SIMP Documentation

Release 0-0

THE SIMP TEAM

June 09, 2016

Contents

This is the 5.1.0-0 release of SIMP compatible with the 7.1 release of CentOS and Red Hat Enterprise Linux (RHEL).

The System Integrity Management Platform (SIMP) is a 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 Guides.

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.

Contents:

CHAPTER 1

SIMP 5.1.0-0

1.1 Changelog

Contents

SIMP 5.1.0-0

Package: 5.1.0-0

This release is known to work with:

- RHEL 7.0 and 7.1 x86_64
- CentOS 7.0 x86_64 (1406 and 1503)

Warning: The default system passwords have changed! Please see the User's Guide for details.

1.1.1 Manual Changes Requred

- Bugs in the *simplib::secure_mountpoints* class (formerly *common::secure_mountpoints*)

Note: This only affects you if you did not have a separate partition for /tmp!

- There were issues in the secure_mountpoints class that caused /tmp and /var/tmp to be mounted against the root filesystem. While the new code addresses this, it cannot determine if your system has been modified incorrectly in the past.
- To fix the issue, you need to do the following: - Unmount /var/tmp (may take multiple unmounts) - Unmount /tmp (may take multiple unmounts) - Remove the 'bind' entries for /tmp and /var/tmp from /etc/fstab - Run **puppet** with the new code in place

1.1.2 Deprecations

- simp-hiera

  The *simp-hiera* RPM has been replaced by the upstream *hiera* package from Puppet Labs. The original simp-hiera fork had been maintained due to a need that the 'alias()' function now serves. Please run the *hiera_upgrade* script to convert your existing SIMP environment. You may also set the environment variable *HIERA_UPGRADE* to a path of your choice to update any other hieradata that you may have on your system.

- pupmod-simp-common

  The *::common* namespace has been deprecated in favor of the new *::simplib* namespace. This removes a commonly conflicting module name from the SIMP ecosystem.

  You will need to run the *migrate_to_simplib* script to update all of the relevant files. This script will only migrate items in the existing SIMP environment. You may also set the environment variable *UPGRADE_PATHS* to run the script on multiple external paths.

  All code was migrated.

- pupmod-simp-functions

  The *::functions* namespace has been deprecated in favor of the new *::simplib* namespace. This removes a commonly conflicting module name from the SIMP ecosystem.

  You will need to run the *migrate_to_simplib* script to update all of the relevant files. This script will only migrate items in the existing SIMP environment. You may also set the environment variable *UPGRADE_PATHS* to run the script on multiple external paths.

  The following items were not migrated:

  - append_if_no_such_line => Use simp_file_line{}
  - delete_lines => Use augeas{}
  - init_mod_nice => Use init_ulimit{}
  - init_mod_open_files => Use init_ulimit{}
  - line => Use augeas{}
  - prepend_if_no_such_line => Use simp_file_line{}
  - renice => No replacement, was not correct
  - replace_line => Use augeas{}

1.1.3 Significant Updates

- FIPS Mode is now enabled by default!
  - This is a SIGNIFICANT change and may impact many of your running applications that use encryption.

  - If you are upgrading, do **NOT** enable FIPS mode without extensive testing as it may cause various applications to not function properly any longer.

- The rsyslog module has been completely rewritten to support rsyslog 7.4. This is a breaking change from previous releases and will require active updates to existing systems. All modules with rsyslog integration ave been updated to accommodate this change:
  - Critical Variable Changes
    - The global *rsyslog::log_server_list* variable is now set to send to **all** of the servers in the Array by default.
      - This variable defaults to the global *log_servers* Array in Hiera.
    - There is a new variable *rsyslog::failover_log_servers* which is an Array of failover log servers to be used for your system. These will be tried, in order, until successful messages can be sent.
  - Updated Modules:
    - aide
    - apache
    - auditd
    - dhcp
    - logstash
    - openldap
    - rsync
    - simp
    - sudosh
- There was a bug in previous versions of SIMP that require the following LDIF to be run manually on the systems to correct the password policy checking.

  dn: cn=default,ou=pwpolicies,dc=your,dc=domain changetype: modify replace: pwdCheckModule pwdCheckModule: simp_check_password.so - dn: cn=noExpire_noLockout,ou=pwpolicies,dc=your,dc=domain changetype: modify replace: pwdCheckModule pwdCheckModule: simp_check_password.so
- The Electrical and SIMP modules for elasticsearch have been combined.

1.1.4 Upgrade Guidance

Fully detailed upgrade guidance can be found in the **Upgrading SIMP** portion of the *User's Guide*.

Warning: You must have at least **2.2GB** of **free** RAM on your system to upgrade to this release.

Note: Upgrading from releases older than 5.0 is not supported.

Expectations

Before you begin, please be aware that the following actions will take place as a result of the migrate_to_environments script:

- The *puppet-server* RPM will be removed

- The *puppetserver* RPM will be installed (no, that's not a typo)
- **ALL** SIMP Puppet code will be migrated into a new *simp* environment
  - This will be located at */etc/puppet/environments/simp*
- A backup of your running environment will be made available at */etc/puppet/environments/pre_migration.simp*
  - You will find timestamped directories under the *pre_migration.simp* directory that correspond to runs of the migration script
  - Your old files will be in a *backup_data* directory and will be linked to a local bare Git repository in the same space

The upgrade steps will also have you install PuppetDB. PuppetDB is installed by default if you kick from the DVD.

1.1.5 Security Announcements

CVEs Addressed

1.1.6 RPM Updates

Numerous RPMs were updated in the creation of this release. Several were included due to our use of *repoclosure* to ensure that RPM dependencies are met when releasing a DVD.

- This version include the latest RedHat 7.1 and CentOS 7.0 (1503) RPMs.
- Facter upgraded to 2.4.
- PuppetDB upgraded to 2.3.8-1

1.1.7 Fixed Bugs

- pupmod-aide
  - Change the call to the *rsyslog* init script to the *service* command to seamlessly support both RHEL6 and RHEL7.
- pupmod-apache
  - Removed all reliance on 'lsb*' facts since some environments do now wish to install the prerequisites for those facts to run.
  - Remove the apache_version fact and simply use the version controls built into the Apache configuration language.
  - Update all custom functions to properly scope definitions.
  - Ensure that mod_ldap is installed in SIMP >= 5.0.
  - Prevent apache from restarting after downloading a CRL.
- pupmod-clamav
  - Change the call to the *rsyslog* init script to the *service* command to seamlessly support both RHEL6 and RHEL7.
- pupmod-common => Deprecated - Replaced by pupmod-simplib!
- pupmod-simplib
  - Fixed the secure_mountpoints code so that it no longer incorrectly bind mounts /tmp or /var/tmp.

  - We no longer supply crontab or anacrontab in global_etcd.
  - Remove dynamic_swappiness cron job if a static value is set.
  - Ensure that the *passgen()* function fails on invalid scenarios. This prevents the accidental cration of empty passwords.
  - Allow the value *2* to be used for `rp_filter` in `simplib::sysctl`.
  - Added ability to return remote ip addrs.
- pupmod-dhcp
  - Change the call to the *rsyslog* init script to the *service* command to seamlessly support both RHEL6 and RHEL7.
- pupmod-elasticsearch
  - Ensured that Elasticsearch works properly with the new version of Apache.
  - Removed our default ES tuning since the default works better for LogStash.
  - Ensure that Puppet manages the Elasticsearch logging file.
- pupmod-functions
  - Fixed sysv.rb to explicitly require puppet/util/selinux, which caused puppet describe to have errors.
- pupmod-iptables
  - Fixed a bug that would cause issues with Ruby 1.8.7.
  - Fixed DNS resolution in IPv6.
  - Prevent IPv6 ::1 spoofed addresses by default.
- pupmod-simp-logstash
  - Fix issues with both TCPWrappers and IPTables when used with LogStash.
- pupmod-nfs
  - Updated the *mountd* port to be *20048* by default for SELinux issues in RHEL7.
- pupmod-ntp
  - Updated against NTP Security Vulnerabilities (Red Hat Article #1305723).
  - Ensure that *restrict* entries use DDQ format.
- pupmod-openldap
  - The Password Policy overlay was getting loaded into the default.ldif even if you didn't want to use it. This has been fixed.
  - Made the password policy overlay align with the latest SIMP build of the plugin.
    * This means that you *must* have version simp-ppolicy-check-password-2.4.39-0 or later available to the system being configured.
  - Change the call to the *rsyslog* init script to the *service* command to seamlessly support both RHEL6 and RHEL7.
  - Fixed reported bugs in syncrepl.pp.
  - Removed all reliance on the 'lsb*' facts since some users do not wish to install the prerequisite RPMs for LSB compliance.
- pupmod-openscap

  - Change the call to the *rsyslog* init script to the *service* command to seamlessly support both RHEL6 and RHEL7.
  - Changed default ssg base path to /usr/share/xml/scap/ssg/content

- pupmod-pam
  - Removed all reliance on the 'lsb*' facts since some users do not wish to install the prerequisite RPMs for LSB compliance.
- pupmod-pki
  - Now allow directories in the cacerts directories. This previously caused failures that needed to be manually addressed on each node.
- pupmod-rsync
  - Fixed provider to run with –dry-run when puppet is run with a –noop.
- pupmod-simp
  - Ensure that SSSD is used by default on EL7+ systems since nscd and nslcd have functionality issues.
  - Removed all reliance on the 'lsb*' facts since some users do not wish to install the prerequisite RPMs for LSB compliance.
- pupmod-ssh
  - Modernized the Ciphers, MACs, and Kex.
  - Added explicit cases for FIPS and non-FIPS mode (as well as reasonable default cases for RHEL7 and below).
  - Updated to use the new augeasproviders module dependencies.
  - Added a function *ssh_format_host_entry_for_sorting()* that will properly sort SSH *Host* entries for inclusion with concat.
- pupmod-stunnel
  - Had a variable **options** in *stunnel.erb* that should have been scoped as **@options**.
- pupmod-sudo
  - Removed all reliance on the 'lsb*' facts since some users do not wish to install the prerequisite RPMs for LSB compliance.
- pupmod-sudosh
  - Change the call to the *rsyslog* init script to the *service* command to seamlessly support both RHEL6 and RHEL7.
- pupmod-sysctl
  - Removed support for the old parsed-file provider and moved to using the new Augeas-based provider.
- pupmod-tftpboot
  - Purging of non-Puppet-managed items in *pxelinux.cfg* is now optional.
- pupmod-simp-tpm
  - IMA is disabled by default.
- simp-gpgkeys
  - Ensure that the keys are set in the correct locations for the target SIMP distribution.
- simp-rsync

  - Removed spurious install messages.
- simp-util
  - Fixed the targets of unpack_dvd.
  - Added a **use_fips** boolean to *simp config*
- pupmod-xinetd
  - Fixed: The default log_type should be 'SYSLOG authpriv' instead of 'SYSLOG daemon info'.
- pupmod-vnc
  - Removed banners that broke some vnc clients.
- simp-cli
  - *simp config -a ANSWERFILE* fails when an item has no answer
  - *simp config -A ANSWERFILE* prompts when an an item has no answer
  - The misleading *–help* documentation for *-ff* has been removed
  - The Config::Item *use_fips* now echoes its command unless *@silent*
  - The *simp doc* command path to the documentation has been corrected.
  - General usability improvements.
- DVD
  - NetworkManager-wait-online is now set by default in the ISO supplied kickstart images. Without ths, it is possible for the 'runpppet' script to attempt to run prior to the network being initialized.
  - A default IP is no longer provided when booting from the ISO; *simp config* will set the network properly.
  - The default kickstart no longer attempts to chkconfig any services in the %post section.

1.1.8 New Features

- pupmod-auditd
  - Completely overhauled the module with a focus on better acceptance testing and format compliance.
- pupmod-augeasproviders
  - This was updated to 2.1.3.
  - The update to 2.1.3 caused the addition of all of the pupmod-augeasproviders modules below.
- augeasproviders_apache
  - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_base
  - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_core
  - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_grub
  - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_mounttab

    - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_nagios
    - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_pam
    - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_postgresql
    - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_puppet
    - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_shellvar
    - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_ssh
    - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_sysctl
    - Imported 2.1.3 to support the Augeasproviders stack.
- pupmod-augeasproviders
    - This was updated to 2.1.3.
    - The update to 2.1.3 caused the addition of all of the pupmod-augeasproviders modules below.
- pupmod-cgroups
    - Added acceptance tests.
- pupmod-common => Deprecated - Replaced by pupmod-simplib!
- pupmod-simplib
    - Created parse_hosts function.
    - Added full tests for evaluating the ability to toggle FIPS mode.
- pupmod-richardc-datacat
    - Incorporated the *richardc/datacat* module into the core for user convenience.
- pupmod-freeradius
    - Split the Freeradius module based on version so that it can be properly selected against the *installed* version of Freeradius. This may take two runs to coalesce.
- pupmod-puppetlabs-inifile
    - Updated to version 1.2.0.
- pupmod-puppetlabs-puppetdb
    - Updated to version 5.0.0-0.
- pupmod-simp-kibana
    - Add Kibana dashboards to the Kibana module.
    - Allows users to apply default SIMP kibana Dashboards.

- pupmod-simp-logstash
  - Integrated SIMP and Electrical Logstash modules.
  - Changes the existing Logstash module to allow users to apply default SIMP filters.
- pupmod-pki
  - Now generate a system RSA public key against the passed private key.
- pupmod-puppetlabs-postgresql
  - Initial import of the Puppet Labs PostgreSQL module.
  - Modifications were made to support the SIMP concat.
- pupmod-puppetlabs-puppetdb
  - New import of the Puppet Labs PuppetDB module.
- pupmod-simp-rsyslog
  - Module has been rewritten to support rsyslog 7.4.
- pupmod-simp-simp
  - Set the SELinux Boolean 'use_nfs_home_dirs' to 'on' if a remote NFS server is used for home directories.
  - The 'runpuppet' script was modified to run 'fixfiles' on systems prior to the final puppet runs since RHEL7, in some cases, does not appear to honor the '/.autorelabel' file.
- pupmod-puppetlabs-stdlib
  - Updated to version 4.5.1.
- pupmod-sysctl
  - Moved the configuration file updates from sysctl.conf to sysctl.d/20-simp.conf to use the latest update mechanisms.
- pupmod-tftpboot
  - Updated to use native packages and pull as muchs possible.
- simp-doc
  - Updated tables across the board to be more readable.
  - Updated documentation relating to user management and user key management using SSH.
  - Rebranded the documentation and updated the color scheme.
  - Updated the default system passwords.
- simp-rsync
  - Content has been restructured to eliminate licensing conflicts.
  - ClamAV has been refactored into a separate (GPL) package.
- simp-utils
  - simp config was rewritten to allow for new features and flexibility.
  - Now provided as a Ruby gem "simp-cli".
- Mcollective
  - Mcollective is now available to be installed and used with SIMP. It uses SSL/TLS along with user certificates for proper encryption and authentication.

- PuppetDB
  - PuppetDB is now supported by SIMP and installed by default.
- Puppetserver
  - The puppet master service has been replaced by the puppetserver service. This is a major rewrite by Puppetlabs. Puppetserver scales better for larger agent deployments with a single puppet master.
  - Uses Environments by default, this allows for tools such as r10K. Production environment is a link to simp by default.
- Facter 2.4
  - Facter now returns the following facts as their actual boolean or integer values, instead of converting them into strings:

    activeprocessorcount is_virtual mtu_<INTERFACE> physicalprocessorcount processorcount selinux_enforced selinux sp_number_processors sp_packages

1.1.9 Known Bugs

- There is a symlink that is created at /etc/puppet/environments/simp/simp which should not be in place. This is being tracked as SIMP-661
- SSSD is currently broken and will allow logins via SSH even if your password has expired. This has been noted by Red Hat and is in the pipeline.
- If you are running libvirtd, when svckill runs it will always attempt to kill dnsmasq unless you are deliberately trying to run the dnsmasq service. This does *not* actually kill the service but is, instead, an error of the startup script and causes no damage to your system.

CHAPTER 2

SIMP Installation Guide

Contents:

2.1 Introduction

This guide will walk a user through the process of managing a *SIMP* system. This system includes, at a minimum, a SIMP server with properly configured networking information and a working Puppet server. Additionally, this document outlines the process of managing clients and users associated with the SIMP system.

2.1.1 Level of Knowledge

SIMP is designed for use by system administrators or users with a strong background using Linux operating systems. The core applications that make up SIMP and require prerequisite knowledge are:

- *Puppet* - 3.7 or later
- *Domain Name System* (DNS) - BIND 9
- *Dynamic Host Configuration Protocol* (DHCP) - Internet Systems Consortium (ISC) DHCP
- *Lightweight Directory Access Protocol* (LDAP) - OpenLDAP
- RedHat Kickstart (including all tools behind it) - *Trivial File Transfer Protocol* (TFTP), PXELinux, etc.
- Apache
- *Yellowdog Updater, Modified* (YUM)
- Rsyslog Version 3+
- *Internet Protocol Tables* (IPtables) (Basic knowledge of the rules)
- *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 does as much initial setup and configuration of these tools as possible. However, without at least some understanding, you will be unable to tailor a SIMP system to fit the desired environment. A general understanding of how to control and manipulate these tools from the *command line interface* (CLI) will be necessary, as SIMP does not come stock with 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.

2.1.2 SIMP Defined

The System Integrity Management Platform (SIMP) is a 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 Guides.

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.

2.2 SIMP Server Installation

This chapter provides guidance on installing and configuring SIMP using the `simp config` utility.

2.2.1 System Requirements

SIMP scales well, but how much depends on a number of factors, including the number of nodes, the processor speed, the total memory, and the complexity of the manifests. The following minimal system requirements are recommended:

- *Central Processing Unit* (CPU) : 2 Cores
- *Random Access Memory* (RAM) : 2.2 GB
- *Hard Disk Drive* (HDD) : 50 GB

2.2.2 Using the SIMP Utility

The SIMP Utility does not assist users through the entire configuration process; however, it does make the initial configuration easier and more repeatable.

Important: Correct time across all systems is important to the proper functioning of SIMP and Puppet in general.

If a user has trouble connecting to the Puppet server and errors regarding certificate validation appear, check the Puppet server and client times to ensure they are synchronized.

Warning: Keep in mind as the installation process begins that Puppet does not work well with capital letters in host names. Therefore, they should not be used.

2.2.3 SIMP Default Passwords and Settings

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

Important: All default passwords should be changed during the initial configuration proceess.

Table: SIMP Default Passwords

Utility	Password
Grub	GrubPassword
Root User	RootPassword
Simp User	UserPassword

A table of settings that can be changed/defined during installation is located in Appendix B, *List of Installation Variables*. Review this if you are unfamiliar with SIMP.

2.2.4 Preparing the SIMP Server Environment

1. Boot the system and ensure the SIMP ISO is selected.
2. Press *Enter** to run the standard SIMP install, or choose from the customized options list.
3. When the installation is complete, the system will restart automatically.
4. Log on as `root` and type the default password shown in **Table 2.1.**
5. Type the default password again when prompted for the (current) UNIX password.
6. Type a new password when prompted for the New Password. Retype the password when prompted.

2.2.5 Installing the SIMP Server

Warning: Keep in mind as the installation process begins that Puppet does not work well with capital letters in host names. Therefore, they should not be used.

1. Log on as `simp` and run `su -` to gain root access.
2. Type `simp config`

1. Type `simp config -a <Config File>` to load a previously generated configuration instead of generating the configuration from the script. This is the option to run for systems that will be rebuilt often.
2. For a list of additional commands, type `simp help`. Type `simp help ***<Command>***` for more information on a specific command.
3. A list of the variables that are set and more details are contained in *List of Installation Variables*.

Note: Once simp config has been run, a simp config file with all your settings is saved in /root/.simp/simp_conf.yaml

3. Configure the system as prompted.
4. Type `simp bootstrap`

Note: If progress bars are of equal length and the bootstrap finishes quickly, a problem has occurred. This is most likely due to an error in SIMP configuration. Refer to the previous step and make sure that all configuration options are correct.

5. Type `reboot`

2.2.6 Performing Post-installation Setup on the SIMP Server

1. Log on as `root`
2. Run puppet for the first time. Errors will appear for DHCP. These can be safely ingored at this stage. Type: `puppet agent -t`

3. Copy CentOS RHEL_MAJOR_MINOR_VERSION ISO(s) to the server and unpack using the `unpack_dvd` utility. This creates a new tree under `/var/www/yum/CentOS`. Execute: `unpack_dvd CentOS-RHEL_MAJOR_MINOR_VERSION- *####*-x86_64-Everything.iso`
4. Update your system using yum. The updates applied will be dependent on what ISO you initially used. Execute: `yum clean all; yum makecache`
5. Run puppet. Ignore the same DHCP errors: `puppet agent -t`
6. Type `reboot`

2.3 Client Management

This chapter provides guidance to install and configure SIMP clients based on the standard SIMP system installed using the SIMP DVD.

