTANCE OF PIV CREDENTIALS FROM OTHER AGENCIES

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The information system accepts and electronically verifies Personal Identity Verification (PIV) credentials from other federal agencies.

Note: This control enhancement applies to logical access control systems (LACS) and physical access control systems (PACS). Personal Identity Verification (PIV) credentials are those credentials issued by federal agencies that conform to FIPS Publication 201 and supporting guidance documents. OMB Memorandum 11-11 requires federal agencies to continue implementing the requirements specified in HSPD-12 to enable agency-wide use of PIV credentials.

Related Controls: *AU-2*, *PE-3*, *SA-4*

IA-8 (2) : ACCEPTANCE OF THIRD-PARTY CREDENTIALS

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The information system accepts only FICAM-approved third-party credentials.

Note: This control enhancement typically applies to organizational information systems that are accessible to the general public, for example, public-facing websites. Third-party credentials are those credentials issued by nonfederal government entities approved by the Federal Identity, Credential, and Access Management (FICAM) Trust Framework Solutions initiative. Approved third-party credentials meet or exceed the set of minimum federal government-wide technical, security, privacy, and organizational maturity requirements. This allows federal government relying parties to trust such credentials at their approved assurance levels.

Related Controls: *AU-2*

IA-8 (3) : USE OF FICAM-APPROVED PRODUCTS

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The organization employs only FICAM-approved information system components in [Assignment: organization-defined information systems] to accept third-party credentials.

Note: This control enhancement typically applies to information systems that are accessible to the general public, for example, public-facing websites. FICAM-approved information system components include, for example, information technology products and software libraries that have been approved by the Federal Identity, Credential, and Access Management conformance program.

Related Controls: *SA-4*

IA-8 (4) : USE OF FICAM-ISSUED PROFILES

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The information system conforms to FICAM-issued profiles.

Note: This control enhancement addresses open identity management standards. To ensure that these standards are viable, robust, reliable, sustainable (e.g., available in commercial information technology products), and interoperable as documented, the United States Government assesses and scopes identity management standards and technology implementations against applicable federal legislation, directives, policies, and requirements. The result is FICAM-issued implementation profiles of approved protocols (e.g., FICAM authentication protocols such as SAML 2.0 and OpenID 2.0, as well as other protocols such as the FICAM Backend Attribute Exchange).

Related Controls: *SA-4*

IA-8 (5) : ACCEPTANCE OF PIV-I CREDENTIALS

> The information system accepts and electronically verifies Personal Identity Verification-I (PIV-I) credentials.

Note: This control enhancement: (i) applies to logical and physical access control systems; and (ii) addresses Non-Federal Issuers (NFIs) of identity cards that desire to interoperate with United States Government Personal Identity Verification (PIV) information systems and that can be trusted by federal government-relying parties. The X.509 certificate policy for the Federal Bridge Certification Authority (FBCA) addresses PIV-I requirements. The PIV-I card is suitable for Assurance Level 4 as defined in OMB Memorandum 04-04 and NIST Special Publication 800-63, and multifactor authentication as defined in NIST Special Publication 800-116. PIV-I credentials are those credentials issued by a PIV-I provider whose PIV-I certificate policy maps to the Federal Bridge PIV-I Certificate Policy. A PIV-I provider is cross-certified (directly or through another PKI bridge) with the FBCA with policies that have been mapped and approved as meeting the requirements of the PIV-I policies defined in the FBCA certificate policy.

Related Controls: *AU-2*

Control Family: IDENTIFICATION AND AUTHENTICATION

IA-9 : SERVICE IDENTIFICATION AND AUTHENTICATION

Priority: P0

> The organization identifies and authenticates [Assignment: organization-defined information system services] using [Assignment: organization-defined security safeguards].

Note: This control supports service-oriented architectures and other distributed architectural approaches requiring the identification and authentication of information system services. In such architectures, external services often appear dynamically. Therefore, information systems should be able to determine in a dynamic manner, if external providers and associated services are authentic. Safeguards implemented by organizational information systems to validate provider and service authenticity include, for example, information or code signing, provenance graphs, and/or electronic signatures indicating or including the sources of services.

IA-9 (1) : INFORMATION EXCHANGE

> The organization ensures that service providers receive, validate, and transmit identification and authentication information.

IA-9 (2) : TRANSMISSION OF DECISIONS

> The organization ensures that identification and authentication decisions are transmitted between [Assignment: organization-defined services] consistent with organizational policies.

Note: For distributed architectures (e.g., service-oriented architectures), the decisions regarding the validation of identification and authentication claims may be made by services separate from the services acting on those decisions. In such situations, it is necessary to provide the identification and authentication decisions (as opposed to the actual identifiers and authenticators) to the services that need to act on those decisions.

Related Controls: *SC-8*

Control Family: IDENTIFICATION AND AUTHENTICATION

IA-10 : ADAPTIVE IDENTIFICATION AND AUTHENTICATION

Priority: P0

> The organization requires that individuals accessing the information system employ [Assignment: organization-defined supplemental authentication techniques or mechanisms] under specific [Assignment: organization-defined circumstances or situations].

Note: Adversaries may compromise individual authentication mechanisms and subsequently attempt to impersonate legitimate users. This situation can potentially occur with any authentication mechanisms employed by organizations. To address this threat, organizations may employ specific techniques/mechanisms and establish protocols to assess suspicious behavior (e.g., individuals accessing information that they do not typically access as part of their normal duties, roles, or responsibilities, accessing greater quantities of information than the individuals would routinely access, or attempting to access information from suspicious network addresses). In these situations when certain preestablished conditions or triggers occur, organizations can require selected individuals to provide additional authentication information. Another potential use for adaptive identification and authentication is to increase the strength of mechanism based on the number and/or types of records being accessed.

Related Controls: *AU-6*, *SI-4*

Control Family: IDENTIFICATION AND AUTHENTICATION

IA-11 : RE-AUTHENTICATION

Priority: P0

> The organization requires users and devices to re-authenticate when [Assignment: organization-defined circumstances or situations requiring re-authentication].

Note: In addition to the re-authentication requirements associated with session locks, organizations may require re-authentication of individuals and/or devices in other situations including, for example: (i) when authenticators change; (ii), when roles change; (iii) when security categories of information systems change; (iv), when the execution of privileged functions occurs; (v) after a fixed period of time; or (vi) periodically.

Related Controls: *AC-11*

Control Family: INCIDENT RESPONSE

IR-1 : INCIDENT RESPONSE POLICY AND PROCEDURES

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: This control addresses the establishment of policy and procedures for the effective implementation of selected security controls and control enhancements in the IR family. Policy and procedures reflect applicable federal laws, Executive Orders, directives, regulations, policies, standards, and guidance. Security program policies and procedures at the organization level may make the need for system-specific policies and procedures unnecessary. The policy can be included as part of the general information security policy for organizations or conversely, can be represented by multiple policies reflecting the complex nature of certain organizations. The procedures can be established for the security program in general and for particular information systems, if needed. The organizational risk management strategy is a key factor in establishing policy and procedures.

Related Controls: *PM-9*

IR-1a.

Develops, documents, and disseminates to [Assignment: organization-defined personnel or roles]:

IR-1a.1.

An incident response policy that addresses purpose, scope, roles, responsibilities, management commitment, coordination among organizational entities, and compliance; and

IR-1a.2.

Procedures to facilitate the implementation of the incident response policy and associated incident response controls; and

IR-1b.

Reviews and updates the current:

IR-1b.1.

Incident response policy [Assignment: organization-defined frequency]; and

IR-1b.2.

Incident response procedures [Assignment: organization-defined frequency].

Control Family: INCIDENT RESPONSE

IR-2 : INCIDENT RESPONSE TRAINING

Priority: P2

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The organization provides incident response training to information system users consistent with assigned roles and responsibilities:

Note: Incident response training provided by organizations is linked to the assigned roles and responsibilities of organizational personnel to ensure the appropriate content and level of detail is included in such training. For example, regular users may only need to know who to call or how to recognize an incident on the information system; system administrators may require additional training on how to handle/remediate incidents; and incident responders may receive more specific training on forensics, reporting, system recovery, and restoration. Incident response training includes user training in the identification and reporting of suspicious activities, both from external and internal sources.

Related Controls: *AT-3*, *CP-3*, *IR-8*

IR-2a.

Within [Assignment: organization-defined time period] of assuming an incident response role or responsibility;

IR-2b.

When required by information system changes; and

IR-2c.

[Assignment: organization-defined frequency] thereafter.

IR-2 (1) : SIMULATED EVENTS

Baseline-Impact: **HIGH**

The organization incorporates simulated events into incident response training to facilitate effective response by personnel in crisis situations.

IR-2 (2) : AUTOMATED TRAINING ENVIRONMENTS

Baseline-Impact: **HIGH**

The organization employs automated mechanisms to provide a more thorough and realistic incident response training environment.

Control Family: INCIDENT RESPONSE

IR-3 : INCIDENT RESPONSE TESTING

Priority: P2

Baseline-Impact: *MODERATE*, **HIGH**

The organization tests the incident response capability for the information system [Assignment: organization-defined frequency] using [Assignment: organization-defined tests] to determine the incident response effectiveness and documents the results.

Note: Organizations test incident response capabilities to determine the overall effectiveness of the capabilities and to identify potential weaknesses or deficiencies. Incident response testing includes, for example, the use of checklists, walk-through or tabletop exercises, simulations (parallel/full interrupt), and comprehensive exercises. Incident response testing can also include a determination of the effects on organizational operations (e.g., reduction in mission capabilities), organizational assets, and individuals due to incident response.

Related Controls: *CP-4*, *IR-8*

IR-3 (1) : AUTOMATED TESTING

The organization employs automated mechanisms to more thoroughly and effectively test the incident response capability.

Note: Organizations use automated mechanisms to more thoroughly and effectively test incident response capabilities, for example: (i) by providing more complete coverage of incident response issues; (ii) by selecting more realistic test scenarios and test environments; and (iii) by stressing the response capability.

Related Controls: *AT-2*

IR-3 (2) : COORDINATION WITH RELATED PLANS

Baseline-Impact: *MODERATE*, **HIGH**

The organization coordinates incident response testing with organizational elements responsible for related plans.

Note: Organizational plans related to incident response testing include, for example, Business Continuity Plans, Contingency Plans, Disaster Recovery Plans, Continuity of Operations Plans, Crisis Communications Plans, Critical Infrastructure Plans, and Occupant Emergency Plans.

Control Family: INCIDENT RESPONSE

IR-4 : INCIDENT HANDLING

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: Organizations recognize that incident response capability is dependent on the capabilities of organizational information systems and the mission/business processes being supported by those systems. Therefore, organizations consider incident response as part of the definition, design, and development of mission/business processes and information systems. Incident-related information can be obtained from a variety of sources including, for example, audit monitoring, network monitoring, physical access monitoring, user/administrator reports, and reported supply chain events. Effective incident handling capability includes coordination among many organizational entities including, for example, mission/business owners, information system owners, authorizing officials, human resources offices, physical and personnel security offices, legal departments, operations personnel, procurement offices, and the risk executive (function).

Related Controls: *AU-6*, *CM-6*, *CP-2*, *CP-4*, *IR-2*, *IR-3*, *IR-8*, *PE-6*, *SC-5*, *SC-7*, *SI-3*, *SI-4*, *SI-7*

IR-4a.

Implements an incident handling capability for security incidents that includes preparation, detection and analysis, containment, eradication, and recovery;

IR-4b.

Coordinates incident handling activities with contingency planning activities; and

IR-4c.

Incorporates lessons learned from ongoing incident handling activities into incident response procedures, training, and testing, and implements the resulting changes accordingly.

IR-4 (1) : AUTOMATED INCIDENT HANDLING PROCESSES

Baseline-Impact: *MODERATE*, **HIGH**

The organization employs automated mechanisms to support the incident handling process.

Note: Automated mechanisms supporting incident handling processes include, for example, online incident management systems.

IR-4 (2) : DYNAMIC RECONFIGURATION

The organization includes dynamic reconfiguration of [Assignment: organization-defined information system components] as part of the incident response capability.

Note: Dynamic reconfiguration includes, for example, changes to router rules, access control lists, intrusion detection/prevention system parameters, and filter rules for firewalls and gateways. Organizations perform dynamic reconfiguration of information systems, for example, to stop attacks, to misdirect attackers, and to isolate components of systems, thus limiting the extent of the damage from breaches or compromises. Organizations include time frames for achieving the reconfiguration of information systems in the definition of the reconfiguration capability, considering the potential need for rapid response in order to effectively address sophisticated cyber threats.

Related Controls: *AC-2*, *AC-4*, *AC-16*, *CM-2*, *CM-3*, *CM-4*

IR-4 (3) : CONTINUITY OF OPERATIONS

The organization identifies [Assignment: organization-defined classes of incidents] and [Assignment: organization-defined actions to take in response to classes of incidents] to ensure continuation of organizational missions and business functions.

Note: Classes of incidents include, for example, malfunctions due to design/implementation errors and omissions, targeted malicious attacks, and untargeted malicious attacks. Appropriate incident response actions include, for example, graceful degradation, information system shutdown, fall back to manual mode/alternative technology whereby the system operates differently, employing deceptive measures, alternate information flows, or operating in a mode that is reserved solely for when systems are under attack.

IR-4 (4) : INFORMATION CORRELATION

Baseline-Impact: HIGH

> The organization correlates incident information and individual incident responses to achieve an organization-wide perspective on incident awareness and response.

Note: Sometimes the nature of a threat event, for example, a hostile cyber attack, is such that it can only be observed by bringing together information from different sources including various reports and reporting procedures established by organizations.

IR-4 (5) : AUTOMATIC DISABLING OF INFORMATION SYSTEM

> The organization implements a configurable capability to automatically disable the information system if [Assignment: organization-defined security violations] are detected.

IR-4 (6) : INSIDER THREATS - SPECIFIC CAPABILITIES

> The organization implements incident handling capability for insider threats.

Note: While many organizations address insider threat incidents as an inherent part of their organizational incident response capability, this control enhancement provides additional emphasis on this type of threat and the need for specific incident handling capabilities (as defined within organizations) to provide appropriate and timely responses.

IR-4 (7) : INSIDER THREATS - INTRA-ORGANIZATION COORDINATION

> The organization coordinates incident handling capability for insider threats across [Assignment: organization-defined components or elements of the organization].

Note: Incident handling for insider threat incidents (including preparation, detection and analysis, containment, eradication, and recovery) requires close coordination among a variety of organizational components or elements to be effective. These components or elements include, for example, mission/business owners, information system owners, human resources offices, procurement offices, personnel/physical security offices, operations personnel, and risk executive (function). In addition, organizations may require external support from federal, state, and local law enforcement agencies.

IR-4 (8) : CORRELATION WITH EXTERNAL ORGANIZATIONS

> The organization coordinates with [Assignment: organization-defined external organizations] to correlate and share [Assignment: organization-defined incident information] to achieve a cross-organization perspective on incident awareness and more effective incident responses.

Note: The coordination of incident information with external organizations including, for example, mission/business partners, military/coalition partners, customers, and multitiered developers, can provide significant benefits. Cross-organizational coordination with respect to incident handling can serve as an important risk management capability.

This capability allows organizations to leverage critical information from a variety of sources to effectively respond to information security-related incidents potentially affecting the organizations operations, assets, and individuals.

IR-4 (9) : DYNAMIC RESPONSE CAPABILITY

> The organization employs [Assignment: organization-defined dynamic response capabilities] to effectively respond to security incidents.

Note: This control enhancement addresses the deployment of replacement or new capabilities in a timely manner in response to security incidents (e.g., adversary actions during hostile cyber attacks). This includes capabilities implemented at the mission/business process level (e.g., activating alternative mission/business processes) and at the information system level.

Related Controls: *CP-10*

IR-4 (10) : SUPPLY CHAIN COORDINATION

> The organization coordinates incident handling activities involving supply chain events with other organizations involved in the supply chain.

Note: Organizations involved in supply chain activities include, for example, system/product developers, integrators, manufacturers, packagers, assemblers, distributors, vendors, and resellers. Supply chain incidents include, for example, compromises/breaches involving information system components, information technology products, development processes or personnel, and distribution processes or warehousing facilities.

Control Family: INCIDENT RESPONSE

IR-5 : INCIDENT MONITORING

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The organization tracks and documents information system security incidents.

Note: Documenting information system security incidents includes, for example, maintaining records about each incident, the status of the incident, and other pertinent information necessary for forensics, evaluating incident details, trends, and handling. Incident information can be obtained from a variety of sources including, for example, incident reports, incident response teams, audit monitoring, network monitoring, physical access monitoring, and user/administrator reports.

Related Controls: *AU-6*, *IR-8*, *PE-6*, *SC-5*, *SC-7*, *SI-3*, *SI-4*, *SI-7*

IR-5 (1) : AUTOMATED TRACKING / DATA COLLECTION / ANALYSIS

Baseline-Impact: **HIGH**

The organization employs automated mechanisms to assist in the tracking of security incidents and in the collection and analysis of incident information.

Note: Automated mechanisms for tracking security incidents and collecting/analyzing incident information include, for example, the Einstein network monitoring device and monitoring online Computer Incident Response Centers (CIRCs) or other electronic databases of incidents.

Related Controls: *AU-7*, *IR-4*

Control Family: INCIDENT RESPONSE

IR-6 : INCIDENT REPORTING

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: The intent of this control is to address both specific incident reporting requirements within an organization and the formal incident reporting requirements for federal agencies and their subordinate organizations. Suspected security incidents include, for example, the receipt of suspicious email communications that can potentially contain malicious code. The types of security incidents reported, the content and timeliness of the reports, and the designated reporting authorities reflect applicable federal laws, Executive Orders, directives, regulations, policies, standards, and guidance. Current federal policy requires that all federal agencies (unless specifically exempted from such requirements) report security incidents to the United States Computer Emergency Readiness Team (US-CERT) within specified time frames designated in the US-CERT Concept of Operations for Federal Cyber Security Incident Handling.

Related Controls: *IR-4*, *IR-5*, *IR-8*

IR-6a.

Requires personnel to report suspected security incidents to the organizational incident response capability within [Assignment: organization-defined time period]; and

IR-6b.

Reports security incident information to [Assignment: organization-defined authorities].

IR-6 (1) : AUTOMATED REPORTING

Baseline-Impact: *MODERATE*, **HIGH**

The organization employs automated mechanisms to assist in the reporting of security incidents.

Related Controls: *IR-7*

IR-6 (2) : VULNERABILITIES RELATED TO INCIDENTS

The organization reports information system vulnerabilities associated with reported security incidents to [Assignment: organization-defined personnel or roles].

IR-6 (3) : COORDINATION WITH SUPPLY CHAIN

The organization provides security incident information to other organizations involved in the supply chain for information systems or information system components related to the incident.

Note: Organizations involved in supply chain activities include, for example, system/product developers, integrators, manufacturers, packagers, assemblers, distributors, vendors, and resellers. Supply chain incidents include, for example, compromises/breaches involving information system components, information technology products, development processes or personnel, and distribution processes or warehousing facilities. Organizations determine the appropriate information to share considering the value gained from support by external organizations with the potential for harm due to sensitive information being released to outside organizations of perhaps questionable trustworthiness.

Control Family: INCIDENT RESPONSE

IR-7 : INCIDENT RESPONSE ASSISTANCE

Priority: P2

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization provides an incident response support resource, integral to the organizational incident response capability that offers advice and assistance to users of the information system for the handling and reporting of security incidents.

Note: Incident response support resources provided by organizations include, for example, help desks, assistance groups, and access to forensics services, when required.

Related Controls: *AT-2*, *IR-4*, *IR-6*, *IR-8*, *SA-9*

IR-7 (1) : AUTOMATION SUPPORT FOR AVAILABILITY OF INFORMATION / SUPPORT

Baseline-Impact: *MODERATE*, **HIGH**

The organization employs automated mechanisms to increase the availability of incident response-related information and support.

Note: Automated mechanisms can provide a push and/or pull capability for users to obtain incident response assistance. For example, individuals might have access to a website to query the assistance capability, or conversely, the assistance capability may have the ability to proactively send information to users (general distribution or targeted) as part of increasing understanding of current response capabilities and support.

IR-7 (2) : COORDINATION WITH EXTERNAL PROVIDERS

The organization:

Note: External providers of information system protection capability include, for example, the Computer Network Defense program within the U.S. Department of Defense. External providers help to protect, monitor, analyze, detect, and respond to unauthorized activity within organizational information systems and networks.

IR-7 (2)(a)

Establishes a direct, cooperative relationship between its incident response capability and external providers of information system protection capability; and

IR-7 (2)(b)

Identifies organizational incident response team members to the external providers.

Control Family: INCIDENT RESPONSE

IR-8 : INCIDENT RESPONSE PLAN

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: It is important that organizations develop and implement a coordinated approach to incident response. Organizational missions, business functions, strategies, goals, and objectives for incident response help to determine the structure of incident response capabilities. As part of a comprehensive incident response capability, organizations consider the coordination and sharing of information with external organizations, including, for example, external service providers and organizations involved in the supply chain for organizational information systems.

Related Controls: *MP-2*, *MP-4*, *MP-5*

IR-8a.

Develops an incident response plan that:

IR-8a.1.

Provides the organization with a roadmap for implementing its incident response capability;

IR-8a.2.

Describes the structure and organization of the incident response capability;

IR-8a.3.

Provides a high-level approach for how the incident response capability fits into the overall organization;

IR-8a.4.

Meets the unique requirements of the organization, which relate to mission, size, structure, and functions;

IR-8a.5.

Defines reportable incidents;

IR-8a.6.

Provides metrics for measuring the incident response capability within the organization;

IR-8a.7.

Defines the resources and management support needed to effectively maintain and mature an incident response capability; and

IR-8a.8.

Is reviewed and approved by [Assignment: organization-defined personnel or roles];

IR-8b.

Distributes copies of the incident response plan to [Assignment: organization-defined incident response personnel (identified by name and/or by role) and organizational elements];

IR-8c.

Reviews the incident response plan [Assignment: organization-defined frequency];

IR-8d.

Updates the incident response plan to address system/organizational changes or problems encountered during plan implementation, execution, or testing;

IR-8e.

Communicates incident response plan changes to [Assignment: organization-defined incident response personnel (identified by name and/or by role) and organizational elements]; and

IR-8f.

Protects the incident response plan from unauthorized disclosure and modification.

Control Family: INCIDENT RESPONSE

IR-9 : INFORMATION SPILLAGE RESPONSE

Priority: P0

The organization responds to information spills by:

Note: Information spillage refers to instances where either classified or sensitive information is inadvertently placed on information systems that are not authorized to process such information. Such information spills often occur when information that is initially thought to be of lower sensitivity is transmitted to an information system and then is subsequently determined to be of higher sensitivity. At that point, corrective action is required. The nature of the organizational response is generally based upon the degree of sensitivity of the spilled information (e.g., security category or classification level), the security capabilities of the information system, the specific nature of contaminated storage media, and the access authorizations (e.g., security clearances) of individuals with authorized access to the contaminated system. The methods used to communicate information about the spill after the fact do not involve methods directly associated with the actual spill to minimize the risk of further spreading the contamination before such contamination is isolated and eradicated.

IR-9a.

Identifying the specific information involved in the information system contamination;

IR-9b.

Alerting [Assignment: organization-defined personnel or roles] of the information spill using a method of communication not associated with the spill;

IR-9c.

Isolating the contaminated information system or system component;

IR-9d.

Eradicating the information from the contaminated information system or component;

IR-9e.

Identifying other information systems or system components that may have been subsequently contaminated; and

IR-9f.

Performing other [Assignment: organization-defined actions].

IR-9 (1) : RESPONSIBLE PERSONNEL

The organization assigns [Assignment: organization-defined personnel or roles] with responsibility for responding to information spills.

IR-9 (2) : TRAINING

The organization provides information spillage response training [Assignment: organization-defined frequency].

IR-9 (3) : POST-SPILL OPERATIONS

The organization implements [Assignment: organization-defined procedures] to ensure that organizational personnel impacted by information spills can continue to carry out assigned tasks while contaminated systems are undergoing corrective actions.

Note: Correction actions for information systems contaminated due to information spillages may be very time-consuming. During those periods, personnel may not have access to the contaminated systems, which may potentially affect their ability to conduct organizational business.

IR-9 (4) : EXPOSURE TO UNAUTHORIZED PERSONNEL

The organization employs [Assignment: organization-defined security safeguards] for personnel exposed to information not within assigned access authorizations.

Note: Security safeguards include, for example, making personnel exposed to spilled information aware of the federal laws, directives, policies, and/or regulations regarding the information and the restrictions imposed based on exposure to such information.

Control Family: INCIDENT RESPONSE

IR-10 : INTEGRATED INFORMATION SECURITY ANALYSIS TEAM

Priority: P0

The organization establishes an integrated team of forensic/malicious code analysts, tool developers, and real-time operations personnel.

Note: Having an integrated team for incident response facilitates information sharing. Such capability allows organizational personnel, including developers, implementers, and operators, to leverage the team knowledge of the threat in order to implement defensive measures that will enable organizations to deter intrusions more effectively. Moreover, it promotes the rapid detection of intrusions, development of appropriate mitigations, and the deployment of effective defensive measures. For example, when an intrusion is detected, the integrated security analysis team can rapidly develop an appropriate response for operators to implement, correlate the new incident with information on past intrusions, and augment ongoing intelligence development. This enables the team to identify adversary TTPs that are linked to the operations tempo or to specific missions/business functions, and to define responsive actions in a way that does not disrupt the mission/business operations. Ideally, information security analysis teams are distributed within organizations to make the capability more resilient.

Control Family: MAINTENANCE

MA-1 : SYSTEM MAINTENANCE POLICY AND PROCEDURES

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: This control addresses the establishment of policy and procedures for the effective implementation of selected security controls and control enhancements in the MA family. Policy and procedures reflect applicable federal laws, Executive Orders, directives, regulations, policies, standards, and guidance. Security program policies and procedures at the organization level may make the need for system-specific policies and procedures unnecessary. The policy can be included as part of the general information security policy for organizations or conversely, can be represented by multiple policies reflecting the complex nature of certain organizations. The procedures can be established for the security program in general and for particular information systems, if needed. The organizational risk management strategy is a key factor in establishing policy and procedures.

Related Controls: *PM-9*

MA-1a.

Develops, documents, and disseminates to [Assignment: organization-defined personnel or roles]:

MA-1a.1.

A system maintenance policy that addresses purpose, scope, roles, responsibilities, management commitment, coordination among organizational entities, and compliance; and

MA-1a.2.

Procedures to facilitate the implementation of the system maintenance policy and associated system maintenance controls; and

MA-1b.

Reviews and updates the current:

MA-1b.1.

System maintenance policy [Assignment: organization-defined frequency]; and

MA-1b.2.

System maintenance procedures [Assignment: organization-defined frequency].

Control Family: MAINTENANCE

MA-2 : CONTROLLED MAINTENANCE

Priority: P2

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: This control addresses the information security aspects of the information system maintenance program and applies to all types of maintenance to any system component (including applications) conducted by any local or nonlocal entity (e.g., in-contract, warranty, in-house, software maintenance agreement). System maintenance also includes those components not directly associated with information processing and/or data/information retention such as scanners, copiers, and printers. Information necessary for creating effective maintenance records includes, for example: (i) date and time of maintenance; (ii) name of individuals or group performing the maintenance; (iii) name of escort, if necessary; (iv) a description of the maintenance performed; and (v) information system components/equipment removed or replaced (including identification numbers, if applicable). The level of detail included in maintenance records can be informed by the security categories of organizational information systems. Organizations consider supply chain issues associated with replacement components for information systems.

Related Controls: *CM-3*, *CM-4*, *MA-4*, *MP-6*, *PE-16*, *SA-12*, *SI-2*

MA-2a.

Schedules, performs, documents, and reviews records of maintenance and repairs on information system components in accordance with manufacturer or vendor specifications and/or organizational requirements;

MA-2b.

Approves and monitors all maintenance activities, whether performed on site or remotely and whether the equipment is serviced on site or removed to another location;

MA-2c.

Requires that [Assignment: organization-defined personnel or roles] explicitly approve the removal of the information system or system components from organizational facilities for off-site maintenance or repairs;

MA-2d.

Sanitizes equipment to remove all information from associated media prior to removal from organizational facilities for off-site maintenance or repairs;

MA-2e.

Checks all potentially impacted security controls to verify that the controls are still functioning properly following maintenance or repair actions; and

MA-2f.

Includes [Assignment: organization-defined maintenance-related information] in organizational maintenance records.

MA-2 (1) : RECORD CONTENT

[Withdrawn: Incorporated into MA-2].

MA-2 (2) : AUTOMATED MAINTENANCE ACTIVITIES

Baseline-Impact: HIGH

The organization:

Related Controls: *CA-7*, *MA-3*

MA-2 (2)(a)

Employs automated mechanisms to schedule, conduct, and document maintenance and repairs; and

MA-2 (2)(b)

Produces up-to date, accurate, and complete records of all maintenance and repair actions requested, scheduled, in process, and completed.

Control Family: MAINTENANCE

MA-3 : MAINTENANCE TOOLS

Priority: P3

Baseline-Impact: *MODERATE*, **HIGH**

The organization approves, controls, and monitors information system maintenance tools.

Note: This control addresses security-related issues associated with maintenance tools used specifically for diagnostic and repair actions on organizational information systems. Maintenance tools can include hardware, software, and firmware items. Maintenance tools are potential vehicles for transporting malicious code, either intentionally or unintentionally, into a facility and subsequently into organizational information systems. Maintenance tools can include, for example, hardware/software diagnostic test equipment and hardware/software packet sniffers. This control does not cover hardware/software components that may support information system maintenance, yet are a part of the system, for example, the software implementing ping, ls, ipconfig, or the hardware and software implementing the monitoring port of an Ethernet switch.

Related Controls: *MA-2*, *MA-5*, *MP-6*

MA-3 (1) : INSPECT TOOLS

Baseline-Impact: *MODERATE*, **HIGH**

The organization inspects the maintenance tools carried into a facility by maintenance personnel for improper or unauthorized modifications.

Note: If, upon inspection of maintenance tools, organizations determine that the tools have been modified in an improper/unauthorized manner or contain malicious code, the incident is handled consistent with organizational policies and procedures for incident handling.

Related Controls: *SI-7*

MA-3 (2) : INSPECT MEDIA

Baseline-Impact: *MODERATE*, **HIGH**

> The organization checks media containing diagnostic and test programs for malicious code before the media are used in the information system.

Note: If, upon inspection of media containing maintenance diagnostic and test programs, organizations determine that the media contain malicious code, the incident is handled consistent with organizational incident handling policies and procedures.

Related Controls: *SI-3*

MA-3 (3) : PREVENT UNAUTHORIZED REMOVAL

Baseline-Impact: **HIGH**

> The organization prevents the unauthorized removal of maintenance equipment containing organizational information by:

Note: Organizational information includes all information specifically owned by organizations and information provided to organizations in which organizations serve as information stewards.

MA-3 (3)(a)

> Verifying that there is no organizational information contained on the equipment;

MA-3 (3)(b)

> Sanitizing or destroying the equipment;

MA-3 (3)(c)

> Retaining the equipment within the facility; or

MA-3 (3)(d)

> Obtaining an exemption from [Assignment: organization-defined personnel or roles] explicitly authorizing removal of the equipment from the facility.

MA-3 (4) : RESTRICTED TOOL USE

The information system restricts the use of maintenance tools to authorized personnel only.

Note: This control enhancement applies to information systems that are used to carry out maintenance functions.

Related Controls: *AC-2*, *AC-3*, *AC-5*, *AC-6*

Control Family: MAINTENANCE

MA-4 : NONLOCAL MAINTENANCE

Priority: P2

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: Nonlocal maintenance and diagnostic activities are those activities conducted by individuals communicating through a network, either an external network (e.g., the Internet) or an internal network. Local maintenance and diagnostic activities are those activities carried out by individuals physically present at the information system or information system component and not communicating across a network connection. Authentication techniques used in the establishment of nonlocal maintenance and diagnostic sessions reflect the network access requirements in IA-2. Typically, strong authentication requires authenticators that are resistant to replay attacks and employ multifactor authentication. Strong authenticators include, for example, PKI where certificates are stored on a token protected by a password, passphrase, or biometric. Enforcing requirements in MA-4 is accomplished in part by other controls.

Related Controls: *AC-2*, *AC-3*, *AC-6*, *AC-17*, *AU-2*, *AU-3*, *IA-2*, *IA-4*, *IA-5*, *IA-8*, *MA-2*, *MA-5*, *MP-6*, *PL-2*, *SC-7*, *SC-10*, *SC-17*

MA-4a.

Approves and monitors nonlocal maintenance and diagnostic activities;

MA-4b.

Allows the use of nonlocal maintenance and diagnostic tools only as consistent with organizational policy and documented in the security plan for the information system;

MA-4c.

Employs strong authenticators in the establishment of nonlocal maintenance and diagnostic sessions;

MA-4d.

Maintains records for nonlocal maintenance and diagnostic activities; and

MA-4e.

Terminates session and network connections when nonlocal maintenance is completed.

MA-4 (1) : AUDITING AND REVIEW

The organization:

Related Controls: *AU-2*, *AU-6*, *AU-12*

MA-4 (1)(a)

Audits nonlocal maintenance and diagnostic sessions [Assignment: organization-defined audit events]; and

MA-4 (1)(b)

Reviews the records of the maintenance and diagnostic sessions.

MA-4 (2) : DOCUMENT NONLOCAL MAINTENANCE

Baseline-Impact: *MODERATE*, **HIGH**

The organization documents in the security plan for the information system, the policies and procedures for the establishment and use of nonlocal maintenance and diagnostic connections.

MA-4 (3) : COMPARABLE SECURITY / SANITIZATION

Baseline-Impact: **HIGH**

The organization:

Note: Comparable security capability on information systems, diagnostic tools, and equipment providing maintenance services implies that the implemented security controls on those systems, tools, and equipment are at least as comprehensive as the controls on the information system being serviced.

Related Controls: *MA-3*, *SA-12*, *SI-3*, *SI-7*

MA-4 (3)(a)

Requires that nonlocal maintenance and diagnostic services be performed from an information system that implements a security capability comparable to the capability implemented on the system being serviced; or

MA-4 (3)(b)

Removes the component to be serviced from the information system prior to nonlocal maintenance or diagnostic services, sanitizes the component (with regard to organizational information) before removal from organizational facilities, and after the service is performed, inspects and sanitizes the component (with regard to potentially malicious software) before reconnecting the component to the information system.

MA-4 (4) : AUTHENTICATION / SEPARATION OF MAINTENANCE SESSIONS

The organization protects nonlocal maintenance sessions by:

Related Controls: *SC-13*

MA-4 (4)(a)

Employing [Assignment: organization-defined authenticators that are replay resistant]; and

MA-4 (4)(b)

Separating the maintenance sessions from other network sessions with the information system by either:

MA-4 (4)(b)(1)

Physically separated communications paths; or

MA-4 (4)(b)(2)

Logically separated communications paths based upon encryption.

MA-4 (5) : APPROVALS AND NOTIFICATIONS

The organization:

Note: Notification may be performed by maintenance personnel. Approval of nonlocal maintenance sessions is accomplished by organizational personnel with sufficient information security and information system knowledge to determine the appropriateness of the proposed maintenance.

MA-4 (5)(a)

Requires the approval of each nonlocal maintenance session by [Assignment: organization-defined personnel or roles]; and

MA-4 (5)(b)

Notifies [Assignment: organization-defined personnel or roles] of the date and time of planned nonlocal maintenance.

MA-4 (6) : CRYPTOGRAPHIC PROTECTION

The information system implements cryptographic mechanisms to protect the integrity and confidentiality of nonlocal maintenance and diagnostic communications.

Related Controls: *SC-8*, *SC-13*

MA-4 (7) : REMOTE DISCONNECT VERIFICATION

The information system implements remote disconnect verification at the termination of nonlocal maintenance and diagnostic sessions.

Note: Remote disconnect verification ensures that remote connections from nonlocal maintenance sessions have been terminated and are no longer available for use.

Related Controls: *SC-13*

Control Family: MAINTENANCE

MA-5 : MAINTENANCE PERSONNEL

Priority: P2

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: This control applies to individuals performing hardware or software maintenance on organizational information systems, while PE-2 addresses physical access for individuals whose maintenance duties place them within the physical protection perimeter of the systems (e.g., custodial staff, physical plant maintenance personnel). Technical competence of supervising individuals relates to the maintenance performed on the information systems while having required access authorizations refers to maintenance on and near the systems. Individuals not previously identified as authorized maintenance personnel, such as information technology manufacturers, vendors, systems integrators, and consultants, may require privileged access to organizational information systems, for example, when required to conduct maintenance activities with little or no notice. Based on organizational assessments of risk, organizations may issue temporary credentials to these individuals. Temporary credentials may be for one-time use or for very limited time periods.

Related Controls: *AC-2*, *IA-8*, *MP-2*, *PE-2*, *PE-3*, *PE-4*, *RA-3*

MA-5a.

Establishes a process for maintenance personnel authorization and maintains a list of authorized maintenance organizations or personnel;

MA-5b.

Ensures that non-escorted personnel performing maintenance on the information system have required access authorizations; and

MA-5c.

Designates organizational personnel with required access authorizations and technical competence to supervise the maintenance activities of personnel who do not possess the required access authorizations.

MA-5 (1) : INDIVIDUALS WITHOUT APPROPRIATE ACCESS

Baseline-Impact: HIGH

The organization:

Note: This control enhancement denies individuals who lack appropriate security clearances (i.e., individuals who do not possess security clearances or possess security clearances at a lower level than required) or who are not U.S. citizens, visual and electronic access to any classified information, Controlled Unclassified Information (CUI), or any other sensitive information contained on organizational information systems. Procedures for the use of maintenance personnel can be documented in security plans for the information systems.

Related Controls: *MP-6*, *PL-2*

MA-5 (1)(a)

Implements procedures for the use of maintenance personnel that lack appropriate security clearances or are not U.S. citizens, that include the following requirements:

MA-5 (1)(a)(1)

Maintenance personnel who do not have needed access authorizations, clearances, or formal access approvals are escorted and supervised during the performance of maintenance and diagnostic activities on the information system by approved organizational personnel who are fully cleared, have appropriate access authorizations, and are technically qualified;

MA-5 (1)(a)(2)

Prior to initiating maintenance or diagnostic activities by personnel who do not have needed access authorizations, clearances or formal access approvals, all volatile information storage components within the information system are sanitized and all nonvolatile storage media are removed or physically disconnected from the system and secured; and

MA-5 (1)(b)

Develops and implements alternate security safeguards in the event an information system component cannot be sanitized, removed, or disconnected from the system.

MA-5 (2) : SECURITY CLEARANCES FOR CLASSIFIED SYSTEMS

The organization ensures that personnel performing maintenance and diagnostic activities on an information system processing, storing, or transmitting classified information possess security clearances and formal access approvals for at least the highest classification level and for all compartments of information on the system.

Related Controls: *PS-3*

MA-5 (3) : CITIZENSHIP REQUIREMENTS FOR CLASSIFIED SYSTEMS

The organization ensures that personnel performing maintenance and diagnostic activities on an information system processing, storing, or transmitting classified information are U.S. citizens.

Related Controls: *PS-3*

MA-5 (4) : FOREIGN NATIONALS

The organization ensures that:

Related Controls: *PS-3*

MA-5 (4)(a)

Cleared foreign nationals (i.e., foreign nationals with appropriate security clearances), are used to conduct maintenance and diagnostic activities on classified information systems only when the systems are jointly owned and operated by the United States and foreign allied governments, or owned and operated solely by foreign allied governments; and

MA-5 (4)(b)

Approvals, consents, and detailed operational conditions regarding the use of foreign nationals to conduct maintenance and diagnostic activities on classified information systems are fully documented within Memoranda of Agreements.

MA-5 (5) : NONSYSTEM-RELATED MAINTENANCE

The organization ensures that non-escorted personnel performing maintenance activities not directly associated with the information system but in the physical proximity of the system, have required access authorizations.

Note: Personnel performing maintenance activities in other capacities not directly related to the information system include, for example, physical plant personnel and janitorial personnel.

Control Family: MAINTENANCE

MA-6 : TIMELY MAINTENANCE

Priority: P2

Baseline-Impact: *MODERATE*, **HIGH**

The organization obtains maintenance support and/or spare parts for [Assignment: organization-defined information system components] within [Assignment: organization-defined time period] of failure.

Note: Organizations specify the information system components that result in increased risk to organizational operations and assets, individuals, other organizations, or the Nation when the functionality provided by those components is not operational. Organizational actions to obtain maintenance support typically include having appropriate contracts in place.

Related Controls: *CM-8*, *CP-2*, *CP-7*, *SA-14*, *SA-15*

MA-6 (1) : PREVENTIVE MAINTENANCE

The organization performs preventive maintenance on [Assignment: organization-defined information system components] at [Assignment: organization-defined time intervals].

Note: Preventive maintenance includes proactive care and servicing of organizational information systems components for the purpose of maintaining equipment and facilities in satisfactory operating condition. Such maintenance provides for the systematic inspection, tests, measurements, adjustments, parts replacement, detection, and correction of incipient failures either before they occur or before they develop into major defects. The primary goal of preventive maintenance is to avoid/mitigate the consequences of equipment failures. Preventive maintenance is designed to preserve and restore equipment reliability by replacing worn components before they actually fail. Methods of determining what preventive (or other) failure management policies to apply include, for example, original equipment manufacturer

(OEM) recommendations, statistical failure records, requirements of codes, legislation, or regulations within a jurisdiction, expert opinion, maintenance that has already been conducted on similar equipment, or measured values and performance indications.

MA-6 (2) : PREDICTIVE MAINTENANCE

> The organization performs predictive maintenance on [Assignment: organization-defined information system components] at [Assignment: organization-defined time intervals].

Note: Predictive maintenance, or condition-based maintenance, attempts to evaluate the condition of equipment by performing periodic or continuous (online) equipment condition monitoring. The goal of predictive maintenance is to perform maintenance at a scheduled point in time when the maintenance activity is most cost-effective and before the equipment loses performance within a threshold. The predictive component of predictive maintenance stems from the goal of predicting the future trend of the equipment's condition. This approach uses principles of statistical process control to determine at what point in the future maintenance activities will be appropriate. Most predictive maintenance inspections are performed while equipment is in service, thereby minimizing disruption of normal system operations. Predictive maintenance can result in substantial cost savings and higher system reliability. Predictive maintenance tends to include measurement of the item. To evaluate equipment condition, predictive maintenance utilizes nondestructive testing technologies such as infrared, acoustic (partial discharge and airborne ultrasonic), corona detection, vibration analysis, sound level measurements, oil analysis, and other specific online tests.

MA-6 (3) : AUTOMATED SUPPORT FOR PREDICTIVE MAINTENANCE

> The organization employs automated mechanisms to transfer predictive maintenance data to a computerized maintenance management system.

Note: A computerized maintenance management system maintains a computer database of information about the maintenance operations of organizations and automates processing equipment condition data in order to trigger maintenance planning, execution, and reporting.

Control Family: MEDIA PROTECTION

MP-1 : MEDIA PROTECTION POLICY AND PROCEDURES

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The organization:

Note: This control addresses the establishment of policy and procedures for the effective implementation of selected security controls and control enhancements in the MP family. Policy and procedures reflect applicable federal laws, Executive Orders, directives, regulations, policies, standards, and guidance. Security program policies and procedures at the organization level may make the need for system-specific policies and procedures unnecessary. The policy can be included as part of the general information security policy for organizations or conversely, can be represented by multiple policies reflecting the complex nature of certain organizations. The procedures can be established for the security program in general and for particular information systems, if needed. The organizational risk management strategy is a key factor in establishing policy and procedures.

Related Controls: *PM-9*

MP-1a.

Develops, documents, and disseminates to [Assignment: organization-defined personnel or roles]:

MP-1a.1.

A media protection policy that addresses purpose, scope, roles, responsibilities, management commitment, coordination among organizational entities, and compliance; and

MP-1a.2.

Procedures to facilitate the implementation of the media protection policy and associated media protection controls; and

MP-1b.

Reviews and updates the current:

MP-1b.1.

Media protection policy [Assignment: organization-defined frequency]; and

MP-1b.2.

Media protection procedures [Assignment: organization-defined frequency].

Control Family: MEDIA PROTECTION

MP-2 : MEDIA ACCESS

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization restricts access to [Assignment: organization-defined types of digital and/or non-digital media] to [Assignment: organization-defined personnel or roles].

Note: Information system media includes both digital and non-digital media. Digital media includes, for example, diskettes, magnetic tapes, external/removable hard disk drives, flash drives, compact disks, and digital video disks. Non-digital media includes, for example, paper and microfilm. Restricting non-digital media access includes, for example, denying access to patient medical records in a community hospital unless the individuals seeking access to such records are authorized healthcare providers. Restricting access to digital media includes, for example, limiting access to design specifications stored on compact disks in the media library to the project leader and the individuals on the development team.

Related Controls: *AC-3*, *IA-2*, *MP-4*, *PE-2*, *PE-3*, *PL-2*

MP-2 (1) : AUTOMATED RESTRICTED ACCESS

[Withdrawn: Incorporated into MP-4 (2)].

MP-2 (2) : CRYPTOGRAPHIC PROTECTION

[Withdrawn: Incorporated into SC-28 (1)].

Control Family: MEDIA PROTECTION

MP-3 : MEDIA MARKING

Priority: P2

Baseline-Impact: *MODERATE*, **HIGH**

The organization:

Note: The term security marking refers to the application/use of human-readable security attributes. The term security labeling refers to the application/use of security attributes with regard to internal data structures within information systems (see AC-16). Information system media includes both digital and non-digital media. Digital media includes, for example, diskettes, magnetic tapes, external/removable hard disk drives, flash drives, compact disks, and digital video disks. Non-digital media includes, for example, paper and microfilm. Security marking is generally not required for media containing information determined by organizations to be in the public domain or to be publicly releasable. However, some organizations may require markings for public information indicating that the information is publicly releasable. Marking of information system media reflects applicable federal laws, Executive Orders, directives, policies, regulations, standards, and guidance.

Related Controls: *AC-16*, *PL-2*, *RA-3*

MP-3a.

Marks information system media indicating the distribution limitations, handling caveats, and applicable security markings (if any) of the information; and

MP-3b.