2.3.1 System Requirements

Before installing clients, the system should consist of the following minimum requirements:

- Hardware/*Virtual Machine* (VM) : Capable of running RHEL 6 or 7 ; 64-bit compatible
- RAM: 512 MB
- HDD: 5 GB

2.3.2 Configuring the Puppet Master

Perform the following actions as `root` on the Puppet Master system prior to attempting to install a client.

2.3.3 Configure DNS

Most static files are pulled over `rsync` by Puppet in this implementation for network efficiency. Specific directories of interest are noted in this section.

It is possible to use an existing DNS setup; however, the following table lists the steps for a local setup.

1. Navigate to `/var/simp/rsync/OSTYPE/MAJORRELEASE/bind_dns`
2. Modify the named files to correctly reflect the environment. At a minimum, the following files under `/srv/rsync/bind_dns/default` should be edited:

- `named/etc/named.conf`
- `named/etc/zones/your.domain`
- `named/var/named/forward/your.domain.db`
- `named/var/named/reverse/0.0.10.db`

Important: For the `named/var/named/forward/your.domain.db` and `named/var/named/reverse/0.0.10.db` files, add clients as needed. Make sure to rename both of these files to more appropriately match your system configuration.

- **At a minimum, review `named/etc/named.conf` and check/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

1. Type `puppet agent -t --tags named` on the Puppet Master to apply the changes. Validate DNS and ensure the `/etc/resolv.conf` is updated appropriately
2. If an error about the rndc.key appears when starting bind, copy the `rndc.key` to `/etc` then re-run the puppet command: `cp -p /var/named/chroot/etc/rndc.key /etc/rndc.key`

2.3.4 Configure DHCP

Perform the following actions as `root` on the Puppet Master system prior to attempting to install a client.

Open the `/var/simp/rsync/OSTYPE/MAJORRELEASE/dhcpd/dhcpd.conf` file and edit it to suit the necessary environment.

Make sure the following is done in the `dhcpd.conf` :

- 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 **router** and **netmask** are correct for your environment.
  - Enter the hardware ethernet and fixed-address for each client that will be kickstarted. SIMP environments should 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**

Save and close the file.

Run `puppet agent -t` on the Puppet Master to apply the changes.

2.3.5 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 locate 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 DHCP server setup, also required for PXE booting, is discussed in and 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 through out 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. **Locate the following files in the `/var/www/ks` directory:**

(a) `pupclient_x86_64.cfg`

(b) `diskdetect.sh`

2. **Open each of the files and follow the instructions provided within them to replace the variables. You need to know the IP A**

   (a) **`pupclient_x86_64.cfg`:** 1.) Note: #KSSERVER# should be replaced with Kickstart Server IP not Yum IP. (They are the same if you used the defaults.) 2.) In the URL line use the YUM-SERVER ip not the Kickstart server IP. (Although on a default SIMP system the YUM and kicktart server are the same server so it is not a problem.) 3.) Use the commands in the top of the file in the comments section to generate the password hashes.

   (b) `diskdetect.sh`: The `diskdetect.sh` script is responsible for detecting the first active disk and applying a disk configuration. Edit this file to meet any necessary requirements or use this file as a starting point for further work. It will work as is for most systems as long as your disk device names are in the list.

3. Type `chown root.apache /var/www/ks/*` to ensure that all files are owned by `root` and in the `apache` group.
4. 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: The URLs and locations in the file are setup 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 then 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.

Static Files

Verify the static files are in the correct location:

Type `cd /var/simp/rsync/OSTYPE/MAJORRELEASE/tftpboot` and then type `ls` to check for the existence of the `linux-install/OSTYPE-MAJORRELEASE_ARCH` directory.

OSTYPE and MAJORRELEASE under rsync are the version of the SIMP server

where OSTYPE and MAJORRELEASE under linux-install are the OS type and OS major version of the systems you will be PXE booting.

Under this directory your should find a directory named OSTYPE-MAJORRELEASE.MINORRELEASE-ARCH and a link to this directory named OSTYPE-MAJORRELEASE-ARCH.

Under OSTYPE-MAJORRELEASE.MINORRELEASE-ARCH your should find the files:

- initrd.img
- vmlinuz

If these are not there then you must 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.

Important: The link is what is used in the TFTP configuration files.

Note: If you want to be able to PXE boot different OS, then add a directory for each on and obtain the pxeboot images and copy them under the linux-install directory. SIMP only provides images for the OS for the SIMP server.

Manifest

Create a site manifest for the TFTP server on the Puppet server.

1. **Create the file `/etc/puppet/environment/simp/modules/site/manifests/tftpboot.pp`. Use the source co**

   (a) Replace KSSERVER with the IP address of Kickstart server (or the code to look up the IP Address using Hiera).

   (b) Replace OSTYPE, MAJORRELEASE and ARCH with the correct value for the systems you will be PXE booting.

   (c) MODEL NAME is usually of the form OSTYPE-MAJORRELEASE-ARCH for consistency.

```
class site::tftpboot {
  include 'tftpboot'

  tftpboot::linux_model { 'MODEL NAME':
    kernel => 'OSTYPE-MAJORRELEASE-ARCH/vmlinuz',
    initrd => 'OSTYPE-MAJORRELEASE-ARCH/initrd.img',
    ks     => "http://KSSERVER/ks/pupclient_x86_64.cfg",
    extra  => "ksdevice=bootif\nipappend 2"
  }

  tftpboot::assign_host { 'default': model => 'MODEL NAME' }
}
```

2. Add the tftpboot site manifest on your puppet server node via Hiera.

Create the file (or edit if it exists): `/etc/puppet/environments/simp/hieradata/hosts/<tftp.server.fqdn>.yam` (By default the TFTP server is the same as your puppet server o in the deault case it will exist.) Add the following example code to that yaml file.

```
---
classes:
  - 'site::tftpboot'
```

3. After updating the above file, type `puppet agent -t --tags tftpboot` on the Puppet server.

Note: To PXE boot more OSs create, in the tftpboot.pp file, a tftpboot::linux_model block for each OS type using the extra directories and kickstart files created using the notes in previous sections. Point individual systems to them by adding assign_host lines with their MAC pointing to the appropriate model name.

2.3.6 Setting Up the Client

The following lists the steps to *PXE* boot the system and set up the client.

1. Set up your client's *BIOS* or virtual settings to boot off 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. Puppet will not autosign puppet certificates by default and waitforcert is enabled. The client will check in every 30 seconds for a signed cert. Log on to the puppet server and run `puppet cert sign <puppet.client.fqdn>`.

Upon successful deployment of a new client, it is highly recommended that *LDAP administrative accounts* be created.

2.3.7 Troubleshooting Issues

If the client has been kickstarted, but is not communicating with the Puppet server, try the following options:

- Check the forward and reverse *DNS* entries on the client and server; both must be correct.
- Check the time on the systems. More than an hour's difference will cause serious issues with certificates.
- Remove `/var/lib/puppet/ssl` on the client system; run `puppet cert --clean ***<Client Host Name>***` on the Puppet server; and try again.

2.3.8 Troubleshoot Certificate Issues

If host certificates do not appear to be working and the banner is not getting rsync'd to the clients, 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 `/etc/puppet/environments/simp/keydist`
2. Run `find . -name "****<Your.Domain>*.pub" -exec openssl verify -CApath cacerts {} \;`

Important: The screen displays `./<Host Name>.<Your.Domain>/<Host Name>.<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.

1. Navigate to `/etc/puppet/environments/simp/keydist;`
2. Run `OPENSSL_CONF=default.cnf openssl ca -revoke ../../keydist/***<Host to Revoke>*/*<Host to Revoke>*.pub**`

2.4 Apply Certificates

This section provides guidance on obtaining official certificates and generating a Fake CA.

2.4.1 Obtaining Official Certificates

All SIMP systems must have *Public Key Infrastructure* (PKI) keypairs generated for the server.

These keys reside in the `/etc/puppet/environments/simp/modules/pki/files/keydist` directory and are served to the clients over the puppet protocol.

Note: These keypairs are not the keys that the Puppet server uses for its operation. Do not get the two confused.

The table below lists the steps to add any keys for the server that were received from a proper CA to `/etc/puppet/environments/simp/modules/pki/files/keydist`.

1. Type `mkdir /etc/puppet/environments/simp/modules/pki/files/keydist/***<Client System FQDN>***`
2. Type `mv ***<Certificate Directory>***/***<FQDN>***.[pem|pub] /etc/puppet/environments/simp/modules/pki/files/keydist/***<FQDN>***`
3. Type `chown -R root.puppet /etc/puppet/environments/simp/modules/pki/files/keydist`
4. Type `chmod -R u=rwX,g=rX,o-rwx /etc/puppet/environments/simp/modules/pki/files/keydist`

Table: Official Certificates Procedure

The table below lists the steps to create and populate the `/etc/puppet/environments/simp/modules/pki/files/keydis` directory.

1. Type `cd /etc/puppet/keydist`
2. Type `mkdir cacerts` and copy the root CA public certificates into cacerts in Privacy Enhanced Mail (PEM) format (one per file).
3. Type `cd cacerts`
4. Type `for file in *.pem; do ln -s $file 'openssl x509 -in $file -hash -noout'.0; done`

Table: `/etc/puppet/environments/simp/modules/pki/files/keydist/cacerts` Directory Creation Procedure

1. Type `cd /etc/puppet/keydist`
2. Type `mkdir cacerts` and copy the root CA public certificates into `cacerts` in *Privacy Enhanced Mail* (PEM) format (one per file).
3. Type `cd cacerts`
4. Type `for file in *.pem; do ln -s $file 'openssl x509 -in $file -hash -noout'.0; done`

Table: `/etc/puppet/keydist/cacerts` Directory Creation Procedure

2.4.2 Generating Fake CAs

If server certificates have not or could not be obtained at the time of client installation, the SIMP team provides a way to create them for the system so that it will work until proper certificates are provided.

Note: This option should not be used for any operational system that can use proper enterprise PKI certificates.

The instructions below lists the steps to generate the Fake CAs.

1. Type `cd /etc/puppet/environments/simp/FakeCA`

2. Type `vi togen`
3. 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 without spaces. For example, `.name,alt.name1,alt.name2.`

4. Type `wc cacertkey`

   Note: Ensure that the `cacertkey` file is not empty. If it is, enter text into the file; then save and close the file.

5. Type `./gencerts_nopass.sh auto`

   Note: To avoid using the default Fake CA values, remove the `auto` statement from the `./gencerts_nopass.sh` command.

Table: Generating Fake CAs Procedure

Warning: If the `clean.sh` command is run after the certificates have been generated, the running system will break. To troubleshoot certificate problems, see the section at the end of this chapter.

If issues arise while generating keys, type `cd /etc/puppet/environments/simp/FakeCA` to navigate to the `/etc/puppet/environments/simp/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.

2.5 Hiera Overview

SIMP now uses Hiera natively instead of Extdata. From Puppet Labs website: Hiera is a key/value lookup tool for configuration data, built to set node-specific data without repeating yourself. It is an attempt to make SIMP more configurable to you, the end user. It configures Puppet in two ways: automatic parameter lookup/hiera lookup functions, and assigning classes to nodes. The former allows you to generate reusable code and concentrates parameter assignment to one directory. The latter is a supplement to the failed inheritance model.

2.5.1 Setting Parameters

Automatic Lookup You can now safely declare any class on any node with 'include', even if the class is parametized. Before Hiera, this was not possible. Puppet will automatically retrieve class parameters from Hiera using keys. Add a key with a value pair to an appropriate yaml file, say default.yaml, as such:

Adding a Key/Value Pair to Hiera Examples

```
---
classfoo::parameter_bar: "Woo"
classfoo::parameter_baz: "Hoo"
```

You can then 'include classfoo' on any node, with parameter_bar and parameter_baz defaulting to Woo and Hoo, respectively.

Lookup Functions You are not required to set up your hierarchy for automatic variable lookup. Using three functionts, you can query Hiera for any key.

The first is `hiera`. This uses standard priority lookup and can retrieve values of any data type from Hiera. If no key is found, a default should be included. `$myvar = hiera('parameter_bar', 'Woo')`

The second is `hiera_array`. This uses an array merge lookup. It retrieves all array values for a given key througout the entire hierarchy and flattens them into a single array.

The third is `hiera_hash`. This uses a hash merge lookup. It retrieves all hash values for a given key throughout the entire hierarchy and merges them into a single hash.

2.5.2 Assigning Classes to Nodes

Assigning classes to nodes is done with the `hiera_include` function. Hiera does an array merge lookup on 'tags' to retrieve classes which should be included on a node. In SIMP, we place `hiera_include('classes')` in `/etc/puppet/manifests/site.pp`. Since site.pp is outside of any node definition and below all top scope variables, every node controlled by puppet will get every class tagged with 'classes' **in its hierarchy**. Additionally, simp_def.yaml in is the hierarchy of every node, so every node will receive those classes (by default).

2.5.3 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, make a site class, `/etc/puppet/modules/site/manifests/my_site.pp`. For example, to include tftpboot linux_model and assign_host on your puppet server, puppet.your.domain:

Adding a Site Manifest Examples

```
# in /etc/puppet/environments/simp/modules/site/manifests/tftpboot.pp
# Set KSSERVER statically or use Hiera for lookup

class site::tftpboot {
  include 'tftpboot'

  tftpboot::linux_model { 'CentOS_RHEL_MAJOR_VERSION':
    kernel => 'centosRHEL_MAJOR_VERSION_x86_64/vmlinuz',
    initrd => 'centosRHEL_MAJOR_VERSION_x86_64/initrd.img',
    ks     => "http://KSSERVER/ks/pupclient_x86_64.cfg",
    extra  => 'ipappend 2'
  }

  tftpboot::assign_host { 'default': model => 'CentOS_RHEL_MAJOR_VERSION' }
}
```

Then, in `/etc/puppetenvironments/simp/hieradata/hosts/puppet.your.domain.yaml`

Adding TFTP Site to Hiera Examples

```
---
classes:
  - 'site::tftpboot'
```

2.5.4 SIMP Hiera File Structure

- `/etc/puppet/hiera.yaml` Hiera's config file, used to control the hierarchy of your backends.
- `/etc/puppet/environments/simp/hieradata/` Default location of the yaml files which contain your node data

- `/etc/puppet/environments/simp/hieradata/simp_classes.yaml` The list of default classes to include on any SIMP system.
- `/etc/puppet/environments/simp/hieradata/simp_def.yaml` Contains the variables needed to configure a working SIMP system. Modified by simp-config.
- `/etc/puppet/environments/simp/hieradata/hosts/` By populating this directory with some.host.name.yaml file, you can assign parameters to host some.host.name
- `/etc/puppet/environments/simp/hieradata/domains/` Same principal as hosts, but domain names.
- `/etc/puppet/manifests/` Contains site.pp and all other node manifests. BE CAREFUL when modifying this directory, site.pp contains your globals. This directory can be used to supplement or even REPLACE Hiera, with nodes. Note that Hiera cannot regex hostnames to apply manifests, so a node manifest will have to be created here if you wish to have that ability.

2.6 SIMP 5.1.0-0

2.6.1 Changelog

Contents

SIMP 5.1.0-0

Package: 5.1.0-0

This release is known to work with:

- RHEL 7.0 and 7.1 x86_64
- CentOS 7.0 x86_64 (1406 and 1503)

Warning: The default system passwords have changed! Please see the User's Guide for details.

Manual Changes Requred

- Bugs in the *simplib::secure_mountpoints* class (formerly *common::secure_mountpoints*)

Note: This only affects you if you did not have a separate partition for /tmp!

- There were issues in the secure_mountpoints class that caused /tmp and /var/tmp to be mounted against the root filesystem. While the new code addresses this, it cannot determine if your system has been modified incorrectly in the past.
- To fix the issue, you need to do the following: - Unmount /var/tmp (may take multiple unmounts) - Unmount /tmp (may take multiple unmounts) - Remove the 'bind' entries for /tmp and /var/tmp from /etc/fstab - Run **puppet** with the new code in place

Deprecations

- simp-hiera

  The *simp-hiera* RPM has been replaced by the upstream *hiera* package from Puppet Labs. The original simp-hiera fork had been maintained due to a need that the 'alias()' function now serves. Please run the *hiera_upgrade* script to convert your existing SIMP environment. You may also set the environment variable *HIERA_UPGRADE* to a path of your choice to update any other hieradata that you may have on your system.

- pupmod-simp-common

  The *::common* namespace has been deprecated in favor of the new *::simplib* namespace. This removes a commonly conflicting module name from the SIMP ecosystem.

  You will need to run the *migrate_to_simplib* script to update all of the relevant files. This script will only migrate items in the existing SIMP environment. You may also set the environment variable *UPGRADE_PATHS* to run the script on multiple external paths.

  All code was migrated.

- pupmod-simp-functions

  The *::functions* namespace has been deprecated in favor of the new *::simplib* namespace. This removes a commonly conflicting module name from the SIMP ecosystem.

  You will need to run the *migrate_to_simplib* script to update all of the relevant files. This script will only migrate items in the existing SIMP environment. You may also set the environment variable *UPGRADE_PATHS* to run the script on multiple external paths.

  The following items were not migrated:

  - append_if_no_such_line => Use simp_file_line{}
  - delete_lines => Use augeas{}
  - init_mod_nice => Use init_ulimit{}
  - init_mod_open_files => Use init_ulimit{}
  - line => Use augeas{}
  - prepend_if_no_such_line => Use simp_file_line{}
  - renice => No replacement, was not correct
  - replace_line => Use augeas{}

Significant Updates

- FIPS Mode is now enabled by default!

  - This is a SIGNIFICANT change and may impact many of your running applications that use encryption.
  - If you are upgrading, do **NOT** enable FIPS mode without extensive testing as it may cause various applications to not function properly any longer.
- The rsyslog module has been completely rewritten to support rsyslog 7.4. This is a breaking change from previous releases and will require active updates to existing systems. All modules with rsyslog integration ave been updated to accommodate this change:
  - Critical Variable Changes
    - The global *rsyslog::log_server_list* variable is now set to send to **all** of the servers in the Array by default.
      - This variable defaults to the global *log_servers* Array in Hiera.
    - There is a new variable *rsyslog::failover_log_servers* which is an Array of failover log servers to be used for your system. These will be tried, in order, until successful messages can be sent.
  - Updated Modules:
    - aide
    - apache
    - auditd
    - dhcp
    - logstash
    - openldap
    - rsync
    - simp
    - sudosh
- There was a bug in previous versions of SIMP that require the following LDIF to be run manually on the systems to correct the password policy checking.

  dn: cn=default,ou=pwpolicies,dc=your,dc=domain changetype: modify replace: pwdCheckModule pwdCheckModule: simp_check_password.so - dn: cn=noExpire_noLockout,ou=pwpolicies,dc=your,dc=domain changetype: modify replace: pwdCheckModule pwdCheckModule: simp_check_password.so
- The Electrical and SIMP modules for elasticsearch have been combined.

Upgrade Guidance

Fully detailed upgrade guidance can be found in the **Upgrading SIMP** portion of the *User's Guide*.

Warning: You must have at least **2.2GB** of **free** RAM on your system to upgrade to this release.

Note: Upgrading from releases older than 5.0 is not supported.

Expectations

Before you begin, please be aware that the following actions will take place as a result of the migrate_to_environments script:

- The *puppet-server* RPM will be removed
- The *puppetserver* RPM will be installed (no, that's not a typo)
- **ALL** SIMP Puppet code will be migrated into a new *simp* environment
  - This will be located at */etc/puppet/environments/simp*
- A backup of your running environment will be made available at */etc/puppet/environments/pre_migration.simp*
  - You will find timestamped directories under the *pre_migration.simp* directory that correspond to runs of the migration script
  - Your old files will be in a *backup_data* directory and will be linked to a local bare Git repository in the same space

The upgrade steps will also have you install PuppetDB. PuppetDB is installed by default if you kick from the DVD.

Security Announcements

CVEs Addressed

RPM Updates

Numerous RPMs were updated in the creation of this release. Several were included due to our use of *repoclosure* to ensure that RPM dependencies are met when releasing a DVD.

- This version include the latest RedHat 7.1 and CentOS 7.0 (1503) RPMs.
- Facter upgraded to 2.4.
- PuppetDB upgraded to 2.3.8-1

Fixed Bugs

- pupmod-aide
  - Change the call to the *rsyslog* init script to the *service* command to seamlessly support both RHEL6 and RHEL7.
- pupmod-apache
  - Removed all reliance on 'lsb*' facts since some environments do now wish to install the prerequisites for those facts to run.
  - Remove the apache_version fact and simply use the version controls built into the Apache configuration language.
  - Update all custom functions to properly scope definitions.
  - Ensure that mod_ldap is installed in SIMP >= 5.0.
  - Prevent apache from restarting after downloading a CRL.
- pupmod-clamav
  - Change the call to the *rsyslog* init script to the *service* command to seamlessly support both RHEL6 and RHEL7.
- pupmod-common => Deprecated - Replaced by pupmod-simplib!
- pupmod-simplib

  - Fixed the secure_mountpoints code so that it no longer incorrectly bind mounts /tmp or /var/tmp.
  - We no longer supply crontab or anacrontab in global_etcd.
  - Remove dynamic_swappiness cron job if a static value is set.
  - Ensure that the *passgen()* function fails on invalid scenarios. This prevents the accidental cration of empty passwords.
  - Allow the value *2* to be used for `rp_filter` in `simplib::sysctl`.
  - Added ability to return remote ip addrs.
- pupmod-dhcp
  - Change the call to the *rsyslog* init script to the *service* command to seamlessly support both RHEL6 and RHEL7.
- pupmod-elasticsearch
  - Ensured that Elasticsearch works properly with the new version of Apache.
  - Removed our default ES tuning since the default works better for LogStash.
  - Ensure that Puppet manages the Elasticsearch logging file.
- pupmod-functions
  - Fixed sysv.rb to explicitly require puppet/util/selinux, which caused puppet describe to have errors.
- pupmod-iptables
  - Fixed a bug that would cause issues with Ruby 1.8.7.
  - Fixed DNS resolution in IPv6.
  - Prevent IPv6 ::1 spoofed addresses by default.
- pupmod-simp-logstash
  - Fix issues with both TCPWrappers and IPTables when used with LogStash.
- pupmod-nfs
  - Updated the *mountd* port to be *20048* by default for SELinux issues in RHEL7.
- pupmod-ntp
  - Updated against NTP Security Vulnerabilities (Red Hat Article #1305723).
  - Ensure that *restrict* entries use DDQ format.
- pupmod-openldap
  - The Password Policy overlay was getting loaded into the default.ldif even if you didn't want to use it. This has been fixed.
  - Made the password policy overlay align with the latest SIMP build of the plugin.
    * This means that you *must* have version simp-ppolicy-check-password-2.4.39-0 or later available to the system being configured.
  - Change the call to the *rsyslog* init script to the *service* command to seamlessly support both RHEL6 and RHEL7.
  - Fixed reported bugs in syncrepl.pp.
  - Removed all reliance on the 'lsb*' facts since some users do not wish to install the prerequisite RPMs for LSB compliance.

- pupmod-openscap
  - Change the call to the *rsyslog* init script to the *service* command to seamlessly support both RHEL6 and RHEL7.
  - Changed default ssg base path to /usr/share/xml/scap/ssg/content
- pupmod-pam
  - Removed all reliance on the 'lsb*' facts since some users do not wish to install the prerequisite RPMs for LSB compliance.
- pupmod-pki
  - Now allow directories in the cacerts directories. This previously caused failures that needed to be manually addressed on each node.
- pupmod-rsync
  - Fixed provider to run with –dry-run when puppet is run with a –noop.
- pupmod-simp
  - Ensure that SSSD is used by default on EL7+ systems since nscd and nslcd have functionality issues.
  - Removed all reliance on the 'lsb*' facts since some users do not wish to install the prerequisite RPMs for LSB compliance.
- pupmod-ssh
  - Modernized the Ciphers, MACs, and Kex.
  - Added explicit cases for FIPS and non-FIPS mode (as well as reasonable default cases for RHEL7 and below).
  - Updated to use the new augeasproviders module dependencies.
  - Added a function *ssh_format_host_entry_for_sorting()* that will properly sort SSH *Host* entries for inclusion with concat.
- pupmod-stunnel
  - Had a variable **options** in *stunnel.erb* that should have been scoped as **@options**.
- pupmod-sudo
  - Removed all reliance on the 'lsb*' facts since some users do not wish to install the prerequisite RPMs for LSB compliance.
- pupmod-sudosh
  - Change the call to the *rsyslog* init script to the *service* command to seamlessly support both RHEL6 and RHEL7.
- pupmod-sysctl
  - Removed support for the old parsed-file provider and moved to using the new Augeas-based provider.
- pupmod-tftpboot
  - Purging of non-Puppet-managed items in *pxelinux.cfg* is now optional.
- pupmod-simp-tpm
  - IMA is disabled by default.
- simp-gpgkeys
  - Ensure that the keys are set in the correct locations for the target SIMP distribution.

- simp-rsync
  - Removed spurious install messages.
- simp-util
  - Fixed the targets of unpack_dvd.
  - Added a **use_fips** boolean to *simp config*
- pupmod-xinetd
  - Fixed: The default log_type should be 'SYSLOG authpriv' instead of 'SYSLOG daemon info'.
- pupmod-vnc
  - Removed banners that broke some vnc clients.
- simp-cli
  - *simp config -a ANSWERFILE* fails when an item has no answer
  - *simp config -A ANSWERFILE* prompts when an an item has no answer
  - The misleading *–help* documentation for *-ff* has been removed
  - The Config::Item *use_fips* now echoes its command unless *@silent*
  - The *simp doc* command path to the documentation has been corrected.
  - General usability improvements.
- DVD
  - NetworkManager-wait-online is now set by default in the ISO supplied kickstart images. Without ths, it is possible for the 'runpppet' script to attempt to run prior to the network being initialized.
  - A default IP is no longer provided when booting from the ISO; *simp config* will set the network properly.
  - The default kickstart no longer attempts to chkconfig any services in the %post section.

New Features

- pupmod-auditd
  - Completely overhauled the module with a focus on better acceptance testing and format compliance.
- pupmod-augeasproviders
  - This was updated to 2.1.3.
  - The update to 2.1.3 caused the addition of all of the pupmod-augeasproviders modules below.
- augeasproviders_apache
  - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_base
  - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_core
  - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_grub
  - Imported 2.1.3 to support the Augeasproviders stack.

- augeasproviders_mounttab
  - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_nagios
  - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_pam
  - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_postgresql
  - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_puppet
  - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_shellvar
  - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_ssh
  - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_sysctl
  - Imported 2.1.3 to support the Augeasproviders stack.
- pupmod-augeasproviders
  - This was updated to 2.1.3.
  - The update to 2.1.3 caused the addition of all of the pupmod-augeasproviders modules below.
- pupmod-cgroups
  - Added acceptance tests.
- pupmod-common => Deprecated - Replaced by pupmod-simplib!
- pupmod-simplib
  - Created parse_hosts function.
  - Added full tests for evaluating the ability to toggle FIPS mode.
- pupmod-richardc-datacat
  - Incorporated the *richardc/datacat* module into the core for user convenience.
- pupmod-freeradius
  - Split the Freeradius module based on version so that it can be properly selected against the *installed* version of Freeradius. This may take two runs to coalesce.
- pupmod-puppetlabs-inifile
  - Updated to version 1.2.0.
- pupmod-puppetlabs-puppetdb
  - Updated to version 5.0.0-0.
- pupmod-simp-kibana
  - Add Kibana dashboards to the Kibana module.

  - Allows users to apply default SIMP kibana Dashboards.
- pupmod-simp-logstash
  - Integrated SIMP and Electrical Logstash modules.
  - Changes the existing Logstash module to allow users to apply default SIMP filters.
- pupmod-pki
  - Now generate a system RSA public key against the passed private key.
- pupmod-puppetlabs-postgresql
  - Initial import of the Puppet Labs PostgreSQL module.
  - Modifications were made to support the SIMP concat.
- pupmod-puppetlabs-puppetdb
  - New import of the Puppet Labs PuppetDB module.
- pupmod-simp-rsyslog
  - Module has been rewritten to support rsyslog 7.4.
- pupmod-simp-simp
  - Set the SELinux Boolean 'use_nfs_home_dirs' to 'on' if a remote NFS server is used for home directories.
  - The 'runpuppet' script was modified to run 'fixfiles' on systems prior to the final puppet runs since RHEL7, in some cases, does not appear to honor the '/.autorelabel' file.
- pupmod-puppetlabs-stdlib
  - Updated to version 4.5.1.
- pupmod-sysctl
  - Moved the configuration file updates from sysctl.conf to sysctl.d/20-simp.conf to use the latest update mechanisms.
- pupmod-tftpboot
  - Updated to use native packages and pull as muchs possible.
- simp-doc
  - Updated tables across the board to be more readable.
  - Updated documentation relating to user management and user key management using SSH.
  - Rebranded the documentation and updated the color scheme.
  - Updated the default system passwords.
- simp-rsync
  - Content has been restructured to eliminate licensing conflicts.
  - ClamAV has been refactored into a separate (GPL) package.
- simp-utils
  - simp config was rewritten to allow for new features and flexibility.
  - Now provided as a Ruby gem "simp-cli".
- Mcollective

  - Mcollective is now available to be installed and used with SIMP. It uses SSL/TLS along with user certificates for proper encryption and authentication.

- PuppetDB
  - PuppetDB is now supported by SIMP and installed by default.
- Puppetserver
  - The puppet master service has been replaced by the puppetserver service. This is a major rewrite by Puppetlabs. Puppetserver scales better for larger agent deployments with a single puppet master.
  - Uses Environments by default, this allows for tools such as r10K. Production environment is a link to simp by default.
- Facter 2.4
  - Facter now returns the following facts as their actual boolean or integer values, instead of converting them into strings:

    activeprocessorcount is_virtual mtu_<INTERFACE> physicalprocessorcount processorcount selinux_enforced selinux sp_number_processors sp_packages

Known Bugs

- There is a symlink that is created at /etc/puppet/environments/simp/simp which should not be in place. This is being tracked as SIMP-661
- SSSD is currently broken and will allow logins via SSH even if your password has expired. This has been noted by Red Hat and is in the pipeline.
- If you are running libvirtd, when svckill runs it will always attempt to kill dnsmasq unless you are deliberately trying to run the dnsmasq service. This does *not* actually kill the service but is, instead, an error of the startup script and causes no damage to your system.

2.7 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 reproducability.

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.

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.

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.

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

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.

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

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

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.

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

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 standard of note in SIMP is FIPS 140-2

Source: FIPS Publications

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.

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 configuration data, built to make *Puppet* better and let you set node-specific data without repeating yourself.

Source: Hiera Overview

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.

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.

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.

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.

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

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.

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.

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.

PSSH, Parallel Secure Shell A tool that provides parallel versions of OpenSSH and other related tools.

Puppet An Open Source configuration management tool written and maintained by Puppet Labs. 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.

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.

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 *RHEL*.

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.

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.

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.

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*.

SSH, Secure Shell An application for secure data communication, remote shell services, or command execution between networked computers. SSH utilitizes 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*.

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.

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.

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*.

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".

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.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

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*.

2.8 Installation_Miscellaney

This sections provides a list of variables that are configurable during the install.

2.8.1 List of Installation Variables

Description
Enable FIPS-140-2 compliance
Do you want to set up network interface - use DHCP or Static for NIC - Hostname of server - IP Address of server - Netmask - Defaul
Your DNS server
The search domain for DNS.
Subnet used for clients managed by the puppet server
NTP servers.
IP addr of primary log server (rsyslog)
IP address of failover log server.
Yum server for simp modules.
Turn on the audit deamon?
Turn on iptable deamon?
The default system run level
Do you want to set SELINUX to enforcing?
Set a grub password on the puppet server?
Make puppet server the master yum server?
The FQDN of the puppet server.
Puppet servers IP address.
FQDN of Puppet Certificate Authority (CA)
The port Puppet CA will listen on.
The DNS name of puppet database server.
The port used by the puppet database server
Do you want to use LDAP?
LDAP Server Base Distinquish Name (DN)
The LDAP Bind Distiquished name.
LDAP Bind password
LDAP Sync Distiquished name.
LDAP Sync password
The LDAP root DN.
LDAP root password This password is used for manually updating LDAP, you will want to set it your self.
The URI for your LDAP server.
The directory that will hold files used to sync oprational directories
The server that remote syncs
Maximum rsync timeout in seconds

2.8.2 Configuration

This briefly describes what is being configured in the different sections indicated in the table above.

You may make changes to the default settings in ‘‘ *puppet config print environment-path*/simp/hieradata/simp_def.yaml‘ ‘‘ or one of the other yaml files in the hieradata directory.

These *Hiera* files can be used after initial set up to change settings. The *Hiera Overview* section gives an introduction of using Hiera in SIMP.

FIPS

- Turning on and off *FIPS* mode sets kernel parameters and systems environment variables to ensure the system is FIPS 140-2 compliant.
- FIPS is on by default. If you ever want to have your system to beFIPS compliant, you will want to ensure that the system is built with this enabled. It may easily be disabled once the system is built.

GRUB

- Grub password in `/etc/grub2.cfg`

DNS

- The /etc/resolv.conf
- The *DNS* server capabilities are not configured by this.

SYSTEM

- Basic network setup.
- Startup files in /etc/init.d.
- Configuration files under /etc/sysconfig.
- Rsyslog settings.

PUPPET

- Autosigning in `*/etc/puppet/autosign.conf`
- File Serving in `*/etc/puppet/fileserver.conf`
- Puppet server and *Certificate Authority* (CA) information in `/etc/puppet/puppet.conf`
- Server certificates for the puppet host (Fake CA)

LDAP

- If you select use_ldap and set this server as your *LDAP* server, OpenLDAP Puppet will enable the LDAP service on this server and all clients will be set to reference it for authentication.
- If you select use_ldap and set another server as your LDAP server, then the clients (including this server) will use the specified server instead.
- If you choose not to use LDAP the system is set up to use traditional local authentication only.

RSYNC

- The puppet server is configured to rsync data directories for services like *DNS*, *DHCP* or *TFTP*.

YUM

- Base *YUM* repositories for *RPM* updates.

2.9 Indices and tables

- genindex
- search

CHAPTER 3

SIMP User Guide

Contents:

3.1 Introduction

This guide will walk a user through the process of managing a *SIMP* system. This system includes, at a minimum, a SIMP server with properly configured networking information and a working Puppet server. Additionally, this document outlines the process of managing clients and users associated with the SIMP system.

3.1.1 Level of Knowledge

SIMP is designed for use by system administrators or users with a strong background using Linux operating systems. The core applications that make up SIMP and require prerequisite knowledge are:

- *Puppet* - 3.7 or later
- *Domain Name System* (DNS) - BIND 9
- *Dynamic Host Configuration Protocol* (DHCP) - Internet Systems Consortium (ISC) DHCP
- *Lightweight Directory Access Protocol* (LDAP) - OpenLDAP
- RedHat Kickstart (including all tools behind it) - *Trivial File Transfer Protocol* (TFTP), PXELinux, etc.
- Apache
- *Yellowdog Updater, Modified* (YUM)
- Rsyslog Version 3+
- *Internet Protocol Tables* (IPtables) (Basic knowledge of the rules)
- *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 does as much initial setup and configuration of these tools as possible. However, without at least some understanding, you will be unable to tailor a SIMP system to fit the desired environment. A general understanding of how to control and manipulate these tools from the *command line interface* (CLI) will be necessary, as SIMP does not come stock with 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.

3.1.2 SIMP Defined

The System Integrity Management Platform (SIMP) is a 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 Guides.

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.

3.2 User Management

This chapter explains how to manage users in the default SIMP environment.

3.2.1 Managing Users with Lightweight Directory Access Protocol (LDAP)

SIMP natively uses OpenLDAP for user and group management. Actionable copies of the *LDAP* Data Interchange Format (.ldif) files can be found on the system in the `/usr/share/doc/simp-doc-<Version>/ldifs` directory.

Users cannot have any extraneous spaces in .ldif files.

```
# Use `:set list` in vim to see hidden spaces at the end of lines.

# Use the following to strip out inappropriate characters

sed -i \
  's/\\(^[[:graph:]]\*:\\)[[:space:]]\*\\ ([[:graph:]]\*\\) \\[[:space:]]\*$/\\1\\2/' \
  file.ldif
```

Note: Use the `[` and `]` characters to scroll right when using ELinks.

Add Users

Users can be added with or without a password. Follow the instructions in the following sections.

Warning: This process should not be used to create users or groups for daemon processes unless the user has experience.

Adding Users With a Password

To add a user to the system, *Secure Shell* (SSH) to the LDAP server and use the `slappasswd` command to generate a password hash for a user.

Create a `/root/ldifs` directory and add the following information to the `/root/ldifs/adduser.ldif` file. Replace the information within < > with the installed system's information.

Example ldif to add a user

```
dn: uid=<User UID>,ou=People,dc=your,dc=domain
uid: <User UID>
cn: <User UID>
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
objectClass: ldapPublicKey
shadowMax: 90
shadowMin: 1
shadowWarning: 7
shadowLastChange: 10167
pwdReset: TRUE
sshPublicKey: <User SSH Public Key>
loginShell: /bin/bash
uidNumber: <User UID Number>
gidNumber: <User Primary GID>
homeDirectory: /home/<User UID>
userPassword: <Password Hash from slappasswd>
```

Type:

```
`ldapadd -Z -x -W -D "cn=LDAPAdmin,ou=People,dc=your,dc=domain" \
-f /root/ldifs/adduser.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). Members of this LDAP group can utilize sudo sudosh for privilege escalation.

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.

Adding Users Without a Password

Create a `/root/ldifs` directory and add the following information to the `/root/ldifs/adduser.ldif` file. Replace the information within < > with the installed system's information.

Example ldif example to add a user

```
dn: uid=<User UID>,ou=People,dc=your,dc=domain
uid: <User UID>
cn: <User UID>
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
objectClass: ldapPublicKey
sshPublicKey: <User SSH Public Key>
loginShell: /bin/bash
uidNumber: <User UID Number>
```

```
gidNumber: <User Primary GID>
homeDirectory: /home/<User UID>
```

Type:

```
ldapadd -Z -x -W -D "cn=LDAPAdmin,ou=People,dc=your,dc=domain" \
 -f /root/ldifs/adduser.ldif
```

Remove Users

To remove a user, create a `/root/ldifs/removeuser.ldif` file. Add the information below to the file and replace the text within < > with the installed system's information.

Example ldif to remove a user

```
dn: cn=<User UID>,ou=Group,dc=example,dc=domain
changeType: delete

dn: uid=<User UID>,ou=People,dc=example,dc=domain
changeType: delete
```

Type:

```
ldapmodify -Z -x -W -D "cn=LDAPAdmin,ou=People,dc=your,dc=domain" \
-f /root/ldifs/removeuser.ldif
```

Additional .ldif File Commands

Other useful commands for .ldif files can be found below. Before using these commands, ensure that the `/root/ldifs` directory has been created.

Changing a Password

To change a password, add the following information to the `/root/ldifs/<.ldif File>` file. Replace the information below within < > with the installed system's information.

Example ldif to change password

```
dn: uid=<User UID>,ou=People,dc=your,dc=domain
changetype: modify
replace: userPassword
userPassword: <Hash from slappasswd>
```

Type:

```
ldapmodify -Z -x -W -D "cn=LDAPAdmin,ou=People,dc=your,dc=domain" \
-f <.ldif_file>
```

Adding a Group

To add a group, add the following information to the `/root/ldifs/<.ldif File>` file. Replace the information below within < > with the installed system's information.

Example ldif to add a group

```
dn: cn=<Group Name>,ou=Group,dc=your,dc=domain
objectClass: posixGroup
objectClass: top
cn: <Group Name>
gidNumber: <GID>
description: "Some Descriptive Text"
```

Type:

```
ldapadd -Z -x -W -D "cn=LDAPAdmin,ou=People,dc=your,dc=domain" \
-f <.ldif_file>
```

Removing a Group

To remove a group, add the following information to the `/root/ldifs/<.ldif File>` file. Replace the information below within < > with the installed system's information.

Example ldif to remove a group

```
dn: cn=<Group Name>,ou=Group,dc=your,dc=domain
changetype: delete
```

Type:

```
ldapmodify -Z -x -W -D "cn=LDAPAdmin,ou=People,dc=your,dc=domain" \
-f <.ldif_file>
```

Adding Users to a Group

To add users to a group, add the following information to the `/root/ldifs/<.ldif File>` file. Replace the information below within < > with the installed system's information.

Example ldif to add to a group

```
dn: cn=<Group Name>,ou=Group,dc=your,dc=domain
changetype: modify
add: memberUid
memberUid: <UID1>
memberUid: <UID2>
...
memberUid: <UIDX>
```

Type:

```
ldapmodify -Z -x -W -D "cn=LDAPAdmin,ou=People,dc=your,dc=domain" \
-f <.ldif_file>
```

Removing Users from a Group

To remove users from a group, add the following information to the `/root/ldifs/<.ldif File>` file. Replace the information below within < > with the installed system's information.

Example ldif to remove a user from a group

```
dn: cn=<Group Name>,ou=Group,dc=your,dc=domain
changetype: modify
delete: memberUid
memberUid: <UID1>
memberUid: <UID2>
...
memberUid: <UIDX>
```

Type:

```
ldapmodify -Z -x -W -D "cn=LDAPAdmin,ou=People,dc=your,dc=domain" \
-f <.ldif_file>
```

Updating an SSH Public Key

To update an SSH public key, add the following information to the `/root/ldifs/<.ldif File>` file. Replace the information below within < > with the installed system's information.

Example ldif to update SSH public key

```
dn: uid=<User UID>,ou=People,dc=your,dc=domain
changetype: modify
replace: sshPublicKey
sshPublicKey: <User OpenSSH Public Key>
```

Type:

```
ldapmodify -Z -x -W -D "cn=LDAPAdmin,ou=People,dc=your,dc=domain" \
-f <.ldif_file>
```

Forcing a Password Reset

To force a password reset, add the following information to the `/root/ldifs/<.ldif File>` file. Replace the information below within < > with the installed system's information.

Example ldif to reset user's shadowLastChange