Exempts [Assignment: organization-defined types of information system media] from marking as long as the media remain within [Assignment: organization-defined controlled areas].

Control Family: MEDIA PROTECTION

MP-4 : MEDIA STORAGE

Priority: P1

Baseline-Impact: *MODERATE*, **HIGH**

The organization:

Note: Information system media includes both digital and non-digital media. Digital media includes, for example, diskettes, magnetic tapes, external/removable hard disk drives, flash drives, compact disks, and digital video disks. Non-digital media includes, for example, paper and microfilm. Physically controlling information system media includes, for example, conducting inventories, ensuring procedures are in place to allow individuals to check out and return media to the media library, and maintaining accountability for all stored media. Secure storage includes, for example, a locked drawer, desk, or cabinet, or a controlled media library. The type of media storage is commensurate with the security category and/or classification of the information residing on the media. Controlled areas are areas for which organizations provide sufficient physical and procedural safeguards to meet the requirements established for protecting information and/or information systems. For media containing information determined by organizations to be in the public domain, to be publicly releasable, or to have limited or no adverse impact on organizations or individuals if accessed by other than authorized personnel, fewer safeguards may be needed. In these situations, physical access controls provide adequate protection.

Related Controls: *CP-6*, *CP-9*, *MP-2*, *MP-7*, *PE-3*

MP-4a.

Physically controls and securely stores [Assignment: organization-defined types of digital and/or non-digital media] within [Assignment: organization-defined controlled areas]; and

MP-4b.

Protects information system media until the media are destroyed or sanitized using approved equipment, techniques, and procedures.

MP-4 (1) : CRYPTOGRAPHIC PROTECTION

[Withdrawn: Incorporated into SC-28 (1)].

MP-4 (2) : AUTOMATED RESTRICTED ACCESS

The organization employs automated mechanisms to restrict access to media storage areas and to audit access attempts and access granted.

Note: Automated mechanisms can include, for example, keypads on the external entries to media storage areas.

Related Controls: *AU-2*, *AU-9*, *AU-6*, *AU-12*

Control Family: MEDIA PROTECTION

MP-5 : MEDIA TRANSPORT

Priority: P1

Baseline-Impact: *MODERATE*, **HIGH**

The organization:

Note:

Information system media includes both digital and non-digital media. Digital media includes, for example, diskettes, magnetic tapes, external/removable hard disk drives, flash drives, compact disks, and digital video disks. Non-digital media includes, for example, paper and microfilm. This control also applies to mobile devices with information storage capability (e.g., smart phones, tablets, E-readers), that are transported outside of controlled areas. Controlled areas are areas or spaces for which organizations provide sufficient physical and/or procedural safeguards to meet the requirements established for protecting information and/or information systems.

Physical and technical safeguards for media are commensurate with the security category or classification of the information residing on the media. Safeguards to protect media during transport include, for example, locked containers and cryptography. Cryptographic mechanisms can provide confidentiality and integrity protections depending upon the mechanisms used. Activities associated with transport include the actual transport as well as those activities such as releasing media for transport and ensuring that media enters the appropriate transport processes. For the actual transport, authorized transport and courier personnel may include individuals from outside the organization (e.g., U.S. Postal Service or a commercial transport or delivery service). Maintaining accountability of media during transport includes, for example, restricting transport activities to authorized personnel, and tracking and/or obtaining explicit records of transport activities as the media moves through the transportation system to prevent and detect loss, destruction, or tampering. Organizations establish documentation requirements for activities associated with the transport of information system media in accordance with organizational assessments of risk to include the flexibility to define different record-keeping methods for the different types of media transport as part of an overall system of transport-related records.

Related Controls: *AC-19*, *CP-9*, *MP-3*, *MP-4*, *RA-3*, *SC-8*, *SC-13*, *SC-28*

MP-5a.

Protects and controls [Assignment: organization-defined types of information system media] during transport outside of controlled areas using [Assignment: organization-defined security safeguards];

MP-5b.

Maintains accountability for information system media during transport outside of controlled areas;

MP-5c.

Documents activities associated with the transport of information system media; and

MP-5d.

Restricts the activities associated with the transport of information system media to authorized personnel.

MP-5 (1) : PROTECTION OUTSIDE OF CONTROLLED AREAS

[Withdrawn: Incorporated into MP-5].

MP-5 (2) : DOCUMENTATION OF ACTIVITIES

[Withdrawn: Incorporated into MP-5].

MP-5 (3) : CUSTODIANS

The organization employs an identified custodian during transport of information system media outside of controlled areas.

Note: Identified custodians provide organizations with specific points of contact during the media transport process and facilitate individual accountability. Custodial responsibilities can be transferred from one individual to another as long as an unambiguous custodian is identified at all times.

MP-5 (4) : CRYPTOGRAPHIC PROTECTION

Baseline-Impact: *MODERATE*, **HIGH**

The information system implements cryptographic mechanisms to protect the confidentiality and integrity of information stored on digital media during transport outside of controlled areas.

Note: This control enhancement applies to both portable storage devices (e.g., USB memory sticks, compact disks, digital video disks, external/removable hard disk drives) and mobile devices with storage capability (e.g., smart phones, tablets, E-readers).

Related Controls: *MP-2*

Control Family: MEDIA PROTECTION

MP-6 : MEDIA SANITIZATION

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: This control applies to all information system media, both digital and non-digital, subject to disposal or reuse, whether or not the media is considered removable. Examples include media found in scanners, copiers, printers, notebook computers, workstations, network components, and mobile devices. The sanitization process removes information from the media such that the information cannot be retrieved or reconstructed. Sanitization techniques, including clearing, purging, cryptographic erase, and destruction, prevent the disclosure of information to unauthorized individuals when such media is reused or released for disposal. Organizations determine the appropriate sanitization methods recognizing that destruction is sometimes necessary when other methods cannot be applied to media requiring sanitization. Organizations use discretion on the employment of approved sanitization techniques and procedures for media containing information deemed to be in the public domain or publicly releasable, or deemed to have no adverse impact on organizations or individuals if released for reuse or disposal. Sanitization of non-digital media includes, for example, removing a classified appendix from an otherwise unclassified document, or redacting selected sections or words from a document by obscuring the redacted sections/words in a manner equivalent in effectiveness to removing them from the document. NSA standards and policies control the sanitization process for media containing classified information.

Related Controls: *MA-2*, *MA-4*, *RA-3*, *SC-4*

MP-6a.

Sanitizes [Assignment: organization-defined information system media] prior to disposal, release out of organizational control, or release for reuse using [Assignment: organization-defined sanitization techniques and procedures] in accordance with applicable federal and organizational standards and policies; and

MP-6b.

Employs sanitization mechanisms with the strength and integrity commensurate with the security category or classification of the information.

MP-6 (1) : REVIEW / APPROVE / TRACK / DOCUMENT / VERIFY

Baseline-Impact: HIGH

The organization reviews, approves, tracks, documents, and verifies media sanitization and disposal actions.

Note: Organizations review and approve media to be sanitized to ensure compliance with records-retention policies. Tracking/documenting actions include, for example, listing personnel who reviewed and approved sanitization and disposal actions, types of media sanitized, specific files stored on the media, sanitization methods used, date and time of the sanitization actions, personnel who performed the sanitization, verification actions taken, personnel who performed the verification, and disposal action taken. Organizations verify that the sanitization of the media was effective prior to disposal.

Related Controls: *SI-12*

MP-6 (2) : EQUIPMENT TESTING

Baseline-Impact: HIGH

> The organization tests sanitization equipment and procedures [Assignment: organization-defined frequency] to verify that the intended sanitization is being achieved.

Note: Testing of sanitization equipment and procedures may be conducted by qualified and authorized external entities (e.g., other federal agencies or external service providers).

MP-6 (3) : NONDESTRUCTIVE TECHNIQUES

Baseline-Impact: HIGH

> The organization applies nondestructive sanitization techniques to portable storage devices prior to connecting such devices to the information system under the following circumstances: [Assignment: organization-defined circumstances requiring sanitization of portable storage devices].

Note: This control enhancement applies to digital media containing classified information and Controlled Unclassified Information (CUI). Portable storage devices can be the source of malicious code insertions into organizational information systems. Many of these devices are obtained from unknown and potentially untrustworthy sources and may contain malicious code that can be readily transferred to information systems through USB ports or other entry portals. While scanning such storage devices is always recommended, sanitization provides additional assurance that the devices are free of malicious code to include code capable of initiating zero-day attacks. Organizations consider nondestructive sanitization of portable storage devices when such devices are first purchased from the manufacturer or vendor prior to initial use or when organizations lose a positive chain of custody for the devices.

Related Controls: *SI-3*

MP-6 (4) : CONTROLLED UNCLASSIFIED INFORMATION

> [Withdrawn: Incorporated into MP-6].

MP-6 (5) : CLASSIFIED INFORMATION

[Withdrawn: Incorporated into MP-6].

MP-6 (6) : MEDIA DESTRUCTION

[Withdrawn: Incorporated into MP-6].

MP-6 (7) : DUAL AUTHORIZATION

The organization enforces dual authorization for the sanitization of [Assignment: organization-defined information system media].

Note: Organizations employ dual authorization to ensure that information system media sanitization cannot occur unless two technically qualified individuals conduct the task. Individuals sanitizing information system media possess sufficient skills/expertise to determine if the proposed sanitization reflects applicable federal/organizational standards, policies, and procedures. Dual authorization also helps to ensure that sanitization occurs as intended, both protecting against errors and false claims of having performed the sanitization actions. Dual authorization may also be known as two-person control.

Related Controls: *AC-3*, *MP-2*

MP-6 (8) : REMOTE PURGING / WIPING OF INFORMATION

The organization provides the capability to purge/wipe information from [Assignment: organization-defined information systems, system components, or devices] either remotely or under the following conditions: [Assignment: organization-defined conditions].

Note: This control enhancement protects data/information on organizational information systems, system components, or devices (e.g., mobile devices) if such systems, components, or devices are obtained by unauthorized individuals. Remote purge/wipe commands require strong authentication to mitigate the risk of unauthorized individuals purging/wiping the system/component/device. The purge/wipe function can be implemented in a variety of ways including, for example, by overwriting data/information multiple times or by destroying the key necessary to decrypt encrypted data.

Control Family: MEDIA PROTECTION

MP-7 : MEDIA USE

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization [Selection: restricts; prohibits] the use of [Assignment: organization-defined types of information system media] on [Assignment: organization-defined information systems or system components] using [Assignment: organization-defined security safeguards].

Note: Information system media includes both digital and non-digital media. Digital media includes, for example, diskettes, magnetic tapes, external/removable hard disk drives, flash drives, compact disks, and digital video disks. Non-digital media includes, for example, paper and microfilm. This control also applies to mobile devices with information storage capability (e.g., smart phones, tablets, E-readers). In contrast to MP-2, which restricts user access to media, this control restricts the use of certain types of media on information systems, for example, restricting/prohibiting the use of flash drives or external hard disk drives. Organizations can employ technical and nontechnical safeguards (e.g., policies, procedures, rules of behavior) to restrict the use of information system media. Organizations may restrict the use of portable storage devices, for example, by using physical cages on workstations to prohibit access to certain external ports, or disabling/removing the ability to insert, read or write to such devices. Organizations may also limit the use of portable storage devices to only approved devices including, for example, devices provided by the organization, devices provided by other approved organizations, and devices that are not personally owned. Finally, organizations may restrict the use of portable storage devices based on the type of device, for example, prohibiting the use of writeable, portable storage devices, and implementing this restriction by disabling or removing the capability to write to such devices.

Related Controls: *AC-19*, *PL-4*

MP-7 (1) : PROHIBIT USE WITHOUT OWNER

Baseline-Impact: *MODERATE*, **HIGH**

> The organization prohibits the use of portable storage devices in organizational information systems when such devices have no identifiable owner.

Note: Requiring identifiable owners (e.g., individuals, organizations, or projects) for portable storage devices reduces the risk of using such technologies by allowing organizations to assign responsibility and accountability for addressing known vulnerabilities in the devices (e.g., malicious code insertion).

Related Controls: *PL-4*

MP-7 (2) : PROHIBIT USE OF SANITIZATION-RESISTANT MEDIA

> The organization prohibits the use of sanitization-resistant media in organizational information systems.

Note: Sanitization-resistance applies to the capability to purge information from media. Certain types of media do not support sanitize commands, or if supported, the interfaces are not supported in a standardized way across these devices. Sanitization-resistant media include, for example, compact flash, embedded flash on boards and devices, solid state drives, and USB removable media.

Related Controls: *MP-6*

Control Family: MEDIA PROTECTION

MP-8 : MEDIA DOWNGRADING

Priority: P0

The organization:

Note: This control applies to all information system media, digital and non-digital, subject to release outside of the organization, whether or not the media is considered removable. The downgrading process, when applied to system media, removes information from the media, typically by security category or classification level, such that the information cannot be retrieved or reconstructed. Downgrading of media includes redacting information to enable wider release and distribution. Downgrading of media also ensures that empty space on the media (e.g., slack space within files) is devoid of information.

MP-8a.

Establishes [Assignment: organization-defined information system media downgrading process] that includes employing downgrading mechanisms with [Assignment: organization-defined strength and integrity];

MP-8b.

Ensures that the information system media downgrading process is commensurate with the security category and/or classification level of the information to be removed and the access authorizations of the potential recipients of the downgraded information;

MP-8c.

Identifies [Assignment: organization-defined information system media requiring downgrading]; and

MP-8d.

Downgrades the identified information system media using the established process.

MP-8 (1) : DOCUMENTATION OF PROCESS

The organization documents information system media downgrading actions.

Note: Organizations can document the media downgrading process by providing information such as the downgrading technique employed, the identification number of the downgraded media, and the identity of the individual that authorized and/or performed the downgrading action.

MP-8 (2) : EQUIPMENT TESTING

The organization employs [Assignment: organization-defined tests] of downgrading equipment and procedures to verify correct performance [Assignment: organization-defined frequency].

MP-8 (3) : CONTROLLED UNCLASSIFIED INFORMATION

The organization downgrades information system media containing [Assignment: organization-defined Controlled Unclassified Information (CUI)] prior to public release in accordance with applicable federal and organizational standards and policies.

MP-8 (4) : CLASSIFIED INFORMATION

The organization downgrades information system media containing classified information prior to release to individuals without required access authorizations in accordance with NSA standards and policies.

Note: Downgrading of classified information uses approved sanitization tools, techniques, and procedures to transfer information confirmed to be unclassified from classified information systems to unclassified media.

Control Family: PHYSICAL AND ENVIRONMENTAL PROTECTION

PE-1 : PHYSICAL AND ENVIRONMENTAL PROTECTION POLICY AND PROCEDURES

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: This control addresses the establishment of policy and procedures for the effective implementation of selected security controls and control enhancements in the PE family. Policy and procedures reflect applicable federal laws, Executive Orders, directives, regulations, policies, standards, and guidance. Security program policies and procedures at the organization level may make the need for system-specific policies and procedures unnecessary. The policy can be included as part of the general information security policy for organizations or conversely, can be represented by multiple policies reflecting the complex nature of certain organizations. The procedures can be established for the security program in general and for particular information systems, if needed. The organizational risk management strategy is a key factor in establishing policy and procedures.

Related Controls: *PM-9*

PE-1a.

Develops, documents, and disseminates to [Assignment: organization-defined personnel or roles]:

PE-1a.1.

A physical and environmental protection policy that addresses purpose, scope, roles, responsibilities, management commitment, coordination among organizational entities, and compliance; and

PE-1a.2.

Procedures to facilitate the implementation of the physical and environmental protection policy and associated physical and environmental protection controls; and

PE-1b.

Reviews and updates the current:

PE-1b.1.

Physical and environmental protection policy [Assignment: organization-defined frequency]; and

PE-1b.2.

Physical and environmental protection procedures [Assignment: organization-defined frequency].

Control Family: PHYSICAL AND ENVIRONMENTAL PROTECTION

PE-2 : PHYSICAL ACCESS AUTHORIZATIONS

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: This control applies to organizational employees and visitors. Individuals (e.g., employees, contractors, and others) with permanent physical access authorization credentials are not considered visitors. Authorization credentials include, for example, badges, identification cards, and smart cards. Organizations determine the strength of authorization credentials needed (including level of forge-proof badges, smart cards, or identification cards) consistent with federal standards, policies, and procedures. This control only applies to areas within facilities that have not been designated as publicly accessible.

Related Controls: *PE-3*, *PE-4*, *PS-3*

PE-2a.

Develops, approves, and maintains a list of individuals with authorized access to the facility where the information system resides;

PE-2b.

Issues authorization credentials for facility access;

PE-2c.

Reviews the access list detailing authorized facility access by individuals [Assignment: organization-defined frequency]; and

PE-2d.

Removes individuals from the facility access list when access is no longer required.

PE-2 (1) : ACCESS BY POSITION / ROLE

The organization authorizes physical access to the facility where the information system resides based on position or role.

Related Controls: *AC-2*, *AC-3*, *AC-6*

PE-2 (2) : TWO FORMS OF IDENTIFICATION

The organization requires two forms of identification from [Assignment: organization-defined list of acceptable forms of identification] for visitor access to the facility where the information system resides.

Note: Acceptable forms of government photo identification include, for example, passports, Personal Identity Verification (PIV) cards, and drivers licenses. In the case of gaining access to facilities using automated mechanisms, organizations may use PIV cards, key cards, PINs, and biometrics.

Related Controls: *IA-2*, *IA-4*, *IA-5*

PE-2 (3) : RESTRICT UNESCORTED ACCESS

The organization restricts unescorted access to the facility where the information system resides to personnel with [Selection (one or more): security clearances for all information contained within the system; formal access authorizations for all information contained within the system; need for access to all information contained within the system; [Assignment: organization-defined credentials]].

Note: Due to the highly sensitive nature of classified information stored within certain facilities, it is important that individuals lacking sufficient security clearances, access approvals, or need to know, be escorted by individuals with appropriate credentials to ensure that such information is not exposed or otherwise compromised.

Related Controls: *PS-2*, *PS-6*

Control Family: PHYSICAL AND ENVIRONMENTAL PROTECTION

PE-3 : PHYSICAL ACCESS CONTROL

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: This control applies to organizational employees and visitors. Individuals (e.g., employees, contractors, and others) with permanent physical access authorization credentials are not considered visitors. Organizations determine the types of facility guards needed including, for example, professional physical security staff or other personnel such as administrative staff or information system users. Physical access devices include, for example, keys, locks, combinations, and card readers. Safeguards for publicly accessible areas within organizational facilities include, for example, cameras, monitoring by guards, and isolating selected information systems and/or system components in secured areas. Physical access control systems comply with applicable federal laws, Executive Orders, directives, policies, regulations, standards, and guidance. The Federal Identity, Credential, and Access Management Program provides implementation guidance for identity, credential, and access management capabilities for physical access control systems. Organizations have flexibility in the types of audit logs employed. Audit logs can be procedural (e.g., a written log of individuals accessing the facility and when such access occurred), automated (e.g., capturing ID provided by a PIV card), or some combination thereof. Physical access points can include facility access points, interior access points to information systems and/or components requiring supplemental access controls, or both. Components of organizational information systems (e.g., workstations, terminals) may be located in areas designated as publicly accessible with organizations safeguarding access to such devices.

Related Controls: *AU-2*, *AU-6*, *MP-2*, *MP-4*, *PE-2*, *PE-4*, *PE-5*, *PS-3*, *RA-3*

PE-3a.

Enforces physical access authorizations at [Assignment: organization-defined entry/exit points to the facility where the information system resides] by;

PE-3a.1.

Verifying individual access authorizations before granting access to the facility; and

PE-3a.2.

Controlling ingress/egress to the facility using [Selection (one or more): [Assignment: organization-defined physical access control systems/devices]; guards];

PE-3b.

Maintains physical access audit logs for [Assignment: organization-defined entry/exit points];

PE-3c.

Provides [Assignment: organization-defined security safeguards] to control access to areas within the facility officially designated as publicly accessible;

PE-3d.

Escorts visitors and monitors visitor activity [Assignment: organization-defined circumstances requiring visitor escorts and monitoring];

PE-3e.

Secures keys, combinations, and other physical access devices;

PE-3f.

Inventories [Assignment: organization-defined physical access devices] every [Assignment: organization-defined frequency]; and

PE-3g.

Changes combinations and keys [Assignment: organization-defined frequency] and/or when keys are lost, combinations are compromised, or individuals are transferred or terminated.

PE-3 (1) : INFORMATION SYSTEM ACCESS

Baseline-Impact: HIGH

The organization enforces physical access authorizations to the information system in addition to the physical access controls for the facility at [Assignment: organization-defined physical spaces containing one or more components of the information system].

Note: This control enhancement provides additional physical security for those areas within facilities where there is a concentration of information system components (e.g., server rooms, media storage areas, data and communications centers).

Related Controls: *PS-2*

PE-3 (2) : FACILITY / INFORMATION SYSTEM BOUNDARIES

The organization performs security checks [Assignment: organization-defined frequency] at the physical boundary of the facility or information system for unauthorized exfiltration of information or removal of information system components.

Note: Organizations determine the extent, frequency, and/or randomness of security checks to adequately mitigate risk associated with exfiltration.

Related Controls: *AC-4*, *SC-7*

PE-3 (3) : CONTINUOUS GUARDS / ALARMS / MONITORING

The organization employs guards and/or alarms to monitor every physical access point to the facility where the information system resides 24 hours per day, 7 days per week.

Related Controls: *CP-6*, *CP-7*

PE-3 (4) : LOCKABLE CASINGS

The organization uses lockable physical casings to protect [Assignment: organization-defined information system components] from unauthorized physical access.

PE-3 (5) : TAMPER PROTECTION

The organization employs [Assignment: organization-defined security safeguards] to [Selection (one or more): detect; prevent] physical tampering or alteration of [Assignment: organization-defined hardware components] within the information system.

Note: Organizations may implement tamper detection/prevention at selected hardware components or tamper detection at some components and tamper prevention at other components. Tamper detection/prevention activities can employ many types of anti-tamper technologies including, for example, tamper-detection seals and anti-tamper coatings. Anti-tamper programs help to detect hardware alterations through counterfeiting and other supply chain-related risks.

Related Controls: *SA-12*

PE-3 (6) : FACILITY PENETRATION TESTING

The organization employs a penetration testing process that includes [Assignment: organization-defined frequency], unannounced attempts to bypass or circumvent security controls associated with physical access points to the facility.

Related Controls: *CA-2*, *CA-7*

Control Family: PHYSICAL AND ENVIRONMENTAL PROTECTION

PE-4 : ACCESS CONTROL FOR TRANSMISSION MEDIUM

Priority: P1

Baseline-Impact: *MODERATE*, **HIGH**

> The organization controls physical access to [Assignment: organization-defined information system distribution and transmission lines] within organizational facilities using [Assignment: organization-defined security safeguards].

Note: Physical security safeguards applied to information system distribution and transmission lines help to prevent accidental damage, disruption, and physical tampering. In addition, physical safeguards may be necessary to help prevent eavesdropping or in transit modification of unencrypted transmissions. Security safeguards to control physical access to system distribution and transmission lines include, for example: (i) locked wiring closets; (ii) disconnected or locked spare jacks; and/or (iii) protection of cabling by conduit or cable trays.

Related Controls: *MP-2*, *MP-4*, *PE-2*, *PE-3*, *PE-5*, *SC-7*, *SC-8*

Control Family: PHYSICAL AND ENVIRONMENTAL PROTECTION

PE-5 : ACCESS CONTROL FOR OUTPUT DEVICES

Priority: P2

Baseline-Impact: *MODERATE*, **HIGH**

> The organization controls physical access to information system output devices to prevent unauthorized individuals from obtaining the output.

Note: Controlling physical access to output devices includes, for example, placing output devices in locked rooms or other secured areas and allowing access to authorized individuals only, and placing output devices in locations that can be monitored by organizational personnel. Monitors, printers, copiers, scanners, facsimile machines, and audio devices are examples of information system output devices.

Related Controls: *PE-2*, *PE-3*, *PE-4*, *PE-18*

PE-5 (1) : ACCESS TO OUTPUT BY AUTHORIZED INDIVIDUALS

> The organization:

Note: Controlling physical access to selected output devices includes, for example, placing printers, copiers, and facsimile machines in controlled areas with keypad access controls or limiting access to individuals with certain types of badges.

PE-5 (1)(a)

Controls physical access to output from [Assignment: organization-defined output devices]; and

PE-5 (1)(b)

Ensures that only authorized individuals receive output from the device.

PE-5 (2) : ACCESS TO OUTPUT BY INDIVIDUAL IDENTITY

The information system:

Note: Controlling physical access to selected output devices includes, for example, installing security functionality on printers, copiers, and facsimile machines that allows organizations to implement authentication (e.g., using a PIN or hardware token) on output devices prior to the release of output to individuals.

PE-5 (2)(a)

Controls physical access to output from [Assignment: organization-defined output devices]; and

PE-5 (2)(b)

Links individual identity to receipt of the output from the device.

PE-5 (3) : MARKING OUTPUT DEVICES

The organization marks [Assignment: organization-defined information system output devices] indicating the appropriate security marking of the information permitted to be output from the device.

Note: Outputs devices include, for example, printers, monitors, facsimile machines, scanners, copiers, and audio devices. This control enhancement is generally applicable to information system output devices other than mobiles devices.

Control Family: PHYSICAL AND ENVIRONMENTAL PROTECTION

PE-6 : MONITORING PHYSICAL ACCESS

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: Organizational incident response capabilities include investigations of and responses to detected physical security incidents. Security incidents include, for example, apparent security violations or suspicious physical access activities. Suspicious physical access activities include, for example: (i) accesses outside of normal work hours; (ii) repeated accesses to areas not normally accessed; (iii) accesses for unusual lengths of time; and (iv) out-of-sequence accesses.

Related Controls: *CA-7*, *IR-4*, *IR-8*

PE-6a.

Monitors physical access to the facility where the information system resides to detect and respond to physical security incidents;

PE-6b.

Reviews physical access logs [Assignment: organization-defined frequency] and upon occurrence of [Assignment: organization-defined events or potential indications of events]; and

PE-6c.

Coordinates results of reviews and investigations with the organizational incident response capability.

PE-6 (1) : INTRUSION ALARMS / SURVEILLANCE EQUIPMENT

Baseline-Impact: *MODERATE*, **HIGH**

The organization monitors physical intrusion alarms and surveillance equipment.

PE-6 (2) : AUTOMATED INTRUSION RECOGNITION / RESPONSES

The organization employs automated mechanisms to recognize [Assignment: organization-defined classes/types of intrusions] and initiate [Assignment: organization-defined response actions].

Related Controls: *SI-4*

PE-6 (3) : VIDEO SURVEILLANCE

The organization employs video surveillance of [Assignment: organization-defined operational areas] and retains video recordings for [Assignment: organization-defined time period].

Note: This control enhancement focuses on recording surveillance video for purposes of subsequent review, if circumstances so warrant (e.g., a break-in detected by other means). It does not require monitoring surveillance video although organizations may choose to do so. Note that there may be legal considerations when performing and retaining video surveillance, especially if such surveillance is in a public location.

PE-6 (4) : MONITORING PHYSICAL ACCESS TO INFORMATION SYSTEMS

Baseline-Impact: **HIGH**

The organization monitors physical access to the information system in addition to the physical access monitoring of the facility as [Assignment: organization-defined physical spaces containing one or more components of the information system].

Note: This control enhancement provides additional monitoring for those areas within facilities where there is a concentration of information system components (e.g., server rooms, media storage areas, communications centers).

Related Controls: *PS-2*, *PS-3*

Control Family: PHYSICAL AND ENVIRONMENTAL PROTECTION

PE-7 : VISITOR CONTROL

[Withdrawn: Incorporated into PE-2 and PE-3].

Control Family: PHYSICAL AND ENVIRONMENTAL PROTECTION

PE-8 : VISITOR ACCESS RECORDS

Priority: P3

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: Visitor access records include, for example, names and organizations of persons visiting, visitor signatures, forms of identification, dates of access, entry and departure times, purposes of visits, and names and organizations of persons visited. Visitor access records are not required for publicly accessible areas.

PE-8a.

Maintains visitor access records to the facility where the information system resides for [Assignment: organization-defined time period]; and

PE-8b.

Reviews visitor access records [Assignment: organization-defined frequency].

PE-8 (1) : AUTOMATED RECORDS MAINTENANCE / REVIEW

Baseline-Impact: **HIGH**

The organization employs automated mechanisms to facilitate the maintenance and review of visitor access records.

PE-8 (2) : PHYSICAL ACCESS RECORDS

[Withdrawn: Incorporated into PE-2].

Control Family: PHYSICAL AND ENVIRONMENTAL PROTECTION

PE-9 : POWER EQUIPMENT AND CABLING

Priority: P1

Baseline-Impact: *MODERATE*, **HIGH**

The organization protects power equipment and power cabling for the information system from damage and destruction.

Note: Organizations determine the types of protection necessary for power equipment and cabling employed at different locations both internal and external to organizational facilities and environments of operation. This includes, for example, generators and power cabling outside of buildings, internal cabling and uninterruptable power sources within an office or data center, and power sources for self-contained entities such as vehicles and satellites.

Related Controls: *PE-4*

PE-9 (1) : REDUNDANT CABLING

The organization employs redundant power cabling paths that are physically separated by [Assignment: organization-defined distance].

Note: Physically separate, redundant power cables help to ensure that power continues to flow in the event one of the cables is cut or otherwise damaged.

PE-9 (2) : AUTOMATIC VOLTAGE CONTROLS

The organization employs automatic voltage controls for [Assignment: organization-defined critical information system components].

Control Family: PHYSICAL AND ENVIRONMENTAL PROTECTION

PE-10 : EMERGENCY SHUTOFF

Priority: P1

Baseline-Impact: *MODERATE*, **HIGH**

The organization:

Note: This control applies primarily to facilities containing concentrations of information system resources including, for example, data centers, server rooms, and mainframe computer rooms.

Related Controls: *PE-15*

PE-10a.

Provides the capability of shutting off power to the information system or individual system components in emergency situations;

PE-10b.

Places emergency shutoff switches or devices in [Assignment: organization-defined location by information system or system component] to facilitate safe and easy access for personnel; and

PE-10c.

Protects emergency power shutoff capability from unauthorized activation.

PE-10 (1) : ACCIDENTAL / UNAUTHORIZED ACTIVATION

[Withdrawn: Incorporated into PE-10].

Control Family: PHYSICAL AND ENVIRONMENTAL PROTECTION

PE-11 : EMERGENCY POWER

Priority: P1

Baseline-Impact: *MODERATE*, **HIGH**

The organization provides a short-term uninterruptible power supply to facilitate [Selection (one or more): an orderly shutdown of the information system; transition of the information system to long-term alternate power] in the event of a primary power source loss.

Related Controls: *AT-3*, *CP-2*, *CP-7*

PE-11 (1) : LONG-TERM ALTERNATE POWER SUPPLY - MINIMAL OPERATIONAL CAPABILITY

Baseline-Impact: **HIGH**

> The organization provides a long-term alternate power supply for the information system that is capable of maintaining minimally required operational capability in the event of an extended loss of the primary power source.

Note: This control enhancement can be satisfied, for example, by the use of a secondary commercial power supply or other external power supply. Long-term alternate power supplies for the information system can be either manually or automatically activated.

PE-11 (2) : LONG-TERM ALTERNATE POWER SUPPLY - SELF-CONTAINED

> The organization provides a long-term alternate power supply for the information system that is:

Note: This control enhancement can be satisfied, for example, by the use of one or more generators with sufficient capacity to meet the needs of the organization. Long-term alternate power supplies for organizational information systems are either manually or automatically activated.

PE-11 (2)(a)

> Self-contained;

PE-11 (2)(b)

> Not reliant on external power generation; and

PE-11 (2)(c)

> Capable of maintaining [Selection: minimally required operational capability; full operational capability] in the event of an extended loss of the primary power source.

Control Family: PHYSICAL AND ENVIRONMENTAL PROTECTION

PE-12 : EMERGENCY LIGHTING

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The organization employs and maintains automatic emergency lighting for the information system that activates in the event of a power outage or disruption and that covers emergency exits and evacuation routes within the facility.

Note: This control applies primarily to facilities containing concentrations of information system resources including, for example, data centers, server rooms, and mainframe computer rooms.

Related Controls: *CP-2*, *CP-7*

PE-12 (1) : ESSENTIAL MISSIONS / BUSINESS FUNCTIONS

The organization provides emergency lighting for all areas within the facility supporting essential missions and business functions.

Control Family: PHYSICAL AND ENVIRONMENTAL PROTECTION

PE-13 : FIRE PROTECTION

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization employs and maintains fire suppression and detection devices/systems for the information system that are supported by an independent energy source.

Note: This control applies primarily to facilities containing concentrations of information system resources including, for example, data centers, server rooms, and mainframe computer rooms. Fire suppression and detection devices/systems include, for example, sprinkler systems, handheld fire extinguishers, fixed fire hoses, and smoke detectors.

PE-13 (1) : DETECTION DEVICES / SYSTEMS

Baseline-Impact: HIGH

The organization employs fire detection devices/systems for the information system that activate automatically and notify [Assignment: organization-defined personnel or roles] and [Assignment: organization-defined emergency responders] in the event of a fire.

Note: Organizations can identify specific personnel, roles, and emergency responders in the event that individuals on the notification list must have appropriate access authorizations and/or clearances, for example, to obtain access to facilities where classified operations are taking place or where there are information systems containing classified information.

PE-13 (2) : SUPPRESSION DEVICES / SYSTEMS

Baseline-Impact: **HIGH**

> The organization employs fire suppression devices/systems for the information system that provide automatic notification of any activation to Assignment: organization-defined personnel or roles] and [Assignment: organization-defined emergency responders].

Note: Organizations can identify specific personnel, roles, and emergency responders in the event that individuals on the notification list must have appropriate access authorizations and/or clearances, for example, to obtain access to facilities where classified operations are taking place or where there are information systems containing classified information.

PE-13 (3) : AUTOMATIC FIRE SUPPRESSION

Baseline-Impact: *MODERATE*, **HIGH**

> The organization employs an automatic fire suppression capability for the information system when the facility is not staffed on a continuous basis.

PE-13 (4) : INSPECTIONS

> The organization ensures that the facility undergoes [Assignment: organization-defined frequency] inspections by authorized and qualified inspectors and resolves identified deficiencies within [Assignment: organization-defined time period].

Control Family: PHYSICAL AND ENVIRONMENTAL PROTECTION

PE-14 : TEMPERATURE AND HUMIDITY CONTROLS

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The organization:

Note: This control applies primarily to facilities containing concentrations of information system resources, for example, data centers, server rooms, and mainframe computer rooms.

Related Controls: *AT-3*

PE-14a.

Maintains temperature and humidity levels within the facility where the information system resides at [Assignment: organization-defined acceptable levels]; and

PE-14b.

Monitors temperature and humidity levels [Assignment: organization-defined frequency].

PE-14 (1) : AUTOMATIC CONTROLS

The organization employs automatic temperature and humidity controls in the facility to prevent fluctuations potentially harmful to the information system.

PE-14 (2) : MONITORING WITH ALARMS / NOTIFICATIONS

The organization employs temperature and humidity monitoring that provides an alarm or notification of changes potentially harmful to personnel or equipment.

Control Family: PHYSICAL AND ENVIRONMENTAL PROTECTION

PE-15 : WATER DAMAGE PROTECTION

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization protects the information system from damage resulting from water leakage by providing master shutoff or isolation valves that are accessible, working properly, and known to key personnel.

Note: This control applies primarily to facilities containing concentrations of information system resources including, for example, data centers, server rooms, and mainframe computer rooms. Isolation valves can be employed in addition to or in lieu of master shutoff valves to shut off water supplies in specific areas of concern, without affecting entire organizations.

Related Controls: *AT-3*

PE-15 (1) : AUTOMATION SUPPORT

Baseline-Impact: HIGH

The organization employs automated mechanisms to detect the presence of water in the vicinity of the information system and alerts [Assignment: organization-defined personnel or roles].

Note: Automated mechanisms can include, for example, water detection sensors, alarms, and notification systems.

Control Family: PHYSICAL AND ENVIRONMENTAL PROTECTION

PE-16 : DELIVERY AND REMOVAL

Priority: P2

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The organization authorizes, monitors, and controls [Assignment: organization-defined types of information system components] entering and exiting the facility and maintains records of those items.

Note: Effectively enforcing authorizations for entry and exit of information system components may require restricting access to delivery areas and possibly isolating the areas from the information system and media libraries.

Related Controls: *CM-3*, *MA-2*, *MA-3*, *MP-5*, *SA-12*

Control Family: PHYSICAL AND ENVIRONMENTAL PROTECTION

PE-17 : ALTERNATE WORK SITE

Priority: P2

Baseline-Impact: *MODERATE*, **HIGH**

> The organization:

Note: Alternate work sites may include, for example, government facilities or private residences of employees. While commonly distinct from alternative processing sites, alternate work sites may provide readily available alternate locations as part of contingency operations. Organizations may define different sets of security controls for specific alternate work sites or types of sites depending on the work-related activities conducted at those sites. This control supports the contingency planning activities of organizations and the federal telework initiative.

Related Controls: *AC-17*, *CP-7*

PE-17a.

> Employs [Assignment: organization-defined security controls] at alternate work sites;

PE-17b.

> Assesses as feasible, the effectiveness of security controls at alternate work sites; and

PE-17c.

Provides a means for employees to communicate with information security personnel in case of security incidents or problems.

Control Family: PHYSICAL AND ENVIRONMENTAL PROTECTION

PE-18 : LOCATION OF INFORMATION SYSTEM COMPONENTS

Priority: P3

Baseline-Impact: HIGH

The organization positions information system components within the facility to minimize potential damage from [Assignment: organization-defined physical and environmental hazards] and to minimize the opportunity for unauthorized access.

Note: Physical and environmental hazards include, for example, flooding, fire, tornadoes, earthquakes, hurricanes, acts of terrorism, vandalism, electromagnetic pulse, electrical interference, and other forms of incoming electromagnetic radiation. In addition, organizations consider the location of physical entry points where unauthorized individuals, while not being granted access, might nonetheless be in close proximity to information systems and therefore increase the potential for unauthorized access to organizational communications (e.g., through the use of wireless sniffers or microphones).

Related Controls: *CP-2*, *PE-19*, *RA-3*

PE-18 (1) : FACILITY SITE

The organization plans the location or site of the facility where the information system resides with regard to physical and environmental hazards and for existing facilities, considers the physical and environmental hazards in its risk mitigation strategy.

Related Controls: *PM-8*

Control Family: PHYSICAL AND ENVIRONMENTAL PROTECTION

PE-19 : INFORMATION LEAKAGE

Priority: P0

The organization protects the information system from information leakage due to electromagnetic signals emanations.

Note: Information leakage is the intentional or unintentional release of information to an untrusted environment from electromagnetic signals emanations. Security categories or classifications of information systems (with respect to confidentiality) and organizational security policies guide the selection of security controls employed to protect systems against information leakage due to electromagnetic signals emanations.

PE-19 (1) : NATIONAL EMISSIONS / TEMPEST POLICIES AND PROCEDURES

The organization ensures that information system components, associated data communications, and networks are protected in accordance with national emissions and TEMPEST policies and procedures based on the security category or classification of the information.

Control Family: PHYSICAL AND ENVIRONMENTAL PROTECTION

PE-20 : ASSET MONITORING AND TRACKING

Priority: P0

The organization:

Note: Asset location technologies can help organizations ensure that critical assets such as vehicles or essential information system components remain in authorized locations. Organizations consult with the Office of the General Counsel and the Senior Agency Official for Privacy (SAOP)/Chief Privacy Officer (CPO) regarding the deployment and use of asset location technologies to address potential privacy concerns.

Related Controls: *CM-8*

PE-20a.

Employs [Assignment: organization-defined asset location technologies] to track and monitor the location and movement of [Assignment: organization-defined assets] within [Assignment: organization-defined controlled areas]; and

PE-20b.

Ensures that asset location technologies are employed in accordance with applicable federal laws, Executive Orders, directives, regulations, policies, standards, and guidance.

Control Family: PLANNING

PL-1 : SECURITY PLANNING POLICY AND PROCEDURES

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: This control addresses the establishment of policy and procedures for the effective implementation of selected security controls and control enhancements in the PL family. Policy and procedures reflect applicable federal laws, Executive Orders, directives, regulations, policies, standards, and guidance. Security program policies and procedures at the organization level may make the need for system-specific policies and procedures unnecessary. The policy can be included as part of the general information security policy for organizations or conversely, can be represented by multiple policies reflecting the complex nature of certain organizations. The procedures can be established for the security program in general and for particular information systems, if needed. The organizational risk management strategy is a key factor in establishing policy and procedures.

Related Controls: *PM-9*

PL-1a.

Develops, documents, and disseminates to [Assignment: organization-defined personnel or roles]:

PL-1a.1.

A security planning policy that addresses purpose, scope, roles, responsibilities, management commitment, coordination among organizational entities, and compliance; and

PL-1a.2.

Procedures to facilitate the implementation of the security planning policy and associated security planning controls; and

PL-1b.

Reviews and updates the current:

PL-1b.1.

Security planning policy [Assignment: organization-defined frequency]; and

PL-1b.2.

Security planning procedures [Assignment: organization-defined frequency].

Control Family: PLANNING

PL-2 : SYSTEM SECURITY PLAN

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note:

Security plans relate security requirements to a set of security controls and control enhancements. Security plans also describe, at a high level, how the security controls and control enhancements meet those security requirements, but do not provide detailed, technical descriptions of the specific design or implementation of the controls/enhancements. Security plans contain sufficient information (including the specification of parameter values for assignment and selection statements either explicitly or by reference) to enable a design and implementation that is unambiguously compliant with the intent of the plans and subsequent determinations of risk to organizational operations and assets, individuals, other organizations, and the Nation if the plan is implemented as intended. Organizations can also apply tailoring guidance to the security control baselines in Appendix D and CNSS Instruction 1253 to develop overlays for community-wide use or to address specialized requirements, technologies, or missions/environments of operation (e.g., DoD-tactical, Federal Public Key Infrastructure, or Federal Identity, Credential, and Access Management, space operations). Appendix I provides guidance on developing overlays.

> Security plans need not be single documents; the plans can be a collection of various documents including documents that already exist. Effective security plans make extensive use of references to policies, procedures, and additional documents (e.g., design and implementation specifications) where more detailed information can be obtained. This reduces the documentation requirements associated with security programs and maintains security-related information in other established management/operational areas related to enterprise architecture, system development life cycle, systems engineering, and acquisition. For example, security plans do not contain detailed contingency plan or incident response plan information but instead provide explicitly or by reference, sufficient information to define what needs to be accomplished by those plans.

Related Controls: *AC-2*, *AC-6*, *AC-14*, *AC-17*, *AC-20*, *CA-2*, *CA-3*, *CA-7*, *CM-9*, *CP-2*, *IR-8*, *MA-4*, *MA-5*, *MP-2*, *MP-4*, *MP-5*, *PL-7*, *PM-1*, *PM-7*, *PM-8*, *PM-9*, *PM-11*, *SA-5*, *SA-17*

PL-2a.

Develops a security plan for the information system that:

PL-2a.1.

Is consistent with the organizations enterprise architecture;

PL-2a.2.

Explicitly defines the authorization boundary for the system;

PL-2a.3.

Describes the operational context of the information system in terms of missions and business processes;

PL-2a.4.

Provides the security categorization of the information system including supporting rationale;

PL-2a.5.

Describes the operational environment for the information system and relationships with or connections to other information systems;

PL-2a.6.

Provides an overview of the security requirements for the system;

PL-2a.7.

Identifies any relevant overlays, if applicable;

PL-2a.8.

Describes the security controls in place or planned for meeting those requirements including a rationale for the tailoring decisions; and

PL-2a.9.

Is reviewed and approved by the authorizing official or designated representative prior to plan implementation;

PL-2b.

Distributes copies of the security plan and communicates subsequent changes to the plan to [Assignment: organization-defined personnel or roles];

PL-2c.

Reviews the security plan for the information system [Assignment: organization-defined frequency];

PL-2d.

Updates the plan to address changes to the information system/environment of operation or problems identified during plan implementation or security control assessments; and

PL-2e.

Protects the security plan from unauthorized disclosure and modification.

PL-2 (1) : CONCEPT OF OPERATIONS

[Withdrawn: Incorporated into PL-7].

PL-2 (2) : FUNCTIONAL ARCHITECTURE

[Withdrawn: Incorporated into PL-8].

PL-2 (3) : PLAN / COORDINATE WITH OTHER ORGANIZATIONAL ENTITIES

Baseline-Impact: *MODERATE*, **HIGH**

The organization plans and coordinates security-related activities affecting the information system with [Assignment: organization-defined individuals or groups] before conducting such activities in order to reduce the impact on other organizational entities.

Note: Security-related activities include, for example, security assessments, audits, hardware and software maintenance, patch management, and contingency plan testing. Advance planning and coordination includes emergency and nonemergency (i.e., planned or nonurgent unplanned) situations. The process defined by organizations to plan and coordinate security-related activities can be included in security plans for information systems or other documents, as appropriate.

Related Controls: *CP-4*, *IR-4*

Control Family: PLANNING

PL-3 : SYSTEM SECURITY PLAN UPDATE

[Withdrawn: Incorporated into PL-2].

Control Family: PLANNING

PL-4 : RULES OF BEHAVIOR

Priority: P2

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: This control enhancement applies to organizational users. Organizations consider rules of behavior based on individual user roles and responsibilities, differentiating, for example, between rules that apply to privileged users and rules that apply to general users. Establishing rules of behavior for some types of non-organizational users including, for example, individuals who simply receive data/information from federal information systems, is often not feasible given the large number of such users and the limited nature of their interactions with the systems. Rules of behavior for both organizational and non-organizational users can also be established in AC-8, System Use Notification. PL-4 b. (the signed acknowledgment portion of this control) may be satisfied by the security awareness training and role-based security training programs conducted by organizations if such training includes rules of behavior. Organizations can use electronic signatures for acknowledging rules of behavior.

Related Controls: *AC-2*, *AC-6*, *AC-8*, *AC-9*, *AC-17*, *AC-18*, *AC-19*, *AC-20*, *AT-2*, *AT-3*, *CM-11*, *IA-2*, *IA-4*, *IA-5*, *MP-7*, *PS-6*, *PS-8*, *SA-5*

PL-4a.