```
dn: uid=<User UID>,ou=People,dc=your,dc=domain
changetype: modify
replace: pwdReset
pwdReset: TRUE
-
replace: shadowLastChange
shadowLastChange: 10000
```

Type:

```
ldapmodify -Z -x -W -D "cn=LDAPAdmin,ou=People,dc=your,dc=domain" \
-f <.ldif_file>
```

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.

Unlocking an LDAP Account

To unlock an LDAP account, add the following information to the `/root/ldifs/<.ldif File>` file. Replace the information below within < > with the installed system's information.

Example ldif to Unlock LDAP Account

```
dn: uid=<User UID>,ou=People,dc=your,dc=domain
changetype: modify
delete: pwdAccountLockedTime
```

Type:

```
ldapmodify -Z -x -W -D "cn=LDAPAdmin,ou=People,dc=your,dc=domain" \
 -f <.ldif_file>
```

Note: The `ldapmodify` command is only effective when using the *ppolicy* overlay.

Troubleshooting Issues

If a user's password is changed in LDAP or the user changes it shortly after its initial setup, the "Password too young to change" error may appear. In this situation, apply the `pwdReset : TRUE` command to the user's account as described Add Users with a Password section.

3.2.2 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 the *site* module to set up your users. The examples may easily be extrapolated into defined types if you wish but are presented as classes for simplicity.

If you are not familiar with setting up *SSH* keys, you may want to follow the relevant GitHub documentation.

Service Account

```
class site::service_account {
  include 'ssh'

  $_svc_account_user           = 'svcuser'
  $_svc_account_group          = 'svcgroup'
  $_svc_account_id             = '1777',
  $_svc_account_homedir        = "/var/local/${_svc_account_user}"

  # Since this is a service account, automatically generate an SSH key for
  # the user and store it on the Puppet master for distribution.
  $_svc_account_ssh_private_key = ssh_keygen($_svc_account_user, '2048', true)
  $_svc_account_ssh_public_key = ssh_keygen($_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   => '0700'
  }

  ssh_authorized_key { $_svc_account_user:
    type    => 'ssh-rsa',
    key     => $_svc_account_ssh_public_key,
    target  => "${_svc_account_homedir}/.ssh/authorized_keys",
    require => [
      File["${_svc_account_homedir}/.ssh"],
      User[$_svc_account_user]
    ]
  }

  file { "${_svc_account_homedir}/.ssh/id_rsa":
    mode    => '0600',
    owner   => $_svc_account_user,
    group   => $_svc_account_group,
    content => $_svc_account_ssh_private_key
  }

  file { "/etc/ssh/local_keys/${_svc_account_user}":
    owner  => 'root',
    group  => $_svc_account_group,
    mode   => '0644',
    source => "puppet:///site/ssh_autokeys/${_svc_account_user}.pub"
  }

  sudo::user_specification { $_svc_account_user:
    user_list => ["(${_svc_account_group})"],
    host_list => [$::fqdn],
    runas     => 'root',
    cmnd      => ['/bin/cat /var/log/app.log'],
    passwd    => false
  }

  # Allow this service account from everywhere
  pam::access::manage { "Allow ${_svc_account_user}":
    users   => ${_svc_account_user},
    origins => ['ALL']
  }
}
```

Local User Account

```
class site::service_account {
  include 'ssh'

  $_local_account_user           = 'localuser'
  $_local_account_group          = 'localgroup'
  $_local_account_id             = '1778',

  # You'll probably want this in /home unless you're using NFS
  $_local_account_homedir        = "/home/${_local_account_user}"

  # You'll 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'
  }

  file { "/etc/ssh/local_keys/${_local_account_user}":
    owner  => 'root',
    group  => $_local_account_group,
    mode   => '0644',
    source => "puppet:///site/ssh_autokeys/${_local_account_user}.pub"
  }

  sudo::user_specification { $_local_account_user:
    user_list => ["(${_local_account_group})"],
    host_list => [$::fqdn],
    runas     => 'root',
    cmnd      => ['/bin/cat /var/log/app.log'],
    passwd    => false
  }

  # Allow this account from everywhere
  pam::access::manage { "Allow ${_local_account_user}":
    users   => ${_local_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.

3.3 Client Management

This chapter provides guidance to install and configure SIMP clients based on the standard SIMP system installed using the SIMP DVD.

3.3.1 System Requirements

Before installing clients, the system should consist of the following minimum requirements:

- Hardware/*Virtual Machine* (VM) : Capable of running RHEL 6 or 7 ; 64-bit compatible
- RAM: 512 MB
- HDD: 5 GB

3.3.2 Configuring the Puppet Master

Perform the following actions as `root` on the Puppet Master system prior to attempting to install a client.

3.3.3 Configure DNS

Most static files are pulled over `rsync` by Puppet in this implementation for network efficiency. Specific directories of interest are noted in this section.

It is possible to use an existing DNS setup; however, the following table lists the steps for a local setup.

1. Navigate to `/var/simp/rsync/OSTYPE/MAJORRELEASE/bind_dns`
2. Modify the named files to correctly reflect the environment. At a minimum, the following files under `/srv/rsync/bind_dns/default` should be edited:

- `named/etc/named.conf`
- `named/etc/zones/your.domain`
- `named/var/named/forward/your.domain.db`
- `named/var/named/reverse/0.0.10.db`

Important: For the `named/var/named/forward/your.domain.db` and `named/var/named/reverse/0.0.10.db` files, add clients as needed. Make sure to rename both of these files to more appropriately match your system configuration.

- **At a minimum, review `named/etc/named.conf` and check/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

1. Type `puppet agent -t --tags named` on the Puppet Master to apply the changes. Validate DNS and ensure the `/etc/resolv.conf` is updated appropriately
2. If an error about the rndc.key appears when starting bind, copy the `rndc.key` to `/etc` then re-run the puppet command: `cp -p /var/named/chroot/etc/rndc.key /etc/rndc.key`

3.3.4 Configure DHCP

Perform the following actions as `root` on the Puppet Master system prior to attempting to install a client.

Open the `/var/simp/rsync/OSTYPE/MAJORRELEASE/dhcpd/dhcpd.conf` file and edit it to suit the necessary environment.

Make sure the following is done in the `dhcpd.conf` :

- 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 **router** and **netmask** are correct for your environment.
  - Enter the hardware ethernet and fixed-address for each client that will be kickstarted. SIMP environments should 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**

Save and close the file.

Run `puppet agent -t` on the Puppet Master to apply the changes.

3.3.5 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 locate 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 DHCP server setup, also required for PXE booting, is discussed in and 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 through out 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. **Locate the following files in the `/var/www/ks` directory:**
   (a) `pupclient_x86_64.cfg`
   (b) `diskdetect.sh`
2. **Open each of the files and follow the instructions provided within them to replace the variables. You need to know the IP A**
   (a) **`pupclient_x86_64.cfg`:** 1.) Note: #KSSERVER# should be replaced with Kickstart Server IP not Yum IP. (They are the same if you used the defaults.) 2.) In the URL line use the YUM-SERVER ip not the Kickstart server IP. (Although on a default SIMP system the YUM and kicktart server are the same server so it is not a problem.) 3.) Use the commands in the top of the file in the comments section to generate the password hashes.
   (b) `diskdetect.sh`: The `diskdetect.sh` script is responsible for detecting the first active disk and applying a disk configuration. Edit this file to meet any necessary requirements or use this file as a starting point for further work. It will work as is for most systems as long as your disk device names are in the list.
3. Type `chown root.apache /var/www/ks/*` to ensure that all files are owned by `root` and in the `apache` group.
4. 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: The URLs and locations in the file are setup 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 then 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.

Static Files

Verify the static files are in the correct location:

Type `cd /var/simp/rsync/OSTYPE/MAJORRELEASE/tftpboot` and then type `ls` to check for the existence of the `linux-install/OSTYPE-MAJORRELEASE_ARCH` directory.

OSTYPE and MAJORRELEASE under rsync are the version of the SIMP server

where OSTYPE and MAJORRELEASE under linux-install are the OS type and OS major version of the systems you will be PXE booting.

Under this directory your should find a directory named OSTYPE-MAJORRELEASE.MINORRELEASE-ARCH and a link to this directory named OSTYPE-MAJORRELEASE-ARCH.

Under OSTYPE-MAJORRELEASE.MINORRELEASE-ARCH your should find the files:

- initrd.img
- vmlinuz

If these are not there then you must 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.

Important: The link is what is used in the TFTP configuration files.

Note: If you want to be able to PXE boot different OS, then add a directory for each on and obtain the pxeboot images and copy them under the linux-install directory. SIMP only provides images for the OS for the SIMP server.

Manifest

Create a site manifest for the TFTP server on the Puppet server.

1. **Create the file `/etc/puppet/environment/simp/modules/site/manifests/tftpboot.pp`. Use the source co**

   (a) Replace KSSERVER with the IP address of Kickstart server (or the code to look up the IP Address using Hiera).

   (b) Replace OSTYPE, MAJORRELEASE and ARCH with the correct value for the systems you will be PXE booting.

   (c) MODEL NAME is usually of the form OSTYPE-MAJORRELEASE-ARCH for consistency.

```
class site::tftpboot {
  include 'tftpboot'

  tftpboot::linux_model { 'MODEL NAME':
    kernel => 'OSTYPE-MAJORRELEASE-ARCH/vmlinuz',
    initrd => 'OSTYPE-MAJORRELEASE-ARCH/initrd.img',
    ks     => "http://KSSERVER/ks/pupclient_x86_64.cfg",
    extra  => "ksdevice=bootif\nipappend 2"
  }

  tftpboot::assign_host { 'default': model => 'MODEL NAME' }
}
```

2. Add the tftpboot site manifest on your puppet server node via Hiera.

Create the file (or edit if it exists): `/etc/puppet/environments/simp/hieradata/hosts/<tftp.server.fqdn>.yam`
(By default the TFTP server is the same as your puppet server o in the deault case it will exist.) Add the following example code to that yaml file.

```
---
classes:
  - 'site::tftpboot'
```

3. After updating the above file, type `puppet agent -t --tags tftpboot` on the Puppet server.

Note: To PXE boot more OSs create, in the tftpboot.pp file, a tftpboot::linux_model block for each OS type using the

extra directories and kickstart files created using the notes in previous sections. Point individual systems to them by adding assign_host lines with their MAC pointing to the appropriate model name.

3.3.6 Setting Up the Client

The following lists the steps to *PXE* boot the system and set up the client.

1. Set up your client's *BIOS* or virtual settings to boot off 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. Puppet will not autosign puppet certificates by default and waitforcert is enabled. The client will check in every 30 seconds for a signed cert. Log on to the puppet server and run `puppet cert sign <puppet.client.fqdn>`.

Upon successful deployment of a new client, it is highly recommended that *LDAP administrative accounts* be created.

3.3.7 Troubleshooting Issues

If the client has been kickstarted, but is not communicating with the Puppet server, try the following options:

- Check the forward and reverse *DNS* entries on the client and server; both must be correct.
- Check the time on the systems. More than an hour's difference will cause serious issues with certificates.
- Remove `/var/lib/puppet/ssl` on the client system; run `puppet cert --clean ***<Client Host Name>***` on the Puppet server; and try again.

3.3.8 Troubleshoot Certificate Issues

If host certificates do not appear to be working and the banner is not getting rsync'd to the clients, 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 `/etc/puppet/environments/simp/keydist`
2. Run `find . -name "****<Your.Domain>*.pub" -exec openssl verify -CApath cacerts {} \;`

Important: The screen displays `./<Host Name>.<Your.Domain>/<Host Name>.<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.

1. Navigate to `/etc/puppet/environments/simp/keydist;`
2. Run `OPENSSL_CONF=default.cnf openssl ca -revoke ../../keydist/***<Host to Revoke>*/*<Host to Revoke>*.pub**`

3.4 Apply Certificates

This section provides guidance on obtaining official certificates and generating a Fake CA.

3.4.1 Obtaining Official Certificates

All SIMP systems must have *Public Key Infrastructure* (PKI) keypairs generated for the server. These keys reside in the `/etc/puppet/keydist` directory and are served to the clients over the Puppet protocol.

Note: These keypairs are not the keys that the Puppet server uses for its operation. Do not get the two confused.

The table below lists the steps to add any keys for the server that were received from a proper CA to `/etc/puppet/keydist`.

1. Type `mkdir /etc/puppet/keydist/***<Client System FQDN>***`
2. Type `mv ***<Certificate Directory>***/***<FQDN>***.[pem|pub] /etc/puppet/keydist/***<FQDN>***`
3. Type `chown -R root.puppet /etc/puppet/keydist`
4. Type `chmod -R u=rwX,g=rX,o-rwx /etc/puppet/keydist`

Table: Official Certificates Procedure

The table below lists the steps to create and populate the `/etc/puppet/keydist/cacerts` directory.

1. Type `cd /etc/puppet/keydist`
2. Type `mkdir cacerts` and copy the root CA public certificates into *cacerts* in *Privacy Enhanced Mail* (PEM) format (one per file).
3. Type `cd cacerts`
4. Type `for file in *.pem; do ln -s $file 'openssl x509 -in $file -hash -noout'.0; done`

Table: */etc/puppet/keydist/cacerts* Directory Creation Procedure

3.4.2 Generating Fake CAs

If server certificates have not or could not be obtained at the time of client installation, the SIMP team provides a way to create them for the system so that it will work until proper certificates are provided.

Note: This option should not be used for any operational system that can use proper enterprise PKI certificates.

The table below lists the steps to generate the Fake CAs.

1. Type `cd /etc/puppet/Config/FakeCA`
2. Type `vi togen`
3. 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 without spaces. For example, .name,alt.name1,alt.name2.

   ---

4. Type `wc cacertkey`

   Note: Ensure that the cacertkey file is not empty. If it is, enter text into the file; then save and close the file.

5. Type `./gencerts_nopass.sh auto`

   Note: To avoid using the default Fake CA values, remove the auto statement from the ./gencerts_nopass.sh command.

Table: Generating Fake CAs Procedure

Warning: If the `clean.sh` command is run after the certificates have been generated, the running system will break. To troubleshoot certificate problems, see the section at the end of this chapter.

If issues arise while generating keys, type `cd /etc/puppet/Config/FakeCA` to navigate to the */etc/puppet/Config/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.

3.5 Maximum Number of Nodes

The maximum number of clients reasonable per each system is dependent on many variables, including number of processors and size of memory. Although it is impossible to predict exactly how many clients a specific server may be able to handle, a simple algorithm can give the user an estimate.

Servers with different hardware have been tested at worst case scenario. This means that all of the server's clients will run Puppet at the exact same time. The most important information collected during these runs was the compile time, which shows the increase in seconds that it takes for each node to compile when another node is added. After a certain number of nodes, nodes begin to drop to compile times lower than 30 seconds. These nodes are not actually completing their Puppet runs. This data can be seen in the following graph:

3.5.1 Number of Nodes vs. Compile Time

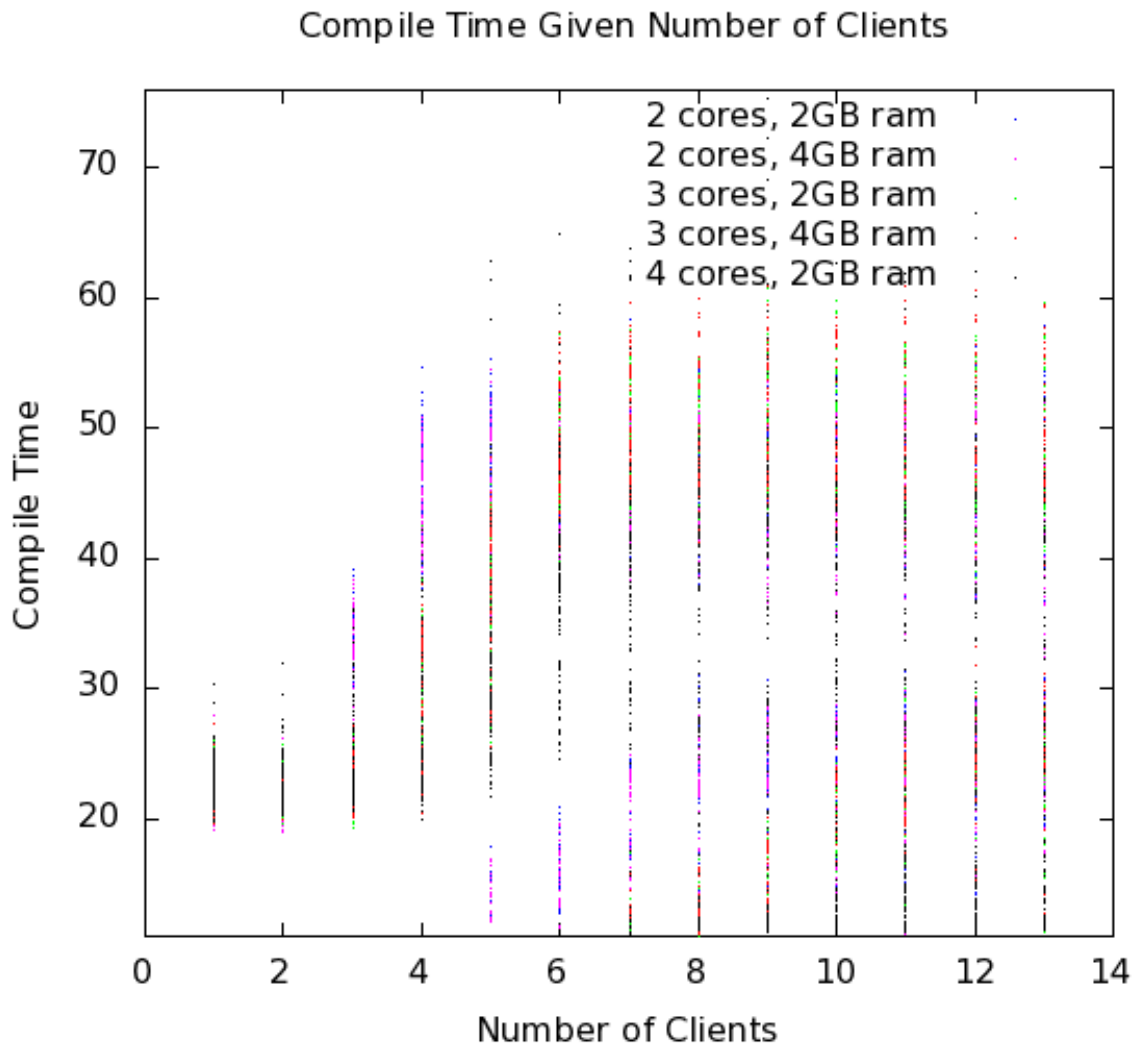
The queue size can be found by looking at the maximum number of clients running Puppet at once before any are dropped. According to the SIMP team's data, a server with two cores has a queue size of four; a server with three cores has a queue size of six; however, a server with four cores has a queue size of six. Although it may appear that the queue size is plateauing as cores are increased, the SIMP team predicts that this is due to the limited memory. However, the team is confident that a system with four cores and 4GB of ram will indeed have a queue size of eight clients. From this, it can be concluded that, given enough memory, **Queue_Size = 2*Cores**.

Also using this data, the compile times for other systems can be predicted given the amount of processors, memory, and nodes. This is done using ordinary least squares in Octave.

In addition, the maximum number of clients can also be predicted with the use of the following equation:

*Max_Num_Of_Total_Clients = (Run_Time_In_Sec / Comp_Time) * Queue_Size*

Where *Run_Time_In_Sec* is the number of seconds per half an hour (1800), *Queue_Size* is the maximum number of clients in the worst case scenario (queue size), and *Comp_Time* is the average compile time of the clients when there are *Max_Num_Worstcase* clients.



3.6 SIMP Administration

This chapter provides basic guidance on how to administer a SIMP environment.

> **Warning:** While working with the system, keep in mind that Puppet does not work well with capital letters in host names. Therefore, they should not be used.

3.6.1 Nightly Updates

All SIMP systems are configured, by default, to do a YUM update of the entire system on a nightly basis.

The configuration pulls updates from all repositories that the system is aware of. To change this behavior, refer to the *Excluding Repositories* FAQ section. This configuration is also helpful because it is easier to manage symlinks in YUM repositories than it is to manage individual package minutia for every single package on every system.

The general technique is to put packages that all systems will receive into the `Updates` repository provided with SIMP. Any packages that will only go to specific system sets will 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 symlinked or hard linked between repositories for maximum space utilization.

3.6.2 Sudosh

By default, a SIMP system uses *Sudosh* to enable logging of sudo sessions to Rsyslog. To open a sudo session as root (or any other user), type su - as simp, or sudo sudosh as anyone else, instead of sudo su.

The logs are stored in /var/log/sudosh.log. Sessions can be replayed by typing sudosh-syslog-replay.

3.6.3 User Accounts

By default, users can add local users to a system or use LDAP to administer users.

It is recommended that LDAP is used for adding all regular 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.

It is also possible that there will be users that are local to the system. To have these users follow the normal password expiration conventions set on the system, use the native Puppet user and group types.

To have a user that does not expire, look at the /etc/puppet/localusers file to enable these users across the systems. The comments in the file provide instructions on generating entries for the desired systems. It is hoped that future versions of Puppet will support the modification of password expiration values via the native types and that the localusers file will be retired.

3.6.4 Certificate Management

This section describes the two different types of certificates used in a SIMP system and how to manage them. For information on initial certificate setup, refer to the *Apply Certificates* section of the Client Management chapter.

3.6.5 Server Certificates

Server certificates are the standard PKI certificates assigned either by an official CA or generated using the FakeCA utility offered by SIMP. They can be found in the /etc/pki/ directory of both the client and server systems. These certificates are set to expire annually. To change this, edit the following files with the number of days for the desired lifespan of the certificates:

Note: This assumes that the user has generated Certificates with the FakeCA provided by SIMP. If official certificates are being used, these settings must be changed within the official CA, not on the SIMP system.

- /etc/puppet/Config/FakeCA/CA
- /etc/puppet/Config/FakeCA/ca.cnf
- /etc/puppet/Config/FakeCA/default_altnames.cnf
- /etc/puppet/Config/FakeCA/default.cnf
- /etc/puppet/Config/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 /etc/puppet/Config/FakeCA/output/conf/.

Important: Editing any entries in the above mentioned config files will not affect the existing certificates. To make changes to an existing certificate it must be re-created and signed.

Below 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 /etc/puppet/Config/FakeCA/)
do
  sed -i 's/365/1825/' $file
done
```

3.6.6 Puppet Certificates

Puppet certificates are issued and maintained strictly within Puppet. They are different from the server certificates and should be managed with the `puppet cert` tool. For the complete documentation on the `puppet cert` tool, visit the Puppet Labs cert manual detailing its capabilities. On a SIMP system, these certificates are located in the `/var/lib/puppet/ssl/` directory and are set to expire every five years.

3.6.7 Applications

This section describes how to add services to the servers. To perform this action, it is important to understand how to use IPtables and what the `svckill.rb` script does on the system.

3.6.8 IPTables

By default, the SIMP system locks down all incoming connections to the server save port 22. Port 22 is allowed from all external sources since it is expected that the user will want to be able to SSH into the systems from the outside at all times.

The default alteration for the IPtables start-up script is such that it will "fail safe". This means that if the IPtables rules are incorrect, the system will not open up the IPtables rule set completely. Instead, the system will deny access to all ports except port 22 to allow for recovery via SSH.

There are many examples of how to use the IPtables module in the source code; the Apache module at `/etc/puppet/modules/apache` is a particularly good example. In addition, look at the definitions in the IPtables module to understand their purpose and choose the best option. Refer to the IPtables page of the Developers Guide for a good summary and example code (HTML version only).

3.6.9 svckill.rb

To ensure that the system does not run more services than are required, the `svckill.rb` script has been implemented to stop any service that is not properly defined in the Puppet catalogue.

To prevent services from stopping, refer to the instructions in the *My Services Are Dying!* FAQ section.

3.6.10 GUI

SIMP was designed as a minimized system, but it is likely that the user will want to have a GUI on some of the systems. Refer to the *Infrastructure Setup* section for information on setting up GUIs for the systems.

3.7 Backing up the Puppet Master

This section details all of the steps required for backing up the Puppet Master.

Note: SIMP, by default, provides two ways to back up data. They are BackupPC and Git. If there is a different preferred method, the user may install it and configure it first.

Warning: BackupPC may, or may not, work properly for you on RHEL7+ systems. The SIMP team is currently evaluating other options for an inbuilt backup system.

1. Backup `/var/lib/puppet/ssl`
2. Backup `/etc/puppet`
3. Backup `/srv/rsync` and/or `/var/simp/rsync`
4. **Optional:** Backup /var/www

Table: SIMP Upgrade Process

3.8 Managing Workstation Infrastructures

This chapter describes how to manage client workstations with a SIMP system including GUIs, repositories, virtualization, Network File System (NFS), printing, and Virtual Network Computing (VNC).

3.8.1 Infrastructure Setup

The following sections provide examples for setting up a SIMP workstation environment.

3.8.2 User Workstation Setup

Below is an example class, `/etc/puppet/modules/site/manifests/workstation.pp`, that could be used to set up a user workstation.

```
class site::workstation {
  include 'site::gui'
  include 'site::repos'
  include 'site::virt'
  include 'site::automount'
  include 'site::print::client'

  # 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::manage { "Allow Users":
    comment => 'Allow all users in the "users" group to access the system from anywhere.',
    users   => '(users)',
    origins => ['ALL']
  }

  # General Use Packages
  package { [
    'pidgin',
```

```
    'git',
    'control-center-extra',
    'gconf-editor',
    'evince',
    'libreoffice-writer',
    'libreoffice-xsltfilter',
    'libreoffice-calc',
    'libreoffice-impress',
    'libreoffice-emailmerge',
    'libreoffice-base',
    'libreoffice-math',
    'libreoffice-pdfimport',
    'bluefish',
    'gnome-media',
    'pulseaudio',
    'file-roller',
    'inkscape',
    'gedit-plugins',
    'planner'
  ]: ensure => 'latest'
  }
}
```

3.8.3 Graphical Desktop Setup

Below is an example manifest called `/etc/puppet/modules/site/manifests/gui.pp` for setting up a graphical desktop on a user workstation.

```
class site::gui {
  include 'xwindows::gdm'
  include 'windowmanager::gnome'
  include 'vnc::client'

   # Compiz Stuff
  package { [
    'fusion-icon',
    'emerald-themes',
    'compiz-fusion-extras',
    'compiz-fusion-extras-gnome',
    'vinagre'
  ]:
    ensure => 'latest'
  }
}
```

3.8.4 Workstation Repositories

Below is an example manifest called `/etc/puppet/modules/site/manifests/repos.pp` for setting up workstation repositories.

```
class site::repos {
  # Whatever local yumrepo statements you need for installing
  # your packages and keeping your systems up to date
}
```

3.8.5 Virtualization on User Workstations

Below is an example manifest called `/etc/puppet/modules/site/manifests/virt.pp` for allowing virtualization on a user workstation.

```
# We allow users to run VMs on their workstations.
# If you don't want this, just don't include this class.
# If this is installed, VM creation and management is still limited by PolicyKit

class site::virt {
  include 'libvirt::kvm'
  include 'libvirt::ksm'
  include 'network::redhat'

  network::redhat::add_eth { "em1":
    bridge => 'br0',
    hwaddr => $::macaddress_em1
  }

  network::redhat::add_eth { "br0":
    net_type => 'Bridge',
    hwaddr => $::macaddress_em1,
    require => Network::Redhat::Add_eth["em1"]
  }

  common::swappiness::conf { 'default':
    high_swappiness => '80',
    max_swappiness => '100'
  }

  # If 80% of memory is used, flush caches.
  exec { 'flush_cache_himem':

    command => '/bin/echo 1 > /proc/sys/vm/drop-caches',
    onlyif => inline_template("/bin/<%= memoryfree.split(/\s/)[0].
    to_f/memorysize.split(/\s/)[0].to_f < 0.2 ? true : false %>")
  }

  package { 'virt-manager': ensure => 'latest' }
}
```

3.8.6 Network File System

Below is an example manifest called `/etc/puppet/modules/site/automount.pp` for Network File System setup.

```
#If you are not using NFS, you do not need to include this.

class site::automount {
  include 'autofs'

  file { '/net':
    ensure => 'directory',
    mode   => '0755'
  }

#A global share
```

```
Autofs::map::master { 'share':
  mount_point => '/net',
  map_name    => '/etc/autofs/share.map'
}
#Map the share
autofs::map::entry { 'share':
  options    => '-fstype=nfs4, port=2049.soft',
location    => "${::nfs_server}:/share'.
Target     => 'share'
}
}
```

3.8.7 Setting up a Printer Environment

Below are example manifests for setting up a printing environment.

Setting up a Print Client

Below is an example manifest called `/etc/puppet/modules/site/manifests/print/client.pp` for setting up a print client.

```
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

Below is an example manifest called `/etc/puppet/modules/site/manifests/print/server.pp` for setting up a print server.

```
class site::print::server {

  # Note, this is *not* set up for being a central print server.
  # You'll 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']
  }
}
```

3.9 VNC

Virtual Network Computing (VNC) is a tool that is used to manage desktops and workstations remotely through the standard setup or a proxy.

3.9.1 VNC Standard Setup

Note: You must have the `pupmod-vnc` RPM installed to use VNC on your system!

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/puppet/modules/vnc/manifests/server.pp <file:///etc/puppet/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.

3.9.2 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/puppet/hieradata/hosts/vserv.your.domain.yaml` file. In the client hosts file, modify or create the entries shown in the examples below. These additional modules will allow vserv to act as a VNC server and vclnt to act as a client.

VNC Server node

```
# vserv.your.domain.yaml
classes:
  - 'windowmanager::gnome'
  - 'mozilla::firefox'
  - 'vnc::server'
```

VNC client node

```
# vclnt.your.domain.yaml
classes:
  - 'windowmanager::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`

Note: 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**

   Note: 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**

   Note: 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, X 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`.

3.10 Upgrading SIMP

This chapter provides information on how to upgrade a running instance to the latest codebase.

3.10.1 Pre-Upgrade Recommendations

The following process should be followed before upgrade.

1. Run `puppet agent --disable` to disable puppet.

   Note: If you think you will need more than 4 hours to complete this task, also disable puppet in root's crontab.

   ---

2. You may wish to block all communications with agents while updating the server. This is not required but could spare you some headaches if something doesn't work properly.

   The simplest way to do this is to set the catalog retrieval capability to 127.0.0.1 in `/etc/puppet/auth.conf` as shown below.

```
path ~ ^/catalog/([^/]+)$
method find
# Uncomment this when complete and delete the other entries
#allow $1
allow 127.0.0.1
```

Using the syntax above, you can add fully qualified domain names, one at a time, to the 'allow' list and only those hosts will be able to retrieve their catalog from the running server. 127.0.0.1 serves as a placeholder so that no host can actually retrieve their catalog.

3.10.2 Migrating To Environments

SIMP 4.1 and 5.0 used the traditional, Rack-based, Puppet Master. Starting with 4.2 and 5.1, SIMP now uses the Clojure-based Puppet Server. Unfortunately, there are some conflicts with directly upgrading from the Puppet Master to the Puppet Server since some of the RPM package prerequisites conflict. This new Puppet Server can properly utilize Puppet Environments. To provide our users with this capability, and to facilitate more dynamic workflows in the future, the SIMP team has migrated **all** existing material to a native *simp* environment. To help facilitate your migration, the SIMP team has created two migration scripts that both upgrade your Puppet Server and migrate your existing data into the new *simp* environment.

Warning: You must have at least **2.2G** of **free memory** to run the new Puppet Server.

3.10.3 Migration Script Features

The migration script will perform the following actions on your system:

- Remove the `puppet-server` package from your system
- Install the `puppetserver` package onto your system
- Update all packages from your repositories
- Create a backup folder at `/etc/puppet/environments/pre_migration.simp`
- Create a Git repository in the backup folder under a timestamped directory
- Commit all current materials from `/etc/puppet` into the backup Git repository
- Checkout the backup Git repository under the timestamped directory as `backup_data` for ease of use
- Migrate all existing data into the new `simp` environment under `/etc/puppet/environments/simp`

Note: All future upgrades will only affect the new `simp` environment. You may create new environments and/or modify the contents of `/etc/puppet/modules` without fear of the SIMP packages overwriting your work.

3.10.4 Migration Script Execution

1. Copy the new SIMP ISO onto your system. For the purposes of these instructions, we will refer to this is SIMP_Update.iso. Please ensure that you are in the directory with the ISO prior to proceeding. Extract the new simp-utils package using the following command:

```
isoinfo -i SIMP_Update.iso -R -x `isoinfo -i SIMP_Update.iso -Rf | grep noarch/simp-utils` > sim
```

2. Install the new simp-utils RPM:

```
yum -y localupdate simp-utils*.rpm
```

3. Unpack the DVD onto the system:

```
/usr/local/bin/unpack_dvd SIMP_Update.iso
```

4. Run the migration script (this may take some time, do NOT hit CTRL-C!):

```
/usr/share/simp/upgrade_script/migrate_to_environments
```

5. Run the puppet agent:

```
puppet agent -t
```

6. Stop the new puppetserver service (it may not be running):

```
service puppetserver stop
```

7. Remove any left over PID files:

```
rm /var/run/puppetserver/puppetserver
```

8. Kill any running puppet master processes:

```
pkill -f 'puppet master'
```

9. Wait for 10 seconds to let things finalize if necessary:

```
    sleep 10
```

10. Start the new Puppet Server:

```
    service puppetserver start
```

Table: Executing the Migration Script

Your new Puppet Server should now be running and a run of `puppet agent -t` should complete as usual.

3.10.5 Converting from Extdata to Hiera

SIMP now uses Hiera natively instead of Extdata. Tools have been put into place by Puppet Labs and SIMP to make the conversion as easy as possible. Two scripts have been provided to automatically convert generic csv files and `simp_def.csv` to yaml. The first example shows how to convert an Extdata csv file called foo.csv into a Hiera yaml file called `bar.yaml`:

```
extdata2hiera -i foo.csv -o bar.yaml
```

The second example shows how to convert an Extdata csv simp_def file called simp_def.csv into a Hiera yaml file called `simp_def.yaml`.

```
simpdef2hiera --in simp_def.csv --out simp_def.yaml
```

Puppet will automatically retrieve class parameters from Hiera, using lookup keys like `myclass::parameter_one`. Puppet classes can optionally include parameters in their definition. This lets the class ask for data to be passed in at the time that it's declared, and it can use that data as normal variables throughout its definition.

There are two main ways to reference Hiera data in puppet manifests. The first, and preferred way, is to use the automatic class variable lookup capability. For each class that you create, the variables will be automatically discovered in hiera should they exist. This is quite powerful in that you no longer need to provide class parameters in your manifests and can finally properly separate your data from your code.

Note: For more information on the lookup functions, see Link the puppet documentation on Hiera.

```
# Some class file in scope...
class foo (
  $param1 = 'default1'
  $param2 = 'default2'
) { .... }

# /etc/puppet/hieradata/default.yaml
---
foo::param1: 'custom1'
```

The second is similar to the old Extdata way, and looks like the following:

```
$var = hiera("some_hiera_variable", "default_value")
```

The following is from the Puppet Labs documentation, and explains the reason for switching to Hiera.

Automatic parameter lookup is good for writing reusable code because it is regular and predictable. Anyone downloading your module can look at the first line of each manifest and easily see which keys they need to set in their own Hiera data. If you use the Hiera functions in the body of a class instead, you will need to clearly document which keys the user needs to set.

Note: For more information on hiera and puppet in general, see http://docs.puppetlabs.com/hiera/1/complete_example.html.

3.10.6 Scope Functions

All scope functions must take arguments in array form. For example in `/etc/puppet/modules/apache/templates/ssl.conf.erb`:

```
<%=scope.function_bracketize(l) %>
becomes
<%=scope.function_bracketize([l]) %>
```

3.10.7 Commands

Deprecated commands mentioned in Puppet 2.7 upgrade are now completely removed.

3.10.8 Lock File

Puppet agent now uses the two lock files instead of one. These are the run-in-progress lockfile (`agent_catalog_run_lockfile`) and the disabled lockfile (`agent_disabled_lockfile`). The `puppetagent_cron file` (made by the pupmod module) must be edited to suit this change.

3.11 Logstash

This chapter gives instruction for getting a basic configuration of Logstash working in a SIMP environment.

3.11.1 Logstash

Logstash is an open source tool that provides a means for SIMP implementations to have logs and events collected, searched, and forwarded (filtered or unfiltered) to another host. SIMP comes with three separate but related modules. The modules are:

- **Logstash:** Installs the RPMs and configuration needed for log inputs, filters, and outputs.
- **Kibana:** Installs the RPMs and configuration needed for the Kibana 3 web interface.
- **Elasticsearch:** Installs the RPMs and configuration needed for Elasticsearch.

Warning: The Logstash class is incompatible with the SIMP `rsyslog::stock::server` class! You cannot enable both of them on the same sever.

3.11.2 Logstash Architecture

The overall model for Logstash is very simple. It takes inputs from various sources, optionally applies filters, and outputs the results to a specified target. It's likely that you can already forward logs to Logstash and output them in a useful format as part of your existing architecture.

Logstash filters can manipulate logs after ingest and before output. Examples of existing filters include fixing logs to split/combine lines, adding fields, normalizing time stamps, and adding GeoIP fields. Depending on the type of log manipulation that is desired, there is likely a filter and associated documentation that already exists.

3.11.3 Logstash SIMP Architecture

Applying the SIMP Logstash, Elasticsearch, and Kibana modules provides an implementation with a functioning log reduction and search capability. Unless scale dictates otherwise, these three modules can easily be applied to a single host.

The intent of providing Logstash in SIMP is to replace the default Rsyslog server with a capability that is easier to search and analyze over time. Once your Logstash server is set up, you simply need to direct your hosts to forward logs to your Logstash server. In a default SIMP configuration, this can be done by setting the $log_server variable in hiera.

Note: SIMP does **NOT** apply any filters to the logs by default.

It is up to each implementation to define and apply filters that meet their local requirements. While multiple output targets may be defined, SIMP only defines the Elasticsearch output by default. Please see the Elasticsearch Puppet module for details on how to define additional output targets.

3.11.4 SIMP Logstash Fow

Logstash, SIMP, and Security

The provided SIMP modules for Logstash, Elasticsearch, and Kibana have been built with connection security in mind. Overriding these settings could adversely affect the security of the logging infrastructure. The following list describes the security features in place with the default SIMP module settings:

Warning: The native (Java) Elasticsearch connections are not encrypted! This will be remedied in the future as sufficient methods are found.

- **User Name and Password Protection for Kibana:** The Kibana web can be exposed to a defined list of hosts. If you are connecting to Kibana from anything other than the localhost, a user name and password is required for authentication. Both LDAP and local database users are supported.
- **Syslog over Stunnel:** The default behavior in SIMP is to encrypt syslog traffic over Stunnel. This remains the case with Logstash. Unencrypted traffic is also supported for network devices.
- **Limiting Web Actions:** The Kibana module restricts what HTTP commands a user can perform on the Elasticsearch data store. Full POST action must be given to the Logstash nodes and some nodes may require DELETE capabilities. Logstash hosts should be tightly controlled so that administrative users cannot modify data inside of Elasticsearch with carefully crafted commands. This is one reason that we use syslog on the local hosts.

Important: The Puppet modules for Logstash, Kibana, and Elasticsearch contain dozens of variables that may be manipulated. You should read each product's documentation and ensure you understand any setting that is changed from the default SIMP values. Changes can affect both security and functionality of the system.

3.11.5 Logstash Setup

3.11.6 Logstash System Requirements

The storage requirements for Logstash and Elasticsearch vary depending on how long you plan on keeping logs. If you use the settings in ?, then your logs are not being filtered and are being sent to Elasticsearch. When using Elasticsearch, the logs are formatted for Elasticsearch and stored in `/var/elasticsearch`. You can also configure how many days of data you wish to keep in Elasticsearch (keep_days => '99'). Therefore, you should ensure you have enough space on `/var` to keep your defined number of days worth of logs.

As you grow your Elasticsearch cluster to handle increasing log loads, you will want to ensure that your keep_days is set to handle your entire cluster appropriately.

Note: You should have at least 4G of memory available on any Elasticsearch node.

Important: You should NOT install Logstash, Elasticsearch, nor Kibana on your Puppet master. There will likely be conflicts with Apache and resource limitations.

3.11.7 Logstash Module Recommended SIMP Setup

The following example manifest can be applied to a single host with a large /var volume and 4GB of memory.

```
---
# Add these settings to only your Logstash node.

apache::ssl::sslverifyclient: %{hiera('kibana::ssl_verify_client')}

kibana::redirect_web_root: true
kibana::ssl_allowroot: %{hiera('client_nets')}
kibana::ssl_verify_client: 'none'
# You can add more groups under ldap_groups if you want others
# to be able to access your Kibana instance.
#
# Remember, whitespace matters!
#
kibana::method_acl:
  'method':
    'ldap':
      'enable': true
  'limits':
    'users':
      'valid-user': 'defaults'
    'ldap_groups':
      'cn=administrators,ou=Group,dc=your,dc=domain': 'defaults'

logstash::simp::keep_days: '30'

elasticsearch::simp::manage_httpd: 'conf'

classes:
  - 'logstash::simp'
  - 'kibana'
```

In the case of the Elasticsearch node setup below, it may be better to use a group match to pull your Hiera settings. To do this, you should add the following to a file like /etc/puppet/manifests/nodegroups.pp

```
if $trusted['certname'] =~ /es\d+\.your\.domain/ {
  $hostgroup = 'elasticsearch'
}
```

Then, ensure that a file called 'elasticsearch.yaml' is present in the .. only:: not simp_4

`/etc/puppet/environments/simp/hieradata/hostgroups/` directory and contains the following

content.