> Establishes and makes readily available to individuals requiring access to the information system, the rules that describe their responsibilities and expected behavior with regard to information and information system usage;

PL-4b.

> Receives a signed acknowledgment from such individuals, indicating that they have read, understand, and agree to abide by the rules of behavior, before authorizing access to information and the information system;

PL-4c.

> Reviews and updates the rules of behavior [Assignment: organization-defined frequency]; and

PL-4d.

> Requires individuals who have signed a previous version of the rules of behavior to read and re-sign when the rules of behavior are revised/updated.

PL-4 (1) : SOCIAL MEDIA AND NETWORKING RESTRICTIONS

Baseline-Impact: *MODERATE*, **HIGH**

> The organization includes in the rules of behavior, explicit restrictions on the use of social media/networking sites and posting organizational information on public websites.

Note: This control enhancement addresses rules of behavior related to the use of social media/networking sites: (i) when organizational personnel are using such sites for official duties or in the conduct of official business; (ii) when organizational information is involved in social media/networking transactions; and (iii) when personnel are accessing social media/networking sites from organizational information systems. Organizations also address specific rules that prevent unauthorized entities from obtaining and/or inferring non-public organizational information (e.g., system account information, personally identifiable information) from social media/networking sites.

Control Family: PLANNING

PL-5 : PRIVACY IMPACT ASSESSMENT

[Withdrawn: Incorporated into Appendix J, AR-2].

Control Family: PLANNING

PL-6 : SECURITY-RELATED ACTIVITY PLANNING

[Withdrawn: Incorporated into PL-2].

Control Family: PLANNING

PL-7 : SECURITY CONCEPT OF OPERATIONS

Priority: P0

The organization:

Note: The security CONOPS may be included in the security plan for the information system or in other system development life cycle-related documents, as appropriate. Changes to the CONOPS are reflected in ongoing updates to the security plan, the information security architecture, and other appropriate organizational documents (e.g., security specifications for procurements/acquisitions, system development life cycle documents, and systems/security engineering documents).

Related Controls: *PL-2*

PL-7a.

Develops a security Concept of Operations (CONOPS) for the information system containing at a minimum, how the organization intends to operate the system from the perspective of information security; and

PL-7b.

Reviews and updates the CONOPS [Assignment: organization-defined frequency].

Control Family: PLANNING

PL-8 : INFORMATION SECURITY ARCHITECTURE

Priority: P1

Baseline-Impact: *MODERATE*, **HIGH**

The organization:

Note:

This control addresses actions taken by organizations in the design and development of information systems. The information security architecture at the individual information system level is consistent with and complements the more global, organization-wide information security architecture described in PM-7 that is integral to and developed as part of the enterprise architecture. The information security architecture includes an architectural description, the placement/allocation of security functionality (including security controls), security-related information for external interfaces, information being exchanged across the interfaces, and the protection mechanisms associated with each interface. In addition, the security architecture can include other important security-related information, for example, user roles and access privileges assigned to each role, unique security requirements, the types of information processed, stored, and transmitted by the information system, restoration priorities of information and information system services, and any other specific protection needs.

In todays modern architecture, it is becoming less common for organizations to control all information resources. There are going to be key dependencies on external information services and service providers. Describing such dependencies in the information security architecture is important to developing a comprehensive mission/business protection strategy. Establishing, developing, documenting, and maintaining under configuration control, a baseline configuration for organizational information systems is critical to implementing and maintaining an effective information security architecture. The development of the information security architecture is coordinated with the Senior Agency Official for Privacy (SAOP)/Chief Privacy Officer (CPO) to ensure that security controls needed to support privacy requirements are identified and effectively implemented. PL-8 is primarily directed at organizations (i.e., internally focused) to help ensure that organizations develop an information security architecture for the information system, and that the security architecture is integrated with or tightly coupled to the enterprise architecture through the organization-wide information security architecture. In contrast, SA-17 is primarily directed at external information technology product/system developers and integrators (although SA-17 could be used internally within organizations for in-house system development). SA-17, which is complementary to PL-8, is selected when organizations outsource the development of information systems or information system components to external entities, and there is a need to demonstrate/show consistency with the organizations enterprise architecture and information security architecture.

Related Controls: *CM-2*, *CM-6*, *PL-2*, *PM-7*, *SA-5*, *SA-17*, Appendix J

PL-8a.

Develops an information security architecture for the information system that:

PL-8a.1.

Describes the overall philosophy, requirements, and approach to be taken with regard to protecting the confidentiality, integrity, and availability of organizational information;

PL-8a.2.

Describes how the information security architecture is integrated into and supports the enterprise architecture; and

PL-8a.3.

Describes any information security assumptions about, and dependencies on, external services;

PL-8b.

Reviews and updates the information security architecture [Assignment: organization-defined frequency] to reflect updates in the enterprise architecture; and

PL-8c.

Ensures that planned information security architecture changes are reflected in the security plan, the security Concept of Operations (CONOPS), and organizational procurements/acquisitions.

PL-8 (1) : DEFENSE-IN-DEPTH

The organization designs its security architecture using a defense-in-depth approach that:

Note: Organizations strategically allocate security safeguards (procedural, technical, or both) in the security architecture so that adversaries have to overcome multiple safeguards to achieve their objective. Requiring adversaries to defeat multiple mechanisms makes it more difficult to successfully attack critical information resources (i.e., increases adversary work factor) and also increases the likelihood of detection. The coordination of allocated safeguards is essential to ensure that an attack that involves one safeguard does not create adverse unintended consequences (e.g., lockout, cascading alarms) by interfering with another safeguard. Placement of security safeguards is a key activity. Greater asset criticality or information value merits additional layering. Thus, an organization may choose to place anti-virus software at organizational boundary layers, email/web servers, notebook computers, and workstations to maximize the number of related safeguards adversaries must penetrate before compromising the information and information systems.

Related Controls: *SC-29*, *SC-36*

PL-8 (1)(a)

Allocates [Assignment: organization-defined security safeguards] to [Assignment: organization-defined locations and architectural layers]; and

PL-8 (1)(b)

Ensures that the allocated security safeguards operate in a coordinated and mutually reinforcing manner.

PL-8 (2) : SUPPLIER DIVERSITY

The organization requires that [Assignment: organization-defined security safeguards] allocated to [Assignment: organization-defined locations and architectural layers] are obtained from different suppliers.

Note: Different information technology products have different strengths and weaknesses. Providing a broad spectrum of products complements the individual offerings. For example, vendors offering malicious code protection typically update their products at different times, often developing solutions for known viruses, Trojans, or worms according to their priorities and development schedules. By having different products at different locations (e.g., server, boundary, desktop) there is an increased likelihood that at least one will detect the malicious code.

Related Controls: *SA-12*

Control Family: PLANNING

PL-9 : CENTRAL MANAGEMENT

Priority: P0

The organization centrally manages [Assignment: organization-defined security controls and related processes].

Note: Central management refers to the organization-wide management and implementation of selected security controls and related processes. Central management includes planning, implementing, assessing, authorizing, and monitoring the organization-defined, centrally managed security controls and processes. As central management of security controls is generally associated with common controls, such management promotes and facilitates standardization of security control implementations and management and judicious use of organizational resources. Centrally-managed security controls and processes may also meet independence requirements for assessments in support of initial and ongoing authorizations to operate as part of organizational continuous monitoring. As part of the security control selection process, organizations determine which controls may be suitable for central management based on organizational resources and capabilities. Organizations consider that it may not always be possible to centrally manage every aspect of a security control. In such cases, the security control is treated as a hybrid control with the control managed and implemented either centrally or at the information system level. Controls and control enhancements that are candidates for full or partial central management include, but are not limited to: AC-2 (1) (2) (3) (4); AC-17 (1) (2) (3) (9); AC-18 (1) (3) (4) (5); AC-19 (4); AC-22; AC-23; AT-2 (1) (2); AT-3 (1) (2) (3); AT-4; AU-6 (1) (3) (5) (6) (9); AU-7 (1) (2); AU-11, AU-13, AU-16, CA-2 (1) (2) (3); CA-3 (1) (2) (3); CA-7 (1); CA-9; CM-2 (1) (2); CM-3 (1) (4); CM-4; CM-6 (1); CM-7 (4) (5); CM-8 (all); CM-9 (1); CM-10; CM-11; CP-7 (all); CP-8 (all); SC-43; SI-2; SI-3; SI-7; and SI-8.

Control Family: PERSONNEL SECURITY

PS-1 : PERSONNEL SECURITY POLICY AND PROCEDURES

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: This control addresses the establishment of policy and procedures for the effective implementation of selected security controls and control enhancements in the PS family. Policy and procedures reflect applicable federal laws, Executive Orders, directives, regulations, policies, standards, and guidance. Security program policies and procedures at the organization level may make the need for system-specific policies and procedures unnecessary. The policy can be included as part of the general information security policy for organizations or conversely, can be represented by multiple policies reflecting the complex nature of certain organizations. The procedures can be established for the security program in general and for particular information systems, if needed. The organizational risk management strategy is a key factor in establishing policy and procedures.

Related Controls: *PM-9*

PS-1a.

Develops, documents, and disseminates to [Assignment: organization-defined personnel or roles]:

PS-1a.1.

A personnel security policy that addresses purpose, scope, roles, responsibilities, management commitment, coordination among organizational entities, and compliance; and

PS-1a.2.

Procedures to facilitate the implementation of the personnel security policy and associated personnel security controls; and

PS-1b.

Reviews and updates the current:

PS-1b.1.

Personnel security policy [Assignment: organization-defined frequency]; and

PS-1b.2.

Personnel security procedures [Assignment: organization-defined frequency].

Control Family: PERSONNEL SECURITY

PS-2 : POSITION RISK DESIGNATION

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: Position risk designations reflect Office of Personnel Management policy and guidance. Risk designations can guide and inform the types of authorizations individuals receive when accessing organizational information and information systems. Position screening criteria include explicit information security role appointment requirements (e.g., training, security clearances).

Related Controls: *AT-3*, *PL-2*, *PS-3*

PS-2a.

Assigns a risk designation to all organizational positions;

PS-2b.

Establishes screening criteria for individuals filling those positions; and

PS-2c.

Reviews and updates position risk designations [Assignment: organization-defined frequency].

Control Family: PERSONNEL SECURITY

PS-3 : PERSONNEL SCREENING

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: Personnel screening and rescreening activities reflect applicable federal laws, Executive Orders, directives, regulations, policies, standards, guidance, and specific criteria established for the risk designations of assigned positions. Organizations may define different rescreening conditions and frequencies for personnel accessing information systems based on types of information processed, stored, or transmitted by the systems.

Related Controls: *AC-2*, *IA-4*, *PE-2*, *PS-2*

PS-3a.

Screens individuals prior to authorizing access to the information system; and

PS-3b.

Rescreens individuals according to [Assignment: organization-defined conditions requiring rescreening and, where rescreening is so indicated, the frequency of such rescreening].

PS-3 (1) : CLASSIFIED INFORMATION

The organization ensures that individuals accessing an information system processing, storing, or transmitting classified information are cleared and indoctrinated to the highest classification level of the information to which they have access on the system.

Related Controls: *AC-3*, *AC-4*

PS-3 (2) : FORMAL INDOCTRINATION

The organization ensures that individuals accessing an information system processing, storing, or transmitting types of classified information which require formal indoctrination, are formally indoctrinated for all of the relevant types of information to which they have access on the system.

Note: Types of classified information requiring formal indoctrination include, for example, Special Access Program (SAP), Restricted Data (RD), and Sensitive Compartment Information (SCI).

Related Controls: *AC-3*, *AC-4*

PS-3 (3) : INFORMATION WITH SPECIAL PROTECTION MEASURES

The organization ensures that individuals accessing an information system processing, storing, or transmitting information requiring special protection:

Note: Organizational information requiring special protection includes, for example, Controlled Unclassified Information (CUI) and Sources and Methods Information (SAMI). Personnel security criteria include, for example, position sensitivity background screening requirements.

PS-3 (3)(a)

Have valid access authorizations that are demonstrated by assigned official government duties; and

PS-3 (3)(b)

Satisfy [Assignment: organization-defined additional personnel screening criteria].

Control Family: PERSONNEL SECURITY

PS-4 : PERSONNEL TERMINATION

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization, upon termination of individual employment:

Note: Information system-related property includes, for example, hardware authentication tokens, system administration technical manuals, keys, identification cards, and building passes. Exit interviews ensure that terminated individuals understand the security constraints imposed by being former employees and that proper accountability is achieved for information system-related property. Security topics of interest at exit interviews can include, for example, reminding terminated individuals of nondisclosure agreements and potential limitations on future employment. Exit interviews may not be possible for some terminated individuals, for example, in cases related to job abandonment, illnesses, and nonavailability of supervisors. Exit interviews are important for individuals with security clearances. Timely execution of termination actions is essential for individuals terminated for cause. In certain situations, organizations consider disabling the information system accounts of individuals that are being terminated prior to the individuals being notified.

Related Controls: *AC-2*, *IA-4*, *PE-2*, *PS-5*, *PS-6*

PS-4a.

Disables information system access within [Assignment: organization-defined time period];

PS-4b.

Terminates/revokes any authenticators/credentials associated with the individual;

PS-4c.

Conducts exit interviews that include a discussion of [Assignment: organization-defined information security topics];

PS-4d.

Retrieves all security-related organizational information system-related property;

PS-4e.

Retains access to organizational information and information systems formerly controlled by terminated individual; and

PS-4f.

Notifies [Assignment: organization-defined personnel or roles] within [Assignment: organization-defined time period].

PS-4 (1) : POST-EMPLOYMENT REQUIREMENTS

The organization:

Note: Organizations consult with the Office of the General Counsel regarding matters of post-employment requirements on terminated individuals.

PS-4 (1)(a)

Notifies terminated individuals of applicable, legally binding post-employment requirements for the protection of organizational information; and

PS-4 (1)(b)

Requires terminated individuals to sign an acknowledgment of post-employment requirements as part of the organizational termination process.

PS-4 (2) : AUTOMATED NOTIFICATION

Baseline-Impact: **HIGH**

> The organization employs automated mechanisms to notify [Assignment: organization-defined personnel or roles] upon termination of an individual.

Note: In organizations with a large number of employees, not all personnel who need to know about termination actions receive the appropriate notificationsor, if such notifications are received, they may not occur in a timely manner. Automated mechanisms can be used to send automatic alerts or notifications to specific organizational personnel or roles (e.g., management personnel, supervisors, personnel security officers, information security officers, systems administrators, or information technology administrators) when individuals are terminated. Such automatic alerts or notifications can be conveyed in a variety of ways, including, for example, telephonically, via electronic mail, via text message, or via websites.

Control Family: PERSONNEL SECURITY

PS-5 : PERSONNEL TRANSFER

Priority: P2

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The organization:

Note: This control applies when reassignments or transfers of individuals are permanent or of such extended durations as to make the actions warranted. Organizations define actions appropriate for the types of reassignments or transfers, whether permanent or extended. Actions that may be required for personnel transfers or reassignments to other positions within organizations include, for example: (i) returning old and issuing new keys, identification cards, and building passes; (ii) closing information system accounts and establishing new accounts; (iii) changing information system access authorizations (i.e., privileges); and (iv) providing for access to official records to which individuals had access at previous work locations and in previous information system accounts.

Related Controls: *AC-2*, *IA-4*, *PE-2*, *PS-4*

PS-5a.

> Reviews and confirms ongoing operational need for current logical and physical access authorizations to information systems/facilities when individuals are reassigned or transferred to other positions within the organization;

PS-5b.

Initiates [Assignment: organization-defined transfer or reassignment actions] within [Assignment: organization-defined time period following the formal transfer action];

PS-5c.

Modifies access authorization as needed to correspond with any changes in operational need due to reassignment or transfer; and

PS-5d.

Notifies [Assignment: organization-defined personnel or roles] within [Assignment: organization-defined time period].

Control Family: PERSONNEL SECURITY

PS-6 : ACCESS AGREEMENTS

Priority: P3

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: Access agreements include, for example, nondisclosure agreements, acceptable use agreements, rules of behavior, and conflict-of-interest agreements. Signed access agreements include an acknowledgment that individuals have read, understand, and agree to abide by the constraints associated with organizational information systems to which access is authorized. Organizations can use electronic signatures to acknowledge access agreements unless specifically prohibited by organizational policy.

Related Controls: *PL-4*, *PS-2*, *PS-3*, *PS-4*, *PS-8*

PS-6a.

Develops and documents access agreements for organizational information systems;

PS-6b.

Reviews and updates the access agreements [Assignment: organization-defined frequency]; and

PS-6c.

Ensures that individuals requiring access to organizational information and information systems:

PS-6c.1.

Sign appropriate access agreements prior to being granted access; and

PS-6c.2.

Re-sign access agreements to maintain access to organizational information systems when access agreements have been updated or [Assignment: organization-defined frequency].

PS-6 (1) : INFORMATION REQUIRING SPECIAL PROTECTION

[Withdrawn: Incorporated into PS-3].

PS-6 (2) : CLASSIFIED INFORMATION REQUIRING SPECIAL PROTECTION

The organization ensures that access to classified information requiring special protection is granted only to individuals who:

Note: Classified information requiring special protection includes, for example, collateral information, Special Access Program (SAP) information, and Sensitive Compartmented Information (SCI). Personnel security criteria reflect applicable federal laws, Executive Orders, directives, regulations, policies, standards, and guidance.

PS-6 (2)(a)

Have a valid access authorization that is demonstrated by assigned official government duties;

PS-6 (2)(b)

Satisfy associated personnel security criteria; and

PS-6 (2)(c)

Have read, understood, and signed a nondisclosure agreement.

PS-6 (3) : POST-EMPLOYMENT REQUIREMENTS

The organization:

Note: Organizations consult with the Office of the General Counsel regarding matters of post-employment requirements on terminated individuals.

PS-6 (3)(a)

Notifies individuals of applicable, legally binding post-employment requirements for protection of organizational information; and

PS-6 (3)(b)

Requires individuals to sign an acknowledgment of these requirements, if applicable, as part of granting initial access to covered information.

Control Family: PERSONNEL SECURITY

PS-7 : THIRD-PARTY PERSONNEL SECURITY

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: Third-party providers include, for example, service bureaus, contractors, and other organizations providing information system development, information technology services, outsourced applications, and network and security management. Organizations explicitly include personnel security requirements in acquisition-related documents. Third-party providers may have personnel working at organizational facilities with credentials, badges, or information system privileges issued by organizations. Notifications of third-party personnel changes ensure appropriate termination of privileges and credentials. Organizations define the transfers and terminations deemed reportable by security-related characteristics that include, for example, functions, roles, and nature of credentials/privileges associated with individuals transferred or terminated.

Related Controls: *PS-2*, *PS-3*, *PS-4*, *PS-5*, *PS-6*, *SA-9*, *SA-21*

PS-7a.

Establishes personnel security requirements including security roles and responsibilities for third-party providers;

PS-7b.

Requires third-party providers to comply with personnel security policies and procedures established by the organization;

PS-7c.

Documents personnel security requirements;

PS-7d.

Requires third-party providers to notify [Assignment: organization-defined personnel or roles] of any personnel transfers or terminations of third-party personnel who possess organizational credentials and/or badges, or who have information system privileges within [Assignment: organization-defined time period]; and

PS-7e.

Monitors provider compliance.

Control Family: PERSONNEL SECURITY

PS-8 : PERSONNEL SANCTIONS

Priority: P3

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: Organizational sanctions processes reflect applicable federal laws, Executive Orders, directives, regulations, policies, standards, and guidance. Sanctions processes are described in access agreements and can be included as part of general personnel policies and procedures for organizations. Organizations consult with the Office of the General Counsel regarding matters of employee sanctions.

Related Controls: *PL-4*, *PS-6*

PS-8a.

Employs a formal sanctions process for individuals failing to comply with established information security policies and procedures; and

PS-8b.

Notifies [Assignment: organization-defined personnel or roles] within [Assignment: organization-defined time period] when a formal employee sanctions process is initiated, identifying the individual sanctioned and the reason for the sanction.

Control Family: RISK ASSESSMENT

RA-1 : RISK ASSESSMENT POLICY AND PROCEDURES

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: This control addresses the establishment of policy and procedures for the effective implementation of selected security controls and control enhancements in the RA family. Policy and procedures reflect applicable federal laws, Executive Orders, directives, regulations, policies, standards, and guidance. Security program policies and procedures at the organization level may make the need for system-specific policies and procedures unnecessary. The policy can be included as part of the general information security policy for organizations or conversely, can be represented by multiple policies reflecting the complex nature of certain organizations. The procedures can be established for the security program in general and for particular information systems, if needed. The organizational risk management strategy is a key factor in establishing policy and procedures.

Related Controls: *PM-9*

RA-1a.

Develops, documents, and disseminates to [Assignment: organization-defined personnel or roles]:

RA-1a.1.

A risk assessment policy that addresses purpose, scope, roles, responsibilities, management commitment, coordination among organizational entities, and compliance; and

RA-1a.2.

Procedures to facilitate the implementation of the risk assessment policy and associated risk assessment controls; and

RA-1b.

Reviews and updates the current:

RA-1b.1.

Risk assessment policy [Assignment: organization-defined frequency]; and

RA-1b.2.

Risk assessment procedures [Assignment: organization-defined frequency].

Control Family: RISK ASSESSMENT

RA-2 : SECURITY CATEGORIZATION

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: Clearly defined authorization boundaries are a prerequisite for effective security categorization decisions. Security categories describe the potential adverse impacts to organizational operations, organizational assets, and individuals if organizational information and information systems are comprised through a loss of confidentiality, integrity, or availability. Organizations conduct the security categorization process as an organization-wide activity with the involvement of chief information officers, senior information security officers, information system owners, mission/business owners, and information owners/stewards. Organizations also consider the potential adverse impacts to other organizations and, in accordance with the USA PATRIOT Act of 2001 and Homeland Security Presidential Directives, potential national-level adverse impacts. Security categorization processes carried out by organizations facilitate the development of inventories of information assets, and along with CM-8, mappings to specific information system components where information is processed, stored, or transmitted.

Related Controls: *CM-8*, *MP-4*, *RA-3*, *SC-7*

RA-2a.

Categorizes information and the information system in accordance with applicable federal laws, Executive Orders, directives, policies, regulations, standards, and guidance;

RA-2b.

Documents the security categorization results (including supporting rationale) in the security plan for the information system; and

RA-2c.

Ensures that the authorizing official or authorizing official designated representative reviews and approves the security categorization decision.

Control Family: RISK ASSESSMENT

RA-3 : RISK ASSESSMENT

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note:

Clearly defined authorization boundaries are a prerequisite for effective risk assessments. Risk assessments take into account threats, vulnerabilities, likelihood, and impact to organizational operations and assets, individuals, other organizations, and the Nation based on the operation and use of information systems. Risk assessments also take into account risk from external parties (e.g., service providers, contractors operating information systems on behalf of the organization, individuals accessing organizational information systems, outsourcing entities). In accordance with OMB policy and related E-authentication initiatives, authentication of public users accessing federal information systems may also be required to protect nonpublic or privacy-related information. As such, organizational assessments of risk also address public access to federal information systems.

Risk assessments (either formal or informal) can be conducted at all three tiers in the risk management hierarchy (i.e., organization level, mission/business process level, or information system level) and at any phase in the system development life cycle. Risk assessments can also be conducted at various steps in the Risk Management Framework, including categorization, security control selection, security control implementation, security control assessment, information system authorization, and security control monitoring. RA-3 is noteworthy in that the control must be partially implemented prior to the implementation of other controls in order to complete the first two steps in the Risk Management Framework. Risk assessments can play an important role in security control selection processes, particularly during the application of tailoring guidance, which includes security control supplementation.

Related Controls: *RA-2*, *PM-9*

RA-3a.

Conducts an assessment of risk, including the likelihood and magnitude of harm, from the unauthorized access, use, disclosure, disruption, modification, or destruction of the information system and the information it processes, stores, or transmits;

RA-3b.

Documents risk assessment results in [Selection: security plan; risk assessment report; [Assignment: organization-defined document]];

RA-3c.

Reviews risk assessment results [Assignment: organization-defined frequency];

RA-3d.

Disseminates risk assessment results to [Assignment: organization-defined personnel or roles]; and

RA-3e.

Updates the risk assessment [Assignment: organization-defined frequency] or whenever there are significant changes to the information system or environment of operation (including the identification of new threats and vulnerabilities), or other conditions that may impact the security state of the system.

Control Family: RISK ASSESSMENT

RA-4 : RISK ASSESSMENT UPDATE

[Withdrawn: Incorporated into RA-3].

Control Family: RISK ASSESSMENT

RA-5 : VULNERABILITY SCANNING

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: Security categorization of information systems guides the frequency and comprehensiveness of vulnerability scans. Organizations determine the required vulnerability scanning for all information system components, ensuring that potential sources of vulnerabilities such as networked printers, scanners, and copiers are not overlooked. Vulnerability analyses for custom software applications may require additional approaches such as static analysis, dynamic analysis, binary analysis, or a hybrid of the three approaches. Organizations can employ these analysis approaches in a variety of tools (e.g., web-based application scanners, static analysis tools, binary analyzers) and in source code reviews. Vulnerability scanning includes, for example: (i) scanning for patch levels; (ii) scanning for functions, ports, protocols, and services that should not be accessible to users or devices; and (iii) scanning for improperly configured or incorrectly operating information flow control mechanisms. Organizations consider using tools that express vulnerabilities in the Common Vulnerabilities and Exposures (CVE) naming convention and that use the Open Vulnerability Assessment Language (OVAL) to determine/test for the presence of vulnerabilities. Suggested sources for vulnerability information include the Common Weakness Enumeration (CWE) listing and the National Vulnerability Database (NVD). In addition, security control assessments such as red team exercises provide other sources of potential vulnerabilities for which to scan. Organizations also consider using tools that express vulnerability impact by the Common Vulnerability Scoring System (CVSS).

Related Controls: *CA-2*, *CA-7*, *CM-4*, *CM-6*, *RA-2*, *RA-3*, *SA-11*, *SI-2*

RA-5a.

Scans for vulnerabilities in the information system and hosted applications [Assignment: organization-defined frequency and/or randomly in accordance with organization-defined process] and when new vulnerabilities potentially affecting the system/applications are identified and reported;

RA-5b.

Employs vulnerability scanning tools and techniques that facilitate interoperability among tools and automate parts of the vulnerability management process by using standards for:

RA-5b.1.

Enumerating platforms, software flaws, and improper configurations;

RA-5b.2.

Formatting checklists and test procedures; and

RA-5b.3.

Measuring vulnerability impact;

RA-5c.

Analyzes vulnerability scan reports and results from security control assessments;

RA-5d.

Remediates legitimate vulnerabilities [Assignment: organization-defined response times] in accordance with an organizational assessment of risk; and

RA-5e.

Shares information obtained from the vulnerability scanning process and security control assessments with [Assignment: organization-defined personnel or roles] to help eliminate similar vulnerabilities in other information systems (i.e., systemic weaknesses or deficiencies).

RA-5 (1) : UPDATE TOOL CAPABILITY

Baseline-Impact: *MODERATE*, **HIGH**

The organization employs vulnerability scanning tools that include the capability to readily update the information system vulnerabilities to be scanned.

Note: The vulnerabilities to be scanned need to be readily updated as new vulnerabilities are discovered, announced, and scanning methods developed. This updating process helps to ensure that potential vulnerabilities in the information system are identified and addressed as quickly as possible.

Related Controls: *SI-3*, *SI-7*

RA-5 (2) : UPDATE BY FREQUENCY / PRIOR TO NEW SCAN / WHEN IDENTIFIED

Baseline-Impact: *MODERATE*, **HIGH**

The organization updates the information system vulnerabilities scanned [Selection (one or more): [Assignment: organization-defined frequency]; prior to a new scan; when new vulnerabilities are identified and reported].

Related Controls: *SI-3*, *SI-5*

RA-5 (3) : BREADTH / DEPTH OF COVERAGE

The organization employs vulnerability scanning procedures that can identify the breadth and depth of coverage (i.e., information system components scanned and vulnerabilities checked).

RA-5 (4) : DISCOVERABLE INFORMATION

Baseline-Impact: HIGH

The organization determines what information about the information system is discoverable by adversaries and subsequently takes [Assignment: organization-defined corrective actions].

Note: Discoverable information includes information that adversaries could obtain without directly compromising or breaching the information system, for example, by collecting information the system is exposing or by conducting extensive searches of the web. Corrective actions can include, for example, notifying appropriate organizational personnel, removing designated information, or changing the information system to make designated information less relevant or attractive to adversaries.

Related Controls: *AU-13*

RA-5 (5) : PRIVILEGED ACCESS

Baseline-Impact: *MODERATE*, **HIGH**

> The information system implements privileged access authorization to [Assignment: organization-identified information system components] for selected [Assignment: organization-defined vulnerability scanning activities].

Note: In certain situations, the nature of the vulnerability scanning may be more intrusive or the information system component that is the subject of the scanning may contain highly sensitive information. Privileged access authorization to selected system components facilitates more thorough vulnerability scanning and also protects the sensitive nature of such scanning.

RA-5 (6) : AUTOMATED TREND ANALYSES

> The organization employs automated mechanisms to compare the results of vulnerability scans over time to determine trends in information system vulnerabilities.

Related Controls: *IR-4*, *IR-5*, *SI-4*

RA-5 (7) : AUTOMATED DETECTION AND NOTIFICATION OF UNAUTHORIZED COMPONENTS

> [Withdrawn: Incorporated into CM-8].

RA-5 (8) : REVIEW HISTORIC AUDIT LOGS

> The organization reviews historic audit logs to determine if a vulnerability identified in the information system has been previously exploited.

Related Controls: *AU-6*

RA-5 (9) : PENETRATION TESTING AND ANALYSES

> [Withdrawn: Incorporated into CA-8].

RA-5 (10) : CORRELATE SCANNING INFORMATION

The organization correlates the output from vulnerability scanning tools to determine the presence of multi-vulnerability/multi-hop attack vectors.

Control Family: RISK ASSESSMENT

RA-6 : TECHNICAL SURVEILLANCE COUNTERMEASURES SURVEY

Priority: P0

The organization employs a technical surveillance countermeasures survey at [Assignment: organization-defined locations] [Selection (one or more): [Assignment: organization-defined frequency]; [Assignment: organization-defined events or indicators occur]].

Note: Technical surveillance countermeasures surveys are performed by qualified personnel to detect the presence of technical surveillance devices/hazards and to identify technical security weaknesses that could aid in the conduct of technical penetrations of surveyed facilities. Such surveys provide evaluations of the technical security postures of organizations and facilities and typically include thorough visual, electronic, and physical examinations in and about surveyed facilities. The surveys also provide useful input into risk assessments and organizational exposure to potential adversaries.

Control Family: SYSTEM AND SERVICES ACQUISITION

SA-1 : SYSTEM AND SERVICES ACQUISITION POLICY AND PROCEDURES

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: This control addresses the establishment of policy and procedures for the effective implementation of selected security controls and control enhancements in the SA family. Policy and procedures reflect applicable federal laws, Executive Orders, directives, regulations, policies, standards, and guidance. Security program policies and procedures at the organization level may make the need for system-specific policies and procedures unnecessary. The policy can be included as part of the general information security policy for organizations or conversely, can be represented by multiple policies reflecting the complex nature of certain organizations. The procedures can be established for the security program in general and for particular information systems, if needed. The organizational risk management strategy is a key factor in establishing policy and procedures.

Related Controls: *PM-9*

SA-1a.

Develops, documents, and disseminates to [Assignment: organization-defined personnel or roles]:

SA-1a.1.

A system and services acquisition policy that addresses purpose, scope, roles, responsibilities, management commitment, coordination among organizational entities, and compliance; and

SA-1a.2.

Procedures to facilitate the implementation of the system and services acquisition policy and associated system and services acquisition controls; and

SA-1b.

Reviews and updates the current:

SA-1b.1.

System and services acquisition policy [Assignment: organization-defined frequency]; and

SA-1b.2.

System and services acquisition procedures [Assignment: organization-defined frequency].

Control Family: SYSTEM AND SERVICES ACQUISITION

SA-2 : ALLOCATION OF RESOURCES

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: Resource allocation for information security includes funding for the initial information system or information system service acquisition and funding for the sustainment of the system/service.

Related Controls: *PM-3*, *PM-11*

SA-2a.

Determines information security requirements for the information system or information system service in mission/business process planning;

SA-2b.

Determines, documents, and allocates the resources required to protect the information system or information system service as part of its capital planning and investment control process; and

SA-2c.

Establishes a discrete line item for information security in organizational programming and budgeting documentation.

Control Family: SYSTEM AND SERVICES ACQUISITION

SA-3 : SYSTEM DEVELOPMENT LIFE CYCLE

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: A well-defined system development life cycle provides the foundation for the successful development, implementation, and operation of organizational information systems. To apply the required security controls within the system development life cycle requires a basic understanding of information security, threats, vulnerabilities, adverse impacts, and risk to critical missions/business functions. The security engineering principles in SA-8 cannot be properly applied if individuals that design, code, and test information systems and system components (including information technology products) do not understand security. Therefore, organizations include qualified personnel, for example, chief information security officers, security architects, security engineers, and information system security officers in system development life cycle activities to ensure that security requirements are incorporated into organizational information systems. It is equally important that developers include individuals on the development team that possess the requisite security expertise and skills to ensure that needed security capabilities are effectively integrated into the information system. Security awareness and training programs can help ensure that individuals having key security roles and responsibilities have the appropriate experience, skills, and expertise to conduct assigned system development life cycle activities. The effective integration of security requirements into enterprise architecture also helps to ensure that important security considerations are addressed early in the system development life cycle and that those considerations are directly related to the organizational mission/business processes. This process also facilitates the integration of the information security architecture into the enterprise architecture, consistent with organizational risk management and information security strategies.

Related Controls: *AT-3*, *PM-7*, *SA-8*

SA-3a.

Manages the information system using [Assignment: organization-defined system development life cycle] that incorporates information security considerations;

SA-3b.

Defines and documents information security roles and responsibilities throughout the system development life cycle;

SA-3c.

Identifies individuals having information security roles and responsibilities; and

SA-3d.

Integrates the organizational information security risk management process into system development life cycle activities.

Control Family: SYSTEM AND SERVICES ACQUISITION

SA-4 : ACQUISITION PROCESS

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization includes the following requirements, descriptions, and criteria, explicitly or by reference, in the acquisition contract for the information system, system component, or information system service in accordance with applicable federal laws, Executive Orders, directives, policies, regulations, standards, guidelines, and organizational mission/business needs:

Note:

Information system components are discrete, identifiable information technology assets (e.g., hardware, software, or firmware) that represent the building blocks of an information system. Information system components include commercial information technology products. Security functional requirements include security capabilities, security functions, and security mechanisms. Security strength requirements associated with such capabilities, functions, and mechanisms include degree of correctness, completeness, resistance to direct attack, and resistance to tampering or bypass. Security assurance requirements include: (i) development processes, procedures, practices, and methodologies; and (ii) evidence from development and assessment activities providing grounds for confidence that the required security functionality has been implemented and the required security strength has been achieved. Security documentation requirements address all phases of the system development life cycle.

Security functionality, assurance, and documentation requirements are expressed in terms of security controls and control enhancements that have been selected through the tailoring process. The security control tailoring process includes, for example, the specification of parameter values through the use of assignment and selection statements and the specification of platform dependencies and implementation information. Security documentation provides user and administrator guidance regarding the implementation and operation of security controls. The level of detail required in security documentation is based on the security category or classification level of the information system and the degree to which organizations depend on the stated security capability, functions, or mechanisms to meet overall risk response expectations (as defined in the organizational risk management strategy). Security requirements can also include organizationally mandated configuration settings specifying allowed functions, ports, protocols, and services. Acceptance criteria for information systems, information system components, and information system services are defined in the same manner as such criteria for any organizational acquisition or procurement. The Federal Acquisition Regulation (FAR) Section 7.103 contains information security requirements from FISMA.

Related Controls: *CM-6*, *PL-2*, *PS-7*, *SA-3*, *SA-5*, *SA-8*, *SA-11*, *SA-12*

SA-4a.

Security functional requirements;

SA-4b.

Security strength requirements;

SA-4c.

Security assurance requirements;

SA-4d.

Security-related documentation requirements;

SA-4e.

Requirements for protecting security-related documentation;

SA-4f.

Description of the information system development environment and environment in which the system is intended to operate; and

SA-4g.

Acceptance criteria.

SA-4 (1) : FUNCTIONAL PROPERTIES OF SECURITY CONTROLS

Baseline-Impact: *MODERATE*, **HIGH**

> The organization requires the developer of the information system, system component, or information system service to provide a description of the functional properties of the security controls to be employed.

Note: Functional properties of security controls describe the functionality (i.e., security capability, functions, or mechanisms) visible at the interfaces of the controls and specifically exclude functionality and data structures internal to the operation of the controls.

Related Controls: *SA-5*

SA-4 (2) : DESIGN / IMPLEMENTATION INFORMATION FOR SECURITY CONTROLS

Baseline-Impact: *MODERATE*, **HIGH**

> The organization requires the developer of the information system, system component, or information system service to provide design and implementation information for the security controls to be employed that includes: [Selection (one or more): security-relevant external system interfaces; high-level design; low-level design; source code or hardware schematics; [Assignment: organization-defined design/implementation information]] at [Assignment: organization-defined level of detail].

Note: Organizations may require different levels of detail in design and implementation documentation for security controls employed in organizational information systems, system components, or information system services based on mission/business requirements, requirements for trustworthiness/resiliency, and requirements for analysis and testing. Information systems can be partitioned into multiple subsystems. Each subsystem within the system can contain one or more modules. The high-level design for the system is expressed in terms of multiple subsystems and the interfaces between subsystems providing security-relevant functionality. The low-level design for the system is expressed in terms of modules with particular emphasis on software and firmware (but not excluding hardware) and the interfaces between modules providing security-relevant functionality. Source code and hardware schematics are typically referred to as the implementation representation of the information system.

Related Controls: *SA-5*

SA-4 (3) : DEVELOPMENT METHODS / TECHNIQUES / PRACTICES

> The organization requires the developer of the information system, system component, or information system service to demonstrate the use of a system development life cycle that includes [Assignment: organization-defined state-of-the-practice system/security engineering methods, software development methods, testing/evaluation/validation techniques, and quality control processes].

Note: Following a well-defined system development life cycle that includes state-of-the-practice software development methods, systems/security engineering methods, quality control processes, and testing, evaluation, and validation techniques helps to reduce the number and severity of latent errors within information systems, system components, and information system services. Reducing the number/severity of such errors reduces the number of vulnerabilities in those systems, components, and services.

Related Controls: *SA-12*

SA-4 (4) : ASSIGNMENT OF COMPONENTS TO SYSTEMS

[Withdrawn: Incorporated into CM-8 (9)].

SA-4 (5) : SYSTEM / COMPONENT / SERVICE CONFIGURATIONS

The organization requires the developer of the information system, system component, or information system service to:

Note: Security configurations include, for example, the U.S. Government Configuration Baseline (USGCB) and any limitations on functions, ports, protocols, and services. Security characteristics include, for example, requiring that all default passwords have been changed.

Related Controls: *CM-8*

SA-4 (5)(a)

Deliver the system, component, or service with [Assignment: organization-defined security configurations] implemented; and

SA-4 (5)(b)

Use the configurations as the default for any subsequent system, component, or service reinstallation or upgrade.

SA-4 (6) : USE OF INFORMATION ASSURANCE PRODUCTS

The organization:

Note: COTS IA or IA-enabled information technology products used to protect classified information by cryptographic means may be required to use NSA-approved key management.

Related Controls: *SC-8*, *SC-12*, *SC-13*

SA-4 (6)(a)

Employs only government off-the-shelf (GOTS) or commercial off-the-shelf (COTS) information assurance (IA) and IA-enabled information technology products that compose an NSA-approved solution to protect classified information when the networks used to transmit the information are at a lower classification level than the information being transmitted; and

SA-4 (6)(b)

Ensures that these products have been evaluated and/or validated by NSA or in accordance with NSA-approved procedures.

SA-4 (7) : NIAP-APPROVED PROTECTION PROFILES

The organization:

Related Controls: *SC-12*, *SC-13*

SA-4 (7)(a)

Limits the use of commercially provided information assurance (IA) and IA-enabled information technology products to those products that have been successfully evaluated against a National Information Assurance partnership (NIAP)-approved Protection Profile for a specific technology type, if such a profile exists; and

SA-4 (7)(b)

Requires, if no NIAP-approved Protection Profile exists for a specific technology type but a commercially provided information technology product relies on cryptographic functionality to enforce its security policy, that the cryptographic module is FIPS-validated.

SA-4 (8) : CONTINUOUS MONITORING PLAN

The organization requires the developer of the information system, system component, or information system service to produce a plan for the continuous monitoring of security control effectiveness that contains [Assignment: organization-defined level of detail].

Note: The objective of continuous monitoring plans is to determine if the complete set of planned, required, and deployed security controls within the information system, system component, or information system service continue to be effective over time based on the inevitable changes that occur. Developer continuous monitoring plans include a sufficient level of detail such that the information can be incorporated into the continuous monitoring strategies and programs implemented by organizations.

Related Controls: *CA-7*

SA-4 (9) : FUNCTIONS / PORTS / PROTOCOLS / SERVICES IN USE

Baseline-Impact: *MODERATE*, **HIGH**

The organization requires the developer of the information system, system component, or information system service to identify early in the system development life cycle, the functions, ports, protocols, and services intended for organizational use.

Note: The identification of functions, ports, protocols, and services early in the system development life cycle (e.g., during the initial requirements definition and design phases) allows organizations to influence the design of the information system, information system component, or information system service. This early involvement in the life cycle helps organizations to avoid or minimize the use of functions, ports, protocols, or services that pose unnecessarily high risks and understand the trade-offs involved in blocking specific ports, protocols, or services (or when requiring information system service providers to do so). Early identification of functions, ports, protocols, and services avoids costly retrofitting of security controls after the information system, system component, or information system service has been implemented. SA-9 describes requirements for external information system services with organizations identifying which functions, ports, protocols, and services are provided from external sources.

Related Controls: *CM-7*, *SA-9*

SA-4 (10) : USE OF APPROVED PIV PRODUCTS

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The organization employs only information technology products on the FIPS 201-approved products list for Personal Identity Verification (PIV) capability implemented within organizational information systems.

Related Controls: *IA-2*, *IA-8*

Control Family: SYSTEM AND SERVICES ACQUISITION

SA-5 : INFORMATION SYSTEM DOCUMENTATION

Priority: P2

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The organization:

Note: This control helps organizational personnel understand the implementation and operation of security controls associated with information systems, system components, and information system services. Organizations consider establishing specific measures to determine the quality/completeness of the content provided. The inability to obtain needed documentation may occur, for example, due to the age of the information system/component or lack of support from developers and contractors. In those situations, organizations may need to recreate selected documentation if such documentation is essential to the effective implementation or operation of security controls. The level of protection provided for selected information system, component, or service documentation is commensurate with the security category or classification of the system. For example, documentation associated with a key DoD weapons system or command and control system would typically require a higher level of protection than a routine administrative system. Documentation that addresses information system vulnerabilities may also require an increased level of protection. Secure operation of the information system, includes, for example, initially starting the system and resuming secure system operation after any lapse in system operation.

Related Controls: *CM-6*, *CM-8*, *PL-2*, *PL-4*, *PS-2*, *SA-3*, *SA-4*

SA-5a.

Obtains administrator documentation for the information system, system component, or information system service that describes:

SA-5a.1.

Secure configuration, installation, and operation of the system, component, or service;

SA-5a.2.

Effective use and maintenance of security functions/mechanisms; and

SA-5a.3.

Known vulnerabilities regarding configuration and use of administrative (i.e., privileged) functions;

SA-5b.

Obtains user documentation for the information system, system component, or information system service that describes:

SA-5b.1.

User-accessible security functions/mechanisms and how to effectively use those security functions/mechanisms;

SA-5b.2.

Methods for user interaction, which enables individuals to use the system, component, or service in a more secure manner; and

SA-5b.3.

User responsibilities in maintaining the security of the system, component, or service;

SA-5c.

Documents attempts to obtain information system, system component, or information system service documentation when such documentation is either unavailable or nonexistent and takes [Assignment: organization-defined actions] in response;

SA-5d.

Protects documentation as required, in accordance with the risk management strategy; and

SA-5e.

Distributes documentation to [Assignment: organization-defined personnel or roles].

SA-5 (1) : FUNCTIONAL PROPERTIES OF SECURITY CONTROLS

[Withdrawn: Incorporated into SA-4 (1)].

SA-5 (2) : SECURITY-RELEVANT EXTERNAL SYSTEM INTERFACES

[Withdrawn: Incorporated into SA-4 (2)].

SA-5 (3) : HIGH-LEVEL DESIGN

[Withdrawn: Incorporated into SA-4 (2)].

SA-5 (4) : LOW-LEVEL DESIGN

[Withdrawn: Incorporated into SA-4 (2)].

SA-5 (5) : SOURCE CODE

[Withdrawn: Incorporated into SA-4 (2)].

Control Family: SYSTEM AND SERVICES ACQUISITION

SA-6 : SOFTWARE USAGE RESTRICTIONS

[Withdrawn: Incorporated into CM-10 and SI-7].

Control Family: SYSTEM AND SERVICES ACQUISITION

SA-7 : USER-INSTALLED SOFTWARE

[Withdrawn: Incorporated into CM-11 and SI-7].

Control Family: SYSTEM AND SERVICES ACQUISITION

SA-8 : SECURITY ENGINEERING PRINCIPLES

Priority: P1

Baseline-Impact: *MODERATE*, **HIGH**

> The organization applies information system security engineering principles in the specification, design, development, implementation, and modification of the information system.

Note: Organizations apply security engineering principles primarily to new development information systems or systems undergoing major upgrades. For legacy systems, organizations apply security engineering principles to system upgrades and modifications to the extent feasible, given the current state of hardware, software, and firmware within those systems. Security engineering principles include, for example: (i) developing layered protections; (ii) establishing sound security policy, architecture, and controls as the foundation for design; (iii) incorporating security requirements into the system development life cycle; (iv) delineating physical and logical security boundaries; (v) ensuring that system developers are trained on how to build secure software; (vi) tailoring security controls to meet organizational and operational needs; (vii) performing threat modeling to identify use cases, threat agents, attack vectors, and attack patterns as well as compensating controls and design patterns needed to mitigate risk; and (viii) reducing risk to acceptable levels, thus enabling informed risk management decisions.

Related Controls: *PM-7*, *SA-3*, *SA-4*, *SA-17*, *SC-2*, *SC-3*

Control Family: SYSTEM AND SERVICES ACQUISITION

SA-9 : EXTERNAL INFORMATION SYSTEM SERVICES

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The organization:

Note: External information system services are services that are implemented outside of the authorization boundaries of organizational information systems. This includes services that are used by, but not a part of, organizational information systems. FISMA and OMB policy require that organizations using external service providers that are processing, storing, or transmitting federal information or operating information systems on behalf of the federal government ensure that such providers meet the same security requirements that federal agencies are required to meet. Organizations establish relationships with external service providers in a variety of ways including, for example, through joint ventures, business partnerships, contracts, interagency agreements, lines of business arrangements, licensing agreements, and supply chain exchanges. The responsibility for managing risks from the use of external information system services remains with authorizing officials. For services external to organizations, a chain of trust requires that organizations establish and retain a level of confidence that each participating provider in the potentially complex consumer-provider relationship provides adequate protection for the services rendered. The extent and nature of this chain of trust varies based on the relationships between organizations and the external providers. Organizations document the basis for trust relationships so the relationships can be monitored over time. External information system services documentation includes government, service providers, end user security roles and responsibilities, and service-level agreements. Service-level agreements define expectations of performance for security controls, describe measurable outcomes, and identify remedies and response requirements for identified instances of noncompliance.

Related Controls: *CA-3*, *IR-7*, *PS-7*

SA-9a.

Requires that providers of external information system services comply with organizational information security requirements and employ [Assignment: organization-defined security controls] in accordance with applicable federal laws, Executive Orders, directives, policies, regulations, standards, and guidance;

SA-9b.

Defines and documents government oversight and user roles and responsibilities with regard to external information system services; and

SA-9c.

Employs [Assignment: organization-defined processes, methods, and techniques] to monitor security control compliance by external service providers on an ongoing basis.

SA-9 (1) : RISK ASSESSMENTS / ORGANIZATIONAL APPROVALS

The organization:

Note: Dedicated information security services include, for example, incident monitoring, analysis and response, operation of information security-related devices such as firewalls, or key management services.

Related Controls: *CA-6*, *RA-3*

SA-9 (1)(a)

Conducts an organizational assessment of risk prior to the acquisition or outsourcing of dedicated information security services; and

SA-9 (1)(b)

Ensures that the acquisition or outsourcing of dedicated information security services is approved by [Assignment: organization-defined personnel or roles].

SA-9 (2) : IDENTIFICATION OF FUNCTIONS / PORTS / PROTOCOLS / SERVICES

Baseline-Impact: *MODERATE*, **HIGH**

The organization requires providers of [Assignment: organization-defined external information system services] to identify the functions, ports, protocols, and other services required for the use of such services.

Note: Information from external service providers regarding the specific functions, ports, protocols, and services used in the provision of such services can be particularly useful when the need arises to understand the trade-offs involved in restricting certain functions/services or blocking certain ports/protocols.

Related Controls: *CM-7*

SA-9 (3) : ESTABLISH / MAINTAIN TRUST RELATIONSHIP WITH PROVIDERS

> The organization establishes, documents, and maintains trust relationships with external service providers based on [Assignment: organization-defined security requirements, properties, factors, or conditions defining acceptable trust relationships].

Note: The degree of confidence that the risk from using external services is at an acceptable level depends on the trust that organizations place in the external providers, individually or in combination. Trust relationships can help organization to gain increased levels of confidence that participating service providers are providing adequate protection for the services rendered. Such relationships can be complicated due to the number of potential entities participating in the consumer-provider interactions, subordinate relationships and levels of trust, and the types of interactions between the parties. In some cases, the degree of trust is based on the amount of direct control organizations are able to exert on external service providers with regard to employment of security controls necessary for the protection of the service/information and the evidence brought forth as to the effectiveness of those controls. The level of control is typically established by the terms and conditions of the contracts or service-level agreements and can range from extensive control (e.g., negotiating contracts or agreements that specify security requirements for the providers) to very limited control (e.g., using contracts or service-level agreements to obtain commodity services such as commercial telecommunications services). In other cases, levels of trust are based on factors that convince organizations that required security controls have been employed and that determinations of control effectiveness exist. For example, separately authorized external information system services provided to organizations through well-established business relationships may provide degrees of trust in such services within the tolerable risk range of the organizations using the services. External service providers may also outsource selected services to other external entities, making the trust relationship more difficult and complicated to manage. Depending on the nature of the services, organizations may find it very difficult to place significant trust in external providers. This is not due to any inherent untrustworthiness on the part of providers, but to the intrinsic level of risk in the services.

SA-9 (4) : CONSISTENT INTERESTS OF CONSUMERS AND PROVIDERS

> The organization employs [Assignment: organization-defined security safeguards] to ensure that the interests of [Assignment: organization-defined external service providers] are consistent with and reflect organizational interests.

Note: As organizations increasingly use external service providers, the possibility exists that the interests of the service providers may diverge from organizational interests. In such situations, simply having the correct technical, procedural, or operational safeguards in place may not be sufficient if the service providers that implement and control those safeguards are not operating in a manner consistent with the interests of the consuming organizations. Possible actions that organizations might take to address such concerns include, for example, requiring background checks for selected service provider personnel, examining ownership records, employing only trustworthy service providers (i.e., providers with which organizations have had positive experiences), and conducting periodic/unscheduled visits to service provider facilities.

SA-9 (5) : PROCESSING, STORAGE, AND SERVICE LOCATION

> The organization restricts the location of [Selection (one or more): information processing; information/data; information system services] to [Assignment: organization-defined locations] based on [Assignment: organization-defined requirements or conditions].

Note: The location of information processing, information/data storage, or information system services that are critical to organizations can have a direct impact on the ability of those organizations to successfully execute their missions/business functions. This situation exists when external providers control the location of processing, storage or services. The criteria external providers use for the selection of processing, storage, or service locations may be different from organizational criteria. For example, organizations may want to ensure that data/information storage locations are restricted to certain locations to facilitate incident response activities (e.g., forensic analyses, after-the-fact investigations) in case of information security breaches/compromises. Such incident response activities may be adversely affected by the governing laws or protocols in the locations where processing and storage occur and/or the locations from which information system services emanate.

Control Family: SYSTEM AND SERVICES ACQUISITION

SA-10 : DEVELOPER CONFIGURATION MANAGEMENT

Priority: P1

Baseline-Impact: *MODERATE*, **HIGH**

> The organization requires the developer of the information system, system component, or information system service to:

Note: This control also applies to organizations conducting internal information systems development and integration. Organizations consider the quality and completeness of the configuration management activities conducted by developers as evidence of applying effective security safeguards. Safeguards include, for example, protecting from unauthorized modification or destruction, the master copies of all material used to generate security-relevant portions of the system hardware, software, and firmware. Maintaining the integrity of changes to the information system, information system component, or information system service requires configuration control throughout the system development life cycle to track authorized changes and prevent unauthorized changes. Configuration items that are placed under configuration management (if existence/use is required by other security controls) include: the formal model; the functional, high-level, and low-level design specifications; other design data; implementation documentation; source code and hardware schematics; the running version of the object code; tools for comparing new versions of security-relevant hardware descriptions and software/firmware source code with previous versions; and test fixtures and documentation. Depending on the mission/business needs of organizations and the nature of the contractual relationships in place, developers may provide configuration management support during the operations and maintenance phases of the life cycle.

Related Controls: *CM-3*, *CM-4*, *CM-9*, *SA-12*, *SI-2*

SA-10a.

Perform configuration management during system, component, or service [Selection (one or more): design; development; implementation; operation];

SA-10b.

Document, manage, and control the integrity of changes to [Assignment: organization-defined configuration items under configuration management];

SA-10c.

Implement only organization-approved changes to the system, component, or service;

SA-10d.

Document approved changes to the system, component, or service and the potential security impacts of such changes; and

SA-10e.

Track security flaws and flaw resolution within the system, component, or service and report findings to [Assignment: organization-defined personnel].

SA-10 (1) : SOFTWARE / FIRMWARE INTEGRITY VERIFICATION

The organization requires the developer of the information system, system component, or information system service to enable integrity verification of software and firmware components.

Note: This control enhancement allows organizations to detect unauthorized changes to software and firmware components through the use of tools, techniques, and/or mechanisms provided by developers. Integrity checking mechanisms can also address counterfeiting of software and firmware components. Organizations verify the integrity of software and firmware components, for example, through secure one-way hashes provided by developers. Delivered software and firmware components also include any updates to such components.

Related Controls: *SI-7*

SA-10 (2) : ALTERNATIVE CONFIGURATION MANAGEMENT PROCESSES

The organization provides an alternate configuration management process using organizational personnel in the absence of a dedicated developer configuration management team.

Note: Alternate configuration management processes may be required, for example, when organizations use commercial off-the-shelf (COTS) information technology products. Alternate configuration management processes include organizational personnel that: (i) are responsible for reviewing/approving proposed changes to information systems, system components, and information system services; and (ii) conduct security impact analyses prior to the implementation of any changes to systems, components, or services (e.g., a configuration control board that considers security impacts of changes during development and includes representatives of both the organization and the developer, when applicable).

SA-10 (3) : HARDWARE INTEGRITY VERIFICATION

> The organization requires the developer of the information system, system component, or information system service to enable integrity verification of hardware components.

Note: This control enhancement allows organizations to detect unauthorized changes to hardware components through the use of tools, techniques, and/or mechanisms provided by developers. Organizations verify the integrity of hardware components, for example, with hard-to-copy labels and verifiable serial numbers provided by developers, and by requiring the implementation of anti-tamper technologies. Delivered hardware components also include updates to such components.

Related Controls: *SI-7*

SA-10 (4) : TRUSTED GENERATION

> The organization requires the developer of the information system, system component, or information system service to employ tools for comparing newly generated versions of security-relevant hardware descriptions and software/firmware source and object code with previous versions.

Note: This control enhancement addresses changes to hardware, software, and firmware components between versions during development. In contrast, SA-10 (1) and SA-10 (3) allow organizations to detect unauthorized changes to hardware, software, and firmware components through the use of tools, techniques, and/or mechanisms provided by developers.

SA-10 (5) : MAPPING INTEGRITY FOR VERSION CONTROL

> The organization requires the developer of the information system, system component, or information system service to maintain the integrity of the mapping between the master build data (hardware drawings and software/firmware code) describing the current version of security-relevant hardware, software, and firmware and the on-site master copy of the data for the current version.

Note: This control enhancement addresses changes to hardware, software, and firmware components during initial development and during system life cycle updates. Maintaining the integrity between the master copies of security-relevant hardware, software, and firmware (including designs and source code) and the equivalent data in master copies on-site in operational environments is essential to ensure the availability of organizational information systems supporting critical missions and/or business functions.

SA-10 (6) : TRUSTED DISTRIBUTION

> The organization requires the developer of the information system, system component, or information system service to execute procedures for ensuring that security-relevant hardware, software, and firmware updates distributed to the organization are exactly as specified by the master copies.

Note: The trusted distribution of security-relevant hardware, software, and firmware updates helps to ensure that such updates are faithful representations of the master copies maintained by the developer and have not been tampered with during distribution.

Control Family: SYSTEM AND SERVICES ACQUISITION

SA-11 : DEVELOPER SECURITY TESTING AND EVALUATION

Priority: P1

Baseline-Impact: *MODERATE*, **HIGH**

> The organization requires the developer of the information system, system component, or information system service to:

Note: Developmental security testing/evaluation occurs at all post-design phases of the system development life cycle. Such testing/evaluation confirms that the required security controls are implemented correctly, operating as intended, enforcing the desired security policy, and meeting established security requirements. Security properties of information systems may be affected by the interconnection of system components or changes to those components. These interconnections or changes (e.g., upgrading or replacing applications and operating systems) may adversely affect previously implemented security controls. This control provides additional types of security testing/evaluation that developers can conduct to reduce or eliminate potential flaws. Testing custom software applications may require approaches such as static analysis, dynamic analysis, binary analysis, or a hybrid of the three approaches. Developers can employ these analysis approaches in a variety of tools (e.g., web-based application scanners, static analysis tools, binary analyzers) and in source code reviews. Security assessment plans provide the specific activities that developers plan to carry out including the types of analyses, testing, evaluation, and reviews of software and firmware components, the degree of rigor to be applied, and the types of artifacts produced during those processes. The depth of security testing/evaluation refers to the rigor and level of detail associated with the assessment process (e.g., black box, gray box, or white box testing). The coverage of security testing/evaluation refers to the scope (i.e., number and type) of the artifacts included in the assessment process. Contracts specify the acceptance criteria for security assessment plans, flaw remediation processes, and the evidence that the plans/processes have been diligently applied. Methods for reviewing and protecting assessment plans, evidence, and documentation are commensurate with the security category or classification level of the information system. Contracts may specify documentation protection requirements.

Related Controls: *CA-2*, *CM-4*, *SA-3*, *SA-4*, *SA-5*, *SI-2*

SA-11a.

Create and implement a security assessment plan;

SA-11b.

Perform [Selection (one or more): unit; integration; system; regression] testing/evaluation at [Assignment: organization-defined depth and coverage];

SA-11c.

Produce evidence of the execution of the security assessment plan and the results of the security testing/evaluation;

SA-11d.

Implement a verifiable flaw remediation process; and

SA-11e.

Correct flaws identified during security testing/evaluation.

SA-11 (1) : STATIC CODE ANALYSIS

The organization requires the developer of the information system, system component, or information system service to employ static code analysis tools to identify common flaws and document the results of the analysis.

Note: Static code analysis provides a technology and methodology for security reviews. Such analysis can be used to identify security vulnerabilities and enforce security coding practices. Static code analysis is most effective when used early in the development process, when each code change can be automatically scanned for potential weaknesses. Static analysis can provide clear remediation guidance along with defects to enable developers to fix such defects. Evidence of correct implementation of static analysis can include, for example, aggregate defect density for critical defect types, evidence that defects were inspected by developers or security professionals, and evidence that defects were fixed. An excessively high density of ignored findings (commonly referred to as ignored or false positives) indicates a potential problem with the analysis process or tool. In such cases, organizations weigh the validity of the evidence against evidence from other sources.

SA-11 (2) : THREAT AND VULNERABILITY ANALYSES

The organization requires the developer of the information system, system component, or information system service to perform threat and vulnerability analyses and subsequent testing/evaluation of the as-built system, component, or service.

Note: Applications may deviate significantly from the functional and design specifications created during the requirements and design phases of the system development life cycle. Therefore, threat and vulnerability analyses of information systems, system components, and information system services prior to delivery are critical to the effective operation of those systems, components, and services. Threat and vulnerability analyses at this phase of the life cycle help to ensure that design or implementation changes have been accounted for, and that any new vulnerabilities created as a result of those changes have been reviewed and mitigated.

Related Controls: *PM-15*, *RA-5*

SA-11 (3) : INDEPENDENT VERIFICATION OF ASSESSMENT PLANS / EVIDENCE

The organization:

Note: Independent agents have the necessary qualifications (i.e., expertise, skills, training, and experience) to verify the correct implementation of developer security assessment plans.

Related Controls: *AT-3*, *CA-7*, *RA-5*, *SA-12*

SA-11 (3)(a)

Requires an independent agent satisfying [Assignment: organization-defined independence criteria] to verify the correct implementation of the developer security assessment plan and the evidence produced during security testing/evaluation; and

SA-11 (3)(b)

Ensures that the independent agent is either provided with sufficient information to complete the verification process or granted the authority to obtain such information.

SA-11 (4) : MANUAL CODE REVIEWS

The organization requires the developer of the information system, system component, or information system service to perform a manual code review of [Assignment: organization-defined specific code] using [Assignment: organization-defined processes, procedures, and/or techniques].

Note: Manual code reviews are usually reserved for the critical software and firmware components of information systems. Such code reviews are uniquely effective at identifying weaknesses that require knowledge of the applications requirements or context which are generally unavailable to more automated analytic tools and techniques such as static or dynamic analysis. Components benefiting from manual review include for example, verifying access control matrices against application controls and reviewing more detailed aspects of cryptographic implementations and controls.

SA-11 (5) : PENETRATION TESTING

> The organization requires the developer of the information system, system component, or information system service to perform penetration testing at [Assignment: organization-defined breadth/depth] and with [Assignment: organization-defined constraints].

Note: Penetration testing is an assessment methodology in which assessors, using all available information technology product and/or information system documentation (e.g., product/system design specifications, source code, and administrator/operator manuals) and working under specific constraints, attempt to circumvent implemented security features of information technology products and information systems. Penetration testing can include, for example, white, gray, or black box testing with analyses performed by skilled security professionals simulating adversary actions. The objective of penetration testing is to uncover potential vulnerabilities in information technology products and information systems resulting from implementation errors, configuration faults, or other operational deployment weaknesses or deficiencies. Penetration tests can be performed in conjunction with automated and manual code reviews to provide greater levels of analysis than would ordinarily be possible.

SA-11 (6) : ATTACK SURFACE REVIEWS

> The organization requires the developer of the information system, system component, or information system service to perform attack surface reviews.

Note: Attack surfaces of information systems are exposed areas that make those systems more vulnerable to cyber attacks. This includes any accessible areas where weaknesses or deficiencies in information systems (including the hardware, software, and firmware components) provide opportunities for adversaries to exploit vulnerabilities. Attack surface reviews ensure that developers: (i) analyze both design and implementation changes to information systems; and (ii) mitigate attack vectors generated as a result of the changes. Correction of identified flaws includes, for example, deprecation of unsafe functions.

SA-11 (7) : VERIFY SCOPE OF TESTING / EVALUATION

> The organization requires the developer of the information system, system component, or information system service to verify that the scope of security testing/evaluation provides complete coverage of required security controls at [Assignment: organization-defined depth of testing/evaluation].

Note: Verifying that security testing/evaluation provides complete coverage of required security controls can be accomplished by a variety of analytic techniques ranging from informal to formal. Each of these techniques provides an increasing level of assurance corresponding to the degree of formality of the analysis. Rigorously demonstrating security control coverage at the highest levels of assurance can be provided by the use of formal modeling and analysis techniques including correlation between control implementation and corresponding test cases.

SA-11 (8) : DYNAMIC CODE ANALYSIS

> The organization requires the developer of the information system, system component, or information system service to employ dynamic code analysis tools to identify common flaws and document the results of the analysis.

Note: Dynamic code analysis provides run-time verification of software programs, using tools capable of monitoring programs for memory corruption, user privilege issues, and other potential security problems. Dynamic code analysis employs run-time tools to help to ensure that security functionality performs in the manner in which it was designed. A specialized type of dynamic analysis, known as fuzz testing, induces program failures by deliberately introducing malformed or random data into software programs. Fuzz testing strategies derive from the intended use of applications and the functional and design specifications for the applications. To understand the scope of dynamic code analysis and hence the assurance provided, organizations may also consider conducting code coverage analysis (checking the degree to which the code has been tested using metrics such as percent of subroutines tested or percent of program statements called during execution of the test suite) and/or concordance analysis (checking for words that are out of place in software code such as non-English language words or derogatory terms).

Control Family: SYSTEM AND SERVICES ACQUISITION

SA-12 : SUPPLY CHAIN PROTECTION

Priority: P1

Baseline-Impact: HIGH

> The organization protects against supply chain threats to the information system, system component, or information system service by employing [Assignment: organization-defined security safeguards] as part of a comprehensive, defense-in-breadth information security strategy.

Note: Information systems (including system components that compose those systems) need to be protected throughout the system development life cycle (i.e., during design, development, manufacturing, packaging, assembly, distribution, system integration, operations, maintenance, and retirement). Protection of organizational information systems is accomplished through threat awareness, by the identification, management, and reduction of vulnerabilities at each phase of the life cycle and the use of complementary, mutually reinforcing strategies to respond to risk. Organizations consider implementing a standardized process to address supply chain risk with respect to information systems and system components, and to educate the acquisition workforce on threats, risk, and required security controls. Organizations use the acquisition/procurement processes to require supply chain entities to implement necessary security safeguards to: (i) reduce the likelihood of unauthorized modifications at each stage in the supply chain; and (ii) protect information systems and information system components, prior to taking delivery of such systems/components. This control also applies to information system services. Security safeguards include, for example: (i) security controls for development systems, development facilities, and external connections to development systems; (ii) vetting development personnel; and (iii) use of tamper-evident packaging during shipping/warehousing. Methods for reviewing and protecting development plans, evidence, and documentation are commensurate with the security category or classification level of the information system. Contracts may specify documentation protection requirements.

Related Controls: *AT-3*, *CM-8*, *IR-4*, *PE-16*, *PL-8*, *SA-3*, *SA-4*, *SA-8*, *SA-10*, *SA-14*, *SA-15*, *SA-18*, *SA-19*, *SC-29*, *SC-30*, *SC-38*, *SI-7*

SA-12 (1) : ACQUISITION STRATEGIES / TOOLS / METHODS

> The organization employs [Assignment: organization-defined tailored acquisition strategies, contract tools, and procurement methods] for the purchase of the information system, system component, or information system service from suppliers.

Note: The use of acquisition and procurement processes by organizations early in the system development life cycle provides an important vehicle to protect the supply chain. Organizations use available all-source intelligence analysis to inform the tailoring of acquisition strategies, tools, and methods. There are a number of different tools and techniques available (e.g., obscuring the end use of an information system or system component, using blind or filtered buys). Organizations also consider creating incentives for suppliers who: (i) implement required security safeguards; (ii) promote transparency into their organizational processes and security practices; (iii) provide additional vetting of the processes and security practices of subordinate suppliers, critical information system components, and services; (iv) restrict purchases from specific suppliers or countries; and (v) provide contract language regarding the prohibition of tainted or counterfeit components. In addition, organizations consider minimizing the time between purchase decisions and required delivery to limit opportunities for adversaries to corrupt information system components or products. Finally, organizations can use trusted/controlled distribution, delivery, and warehousing options to reduce supply chain risk (e.g., requiring tamper-evident packaging of information system components during shipping and warehousing).

Related Controls: *SA-19*

SA-12 (2) : SUPPLIER REVIEWS

> The organization conducts a supplier review prior to entering into a contractual agreement to acquire the information system, system component, or information system service.

Note: Supplier reviews include, for example: (i) analysis of supplier processes used to design, develop, test, implement, verify, deliver, and support information systems, system components, and information system services; and (ii) assessment of supplier training and experience in developing systems, components, or services with the required security capability. These reviews provide organizations with increased levels of visibility into supplier activities during the system development life cycle to promote more effective supply chain risk management. Supplier reviews can also help to determine whether primary suppliers have security safeguards in place and a practice for vetting subordinate suppliers, for example, second- and third-tier suppliers, and any subcontractors.

SA-12 (3) : TRUSTED SHIPPING AND WAREHOUSING

> [Withdrawn: Incorporated into SA-12 (1)].

SA-12 (4) : DIVERSITY OF SUPPLIERS

> [Withdrawn: Incorporated into SA-12 (13)].

SA-12 (5) : LIMITATION OF HARM

> The organization employs [Assignment: organization-defined security safeguards] to limit harm from potential adversaries identifying and targeting the organizational supply chain.

Note: Supply chain risk is part of the advanced persistent threat (APT). Security safeguards and countermeasures to reduce the probability of adversaries successfully identifying and targeting the supply chain include, for example: (i) avoiding the purchase of custom configurations to reduce the risk of acquiring information systems, components, or products that have been corrupted via supply chain actions targeted at specific organizations; (ii) employing a diverse set of suppliers to limit the potential harm from any given supplier in the supply chain; (iii) employing approved vendor lists with standing reputations in industry, and (iv) using procurement carve outs (i.e., exclusions to commitments or obligations).

SA-12 (6) : MINIMIZING PROCUREMENT TIME

> [Withdrawn: Incorporated into SA-12 (1)].

SA-12 (7) : ASSESSMENTS PRIOR TO SELECTION / ACCEPTANCE / UPDATE

> The organization conducts an assessment of the information system, system component, or information system service prior to selection, acceptance, or update.

Note: Assessments include, for example, testing, evaluations, reviews, and analyses. Independent, third-party entities or organizational personnel conduct assessments of systems, components, products, tools, and services. Organizations conduct assessments to uncover unintentional vulnerabilities and intentional vulnerabilities including, for example, malicious code, malicious processes, defective software, and counterfeits. Assessments can include, for example, static analyses, dynamic analyses, simulations, white, gray, and black box testing, fuzz testing, penetration testing, and ensuring that components or services are genuine (e.g., using tags, cryptographic hash verifications, or digital signatures). Evidence generated during security assessments is documented for follow-on actions carried out by organizations.

Related Controls: *CA-2*, *SA-11*

SA-12 (8) : USE OF ALL-SOURCE INTELLIGENCE

> The organization uses all-source intelligence analysis of suppliers and potential suppliers of the information system, system component, or information system service.

Note: All-source intelligence analysis is employed by organizations to inform engineering, acquisition, and risk management decisions. All-source intelligence consists of intelligence products and/or organizations and activities that incorporate all sources of information, most frequently including human intelligence, imagery intelligence, measurement and signature intelligence, signals intelligence, and open source data in the production of finished intelligence. Where available, such information is used to analyze the risk of both intentional and unintentional vulnerabilities from development, manufacturing, and delivery processes, people, and the environment. This review is performed on suppliers at multiple tiers in the supply chain sufficient to manage risks.

Related Controls: *SA-15*

SA-12 (9) : OPERATIONS SECURITY

> The organization employs [Assignment: organization-defined Operations Security (OPSEC) safeguards] in accordance with classification guides to protect supply chain-related information for the information system, system component, or information system service.

Note: Supply chain information includes, for example: user identities; uses for information systems, information system components, and information system services; supplier identities; supplier processes; security requirements; design specifications; testing and evaluation results; and system/component configurations. This control enhancement expands the scope of OPSEC to include suppliers and potential suppliers. OPSEC is a process of identifying critical information and subsequently analyzing friendly actions attendant to operations and other activities to: (i) identify those actions that can be observed by potential adversaries; (ii) determine indicators that adversaries might obtain that could be interpreted or pieced together to derive critical information in sufficient time to cause harm to organizations; (iii) implement safeguards or countermeasures to eliminate or reduce to an acceptable level, exploitable vulnerabilities; and (iv) consider how aggregated information may compromise the confidentiality of users or uses of the supply chain. OPSEC may require organizations to withhold critical mission/business information from suppliers and may include the use of intermediaries to hide the end use, or users, of information systems, system components, or information system services.

SA-12 (10) : VALIDATE AS GENUINE AND NOT ALTERED

> The organization employs [Assignment: organization-defined security safeguards] to validate that the information system or system component received is genuine and has not been altered.

Note: For some information system components, especially hardware, there are technical means to help determine if the components are genuine or have been altered. Security safeguards used to validate the authenticity of information systems and information system components include, for example, optical/nanotechnology tagging and side-channel analysis. For hardware, detailed bill of material information can highlight the elements with embedded logic complete with component and production location.

SA-12 (11) : PENETRATION TESTING / ANALYSIS OF ELEMENTS, PROCESSES, AND ACTORS

> The organization employs [Selection (one or more): organizational analysis, independent third-party analysis, organizational penetration testing, independent third-party penetration testing] of [Assignment: organization-defined supply chain elements, processes, and actors] associated with the information system, system component, or information system service.

Note: This control enhancement addresses analysis and/or testing of the supply chain, not just delivered items. Supply chain elements are information technology products or product components that contain programmable logic and that are critically important to information system functions. Supply chain processes include, for example: (i) hardware, software, and firmware development processes; (ii) shipping/handling procedures; (iii) personnel and physical security programs; (iv) configuration management tools/measures to maintain provenance; or (v) any other programs, processes, or procedures associated with the production/distribution of supply chain elements. Supply chain actors are individuals with specific roles and responsibilities in the supply chain. The evidence generated during analyses and testing of supply chain elements, processes, and actors is documented and used to inform organizational risk management activities and decisions.

Related Controls: *RA-5*

SA-12 (12) : INTER-ORGANIZATIONAL AGREEMENTS

> The organization establishes inter-organizational agreements and procedures with entities involved in the supply chain for the information system, system component, or information system service.

Note: The establishment of inter-organizational agreements and procedures provides for notification of supply chain compromises. Early notification of supply chain compromises that can potentially adversely affect or have adversely affected organizational information systems, including critical system components, is essential for organizations to provide appropriate responses to such incidents.

SA-12 (13) : CRITICAL INFORMATION SYSTEM COMPONENTS

> The organization employs [Assignment: organization-defined security safeguards] to ensure an adequate supply of [Assignment: organization-defined critical information system components].

Note: Adversaries can attempt to impede organizational operations by disrupting the supply of critical information system components or corrupting supplier operations. Safeguards to ensure adequate supplies of critical information system components include, for example: (i) the use of multiple suppliers throughout the supply chain for the identified critical components; and (ii) stockpiling of spare components to ensure operation during mission-critical times.

SA-12 (14) : IDENTITY AND TRACEABILITY

> The organization establishes and retains unique identification of [Assignment: organization-defined supply chain elements, processes, and actors] for the information system, system component, or information system service.

Note: Knowing who and what is in the supply chains of organizations is critical to gaining visibility into what is happening within such supply chains, as well as monitoring and identifying high-risk events and activities. Without reasonable visibility and traceability into supply chains (i.e., elements, processes, and actors), it is very difficult for organizations to understand and therefore manage risk, and to reduce the likelihood of adverse events. Uniquely identifying acquirer and integrator roles, organizations, personnel, mission and element processes, testing and evaluation procedures, delivery mechanisms, support mechanisms, communications/delivery paths, and disposal/final disposition activities as well as the components and tools used, establishes a foundational identity structure for assessment of supply chain activities. For example, labeling (using serial numbers) and tagging (using radio-frequency identification [RFID] tags) individual supply chain elements including software packages, modules, and hardware devices, and processes associated with those elements can be used for this purpose. Identification methods are sufficient to support the provenance in the event of a supply chain issue or adverse supply chain event.

SA-12 (15) : PROCESSES TO ADDRESS WEAKNESSES OR DEFICIENCIES

The organization establishes a process to address weaknesses or deficiencies in supply chain elements identified during independent or organizational assessments of such elements.

Note: Evidence generated during independent or organizational assessments of supply chain elements (e.g., penetration testing, audits, verification/validation activities) is documented and used in follow-on processes implemented by organizations to respond to the risks related to the identified weaknesses and deficiencies. Supply chain elements include, for example, supplier development processes and supplier distribution systems.

Control Family: SYSTEM AND SERVICES ACQUISITION

SA-13 : TRUSTWORTHINESS

Priority: P0

The organization:

Note:

This control helps organizations to make explicit trustworthiness decisions when designing, developing, and implementing information systems that are needed to conduct critical organizational missions/business functions. Trustworthiness is a characteristic/property of an information system that expresses the degree to which the system can be expected to preserve the confidentiality, integrity, and availability of the information it processes, stores, or transmits. Trustworthy information systems are systems that are capable of being trusted to operate within defined levels of risk despite the environmental disruptions, human errors, and purposeful attacks that are expected to occur in the specified environments of operation. Trustworthy systems are important to mission/business success. Two factors affecting the trustworthiness of information systems include: (i) security functionality (i.e., the security features, functions, and/or mechanisms employed within the system and its environment of operation); and (ii) security assurance (i.e., the grounds for confidence that the security functionality is effective in its application). Developers, implementers, operators, and maintainers of organizational information systems can increase the level of assurance (and trustworthiness), for example, by employing well-defined security policy models, structured and rigorous hardware, software, and firmware development techniques, sound system/security engineering principles, and secure configuration settings (defined by a set of assurance-related security controls in Appendix E).

Assurance is also based on the assessment of evidence produced during the system development life cycle. Critical missions/business functions are supported by high-impact systems and the associated assurance requirements for such systems. The additional assurance controls in Table E-4 in Appendix E (designated as optional) can be used to develop and implement high-assurance solutions for specific information systems and system components using the concept of overlays described in Appendix I. Organizations select assurance overlays that have been developed, validated, and approved for community adoption (e.g., cross-organization, governmentwide), limiting the development of such overlays on an organization-by-organization basis. Organizations can conduct criticality analyses as described in SA-14, to determine the information systems, system components, or information system services that require high-assurance solutions. Trustworthiness requirements and assurance overlays can be described in the security plans for organizational information systems.

Related Controls: *RA-2*, *SA-4*, *SA-8*, *SA-14*, *SC-3*

SA-13a.

Describes the trustworthiness required in the [Assignment: organization-defined information system, information system component, or information system service] supporting its critical missions/business functions; and

SA-13b.

Implements [Assignment: organization-defined assurance overlay] to achieve such trustworthiness.

Control Family: SYSTEM AND SERVICES ACQUISITION

SA-14 : CRITICALITY ANALYSIS

Priority: P0

The organization identifies critical information system components and functions by performing a criticality analysis for [Assignment: organization-defined information systems, information system components, or information system services] at [Assignment: organization-defined decision points in the system development life cycle].

Note: Criticality analysis is a key tenet of supply chain risk management and informs the prioritization of supply chain protection activities such as attack surface reduction, use of all-source intelligence, and tailored acquisition strategies. Information system engineers can conduct an end-to-end functional decomposition of an information system to identify mission-critical functions and components. The functional decomposition includes the identification of core organizational missions supported by the system, decomposition into the specific functions to perform those missions, and traceability to the hardware, software, and firmware components that implement those functions, including when the functions are shared by many components within and beyond the information system boundary. Information system components that allow for unmediated access to critical components or functions are considered critical due to the inherent vulnerabilities such components create. Criticality is assessed in terms of the impact of the function or component failure on the ability of the component to complete the organizational missions supported by the information system. A criticality analysis is performed whenever an architecture or design is being developed or modified, including upgrades.

Related Controls: *CP-2*, *PL-2*, *PL-8*, *PM-1*, *SA-8*, *SA-12*, *SA-13*, *SA-15*, *SA-20*

SA-14 (1) : CRITICAL COMPONENTS WITH NO VIABLE ALTERNATIVE SOURCING

[Withdrawn: Incorporated into SA-20].

Control Family: SYSTEM AND SERVICES ACQUISITION

SA-15 : DEVELOPMENT PROCESS, STANDARDS, AND TOOLS

Priority: P2

Baseline-Impact: HIGH

The organization:

Note: Development tools include, for example, programming languages and computer-aided design (CAD) systems. Reviews of development processes can include, for example, the use of maturity models to determine the potential effectiveness of such processes. Maintaining the integrity of changes to tools and processes enables accurate supply chain risk assessment and mitigation, and requires robust configuration control throughout the life cycle (including design, development, transport, delivery, integration, and maintenance) to track authorized changes and prevent unauthorized changes.

Related Controls: *SA-3*, *SA-8*

SA-15a.

Requires the developer of the information system, system component, or information system service to follow a documented development process that:

SA-15a.1.

Explicitly addresses security requirements;

SA-15a.2.

Identifies the standards and tools used in the development process;

SA-15a.3.

Documents the specific tool options and tool configurations used in the development process; and

SA-15a.4.

Documents, manages, and ensures the integrity of changes to the process and/or tools used in development; and

SA-15b.

Reviews the development process, standards, tools, and tool options/configurations [Assignment: organization-defined frequency] to determine if the process, standards, tools, and tool options/configurations selected and employed can satisfy [Assignment: organization-defined security requirements].

SA-15 (1) : QUALITY METRICS

The organization requires the developer of the information system, system component, or information system service to:

Note: Organizations use quality metrics to establish minimum acceptable levels of information system quality. Metrics may include quality gates which are collections of completion criteria or sufficiency standards representing the satisfactory execution of particular phases of the system development project. A quality gate, for example, may require the elimination of all compiler warnings or an explicit determination that the warnings have no impact on the effectiveness of required security capabilities. During the execution phases of development projects, quality gates provide clear, unambiguous indications of progress. Other metrics apply to the entire development project. These metrics can include defining the severity thresholds of vulnerabilities, for example, requiring no known vulnerabilities in the delivered information system with a Common Vulnerability Scoring System (CVSS) severity of Medium or High.

SA-15 (1)(a)

Define quality metrics at the beginning of the development process; and

SA-15 (1)(b)

Provide evidence of meeting the quality metrics [Selection (one or more): [Assignment: organization-defined frequency]; [Assignment: organization-defined program review milestones]; upon delivery].

SA-15 (2) : SECURITY TRACKING TOOLS

The organization requires the developer of the information system, system component, or information system service to select and employ a security tracking tool for use during the development process.

Note: Information system development teams select and deploy security tracking tools, including, for example, vulnerability/work item tracking systems that facilitate assignment, sorting, filtering, and tracking of completed work items or tasks associated with system development processes.

SA-15 (3) : CRITICALITY ANALYSIS

> The organization requires the developer of the information system, system component, or information system service to perform a criticality analysis at [Assignment: organization-defined breadth/depth] and at [Assignment: organization-defined decision points in the system development life cycle].

Note: This control enhancement provides developer input to the criticality analysis performed by organizations in SA-14. Developer input is essential to such analysis because organizations may not have access to detailed design documentation for information system components that are developed as commercial off-the-shelf (COTS) information technology products (e.g., functional specifications, high-level designs, low-level designs, and source code/hardware schematics).

Related Controls: *SA-4*, *SA-14*

SA-15 (4) : THREAT MODELING / VULNERABILITY ANALYSIS

> The organization requires that developers perform threat modeling and a vulnerability analysis for the information system at [Assignment: organization-defined breadth/depth] that:

Related Controls: *SA-4*

SA-15 (4)(a)

> Uses [Assignment: organization-defined information concerning impact, environment of operations, known or assumed threats, and acceptable risk levels];

SA-15 (4)(b)

> Employs [Assignment: organization-defined tools and methods]; and

SA-15 (4)(c)

> Produces evidence that meets [Assignment: organization-defined acceptance criteria].

SA-15 (5) : ATTACK SURFACE REDUCTION

> The organization requires the developer of the information system, system component, or information system service to reduce attack surfaces to [Assignment: organization-defined thresholds].

Note: Attack surface reduction is closely aligned with developer threat and vulnerability analyses and information system architecture and design. Attack surface reduction is a means of reducing risk to organizations by giving attackers less opportunity to exploit weaknesses or deficiencies (i.e., potential vulnerabilities) within information systems, information system components, and information system services. Attack surface reduction includes, for example, applying the principle of least privilege, employing layered defenses, applying the principle of least functionality (i.e., restricting ports, protocols, functions, and services), deprecating unsafe functions, and eliminating application programming interfaces (APIs) that are vulnerable to cyber attacks.

Related Controls: *CM-7*

SA-15 (6) : CONTINUOUS IMPROVEMENT

> The organization requires the developer of the information system, system component, or information system service to implement an explicit process to continuously improve the development process.

Note: Developers of information systems, information system components, and information system services consider the effectiveness/efficiency of current development processes for meeting quality objectives and addressing security capabilities in current threat environments.

SA-15 (7) : AUTOMATED VULNERABILITY ANALYSIS

> The organization requires the developer of the information system, system component, or information system service to:

Related Controls: *RA-5*

SA-15 (7)(a)

> Perform an automated vulnerability analysis using [Assignment: organization-defined tools];

SA-15 (7)(b)

> Determine the exploitation potential for discovered vulnerabilities;

SA-15 (7)(c)

> Determine potential risk mitigations for delivered vulnerabilities; and

SA-15 (7)(d)

> Deliver the outputs of the tools and results of the analysis to [Assignment: organization-defined personnel or roles].

SA-15 (8) : REUSE OF THREAT / VULNERABILITY INFORMATION

> The organization requires the developer of the information system, system component, or information system service to use threat modeling and vulnerability analyses from similar systems, components, or services to inform the current development process.

Note: Analysis of vulnerabilities found in similar software applications can inform potential design or implementation issues for information systems under development. Similar information systems or system components may exist within developer organizations. Authoritative vulnerability information is available from a variety of public and private sector sources including, for example, the National Vulnerability Database.

SA-15 (9) : USE OF LIVE DATA

> The organization approves, documents, and controls the use of live data in development and test environments for the information system, system component, or information system service.

Note: The use of live data in preproduction environments can result in significant risk to organizations. Organizations can minimize such risk by using test or dummy data during the development and testing of information systems, information system components, and information system services.

SA-15 (10) : INCIDENT RESPONSE PLAN

> The organization requires the developer of the information system, system component, or information system service to provide an incident response plan.

Note: The incident response plan for developers of information systems, system components, and information system services is incorporated into organizational incident response plans to provide the type of incident response information not readily available to organizations. Such information may be extremely helpful, for example, when organizations respond to vulnerabilities in commercial off-the-shelf (COTS) information technology products.

Related Controls: *IR-8*

SA-15 (11) : ARCHIVE INFORMATION SYSTEM / COMPONENT

> The organization requires the developer of the information system or system component to archive the system or component to be released or delivered together with the corresponding evidence supporting the final security review.

Note: Archiving relevant documentation from the development process can provide a readily available baseline of information that can be helpful during information system/component upgrades or modifications.

Control Family: SYSTEM AND SERVICES ACQUISITION

SA-16 : DEVELOPER-PROVIDED TRAINING

Priority: P2

Baseline-Impact: HIGH

> The organization requires the developer of the information system, system component, or information system service to provide [Assignment: organization-defined training] on the correct use and operation of the implemented security functions, controls, and/or mechanisms.

Note: This control applies to external and internal (in-house) developers. Training of personnel is an essential element to ensure the effectiveness of security controls implemented within organizational information systems. Training options include, for example, classroom-style training, web-based/computer-based training, and hands-on training.

Organizations can also request sufficient training materials from developers to conduct in-house training or offer self-training to organizational personnel. Organizations determine the type of training necessary and may require different types of training for different security functions, controls, or mechanisms.

Related Controls: *AT-2*, *AT-3*, *SA-5*

Control Family: SYSTEM AND SERVICES ACQUISITION

SA-17 : DEVELOPER SECURITY ARCHITECTURE AND DESIGN

Priority: P1

Baseline-Impact: HIGH

The organization requires the developer of the information system, system component, or information system service to produce a design specification and security architecture that:

Note: This control is primarily directed at external developers, although it could also be used for internal (in-house) development. In contrast, PL-8 is primarily directed at internal developers to help ensure that organizations develop an information security architecture and such security architecture is integrated or tightly coupled to the enterprise architecture. This distinction is important if/when organizations outsource the development of information systems, information system components, or information system services to external entities, and there is a requirement to demonstrate consistency with the organizations enterprise architecture and information security architecture.

Related Controls: *PL-8*, *PM-7*, *SA-3*, *SA-8*

SA-17a.

Is consistent with and supportive of the organizations security architecture which is established within and is an integrated part of the organizations enterprise architecture;

SA-17b.

Accurately and completely describes the required security functionality, and the allocation of security controls among physical and logical components; and

SA-17c.

Expresses how individual security functions, mechanisms, and services work together to provide required security capabilities and a unified approach to protection.

SA-17 (1) : FORMAL POLICY MODEL

The organization requires the developer of the information system, system component, or information system service to:

Note: Formal models describe specific behaviors or security policies using formal languages, thus enabling the correctness of those behaviors/policies to be formally proven. Not all components of information systems can be modeled, and generally, formal specifications are scoped to specific behaviors or policies of interest (e.g., nondiscretionary access control policies). Organizations choose the particular formal modeling language and approach based on the nature of the behaviors/policies to be described and the available tools. Formal modeling tools include, for example, Gypsy and Zed.

SA-17 (1)(a)

Produce, as an integral part of the development process, a formal policy model describing the [Assignment: organization-defined elements of organizational security policy] to be enforced; and

SA-17 (1)(b)

Prove that the formal policy model is internally consistent and sufficient to enforce the defined elements of the organizational security policy when implemented.

SA-17 (2) : SECURITY-RELEVANT COMPONENTS

The organization requires the developer of the information system, system component, or information system service to:

Note: Security-relevant hardware, software, and firmware represent the portion of the information system, component, or service that must be trusted to perform correctly in order to maintain required security properties.

Related Controls: *SA-5*

SA-17 (2)(a)

Define security-relevant hardware, software, and firmware; and

SA-17 (2)(b)

Provide a rationale that the definition for security-relevant hardware, software, and firmware is complete.

SA-17 (3) : FORMAL CORRESPONDENCE

The organization requires the developer of the information system, system component, or information system service to:

Note: Correspondence is an important part of the assurance gained through modeling. It demonstrates that the implementation is an accurate transformation of the model, and that any additional code or implementation details present have no impact on the behaviors or policies being modeled. Formal methods can be used to show that the high-level security properties are satisfied by the formal information system description, and that the formal system description is correctly implemented by a description of some lower level, for example a hardware description. Consistency between the formal top-level specification and the formal policy models is generally not amenable to being fully proven. Therefore, a combination of formal/informal methods may be needed to show such consistency. Consistency between the formal top-level specification and the implementation may require the use of an informal demonstration due to limitations in the applicability of formal methods to prove that the specification accurately reflects the implementation. Hardware, software, and firmware mechanisms strictly internal to security-relevant hardware, software, and firmware include, for example, mapping registers and direct memory input/output.

Related Controls: *SA-5*

SA-17 (3)(a)

Produce, as an integral part of the development process, a formal top-level specification that specifies the interfaces to security-relevant hardware, software, and firmware in terms of exceptions, error messages, and effects;

SA-17 (3)(b)

Show via proof to the extent feasible with additional informal demonstration as necessary, that the formal top-level specification is consistent with the formal policy model;

SA-17 (3)(c)

Show via informal demonstration, that the formal top-level specification completely covers the interfaces to security-relevant hardware, software, and firmware;

SA-17 (3)(d)

Show that the formal top-level specification is an accurate description of the implemented security-relevant hardware, software, and firmware; and

SA-17 (3)(e)

Describe the security-relevant hardware, software, and firmware mechanisms not addressed in the formal top-level specification but strictly internal to the security-relevant hardware, software, and firmware.

SA-17 (4) : INFORMAL CORRESPONDENCE

The organization requires the developer of the information system, system component, or information system service to:

Note: Correspondence is an important part of the assurance gained through modeling. It demonstrates that the implementation is an accurate transformation of the model, and that any additional code or implementation details present has no impact on the behaviors or policies being modeled. Consistency between the descriptive top-level specification (i.e., high-level/low-level design) and the formal policy model is generally not amenable to being fully proven. Therefore, a combination of formal/informal methods may be needed to show such consistency. Hardware, software, and firmware mechanisms strictly internal to security-relevant hardware, software, and firmware include, for example, mapping registers and direct memory input/output.