```
---
# All nodes running elasticsearch in your cluster should use
# these settings.
elasticsearch::simp::cluster_name: 'a_unique_hard_to_guess_name'
# This can be no more than the total number of ES nodes that you
# have in your cluster.
elasticsearch::simp::replicas: '2'
elasticsearch::simp::java_install: true

classes:
  - 'elasticsearch::simp'
```

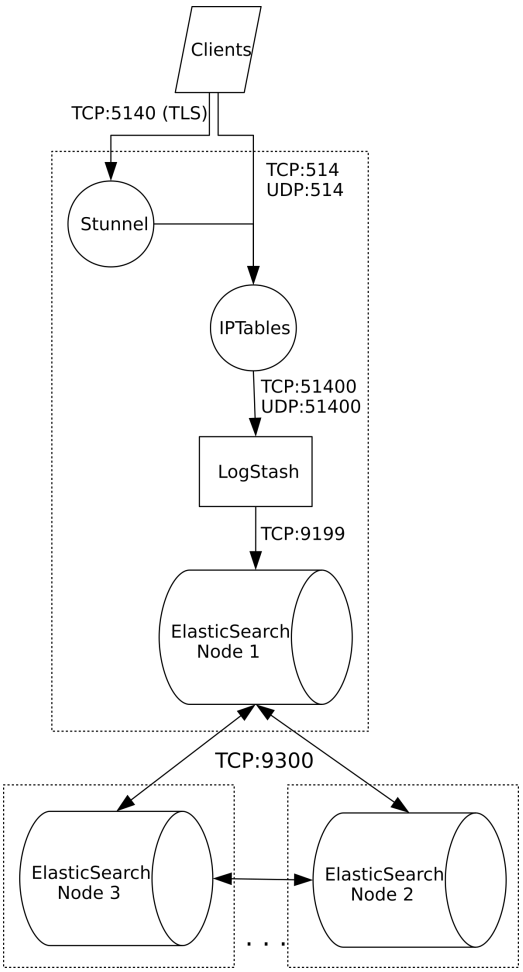
Make sure you point your clients to the Logstash server by setting the `log_server` variable to the fqdn of the Logstash server in hiera. This is further covered in ?.

Using LogStash and ElasticSearch

With the default settings applied, you should be able to connect to port 443 on your Kibana host. If connecting from localhost, you will not be prompted for a password. If you are connecting from an external host, a valid LDAP account with that user being defined in the Kibana Class is needed. The page is SSL protected so use https://<hostname>/kibana

With the web interface up, you now have the ability to search logs.

There are several resources available to help with searching. The Kibana Overview Page and Elasticsearch Guide are a good place to start. You should also visit the main Logstash page to see demonstrations and read their tips for searching logs.



3.12 Using Kerberos 5 in SIMP

The *Kerberos* module helps an administrator obtain a working *Key Distribution Center* (KDC) setup and configure clients to use the KDC.

Important: Given the highly sensitive nature of Kerberos passwords and tokens, this module does not store or use any passwords related to the Kerberos KDC.

Remember the passwords chosen for the Kerberos KDC. Puppet does not have the ability to retrieve forgotten passwords.

As a result of the nature of Kerberos, an administrator must run `/usr/sbin/kdb5_util create -s` on the KDC to set the principal administrator password and initialize the database.

The following sections provide instruction on how to get started with Kerberos 5. For more detailed information, review the official Red Hat documentation.

3.12.1 Creating Principals

Once all of the systems using Kerberos are properly configured, either via the `krb::stock` classes or otherwise, the administrator must register principals with the KDC.

Create the Admin Principal

The first principal to be registered is an admin principal that manages the environment, since it is in the admin group. This principal must be created on the KDC system.

Before creating the admin principal, the user must first create an *Access Control List* (ACL). To accomplish this, add the following Puppet code to the site manifest for the KDC system. If a custom implementation of Kerberos is being used, changes may need to be made to the code.

Code for Creating an Admin Principal Kerberos

```
krb5_acl{ "${::domain}_admin":
  principal       => "*/admin@${::domain}",
  operation_mask  => '*'
}
```

The table below lists the steps to create an admin principal that is appropriate for common organizations. These steps should be accomplished after creating the ACL by using the code provided in the previous example.

1. After using the code from the previous example, run `puppet agent -t` to allow the changes to take effect.
2. To finish creating the principal, type `/usr/bin/kadmin.local -r ***<Your.Domain>* -q "addprinc *<User Name>*/admin"`

   Note: By following this step, all features of the admin principal can be used remotely.

3. To load the principal, type `/usr/bin/kinit ***<User Name>*/admin**`

Table: Creating the Admin Principal Procedure

Create the Host Principal(s)

Once the admin principal has been created, host principals for each host can be made. The table below lists the steps to complete this action.

1. On the KDC, generate a principal for each host in the environment by typing `/usr/sbin/kadmin.local -r ***<Your.Domain>* -q 'addprinc -randkey host/*<FQDN>*'`

   Note: To use much of the functionality of the host, the user must first ensure that each host has a keytab. SIMP uses the `/etc/puppet/keydist` directory for each host to distribute keytabs securely to the clients.

2. To create a keytab file for each of the above hosts, type `/usr/sbin/kadmin.local -r ***<Your.Domain>* -q 'ktadd -k *<FQDN>*.keytab host/*<FQDN>*'`
3. Propagate all keytabs to the Puppet server by moving all of the resulting keytab files securely to the `/etc/puppet/keydist/<FQDN>/keytabs` directory on the Puppet server, as appropriate for each file.
4. Update the node declarations to include `krb::keytab`.

   Note: Ensure that all keytab directories are readable by the group Puppet, but not globally.

Table: Creating Host Principals Procedure

Once the Puppet Agent runs on the clients, the keytabs are copied to the `/etc/krb5_keytabs` directory. The keytab matching the FQDN is set in place as the default keytab, `/etc/krb5.keytab`.

3.13 Troubleshooting Common Issues

How to troubleshoot common problems that occur when installing and using SIMP.

3.13.1 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.rb*

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 type 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.rb* script that runs with every Puppet run.

The *svckill.rb* script:

- Collects a list of all services on the system. These are the same services that the user sees after typing `chkconfig --list`
- Ignores certain critical services, including Puppet, IPtables, 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, declare them in the node manifest space.

Note: The key is to declare the services and not set them to any other option. By adding them to the manifest, the *svckill.rb* script will ignore them.

The example below demonstrates this action, assuming that the *keepmealive* service is added to the *chkconfig*.

preventing a service from being killed by svckill.rb

```
service { "keepmealive": }
```

3.13.2 Why Can't I Login?!

If you've reached this page, you're 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 setup incorrectly, or your *PKI* certificates are invalid.

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 don't 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::manage define as shown below.

```
pam::access::manage { 'Allow the security group into the system':
  users   => ['(security)'],
  origins => ['ALL'],
  comment => 'The core security team'
}

pam::access::manage { 'Allow bob into the system from the proxy only':
  users   => ['bob'],
  origins => ["proxy.${::domain}"],
  comment => 'Bob the proxied'
}
```

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.

```
$ nslookup 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.

```
$ nslookup 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.

3.13.3 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 contains its own Fake CA and, therefore, you must know which environment is serving the systems that are having issues prior to proceeding.

For this section, we will assume that it is the 'simp' environment located at the active environment path.

Note: 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.

For the remainder of this section, we will assume that the *FQDN* of the system with issues is 'system.my.domain' and the LDAP server to which it is attempting to connect is 'ldap.my.domain'.

Navigate to the environment *keydist* directory and validate the system certificates.

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'.

```
$ cd `puppet config print environmentpath`/simp/keydist

# Validate the client system
$ openssl verify -CApath cacerts system.my.domain

# Validate the LDAP system
$ openssl verify -CApath cacerts ldap.my.domain
```

If there are any issues, you may need to follow the Fake CA README to generate new certificates for one or more of your hosts.

3.13.4 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 mmaster and the clients!

If you need to fix client certificate issues outside of time, first make sure that you don't have a certificate already in place on your Puppet server.

```
$ puppet cert 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.

```
$ puppet cert clean <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!

Warning: **NEVER RUN ‘‘puppet cert clean –all‘‘**

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
```

Puppet Server Certificate Issues

Warning: This is destructive to your Puppet communications. This should only be used if you have no other options.

If the Puppet server has certificate issues, regenerate the server CAs. To do this, remove the contents of the *ssl* folder and regenerate those `.pem` files.

The following table lists the steps to regenerate the server CAs:

```
$ service puppetserver stop
$ rm -rf /var/lib/puppet/ssl
$ puppet cert list --all
$ puppet cert --generate ***<fqdn>***
$ service puppetserver start
$ puppet agent --test
```

3.14 SIMP FAQs

This chapter answers some of the frequently asked questions (FAQs) about SIMP.

3.14.1 Centralized Logging

SIMP provides a pre-built set of classes within the *rsyslog* module for enabling centralized logging within the infrastructure.

After completing these steps, run Puppet on the server and clients, or wait until after the next run to see logs start to flow.

Enable the Server

To enable the pre-built log server, add the following example code to the designated logging node.

Code to Enable the Server Logging Examples

```
classes :
  - 'simp::rsyslog::stock'
```

Enable the Clients

To have clients send data to the server, make the following changes to the `/etc/puppet/hieradata/simp_def.yaml` file.

Code to Enable the Client Logging Examples:

```
log_server="fqdn.of.your.log.server"
```

3.14.2 Changing Puppet Masters

It may be necessary to change the Puppet Master. To point a particular client to a new Puppet Master, follow the steps in the sections below.

On the Client

Enter the following changes into the */etc/puppet/puppet.conf* file.

Code Changes on Client to Switch Puppet Masters

```
server = new.puppet.master.fqdn
ca_server = new.puppet.master.fqdn
ca_port = 8141
```

To remove all files and sub-directories in the `/var/lib/puppet/ssl` directory, type `cd /var/lib/puppet/ssl`. Then type `rm -rf ./*`.

Assuming the new Puppet Master has been set up to properly accept the client, type `puppet agent --test` to run a full Puppet run while pointing to the new server.

If all goes well, the client will now be synchronized with the new Puppet Master. If not, refer to the SIMP Server Installation section of the SIMP Install Guide and ensure that the new Puppet Master was set up properly.

On the Old Puppet Master

Remove or comment out all items for the client node in the `/etc/puppet/hieradata/hosts` space.

To run `puppet agent` in *noop* mode to ensure that there are no inadvertent errors, type `puppet agent --test --noop`.

3.14.3 Building a Bootable DVD from the SIMP tarball

SIMP is an overlay on top of RHEL, not a complete distribution. As such, the user must build a bootable DVD if provided with the SIMP source code or *tar* file.

To build a bootable SIMP DVD, if provided a RHEL DVD and the SIMP *tar* file, follow the steps in the sections below.

Build the DVD

The table below lists the steps to build a SIMP DVD, assuming that the user has copied the DVD to a location with enough space to house and unpack the ISO (around 10G).

Starting from the directory with the ISO, complete the steps outlined below. These steps are based on an example ISO of `rhel-server-6.7-x86_64-dvd.iso`.

1. Type

```
for file in `isoinfo -Rf -i rhel-server-6.7-x86_64-dvd.iso | \
  tac`; do mkdir -p RHEL6.7-x86_64`dirname $file`; \
  isoinfo -R -x $file -i rhel-server-6.7-x86_64-dvd.iso > RHEL6.7-x86_64$file; done
```

2. Type `tar -C RHEL6.7-x86_64 -xzf ***<SIMP tarball>***`

3. Type

```
mkisofs -o SIMP-6.7-***<SIMP Version>-x86_64.iso \***
  -b isolinux/isolinux.bin -c boot.cat -no-emul-boot -boot-load-size 4 \
  -boot-info-table -R -m TRANS.TBL -uid 0 -gid 0 RHEL6.7-x86_64
```

Table: Build a SIMP DVD Procedure

The fully bootable SIMP DVD is ready to install on a new system. Replace the RHEL version and architecture to fit the user's needs. See the Changelog for compatible RHEL versions.

Use the Alternative Method

If the Ruby *rake* utility is installed, use the *Rakefile* provided in the *Docs/examples* directory of the *tar* file.

3.14.4 Excluding 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 `common::yum_schedule::repos` and `common::yum_schedule::disable` variables in the pupmod-common module control which repositories are enabled for nightly updating. Both variables must be specified in array format.

`common::yum_schedule::repos` is used to specify an array of repositories from which updates are provided; no other repositories will be used.

`common::yum_schedule::disable` is used to specify an array of repositories from which updates are not provided; all other repositories will be used.

3.14.5 IPtables NAT Rules

See the IPtables Module Reference for notes on using the basic IPtables Module.

Add NAT Rules

The user may be required to add *Network Address Translation* (NAT) rules to the IPtables ruleset. To achieve this using the IPtables module, SIMP 1.1.3 or later is required and the `iptables::add_rules` input statement should be used to affect the appropriate changes.

The example below shows an IPtable NAT rule.

Example of an IPtable NAT Rule

```
 iptables::add_rules { "nat_global":
    table => "nat",
    first => "true",
    absolute => "true",
    header => "false",
    content => "
    :PREROUTING ACCEPT [0:0]
    :POSTROUTING ACCEPT [0:0]
    :OUTPUT ACCEPT [0:0]
    "
  }

iptables::add_rules { "nat_test":
    table   => "nat",
    header  => "false",
    content => "
    -A PREROUTING --physdev-in
    eth1 -j DROP
    "
  }
```

3.14.6 Network-based Initial Server Build

This section provides guidance to install the initial SIMP server via an existing kickstart infrastructure.

Prepare the Kickstart

To kickstart the initial server, copy the `netboot.cfg` file into the kickstart location from `ks/` at the root level of the extracted DVD.

Replace the *KS_SERVER* and *KS_BASE* variables in the `netboot.cfg` file to match the system settings.

Kickstart the System

Kickstart the system against the `netboot.cfg` file; this will build a functional SIMP server identical to the one that the user would have received from the DVD.

Post-Installation

This section describes the post installation procedures to use the server.

Setting up the new YUM repo

All of the SIMP systems must be able to reference two YUM locations after install. The first is the *Local* repo, which is spawned from the *Local* directory at the top of the DVD. This is expected to be referenced as `http://yum_server/yum/SIMP/<Architecture>` by the clients.

The second location is the *Updates* repo, which contains a repo with all of the base operating system RPMs. This is expected to be referenced as `http://yum_server/yum/(RedHat|CentOS)/<Version>/<Architecture>/Updates` by the clients.

The user is responsible for adjusting these locations in the pre-existing system; however, the table below lists the steps to adjust these locations on the newly built SIMP server.

Note: These steps assume that the SIMP DVD material is copied in its unpacked form to the `/srv/SIMP` directory and that the version unpacked is RHEL 5.8. Adjust the paths appropriately if the CentOS or 5.7 version is being used.

---

1. Copy the entire SIMP DVD material to the SIMP server.
2. Type `cd /srv;`
3. Type `mkdir -p www/yum/RedHat/5.8/x86_64;`
4. Type `mv /srv/SIMP/SIMP www/yum;`
5. Type `mv /srv/SIMP/ks www;`
6. Type `cd www/yum/RedHat`
7. Type `ln -s 5.8 6; and then cd 5.8/x86_64;` to be able to move to newer versions more easily.
8. Type `mkdir Updates;`
9. Type `cd Updates;`
10. Type `find ..  -type f -name "*.rpm" -exec ln -s {} \;`
11. Type `createrepo -p .`
12. Type `cd /var/www/yum/SIMP;`
13. Type `updaterepos;`
14. Type `chown -R root.apache /var/www;`
15. Type `chmod -R u+rwX,g+rX,o-rwx /var/www;`
16. Enter the following commands into the command line to adjust the file.

```
cat << EOF >> /etc/yum.repos.d/filesystem.repo
[flocal-x86_64]
name=Local within the filesystem
baseurl=file:///var/www/yum/SIMP/x86_64
enabled=1
gpgcheck=0
EOF
```

17. Enter the following commands into the command line to adjust the file.

```
cat << EOF >> /etc/yum.repos.d/filesystem.repo
[frhbase]
name=$ostype $rhversion base repo
baseurl=file:///var/www/yum/RedHat/6/x86_64/Server
enabled=1
gpgcheck=0
EOF
```

Follow the instructions in the *Client Management* for additional assistance.

3.14.7 Performing One-shot Operations

This section introduces the options provided for performing one-shot commands on all Puppet-managed systems without using Puppet. This is useful when the user needs to perform an action one time in every location, but does not want to enforce that action over time.

Use the PSSH Utility

Parallel Secure Shell (PSSH) has been included in SIMP for some time, but has not been installed by default.

The table below lists the steps to use PSSH.

Table: Use PSSH Procedure

Note: There is no manual page provided with PSSH; type `pssh --help` for further explanation.

Other SSH Options

Using the `-f` option forces *TTY* for SSH, which allows the user to run sudo commands via PSSH.

Using the *-OStrictHostKeyChecking=no* option connects the user to the target servers via SSH even if there is an issue with `~/.ssh/known_hosts`.

3.14.8 Puppet Server Behind a NAT

This section provides guidance for when the Puppet server is behind a NAT but is managing hosts outside the NAT.

To resolve this issue, open the `/etc/puppet/manifests/vars.pp` file and rename the `puppet_servers` variable to `puppet_server_hosts_mod`. Then, create a new `$puppet_servers` variable and point it to `template('site/nat_ip_switch.erb')`.

The entries in `vars.pp` should look like the following example.

Example Sample Entries in `vars.pp`

```
$puppet_server_hosts_mod = "puppet.$dns_domain|1.2.3.4 puppet2.$dns_domain|2.3.4.5"
$puppet_servers = template('site/nat_ip_switch.erb')
```

Create a `/etc/puppet/modules/site/templates/nat_ip_switch.erb` file with the content shown in the next example. Change the appropriate portions of the content to meet the needs of the user environment.

Important: Ensure that the `.erb` file is owned by *root.puppet* and mode *640*.

Source Create the `nat_ip_switch.erb`

```
<%
# Edit this variable to provide the IP address mappings.
# The left-hand side should contain the internal addresses.
# The right-hand side should contain the external addresses.
t_ipmap = {
    "1.2.3.4" => "10.10.10.10",
    "2.3.4.5" => '10.2.3.4'
}

# Edit this regex to match the hosts.
# This is done with a Regexp; the user can use whichever is preferred.
# Pure IP matching would be faster using the IPAddr class.
t_inside_nets = Regexp.new("^5\.*")

t_pupsrvs = puppet_server_hosts_mod.split(/\s|,|;/)

# Change the ipaddress variable to the host that the regexp above is matching.
if not t_inside_nets.match(ipaddress) then
```

```
  t_pupsrvs.each_index do |t_i|
    t_vals = t_pupsrvs[t_i].split(/\|/)
    if t_ipmap.include?(t_vals.last) then
      t_vals[-1] = t_ipmap[t_vals.last]
      t_pupsrvs[t_i] = t_vals.join('|')
    end
  end

  t_pupsrvs = t_pupsrvs.join(' ')
end
-%>
<%= t_pupsrvs -%>
```

Run `puppet agent -t` on the client to receive the appropriately mapped NAT address of the Puppet server.

If the user cannot connect to the NAT'd Puppet server, change the values in the `/etc/hosts` directory to the correct values and try running `puppet agent -t` again.

3.14.9 Redundant LDAP

This section describes how to set up redundant OpenLDAP servers in SIMP.

The version of OpenLDAP in RHEL5 only supports *syncrepl*. Multi-master replication has been added in a more recent version of OpenLDAP but is not currently supported in SIMP. *Syncrepl* is optimal for *Wide Area Network* (WAN) situations and is the SIMP default.

Set up the Master

If the standard `puppet_servers.pp` file in SIMP is being used, the user has a working master server. If not, the following example demonstrates how to use the SIMP *openldap* module to create a server using the `puppet_servers.pp` file .

Source Code for Using an OpenLDAP Server openldap

```
# These are some common variables.
# See /etc/puppet/manifests/vars.pp for the stock version.

$ldap_master = 'ldap://ldapmaster.your.domain'

class ldap_common {
  include 'openldap::slapd_pki'

  openldap::slapd::conf { 'default':
    suffix => 'dc=your,dc=domain',
    rootdn => 'dn=LDAPAdmin,ou=People,dc=your,dc=domain',
    rootpw => '{SSHA}$klskf$asoghaagasgasgaggawawg',
    tlsCertificateFile => "/etc/pki/public/${fqdn}.pub",
    tlsCertificateKeyFile => "/etc/pki/private/${fqdn}.pem",
    client_nets => [ '1.2.3.4/16' ]
  }
}

class ldap_master inherits ldap_common {
  include 'openldap::slapo::syncprov'

  openldap::slapo::syncprov::conf { "default": }
}
```

```
node ldapmaster {
  include 'ldap_master'
}
```

Set up the Replicated Servers

Once the master is ready, LDAP slave nodes must be configured to replicate data from the master. The example below shows an the code that should be added to the slave node in Puppet. The actual order of which gets done first is irrelevant; the replicated servers will attempt to contact the master until they are successful.