Related Controls: *SA-5*

SA-17 (4)(a)

Produce, as an integral part of the development process, an informal descriptive top-level specification that specifies the interfaces to security-relevant hardware, software, and firmware in terms of exceptions, error messages, and effects;

SA-17 (4)(b)

Show via [Selection: informal demonstration, convincing argument with formal methods as feasible] that the descriptive top-level specification is consistent with the formal policy model;

SA-17 (4)(c)

Show via informal demonstration, that the descriptive top-level specification completely covers the interfaces to security-relevant hardware, software, and firmware;

SA-17 (4)(d)

Show that the descriptive top-level specification is an accurate description of the interfaces to security-relevant hardware, software, and firmware; and

SA-17 (4)(e)

Describe the security-relevant hardware, software, and firmware mechanisms not addressed in the descriptive top-level specification but strictly internal to the security-relevant hardware, software, and firmware.

SA-17 (5) : CONCEPTUALLY SIMPLE DESIGN

The organization requires the developer of the information system, system component, or information system service to:

Related Controls: *SC-3*

SA-17 (5)(a)

Design and structure the security-relevant hardware, software, and firmware to use a complete, conceptually simple protection mechanism with precisely defined semantics; and

SA-17 (5)(b)

Internally structure the security-relevant hardware, software, and firmware with specific regard for this mechanism.

SA-17 (6) : STRUCTURE FOR TESTING

The organization requires the developer of the information system, system component, or information system service to structure security-relevant hardware, software, and firmware to facilitate testing.

Related Controls: *SA-11*

SA-17 (7) : STRUCTURE FOR LEAST PRIVILEGE

The organization requires the developer of the information system, system component, or information system service to structure security-relevant hardware, software, and firmware to facilitate controlling access with least privilege.

Related Controls: *AC-5*, *AC-6*

Control Family: SYSTEM AND SERVICES ACQUISITION

SA-18 : TAMPER RESISTANCE AND DETECTION

Priority: P0

The organization implements a tamper protection program for the information system, system component, or information system service.

Note: Anti-tamper technologies and techniques provide a level of protection for critical information systems, system components, and information technology products against a number of related threats including modification, reverse engineering, and substitution. Strong identification combined with tamper resistance and/or tamper detection is essential to protecting information systems, components, and products during distribution and when in use.

Related Controls: *PE-3*, *SA-12*, *SI-7*

SA-18 (1) : MULTIPLE PHASES OF SDLC

> The organization employs anti-tamper technologies and techniques during multiple phases in the system development life cycle including design, development, integration, operations, and maintenance.

Note: Organizations use a combination of hardware and software techniques for tamper resistance and detection. Organizations employ obfuscation and self-checking, for example, to make reverse engineering and modifications more difficult, time-consuming, and expensive for adversaries. Customization of information systems and system components can make substitutions easier to detect and therefore limit damage.

Related Controls: *SA-3*

SA-18 (2) : INSPECTION OF INFORMATION SYSTEMS, COMPONENTS, OR DEVICES

> The organization inspects [Assignment: organization-defined information systems, system components, or devices] [Selection (one or more): at random; at [Assignment: organization-defined frequency], upon [Assignment: organization-defined indications of need for inspection]] to detect tampering.

Note: This control enhancement addresses both physical and logical tampering and is typically applied to mobile devices, notebook computers, or other system components taken out of organization-controlled areas. Indications of need for inspection include, for example, when individuals return from travel to high-risk locations.

Related Controls: *SI-4*

Control Family: SYSTEM AND SERVICES ACQUISITION

SA-19 : COMPONENT AUTHENTICITY

Priority: P0

> The organization:

Note: Sources of counterfeit components include, for example, manufacturers, developers, vendors, and contractors. Anti-counterfeiting policy and procedures support tamper resistance and provide a level of protection against the introduction of malicious code. External reporting organizations include, for example, US-CERT.

Related Controls: *PE-3*, *SA-12*, *SI-7*

SA-19a.

Develops and implements anti-counterfeit policy and procedures that include the means to detect and prevent counterfeit components from entering the information system; and

SA-19b.

Reports counterfeit information system components to [Selection (one or more): source of counterfeit component; [Assignment: organization-defined external reporting organizations]; [Assignment: organization-defined personnel or roles]].

SA-19 (1) : ANTI-COUNTERFEIT TRAINING

The organization trains [Assignment: organization-defined personnel or roles] to detect counterfeit information system components (including hardware, software, and firmware).

SA-19 (2) : CONFIGURATION CONTROL FOR COMPONENT SERVICE / REPAIR

The organization maintains configuration control over [Assignment: organization-defined information system components] awaiting service/repair and serviced/repaired components awaiting return to service.

SA-19 (3) : COMPONENT DISPOSAL

The organization disposes of information system components using [Assignment: organization-defined techniques and methods].

Note: Proper disposal of information system components helps to prevent such components from entering the gray market.

SA-19 (4) : ANTI-COUNTERFEIT SCANNING

The organization scans for counterfeit information system components [Assignment: organization-defined frequency].

Control Family: SYSTEM AND SERVICES ACQUISITION

SA-20 : CUSTOMIZED DEVELOPMENT OF CRITICAL COMPONENTS

Priority: P0

The organization re-implements or custom develops [Assignment: organization-defined critical information system components].

Note: Organizations determine that certain information system components likely cannot be trusted due to specific threats to and vulnerabilities in those components, and for which there are no viable security controls to adequately mitigate the resulting risk. Re-implementation or custom development of such components helps to satisfy requirements for higher assurance. This is accomplished by initiating changes to system components (including hardware, software, and firmware) such that the standard attacks by adversaries are less likely to succeed. In situations where no alternative sourcing is available and organizations choose not to re-implement or custom develop critical information system components, additional safeguards can be employed (e.g., enhanced auditing, restrictions on source code and system utility access, and protection from deletion of system and application files.

Related Controls: *CP-2*, *SA-8*, *SA-14*

Control Family: SYSTEM AND SERVICES ACQUISITION

SA-21 : DEVELOPER SCREENING

Priority: P0

> The organization requires that the developer of [Assignment: organization-defined information system, system component, or information system service]:

Note: Because the information system, system component, or information system service may be employed in critical activities essential to the national and/or economic security interests of the United States, organizations have a strong interest in ensuring that the developer is trustworthy. The degree of trust required of the developer may need to be consistent with that of the individuals accessing the information system/component/service once deployed. Examples of authorization and personnel screening criteria include clearance, satisfactory background checks, citizenship, and nationality. Trustworthiness of developers may also include a review and analysis of company ownership and any relationships the company has with entities potentially affecting the quality/reliability of the systems, components, or services being developed.

Related Controls: *PS-3*, *PS-7*

SA-21a.

> Have appropriate access authorizations as determined by assigned [Assignment: organization-defined official government duties]; and

SA-21b.

> Satisfy [Assignment: organization-defined additional personnel screening criteria].

SA-21 (1) : VALIDATION OF SCREENING

> The organization requires the developer of the information system, system component, or information system service take [Assignment: organization-defined actions] to ensure that the required access authorizations and screening criteria are satisfied.

Note: Satisfying required access authorizations and personnel screening criteria includes, for example, providing a listing of all the individuals authorized to perform development activities on the selected information system, system component, or information system service so that organizations can validate that the developer has satisfied the necessary authorization and screening requirements.

Control Family: SYSTEM AND SERVICES ACQUISITION

SA-22 : UNSUPPORTED SYSTEM COMPONENTS

Priority: P0

The organization:

Note: Support for information system components includes, for example, software patches, firmware updates, replacement parts, and maintenance contracts. Unsupported components (e.g., when vendors are no longer providing critical software patches), provide a substantial opportunity for adversaries to exploit new weaknesses discovered in the currently installed components. Exceptions to replacing unsupported system components may include, for example, systems that provide critical mission/business capability where newer technologies are not available or where the systems are so isolated that installing replacement components is not an option.

Related Controls: *PL-2*, *SA-3*

SA-22a.

Replaces information system components when support for the components is no longer available from the developer, vendor, or manufacturer; and

SA-22b.

Provides justification and documents approval for the continued use of unsupported system components required to satisfy mission/business needs.

SA-22 (1) : ALTERNATIVE SOURCES FOR CONTINUED SUPPORT

The organization provides [Selection (one or more): in-house support; [Assignment: organization-defined support from external providers]] for unsupported information system components.

Note: This control enhancement addresses the need to provide continued support for selected information system components that are no longer supported by the original developers, vendors, or manufacturers when such components remain essential to mission/business operations. Organizations can establish in-house support, for example, by developing customized patches for critical software components or secure the services of external providers who through contractual relationships, provide ongoing support for the designated unsupported components. Such contractual relationships can include, for example, Open Source Software value-added vendors.

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-1 : SYSTEM AND COMMUNICATIONS PROTECTION POLICY AND PROCEDURES

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: This control addresses the establishment of policy and procedures for the effective implementation of selected security controls and control enhancements in the SC family. Policy and procedures reflect applicable federal laws, Executive Orders, directives, regulations, policies, standards, and guidance. Security program policies and procedures at the organization level may make the need for system-specific policies and procedures unnecessary. The policy can be included as part of the general information security policy for organizations or conversely, can be represented by multiple policies reflecting the complex nature of certain organizations. The procedures can be established for the security program in general and for particular information systems, if needed. The organizational risk management strategy is a key factor in establishing policy and procedures.

Related Controls: *PM-9*

SC-1a.

Develops, documents, and disseminates to [Assignment: organization-defined personnel or roles]:

SC-1a.1.

A system and communications protection policy that addresses purpose, scope, roles, responsibilities, management commitment, coordination among organizational entities, and compliance; and

SC-1a.2.

Procedures to facilitate the implementation of the system and communications protection policy and associated system and communications protection controls; and

SC-1b.

Reviews and updates the current:

SC-1b.1.

System and communications protection policy [Assignment: organization-defined frequency]; and

SC-1b.2.

System and communications protection procedures [Assignment: organization-defined frequency].

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-2 : APPLICATION PARTITIONING

Priority: P1

Baseline-Impact: *MODERATE*, **HIGH**

> The information system separates user functionality (including user interface services) from information system management functionality.

Note: Information system management functionality includes, for example, functions necessary to administer databases, network components, workstations, or servers, and typically requires privileged user access. The separation of user functionality from information system management functionality is either physical or logical. Organizations implement separation of system management-related functionality from user functionality by using different computers, different central processing units, different instances of operating systems, different network addresses, virtualization techniques, or combinations of these or other methods, as appropriate. This type of separation includes, for example, web administrative interfaces that use separate authentication methods for users of any other information system resources. Separation of system and user functionality may include isolating administrative interfaces on different domains and with additional access controls.

Related Controls: *SA-4*, *SA-8*, *SC-3*

SC-2 (1) : INTERFACES FOR NON-PRIVILEGED USERS

> The information system prevents the presentation of information system management-related functionality at an interface for non-privileged users.

Note: This control enhancement ensures that administration options (e.g., administrator privileges) are not available to general users (including prohibiting the use of the grey-out option commonly used to eliminate accessibility to such information). Such restrictions include, for example, not presenting administration options until users establish sessions with administrator privileges.

Related Controls: *AC-3*

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-3 : SECURITY FUNCTION ISOLATION

Priority: P1

Baseline-Impact: HIGH

> The information system isolates security functions from nonsecurity functions.

Note: The information system isolates security functions from nonsecurity functions by means of an isolation boundary (implemented via partitions and domains). Such isolation controls access to and protects the integrity of the hardware, software, and firmware that perform those security functions. Information systems implement code separation (i.e., separation of security functions from nonsecurity functions) in a number of ways, including, for example, through the provision of security kernels via processor rings or processor modes. For non-kernel code, security function isolation is often achieved through file system protections that serve to protect the code on disk, and address space protections that protect executing code. Information systems restrict access to security functions through the use of access control mechanisms and by implementing least privilege capabilities. While the ideal is for all of the code within the security function isolation boundary to only contain security-relevant code, it is sometimes necessary to include nonsecurity functions within the isolation boundary as an exception.

Related Controls: *AC-3*, *AC-6*, *SA-4*, *SA-5*, *SA-8*, *SA-13*, *SC-2*, *SC-7*, *SC-39*

SC-3 (1) : HARDWARE SEPARATION

> The information system utilizes underlying hardware separation mechanisms to implement security function isolation.

Note: Underlying hardware separation mechanisms include, for example, hardware ring architectures, commonly implemented within microprocessors, and hardware-enforced address segmentation used to support logically distinct storage objects with separate attributes (i.e., readable, writeable).

SC-3 (2) : ACCESS / FLOW CONTROL FUNCTIONS

> The information system isolates security functions enforcing access and information flow control from nonsecurity functions and from other security functions.

Note: Security function isolation occurs as a result of implementation; the functions can still be scanned and monitored. Security functions that are potentially isolated from access and flow control enforcement functions include, for example, auditing, intrusion detection, and anti-virus functions.

SC-3 (3) : MINIMIZE NONSECURITY FUNCTIONALITY

> The organization minimizes the number of nonsecurity functions included within the isolation boundary containing security functions.

Note: In those instances where it is not feasible to achieve strict isolation of nonsecurity functions from security functions, it is necessary to take actions to minimize the nonsecurity-relevant functions within the security function boundary. Nonsecurity functions contained within the isolation boundary are considered security-relevant because errors or maliciousness in such software, by virtue of being within the boundary, can impact the security functions of organizational information systems. The design objective is that the specific portions of information systems providing information security are of minimal size/complexity. Minimizing the number of nonsecurity functions in the security-relevant components of information systems allows designers and implementers to focus only on those functions which are necessary to provide the desired security capability (typically access enforcement). By minimizing nonsecurity functions within the isolation boundaries, the amount of code that must be trusted to enforce security policies is reduced, thus contributing to understandability.

SC-3 (4) : MODULE COUPLING AND COHESIVENESS

> The organization implements security functions as largely independent modules that maximize internal cohesiveness within modules and minimize coupling between modules.

Note: The reduction in inter-module interactions helps to constrain security functions and to manage complexity. The concepts of coupling and cohesion are important with respect to modularity in software design. Coupling refers to the dependencies that one module has on other modules. Cohesion refers to the relationship between the different functions within a particular module. Good software engineering practices rely on modular decomposition, layering, and minimization to reduce and manage complexity, thus producing software modules that are highly cohesive and loosely coupled.

SC-3 (5) : LAYERED STRUCTURES

> The organization implements security functions as a layered structure minimizing interactions between layers of the design and avoiding any dependence by lower layers on the functionality or correctness of higher layers.

Note: The implementation of layered structures with minimized interactions among security functions and non-looping layers (i.e., lower-layer functions do not depend on higher-layer functions) further enables the isolation of security functions and management of complexity.

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-4 : INFORMATION IN SHARED RESOURCES

Priority: P1

Baseline-Impact: *MODERATE*, **HIGH**

> The information system prevents unauthorized and unintended information transfer via shared system resources.

Note: This control prevents information, including encrypted representations of information, produced by the actions of prior users/roles (or the actions of processes acting on behalf of prior users/roles) from being available to any current users/roles (or current processes) that obtain access to shared system resources (e.g., registers, main memory, hard disks) after those resources have been released back to information systems. The control of information in shared resources is also commonly referred to as object reuse and residual information protection. This control does not address: (i) information remanence which refers to residual representation of data that has been nominally erased or removed; (ii) covert channels (including storage and/or timing channels) where shared resources are manipulated to violate information flow restrictions; or (iii) components within information systems for which there are only single users/roles.

Related Controls: *AC-3*, *AC-4*, *MP-6*

SC-4 (1) : SECURITY LEVELS

> [Withdrawn: Incorporated into SC-4].

SC-4 (2) : PERIODS PROCESSING

> The information system prevents unauthorized information transfer via shared resources in accordance with [Assignment: organization-defined procedures] when system processing explicitly switches between different information classification levels or security categories.

Note: This control enhancement applies when there are explicit changes in information processing levels during information system operations, for example, during multilevel processing and periods processing with information at different classification levels or security categories. Organization-defined procedures may include, for example, approved sanitization processes for electronically stored information.

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-5 : DENIAL OF SERVICE PROTECTION

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The information system protects against or limits the effects of the following types of denial of service attacks: [Assignment: organization-defined types of denial of service attacks or references to sources for such information] by employing [Assignment: organization-defined security safeguards].

Note: A variety of technologies exist to limit, or in some cases, eliminate the effects of denial of service attacks. For example, boundary protection devices can filter certain types of packets to protect information system components on internal organizational networks from being directly affected by denial of service attacks. Employing increased capacity and bandwidth combined with service redundancy may also reduce the susceptibility to denial of service attacks.

Related Controls: *SC-6*, *SC-7*

SC-5 (1) : RESTRICT INTERNAL USERS

> The information system restricts the ability of individuals to launch [Assignment: organization-defined denial of service attacks] against other information systems.

Note: Restricting the ability of individuals to launch denial of service attacks requires that the mechanisms used for such attacks are unavailable. Individuals of concern can include, for example, hostile insiders or external adversaries that have successfully breached the information system and are using the system as a platform to launch cyber attacks on third parties. Organizations can restrict the ability of individuals to connect and transmit arbitrary information on the transport medium (i.e., network, wireless spectrum). Organizations can also limit the ability of individuals to use excessive information system resources. Protection against individuals having the ability to launch denial of service attacks may be implemented on specific information systems or on boundary devices prohibiting egress to potential target systems.

SC-5 (2) : EXCESS CAPACITY / BANDWIDTH / REDUNDANCY

> The information system manages excess capacity, bandwidth, or other redundancy to limit the effects of information flooding denial of service attacks.

Note: Managing excess capacity ensures that sufficient capacity is available to counter flooding attacks. Managing excess capacity may include, for example, establishing selected usage priorities, quotas, or partitioning.

SC-5 (3) : DETECTION / MONITORING

> The organization:

Note: Organizations consider utilization and capacity of information system resources when managing risk from denial of service due to malicious attacks. Denial of service attacks can originate from external or internal sources. Information system resources sensitive to denial of service include, for example, physical disk storage, memory, and CPU cycles. Common safeguards to prevent denial of service attacks related to storage utilization and capacity include, for example, instituting disk quotas, configuring information systems to automatically alert administrators when specific storage capacity thresholds are reached, using file compression technologies to maximize available storage space, and imposing separate partitions for system and user data.

Related Controls: *CA-7*, *SI-4*

SC-5 (3)(a)

Employs [Assignment: organization-defined monitoring tools] to detect indicators of denial of service attacks against the information system; and

SC-5 (3)(b)

Monitors [Assignment: organization-defined information system resources] to determine if sufficient resources exist to prevent effective denial of service attacks.

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-6 : RESOURCE AVAILABILITY

Priority: P0

The information system protects the availability of resources by allocating [Assignment: organization-defined resources] by [Selection (one or more); priority; quota; [Assignment: organization-defined security safeguards]].

Note: Priority protection helps prevent lower-priority processes from delaying or interfering with the information system servicing any higher-priority processes. Quotas prevent users or processes from obtaining more than predetermined amounts of resources. This control does not apply to information system components for which there are only single users/roles.

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-7 : BOUNDARY PROTECTION

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The information system:

Note: Managed interfaces include, for example, gateways, routers, firewalls, guards, network-based malicious code analysis and virtualization systems, or encrypted tunnels implemented within a security architecture (e.g., routers protecting firewalls or application gateways residing on protected subnetworks). Subnetworks that are physically or logically separated from internal networks are referred to as demilitarized zones or DMZs. Restricting or prohibiting interfaces within organizational information systems includes, for example, restricting external web traffic to designated web servers within managed interfaces and prohibiting external traffic that appears to be spoofing internal addresses. Organizations consider the shared nature of commercial telecommunications services in the implementation of security controls associated with the use of such services. Commercial telecommunications services are commonly based on network components and consolidated management systems shared by all attached commercial customers, and may also include third party-provided access lines and other service elements. Such transmission services may represent sources of increased risk despite contract security provisions.

Related Controls: *AC-4*, *AC-17*, *CA-3*, *CM-7*, *CP-8*, *IR-4*, *RA-3*, *SC-5*, *SC-13*

SC-7a.

Monitors and controls communications at the external boundary of the system and at key internal boundaries within the system;

SC-7b.

Implements subnetworks for publicly accessible system components that are [Selection: physically; logically] separated from internal organizational networks; and

SC-7c.

Connects to external networks or information systems only through managed interfaces consisting of boundary protection devices arranged in accordance with an organizational security architecture.

SC-7 (1) : PHYSICALLY SEPARATED SUBNETWORKS

[Withdrawn: Incorporated into SC-7].

SC-7 (2) : PUBLIC ACCESS

[Withdrawn: Incorporated into SC-7].

SC-7 (3) : ACCESS POINTS

Baseline-Impact: *MODERATE*, **HIGH**

The organization limits the number of external network connections to the information system.

Note: Limiting the number of external network connections facilitates more comprehensive monitoring of inbound and outbound communications traffic. The Trusted Internet Connection (TIC) initiative is an example of limiting the number of external network connections.

SC-7 (4) : EXTERNAL TELECOMMUNICATIONS SERVICES

Baseline-Impact: *MODERATE*, **HIGH**

The organization:

Related Controls: *SC-8*

SC-7 (4)(a)

Implements a managed interface for each external telecommunication service;

SC-7 (4)(b)

Establishes a traffic flow policy for each managed interface;

SC-7 (4)(c)

Protects the confidentiality and integrity of the information being transmitted across each interface;

SC-7 (4)(d)

Documents each exception to the traffic flow policy with a supporting mission/business need and duration of that need; and

SC-7 (4)(e)

Reviews exceptions to the traffic flow policy [Assignment: organization-defined frequency] and removes exceptions that are no longer supported by an explicit mission/business need.

SC-7 (5) : DENY BY DEFAULT / ALLOW BY EXCEPTION

Baseline-Impact: *MODERATE*, **HIGH**

The information system at managed interfaces denies network communications traffic by default and allows network communications traffic by exception (i.e., deny all, permit by exception).

Note: This control enhancement applies to both inbound and outbound network communications traffic. A deny-all, permit-by-exception network communications traffic policy ensures that only those connections which are essential and approved are allowed.

SC-7 (6) : RESPONSE TO RECOGNIZED FAILURES

[Withdrawn: Incorporated into SC-7 (18)].

SC-7 (7) : PREVENT SPLIT TUNNELING FOR REMOTE DEVICES

Baseline-Impact: *MODERATE*, **HIGH**

> The information system, in conjunction with a remote device, prevents the device from simultaneously establishing non-remote connections with the system and communicating via some other connection to resources in external networks.

Note: This control enhancement is implemented within remote devices (e.g., notebook computers) through configuration settings to disable split tunneling in those devices, and by preventing those configuration settings from being readily configurable by users. This control enhancement is implemented within the information system by the detection of split tunneling (or of configuration settings that allow split tunneling) in the remote device, and by prohibiting the connection if the remote device is using split tunneling. Split tunneling might be desirable by remote users to communicate with local information system resources such as printers/file servers. However, split tunneling would in effect allow unauthorized external connections, making the system more vulnerable to attack and to exfiltration of organizational information. The use of VPNs for remote connections, when adequately provisioned with appropriate security controls, may provide the organization with sufficient assurance that it can effectively treat such connections as non-remote connections from the confidentiality and integrity perspective. VPNs thus provide a means for allowing non-remote communications paths from remote devices. The use of an adequately provisioned VPN does not eliminate the need for preventing split tunneling.

SC-7 (8) : ROUTE TRAFFIC TO AUTHENTICATED PROXY SERVERS

Baseline-Impact: **HIGH**

> The information system routes [Assignment: organization-defined internal communications traffic] to [Assignment: organization-defined external networks] through authenticated proxy servers at managed interfaces.

Note: External networks are networks outside of organizational control. A proxy server is a server (i.e., information system or application) that acts as an intermediary for clients requesting information system resources (e.g., files, connections, web pages, or services) from other organizational servers. Client requests established through an initial connection to the proxy server are evaluated to manage complexity and to provide additional protection by limiting direct connectivity. Web content filtering devices are one of the most common proxy servers providing access to the Internet. Proxy servers support logging individual Transmission Control Protocol (TCP) sessions and blocking specific Uniform Resource Locators (URLs), domain names, and Internet Protocol (IP) addresses. Web proxies can be configured with organization-defined lists of authorized and unauthorized websites.

Related Controls: *AC-3*, *AU-2*

SC-7 (9) : RESTRICT THREATENING OUTGOING COMMUNICATIONS TRAFFIC

> The information system:

Note: Detecting outgoing communications traffic from internal actions that may pose threats to external information systems is sometimes termed extrusion detection. Extrusion detection at information system boundaries as part of managed interfaces includes the analysis of incoming and outgoing communications traffic searching for indications of internal threats to the security of external systems. Such threats include, for example, traffic indicative of denial of service attacks and traffic containing malicious code.

Related Controls: *AU-2*, *AU-6*, *SC-38*, *SC-44*, *SI-3*, *SI-4*

SC-7 (9)(a)

Detects and denies outgoing communications traffic posing a threat to external information systems; and

SC-7 (9)(b)

Audits the identity of internal users associated with denied communications.

SC-7 (10) : PREVENT UNAUTHORIZED EXFILTRATION

The organization prevents the unauthorized exfiltration of information across managed interfaces.

Note: Safeguards implemented by organizations to prevent unauthorized exfiltration of information from information systems include, for example: (i) strict adherence to protocol formats; (ii) monitoring for beaconing from information systems; (iii) monitoring for steganography; (iv) disconnecting external network interfaces except when explicitly needed; (v) disassembling and reassembling packet headers; and (vi) employing traffic profile analysis to detect deviations from the volume/types of traffic expected within organizations or call backs to command and control centers. Devices enforcing strict adherence to protocol formats include, for example, deep packet inspection firewalls and XML gateways. These devices verify adherence to protocol formats and specification at the application layer and serve to identify vulnerabilities that cannot be detected by devices operating at the network or transport layers. This control enhancement is closely associated with cross-domain solutions and system guards enforcing information flow requirements.

Related Controls: *SI-3*

SC-7 (11) : RESTRICT INCOMING COMMUNICATIONS TRAFFIC

The information system only allows incoming communications from [Assignment: organization-defined authorized sources] to be routed to [Assignment: organization-defined authorized destinations].

Note: This control enhancement provides determinations that source and destination address pairs represent authorized/allowed communications. Such determinations can be based on several factors including, for example, the presence of source/destination address pairs in lists of authorized/allowed communications, the absence of address pairs in lists of unauthorized/disallowed pairs, or meeting more general rules for authorized/allowed source/destination pairs.

Related Controls: *AC-3*

SC-7 (12) : HOST-BASED PROTECTION

> The organization implements [Assignment: organization-defined host-based boundary protection mechanisms] at [Assignment: organization-defined information system components].

Note: Host-based boundary protection mechanisms include, for example, host-based firewalls. Information system components employing host-based boundary protection mechanisms include, for example, servers, workstations, and mobile devices.

SC-7 (13) : ISOLATION OF SECURITY TOOLS / MECHANISMS / SUPPORT COMPONENTS

> The organization isolates [Assignment: organization-defined information security tools, mechanisms, and support components] from other internal information system components by implementing physically separate subnetworks with managed interfaces to other components of the system.

Note: Physically separate subnetworks with managed interfaces are useful, for example, in isolating computer network defenses from critical operational processing networks to prevent adversaries from discovering the analysis and forensics techniques of organizations.

Related Controls: *SA-8*, *SC-2*, *SC-3*

SC-7 (14) : PROTECTS AGAINST UNAUTHORIZED PHYSICAL CONNECTIONS

> The organization protects against unauthorized physical connections at [Assignment: organization-defined managed interfaces].

Note: Information systems operating at different security categories or classification levels may share common physical and environmental controls, since the systems may share space within organizational facilities. In practice, it is possible that these separate information systems may share common equipment rooms, wiring closets, and cable distribution paths. Protection against unauthorized physical connections can be achieved, for example, by employing clearly identified and physically separated cable trays, connection frames, and patch panels for each side of managed interfaces with physical access controls enforcing limited authorized access to these items.

Related Controls: *PE-4*, *PE-19*

SC-7 (15) : ROUTE PRIVILEGED NETWORK ACCESSES

> The information system routes all networked, privileged accesses through a dedicated, managed interface for purposes of access control and auditing.

Related Controls: *AC-2*, *AC-3*, *AU-2*, *SI-4*

SC-7 (16) : PREVENT DISCOVERY OF COMPONENTS / DEVICES

> The information system prevents discovery of specific system components composing a managed interface.

Note: This control enhancement protects network addresses of information system components that are part of managed interfaces from discovery through common tools and techniques used to identify devices on networks. Network addresses are not available for discovery (e.g., network address not published or entered in domain name systems), requiring prior knowledge for access. Another obfuscation technique is to periodically change network addresses.

SC-7 (17) : AUTOMATED ENFORCEMENT OF PROTOCOL FORMATS

> The information system enforces adherence to protocol formats.

Note: Information system components that enforce protocol formats include, for example, deep packet inspection firewalls and XML gateways. Such system components verify adherence to protocol formats/specifications (e.g., IEEE) at the application layer and identify significant vulnerabilities that cannot be detected by devices operating at the network or transport layers.

Related Controls: *SC-4*

SC-7 (18) : FAIL SECURE

Baseline-Impact: HIGH

> The information system fails securely in the event of an operational failure of a boundary protection device.

Note: Fail secure is a condition achieved by employing information system mechanisms to ensure that in the event of operational failures of boundary protection devices at managed interfaces (e.g., routers, firewalls, guards, and application gateways residing on protected subnetworks commonly referred to as demilitarized zones), information systems do not enter into unsecure states where intended security properties no longer hold. Failures of boundary protection devices cannot lead to, or cause information external to the devices to enter the devices, nor can failures permit unauthorized information releases.

Related Controls: *CP-2*, *SC-24*

SC-7 (19) : BLOCKS COMMUNICATION FROM NON-ORGANIZATIONALLY CONFIGURED HOSTS

> The information system blocks both inbound and outbound communications traffic between [Assignment: organization-defined communication clients] that are independently configured by end users and external service providers.

Note: Communication clients independently configured by end users and external service providers include, for example, instant messaging clients. Traffic blocking does not apply to communication clients that are configured by organizations to perform authorized functions.

SC-7 (20) : DYNAMIC ISOLATION / SEGREGATION

> The information system provides the capability to dynamically isolate/segregate [Assignment: organization-defined information system components] from other components of the system.

Note: The capability to dynamically isolate or segregate certain internal components of organizational information systems is useful when it is necessary to partition or separate certain components of dubious origin from those components possessing greater trustworthiness. Component isolation reduces the attack surface of organizational information systems. Isolation of selected information system components is also a means of limiting the damage from successful cyber attacks when those attacks occur.

SC-7 (21) : ISOLATION OF INFORMATION SYSTEM COMPONENTS

Baseline-Impact: HIGH

> The organization employs boundary protection mechanisms to separate [Assignment: organization-defined information system components] supporting [Assignment: organization-defined missions and/or business functions].

Note: Organizations can isolate information system components performing different missions and/or business functions. Such isolation limits unauthorized information flows among system components and also provides the opportunity to deploy greater levels of protection for selected components. Separating system components with boundary protection mechanisms provides the capability for increased protection of individual components and to more effectively control information flows between those components. This type of enhanced protection limits the potential harm from cyber attacks and errors. The degree of separation provided varies depending upon the mechanisms chosen. Boundary protection mechanisms include, for example, routers, gateways, and firewalls separating system components into physically separate networks or subnetworks, cross-domain devices separating subnetworks, virtualization techniques, and encrypting information flows among system components using distinct encryption keys.

Related Controls: *CA-9*, *SC-3*

SC-7 (22) : SEPARATE SUBNETS FOR CONNECTING TO DIFFERENT SECURITY DOMAINS

> The information system implements separate network addresses (i.e., different subnets) to connect to systems in different security domains.

Note: Decomposition of information systems into subnets helps to provide the appropriate level of protection for network connections to different security domains containing information with different security categories or classification levels.

SC-7 (23) : DISABLE SENDER FEEDBACK ON PROTOCOL VALIDATION FAILURE

> The information system disables feedback to senders on protocol format validation failure.

Note: Disabling feedback to senders when there is a failure in protocol validation format prevents adversaries from obtaining information which would otherwise be unavailable.

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-8 : TRANSMISSION CONFIDENTIALITY AND INTEGRITY

Priority: P1

Baseline-Impact: *MODERATE*, **HIGH**

> The information system protects the [Selection (one or more): confidentiality; integrity] of transmitted information.

Note: This control applies to both internal and external networks and all types of information system components from which information can be transmitted (e.g., servers, mobile devices, notebook computers, printers, copiers, scanners, facsimile machines). Communication paths outside the physical protection of a controlled boundary are exposed to the possibility of interception and modification. Protecting the confidentiality and/or integrity of organizational information can be accomplished by physical means (e.g., by employing protected distribution systems) or by logical means (e.g., employing encryption techniques). Organizations relying on commercial providers offering transmission services as commodity services rather than as fully dedicated services (i.e., services which can be highly specialized to individual customer needs), may find it difficult to obtain the necessary assurances regarding the implementation of needed security controls for transmission confidentiality/integrity. In such situations, organizations determine what types of confidentiality/integrity services are available in standard, commercial telecommunication service packages. If it is infeasible or impractical to obtain the necessary security controls and assurances of control effectiveness through appropriate contracting vehicles, organizations implement appropriate compensating security controls or explicitly accept the additional risk.

Related Controls: *AC-17*, *PE-4*

SC-8 (1) : CRYPTOGRAPHIC OR ALTERNATE PHYSICAL PROTECTION

Baseline-Impact: *MODERATE*, **HIGH**

> The information system implements cryptographic mechanisms to [Selection (one or more): prevent unauthorized disclosure of information; detect changes to information] during transmission unless otherwise protected by [Assignment: organization-defined alternative physical safeguards].

Note: Encrypting information for transmission protects information from unauthorized disclosure and modification. Cryptographic mechanisms implemented to protect information integrity include, for example, cryptographic hash functions which have common application in digital signatures, checksums, and message authentication codes. Alternative physical security safeguards include, for example, protected distribution systems.

Related Controls: *SC-13*

SC-8 (2) : PRE / POST TRANSMISSION HANDLING

> The information system maintains the [Selection (one or more): confidentiality; integrity] of information during preparation for transmission and during reception.

Note: Information can be either unintentionally or maliciously disclosed or modified during preparation for transmission or during reception including, for example, during aggregation, at protocol transformation points, and during packing/unpacking. These unauthorized disclosures or modifications compromise the confidentiality or integrity of the information.

Related Controls: *AU-10*

SC-8 (3) : CRYPTOGRAPHIC PROTECTION FOR MESSAGE EXTERNALS

> The information system implements cryptographic mechanisms to protect message externals unless otherwise protected by [Assignment: organization-defined alternative physical safeguards].

Note: This control enhancement addresses protection against unauthorized disclosure of information. Message externals include, for example, message headers/routing information. This control enhancement prevents the exploitation of message externals and applies to both internal and external networks or links that may be visible to individuals who are not authorized users. Header/routing information is sometimes transmitted unencrypted because the information is not properly identified by organizations as having significant value or because encrypting the information can result in lower network performance and/or higher costs. Alternative physical safeguards include, for example, protected distribution systems.

Related Controls: *SC-12*, *SC-13*

SC-8 (4) : CONCEAL / RANDOMIZE COMMUNICATIONS

> The information system implements cryptographic mechanisms to conceal or randomize communication patterns unless otherwise protected by [Assignment: organization-defined alternative physical safeguards].

Note: This control enhancement addresses protection against unauthorized disclosure of information. Communication patterns include, for example, frequency, periods, amount, and predictability. Changes to communications patterns can reveal information having intelligence value especially when combined with other available information related to missions/business functions supported by organizational information systems. This control enhancement prevents the derivation of intelligence based on communications patterns and applies to both internal and external networks or links that may be visible to individuals who are not authorized users. Encrypting the links and transmitting in continuous, fixed/random patterns prevents the derivation of intelligence from the system communications patterns. Alternative physical safeguards include, for example, protected distribution systems.

Related Controls: *SC-12*, *SC-13*

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-9 : TRANSMISSION CONFIDENTIALITY

> [Withdrawn: Incorporated into SC-8].

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-10 : NETWORK DISCONNECT

Priority: P2

Baseline-Impact: *MODERATE*, **HIGH**

> The information system terminates the network connection associated with a communications session at the end of the session or after [Assignment: organization-defined time period] of inactivity.

Note: This control applies to both internal and external networks. Terminating network connections associated with communications sessions include, for example, de-allocating associated TCP/IP address/port pairs at the operating system level, or de-allocating networking assignments at the application level if multiple application sessions are using a single, operating system-level network connection. Time periods of inactivity may be established by organizations and include, for example, time periods by type of network access or for specific network accesses.

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-11 : TRUSTED PATH

Priority: P0

> The information system establishes a trusted communications path between the user and the following security functions of the system: [Assignment: organization-defined security functions to include at a minimum, information system authentication and re-authentication].

Note: Trusted paths are mechanisms by which users (through input devices) can communicate directly with security functions of information systems with the requisite assurance to support information security policies. The mechanisms can be activated only by users or the security functions of organizational information systems. User responses via trusted paths are protected from modifications by or disclosure to untrusted applications. Organizations employ trusted paths for high-assurance connections between security functions of information systems and users (e.g., during system logons). Enforcement of trusted communications paths is typically provided via an implementation that meets the reference monitor concept.

Related Controls: *AC-16*, *AC-25*

SC-11 (1) : LOGICAL ISOLATION

The information system provides a trusted communications path that is logically isolated and distinguishable from other paths.

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-12 : CRYPTOGRAPHIC KEY ESTABLISHMENT AND MANAGEMENT

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The organization establishes and manages cryptographic keys for required cryptography employed within the information system in accordance with [Assignment: organization-defined requirements for key generation, distribution, storage, access, and destruction].

Note: Cryptographic key management and establishment can be performed using manual procedures or automated mechanisms with supporting manual procedures. Organizations define key management requirements in accordance with applicable federal laws, Executive Orders, directives, regulations, policies, standards, and guidance, specifying appropriate options, levels, and parameters. Organizations manage trust stores to ensure that only approved trust anchors are in such trust stores. This includes certificates with visibility external to organizational information systems and certificates related to the internal operations of systems.

Related Controls: *SC-13*, *SC-17*

SC-12 (1) : AVAILABILITY

Baseline-Impact: **HIGH**

> The organization maintains availability of information in the event of the loss of cryptographic keys by users.

Note: Escrowing of encryption keys is a common practice for ensuring availability in the event of loss of keys (e.g., due to forgotten passphrase).

SC-12 (2) : SYMMETRIC KEYS

> The organization produces, controls, and distributes symmetric cryptographic keys using [Selection: NIST FIPS-compliant; NSA-approved] key management technology and processes.

SC-12 (3) : ASYMMETRIC KEYS

The organization produces, controls, and distributes asymmetric cryptographic keys using [Selection: NSA-approved key management technology and processes; approved PKI Class 3 certificates or prepositioned keying material; approved PKI Class 3 or Class 4 certificates and hardware security tokens that protect the users private key].

SC-12 (4) : PKI CERTIFICATES

[Withdrawn: Incorporated into SC-12].

SC-12 (5) : PKI CERTIFICATES / HARDWARE TOKENS

[Withdrawn: Incorporated into SC-12].

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-13 : CRYPTOGRAPHIC PROTECTION

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The information system implements [Assignment: organization-defined cryptographic uses and type of cryptography required for each use] in accordance with applicable federal laws, Executive Orders, directives, policies, regulations, and standards.

Note: Cryptography can be employed to support a variety of security solutions including, for example, the protection of classified and Controlled Unclassified Information, the provision of digital signatures, and the enforcement of information separation when authorized individuals have the necessary clearances for such information but lack the necessary formal access approvals. Cryptography can also be used to support random number generation and hash generation. Generally applicable cryptographic standards include FIPS-validated cryptography and NSA-approved cryptography. This control does not impose any requirements on organizations to use cryptography. However, if cryptography is required based on the selection of other security controls, organizations define each type of cryptographic use and the type of cryptography required (e.g., protection of classified information: NSA-approved cryptography; provision of digital signatures: FIPS-validated cryptography).

Related Controls: *AC-2*, *AC-3*, *AC-7*, *AC-17*, *AC-18*, *AU-9*, *AU-10*, *CM-11*, *CP-9*, *IA-3*, *IA-7*, *MA-4*, *MP-2*, *MP-4*, *MP-5*, *SA-4*, *SC-8*, *SC-12*, *SC-28*, *SI-7*

SC-13 (1) : FIPS-VALIDATED CRYPTOGRAPHY

[Withdrawn: Incorporated into SC-13].

SC-13 (2) : NSA-APPROVED CRYPTOGRAPHY

[Withdrawn: Incorporated into SC-13].

SC-13 (3) : INDIVIDUALS WITHOUT FORMAL ACCESS APPROVALS

[Withdrawn: Incorporated into SC-13].

SC-13 (4) : DIGITAL SIGNATURES

[Withdrawn: Incorporated into SC-13].

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-14 : PUBLIC ACCESS PROTECTIONS

[Withdrawn: Capability provided by AC-2, AC-3, AC-5, AC-6, SI-3, SI-4, SI-5, SI-7, SI-10].

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-15 : COLLABORATIVE COMPUTING DEVICES

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The information system:

Note: Collaborative computing devices include, for example, networked white boards, cameras, and microphones. Explicit indication of use includes, for example, signals to users when collaborative computing devices are activated.

Related Controls: *AC-21*

SC-15a.

Prohibits remote activation of collaborative computing devices with the following exceptions: [Assignment: organization-defined exceptions where remote activation is to be allowed]; and

SC-15b.

Provides an explicit indication of use to users physically present at the devices.

SC-15 (1) : PHYSICAL DISCONNECT

The information system provides physical disconnect of collaborative computing devices in a manner that supports ease of use.

Note: Failing to physically disconnect from collaborative computing devices can result in subsequent compromises of organizational information. Providing easy methods to physically disconnect from such devices after a collaborative computing session helps to ensure that participants actually carry out the disconnect activity without having to go through complex and tedious procedures.

SC-15 (2) : BLOCKING INBOUND / OUTBOUND COMMUNICATIONS TRAFFIC

[Withdrawn: Incorporated into SC-7].

SC-15 (3) : DISABLING / REMOVAL IN SECURE WORK AREAS

The organization disables or removes collaborative computing devices from [Assignment: organization-defined information systems or information system components] in [Assignment: organization-defined secure work areas].

Note: Failing to disable or remove collaborative computing devices from information systems or information system components can result in subsequent compromises of organizational information including, for example, eavesdropping on conversations.

SC-15 (4) : EXPLICITLY INDICATE CURRENT PARTICIPANTS

The information system provides an explicit indication of current participants in [Assignment: organization-defined online meetings and teleconferences].

Note: This control enhancement helps to prevent unauthorized individuals from participating in collaborative computing sessions without the explicit knowledge of other participants.

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-16 : TRANSMISSION OF SECURITY ATTRIBUTES

Priority: P0

> The information system associates [Assignment: organization-defined security attributes] with information exchanged between information systems and between system components.

Note: Security attributes can be explicitly or implicitly associated with the information contained in organizational information systems or system components.

Related Controls: *AC-3*, *AC-4*, *AC-16*

SC-16 (1) : INTEGRITY VALIDATION

> The information system validates the integrity of transmitted security attributes.

Note: This control enhancement ensures that the verification of the integrity of transmitted information includes security attributes.

Related Controls: *AU-10*, *SC-8*

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-17 : PUBLIC KEY INFRASTRUCTURE CERTIFICATES

Priority: P1

Baseline-Impact: *MODERATE*, **HIGH**

> The organization issues public key certificates under an [Assignment: organization-defined certificate policy] or obtains public key certificates from an approved service provider.

Note: For all certificates, organizations manage information system trust stores to ensure only approved trust anchors are in the trust stores. This control addresses both certificates with visibility external to organizational information systems and certificates related to the internal operations of systems, for example, application-specific time services.

Related Controls: *SC-12*

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-18 : MOBILE CODE

Priority: P2

Baseline-Impact: *MODERATE*, **HIGH**

The organization:

Note: Decisions regarding the employment of mobile code within organizational information systems are based on the potential for the code to cause damage to the systems if used maliciously. Mobile code technologies include, for example, Java, JavaScript, ActiveX, Postscript, PDF, Shockwave movies, Flash animations, and VBScript. Usage restrictions and implementation guidance apply to both the selection and use of mobile code installed on servers and mobile code downloaded and executed on individual workstations and devices (e.g., smart phones). Mobile code policy and procedures address preventing the development, acquisition, or introduction of unacceptable mobile code within organizational information systems.

Related Controls: *AU-2*, *AU-12*, *CM-2*, *CM-6*, *SI-3*

SC-18a.

Defines acceptable and unacceptable mobile code and mobile code technologies;

SC-18b.

Establishes usage restrictions and implementation guidance for acceptable mobile code and mobile code technologies; and

SC-18c.

Authorizes, monitors, and controls the use of mobile code within the information system.

SC-18 (1) : IDENTIFY UNACCEPTABLE CODE / TAKE CORRECTIVE ACTIONS

The information system identifies [Assignment: organization-defined unacceptable mobile code] and takes [Assignment: organization-defined corrective actions].

Note: Corrective actions when unacceptable mobile code is detected include, for example, blocking, quarantine, or alerting administrators. Blocking includes, for example, preventing transmission of word processing files with embedded macros when such macros have been defined to be unacceptable mobile code.

SC-18 (2) : ACQUISITION / DEVELOPMENT / USE

The organization ensures that the acquisition, development, and use of mobile code to be deployed in the information system meets [Assignment: organization-defined mobile code requirements].

SC-18 (3) : PREVENT DOWNLOADING / EXECUTION

The information system prevents the download and execution of [Assignment: organization-defined unacceptable mobile code].

SC-18 (4) : PREVENT AUTOMATIC EXECUTION

The information system prevents the automatic execution of mobile code in [Assignment: organization-defined software applications] and enforces [Assignment: organization-defined actions] prior to executing the code.

Note: Actions enforced before executing mobile code, include, for example, prompting users prior to opening electronic mail attachments. Preventing automatic execution of mobile code includes, for example, disabling auto execute features on information system components employing portable storage devices such as Compact Disks (CDs), Digital Video Disks (DVDs), and Universal Serial Bus (USB) devices.

SC-18 (5) : ALLOW EXECUTION ONLY IN CONFINED ENVIRONMENTS

The organization allows execution of permitted mobile code only in confined virtual machine environments.

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-19 : VOICE OVER INTERNET PROTOCOL

Priority: P1

Baseline-Impact: *MODERATE*, **HIGH**

The organization:

Related Controls: *CM-6*, *SC-7*, *SC-15*

SC-19a.

Establishes usage restrictions and implementation guidance for Voice over Internet Protocol (VoIP) technologies based on the potential to cause damage to the information system if used maliciously; and

SC-19b.

Authorizes, monitors, and controls the use of VoIP within the information system.

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-20 : SECURE NAME / ADDRESS RESOLUTION SERVICE (AUTHORITATIVE SOURCE)

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The information system:

Note: This control enables external clients including, for example, remote Internet clients, to obtain origin authentication and integrity verification assurances for the host/service name to network address resolution information obtained through the service. Information systems that provide name and address resolution services include, for example, domain name system (DNS) servers. Additional artifacts include, for example, DNS Security (DNSSEC) digital signatures and cryptographic keys. DNS resource records are examples of authoritative data. The means to indicate the security status of child zones includes, for example, the use of delegation signer resource records in the DNS. The DNS security controls reflect (and are referenced from) OMB Memorandum 08-23. Information systems that use technologies other than the DNS to map between host/service names and network addresses provide other means to assure the authenticity and integrity of response data.

Related Controls: *AU-10*, *SC-8*, *SC-12*, *SC-13*, *SC-21*, *SC-22*

SC-20a.

Provides additional data origin authentication and integrity verification artifacts along with the authoritative name resolution data the system returns in response to external name/address resolution queries; and

SC-20b.

Provides the means to indicate the security status of child zones and (if the child supports secure resolution services) to enable verification of a chain of trust among parent and child domains, when operating as part of a distributed, hierarchical namespace.

SC-20 (1) : CHILD SUBSPACES

[Withdrawn: Incorporated into SC-20].

SC-20 (2) : DATA ORIGIN / INTEGRITY

The information system provides data origin and integrity protection artifacts for internal name/address resolution queries.

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-21 : SECURE NAME / ADDRESS RESOLUTION SERVICE (RECURSIVE OR CACHING RESOLVER)

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The information system requests and performs data origin authentication and data integrity verification on the name/address resolution responses the system receives from authoritative sources.

Note: Each client of name resolution services either performs this validation on its own, or has authenticated channels to trusted validation providers. Information systems that provide name and address resolution services for local clients include, for example, recursive resolving or caching domain name system (DNS) servers. DNS client resolvers either perform validation of DNSSEC signatures, or clients use authenticated channels to recursive resolvers that perform such validations. Information systems that use technologies other than the DNS to map between host/service names and network addresses provide other means to enable clients to verify the authenticity and integrity of response data.

Related Controls: *SC-20*, *SC-22*

SC-21 (1) : DATA ORIGIN / INTEGRITY

[Withdrawn: Incorporated into SC-21].

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-22 : ARCHITECTURE AND PROVISIONING FOR NAME / ADDRESS RESOLUTION SERVICE

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The information systems that collectively provide name/address resolution service for an organization are fault-tolerant and implement internal/external role separation.

Note: Information systems that provide name and address resolution services include, for example, domain name system (DNS) servers. To eliminate single points of failure and to enhance redundancy, organizations employ at least two authoritative domain name system servers, one configured as the primary server and the other configured as the secondary server. Additionally, organizations typically deploy the servers in two geographically separated network subnetworks (i.e., not located in the same physical facility). For role separation, DNS servers with internal roles only process name and address resolution requests from within organizations (i.e., from internal clients). DNS servers with external roles only process name and address resolution information requests from clients external to organizations (i.e., on external networks including the Internet). Organizations specify clients that can access authoritative DNS servers in particular roles (e.g., by address ranges, explicit lists).

Related Controls: *SC-2*, *SC-20*, *SC-21*, *SC-24*

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-23 : SESSION AUTHENTICITY

Priority: P1

Baseline-Impact: *MODERATE*, **HIGH**

The information system protects the authenticity of communications sessions.

Note: This control addresses communications protection at the session, versus packet level (e.g., sessions in service-oriented architectures providing web-based services) and establishes grounds for confidence at both ends of communications sessions in ongoing identities of other parties and in the validity of information transmitted. Authenticity protection includes, for example, protecting against man-in-the-middle attacks/session hijacking and the insertion of false information into sessions.

Related Controls: *SC-8*, *SC-10*, *SC-11*

SC-23 (1) : INVALIDATE SESSION IDENTIFIERS AT LOGOUT

The information system invalidates session identifiers upon user logout or other session termination.

Note: This control enhancement curtails the ability of adversaries from capturing and continuing to employ previously valid session IDs.

SC-23 (2) : USER-INITIATED LOGOUTS / MESSAGE DISPLAYS

[Withdrawn: Incorporated into AC-12 (1)].

SC-23 (3) : UNIQUE SESSION IDENTIFIERS WITH RANDOMIZATION

The information system generates a unique session identifier for each session with [Assignment: organization-defined randomness requirements] and recognizes only session identifiers that are system-generated.

Note: This control enhancement curtails the ability of adversaries from reusing previously valid session IDs. Employing the concept of randomness in the generation of unique session identifiers helps to protect against brute-force attacks to determine future session identifiers.

Related Controls: *SC-13*

SC-23 (4) : UNIQUE SESSION IDENTIFIERS WITH RANDOMIZATION

[Withdrawn: Incorporated into SC-23 (3)].

SC-23 (5) : ALLOWED CERTIFICATE AUTHORITIES

The information system only allows the use of [Assignment: organization-defined certificate authorities] for verification of the establishment of protected sessions.

Note: Reliance on certificate authorities (CAs) for the establishment of secure sessions includes, for example, the use of Secure Socket Layer (SSL) and/or Transport Layer Security (TLS) certificates. These certificates, after verification by the respective certificate authorities, facilitate the establishment of protected sessions between web clients and web servers.

Related Controls: *SC-13*

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-24 : FAIL IN KNOWN STATE

Priority: P1

Baseline-Impact: HIGH

The information system fails to a [Assignment: organization-defined known-state] for [Assignment: organization-defined types of failures] preserving [Assignment: organization-defined system state information] in failure.

Note: Failure in a known state addresses security concerns in accordance with the mission/business needs of organizations. Failure in a known secure state helps to prevent the loss of confidentiality, integrity, or availability of information in the event of failures of organizational information systems or system components. Failure in a known safe state helps to prevent systems from failing to a state that may cause injury to individuals or destruction to property. Preserving information system state information facilitates system restart and return to the operational mode of organizations with less disruption of mission/business processes.

Related Controls: *CP-2*, *CP-10*, *CP-12*, *SC-7*, *SC-22*

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-25 : THIN NODES

Priority: P0

The organization employs [Assignment: organization-defined information system components] with minimal functionality and information storage.

Note: The deployment of information system components with reduced/minimal functionality (e.g., diskless nodes and thin client technologies) reduces the need to secure every user endpoint, and may reduce the exposure of information,

information systems, and services to cyber attacks.

Related Controls: *SC-30*

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-26 : HONEYPOTS

Priority: P0

> The information system includes components specifically designed to be the target of malicious attacks for the purpose of detecting, deflecting, and analyzing such attacks.

Note: A honeypot is set up as a decoy to attract adversaries and to deflect their attacks away from the operational systems supporting organizational missions/business function. Depending upon the specific usage of the honeypot, consultation with the Office of the General Counsel before deployment may be needed.

Related Controls: *SC-30*, *SC-44*, *SI-3*, *SI-4*

SC-26 (1) : DETECTION OF MALICIOUS CODE

> [Withdrawn: Incorporated into SC-35].

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-27 : PLATFORM-INDEPENDENT APPLICATIONS

Priority: P0

> The information system includes: [Assignment: organization-defined platform-independent applications].

Note: Platforms are combinations of hardware and software used to run software applications. Platforms include: (i) operating systems; (ii) the underlying computer architectures, or (iii) both. Platform-independent applications are applications that run on multiple platforms. Such applications promote portability and reconstitution on different platforms, increasing the availability of critical functions within organizations while information systems with specific operating systems are under attack.

Related Controls: *SC-29*

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-28 : PROTECTION OF INFORMATION AT REST

Priority: P1

Baseline-Impact: *MODERATE*, **HIGH**

> The information system protects the [Selection (one or more): confidentiality; integrity] of [Assignment: organization-defined information at rest].

Note: This control addresses the confidentiality and integrity of information at rest and covers user information and system information. Information at rest refers to the state of information when it is located on storage devices as specific components of information systems. System-related information requiring protection includes, for example, configurations or rule sets for firewalls, gateways, intrusion detection/prevention systems, filtering routers, and authenticator content. Organizations may employ different mechanisms to achieve confidentiality and integrity protections, including the use of cryptographic mechanisms and file share scanning. Integrity protection can be achieved, for example, by implementing Write-Once-Read-Many (WORM) technologies. Organizations may also employ other security controls including, for example, secure off-line storage in lieu of online storage when adequate protection of information at rest cannot otherwise be achieved and/or continuous monitoring to identify malicious code at rest.

Related Controls: *AC-3*, *AC-6*, *CA-7*, *CM-3*, *CM-5*, *CM-6*, *PE-3*, *SC-8*, *SC-13*, *SI-3*, *SI-7*

SC-28 (1) : CRYPTOGRAPHIC PROTECTION

> The information system implements cryptographic mechanisms to prevent unauthorized disclosure and modification of [Assignment: organization-defined information] on [Assignment: organization-defined information system components].

Note: Selection of cryptographic mechanisms is based on the need to protect the confidentiality and integrity of organizational information. The strength of mechanism is commensurate with the security category and/or classification of the information. This control enhancement applies to significant concentrations of digital media in organizational areas designated for media storage and also to limited quantities of media generally associated with information system components in operational environments (e.g., portable storage devices, mobile devices). Organizations have the flexibility to either encrypt all information on storage devices (i.e., full disk encryption) or encrypt specific data structures (e.g., files, records, or fields). Organizations employing cryptographic mechanisms to protect information at rest also consider cryptographic key management solutions.

Related Controls: *AC-19*, *SC-12*

SC-28 (2) : OFF-LINE STORAGE

> The organization removes from online storage and stores off-line in a secure location [Assignment: organization-defined information].

Note: Removing organizational information from online information system storage to off-line storage eliminates the possibility of individuals gaining unauthorized access to the information through a network. Therefore, organizations may choose to move information to off-line storage in lieu of protecting such information in online storage.

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-29 : HETEROGENEITY

Priority: P0

> The organization employs a diverse set of information technologies for [Assignment: organization-defined information system components] in the implementation of the information system.

Note: Increasing the diversity of information technologies within organizational information systems reduces the impact of potential exploitations of specific technologies and also defends against common mode failures, including those failures induced by supply chain attacks. Diversity in information technologies also reduces the likelihood that the means adversaries use to compromise one information system component will be equally effective against other system components, thus further increasing the adversary work factor to successfully complete planned cyber attacks. An increase in diversity may add complexity and management overhead which could ultimately lead to mistakes and unauthorized configurations.

Related Controls: *SA-12*, *SA-14*, *SC-27*

SC-29 (1) : VIRTUALIZATION TECHNIQUES

> The organization employs virtualization techniques to support the deployment of a diversity of operating systems and applications that are changed [Assignment: organization-defined frequency].

Note: While frequent changes to operating systems and applications pose configuration management challenges, the changes can result in an increased work factor for adversaries in order to carry out successful cyber attacks. Changing virtual operating systems or applications, as opposed to changing actual operating systems/applications, provide virtual changes that impede attacker success while reducing configuration management efforts. In addition, virtualization techniques can assist organizations in isolating untrustworthy software and/or software of dubious provenance into confined execution environments.

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-30 : CONCEALMENT AND MISDIRECTION

Priority: P0

> The organization employs [Assignment: organization-defined concealment and misdirection techniques] for [Assignment: organization-defined information systems] at [Assignment: organization-defined time periods] to confuse and mislead adversaries.

Note: Concealment and misdirection techniques can significantly reduce the targeting capability of adversaries (i.e., window of opportunity and available attack surface) to initiate and complete cyber attacks. For example, virtualization techniques provide organizations with the ability to disguise information systems, potentially reducing the likelihood of successful attacks without the cost of having multiple platforms. Increased use of concealment/misdirection techniques including, for example, randomness, uncertainty, and virtualization, may sufficiently confuse and mislead adversaries and subsequently increase the risk of discovery and/or exposing tradecraft. Concealment/misdirection techniques may also provide organizations additional time to successfully perform core missions and business functions. Because of the time and effort required to support concealment/misdirection techniques, it is anticipated that such techniques would be used by organizations on a very limited basis.

Related Controls: *SC-26*, *SC-29*, *SI-14*

SC-30 (1) : VIRTUALIZATION TECHNIQUES

[Withdrawn: Incorporated into SC-29 (1)].

SC-30 (2) : RANDOMNESS

The organization employs [Assignment: organization-defined techniques] to introduce randomness into organizational operations and assets.

Note: Randomness introduces increased levels of uncertainty for adversaries regarding the actions organizations take in defending against cyber attacks. Such actions may impede the ability of adversaries to correctly target information resources of organizations supporting critical missions/business functions. Uncertainty may also cause adversaries to hesitate before initiating or continuing attacks. Misdirection techniques involving randomness include, for example, performing certain routine actions at different times of day, employing different information technologies (e.g., browsers, search engines), using different suppliers, and rotating roles and responsibilities of organizational personnel.

SC-30 (3) : CHANGE PROCESSING / STORAGE LOCATIONS

The organization changes the location of [Assignment: organization-defined processing and/or storage] [Selection: [Assignment: organization-defined time frequency]; at random time intervals]].

Note: Adversaries target critical organizational missions/business functions and the information resources supporting those missions and functions while at the same time, trying to minimize exposure of their existence and tradecraft. The static, homogeneous, and deterministic nature of organizational information systems targeted by adversaries, make such systems more susceptible to cyber attacks with less adversary cost and effort to be successful. Changing organizational processing and storage locations (sometimes referred to as moving target defense) addresses the advanced persistent threat (APT) using techniques such as virtualization, distributed processing, and replication. This enables organizations to relocate the information resources (i.e., processing and/or storage) supporting critical missions and business functions. Changing locations of processing activities and/or storage sites introduces uncertainty into the targeting activities by adversaries. This uncertainty increases the work factor of adversaries making compromises or breaches to organizational information systems much more difficult and time-consuming, and increases the chances that adversaries may inadvertently disclose aspects of tradecraft while attempting to locate critical organizational resources.

SC-30 (4) : MISLEADING INFORMATION

> The organization employs realistic, but misleading information in [Assignment: organization-defined information system components] with regard to its security state or posture.

Note: This control enhancement misleads potential adversaries regarding the nature and extent of security safeguards deployed by organizations. As a result, adversaries may employ incorrect (and as a result ineffective) attack techniques. One way of misleading adversaries is for organizations to place misleading information regarding the specific security controls deployed in external information systems that are known to be accessed or targeted by adversaries. Another technique is the use of deception nets (e.g., honeynets, virtualized environments) that mimic actual aspects of organizational information systems but use, for example, out-of-date software configurations.

SC-30 (5) : CONCEALMENT OF SYSTEM COMPONENTS

> The organization employs [Assignment: organization-defined techniques] to hide or conceal [Assignment: organization-defined information system components].

Note: By hiding, disguising, or otherwise concealing critical information system components, organizations may be able to decrease the probability that adversaries target and successfully compromise those assets. Potential means for organizations to hide and/or conceal information system components include, for example, configuration of routers or the use of honeynets or virtualization techniques.

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-31 : COVERT CHANNEL ANALYSIS

Priority: P0

> The organization:

Note: Developers are in the best position to identify potential areas within systems that might lead to covert channels. Covert channel analysis is a meaningful activity when there is the potential for unauthorized information flows across security domains, for example, in the case of information systems containing export-controlled information and having connections to external networks (i.e., networks not controlled by organizations). Covert channel analysis is also meaningful for multilevel secure (MLS) information systems, multiple security level (MSL) systems, and cross-domain systems.

Related Controls: *AC-3*, *AC-4*, *PL-2*

SC-31a.

Performs a covert channel analysis to identify those aspects of communications within the information system that are potential avenues for covert [Selection (one or more): storage; timing] channels; and

SC-31b.

Estimates the maximum bandwidth of those channels.

SC-31 (1) : TEST COVERT CHANNELS FOR EXPLOITABILITY

The organization tests a subset of the identified covert channels to determine which channels are exploitable.

SC-31 (2) : MAXIMUM BANDWIDTH

The organization reduces the maximum bandwidth for identified covert [Selection (one or more); storage; timing] channels to [Assignment: organization-defined values].

Note: Information system developers are in the best position to reduce the maximum bandwidth for identified covert storage and timing channels.

SC-31 (3) : MEASURE BANDWIDTH IN OPERATIONAL ENVIRONMENTS

The organization measures the bandwidth of [Assignment: organization-defined subset of identified covert channels] in the operational environment of the information system.

Note: This control enhancement addresses covert channel bandwidth in operational environments versus developmental environments. Measuring covert channel bandwidth in operational environments helps organizations to determine how much information can be covertly leaked before such leakage adversely affects organizational missions/business functions. Covert channel bandwidth may be significantly different when measured in those settings that are independent of the particular environments of operation (e.g., laboratories or development environments).

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-32 : INFORMATION SYSTEM PARTITIONING

Priority: P0

The organization partitions the information system into [Assignment: organization-defined information system components] residing in separate physical domains or environments based on [Assignment: organization-defined circumstances for physical separation of components].

Note: Information system partitioning is a part of a defense-in-depth protection strategy. Organizations determine the degree of physical separation of system components from physically distinct components in separate racks in the same room, to components in separate rooms for the more critical components, to more significant geographical separation of the most critical components. Security categorization can guide the selection of appropriate candidates for domain partitioning. Managed interfaces restrict or prohibit network access and information flow among partitioned information system components.

Related Controls: *AC-4*, *SA-8*, *SC-2*, *SC-3*, *SC-7*

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-33 : TRANSMISSION PREPARATION INTEGRITY

[Withdrawn: Incorporated into SC-8].

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-34 : NON-MODIFIABLE EXECUTABLE PROGRAMS

Priority: P0

The information system at [Assignment: organization-defined information system components]:

Note: The term operating environment is defined as the specific code that hosts applications, for example, operating systems, executives, or monitors including virtual machine monitors (i.e., hypervisors). It can also include certain applications running directly on hardware platforms. Hardware-enforced, read-only media include, for example, Compact Disk-Recordable (CD-R)/Digital Video Disk-Recordable (DVD-R) disk drives and one-time programmable read-only memory. The use of non-modifiable storage ensures the integrity of software from the point of creation of the read-only image. The use of reprogrammable read-only memory can be accepted as read-only media provided: (i) integrity can be adequately protected from the point of initial writing to the insertion of the memory into the information system; and (ii) there are reliable hardware protections against reprogramming the memory while installed in organizational information systems.

Related Controls: *AC-3*, *SI-7*

SC-34a.

Loads and executes the operating environment from hardware-enforced, read-only media; and

SC-34b.

Loads and executes [Assignment: organization-defined applications] from hardware-enforced, read-only media.

SC-34 (1) : NO WRITABLE STORAGE

The organization employs [Assignment: organization-defined information system components] with no writeable storage that is persistent across component restart or power on/off.

Note: This control enhancement: (i) eliminates the possibility of malicious code insertion via persistent, writeable storage within the designated information system components; and (ii) applies to both fixed and removable storage, with the latter being addressed directly or as specific restrictions imposed through access controls for mobile devices.

Related Controls: *AC-19*, *MP-7*

SC-34 (2) : INTEGRITY PROTECTION / READ-ONLY MEDIA

The organization protects the integrity of information prior to storage on read-only media and controls the media after such information has been recorded onto the media.

Note: Security safeguards prevent the substitution of media into information systems or the reprogramming of programmable read-only media prior to installation into the systems. Security safeguards include, for example, a combination of prevention, detection, and response.

Related Controls: *AC-5*, *CM-3*, *CM-5*, *CM-9*, *MP-2*, *MP-4*, *MP-5*, *SA-12*, *SC-28*, *SI-3*

SC-34 (3) : HARDWARE-BASED PROTECTION

The organization:

SC-34 (3)(a)

Employs hardware-based, write-protect for [Assignment: organization-defined information system firmware components]; and

SC-34 (3)(b)

Implements specific procedures for [Assignment: organization-defined authorized individuals] to manually disable hardware write-protect for firmware modifications and re-enable the write-protect prior to returning to operational mode.

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-35 : HONEYCLIENTS

Priority: P0

The information system includes components that proactively seek to identify malicious websites and/or web-based malicious code.

Note: Honeyclients differ from honeypots in that the components actively probe the Internet in search of malicious code (e.g., worms) contained on external websites. As with honeypots, honeyclients require some supporting isolation measures (e.g., virtualization) to ensure that any malicious code discovered during the search and subsequently executed does not infect organizational information systems.

Related Controls: *SC-26*, *SC-44*, *SI-3*, *SI-4*

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-36 : DISTRIBUTED PROCESSING AND STORAGE

Priority: P0

> The organization distributes [Assignment: organization-defined processing and storage] across multiple physical locations.

Note: Distributing processing and storage across multiple physical locations provides some degree of redundancy or overlap for organizations, and therefore increases the work factor of adversaries to adversely impact organizational operations, assets, and individuals. This control does not assume a single primary processing or storage location, and thus allows for parallel processing and storage.

Related Controls: *CP-6*, *CP-7*

SC-36 (1) : POLLING TECHNIQUES

> The organization employs polling techniques to identify potential faults, errors, or compromises to [Assignment: organization-defined distributed processing and storage components].

Note: Distributed processing and/or storage may be employed to reduce opportunities for adversaries to successfully compromise the confidentiality, integrity, or availability of information and information systems. However, distribution of processing and/or storage components does not prevent adversaries from compromising one (or more) of the distributed components. Polling compares the processing results and/or storage content from the various distributed components and subsequently voting on the outcomes. Polling identifies potential faults, errors, or compromises in distributed processing and/or storage components.

Related Controls: *SI-4*

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-37 : OUT-OF-BAND CHANNELS

Priority: P0

> The organization employs [Assignment: organization-defined out-of-band channels] for the physical delivery or electronic transmission of [Assignment: organization-defined information, information system components, or devices] to [Assignment: organization-defined individuals or information systems].

Note: Out-of-band channels include, for example, local (nonnetwork) accesses to information systems, network paths physically separate from network paths used for operational traffic, or nonelectronic paths such as the US Postal Service. This is in contrast with using the same channels (i.e., in-band channels) that carry routine operational traffic. Out-of-band channels do not have the same vulnerability/exposure as in-band channels, and hence the confidentiality, integrity, or availability compromises of in-band channels will not compromise the out-of-band channels. Organizations may employ out-of-band channels in the delivery or transmission of many organizational items including, for example, identifiers/authenticators, configuration management changes for hardware, firmware, or software, cryptographic key management information, security updates, system/data backups, maintenance information, and malicious code protection updates.

Related Controls: *AC-2*, *CM-3*, *CM-5*, *CM-7*, *IA-4*, *IA-5*, *MA-4*, *SC-12*, *SI-3*, *SI-4*, *SI-7*

SC-37 (1) : ENSURE DELIVERY / TRANSMISSION

> The organization employs [Assignment: organization-defined security safeguards] to ensure that only [Assignment: organization-defined individuals or information systems] receive the [Assignment: organization-defined information, information system components, or devices].

Note: Techniques and/or methods employed by organizations to ensure that only designated information systems or individuals receive particular information, system components, or devices include, for example, sending authenticators via courier service but requiring recipients to show some form of government-issued photographic identification as a condition of receipt.

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-38 : OPERATIONS SECURITY

Priority: P0

> The organization employs [Assignment: organization-defined operations security safeguards] to protect key organizational information throughout the system development life cycle.

Note: Operations security (OPSEC) is a systematic process by which potential adversaries can be denied information about the capabilities and intentions of organizations by identifying, controlling, and protecting generally unclassified information that specifically relates to the planning and execution of sensitive organizational activities. The OPSEC process involves five steps: (i) identification of critical information (e.g., the security categorization process); (ii) analysis of threats; (iii) analysis of vulnerabilities; (iv) assessment of risks; and (v) the application of appropriate countermeasures. OPSEC safeguards are applied to both organizational information systems and the environments in which those systems operate. OPSEC safeguards help to protect the confidentiality of key information including, for example, limiting the sharing of information with suppliers and potential suppliers of information system components, information technology products and services, and with other non-organizational elements and individuals. Information critical to mission/business success includes, for example, user identities, element uses, suppliers, supply chain processes, functional and security requirements, system design specifications, testing protocols, and security control implementation details.

Related Controls: *RA-2*, *RA-5*, *SA-12*

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-39 : PROCESS ISOLATION

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The information system maintains a separate execution domain for each executing process.

Note: Information systems can maintain separate execution domains for each executing process by assigning each process a separate address space. Each information system process has a distinct address space so that communication between processes is performed in a manner controlled through the security functions, and one process cannot modify the executing code of another process. Maintaining separate execution domains for executing processes can be achieved, for example, by implementing separate address spaces. This capability is available in most commercial operating systems that employ multi-state processor technologies.

Related Controls: *AC-3*, *AC-4*, *AC-6*, *SA-4*, *SA-5*, *SA-8*, *SC-2*, *SC-3*

SC-39 (1) : HARDWARE SEPARATION

> The information system implements underlying hardware separation mechanisms to facilitate process separation.

Note: Hardware-based separation of information system processes is generally less susceptible to compromise than software-based separation, thus providing greater assurance that the separation will be enforced. Underlying hardware separation mechanisms include, for example, hardware memory management.

SC-39 (2) : THREAD ISOLATION

> The information system maintains a separate execution domain for each thread in [Assignment: organization-defined multi-threaded processing].

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-40 : WIRELESS LINK PROTECTION

Priority: P0

> The information system protects external and internal [Assignment: organization-defined wireless links] from [Assignment: organization-defined types of signal parameter attacks or references to sources for such attacks].

Note: This control applies to internal and external wireless communication links that may be visible to individuals who are not authorized information system users. Adversaries can exploit the signal parameters of wireless links if such links are not adequately protected. There are many ways to exploit the signal parameters of wireless links to gain intelligence, deny service, or to spoof users of organizational information systems. This control reduces the impact of attacks that are unique to wireless systems. If organizations rely on commercial service providers for transmission services as commodity items rather than as fully dedicated services, it may not be possible to implement this control.

Related Controls: *AC-18*, *SC-5*

SC-40 (1) : ELECTROMAGNETIC INTERFERENCE

> The information system implements cryptographic mechanisms that achieve [Assignment: organization-defined level of protection] against the effects of intentional electromagnetic interference.

Note: This control enhancement protects against intentional jamming that might deny or impair communications by ensuring that wireless spread spectrum waveforms used to provide anti-jam protection are not predictable by unauthorized individuals. The control enhancement may also coincidentally help to mitigate the effects of unintentional jamming due to interference from legitimate transmitters sharing the same spectrum. Mission requirements, projected threats, concept of operations, and applicable legislation, directives, regulations, policies, standards, and guidelines determine levels of wireless link availability and performance/cryptography needed.

Related Controls: *SC-12*, *SC-13*

SC-40 (2) : REDUCE DETECTION POTENTIAL

> The information system implements cryptographic mechanisms to reduce the detection potential of wireless links to [Assignment: organization-defined level of reduction].

Note: This control enhancement is needed for covert communications and protecting wireless transmitters from being geo-located by their transmissions. The control enhancement ensures that spread spectrum waveforms used to achieve low probability of detection are not predictable by unauthorized individuals. Mission requirements, projected threats, concept of operations, and applicable legislation, directives, regulations, policies, standards, and guidelines determine the levels to which wireless links should be undetectable.

Related Controls: *SC-12*, *SC-13*

SC-40 (3) : IMITATIVE OR MANIPULATIVE COMMUNICATIONS DECEPTION

> The information system implements cryptographic mechanisms to identify and reject wireless transmissions that are deliberate attempts to achieve imitative or manipulative communications deception based on signal parameters.

Note: This control enhancement ensures that the signal parameters of wireless transmissions are not predictable by unauthorized individuals. Such unpredictability reduces the probability of imitative or manipulative communications deception based upon signal parameters alone.

Related Controls: *SC-12*, *SC-13*

SC-40 (4) : SIGNAL PARAMETER IDENTIFICATION

The information system implements cryptographic mechanisms to prevent the identification of [Assignment: organization-defined wireless transmitters] by using the transmitter signal parameters.

Note: Radio fingerprinting techniques identify the unique signal parameters of transmitters to fingerprint such transmitters for purposes of tracking and mission/user identification. This control enhancement protects against the unique identification of wireless transmitters for purposes of intelligence exploitation by ensuring that anti-fingerprinting alterations to signal parameters are not predictable by unauthorized individuals. This control enhancement helps assure mission success when anonymity is required.

Related Controls: *SC-12*, *SC-13*

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-41 : PORT AND I/O DEVICE ACCESS

Priority: P0

The organization physically disables or removes [Assignment: organization-defined connection ports or input/output devices] on [Assignment: organization-defined information systems or information system components].

Note: Connection ports include, for example, Universal Serial Bus (USB) and Firewire (IEEE 1394). Input/output (I/O) devices include, for example, Compact Disk (CD) and Digital Video Disk (DVD) drives. Physically disabling or removing such connection ports and I/O devices helps prevent exfiltration of information from information systems and the introduction of malicious code into systems from those ports/devices.

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-42 : SENSOR CAPABILITY AND DATA

Priority: P0

The information system:

Note: This control often applies to types of information systems or system components characterized as mobile devices, for example, smart phones, tablets, and E-readers. These systems often include sensors that can collect and record data regarding the environment where the system is in use. Sensors that are embedded within mobile devices include, for example, cameras, microphones, Global Positioning System (GPS) mechanisms, and accelerometers. While the sensors on mobiles devices provide an important function, if activated covertly, such devices can potentially provide a means for adversaries to learn valuable information about individuals and organizations. For example, remotely activating the GPS function on a mobile device could provide an adversary with the ability to track the specific movements of an individual.

SC-42a.

Prohibits the remote activation of environmental sensing capabilities with the following exceptions: [Assignment: organization-defined exceptions where remote activation of sensors is allowed]; and

SC-42b.

Provides an explicit indication of sensor use to [Assignment: organization-defined class of users].

SC-42 (1) : REPORTING TO AUTHORIZED INDIVIDUALS OR ROLES

The organization ensures that the information system is configured so that data or information collected by the [Assignment: organization-defined sensors] is only reported to authorized individuals or roles.

Note: In situations where sensors are activated by authorized individuals (e.g., end users), it is still possible that the data/information collected by the sensors will be sent to unauthorized entities.

SC-42 (2) : AUTHORIZED USE

The organization employs the following measures: [Assignment: organization-defined measures], so that data or information collected by [Assignment: organization-defined sensors] is only used for authorized purposes.

Note: Information collected by sensors for a specific authorized purpose potentially could be misused for some unauthorized purpose. For example, GPS sensors that are used to support traffic navigation could be misused to track movements of individuals. Measures to mitigate such activities include, for example, additional training to ensure that authorized parties do not abuse their authority, or (in the case where sensor data/information is maintained by external parties) contractual restrictions on the use of the data/information.

SC-42 (3) : PROHIBIT USE OF DEVICES

The organization prohibits the use of devices possessing [Assignment: organization-defined environmental sensing capabilities] in [Assignment: organization-defined facilities, areas, or systems].

Note: For example, organizations may prohibit individuals from bringing cell phones or digital cameras into certain facilities or specific controlled areas within facilities where classified information is stored or sensitive conversations are taking place.

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-43 : USAGE RESTRICTIONS

Priority: P0

The organization:

Note: Information system components include hardware, software, or firmware components (e.g., Voice Over Internet Protocol, mobile code, digital copiers, printers, scanners, optical devices, wireless technologies, mobile devices).

Related Controls: *CM-6*, *SC-7*

SC-43a.

Establishes usage restrictions and implementation guidance for [Assignment: organization-defined information system components] based on the potential to cause damage to the information system if used maliciously; and

SC-43b.

Authorizes, monitors, and controls the use of such components within the information system.

Control Family: SYSTEM AND COMMUNICATIONS PROTECTION

SC-44 : DETONATION CHAMBERS

Priority: P0

The organization employs a detonation chamber capability within [Assignment: organization-defined information system, system component, or location].

Note: Detonation chambers, also known as dynamic execution environments, allow organizations to open email attachments, execute untrusted or suspicious applications, and execute Universal Resource Locator (URL) requests in the safety of an isolated environment or virtualized sandbox. These protected and isolated execution environments provide a means of determining whether the associated attachments/applications contain malicious code. While related to the concept of deception nets, the control is not intended to maintain a long-term environment in which adversaries can operate and their actions can be observed. Rather, it is intended to quickly identify malicious code and reduce the likelihood that the code is propagated to user environments of operation (or prevent such propagation completely).

Related Controls: *SC-7*, *SC-25*, *SC-26*, *SC-30*

Control Family: SYSTEM AND INFORMATION INTEGRITY

SI-1 : SYSTEM AND INFORMATION INTEGRITY POLICY AND PROCEDURES

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: This control addresses the establishment of policy and procedures for the effective implementation of selected security controls and control enhancements in the SI family. Policy and procedures reflect applicable federal laws, Executive Orders, directives, regulations, policies, standards, and guidance. Security program policies and procedures at the organization level may make the need for system-specific policies and procedures unnecessary. The policy can be included as part of the general information security policy for organizations or conversely, can be represented by multiple policies reflecting the complex nature of certain organizations. The procedures can be established for the security program in general and for particular information systems, if needed. The organizational risk management strategy is a key factor in establishing policy and procedures.

Related Controls: *PM-9*

SI-1a.

Develops, documents, and disseminates to [Assignment: organization-defined personnel or roles]:

SI-1a.1.

A system and information integrity policy that addresses purpose, scope, roles, responsibilities, management commitment, coordination among organizational entities, and compliance; and

SI-1a.2.

Procedures to facilitate the implementation of the system and information integrity policy and associated system and information integrity controls; and

SI-1b.

Reviews and updates the current:

SI-1b.1.

System and information integrity policy [Assignment: organization-defined frequency]; and

SI-1b.2.

System and information integrity procedures [Assignment: organization-defined frequency].

Control Family: SYSTEM AND INFORMATION INTEGRITY

SI-2 : FLAW REMEDIATION

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: Organizations identify information systems affected by announced software flaws including potential vulnerabilities resulting from those flaws, and report this information to designated organizational personnel with information security responsibilities. Security-relevant software updates include, for example, patches, service packs, hot fixes, and anti-virus signatures. Organizations also address flaws discovered during security assessments, continuous monitoring, incident response activities, and system error handling. Organizations take advantage of available resources such as the Common Weakness Enumeration (CWE) or Common Vulnerabilities and Exposures (CVE) databases in remediating flaws discovered in organizational information systems. By incorporating flaw remediation into ongoing configuration management processes, required/anticipated remediation actions can be tracked and verified. Flaw remediation actions that can be tracked and verified include, for example, determining whether organizations follow US-CERT guidance and Information Assurance Vulnerability Alerts. Organization-defined time periods for updating security-relevant software and firmware may vary based on a variety of factors including, for example, the security category of the information system or the criticality of the update (i.e., severity of the vulnerability related to the discovered flaw). Some types of flaw remediation may require more testing than other types. Organizations determine the degree and type of testing needed for the specific type of flaw remediation activity under consideration and also the types of changes that are to be configuration-managed. In some situations, organizations may determine that the testing of software and/or firmware updates is not necessary or practical, for example, when implementing simple anti-virus signature updates. Organizations may also consider in testing decisions, whether security-relevant software or firmware updates are obtained from authorized sources with appropriate digital signatures.

Related Controls: *CA-2*, *CA-7*, *CM-3*, *CM-5*, *CM-8*, *MA-2*, *IR-4*, *RA-5*, *SA-10*, *SA-11*, *SI-11*

SI-2a.

Identifies, reports, and corrects information system flaws;

SI-2b.

Tests software and firmware updates related to flaw remediation for effectiveness and potential side effects before installation;

SI-2c.

Installs security-relevant software and firmware updates within [Assignment: organization-defined time period] of the release of the updates; and

SI-2d.

Incorporates flaw remediation into the organizational configuration management process.

SI-2 (1) : CENTRAL MANAGEMENT

Baseline-Impact: HIGH

The organization centrally manages the flaw remediation process.

Note: Central management is the organization-wide management and implementation of flaw remediation processes. Central management includes planning, implementing, assessing, authorizing, and monitoring the organization-defined, centrally managed flaw remediation security controls.

SI-2 (2) : AUTOMATED FLAW REMEDIATION STATUS

Baseline-Impact: *MODERATE*, **HIGH**

The organization employs automated mechanisms [Assignment: organization-defined frequency] to determine the state of information system components with regard to flaw remediation.

Related Controls: *CM-6*, *SI-4*

SI-2 (3) : TIME TO REMEDIATE FLAWS / BENCHMARKS FOR CORRECTIVE ACTIONS

The organization:

Note: This control enhancement requires organizations to determine the current time it takes on the average to correct information system flaws after such flaws have been identified, and subsequently establish organizational benchmarks (i.e., time frames) for taking corrective actions. Benchmarks can be established by type of flaw and/or severity of the potential vulnerability if the flaw can be exploited.

SI-2 (3)(a)

Measures the time between flaw identification and flaw remediation; and

SI-2 (3)(b)

Establishes [Assignment: organization-defined benchmarks] for taking corrective actions.

SI-2 (4) : AUTOMATED PATCH MANAGEMENT TOOLS

[Withdrawn: Incorporated into SI-2].

SI-2 (5) : AUTOMATIC SOFTWARE / FIRMWARE UPDATES

The organization installs [Assignment: organization-defined security-relevant software and firmware updates] automatically to [Assignment: organization-defined information system components].

Note: Due to information system integrity and availability concerns, organizations give careful consideration to the methodology used to carry out automatic updates. Organizations must balance the need to ensure that the updates are installed as soon as possible with the need to maintain configuration management and with any mission or operational impacts that automatic updates might impose.

SI-2 (6) : REMOVAL OF PREVIOUS VERSIONS OF SOFTWARE / FIRMWARE

The organization removes [Assignment: organization-defined software and firmware components] after updated versions have been installed.

Note: Previous versions of software and/or firmware components that are not removed from the information system after updates have been installed may be exploited by adversaries. Some information technology products may remove older versions of software and/or firmware automatically from the information system.

Control Family: SYSTEM AND INFORMATION INTEGRITY

SI-3 : MALICIOUS CODE PROTECTION

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: Information system entry and exit points include, for example, firewalls, electronic mail servers, web servers, proxy servers, remote-access servers, workstations, notebook computers, and mobile devices. Malicious code includes, for example, viruses, worms, Trojan horses, and spyware. Malicious code can also be encoded in various formats (e.g., UUENCODE, Unicode), contained within compressed or hidden files, or hidden in files using steganography. Malicious code can be transported by different means including, for example, web accesses, electronic mail, electronic mail attachments, and portable storage devices. Malicious code insertions occur through the exploitation of information system vulnerabilities. Malicious code protection mechanisms include, for example, anti-virus signature definitions and reputation-based technologies. A variety of technologies and methods exist to limit or eliminate the effects of malicious code. Pervasive configuration management and comprehensive software integrity controls may be effective in preventing execution of unauthorized code. In addition to commercial off-the-shelf software, malicious code may also be present in custom-built software. This could include, for example, logic bombs, back doors, and other types of cyber attacks that could affect organizational missions/business functions. Traditional malicious code protection mechanisms cannot always detect such code. In these situations, organizations rely instead on other safeguards including, for example, secure coding practices, configuration management and control, trusted procurement processes, and monitoring practices to help ensure that software does not perform functions other than the functions intended. Organizations may determine that in response to the detection of malicious code, different actions may be warranted. For example, organizations can define actions in response to malicious code detection during periodic scans, actions in response to detection of malicious downloads, and/or actions in response to detection of maliciousness when attempting to open or execute files.

Related Controls: *CM-3*, *MP-2*, *SA-4*, *SA-8*, *SA-12*, *SA-13*, *SC-7*, *SC-26*, *SC-44*, *SI-2*, *SI-4*, *SI-7*

SI-3a.

Employs malicious code protection mechanisms at information system entry and exit points to detect and eradicate malicious code;

SI-3b.

Updates malicious code protection mechanisms whenever new releases are available in accordance with organizational configuration management policy and procedures;

SI-3c.

Configures malicious code protection mechanisms to:

SI-3c.1.

Perform periodic scans of the information system [Assignment: organization-defined frequency] and real-time scans of files from external sources at [Selection (one or more); endpoint; network entry/exit points] as the files are downloaded, opened, or executed in accordance with organizational security policy; and

SI-3c.2.

[Selection (one or more): block malicious code; quarantine malicious code; send alert to administrator; [Assignment: organization-defined action]] in response to malicious code detection; and

SI-3d.

Addresses the receipt of false positives during malicious code detection and eradication and the resulting potential impact on the availability of the information system.

SI-3 (1) : CENTRAL MANAGEMENT

Baseline-Impact: *MODERATE*, **HIGH**

The organization centrally manages malicious code protection mechanisms.

Note: Central management is the organization-wide management and implementation of malicious code protection mechanisms. Central management includes planning, implementing, assessing, authorizing, and monitoring the organization-defined, centrally managed flaw malicious code protection security controls.

Related Controls: *AU-2*, *SI-8*

SI-3 (2) : AUTOMATIC UPDATES

Baseline-Impact: *MODERATE*, **HIGH**

The information system automatically updates malicious code protection mechanisms.

Note: Malicious code protection mechanisms include, for example, signature definitions. Due to information system integrity and availability concerns, organizations give careful consideration to the methodology used to carry out automatic updates.

Related Controls: *SI-8*

SI-3 (3) : NON-PRIVILEGED USERS

[Withdrawn: Incorporated into AC-6 (10)].

SI-3 (4) : UPDATES ONLY BY PRIVILEGED USERS

The information system updates malicious code protection mechanisms only when directed by a privileged user.

Note: This control enhancement may be appropriate for situations where for reasons of security or operational continuity, updates are only applied when selected/approved by designated organizational personnel.

Related Controls: *AC-6*, *CM-5*

SI-3 (5) : PORTABLE STORAGE DEVICES

[Withdrawn: Incorporated into MP-7].

SI-3 (6) : TESTING / VERIFICATION

The organization:

Related Controls: *CA-2*, *CA-7*, *RA-5*

SI-3 (6)(a)

Tests malicious code protection mechanisms [Assignment: organization-defined frequency] by introducing a known benign, non-spreading test case into the information system; and

SI-3 (6)(b)

Verifies that both detection of the test case and associated incident reporting occur.

SI-3 (7) : NONSIGNATURE-BASED DETECTION

The information system implements nonsignature-based malicious code detection mechanisms.

Note: Nonsignature-based detection mechanisms include, for example, the use of heuristics to detect, analyze, and describe the characteristics or behavior of malicious code and to provide safeguards against malicious code for which signatures do not yet exist or for which existing signatures may not be effective. This includes polymorphic malicious code (i.e., code that changes signatures when it replicates). This control enhancement does not preclude the use of signature-based detection mechanisms.

SI-3 (8) : DETECT UNAUTHORIZED COMMANDS

The information system detects [Assignment: organization-defined unauthorized operating system commands] through the kernel application programming interface at [Assignment: organization-defined information system hardware components] and [Selection (one or more): issues a warning; audits the command execution; prevents the execution of the command].

Note: This control enhancement can also be applied to critical interfaces other than kernel-based interfaces, including for example, interfaces with virtual machines and privileged applications. Unauthorized operating system commands include, for example, commands for kernel functions from information system processes that are not trusted to initiate such commands, or commands for kernel functions that are suspicious even though commands of that type are reasonable for processes to initiate. Organizations can define the malicious commands to be detected by a combination of command types, command classes, or specific instances of commands. Organizations can define hardware components by specific component, component type, location in the network, or combination therein. Organizations may select different actions for different types/classes/specific instances of potentially malicious commands.

Related Controls: *AU-6*

SI-3 (9) : AUTHENTICATE REMOTE COMMANDS

The information system implements [Assignment: organization-defined security safeguards] to authenticate [Assignment: organization-defined remote commands].

Note: This control enhancement protects against unauthorized commands and replay of authorized commands. This capability is important for those remote information systems whose loss, malfunction, misdirection, or exploitation would have immediate and/or serious consequences (e.g., injury or death, property damage, loss of high-valued assets or sensitive information, or failure of important missions/business functions). Authentication safeguards for remote commands help to ensure that information systems accept and execute in the order intended, only authorized commands, and that unauthorized commands are rejected. Cryptographic mechanisms can be employed, for example, to authenticate remote commands.

Related Controls: *SC-12*, *SC-13*, *SC-23*

SI-3 (10) : MALICIOUS CODE ANALYSIS

The organization:

Note: The application of selected malicious code analysis tools and techniques provides organizations with a more in-depth understanding of adversary tradecraft (i.e., tactics, techniques, and procedures) and the functionality and purpose of specific instances of malicious code. Understanding the characteristics of malicious code facilitates more effective organizational responses to current and future threats. Organizations can conduct malicious code analyses by using reverse engineering techniques or by monitoring the behavior of executing code.

SI-3 (10)(a)

Employs [Assignment: organization-defined tools and techniques] to analyze the characteristics and behavior of malicious code; and

SI-3 (10)(b)

Incorporates the results from malicious code analysis into organizational incident response and flaw remediation processes.