Source Code to Configure an LDAP Slave Node replication

```
class ldap_repl inherits ldap_common {
  include 'openldap::slapd::syncrepl'

  openldap::slapd::syncrepl::conf { "111":
    provider => $ldap_master,
    syncrepl_retry => '60 10 600 +',
    searchbase => 'dc=your,dc=domain',
    starttls => 'critical',
    bindmethod => 'simple',
    binddn => 'cn=LDAPSync,ou=People,dc=your,dc=domain',
    credentials => '<plain text password>',
    updateref => $ldap_master
  }
}

node ldaprepl1 {
  include "ldap_repl"
}

node ldaprepl2 {
  include "ldap_repl"
}
```

Promote a Slave Node

Slave nodes can be promoted to act as the LDAP master node. To do this, change the node classifications of the relevant hosts. The following example shows the promotion of the *ldaprepl1* server to the master server.

Source Promoting a Slave Node LDAP

```
# Change the common ldap server variable to promote the slave node.

$ldap_master = 'ldap://ldaprepl1.your.domain'

node ldapmaster {
  # include 'ldap_master'
}

node ldaprepl1 {
  # include 'ldap_repl'
  include 'ldap_master'
}
```

After the next Puppet run on all hosts, *ldaprepl1* will be promoted to the master and all slave nodes will point to it.

Troubleshooting

If the system is not replicating, it is possible that another user has updated the `$ldap_sync_passwd` and `$ldap_sync_hash` entries in the `/etc/puppet/manifests/vars.pp` file without also updating the value in LDAP itself; this is the most common issue reported by users.

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 OpenLDAP.

The example below shows the changes necessary to update the `$ldap_sync` information in LDAP.

Update `$ldap_sync` Information in LDAP Examples

```
dn: cn=LDAPSync,ou=People,dc=your,dc=domain
changetype: modify
replace: userPassword
userPassword: <Hash from $ldap_sync_hash>
```

Master Node Demotion

In the event that multiple master nodes have been set up, it may be necessary to demote one or more of them to slave instances. To do this, add the replication code shown in the previous section titled *Set up the Replicated Servers* to the manifest of the master node being demoted.

Once this is complete, manually remove the active database from the LDAP server being demoted and then run Puppet. The SIMP team is working to enable SIMP to handle this transition automatically in the future.

3.14.10 SFTP Restricted Account

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 => <Path to SFTP Server>
}
```

On a SIMP system, shell would be: `"/usr/libexec/openssh/sftp-server"`

Modify `/etc/shells`

To modify `/etc/shells` to include the shell information provided in the previous user account example, add `common::shells` in Hiera, and add `/usr/libexec/openssh/sftp-server` to the list.

3.14.11 SSH Authorized Keys Setup

This section provides guidance on managing SSH keys within the SIMP environment.

LDAP Enabled

When enabled, ssh 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 location can be changed by setting the `::ssh::server::conf::authorizedkeysfile` parameter in *Hiera* or your *ENC*.

See Also: *Managing Local/Service Users*

3.15 SIMP RPMs

This provides a comprehensive list of all SIMP RPMs and related metadata. Most importantly, it provides a list of which modules are installed by default and which are simply available in the repository.

Name	Version	Default
pupmod-acpid	0.0.1-1	**true**
pupmod-aide	4.1.0-7	**true**
pupmod-apache	4.1.0-18	**true**
pupmod-auditd	5.0.0-0	**true**
pupmod-augeasproviders	2.1.3-0	**true**
pupmod-augeasproviders_apache	2.0.1-0	false
pupmod-augeasproviders_base	2.0.1-0	**true**
pupmod-augeasproviders_core	2.0.1-0	**true**
pupmod-augeasproviders_grub	2.0.1-0	**true**
pupmod-augeasproviders_mounttab	2.0.1-0	false
pupmod-augeasproviders_nagios	2.0.1-0	false
pupmod-augeasproviders_pam	2.0.1-0	false
pupmod-augeasproviders_postgresql	2.0.1-0	false
pupmod-augeasproviders_puppet	2.0.1-0	false
pupmod-augeasproviders_shellvar	2.0.1-0	false
pupmod-augeasproviders_ssh	2.5.0-0	**true**
pupmod-augeasproviders_sysctl	2.0.1-0	false
pupmod-autofs	4.1.0-6	false
pupmod-clamav	4.1.0-6	**true**
pupmod-dhcp	4.1.0-4	**true**
pupmod-freeradius	4.2.0-4	false
pupmod-iptables	4.1.0-13	**true**
pupmod-libvirt	4.1.0-15	false
pupmod-logrotate	4.1.0-2	**true**
pupmod-mcafee	4.1.0-2	false
pupmod-mozilla	4.1.0-1	false
pupmod-named	4.2.0-6	**true**
pupmod-network	4.1.0-4	**true**
pupmod-nfs	4.1.0-13	false
pupmod-nscd	5.0.0-4	**true**

Continued on next page

Table 3.1 – continued from previous page

Name	Version	Default
pupmod-ntpd	4.1.0-8	**true**
pupmod-oddjob	1.0.0-1	false
pupmod-onyxpoint-gpasswd	1.0.0-1	**true**
pupmod-openldap	4.1.1-6	**true**
pupmod-openscap	4.2.0-2	false
pupmod-pam	4.1.0-12	**true**
pupmod-pki	4.1.0-5	**true**
pupmod-polkit	4.1.0-1	false
pupmod-postfix	4.1.0-4	**true**
pupmod-pupmod	6.0.0-20	**true**
pupmod-puppetlabs-apache	1.0.1-2	false
pupmod-puppetlabs-inifile	1.2.0-1	**true**
pupmod-puppetlabs-java	1.2.0-0	false
pupmod-puppetlabs-java_ks	1.2.0-1	false
pupmod-puppetlabs-mysql	2.2.3-1	false
pupmod-richardc-datacat	0.6.1-0	false
pupmod-rsync	4.2.0-2	**true**
pupmod-rsyslog	5.0.0-0	**true**
pupmod-selinux	1.0.0-4	**true**
pupmod-simp	1.1.0-4	**true**
pupmod-simp-activemq	2.0.0-0	false
pupmod-simp-elasticsearch	2.0.0-3	false
pupmod-simp-kibana	3.0.1-3	false
pupmod-simp-logstash	1.0.0-6	false
pupmod-simp-mcollective	2.0.0-0	false
pupmod-simpcat	5.0.0-0	**true**
pupmod-simplib	1.0.0-0	false
pupmod-site	2.0.0-3	**true**
pupmod-snmpd	4.1.0-3	false
pupmod-ssh	4.1.0-10	**true**
pupmod-ssh-augeas-lenses	4.1.0-10	**true**
pupmod-sssd	4.1.0-6	false
pupmod-stunnel	4.2.0-9	**true**
pupmod-sudo	4.1.0-2	**true**
pupmod-sudosh	4.1.0-3	**true**
pupmod-svckill	1.0.0-4	**true**
pupmod-sysctl	4.1.0-5	**true**
pupmod-tcpwrappers	3.0.0-2	**true**
pupmod-tftpboot	4.1.0-7	**true**
pupmod-tpm	0.0.1-8	**true**
pupmod-upstart	4.1.0-3	**true**
pupmod-vnc	4.1.0-3	false
pupmod-vsftpd	5.0.0-0	false
pupmod-windowmanager	4.1.0-2	false
pupmod-xinetd	2.1.0-3	false
pupmod-xwindows	4.1.0-3	false
puppetlabs-postgresql	4.1.0-1.SIMP	**true**
puppetlabs-puppetdb	5.0.0-0	**true**
puppetlabs-stdlib	4.9.0-0.SIMP	**true**

Continued on next page

Table 3.1 – continued from previous page

Name	Version	Default
rubygem-simp-cli	1.0.10-0.el7	**true**
rubygem-simp-cli-doc	1.0.10-0.el7	**true**
simp	5.1.0-RC1.1446834077	**true**
simp-bootstrap	5.2.1-1	**true**
simp-doc	5.1.0-0	**true**
simp-gpgkeys	2.0.0-3.el7	**true**
simp-rsync	5.1.0-2.el7	**true**
simp-rsync-clamav	5.1.0-2.el7	**true**
simp-utils	5.0.0-7	**true**

3.16 SIMP 5.1.0-0

3.16.1 Changelog

Contents

SIMP 5.1.0-0

Package: 5.1.0-0

This release is known to work with:

- RHEL 7.0 and 7.1 x86_64
- CentOS 7.0 x86_64 (1406 and 1503)

Warning: The default system passwords have changed! Please see the User's Guide for details.

Manual Changes Requred

- Bugs in the *simplib::secure_mountpoints* class (formerly *common::secure_mountpoints*)

Note: This only affects you if you did not have a separate partition for /tmp!

- There were issues in the secure_mountpoints class that caused /tmp and /var/tmp to be mounted against the root filesystem. While the new code addresses this, it cannot determine if your system has been modified incorrectly in the past.
- To fix the issue, you need to do the following: - Unmount /var/tmp (may take multiple unmounts) - Unmount /tmp (may take multiple unmounts) - Remove the 'bind' entries for /tmp and /var/tmp from /etc/fstab - Run **puppet** with the new code in place

Deprecations

- simp-hiera

  The *simp-hiera* RPM has been replaced by the upstream *hiera* package from Puppet Labs. The original simp-hiera fork had been maintained due to a need that the 'alias()' function now serves. Please run the *hiera_upgrade* script to convert your existing SIMP environment. You may also set the environment variable *HIERA_UPGRADE* to a path of your choice to update any other hieradata that you may have on your system.

- pupmod-simp-common

  The *::common* namespace has been deprecated in favor of the new *::simplib* namespace. This removes a commonly conflicting module name from the SIMP ecosystem.

  You will need to run the *migrate_to_simplib* script to update all of the relevant files. This script will only migrate items in the existing SIMP environment. You may also set the environment variable *UPGRADE_PATHS* to run the script on multiple external paths.

  All code was migrated.

- pupmod-simp-functions

  The *::functions* namespace has been deprecated in favor of the new *::simplib* namespace. This removes a commonly conflicting module name from the SIMP ecosystem.

  You will need to run the *migrate_to_simplib* script to update all of the relevant files. This script will only migrate items in the existing SIMP environment. You may also set the environment variable *UPGRADE_PATHS* to run the script on multiple external paths.

  The following items were not migrated:

  - append_if_no_such_line => Use simp_file_line{}
  - delete_lines => Use augeas{}
  - init_mod_nice => Use init_ulimit{}
  - init_mod_open_files => Use init_ulimit{}
  - line => Use augeas{}
  - prepend_if_no_such_line => Use simp_file_line{}
  - renice => No replacement, was not correct
  - replace_line => Use augeas{}

Significant Updates

- FIPS Mode is now enabled by default!
  - This is a SIGNIFICANT change and may impact many of your running applications that use encryption.
  - If you are upgrading, do **NOT** enable FIPS mode without extensive testing as it may cause various applications to not function properly any longer.

- The rsyslog module has been completely rewritten to support rsyslog 7.4. This is a breaking change from previous releases and will require active updates to existing systems. All modules with rsyslog integration ave been updated to accommodate this change:
  - Critical Variable Changes
    - The global *rsyslog::log_server_list* variable is now set to send to **all** of the servers in the Array by default.
      - This variable defaults to the global *log_servers* Array in Hiera.
    - There is a new variable *rsyslog::failover_log_servers* which is an Array of failover log servers to be used for your system. These will be tried, in order, until successful messages can be sent.
  - Updated Modules:
    - aide
    - apache
    - auditd
    - dhcp
    - logstash
    - openldap
    - rsync
    - simp
    - sudosh
- There was a bug in previous versions of SIMP that require the following LDIF to be run manually on the systems to correct the password policy checking.

  dn: cn=default,ou=pwpolicies,dc=your,dc=domain changetype: modify replace: pwdCheckModule pwdCheckModule: simp_check_password.so - dn: cn=noExpire_noLockout,ou=pwpolicies,dc=your,dc=domain changetype: modify replace: pwdCheckModule pwdCheckModule: simp_check_password.so
- The Electrical and SIMP modules for elasticsearch have been combined.

Upgrade Guidance

Fully detailed upgrade guidance can be found in the **Upgrading SIMP** portion of the *User's Guide*.

Warning: You must have at least **2.2GB** of **free** RAM on your system to upgrade to this release.

Note: Upgrading from releases older than 5.0 is not supported.

Expectations

Before you begin, please be aware that the following actions will take place as a result of the migrate_to_environments script:

- The *puppet-server* RPM will be removed
- The *puppetserver* RPM will be installed (no, that's not a typo)
- **ALL** SIMP Puppet code will be migrated into a new *simp* environment

  - This will be located at */etc/puppet/environments/simp*
- A backup of your running environment will be made available at */etc/puppet/environments/pre_migration.simp*
  - You will find timestamped directories under the *pre_migration.simp* directory that correspond to runs of the migration script
  - Your old files will be in a *backup_data* directory and will be linked to a local bare Git repository in the same space

The upgrade steps will also have you install PuppetDB. PuppetDB is installed by default if you kick from the DVD.

Security Announcements

CVEs Addressed

RPM Updates

Numerous RPMs were updated in the creation of this release. Several were included due to our use of *repoclosure* to ensure that RPM dependencies are met when releasing a DVD.

- This version include the latest RedHat 7.1 and CentOS 7.0 (1503) RPMs.
- Facter upgraded to 2.4.
- PuppetDB upgraded to 2.3.8-1

Fixed Bugs

- pupmod-aide
  - Change the call to the *rsyslog* init script to the *service* command to seamlessly support both RHEL6 and RHEL7.
- pupmod-apache
  - Removed all reliance on 'lsb*' facts since some environments do now wish to install the prerequisites for those facts to run.
  - Remove the apache_version fact and simply use the version controls built into the Apache configuration language.
  - Update all custom functions to properly scope definitions.
  - Ensure that mod_ldap is installed in SIMP >= 5.0.
  - Prevent apache from restarting after downloading a CRL.
- pupmod-clamav
  - Change the call to the *rsyslog* init script to the *service* command to seamlessly support both RHEL6 and RHEL7.
- pupmod-common => Deprecated - Replaced by pupmod-simplib!
- pupmod-simplib
  - Fixed the secure_mountpoints code so that it no longer incorrectly bind mounts /tmp or /var/tmp.
  - We no longer supply crontab or anacrontab in global_etcd.
  - Remove dynamic_swappiness cron job if a static value is set.

  - Ensure that the *passgen()* function fails on invalid scenarios. This prevents the accidental cration of empty passwords.
  - Allow the value *2* to be used for `rp_filter` in `simplib::sysctl`.
  - Added ability to return remote ip addrs.
- pupmod-dhcp
  - Change the call to the *rsyslog* init script to the *service* command to seamlessly support both RHEL6 and RHEL7.
- pupmod-elasticsearch
  - Ensured that Elasticsearch works properly with the new version of Apache.
  - Removed our default ES tuning since the default works better for LogStash.
  - Ensure that Puppet manages the Elasticsearch logging file.
- pupmod-functions
  - Fixed sysv.rb to explicitly require puppet/util/selinux, which caused puppet describe to have errors.
- pupmod-iptables
  - Fixed a bug that would cause issues with Ruby 1.8.7.
  - Fixed DNS resolution in IPv6.
  - Prevent IPv6 ::1 spoofed addresses by default.
- pupmod-simp-logstash
  - Fix issues with both TCPWrappers and IPTables when used with LogStash.
- pupmod-nfs
  - Updated the *mountd* port to be *20048* by default for SELinux issues in RHEL7.
- pupmod-ntp
  - Updated against NTP Security Vulnerabilities (Red Hat Article #1305723).
  - Ensure that *restrict* entries use DDQ format.
- pupmod-openldap
  - The Password Policy overlay was getting loaded into the default.ldif even if you didn't want to use it. This has been fixed.
  - Made the password policy overlay align with the latest SIMP build of the plugin.
    * This means that you *must* have version simp-ppolicy-check-password-2.4.39-0 or later available to the system being configured.
  - Change the call to the *rsyslog* init script to the *service* command to seamlessly support both RHEL6 and RHEL7.
  - Fixed reported bugs in syncrepl.pp.
  - Removed all reliance on the 'lsb*' facts since some users do not wish to install the prerequisite RPMs for LSB compliance.
- pupmod-openscap
  - Change the call to the *rsyslog* init script to the *service* command to seamlessly support both RHEL6 and RHEL7.

  - Changed default ssg base path to /usr/share/xml/scap/ssg/content

- pupmod-pam
  - Removed all reliance on the 'lsb*' facts since some users do not wish to install the prerequisite RPMs for LSB compliance.
- pupmod-pki
  - Now allow directories in the cacerts directories. This previously caused failures that needed to be manually addressed on each node.
- pupmod-rsync
  - Fixed provider to run with –dry-run when puppet is run with a –noop.
- pupmod-simp
  - Ensure that SSSD is used by default on EL7+ systems since nscd and nslcd have functionality issues.
  - Removed all reliance on the 'lsb*' facts since some users do not wish to install the prerequisite RPMs for LSB compliance.
- pupmod-ssh
  - Modernized the Ciphers, MACs, and Kex.
  - Added explicit cases for FIPS and non-FIPS mode (as well as reasonable default cases for RHEL7 and below).
  - Updated to use the new augeasproviders module dependencies.
  - Added a function *ssh_format_host_entry_for_sorting()* that will properly sort SSH *Host* entries for inclusion with concat.
- pupmod-stunnel
  - Had a variable **options** in *stunnel.erb* that should have been scoped as **@options**.
- pupmod-sudo
  - Removed all reliance on the 'lsb*' facts since some users do not wish to install the prerequisite RPMs for LSB compliance.
- pupmod-sudosh
  - Change the call to the *rsyslog* init script to the *service* command to seamlessly support both RHEL6 and RHEL7.
- pupmod-sysctl
  - Removed support for the old parsed-file provider and moved to using the new Augeas-based provider.
- pupmod-tftpboot
  - Purging of non-Puppet-managed items in *pxelinux.cfg* is now optional.
- pupmod-simp-tpm
  - IMA is disabled by default.
- simp-gpgkeys
  - Ensure that the keys are set in the correct locations for the target SIMP distribution.
- simp-rsync
  - Removed spurious install messages.

- simp-util
  - Fixed the targets of unpack_dvd.
  - Added a **use_fips** boolean to *simp config*
- pupmod-xinetd
  - Fixed: The default log_type should be 'SYSLOG authpriv' instead of 'SYSLOG daemon info'.
- pupmod-vnc
  - Removed banners that broke some vnc clients.
- simp-cli
  - *simp config -a ANSWERFILE* fails when an item has no answer
  - *simp config -A ANSWERFILE* prompts when an an item has no answer
  - The misleading *–help* documentation for *-ff* has been removed
  - The Config::Item *use_fips* now echoes its command unless *@silent*
  - The *simp doc* command path to the documentation has been corrected.
  - General usability improvements.
- DVD
  - NetworkManager-wait-online is now set by default in the ISO supplied kickstart images. Without ths, it is possible for the 'runpppet' script to attempt to run prior to the network being initialized.
  - A default IP is no longer provided when booting from the ISO; *simp config* will set the network properly.
  - The default kickstart no longer attempts to chkconfig any services in the %post section.

New Features

- pupmod-auditd
  - Completely overhauled the module with a focus on better acceptance testing and format compliance.
- pupmod-augeasproviders
  - This was updated to 2.1.3.
  - The update to 2.1.3 caused the addition of all of the pupmod-augeasproviders modules below.
- augeasproviders_apache
  - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_base
  - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_core
  - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_grub
  - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_mounttab
  - Imported 2.1.3 to support the Augeasproviders stack.

- augeasproviders_nagios
  - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_pam
  - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_postgresql
  - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_puppet
  - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_shellvar
  - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_ssh
  - Imported 2.1.3 to support the Augeasproviders stack.
- augeasproviders_sysctl
  - Imported 2.1.3 to support the Augeasproviders stack.
- pupmod-augeasproviders
  - This was updated to 2.1.3.
  - The update to 2.1.3 caused the addition of all of the pupmod-augeasproviders modules below.
- pupmod-cgroups
  - Added acceptance tests.
- pupmod-common => Deprecated - Replaced by pupmod-simplib!
- pupmod-simplib
  - Created parse_hosts function.
  - Added full tests for evaluating the ability to toggle FIPS mode.
- pupmod-richardc-datacat
  - Incorporated the *richardc/datacat* module into the core for user convenience.
- pupmod-freeradius
  - Split the Freeradius module based on version so that it can be properly selected against the *installed* version of Freeradius. This may take two runs to coalesce.
- pupmod-puppetlabs-inifile
  - Updated to version 1.2.0.
- pupmod-puppetlabs-puppetdb
  - Updated to version 5.0.0-0.
- pupmod-simp-kibana
  - Add Kibana dashboards to the Kibana module.
  - Allows users to apply default SIMP kibana Dashboards.
- pupmod-simp-logstash

  - Integrated SIMP and Electrical Logstash modules.
  - Changes the existing Logstash module to allow users to apply default SIMP filters.
- pupmod-pki
  - Now generate a system RSA public key against the passed private key.
- pupmod-puppetlabs-postgresql
  - Initial import of the Puppet Labs PostgreSQL module.
  - Modifications were made to support the SIMP concat.
- pupmod-puppetlabs-puppetdb
  - New import of the Puppet Labs PuppetDB module.
- pupmod-simp-rsyslog
  - Module has been rewritten to support rsyslog 7.4.
- pupmod-simp-simp
  - Set the SELinux Boolean 'use_nfs_home_dirs' to 'on' if a remote NFS server is used for home directories.
  - The 'runpuppet' script was modified to run 'fixfiles' on systems prior to the final puppet runs since RHEL7, in some cases, does not appear to honor the '/.autorelabel' file.
- pupmod-puppetlabs-stdlib
  - Updated to version 4.5.1.
- pupmod-sysctl
  - Moved the configuration file updates from sysctl.conf to sysctl.d/20-simp.conf to use the latest update mechanisms.
- pupmod-tftpboot
  - Updated to use native packages and pull as muchs possible.
- simp-doc
  - Updated tables across the board to be more readable.
  - Updated documentation relating to user management and user key management using SSH.
  - Rebranded the documentation and updated the color scheme.
  - Updated the default system passwords.
- simp-rsync
  - Content has been restructured to eliminate licensing conflicts.
  - ClamAV has been refactored into a separate (GPL) package.
- simp-utils
  - simp config was rewritten to allow for new features and flexibility.
  - Now provided as a Ruby gem "simp-cli".
- Mcollective
  - Mcollective is now available to be installed and used with SIMP. It uses SSL/TLS along with user certificates for proper encryption and authentication.
- PuppetDB

  - PuppetDB is now supported by SIMP and installed by default.

- Puppetserver
  - The puppet master service has been replaced by the puppetserver service. This is a major rewrite by Puppetlabs. Puppetserver scales better for larger agent deployments with a single puppet master.
  - Uses Environments by default, this allows for tools such as r10K. Production environment is a link to simp by default.
- Facter 2.4
  - Facter now returns the following facts as their actual boolean or integer values, instead of converting them into strings:

    activeprocessorcount is_virtual mtu_<INTERFACE> physicalprocessorcount processorcount selinux_enforced selinux sp_number_processors sp_packages

Known Bugs

- There is a symlink that is created at /etc/puppet/environments/simp/simp which should not be in place. This is being tracked as SIMP-661
- SSSD is currently broken and will allow logins via SSH even if your password has expired. This has been noted by Red Hat and is in the pipeline.
- If you are running libvirtd, when svckill runs it will always attempt to kill dnsmasq unless you are deliberately trying to run the dnsmasq service. This does *not* actually kill the service but is, instead, an error of the startup script and causes no damage to your system.

3.17 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 reproducability.

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.

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.

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.

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

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.

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

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

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.

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

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 standard of note in SIMP is FIPS 140-2

Source: FIPS Publications

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.

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 configuration data, built to make *Puppet* better and let you set node-specific data without repeating yourself.

Source: Hiera Overview

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.

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.

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.

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.

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

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.

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.

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.

PSSH, Parallel Secure Shell A tool that provides parallel versions of OpenSSH and other related tools.

Puppet An Open Source configuration management tool written and maintained by Puppet Labs. 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.

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.

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 *RHEL*.

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.

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.

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.

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*.

SSH, Secure Shell An application for secure data communication, remote shell services, or command execution between networked computers. SSH utilitizes 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*.

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.

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.

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*.

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".

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.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

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*.

3.18 Indices and tables

- genindex
- search

CHAPTER 4

SIMP Security Concepts

This is the 5.1.0-0 release of SIMP compatible with the 7.1 release of CentOS and Red Hat Enterprise Linux (RHEL).

This document provides a foundational Security Concept of Operations for the SIMP framework.

Contents:

4.1 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 Controls Traceability Matrix (SCTM).

When possible, the 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.

4.2 Technical Security

This chapter contains SIMP security concepts that are related to the technical security controls described in *NIST 800-53*.

4.2.1 Identification and Authentication

This section addresses the identification and authentication of users and devices.

4.2.2 User Identification and Authentication

Identification and authentication of system and service users can occur at the system level or globally in the SIMP architecture. While local accounts and groups can be created manually, the SIMP team suggests adding users via the `/etc/puppet/localusers` file or by 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]

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 Lightweight 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)].

4.2.3 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, IA-3(3)]

4.2.4 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-4(e)]

4.2.5 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 log on. 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. By default, SIMP requires 14-character passwords with at least one character from three of the four designated categories (i.e., upper case letters, lower case letters, numbers, or special characters), and no more than three consecutive characters from each category. [IA-5, IA-5(1), IA-5(4)] Password ageing 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.

4.2.6 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.

4.2.7 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. 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)]
- Password expiration. Local accounts expire 35 days after password expiration. [AC-2(3)]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 for ldap accounts and `/var/log/audit.log` for local accounts. [AC-2(4)]
- Shell session timeouts after 15 minutes of inactivity. [AC-2(5)] 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)] By default, SIMP will have an administrators groups that has the ability to run `sudosh`. Implementations should further define administrators or user groups and limit them with the Puppet `sudo` class.

4.2.8 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]

4.2.9 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]

Default Server Ports

Application	Direction	Protocol	Transport	Ports	Comment
Puppet	Localhost	HTTP	TCP	8140	The port upon which the Puppet master 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	80	This is used for YUM and is unencrypted, since YUM will not work otherwise.
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.

4.2.10 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 posixGroups in LDAP for full operating System support. [AC-5]

4.2.11 Least Privilege

SIMP does not allow `root` to directly SSH into a system. The `root` user must be at 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` or `sudosh`. [AC-6]

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]

4.2.12 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]
- System banners are presented to a user both before and after logging on. The default banner should be customized for each implementation. [AC-8]
- 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 and AC-9(1)]
- 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 operational settings, this default value is reasonable. [AC-10]
- 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]

4.2.13 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]

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

4.2.14 Security Attributes

SELinux is now available 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]

4.2.15 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 and sudosh. [AC-17]

4.2.16 Systems and Communications Protection

The following sections provide information regarding application partitioning, shared resources, and various levels of protection for systems and communications.

4.2.17 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 or sudosh privileges via Puppet.

4.2.18 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]

4.2.19 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 syncookies. [SC-5]

4.2.20 Boundary Protection

SIMP does not provide boundary protection. [SC-7]

4.2.21 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, SSL, and TLS all provide data transmission integrity and confidentiality. The software that controls them on Red Hat and CentOS are OpenSSH and OpenSSL. 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 removed. [SC-8, SC-9, SC-23, SC-7]

4.2.22 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. Grub passwords are also set to prevent unauthorized modifications to boot parameters. [SC-24]

4.2.23 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. More information on the Puppet CA can be found in the Puppet Labs security documentation. [SC-17, SC-13]

> **Warning:** Fake CA certificates should not be used in an operational setting.

4.2.24 Mobile Code

SIMP does not use mobile code; however, there are not any particular tools that will prevent its use. [SC-18]

4.2.25 Protection of Information at Rest

There are no additional protections for information at rest beyond operating system capabilities in SIMP. There are also no measures in place to encrypt or sign data before transmission. Each implementation should determine how to further protect information at rest. [SC-28]

4.2.26 Audit and Accountability

This section discusses the content, storage, and protection of auditable events.

4.2.27 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, which 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]

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)] Privileged commands are audited as part of the SIMP auditing configuration. This is accomplished by monitoring `sudo` commands with auditd. Keystrokes for administrators that use `sudosh` are also logged. Each session can be replayed using `sudosh-replay`. [AU-2(4)]

4.2.28 Content of Audit Records

Audit records capture the following information [AU-3]:

- Date and Time
- UID and GID of the user performing the action
- Command
- Event ID
- Key
- Node Hostname/IP Address
- Login Session ID
- Executable

4.2.29 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 e-mail infrastructure is in place.) [AU-5(a), AU-5(1)]
- The log file rotates once it reaches 30MB.

4.2.30 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]

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. However, SIMP will not shut down, but will provide an alert via syslog when this happens. [AU-5(1)]

SIMP also comes with an optional module for the Elasticsearch/Logstash/Kibana (ELK) stack. These three open source tools can be combined to parse, index, and visualize logs. There are also SIMP provided dashboards for the Kibana web interface. Implementations can build their own dashboards to meet local security or functional needs for log reduction and management. [AU-6]

4.2.31 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.

4.2.32 Time Synchronization

Each SIMP client (including the Puppet Master) 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 Master 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 time stamps in audit logs to be accurate.

4.3 Operational Security

This chapter contains SIMP security concepts that are related to the operational security controls in NIST 800-53.

4.3.1 Configuration Management

This section describes the management of various configurations within SIMP.

Baseline Configurations

SIMP baselines include configuration settings and Puppet modules. Currently, baselines are maintained for both Red Hat/CentOS 6.x, and Red Hat/CentOS 7.x. Each configuration item that is managed by a Puppet module has an RPM installed on the Puppet Master 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 Master, which denotes the version number of SIMP that is installed. [CM-2, CM-2(2), CM-2(3), CM-6]

SIMP installs a minimal set of RPMs, which can be found in ?. RPMs, services, and IPtables rules all use a deny-all, but allow-by-exception module. 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. [CM-2(5)]

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 all (or selected) Puppet clients. All changes made are audited with auditd or are logged to other files via syslog. [CM-3(a), CM-3(3)]

UNIX systems are made up of hundreds of configuration files that can contain dozens of 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]

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, SI-2, SI-2(1), SI-2(4)]:

- **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.
- **PSSH:** Allow commands to run across a set of nodes with the PSSH utility. Through the use of keys, this becomes a powerful way to run a one-time operation against a large number of nodes.

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]

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] 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]

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 Master may not have the ability to retake control over that client. However, the Puppet Master 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, SI-7(1), SI-7(2), SI-7(3)]

4.3.2 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 implementation to decide how to distribute authentication information for remote maintenance. [MA-4, MA-4(1), MA-6]

4.3.3 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]

4.3.4 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.

4.3.5 System Backup

SIMP comes with a module called `backuppc`. This module provides a base configuration of the BackupPC software and allows Puppet servers and clients to perform backups.

4.4 Information System Management

This chapter contains SIMP security concepts that are related to the management security controls in NIST 800-53.

4.4.1 Risk Assessment

This section describes the process of identifying risks within a system.

4.4.2 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	Possible Mitigations
Disabling Puppet: This can cause the clients to be out of sync with the Puppet Master.	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 Master 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 LDIFs 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

4.4.3 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, RA-5]

4.4.4 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]

4.5 Security Concepts Appendices

4.5.1 Default Files Watched by AIDE