Control Family: SYSTEM AND INFORMATION INTEGRITY

SI-4 : INFORMATION SYSTEM MONITORING

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: Information system monitoring includes external and internal monitoring. External monitoring includes the observation of events occurring at the information system boundary (i.e., part of perimeter defense and boundary protection). Internal monitoring includes the observation of events occurring within the information system. Organizations can monitor information systems, for example, by observing audit activities in real time or by observing other system aspects such as access patterns, characteristics of access, and other actions. The monitoring objectives may guide determination of the events. Information system monitoring capability is achieved through a variety of tools and techniques (e.g., intrusion detection systems, intrusion prevention systems, malicious code protection software, scanning tools, audit record monitoring software, network monitoring software). Strategic locations for monitoring devices include, for example, selected perimeter locations and near server farms supporting critical applications, with such devices typically being employed at the managed interfaces associated with controls SC-7 and AC-17. Einstein network monitoring devices from the Department of Homeland Security can also be included as monitoring devices. The granularity of monitoring information collected is based on organizational monitoring objectives and the capability of information systems to support such objectives. Specific types of transactions of interest include, for example, Hyper Text Transfer Protocol (HTTP) traffic that bypasses HTTP proxies. Information system monitoring is an integral part of organizational continuous monitoring and incident response programs. Output from system monitoring serves as input to continuous monitoring and incident response programs. A network connection is any connection with a device that communicates through a network (e.g., local area network, Internet). A remote connection is any connection with a device communicating through an external network (e.g., the Internet). Local, network, and remote connections can be either wired or wireless.

Related Controls: *AC-3*, *AC-4*, *AC-8*, *AC-17*, *AU-2*, *AU-6*, *AU-7*, *AU-9*, *AU-12*, *CA-7*, *IR-4*, *PE-3*, *RA-5*, *SC-7*, *SC-26*, *SC-35*, *SI-3*, *SI-7*

SI-4a.

Monitors the information system to detect:

SI-4a.1.

Attacks and indicators of potential attacks in accordance with [Assignment: organization-defined monitoring objectives]; and

SI-4a.2.

Unauthorized local, network, and remote connections;

SI-4b.

Identifies unauthorized use of the information system through [Assignment: organization-defined techniques and methods];

SI-4c.

Deploys monitoring devices:

SI-4c.1.

Strategically within the information system to collect organization-determined essential information; and

SI-4c.2.

At ad hoc locations within the system to track specific types of transactions of interest to the organization;

SI-4d.

Protects information obtained from intrusion-monitoring tools from unauthorized access, modification, and deletion;

SI-4e.

Heightens the level of information system monitoring activity whenever there is an indication of increased risk to organizational operations and assets, individuals, other organizations, or the Nation based on law enforcement information, intelligence information, or other credible sources of information;

SI-4f.

Obtains legal opinion with regard to information system monitoring activities in accordance with applicable federal laws, Executive Orders, directives, policies, or regulations; and

SI-4g.

Provides [Assignment: organization-defined information system monitoring information] to [Assignment: organization-defined personnel or roles] [Selection (one or more): as needed; [Assignment: organization-defined frequency]].

SI-4 (1) : SYSTEM-WIDE INTRUSION DETECTION SYSTEM

The organization connects and configures individual intrusion detection tools into an information system-wide intrusion detection system.

SI-4 (2) : AUTOMATED TOOLS FOR REAL-TIME ANALYSIS

Baseline-Impact: *MODERATE*, **HIGH**

The organization employs automated tools to support near real-time analysis of events.

Note: Automated tools include, for example, host-based, network-based, transport-based, or storage-based event monitoring tools or Security Information and Event Management (SIEM) technologies that provide real time analysis of alerts and/or notifications generated by organizational information systems.

SI-4 (3) : AUTOMATED TOOL INTEGRATION

The organization employs automated tools to integrate intrusion detection tools into access control and flow control mechanisms for rapid response to attacks by enabling reconfiguration of these mechanisms in support of attack isolation and elimination.

SI-4 (4) : INBOUND AND OUTBOUND COMMUNICATIONS TRAFFIC

Baseline-Impact: *MODERATE*, **HIGH**

The information system monitors inbound and outbound communications traffic [Assignment: organization-defined frequency] for unusual or unauthorized activities or conditions.

Note: Unusual/unauthorized activities or conditions related to information system inbound and outbound communications traffic include, for example, internal traffic that indicates the presence of malicious code within organizational information systems or propagating among system components, the unauthorized exporting of information, or signaling to external information systems. Evidence of malicious code is used to identify potentially compromised information systems or information system components.

SI-4 (5) : SYSTEM-GENERATED ALERTS

Baseline-Impact: *MODERATE*, **HIGH**

The information system alerts [Assignment: organization-defined personnel or roles] when the following indications of compromise or potential compromise occur: [Assignment: organization-defined compromise indicators].

Note: Alerts may be generated from a variety of sources, including, for example, audit records or inputs from malicious code protection mechanisms, intrusion detection or prevention mechanisms, or boundary protection devices such as firewalls, gateways, and routers. Alerts can be transmitted, for example, telephonically, by electronic mail messages, or by text messaging. Organizational personnel on the notification list can include, for example, system administrators, mission/business owners, system owners, or information system security officers.

Related Controls: *AU-5*, *PE-6*

SI-4 (6) : RESTRICT NON-PRIVILEGED USERS

[Withdrawn: Incorporated into AC-6 (10)].

SI-4 (7) : AUTOMATED RESPONSE TO SUSPICIOUS EVENTS

> The information system notifies [Assignment: organization-defined incident response personnel (identified by name and/or by role)] of detected suspicious events and takes [Assignment: organization-defined least-disruptive actions to terminate suspicious events].

Note: Least-disruptive actions may include, for example, initiating requests for human responses.

SI-4 (8) : PROTECTION OF MONITORING INFORMATION

> [Withdrawn: Incorporated into SI-4].

SI-4 (9) : TESTING OF MONITORING TOOLS

> The organization tests intrusion-monitoring tools [Assignment: organization-defined frequency].

Note: Testing intrusion-monitoring tools is necessary to ensure that the tools are operating correctly and continue to meet the monitoring objectives of organizations. The frequency of testing depends on the types of tools used by organizations and methods of deployment.

Related Controls: *CP-9*

SI-4 (10) : VISIBILITY OF ENCRYPTED COMMUNICATIONS

> The organization makes provisions so that [Assignment: organization-defined encrypted communications traffic] is visible to [Assignment: organization-defined information system monitoring tools].

Note: Organizations balance the potentially conflicting needs for encrypting communications traffic and for having insight into such traffic from a monitoring perspective. For some organizations, the need to ensure the confidentiality of communications traffic is paramount; for others, mission-assurance is of greater concern. Organizations determine whether the visibility requirement applies to internal encrypted traffic, encrypted traffic intended for external destinations, or a subset of the traffic types.

SI-4 (11) : ANALYZE COMMUNICATIONS TRAFFIC ANOMALIES

> The organization analyzes outbound communications traffic at the external boundary of the information system and selected [Assignment: organization-defined interior points within the system (e.g., subnetworks, subsystems)] to discover anomalies.

Note: Anomalies within organizational information systems include, for example, large file transfers, long-time persistent connections, unusual protocols and ports in use, and attempted communications with suspected malicious external addresses.

SI-4 (12) : AUTOMATED ALERTS

The organization employs automated mechanisms to alert security personnel of the following inappropriate or unusual activities with security implications: [Assignment: organization-defined activities that trigger alerts].

Note: This control enhancement focuses on the security alerts generated by organizations and transmitted using automated means. In contrast to the alerts generated by information systems in SI-4 (5), which tend to focus on information sources internal to the systems (e.g., audit records), the sources of information for this enhancement can include other entities as well (e.g., suspicious activity reports, reports on potential insider threats).

Related Controls: *AC-18*, *IA-3*

SI-4 (13) : ANALYZE TRAFFIC / EVENT PATTERNS

The organization:

SI-4 (13)(a)

Analyzes communications traffic/event patterns for the information system;

SI-4 (13)(b)

Develops profiles representing common traffic patterns and/or events; and

SI-4 (13)(c)

Uses the traffic/event profiles in tuning system-monitoring devices to reduce the number of false positives and the number of false negatives.

SI-4 (14) : WIRELESS INTRUSION DETECTION

The organization employs a wireless intrusion detection system to identify rogue wireless devices and to detect attack attempts and potential compromises/breaches to the information system.

Note: Wireless signals may radiate beyond the confines of organization-controlled facilities. Organizations proactively search for unauthorized wireless connections including the conduct of thorough scans for unauthorized wireless access points. Scans are not limited to those areas within facilities containing information systems, but also include areas outside of facilities as needed, to verify that unauthorized wireless access points are not connected to the systems.

Related Controls: *AC-18*, *IA-3*

SI-4 (15) : WIRELESS TO WIRELINE COMMUNICATIONS

> The organization employs an intrusion detection system to monitor wireless communications traffic as the traffic passes from wireless to wireline networks.

Related Controls: *AC-18*

SI-4 (16) : CORRELATE MONITORING INFORMATION

> The organization correlates information from monitoring tools employed throughout the information system.

Note: Correlating information from different monitoring tools can provide a more comprehensive view of information system activity. The correlation of monitoring tools that usually work in isolation (e.g., host monitoring, network monitoring, anti-virus software) can provide an organization-wide view and in so doing, may reveal otherwise unseen attack patterns. Understanding the capabilities/limitations of diverse monitoring tools and how to maximize the utility of information generated by those tools can help organizations to build, operate, and maintain effective monitoring programs.

Related Controls: *AU-6*

SI-4 (17) : INTEGRATED SITUATIONAL AWARENESS

> The organization correlates information from monitoring physical, cyber, and supply chain activities to achieve integrated, organization-wide situational awareness.

Note: This control enhancement correlates monitoring information from a more diverse set of information sources to achieve integrated situational awareness. Integrated situational awareness from a combination of physical, cyber, and supply chain monitoring activities enhances the capability of organizations to more quickly detect sophisticated cyber attacks and investigate the methods and techniques employed to carry out such attacks. In contrast to SI-4 (16) which correlates the various cyber monitoring information, this control enhancement correlates monitoring beyond just the cyber domain. Such monitoring may help reveal attacks on organizations that are operating across multiple attack vectors.

Related Controls: *SA-12*

SI-4 (18) : ANALYZE TRAFFIC / COVERT EXFILTRATION

> The organization analyzes outbound communications traffic at the external boundary of the information system (i.e., system perimeter) and at [Assignment: organization-defined interior points within the system (e.g., subsystems, subnetworks)] to detect covert exfiltration of information.

Note: Covert means that can be used for the unauthorized exfiltration of organizational information include, for example, steganography.

SI-4 (19) : INDIVIDUALS POSING GREATER RISK

> The organization implements [Assignment: organization-defined additional monitoring] of individuals who have been identified by [Assignment: organization-defined sources] as posing an increased level of risk.

Note: Indications of increased risk from individuals can be obtained from a variety of sources including, for example, human resource records, intelligence agencies, law enforcement organizations, and/or other credible sources. The monitoring of individuals is closely coordinated with management, legal, security, and human resources officials within organizations conducting such monitoring and complies with federal legislation, Executive Orders, policies, directives, regulations, and standards.

SI-4 (20) : PRIVILEGED USERS

> The organization implements [Assignment: organization-defined additional monitoring] of privileged users.

SI-4 (21) : PROBATIONARY PERIODS

> The organization implements [Assignment: organization-defined additional monitoring] of individuals during [Assignment: organization-defined probationary period].

SI-4 (22) : UNAUTHORIZED NETWORK SERVICES

> The information system detects network services that have not been authorized or approved by [Assignment: organization-defined authorization or approval processes] and [Selection (one or more): audits; alerts [Assignment: organization-defined personnel or roles]].

Note: Unauthorized or unapproved network services include, for example, services in service-oriented architectures that lack organizational verification or validation and therefore may be unreliable or serve as malicious rogues for valid services.

Related Controls: *AC-6*, *CM-7*, *SA-5*, *SA-9*

SI-4 (23) : HOST-BASED DEVICES

> The organization implements [Assignment: organization-defined host-based monitoring mechanisms] at [Assignment: organization-defined information system components].

Note: Information system components where host-based monitoring can be implemented include, for example, servers, workstations, and mobile devices. Organizations consider employing host-based monitoring mechanisms from multiple information technology product developers.

SI-4 (24) : INDICATORS OF COMPROMISE

The information system discovers, collects, distributes, and uses indicators of compromise.

Note: Indicators of compromise (IOC) are forensic artifacts from intrusions that are identified on organizational information systems (at the host or network level). IOCs provide organizations with valuable information on objects or information systems that have been compromised. IOCs for the discovery of compromised hosts can include for example, the creation of registry key values. IOCs for network traffic include, for example, Universal Resource Locator (URL) or protocol elements that indicate malware command and control servers. The rapid distribution and adoption of IOCs can improve information security by reducing the time that information systems and organizations are vulnerable to the same exploit or attack.

Control Family: SYSTEM AND INFORMATION INTEGRITY

SI-5 : SECURITY ALERTS, ADVISORIES, AND DIRECTIVES

Priority: P1

Baseline-Impact: LOW, *MODERATE*, **HIGH**

The organization:

Note: The United States Computer Emergency Readiness Team (US-CERT) generates security alerts and advisories to maintain situational awareness across the federal government. Security directives are issued by OMB or other designated organizations with the responsibility and authority to issue such directives. Compliance to security directives is essential due to the critical nature of many of these directives and the potential immediate adverse effects on organizational operations and assets, individuals, other organizations, and the Nation should the directives not be implemented in a timely manner. External organizations include, for example, external mission/business partners, supply chain partners, external service providers, and other peer/supporting organizations.

Related Controls: *SI-2*

SI-5a.

Receives information system security alerts, advisories, and directives from [Assignment: organization-defined external organizations] on an ongoing basis;

SI-5b.

Generates internal security alerts, advisories, and directives as deemed necessary;

SI-5c.

Disseminates security alerts, advisories, and directives to: [Selection (one or more): [Assignment: organization-defined personnel or roles]; [Assignment: organization-defined elements within the organization]; [Assignment: organization-defined external organizations]]; and

SI-5d.

Implements security directives in accordance with established time frames, or notifies the issuing organization of the degree of noncompliance.

SI-5 (1) : AUTOMATED ALERTS AND ADVISORIES

Baseline-Impact: HIGH

The organization employs automated mechanisms to make security alert and advisory information available throughout the organization.

Note: The significant number of changes to organizational information systems and the environments in which those systems operate requires the dissemination of security-related information to a variety of organizational entities that have a direct interest in the success of organizational missions and business functions. Based on the information provided by the security alerts and advisories, changes may be required at one or more of the three tiers related to the management of information security risk including the governance level, mission/business process/enterprise architecture level, and the information system level.

Control Family: SYSTEM AND INFORMATION INTEGRITY

SI-6 : SECURITY FUNCTION VERIFICATION

Priority: P1

Baseline-Impact: HIGH

The information system:

Note: Transitional states for information systems include, for example, system startup, restart, shutdown, and abort. Notifications provided by information systems include, for example, electronic alerts to system administrators, messages to local computer consoles, and/or hardware indications such as lights.

Related Controls: *CA-7*, *CM-6*

SI-6a.

Verifies the correct operation of [Assignment: organization-defined security functions];

SI-6b.

Performs this verification [Selection (one or more): [Assignment: organization-defined system transitional states]; upon command by user with appropriate privilege; [Assignment: organization-defined frequency]];

SI-6c.

Notifies [Assignment: organization-defined personnel or roles] of failed security verification tests; and

SI-6d.

[Selection (one or more): shuts the information system down; restarts the information system; [Assignment: organization-defined alternative action(s)]] when anomalies are discovered.

SI-6 (1) : NOTIFICATION OF FAILED SECURITY TESTS

[Withdrawn: Incorporated into SI-6].

SI-6 (2) : AUTOMATION SUPPORT FOR DISTRIBUTED TESTING

The information system implements automated mechanisms to support the management of distributed security testing.

Related Controls: *SI-2*

SI-6 (3) : REPORT VERIFICATION RESULTS

The organization reports the results of security function verification to [Assignment: organization-defined personnel or roles].

Note: Organizational personnel with potential interest in security function verification results include, for example, senior information security officers, information system security managers, and information systems security officers.

Related Controls: *SA-12*, *SI-4*, *SI-5*

Control Family: SYSTEM AND INFORMATION INTEGRITY

SI-7 : SOFTWARE, FIRMWARE, AND INFORMATION INTEGRITY

Priority: P1

Baseline-Impact: *MODERATE*, **HIGH**

> The organization employs integrity verification tools to detect unauthorized changes to [Assignment: organization-defined software, firmware, and information].

Note: Unauthorized changes to software, firmware, and information can occur due to errors or malicious activity (e.g., tampering). Software includes, for example, operating systems (with key internal components such as kernels, drivers), middleware, and applications. Firmware includes, for example, the Basic Input Output System (BIOS). Information includes metadata such as security attributes associated with information. State-of-the-practice integrity-checking mechanisms (e.g., parity checks, cyclical redundancy checks, cryptographic hashes) and associated tools can automatically monitor the integrity of information systems and hosted applications.

Related Controls: *SA-12*, *SC-8*, *SC-13*, *SI-3*

SI-7 (1) : INTEGRITY CHECKS

Baseline-Impact: *MODERATE*, **HIGH**

> The information system performs an integrity check of [Assignment: organization-defined software, firmware, and information] [Selection (one or more): at startup; at [Assignment: organization-defined transitional states or security-relevant events]; [Assignment: organization-defined frequency]].

Note: Security-relevant events include, for example, the identification of a new threat to which organizational information systems are susceptible, and the installation of new hardware, software, or firmware. Transitional states include, for example, system startup, restart, shutdown, and abort.

SI-7 (2) : AUTOMATED NOTIFICATIONS OF INTEGRITY VIOLATIONS

Baseline-Impact: **HIGH**

> The organization employs automated tools that provide notification to [Assignment: organization-defined personnel or roles] upon discovering discrepancies during integrity verification.

Note: The use of automated tools to report integrity violations and to notify organizational personnel in a timely matter is an essential precursor to effective risk response. Personnel having an interest in integrity violations include, for example, mission/business owners, information system owners, systems administrators, software developers, systems integrators, and information security officers.

SI-7 (3) : CENTRALLY-MANAGED INTEGRITY TOOLS

> The organization employs centrally managed integrity verification tools.

Related Controls: *AU-3*, *SI-2*, *SI-8*

SI-7 (4) : TAMPER-EVIDENT PACKAGING

> [Withdrawn: Incorporated into SA-12].

SI-7 (5) : AUTOMATED RESPONSE TO INTEGRITY VIOLATIONS

Baseline-Impact: HIGH

> The information system automatically [Selection (one or more): shuts the information system down; restarts the information system; implements [Assignment: organization-defined security safeguards]] when integrity violations are discovered.

Note: Organizations may define different integrity checking and anomaly responses: (i) by type of information (e.g., firmware, software, user data); (ii) by specific information (e.g., boot firmware, boot firmware for a specific types of machines); or (iii) a combination of both. Automatic implementation of specific safeguards within organizational information systems includes, for example, reversing the changes, halting the information system, or triggering audit alerts when unauthorized modifications to critical security files occur.

SI-7 (6) : CRYPTOGRAPHIC PROTECTION

> The information system implements cryptographic mechanisms to detect unauthorized changes to software, firmware, and information.

Note: Cryptographic mechanisms used for the protection of integrity include, for example, digital signatures and the computation and application of signed hashes using asymmetric cryptography, protecting the confidentiality of the key used to generate the hash, and using the public key to verify the hash information.

Related Controls: *SC-13*

SI-7 (7) : INTEGRATION OF DETECTION AND RESPONSE

Baseline-Impact: *MODERATE*, **HIGH**

> The organization incorporates the detection of unauthorized [Assignment: organization-defined security-relevant changes to the information system] into the organizational incident response capability.

Note: This control enhancement helps to ensure that detected events are tracked, monitored, corrected, and available for historical purposes. Maintaining historical records is important both for being able to identify and discern adversary actions over an extended period of time and for possible legal actions. Security-relevant changes include, for example, unauthorized changes to established configuration settings or unauthorized elevation of information system privileges.

Related Controls: *IR-4*, *IR-5*, *SI-4*

SI-7 (8) : AUDITING CAPABILITY FOR SIGNIFICANT EVENTS

> The information system, upon detection of a potential integrity violation, provides the capability to audit the event and initiates the following actions: [Selection (one or more): generates an audit record; alerts current user; alerts [Assignment: organization-defined personnel or roles]; [Assignment: organization-defined other actions]].

Note: Organizations select response actions based on types of software, specific software, or information for which there are potential integrity violations.

Related Controls: *AU-2*, *AU-6*, *AU-12*

SI-7 (9) : VERIFY BOOT PROCESS

> The information system verifies the integrity of the boot process of [Assignment: organization-defined devices].

Note: Ensuring the integrity of boot processes is critical to starting devices in known/trustworthy states. Integrity verification mechanisms provide organizational personnel with assurance that only trusted code is executed during boot processes.

SI-7 (10) : PROTECTION OF BOOT FIRMWARE

> The information system implements [Assignment: organization-defined security safeguards] to protect the integrity of boot firmware in [Assignment: organization-defined devices].

Note: Unauthorized modifications to boot firmware may be indicative of a sophisticated, targeted cyber attack. These types of cyber attacks can result in a permanent denial of service (e.g., if the firmware is corrupted) or a persistent malicious code presence (e.g., if code is embedded within the firmware). Devices can protect the integrity of the boot firmware in organizational information systems by: (i) verifying the integrity and authenticity of all updates to the boot firmware prior to applying changes to the boot devices; and (ii) preventing unauthorized processes from modifying the boot firmware.

SI-7 (11) : CONFINED ENVIRONMENTS WITH LIMITED PRIVILEGES

> The organization requires that [Assignment: organization-defined user-installed software] execute in a confined physical or virtual machine environment with limited privileges.

Note: Organizations identify software that may be of greater concern with regard to origin or potential for containing malicious code. For this type of software, user installations occur in confined environments of operation to limit or contain damage from malicious code that may be executed.

SI-7 (12) : INTEGRITY VERIFICATION

The organization requires that the integrity of [Assignment: organization-defined user-installed software] be verified prior to execution.

Note: Organizations verify the integrity of user-installed software prior to execution to reduce the likelihood of executing malicious code or code that contains errors from unauthorized modifications. Organizations consider the practicality of approaches to verifying software integrity including, for example, availability of checksums of adequate trustworthiness from software developers or vendors.

SI-7 (13) : CODE EXECUTION IN PROTECTED ENVIRONMENTS

The organization allows execution of binary or machine-executable code obtained from sources with limited or no warranty and without the provision of source code only in confined physical or virtual machine environments and with the explicit approval of [Assignment: organization-defined personnel or roles].

Note: This control enhancement applies to all sources of binary or machine-executable code including, for example, commercial software/firmware and open source software.

SI-7 (14) : BINARY OR MACHINE EXECUTABLE CODE

Baseline-Impact: HIGH

The organization:

Note: This control enhancement applies to all sources of binary or machine-executable code including, for example, commercial software/firmware and open source software. Organizations assess software products without accompanying source code from sources with limited or no warranty for potential security impacts. The assessments address the fact that these types of software products may be very difficult to review, repair, or extend, given that organizations, in most cases, do not have access to the original source code, and there may be no owners who could make such repairs on behalf of organizations.

Related Controls: *SA-5*

SI-7 (14)(a)

Prohibits the use of binary or machine-executable code from sources with limited or no warranty and without the provision of source code; and

SI-7 (14)(b)

Provides exceptions to the source code requirement only for compelling mission/operational requirements and with the approval of the authorizing official.

SI-7 (15) : CODE AUTHENTICATION

The information system implements cryptographic mechanisms to authenticate [Assignment: organization-defined software or firmware components] prior to installation.

Note: Cryptographic authentication includes, for example, verifying that software or firmware components have been digitally signed using certificates recognized and approved by organizations. Code signing is an effective method to protect against malicious code.

SI-7 (16) : TIME LIMIT ON PROCESS EXECUTION W/O SUPERVISION

The organization does not allow processes to execute without supervision for more than [Assignment: organization-defined time period].

Note: This control enhancement addresses processes for which normal execution periods can be determined and situations in which organizations exceed such periods. Supervision includes, for example, operating system timers, automated responses, or manual oversight and response when information system process anomalies occur.

Control Family: SYSTEM AND INFORMATION INTEGRITY

SI-8 : SPAM PROTECTION

Priority: P2

Baseline-Impact: *MODERATE*, **HIGH**

The organization:

Note: Information system entry and exit points include, for example, firewalls, electronic mail servers, web servers, proxy servers, remote-access servers, workstations, mobile devices, and notebook/laptop computers. Spam can be transported by different means including, for example, electronic mail, electronic mail attachments, and web accesses. Spam protection mechanisms include, for example, signature definitions.

Related Controls: *AT-2*, *AT-3*, *SC-5*, *SC-7*, *SI-3*

SI-8a.

Employs spam protection mechanisms at information system entry and exit points to detect and take action on unsolicited messages; and

SI-8b.

Updates spam protection mechanisms when new releases are available in accordance with organizational configuration management policy and procedures.

SI-8 (1) : CENTRAL MANAGEMENT

Baseline-Impact: *MODERATE*, **HIGH**

The organization centrally manages spam protection mechanisms.

Note: Central management is the organization-wide management and implementation of spam protection mechanisms. Central management includes planning, implementing, assessing, authorizing, and monitoring the organization-defined, centrally managed spam protection security controls.

Related Controls: *AU-3*, *SI-2*, *SI-7*

SI-8 (2) : AUTOMATIC UPDATES

Baseline-Impact: *MODERATE*, **HIGH**

The information system automatically updates spam protection mechanisms.

SI-8 (3) : CONTINUOUS LEARNING CAPABILITY

The information system implements spam protection mechanisms with a learning capability to more effectively identify legitimate communications traffic.

Note: Learning mechanisms include, for example, Bayesian filters that respond to user inputs identifying specific traffic as spam or legitimate by updating algorithm parameters and thereby more accurately separating types of traffic.

Control Family: SYSTEM AND INFORMATION INTEGRITY

SI-9 : INFORMATION INPUT RESTRICTIONS

[Withdrawn: Incorporated into AC-2, AC-3, AC-5, AC-6].

Control Family: SYSTEM AND INFORMATION INTEGRITY

SI-10 : INFORMATION INPUT VALIDATION

Priority: P1

Baseline-Impact: *MODERATE*, **HIGH**

The information system checks the validity of [Assignment: organization-defined information inputs].

Note: Checking the valid syntax and semantics of information system inputs (e.g., character set, length, numerical range, and acceptable values) verifies that inputs match specified definitions for format and content. Software applications typically follow well-defined protocols that use structured messages (i.e., commands or queries) to communicate between software modules or system components. Structured messages can contain raw or unstructured data interspersed with metadata or control information. If software applications use attacker-supplied inputs to construct structured messages without properly encoding such messages, then the attacker could insert malicious commands or special characters that can cause the data to be interpreted as control information or metadata. Consequently, the module or component that receives the tainted output will perform the wrong operations or otherwise interpret the data incorrectly. Prescreening inputs prior to passing to interpreters prevents the content from being unintentionally interpreted as commands. Input validation helps to ensure accurate and correct inputs and prevent attacks such as cross-site scripting and a variety of injection attacks.

SI-10 (1) : MANUAL OVERRIDE CAPABILITY

The information system:

Related Controls: *CM-3*, *CM-5*

SI-10 (1)(a)

Provides a manual override capability for input validation of [Assignment: organization-defined inputs];

SI-10 (1)(b)

Restricts the use of the manual override capability to only [Assignment: organization-defined authorized individuals]; and

SI-10 (1)(c)

Audits the use of the manual override capability.

SI-10 (2) : REVIEW / RESOLUTION OF ERRORS

The organization ensures that input validation errors are reviewed and resolved within [Assignment: organization-defined time period].

Note: Resolution of input validation errors includes, for example, correcting systemic causes of errors and resubmitting transactions with corrected input.

SI-10 (3) : PREDICTABLE BEHAVIOR

The information system behaves in a predictable and documented manner that reflects organizational and system objectives when invalid inputs are received.

Note: A common vulnerability in organizational information systems is unpredictable behavior when invalid inputs are received. This control enhancement ensures that there is predictable behavior in the face of invalid inputs by specifying information system responses that facilitate transitioning the system to known states without adverse, unintended side effects.

SI-10 (4) : REVIEW / TIMING INTERACTIONS

The organization accounts for timing interactions among information system components in determining appropriate responses for invalid inputs.

Note: In addressing invalid information system inputs received across protocol interfaces, timing interactions become relevant, where one protocol needs to consider the impact of the error response on other protocols within the protocol stack. For example, 802.11 standard wireless network protocols do not interact well with Transmission Control Protocols (TCP) when packets are dropped (which could be due to invalid packet input). TCP assumes packet losses are due to congestion, while packets lost over 802.11 links are typically dropped due to collisions or noise on the link. If TCP makes a congestion response, it takes precisely the wrong action in response to a collision event. Adversaries may be able to use apparently acceptable individual behaviors of the protocols in concert to achieve adverse effects through suitable construction of invalid input.

SI-10 (5) : RESTRICT INPUTS TO TRUSTED SOURCES AND APPROVED FORMATS

The organization restricts the use of information inputs to [Assignment: organization-defined trusted sources] and/or [Assignment: organization-defined formats].

Note: This control enhancement applies the concept of whitelisting to information inputs. Specifying known trusted sources for information inputs and acceptable formats for such inputs can reduce the probability of malicious activity.

Control Family: SYSTEM AND INFORMATION INTEGRITY

SI-11 : ERROR HANDLING

Priority: P2

Baseline-Impact: *MODERATE*, **HIGH**

> The information system:

Note: Organizations carefully consider the structure/content of error messages. The extent to which information systems are able to identify and handle error conditions is guided by organizational policy and operational requirements. Information that could be exploited by adversaries includes, for example, erroneous logon attempts with passwords entered by mistake as the username, mission/business information that can be derived from (if not stated explicitly by) information recorded, and personal information such as account numbers, social security numbers, and credit card numbers. In addition, error messages may provide a covert channel for transmitting information.

Related Controls: *AU-2*, *AU-3*, *SC-31*

SI-11a.

> Generates error messages that provide information necessary for corrective actions without revealing information that could be exploited by adversaries; and

SI-11b.

> Reveals error messages only to [Assignment: organization-defined personnel or roles].

Control Family: SYSTEM AND INFORMATION INTEGRITY

SI-12 : INFORMATION HANDLING AND RETENTION

Priority: P2

Baseline-Impact: LOW, *MODERATE*, **HIGH**

> The organization handles and retains information within the information system and information output from the system in accordance with applicable federal laws, Executive Orders, directives, policies, regulations, standards, and operational requirements.

Note: Information handling and retention requirements cover the full life cycle of information, in some cases extending beyond the disposal of information systems. The National Archives and Records Administration provides guidance on records retention.

Related Controls: *AC-16*, *AU-5*, *AU-11*, *MP-2*, *MP-4*

Control Family: SYSTEM AND INFORMATION INTEGRITY

SI-13 : PREDICTABLE FAILURE PREVENTION

Priority: P0

The organization:

Note: While MTTF is primarily a reliability issue, this control addresses potential failures of specific information system components that provide security capability. Failure rates reflect installation-specific consideration, not industry-average. Organizations define criteria for substitution of information system components based on MTTF value with consideration for resulting potential harm from component failures. Transfer of responsibilities between active and standby components does not compromise safety, operational readiness, or security capability (e.g., preservation of state variables). Standby components remain available at all times except for maintenance issues or recovery failures in progress.

Related Controls: *CP-2*, *CP-10*, *MA-6*

SI-13a.

Determines mean time to failure (MTTF) for [Assignment: organization-defined information system components] in specific environments of operation; and

SI-13b.

Provides substitute information system components and a means to exchange active and standby components at [Assignment: organization-defined MTTF substitution criteria].

SI-13 (1) : TRANSFERRING COMPONENT RESPONSIBILITIES

The organization takes information system components out of service by transferring component responsibilities to substitute components no later than [Assignment: organization-defined fraction or percentage] of mean time to failure.

SI-13 (2) : TIME LIMIT ON PROCESS EXECUTION WITHOUT SUPERVISION

[Withdrawn: Incorporated into SI-7 (16)].

SI-13 (3) : MANUAL TRANSFER BETWEEN COMPONENTS

The organization manually initiates transfers between active and standby information system components [Assignment: organization-defined frequency] if the mean time to failure exceeds [Assignment: organization-defined time period].

SI-13 (4) : STANDBY COMPONENT INSTALLATION / NOTIFICATION

The organization, if information system component failures are detected:

Note: Automatic or manual transfer of components from standby to active mode can occur, for example, upon detection of component failures.

SI-13 (4)(a)

Ensures that the standby components are successfully and transparently installed within [Assignment: organization-defined time period]; and

SI-13 (4)(b)

[Selection (one or more): activates [Assignment: organization-defined alarm]; automatically shuts down the information system].

SI-13 (5) : FAILOVER CAPABILITY

The organization provides [Selection: real-time; near real-time] [Assignment: organization-defined failover capability] for the information system.

Note: Failover refers to the automatic switchover to an alternate information system upon the failure of the primary information system. Failover capability includes, for example, incorporating mirrored information system operations at alternate processing sites or periodic data mirroring at regular intervals defined by recovery time periods of organizations.

Control Family: SYSTEM AND INFORMATION INTEGRITY

SI-14 : NON-PERSISTENCE

Priority: P0

The organization implements non-persistent [Assignment: organization-defined information system components and services] that are initiated in a known state and terminated [Selection (one or more): upon end of session of use; periodically at [Assignment: organization-defined frequency]].

Note:

This control mitigates risk from advanced persistent threats (APTs) by significantly reducing the targeting capability of adversaries (i.e., window of opportunity and available attack surface) to initiate and complete cyber attacks. By implementing the concept of non-persistence for selected information system components, organizations can provide a known state computing resource for a specific period of time that does not give adversaries sufficient time on target to exploit vulnerabilities in organizational information systems and the environments in which those systems operate. Since the advanced persistent threat is a high-end threat with regard to capability, intent, and targeting, organizations assume that over an extended period of time, a percentage of cyber attacks will be successful. Non-persistent information system components and services are activated as required using protected information and terminated periodically or upon the end of sessions. Non-persistence increases the work factor of adversaries in attempting to compromise or breach organizational information systems.

> Non-persistent system components can be implemented, for example, by periodically re-imaging components or by using a variety of common virtualization techniques. Non-persistent services can be implemented using virtualization techniques as part of virtual machines or as new instances of processes on physical machines (either persistent or non-persistent).The benefit of periodic refreshes of information system components/services is that it does not require organizations to first determine whether compromises of components or services have occurred (something that may often be difficult for organizations to determine). The refresh of selected information system components and services occurs with sufficient frequency to prevent the spread or intended impact of attacks, but not with such frequency that it makes the information system unstable. In some instances, refreshes of critical components and services may be done periodically in order to hinder the ability of adversaries to exploit optimum windows of vulnerabilities.

Related Controls: *SC-30*, *SC-34*

SI-14 (1) : REFRESH FROM TRUSTED SOURCES

> The organization ensures that software and data employed during information system component and service refreshes are obtained from [Assignment: organization-defined trusted sources].

Note: Trusted sources include, for example, software/data from write-once, read-only media or from selected off-line secure storage facilities.

Control Family: SYSTEM AND INFORMATION INTEGRITY

SI-15 : INFORMATION OUTPUT FILTERING

Priority: P0

> The information system validates information output from [Assignment: organization-defined software programs and/or applications] to ensure that the information is consistent with the expected content.

Note: Certain types of cyber attacks (e.g., SQL injections) produce output results that are unexpected or inconsistent with the output results that would normally be expected from software programs or applications. This control enhancement focuses on detecting extraneous content, preventing such extraneous content from being displayed, and alerting monitoring tools that anomalous behavior has been discovered.

Related Controls: *SI-3*, *SI-4*

Control Family: SYSTEM AND INFORMATION INTEGRITY

SI-16 : MEMORY PROTECTION

Priority: P1

Baseline-Impact: *MODERATE*, **HIGH**

> The information system implements [Assignment: organization-defined security safeguards] to protect its memory from unauthorized code execution.

Note: Some adversaries launch attacks with the intent of executing code in non-executable regions of memory or in memory locations that are prohibited. Security safeguards employed to protect memory include, for example, data execution prevention and address space layout randomization. Data execution prevention safeguards can either be hardware-enforced or software-enforced with hardware providing the greater strength of mechanism.

Related Controls: *AC-25*, *SC-3*

Control Family: SYSTEM AND INFORMATION INTEGRITY

SI-17 : FAIL-SAFE PROCEDURES

Priority: P0

> The information system implements [Assignment: organization-defined fail-safe procedures] when [Assignment: organization-defined failure conditions occur].

Note: Failure conditions include, for example, loss of communications among critical system components or between system components and operational facilities. Fail-safe procedures include, for example, alerting operator personnel and providing specific instructions on subsequent steps to take (e.g., do nothing, reestablish system settings, shut down processes, restart the system, or contact designated organizational personnel).

Related Controls: *CP-12*, *CP-13*, *SC-24*, *SI-13*

Control Family: PROGRAM MANAGEMENT

PM-1 : INFORMATION SECURITY PROGRAM PLAN

> The organization:

Note:

Information security program plans can be represented in single documents or compilations of documents at the discretion of organizations. The plans document the program management controls and organization-defined common controls. Information security program plans provide sufficient information about the program management controls/common controls (including specification of parameters for any assignment and selection statements either explicitly or by reference) to enable implementations that are unambiguously compliant with the intent of the plans and a determination of the risk to be incurred if the plans are implemented as intended.

The security plans for individual information systems and the organization-wide information security program plan together, provide complete coverage for all security controls employed within the organization. Common controls are documented in an appendix to the organizations information security program plan unless the controls are included in a separate security plan for an information system (e.g., security controls employed as part of an intrusion detection system providing organization-wide boundary protection inherited by one or more organizational information systems). The organization-wide information security program plan will indicate which separate security plans contain descriptions of common controls. Organizations have the flexibility to describe common controls in a single document or in multiple documents. In the case of multiple documents, the documents describing common controls are included as attachments to the information security program plan. If the information security program plan contains multiple documents, the organization specifies in each document the organizational official or officials responsible for the development, implementation, assessment, authorization, and monitoring of the respective common controls. For example, the organization may require that the Facilities Management Office develop, implement, assess, authorize, and continuously monitor common physical and environmental protection controls from the PE family when such controls are not associated with a particular information system but instead, support multiple information systems.

Related Controls: *PM-8*

PM-1a.

Develops and disseminates an organization-wide information security program plan that:

PM-1a.1.

Provides an overview of the requirements for the security program and a description of the security program management controls and common controls in place or planned for meeting those requirements;

PM-1a.2.

Includes the identification and assignment of roles, responsibilities, management commitment, coordination among organizational entities, and compliance;

PM-1a.3.

Reflects coordination among organizational entities responsible for the different aspects of information security (i.e., technical, physical, personnel, cyber-physical); and

PM-1a.4.

Is approved by a senior official with responsibility and accountability for the risk being incurred to organizational operations (including mission, functions, image, and reputation), organizational assets, individuals, other organizations, and the Nation;

PM-1b.

Reviews the organization-wide information security program plan [Assignment: organization-defined frequency];

PM-1c.

Updates the plan to address organizational changes and problems identified during plan implementation or security control assessments; and

PM-1d.

Protects the information security program plan from unauthorized disclosure and modification.

Control Family: PROGRAM MANAGEMENT

PM-2 : SENIOR INFORMATION SECURITY OFFICER

The organization appoints a senior information security officer with the mission and resources to coordinate, develop, implement, and maintain an organization-wide information security program.

Note: The security officer described in this control is an organizational official. For a federal agency (as defined in applicable federal laws, Executive Orders, directives, policies, or regulations) this official is the Senior Agency Information Security Officer. Organizations may also refer to this official as the Senior Information Security Officer or Chief Information Security Officer.

Control Family: PROGRAM MANAGEMENT

PM-3 : INFORMATION SECURITY RESOURCES

The organization:

Note: Organizations consider establishing champions for information security efforts and as part of including the necessary resources, assign specialized expertise and resources as needed. Organizations may designate and empower an Investment Review Board (or similar group) to manage and provide oversight for the information security-related aspects of the capital planning and investment control process.

Related Controls: *PM-4*, *SA-2*

PM-3a.

Ensures that all capital planning and investment requests include the resources needed to implement the information security program and documents all exceptions to this requirement;

PM-3b.

Employs a business case/Exhibit 300/Exhibit 53 to record the resources required; and

PM-3c.

Ensures that information security resources are available for expenditure as planned.

Control Family: PROGRAM MANAGEMENT

PM-4 : PLAN OF ACTION AND MILESTONES PROCESS

The organization:

Note: The plan of action and milestones is a key document in the information security program and is subject to federal reporting requirements established by OMB. With the increasing emphasis on organization-wide risk management across all three tiers in the risk management hierarchy (i.e., organization, mission/business process, and information system), organizations view plans of action and milestones from an organizational perspective, prioritizing risk response actions and ensuring consistency with the goals and objectives of the organization. Plan of action and milestones updates are based on findings from security control assessments and continuous monitoring activities. OMB FISMA reporting guidance contains instructions regarding organizational plans of action and milestones.

Related Controls: *CA-5*

PM-4a.

Implements a process for ensuring that plans of action and milestones for the security program and associated organizational information systems:

PM-4a.1.

Are developed and maintained;

PM-4a.2.

Document the remedial information security actions to adequately respond to risk to organizational operations and assets, individuals, other organizations, and the Nation; and

PM-4a.3.

Are reported in accordance with OMB FISMA reporting requirements.

PM-4b.

Reviews plans of action and milestones for consistency with the organizational risk management strategy and organization-wide priorities for risk response actions.

Control Family: PROGRAM MANAGEMENT

PM-5 : INFORMATION SYSTEM INVENTORY

The organization develops and maintains an inventory of its information systems.

Note: This control addresses the inventory requirements in FISMA. OMB provides guidance on developing information systems inventories and associated reporting requirements. For specific information system inventory reporting requirements, organizations consult OMB annual FISMA reporting guidance.

Control Family: PROGRAM MANAGEMENT

PM-6 : INFORMATION SECURITY MEASURES OF PERFORMANCE

The organization develops, monitors, and reports on the results of information security measures of performance.

Note: Measures of performance are outcome-based metrics used by an organization to measure the effectiveness or efficiency of the information security program and the security controls employed in support of the program.

Control Family: PROGRAM MANAGEMENT

PM-7 : ENTERPRISE ARCHITECTURE

The organization develops an enterprise architecture with consideration for information security and the resulting risk to organizational operations, organizational assets, individuals, other organizations, and the Nation.

Note: The enterprise architecture developed by the organization is aligned with the Federal Enterprise Architecture. The integration of information security requirements and associated security controls into the organizations enterprise architecture helps to ensure that security considerations are addressed by organizations early in the system development life cycle and are directly and explicitly related to the organizations mission/business processes. This process of security requirements integration also embeds into the enterprise architecture, an integral information security architecture consistent with organizational risk management and information security strategies. For PM-7, the information security architecture is developed at a system-of-systems level (organization-wide), representing all of the organizational information systems. For PL-8, the information security architecture is developed at a level representing an individual information system but at the same time, is consistent with the information security architecture defined for the organization. Security requirements and security control integration are most effectively accomplished through the application of the Risk Management Framework and supporting security standards and guidelines. The Federal Segment Architecture Methodology provides guidance on integrating information security requirements and security controls into enterprise architectures.

Related Controls: *PL-2*, *PL-8*, *PM-11*, *RA-2*, *SA-3*

Control Family: PROGRAM MANAGEMENT

PM-8 : CRITICAL INFRASTRUCTURE PLAN

> The organization addresses information security issues in the development, documentation, and updating of a critical infrastructure and key resources protection plan.

Note: Protection strategies are based on the prioritization of critical assets and resources. The requirement and guidance for defining critical infrastructure and key resources and for preparing an associated critical infrastructure protection plan are found in applicable federal laws, Executive Orders, directives, policies, regulations, standards, and guidance.

Related Controls: *PM-1*, *PM-9*, *PM-11*, *RA-3*

Control Family: PROGRAM MANAGEMENT

PM-9 : RISK MANAGEMENT STRATEGY

> The organization:

Note: An organization-wide risk management strategy includes, for example, an unambiguous expression of the risk tolerance for the organization, acceptable risk assessment methodologies, risk mitigation strategies, a process for consistently evaluating risk across the organization with respect to the organizations risk tolerance, and approaches for monitoring risk over time. The use of a risk executive function can facilitate consistent, organization-wide application of the risk management strategy. The organization-wide risk management strategy can be informed by risk-related inputs from other sources both internal and external to the organization to ensure the strategy is both broad-based and comprehensive.

Related Controls: *RA-3*

PM-9a.

Develops a comprehensive strategy to manage risk to organizational operations and assets, individuals, other organizations, and the Nation associated with the operation and use of information systems;

PM-9b.

Implements the risk management strategy consistently across the organization; and

PM-9c.

Reviews and updates the risk management strategy [Assignment: organization-defined frequency] or as required, to address organizational changes.

Control Family: PROGRAM MANAGEMENT

PM-10 : SECURITY AUTHORIZATION PROCESS

The organization:

Note: Security authorization processes for information systems and environments of operation require the implementation of an organization-wide risk management process, a Risk Management Framework, and associated security standards and guidelines. Specific roles within the risk management process include an organizational risk executive (function) and designated authorizing officials for each organizational information system and common control provider. Security authorization processes are integrated with organizational continuous monitoring processes to facilitate ongoing understanding and acceptance of risk to organizational operations and assets, individuals, other organizations, and the Nation.

Related Controls: *CA-6*

PM-10a.

Manages (i.e., documents, tracks, and reports) the security state of organizational information systems and the environments in which those systems operate through security authorization processes;

PM-10b.

Designates individuals to fulfill specific roles and responsibilities within the organizational risk management process; and

PM-10c.

Fully integrates the security authorization processes into an organization-wide risk management program.

Control Family: PROGRAM MANAGEMENT

PM-11 : MISSION/BUSINESS PROCESS DEFINITION

The organization:

Note: Information protection needs are technology-independent, required capabilities to counter threats to organizations, individuals, or the Nation through the compromise of information (i.e., loss of confidentiality, integrity, or availability). Information protection needs are derived from the mission/business needs defined by the organization, the mission/business processes selected to meet the stated needs, and the organizational risk management strategy. Information protection needs determine the required security controls for the organization and the associated information systems supporting the mission/business processes. Inherent in defining an organizations information protection needs is an understanding of the level of adverse impact that could result if a compromise of information occurs. The security categorization process is used to make such potential impact determinations. Mission/business process definitions and associated information protection requirements are documented by the organization in accordance with organizational policy and procedure.

Related Controls: *PM-7*, *PM-8*, *RA-2*

PM-11a.

Defines mission/business processes with consideration for information security and the resulting risk to organizational operations, organizational assets, individuals, other organizations, and the Nation; and

PM-11b.

Determines information protection needs arising from the defined mission/business processes and revises the processes as necessary, until achievable protection needs are obtained.

Control Family: PROGRAM MANAGEMENT

PM-12 : INSIDER THREAT PROGRAM

The organization implements an insider threat program that includes a cross-discipline insider threat incident handling team.

Note:

Organizations handling classified information are required, under Executive Order 13587 and the National Policy on Insider Threat, to establish insider threat programs. The standards and guidelines that apply to insider threat programs in classified environments can also be employed effectively to improve the security of Controlled Unclassified Information in non-national security systems. Insider threat programs include security controls to detect and prevent malicious insider activity through the centralized integration and analysis of both technical and non-technical information to identify potential insider threat concerns. A senior organizational official is designated by the department/agency head as the responsible individual to implement and provide oversight for the program. In addition to the centralized integration and analysis capability, insider threat programs as a minimum, prepare department/agency insider threat policies and implementation plans, conduct host-based user monitoring of individual employee activities on government-owned classified computers, provide insider threat awareness training to employees, receive access to information from all offices within the department/agency (e.g., human resources, legal, physical security, personnel security, information technology, information system security, and law enforcement) for insider threat analysis, and conduct self-assessments of department/agency insider threat posture.

Insider threat programs can leverage the existence of incident handling teams organizations may already have in place, such as computer security incident response teams. Human resources records are especially important in this effort, as there is compelling evidence to show that some types of insider crimes are often preceded by nontechnical behaviors in the workplace (e.g., ongoing patterns of disgruntled behavior and conflicts with coworkers and other colleagues). These precursors can better inform and guide organizational officials in more focused, targeted monitoring efforts. The participation of a legal team is important to ensure that all monitoring activities are performed in accordance with appropriate legislation, directives, regulations, policies, standards, and guidelines.

Related Controls: *AC-6*, *AT-2*, *AU-6*, *AU-7*, *AU-10*, *AU-12*, *AU-13*, *CA-7*, *IA-4*, *IR-4*, *MP-7*, *PE-2*, *PS-3*, *PS-4*, *PS-5*, *PS-8*, *SC-7*, *SC-38*, *SI-4*, *PM-1*, *PM-14*

Control Family: PROGRAM MANAGEMENT

PM-13 : INFORMATION SECURITY WORKFORCE

The organization establishes an information security workforce development and improvement program.

Note: Information security workforce development and improvement programs include, for example: (i) defining the knowledge and skill levels needed to perform information security duties and tasks; (ii) developing role-based training programs for individuals assigned information security roles and responsibilities; and (iii) providing standards for measuring and building individual qualifications for incumbents and applicants for information security-related positions. Such workforce programs can also include associated information security career paths to encourage: (i) information security professionals to advance in the field and fill positions with greater responsibility; and (ii) organizations to fill information security-related positions with qualified personnel. Information security workforce development and improvement programs are complementary to organizational security awareness and training programs. Information security workforce development and improvement programs focus on developing and institutionalizing core information security capabilities of selected personnel needed to protect organizational operations, assets, and individuals.

Related Controls: *AT-2*, *AT-3*

Control Family: PROGRAM MANAGEMENT

PM-14 : TESTING, TRAINING, AND MONITORING

The organization:

Note: This control ensures that organizations provide oversight for the security testing, training, and monitoring activities conducted organization-wide and that those activities are coordinated. With the importance of continuous monitoring programs, the implementation of information security across the three tiers of the risk management hierarchy, and the widespread use of common controls, organizations coordinate and consolidate the testing and monitoring activities that are routinely conducted as part of ongoing organizational assessments supporting a variety of security controls. Security training activities, while typically focused on individual information systems and specific roles, also necessitate coordination across all organizational elements. Testing, training, and monitoring plans and activities are informed by current threat and vulnerability assessments.

Related Controls: *AT-3*, *CA-7*, *CP-4*, *IR-3*, *SI-4*

PM-14a.

Implements a process for ensuring that organizational plans for conducting security testing, training, and monitoring activities associated with organizational information systems:

PM-14a.1.

Are developed and maintained; and

PM-14a.2.

Continue to be executed in a timely manner;

PM-14b.

Reviews testing, training, and monitoring plans for consistency with the organizational risk management strategy and organization-wide priorities for risk response actions.

Control Family: PROGRAM MANAGEMENT

PM-15 : CONTACTS WITH SECURITY GROUPS AND ASSOCIATIONS

The organization establishes and institutionalizes contact with selected groups and associations within the security community:

Note: Ongoing contact with security groups and associations is of paramount importance in an environment of rapidly changing technologies and threats. Security groups and associations include, for example, special interest groups, forums, professional associations, news groups, and/or peer groups of security professionals in similar organizations. Organizations select groups and associations based on organizational missions/business functions. Organizations share threat, vulnerability, and incident information consistent with applicable federal laws, Executive Orders, directives, policies, regulations, standards, and guidance.

Related Controls: *SI-5*

PM-15a.

To facilitate ongoing security education and training for organizational personnel;

PM-15b.

To maintain currency with recommended security practices, techniques, and technologies; and

PM-15c.

To share current security-related information including threats, vulnerabilities, and incidents.

Control Family: PROGRAM MANAGEMENT

PM-16 : THREAT AWARENESS PROGRAM

The organization implements a threat awareness program that includes a cross-organization information-sharing capability.

Note: Because of the constantly changing and increasing sophistication of adversaries, especially the advanced persistent threat (APT), it is becoming more likely that adversaries may successfully breach or compromise organizational information systems. One of the best techniques to address this concern is for organizations to share threat information. This can include, for example, sharing threat events (i.e., tactics, techniques, and procedures) that organizations have experienced, mitigations that organizations have found are effective against certain types of threats, threat intelligence (i.e., indications and warnings about threats that are likely to occur). Threat information sharing may be bilateral (e.g., government-commercial cooperatives, government-government cooperatives), or multilateral (e.g., organizations taking part in threat-sharing consortia). Threat information may be highly sensitive requiring special agreements and protection, or less sensitive and freely shared.

Related Controls: *PM-12*, *PM-16*

4.10 Vulnerability Supplement

This section provides supplementary guidance regarding specific actions or information relevant to vulnerability reactions on SIMP systems.

Warning: This should in **no way** take the place of vendor recommendations. Users should, in all cases, refer to the vendor documentation for the latest information and any official mitigations.

Contents:

4.10.1 Stack Clash

Official Information

- Red Hat Stack Guard Page
- CVE-2017-1000364
- CVE-2017-1000366
- Qualys Blog Post

Supplementary Guidance

In accordance with the Qualys Blog Post, we attempted to determine if there was some method of partially mitigating the attack without needing to reboot a given system.

Unfortunately, as noted in the post, our results were found to be **very** dangerous for production systems so we have added the appropriate information here in case, for some reason, you cannot reboot your systems.

The information tested below was tested to allow a graphical desktop to run and will need to be set much lower to mitigate against a large number of attacks. But, it may be better than nothing.

Warning: Setting stack and address limits in PAM affects ALL processes on the system. It is *very* important that you do not limit space needed by critical applications.

Note: Setting PAM limits will NOT guarantee immunity from Stack Clash exploits, but will reduce the chance of large-footprint attacks. It is *highly* recommended you upgrade your kernel or obtain a vendor patch to protect your machines from attack. Setting limits in PAM is a last ditch effort when upgrading is not feasible.

Before applying limits to your system, estimate the amount of stack and address space used by critical applications. We have created a space estimation script to assist with limit calculation.

We have run the script on a SIMP system with many of the component modules installed, and have created the following class with defaults based off of the results. You may find you need to adjust limits or supplement the class with additional limits.

```
# Mitigate susceptibility to the 'stack clash' exploit by limiting
# the stack size and address size for local and remote users.
#
# These limitations do NOT guarantee immunity from the exploit, but
# reduce the chance of large-footprint attacks.
#
# NOTE: Before applying this to your system, you should estimate
# the amount of stack and address space used by authorized
# applications.  You may find you need to adjust limits or augment
# this list to best suit your system.
#
# A tool for calculating the largest consumers of stack and address
# space can be found here:
#   https://gist.github.com/trevor-vaughan/8f28ac8d908b3379fa9cee97b910ac54
#
# @param ignore_list
```

(continues on next page)

(continued from previous page)

```
#   Any parameter in this list will be given unlimited stack and
#   address space.
#
#   Default: Ignore root, dbus, gdm. Root and dbus serve critical
#   roles and are unlimited for obvious reasons. GDM is a stack
#   heavy application that must be unlimited, or users run the
#   risk of loosing GUI access to their system.
#
# @param stack_limit
#   The max stack size, in KB, that applications not in the
#   ignore_list will be limited to.
#
# @param address_limit
#   The max address size, in KB, that applications not in the
#   ignore_list will be limited to.
#
# @author SIMP Team
#
class simp::pam_limits::stack_clash(
  Array[String] $ignore_list   = ['root','dbus','gdm'],
  Integer       $stack_limit   = 262144,
  Integer       $address_limit = 4194304
){

  pam::limits::rule { 'ignore_stack':
    domains => $ignore_list,
    type    => '-',
    item    => 'stack',
    value   => 'unlimited',
    order   =>   1
  }

  pam::limits::rule { 'ignore_as':
    domains => $ignore_list,
    type    => '-',
    item    => 'as',
    value   => 'unlimited',
    order   =>   1
  }

  pam::limits::rule { 'limit_stack':
    domains => ['*'],
    type    => '-',
    item    => 'stack',
    value   => $stack_limit,
    order   =>  999
  }

  pam::limits::rule { 'limit_as':
    domains => ['*'],
    type    => '-',
    item    => 'as',
```

(continues on next page)

(continued from previous page)

```
    value   => $address_limit,
    order   =>  999
  }

}
```

4.11 Help

The SIMP team is here to help!

Please see the following for a list of resources.

4.11.1 Public Resources

Many resources are available for getting help with SIMP. For FOSS support, as the community can handle it, you are welcome to use one of the following resources.

Live Chat

- SIMP Project Chat
  - The SIMP Project wants to ensure that active communication is available to all SIMP users. Please join us to discuss SIMP use and development.

Mailing Lists

- SIMP Users Mailing List
  - General user discussion
- SIMP Announcement List
  - Announcements about changes to the SIMP environment
- SIMP Security List (security@simp-project.com)
  - A post-only alias for alerting the SIMP team to security issues

Bug Tracking

If you find a bug, we'd like to encourage you to file a bug in the associated project in the SIMP GitHub space.

If you do not know what project to choose, just drop it in simp-core and we'll figure it out!

If you do not use GitHub, feel free to email your issues or use Slack via the links in the previous section.

4.11.2 Commercial Resources

Full commercial support and consulting services are available for SIMP and the underlying components!

Details and contact information are available on the SIMP Project Homepage.

4.12 License

4.12.1 Legal Notice

Per Section 105 of the Copyright Act of 1976, these works are not entitled to domestic copyright protection under US Federal law. The US Government retains the right to pursue copyright protections outside of the United States. The United States Government has unlimited rights in this documentation and all derivatives thereof, pursuant to the contracts under which it was developed and the License under which it falls.

Material submitted by entities outside the United States Government may pursue copyright enforcement on those portions to which they hold copyright. These portions are explicitly marked within the source of this documentation.

This material may only be distributed subject to the terms and conditions set forth in the Apache License, Version 2.0 (the latest version is available at the Apache License website.

The SIMP Development Team makes no representation about the suitability of the SIMP product for any purpose. It is provided "as is" without expressed or implied warranty. If SIMP is modified in any way, except for designed customization, please identify the new copy as a variant of SIMP.

Additional products are distributed as part of the SIMP suite. By using SIMP, the user agrees to abide by the licenses for the included products.

4.13 Contact

If you have questions please contact the SIMP team. simp@simp-project.org

4.14 Glossary of Terms

Note: Many terms here have been reproduced from various locations across the Internet and are governed by the licenses surrounding the source material. Please see the reference links for specifics on usage and reproducibility.

389 DS
389-DS
389 Directory Server

An enterprise-class *FOSS LDAP* server for Linux, capable of providing a full multi-master LDAPv3 infrastructure.

389 DS provides the main data store and directory service components for *FreeIPA* and IdM. *Red Hat Directory Server* is the commercially-supported version of 389 DS.

In SIMP, 389 DS replaces the (deprecated) *OpenLDAP* server as the default LDAP service for EL8+.

Website: https://directory.fedoraproject.org

ACL
Access Control List

A list of permissions attached to an object. An ACL specifies which users or system processes are granted access to objects, as well as what operations are allowed on given objects. Each entry in a typical ACL specifies a subject and an operation.

See: `man 5 acl`

AIDE
Advanced Intrusion Detection Environment

An intrusion detection system for checking the integrity of files under Linux. AIDE can be used to help track file integrity by comparing a snapshot of the system's files prior to and after a suspected incident. It is maintained by Rami Lehti and Pablo Virolainen.

Apache 2.0 License
Apache License

A software license provided by the Apache Software Foundation.

See: Apache License, Version 2.0

APL
Automatic Parameter Lookup

A method where Puppet automatically looks up *class* parameters in *Hiera* using the fully-qualified name of the parameter.

See: Looking up data with Hiera

Auditd

The userspace component to the Linux Auditing System. It is responsible for writing audit records to the disk. Viewing the logs is done with the ausearch or aureport utilities. Configuring the audit rules is done with the auditctl utility. During startup, the rules in /etc/audit/audit.rules are read by auditctl. The audit daemon itself has some configuration options that the admin may wish to customize. They are found in the auditd.conf file.

Beaker

A Ruby-based acceptance testing harness, created by the Puppet team and maintained by Vox Pupuli.

Source: Beaker Source Repository:

BIND

A flexible, full-featurend DNS system.

Source: BIND 9 Homepage:

BIOS
Basic Input/Output System

A type of firmware used to perform hardware initialization during the booting process (power-on startup) on IBM PC compatible computers.

Source: Wikipedia: BIOS

Bolt
Puppet Bolt

An agentless open-source automation and orchestration tool that can execute any scripts or commands as ad hoc tasks, coordinating them with logic in Puppet *DSL* plans. Bolt's plugins, plans, and tasks are designed to be distributed in Puppet modules.

Source: Puppetlabs Bolt

CA
Certificate Authority

An entity that issues *X.509* digital certificates.

CentOS
Community Enterprise Operating System

An Enterprise-grade Operating System that is mostly compatible with a prominent Linux distribution.

Homepage: https://centos.org/

Class
Classes
Puppet Class
Puppet Classes

Classes are named blocks of Puppet code that are stored in modules and applied later when they are invoked by name.

Source: Classes

Class Parameter

Parameters allow a *class* to request external data. If a class needs to configure itself with data other than facts, that data should usually enter the class via a parameter.

Each class parameter can be used as a normal variable inside the class definition. The values of these variables are not set with normal assignment statements or looked up from top or node scope; instead, they are set based on user input when the class is declared.

Note that if a class parameter lacks a default value, the module's user must set a value themselves (either in their external data or an override). As such, you should supply defaults wherever possible.

Each parameter can be preceded by an optional data type. If you include one, Puppet will check the parameter's value at runtime to make sure that it has the right data type, and raise an error if the value is illegal. If no data type is provided, the parameter will accept values of any data type.

The special variables `$title` and `$name` are both set to the class name automatically, so they can't be used as parameters.

Example:

```
class foo (
  # '$bar' is the class parameter and can be references as '$foo::bar'
  # from locations outside of the class and simply '$bar' from inside
  # the class.

  String $bar = 'An Example Parameter'
) { }
```

Source: Class parameters and variables

CLI
Command Line Interface

A means of interacting with a computer program where the user (or client) issues commands to the program in the form of successive lines of text (command lines).

Source: Wikipedia: Command Line Interface

Code Manager

[Puppet] Code Manager automates the management and deployment of your *Puppet* code. Push code updates to your source control repo, and then Puppet syncs the code to your servers, so that all your servers start running the new code at the same time, without interrupting agent runs.

Source: Managing code with Code Manager See Also: *r10k*

Compliance Profile
SIMP Compliance Profile

A collection of data that maps policy directly to Puppet *class* and *defined type* parameters. These profiles are used by the *SIMP Compliance Engine*.

Control Repo
Control Repository

A version control repository containing all of the required modules, data, and configuration for one or more Puppet environments. Each branch of the control repository deploys as a separate Puppet environment using *r10k* or *Code Manager*.

See:

- Puppet's documentation at https://docs.puppet.com/pe/latest/cmgmt_control_repo.html
- *HOWTO Set up a SIMP Environment in a Control Repository*
- *Setting up a SIMP Environment in a Control Repository using Remote Git repositories*

Core Dump

A dump of the contents of main memory.

CPU
Central Processing Unit

A central processing unit (CPU) is the electronic circuitry within a computer that carries out the instructions of a computer program by performing the basic arithmetic, logical, control and input/output (I/O) operations specified by the instructions

Source: Wikipedia: Central Processing Unit

CRL
Certificate Revocation List

A list of digitical certificates that have been revoked by the *CA*.

Source: Wikipedia: Certificate revocation list

DAC
Discretionary Access Control

A type of access control defined by the Trusted Computer System Evaluation Criteria "as a means of restricting access to objects based on the identity of subjects and/or groups to which they belong. The controls are discretionary in the sense that a subject with a certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject (unless restrained by mandatory access control)".

Source: Wikipedia: Discretionary access control

Defined Type
Defined Types
Defined Resource Type
Defined Resource Types
Puppet Defined Type
Puppet Defined Types

Defined resource types, sometimes called defined types or defines, are blocks of Puppet code that can be evaluated multiple times with different parameters.

Source: Defined resource types

DevOps

A set of software development practices that combines software development (Dev) and information technology operations (Ops) to shorten the systems development life cycle while delivering features, fixes, and updates frequently in close alignment with business objectives

Source: Wikipedia: DevOps

DHCP
Dynamic Host Configuration Protocol
A network protocol that enables a server to automatically assign an IP address to a computer.

DNS
Domain Name System
A database system that translates a computer's fully qualified domain name into an IP address and the reverse.

Docker
Docker containers wrap a piece of software in a complete filesystem that contains everything needed to run: code, runtime, system tools, system libraries – anything that can be installed on a server. This guarantees that the software will always run the same, regardless of its environment.

Source: Docker: What is Docker?

DoS
Denial of Service
Denial of Service Attack
An attempt to make a machine or network resource unavailable to its intended users, such as to temporarily or indefinitely interrupt or suspend services of a host connected to the Internet.

Source: Wikipedia: Denial-of-service attack

DSL
Domain Specific Language
A computer language specialized to a particular application domain.

Source: Wikipedia: Domain-specific language

EL
Enterprise Linux
In the context of SIMP, EL is a generic term for *Enterprise Linux* and covers both *RHEL* and *CentOS* as well as other *RHEL* derivatives such as Oracle Linux.

Elasticsearch
A distributed, RESTful search and analytics engine capable of solving a growing number of use cases. As the heart of the Elastic Stack, it centrally stores your data so you can discover the expected and uncover the unexpected.

Source: Elasticsearch Homepage

ELG
An acronym for *Elasticsearch*, *Logstash*, and *Grafana*

ENC
External Node Classifier
An arbitrary script or application which can tell *Puppet* which *classes* a node should have. It can replace or work in concert with the node definitions in the main site manifest (site.pp).

The Puppet Enterprise Console and The Foreman are two examples of External Node Classifiers.

Source: External Node Classifiers

EPEL
Extra Packages for Enterprise Linux
A Fedora Special Interest Group that creates, maintains, and manages a high quality set of additional packages for *Enterprise Linux*, including, but not limited to, Red Hat Enterprise Linux (*RHEL*), *CentOS* and Scientific Linux (SL), Oracle Linux (OL).E

EPEL packages are usually based on their Fedora counterparts and will never conflict with or replace packages in the base Enterprise Linux distributions. EPEL uses much of the same infrastructure as Fedora, including buildsystem, bugzilla instance, updates manager, mirror manager and more.

Source: EPEL Homepage

FACL
File Access Control List

An *ACL* applied to a file on the filesystem.

Facter

Cross-platform system profiling library for use with *Puppet* and other management tools. It discovers and reports per-node facts, which are available in your Puppet manifests as variables.

Source: Facter Documentation

FIPS
Federal Information Processing Standard

Federal Information Processing Standards (FIPS) Publications are standards issued by *NIST* after approval by the Secretary of Commerce pursuant to the Federal Information Security Management Act (FISMA)

The particular standards of note in SIMP are:

- FIPS 140-2
- FIPS 140-3

Source: FIPS Publications

Firewalld

A dynamically managed firewall with support for network/firewall zones that define the trust level of network connections or interfaces.

Source: Firewalld Homepage

FOSS
Open Source

Software licensed in a manner that keeps its source code publicly accessible and free to modify and redistribute.

See: The Open Source Definition

FQDN
Fully Qualified Domain Name

A domain name that specifies its exact location in the tree hierarchy of the *DNS*. It specifies all domain levels, including the top-level domain and the root zone. An FQDN is distinguished by its unambiguity; it can only be interpreted one way.

FreeIPA

A *FOSS* identity management solution for Linux, with an emphasis on ease-of-use.

FreeIPA is managed from simple web interface and CLI commands. Behind the scenes, it combines *389 DS*, MIT *Kerberos*, *NTP*, *DNS*, and the Dogtag PKI certificate manager. Client systems interact with FreeIPA via *SSSD*.

Red Hat Identity Management (IdM) is the commercial subscription version of FreeIPA.

Website: https://www.freeipa.org/page/Main_Page

Git

A distributed version control system that supports branches.

Website: https://www.git-scm.com

GPG
GnuPG
Gnu Privacy Guard

A complete and free implementation of the OpenPGP standard as defined by RFC4880 (also known as PGP).

Source: GnuPG Homepage

Grafana

A system of pluggable panels and data sources allowing easy extensibility and a variety of panels, including fully featured graph panels with rich visualization options. There is built in support for many of the most popular time series data sources.

Source: Grafana Homepage

GUI
Graphical User Interface

A type of interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, as opposed to text-based interfaces, typed command labels or text navigation.

Source: Wikipedia: Graphical User Interface

HDD
Hard Disk Drive

A device for storing and retrieving digital information, primarily computer data.

Hiera

A key/value lookup tool for user-defined data hierarchies, generally organized based on *Facter* facts. *Puppet* uses Hiera as it compiles catalogs to provide data during *Automatic Parameter Lookup* or explicit `lookup()` calls.

Hiera's hierarchy and data are usually stored as *YAML* or *EYAML* files in the *Control Repository*. However, data can be provided from any source with a custom *Hiera backend*.

Source: Hiera Overview

Hiera backend

A *Hiera* plugin used to retrieve information from a data source and return it appropriately for use in *Puppet*.

See: Hiera: Writing new data backends

The *SIMP Compliance Engine* uses a custom Hiera backend to enforce *Compliance Profile* settings.

hiera-eyaml
Hiera eyaml
EYAML

Hiera eyaml provides a way to keep encrypted data in *Hiera* data files. It consists of:

- The `eyaml` *Hiera backend*
- The **`eyaml`** *CLI* command
- `.eyaml` Hiera data files

The Hiera backend is configured with decryption keys and reads from `.eyaml` files. The CLI command is used to generate keys, encrypt/decrypt data, and edit `.eyaml` files. The `.eyaml` files are YAML that may contain a mix of encrypted and plaintext values—although for performance reasons, it is common practice to limit `.eyaml` files to encrypted data and use plaintext `.yaml` files for everything else.

Encrypted values are prefixed with their encryption method and wrapped with `ENC[]`. They can occur within arrays, hashes, nested arrays and nested hashes. The default encryption mechanism is PKCS7, however further encryption types (like *GPG*) can be added with plugins, which are distributed as RubyGems.

The `eyaml` backend and CLI command are provided by the hiera-eyaml RubyGem, which is been packaged as part of the *Puppet Server*.

Puppet documentation: Configuring a hierarchy level: hiera-eyaml
GitHub project for the hiera-eyaml gem: https://github.com/voxpupuli/hiera-eyaml

HIRS
Host Integrity at Runtime and Start-up

Attestation Certificate Authority (ACA) and *TPM* Provisioning with trusted computing-based supply chain validation.

Source: HIRS

IMA
Integrity Management Architecture
The integrity subsystem is to detect if files have been accidentally or maliciously altered, both remotely and locally.

Source: IMA Sourceforge Page

initrd
The *Initial RAMDisk*. A complete environment that is loaded at boot time to enable booting the rest of the operating system.

InSpec
An open-source testing framework for infrastructure with a human-readable language for specifying compliance, security and other policy requirements.

Source: InSpec Homepage

IP
IP Address
Internet Protocol Address
A numerical label assigned to each device (e.g., computer, printer) participating in a computer network that uses the Internet Protocol for communication.

Source: Wikipedia: IP Address

IP6Tables
Internet Protocol 6 Tables
A user space application that provides an interface to the IPv6 firewall rules on modern Linux systems.

IPTables
Internet Protocol Tables
A user space application that provides an interface to the IPv4 firewall rules on modern Linux systems.

ISO
ISO 9660
A file system standard published by the International Organization for Standardization (ISO) or optical disc media.

Source: Wikipedia: ISO_9660

JSON
JavaScript Object Notation
A lightweight data-interchange format defined by ECMA-404.

Source: JSON Homepage

KDC
Key Distribution Center
Part of a cryptosystem intended to reduce the risks inherent in exchanging keys. KDCs often operate in systems within which some users may have permission to use certain services at some times and not at others.

Kerberos
A computer network authentication protocol that works on the basis of "tickets" to allow nodes communicating over a non-secure network to prove their identity to one another in a secure manner.

Kickstart
Automated installation procedure for Red Hat Linux and other Linux distributions.

See: Kickstart

LDAP
Lightweight Directory Access Protocol

A protocol for querying and modifying LDAP directory services including information such as names, addresses, email, phone numbers, and other information from an online directory.

LDIF
Lightweight Directory Interchange Format

A standard plain text data interchange format for representing *LDAP* (Lightweight Directory Access Protocol) directory content and update requests. LDIF conveys directory content as a set of records, one record for each object (or entry). It also represents update requests, such as Add, Modify, Delete, and Rename, as a set of records, one record for each update request.

Source: Wikipedia: LDAP Data Interchange Format

Logstash

An open source, server-side data processing pipeline that ingests data from a multitude of sources simultaneously, transforms it, and then sends it to your favorite "stash."

Source: Logstash Homepage

LUKS
Linux Unified Key Setup

The standard for Linux hard disk encryption.

See: The LUKS Homepage

MAC
MAC Address
Media Access Control
Media Access Control Address

A unique identifier assigned to network interfaces for communications on the physical network segment.

Source: Wikipedia: MAC address

Mandatory Access Control

A type of access control by which the operating system constrains the ability of a subject or initiator to access or generally perform some sort of operation on an object or target.

Source: Wikipedia: Mandatory access control

Meltdown

A hardware vulnerability affecting Intel x86 microprocessors, IBM POWER processors, and some ARM-based microprocessors. It allows a rogue process to read all memory, even when it is not authorized to do so.

Source: Wikipedia: Meltdown (security vulnerability)

NAT
Network Address Translation

The process of modifying IP address information in IP packet headers while in transit across a traffic routing device.

NFS
Network File System

A distributed file system protocol that allows a user on a client computer to access files over a network in a manner similar to how local storage is accessed.

NIST
National Institute of Standards and Technology

The National Institute of Standards and Technology (NIST) was founded in 1901 and now part of the U.S. Department of Commerce. NIST is one of the nation's oldest physical science laboratories.

Source: NIST - About NIST

NIST 800-171
NIST SP 800-171
NIST Special Publication 800-171

Protecting Controlled Unclassified Information in Nonfederal Information Systems and Organizations

See: SP 800-171

NIST 800-53
NIST SP 800-53
NIST Special Publication 800-53

Security and Privacy Controls for Federal Information Systems and Organizations

See: SP 800-53

NIST information limiting requirements

Specific *NIST 800-53* controls that prohibit passing information to vendors without justification.

Per NIST 800-53r4 AC-20(1) and SC-38:

"*OPSEC* safeguards help to protect the confidentiality of key information including, for example, limiting the sharing of information with suppliers and potential suppliers of information system components, information technology products and services, and with other non-organizational elements and individuals."

NIST SP
NIST Special Publication

A set of publications that provide computer/cyber/information security and guidelines, recommendations, and reference materials.

See: NIST Special Publications

NTP
Network Time Protocol

A networking protocol designed to keep computer systems' clocks within a few milliseconds of each other.

SIMP manages NTP servers and clients on EL7 using simp/ntpd and on EL8 using voxpopuli/chrony.

See: Wikipedia: Network Time Protocol

Website: http://www.ntp.org

OATH
Initiative for Open AuTHentication

A technical framework for open authentication.

Source: OATH Reference Architecture

OpenLDAP

A *FOSS* implementation of *LDAP*. The full suite includes a server, libraries, utilities, and "sample" LDAP clients.

SIMP managed the OpenLDAP server as a default LDAP service through EL7. However, the openldap-servers package was deprecated in EL7.4 and removed in EL8. SIMP supports the *389 Directory Server* as a replacement on EL8+.

See: OpenLDAP website

OpenSCAP

The OpenSCAP project provides tools that are free to use anywhere you like, for any purpose. Availability of the code results in greater portability – anyone can send patches to add support for their platform of choice.

Source: OpenSCAP Features

OPSEC
Operations Security

A process that identifies critical information to determine if friendly actions can be observed by enemy intelligence, determines if information obtained by adversaries could be interpreted to be useful to them, and then executes selected measures that eliminate or reduce adversary exploitation of friendly critical information

Source: Wikipedia: Operations Security

OS
Operating System

System software that manages computer hardware and software resources and provides common services for computer programs. All computer programs, excluding firmware, require an operating system to function.

Source: Wikipedia: Operating system

PAM
Pluggable Authentication Modules

A mechanism to integrate multiple low-level authentication schemes into a high-level application programming interface (API). It allows programs that rely on authentication to be written independent of the underlying authentication scheme.

PE
Puppet Enterprise

A version of Puppet with commercial support by Puppet, Inc. that provides additional capabilities beyond *Puppet*.

PEM
Privacy Enhanced Mail

An early standard for securing electronic mail. This is the public-key of a specific certificate. This is also the format used for Certificate Authority certificates.

PERL
Practical Extraction and Report Language

A high-level, general-purpose, interpreted, dynamic programming language. PERL was originally developed by Larry Wall in 1987 as a general-purpose Unix scripting language to make report processing easier.

PKI
Public Key Infrastructure

A security architecture that has been introduced to provide an increased level of confidence for exchanging information over an increasingly insecure Internet. PKI enables users of a basically insecure public networks, such as the Internet, to securely authenticate to systems and exchange data. The exchange of data is done by using a combination of cryptographically bound public and private keys.

Puppet

An *Open Source* configuration management tool written and maintained by Puppet, Inc.. Written as a Ruby *DSL*, Puppet provides a declarative language that allows system administrators to provide a consistently applied management infrastructure. Users describes system resource and resource state in the Puppet language. Puppet discovers system specific information via *Facter* and compiles Puppet manifests into a system-specific catalog containing resources and resource dependencies, which are applied to each client system.

Puppet Custom Type
Custom Type

New *Puppet Resources*, written in *Ruby*, that add custom client-side capabilities to the Puppet language.

See: Custom Types:

Puppet Data Type

Added in Puppet version 4, strong data types allow for the enforcement of inherent parameter validation as well as a better understanding of what function the data performs in classes.

See: Language: Data Types

Puppet Environment
Puppet Environments

As the Puppet Environment documentation puts it:

> "An environment is a branch that gets turned into a directory on your primary server."

Depending on the context, the term has several implications:

- When describing **infrastructure**, a Puppet environment is a group of Puppet agent nodes the *Puppet Server* manages in isolation from other agent nodes (which belong to other Puppet environments).
- When discussing **file paths** or **directories** on a *Puppet Server*: a Puppet environment is a directory of source, data, and module files used to compile Puppet catalogs for such a group of agents.

  Each Puppet environment directory is located on a Puppet server at `/etc/puppetlabs/code/environments/environment_name`.
- In a *Control Repository*, each **git branch** represents a separate Puppet environment, which *r10k* or PE *Code Manager* will deploy to a Puppet server as a Puppet environment directory with the same name.
- From the perspective of the Puppet compiler and language, a Puppet environment can be seen as a global **namespace**.

See: https://puppet.com/docs/puppet/latest/environments_about.html

Puppet Module
Puppet Modules

A self-contained bundle of code and data able to be processed by the `puppet` application.

Puppet Namespace

A mechanism used by the `puppet` compiler to uniquely identify code during compilation. Generally, namespaces align with *Puppet Module* file paths and are separated by two colons at each directory.

See: Namespaces and Autoloading

Puppet Resource
Puppet Resources

The fundamental unit for modeling system configurations in *Puppet*.

See: Resources:

PuppetDB

An *Open Source* project, PuppetDB collects data generated by *Puppet*. It enables advanced Puppet features like exported resources, and can be the foundation for other applications that use Puppet's data.

Source: *PuppetDB Overview <https://puppet.com/docs/puppetdb/latest>*

Puppetfile

A Ruby file that contains references to *Puppet modules*.

See the Puppetfile spec: https://github.com/puppetlabs/r10k/blob/main/doc/puppetfile.mkd

PuppetForge

An official repository for Puppet modules

See: https://forge.puppet.com/

Puppetserver
Puppet Server

An application that runs on the Java Virtual Machine (JVM) and provides a puppet compiler service by running several JRuby interpreters running the Puppet compiler through a Clojure-based service.

Source: Puppet's Services: Puppet Server

PXE
Preboot Execution Environment

An environment to boot computers using a network interface independently of data storage devices (like hard disks) or installed operating systems.

r10k

A code management tool that uses *git* branches to automate the development and deployment of *Puppet* code.

Rake
Ruby Make

A Make-like program implemented in Ruby.

Source: Rake Homepage

RAM
Random Access Memory

A form of computer data storage. A random access device allows stored data to be accessed in nearly the same amount of time for any storage location, so data can be accessed quickly in any random order.

Red Hat
Red Hat®
Red Hat®, Inc.

A collection of many different software programs, developed by Red Hat®, Inc. and other members of the Open Source community. All software programs included in Red Hat Enterprise Linux® are GPG signed by Red Hat®, Inc. to indicate that they were supplied by Red Hat®, Inc.

See also *Red Hat Enterprise Linux*

Red Hat Directory Server
RHDS

A commercially-supported *LDAP* server from *Red Hat*, based on *389 DS*.

Website: Red Hat Directory Server

RHEL
Red Hat Enterprise Linux

A commercial Linux operating system produced by *Red Hat®*, Inc. RHEL is designed to provide an Enterprise-ready Linux distribution suitable to multiple target applications.

Homepage: https://www.redhat.com/en/technologies/linux-platforms/enterprise-linux

RPM
RPM Package Manager

A package management system. The name RPM is associated with the .rpm file format, files in this format, software packaged in such files, and the package manager itself. RPM was developed primarily for GNU/Linux distributions; the file format is the baseline package format of the Linux Standard Base.

RSA

An algorithm for public-key cryptography that is based on the presumed difficulty of factoring large integers, the factoring problem. RSA stands for Ron Rivest, Adi Shamir and Leonard Adleman, who first publicly described it in 1977.

Rsync

An open source utility that provides fast incremental file transfer.

Source: Rsync Home Page

Rsyslog

An open-source software utility used on UNIX and Unix-like computer systems for forwarding log messages in an IP network. It implements the basic syslog protocol, extends it with content-based filtering, rich filtering capabilities, flexible configuration options and adds features such as using TCP for transport.

Source: Wikipedia: Rsyslog

Ruby

A dynamic, reflective, general-purpose object-oriented programming language that combines syntax inspired by Perl with Smalltalk-like features. Ruby originated in Japan during the mid-1990s and was first developed and designed by Yukihiro "Matz" Matsumoto. It was influenced primarily by Perl, Smalltalk, Eiffel, and Lisp. Ruby supports multiple programming paradigms, including functional, object oriented, imperative and reflective. It also has a dynamic type system and automatic memory management; it is therefore similar in varying respects to Smalltalk, Python, Perl, Lisp, Dylan, Pike, and CLU.

RVM
Ruby Version Manager

A command-line tool which allows you to easily install, manage, and work with multiple *Ruby* environments from interpreters to sets of gems.

Source: RVM Homepage

SCAP
Security Content Automation Protocol

A synthesis of interoperable specifications derived from community ideas.

Source: SCAP Homepage

SELinux

A Linux kernel security module that provides a mechanism for supporting access control security policies, including United States Department of Defense–style mandatory access controls (MAC).

Source: Wikipedia: Security-Enhanced Linux

Service Account

An account that is not for use by a human user but which still requires login access to a host.

SFTP
SSH File Transfer Protocol

A network protocol that provides file access, file transfer, and file management functionalities over any reliable data stream. It was designed by the Internet Engineering Task Force (IETF) as an extension of the Secure Shell protocol (*SSH*) version 2.0 to provide secure file transfer capability, but is also intended to be usable with other protocols.

SIMP
System Integrity Management Platform

A security framework that sits on top of *RHEL* or *CentOS*.

SIMP CE
SIMP Community Edition

The *FOSS* version of SIMP made freely available under the *Apache 2.0 license*.

Comparison: SIMP Editions Overview

SIMP Compliance Engine

A SIMP component that adds the capability to evaluate your *Puppet* code for compliance with a policy as well as enforcing that the code enacts the specified policy.

See: SIMP Compliance Engine Repository

SIMP EE
SIMP Enterprise Edition

A version of SIMP with commercial support by Onyx Point, Inc. that provides additional capabilities beyond *SIMP CE*.

Comparison: SIMP Editions Overview Documentation: SIMP Enterprise Edition

SIMP Omni-Environment

A set of 3 environments (directories) on the Puppet server that is required for a SIMP Puppet environment to operate. Includes a *Puppet Environment*, a *SIMP Secondary Environment*, and a *SIMP Writable Environment*.

See: *SIMP Environments*

SIMP Omni-Environment skeleton

A *SIMP Omni-Environment* in which modules have not been deployed yet.

SIMP Secondary Environment
Secondary Environment
Secondary Environments

SIMP-specific assets that support a corresponding Puppet environment, but which must be maintained independently. The Secondary Environment directory is `/var/simp/environments/<environment_name>`.

SIMP Server

The first server that is built in a SIMP environment and the server that is expected to be the nexus of control for the managed infrastructure.

See: *Puppet Server*

SIMP Writable Environment
Writable Environment
Writable Environments

Puppet environment-specific SIMP data generated and/or read in by SIMP Puppet functions on the Puppet Server during catalog compilation. The Writable Environment directory is `/opt/puppetlabs/server/data/puppetserver/simp/environments/<environment_name>`.

Site Manifest

Puppet always starts compiling with either a single manifest file or a directory of manifests that get treated like a single file. This main starting point is called the main manifest or site manifest. By default, the main manifest for a given environment is <ENVIRONMENTS DIRECTORY>/<ENVIRONMENT>/manifests.

Source: Puppet Documentation: Main manifest directory

Site Profile

This term is used throughout the documentation to refer to a *Puppet Module* that is specific to your site. This simply allows for a common isolated *Puppet namespace* to reduce confusion in the documentation. You could add a module literally called `site` to your environment which would make the examples generally able to be copied and pasted into files in the new module.

You may see various shorthand code snippets that refer to `site::<name>`. This indicates that the *class* should be created somewhere specific to your site and does not dictate the naming of the class.

When referred to by path, the path will start at the `modules` directory for easy reference. This should be expanded to the target *Puppet environment* path.

The following code snippet can be used to determine your module path.

```
$env_path=`puppet config print environmentpath`
$env=`puppet config print environment`

echo "${env_path}/${env}/modules/site"
```

SNMP
Simple Network Management Protocol

A protocl for collecting and organizing information about managed devices on IP networks.

Source: Wikipedia: Simple Network Management Protocol

Spectre

A vulnerability that affects modern microprocessors that perform branch prediction.

Source: Wikipedia: Spectre (security vulnerability)

SSG
SCAP Security Guide

A security policy written in a form of *SCAP* documents. The security policy created in SCAP Security Guide covers many areas of computer security and provides the best-practice solutions. The guide consists of rules with very detailed description and also includes proven remediation scripts, optimized for target systems. SCAP Security Guide, together with *OpenSCAP* tools, can be used for auditing your system in an automated way.

Source: OpenSCAP Homepage

See Also: *SCAP*

SSH
Secure Shell

An application for secure data communication, remote shell services, or command execution between networked computers. SSH utilizes a server/client model for point-to-point secure communication.

SSL
Secure Sockets Layer

The standard security technology for using *PKI* keys to provide a secure channel between two servers.

See also *TLS*.

SSSD
System Security Services Daemon

A daemon that provides access to identity and authentication remote resource through a common framework that can provide caching and offline support to the system.

Source: SSSD Homepage

STIG
DISA STIG
Defense Information Systems Agency Secure Technical Implementation Guide

Configuration standards for DOD IA and IA-enabled devices/systems.

Source: DoD Cyber Exchange

Stunnel

A proxy designed to add *TLS* encryption functionality to existing clients and servers without any changes in the programs' code.

Source: Stunnel Home Page

Sudo

sudo allows a permitted user to execute a command as the superuser or another user, as specified by the security policy. The invoking user's real (not effective) user ID is used to determine the user name with which to query the security policy.

Source: The `SUDO(8)` man page

Sudosh

An application that acts as an echo logger to enhance the auditing of privileged activities at the command line of the operating system. Utilities are available for playing back sudosh sessions in real time.

Sudosh has been replaced by *Tlog* in the latest SIMP distributions.

Swappiness

Swappiness is a Linux kernel parameter that controls the relative weight given to swapping out of runtime memory, as opposed to dropping pages from the system page cache.

Source: Wikipedia: Swappiness

SYN cookies
syncookies

A technique used to resist SYN flood attacks.

Source: Wikipedia: SYN cookies

Syslog

A set of standards for sending log messages across the network.

Source: Wikipedia: syslog

tboot
Trusted Boot

See *TXT*.

TCPWrappers
TCP Wrappers

A host-based networking *ACL* system, used to filter network access to Internet Protocol servers on (Unix-like) operating systems such as Linux or BSD. It allows host or subnetwork *IP* addresses, names and/or ident query replies, to be used as tokens on which to filter for access control purposes.

Source: Wikipedia: TCP Wrappers

TFTP
Trivial File Transfer Protocol

A file transfer protocol generally used for automated transfer of configuration or boot files between machines in a local environment.

Tlog

Tlog is a terminal I/O recording and playback package suitable for implementing centralized user session recording.

Tlog has replaced *Sudosh* as the preferred terminal logging program in SIMP.

source: The Tlog home page

TLS
Transport Layer Security

A cryptographic protocol that provides network communications security. TLS and *SSL* encrypt the segments of network connections above the Transport Layer, using asymmetric cryptography for privacy and a keyed message authentication codes for message reliability.

See also *SSL*.

TOTP
Time-based One-Time Password algorithm

An algorithm that generates a one-time password, taking uniqueness from the current time. It has been adopted by IETF standard RFC 6238.

Source: Wikipedia: Time-based One-time Password algorithm

TPM
Trusted Platform Module

An international standard for a secure cryptoprocessor, which is a dedicated microcontroller designed to secure hardware by integrating cryptographic keys into devices.

Source: Wikipedia: Trusted Platform Module

TTY

A Unix command that prints to standard output the name of the terminal connected to standard input. The name of the program comes from teletypewriter, abbreviated "TTY".

TXT
Trusted Execution Technology

A hardware feature designed to harden platforms from the emerging threats of hypervisor attacks, BIOS, or other firmware attacks, malicious root kit installations, or other software-based attacks. It increases protection by allowing greater control of the launch stack through a Measured Launch Environment (MLE) and enabling isolation in the boot process.

Source: Intel Trusted Execution Technology: White Paper

UEFI
Unified Extensive Firmware Interface

A specification that defines a software interface between an operating system and platform firmware. UEFI replaces the Basic Input/Output System (BIOS) firmware interface.

Source: Wikipedia: UEFI

umask

Umask is a command that determines the settings of a mask that controls how file permissions are set for newly created files. It also may refer to a function that sets the mask, or it may refer to the mask itself, which is formally known as the file mode creation mask. The mask is a grouping of bits, each of which restricts how its corresponding permission is set for newly created files. The bits in the mask may be changed by invoking the umask command.

Source: Wikipedia: umask

UUID
Universally Unique Identifier

A 128-bit unique value that is generally written as groups of hexadecimal digits separated by hyphens.

See also: UUIDGEN(1)

Vagrant

A tool for building complete development environments. With an easy-to-use workflow and focus on automation, Vagrant lowers development environment setup time, increases development/production parity, and makes the "works on my machine" excuse a relic of the past.

Source: Vagrant: About Vagrant

VirtualBox

A general-purpose full virtualizer for x86 hardware, targeted at server, desktop and embedded use.

Source: Wikipedia: VirtualBox

VM
Virtual Machine

An isolated guest operating system installation running within a host operating system.

VNC
Virtual Network Computing

A graphical desktop sharing system that uses the remote framebuffer (RFB) protocol to control another computer remotely. It transmits the keyboard and mouse events from one computer to another, relaying the graphical screen updates back in the other direction, over a network.

WAN
Wide Area Network

A computer networking technology used to transmit ata over long distances, and between different Local Area Networks (LANs), Metropolitan Area Networks (MANs), and other localized computer networking architectures.

X
X11
X Windows
X Window System

The X Window System (X11, or shortened to simply X) is a windowing system for bitmap displays, common on UNIX-like computer operating systems.

Source: Wikipedia: X Window System

X.509

An ITU-T standard for a public key infrastructure (PKI) and Privilege Management Infrastructure (PMI). X.509 specifies, amongst other things, standard formats for public key certificates, certificate revocation lists, attribute certificates, and a certification path validation algorithm.

Source: Wikipedia: X.509

YAML
YAML Ain't Markup Language

A human friendly data serialization standard for all programming languages.

Source: YAML Homepage

YUM
Yellowdog Updater, Modified

A software installation tool for Linux. It is a complete software management system that works with RPM files. YUM is designed to be used over a network or the Internet.

See also *RPM*.

INDEX