```
/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
/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
/etc/cups LSPP
!/var/log/and-httpd
```

4.5.2 Audit Rules

```
## 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 16394

## Set failure mode to panic
# Default: 2
-f 2

## 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!=4294967295

# 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
# CCE-26712-0
# CCE-26651-0
-a always,exit -F arch=b64 -S creat -S mkdir -S mknod -S link -S symlink -S mkdirat -S mknodat -S lin
-a always,exit -F arch=b64 -S creat -S mkdir -S mknod -S link -S symlink -S mkdirat -S mknodat -S lin
-a always,exit -F arch=b32 -S creat -S mkdir -S mknod -S link -S symlink -S mkdirat -S mknodat -S lin
-a always,exit -F arch=b32 -S creat -S mkdir -S mknod -S link -S symlink -S mkdirat -S mknodat -S lin

-a always,exit -F perm=a -F exit=-EACCES -k access
-a always,exit -F perm=a -F exit=-EPERM -k access

# Permissions auditing
# CCE-26280-8
# CCE-27173-4
# CCE-27174-2
# CCE-27175-9
# CCE-27177-5
# CCE-27178-3
# CCE-27179-1
# CCE-27180-9
# CCE-27181-7
# CCE-27182-5
# CCE-27183-3
```

```
# CCE-27184-1
# CCE-27185-8
-a always,exit -F arch=b64 -S chown -S fchmod -S fchmodat -S fchown -S fchownat -S lchown -S setxattr
-a always,exit -F arch=b32 -S chown -S fchmod -S fchmodat -S fchown -S fchownat -S lchown -S setxattr

# 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
-a always,exit -F arch=b32 -F auid!=0 -F uid=0 -S capset -S mknod -S pivot_root -S quotactl -S setsid

# Audit the execution of suid and sgid binaries.
# CCE-26457-2
-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.
# CCE-26611-4
-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
# CCE-27172-6
# CCE-27203-9
# CCE-27169-2
# CCE-27170-0
-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

# CCE-27172-6
-w /etc/localtime -p wa -k audit_time_rules

## Things that could affect system locale
# CCE-26648-6
-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
-w /etc/sysconfig/network -p wa -k audit_network_modifications

# Mount options.
# CCE-26573-6
-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

# CCE-26664-3
-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

# CCE-26657-7
-w /etc/selinux/ -p wa -k MAC-policy

# CCE-26691-6
-w /var/log/faillog -p wa -k logins
-w /var/log/lastlog -p wa -k logins

# CCE-26610-6
-w /var/run/utmp -p wa -k session
-w /var/run/btmp -p wa -k session
-w /var/run/wtmp -p wa -k session

# CCE-26662-7
-w /etc/sudoers -p wa -k CFG_sys

# Generally good things to audit.
-w /var/spool/at -p wa -k CFG_sys
-w /etc/at.deny -p wa -k CFG_sys
-w /etc/cron.deny -p wa -k CFG_cron
-w /etc/cron.d -p wa -k CFG_cron
-w /etc/cron.daily -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/anacrontab -p wa -k CFG_cron
-w /etc/login.defs -p wa -k CFG_sys
-w /etc/securetty -p wa -k CFG_sys
-w /etc/shells -p wa -k CFG_shell
-w /etc/profile -p wa -k CFG_shell
-w /etc/bashrc -p wa -k CFG_shell
-w /etc/csh.cshrc -p wa -k CFG_shell
-w /etc/csh.login -p wa -k CFG_shell
-w /etc/sysconfig -p wa -k CFG_sys
-w /etc/inittab -p wa -k CFG_sys
-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/xinetd.d -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/sysctl.conf -p wa -k CFG_sys
-w /etc/modprobe.d/00_simp_blacklist.conf -p wa -k CFG_sys
-w /etc/modprobe.conf.d -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/aliases -p wa -k CFG_sys
-w /etc/ssh/sshd_config -p wa -k CFG_sys
-w /etc/issue -p wa -k CFG_sys
-w /etc/issue.net -p wa -k CFG_sys
-w /etc/snmp/snmpd.conf -p wa -k CFG_sys
```

```
-w /etc/resolv.conf -p wa -k CFG_sys
-w /etc/nsswitch.conf -p wa -k CFG_sys
-w /etc/host.conf -p wa -k CFG_sys
-w /etc/krb5.conf -p wa -k CFG_sys
-w /etc/initlog.conf -p wa -k CFG_sys
-w /etc/default -p wa -k CFG_sys
-w /lib/firmware/microcode.dat -p wa -k CFG_sys
-w /etc/fstab -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/exports -p wa -k CFG_sys
-w /etc/yum.conf -p wa -k yum-config
-w /etc/yum.repos.d -p wa -k yum-config
-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/ntp.conf -p wa -k CFG_ntp
-w /etc/ntp/keys -p wa -k CFG_ntp
-w /etc/ntp/ntpservers -p wa -k CFG_ntp
-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/ws69.kw.awesome.sauce.pem -p wa -k PKI
-w /etc/pki/public/ws69.kw.awesome.sauce.pub -p wa -k PKI
-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

-w /etc/security/access.conf -p wa -k CFG_security
-w /etc/security/console.perms -p wa -k CFG_security
-w /etc/security/chroot.conf -p wa -k CFG_security
-w /etc/security/limits.conf -p wa -k CFG_security
-w /etc/security/group.conf -p wa -k CFG_security
-w /etc/security/time.conf -p wa -k CFG_security
-w /etc/security/pam_env.conf -p wa -k CFG_security
-w /etc/grub.conf -p wa -k CFG_grub
-w /etc/xinted.conf -p wa -k CFG_xinted
-w /etc/services -p wa -k CFG_services
-w /etc/default/nss -p wa -k CFG_defaults
-w /etc/xinetd.d/chargen -p wa -k CFG_xinted.d
-w /etc/xinetd.d/chargen-udp -p wa -k CFG_xinted.d
-w /etc/xinetd.d/cups-lpd -p wa -k CFG_xinted.d
-w /etc/xinetd.d/daytime -p wa -k CFG_xinted.d
-w /etc/xinetd.d/daytime-udp -p wa -k CFG_xinted.d
-w /etc/xinetd.d/echo -p wa -k CFG_xinted.d
-w /etc/xinetd.d/echo-udp -p wa -k CFG_xinted.d
-w /etc/xinetd.d/rsync -p wa -k CFG_xinted.d
-w /etc/xinetd.d/time -p wa -k CFG_xinted.d
-w /etc/xinetd.d/time-udp -p wa -k CFG_xinted.d
-w /usr/share/gdm/defaults.conf -p wa -k CFG_sys
-w /etc/init/ -p wa -k CFG_upstart
# CCE-26612-2 deliberiately ignored so that audit rules may be manipulated by
```

```
# Puppet.
```

4.5.3 Default Kickstart Files

Default Puppet Master Kickstart file (contains default RPMs)

```
#
# Use the following Ruby code to generate your password hashes:
#   ruby -r 'digest/sha2' -e 'puts "password".crypt("$6$" + rand(36**8).to_s(36))'
#
# Use the following command to generate your grub password hash:
#   grub2-mkpasswd-pbkdf2
#
# Replace the following strings in this file
# #BOOTPASS# - Your hashed bootloader password
# #ROOTPASS# - Your hashed root password
# #KSSERVER# - The IP address of your YUM server
# #YUMSERVER# - The IP address of your YUM server
# #LINUXDIST# - The LINUX Distribution you are kickstarting
#        - Current CASE SENSITIVE options: RedHat CentOS

authconfig --enableshadow --passalgo=sha512
bootloader --location=mbr --append="console=ttyS1,57600 console=tty1" --iscrypted --password=#BOOTPAS
rootpw --iscrypted #ROOTPASS#
zerombr
firewall --enabled --ssh
firstboot --disable
logging --level=info
network --bootproto=dhcp
reboot
selinux --permissive
timezone --utc GMT

install
skipx

%include /tmp/repo-include

text
keyboard us
lang en_US
url --url http://#KSSERVER#/yum/#LINUXDIST#/7/x86_64

%include /tmp/part-include

%packages --nobase
-sendmail
-sysklogd
acl
aide
anacron
audit
bzip2
coolkey
crontabs
cryptsetup-luks
dhclient
```

```
git
gnupg
iptables
iptables-ipv6
irqbalance
krb5-workstation
libaio
libutempter
logrotate
logwatch
lsof
lsscsi
mdadm
microcode_ctl
mutt
net-snmp
net-tools
netlabel_tools
ntp
openssh-clients
openssh-server
pam_krb5
pam_pkcs11
pciutils
psacct
quota
redhat-lsb
rpm
rsync
rsyslog
smartmontools
sssd
stunnel
subversion
sudo
sysstat
tcp_wrappers
tmpwatch
unzip
usbutils
vim-enhanced
vlock
wget
which
zip
# Puppet stuff
rsync
facter
puppet

# In case of broken repo, these should be installed.
hdparm
kbd
libhugetlbfs
policycoreutils
prelink
rootfiles
selinux-policy-targeted
```

```
setserial
sysfsutils
udftools

# Don't install these
-rhn-check
-rhn-setup
-rhnsd
-subscription-manager
-yum-rhn-plugin
%end

%pre
ksserver="#KSSERVER#"
wget -O /tmp/diskdetect.sh http://$ksserver/ks/diskdetect.sh;
chmod 750 /tmp/diskdetect.sh;
/tmp/diskdetect.sh;
wget -O /tmp/repodetect.sh http://$ksserver/ks/repodetect.sh;
chmod 750 /tmp/repodetect.sh;
/tmp/repodetect.sh '7' $ksserver;
%end

%post
ostype="#LINUXDIST#"
if [ $ostype == "CentOS" ]; then
    sed -i '/enabled=/d' /etc/yum.repos.d/CentOS-Base.repo;
      sed -i '/\[.*\]/ a\
      enabled=0' /etc/yum.repos.d/CentOS-Base.repo;
      fi
      ksserver="#KSSERVER#"

# Notify users that bootstrap will run on firstboot
echo "Welcome to SIMP!  If this is firstboot, SIMP bootstrap is scheduled to run.
If this host is not autosigned by Puppet, sign your Puppet certs to begin bootstrap.
Otherwise, it should already be running! Tail /root/puppet.bootstrap.log for details.
Wait for completion and reboot.

To remove this message, delete /root/.bootstrap_msg" > /root/.bootstrap_msg
sed -i "2i if [ -f /root/.bootstrap_msg ]\nthen\n  cat /root/.bootstrap_msg\nfi" /root/.bashrc
source /root/.bashrc

# Enable the firstboot bootstrapping script.
wget --no-check-certificate -O /etc/init.d/runpuppet http://$ksserver/ks/runpuppet;
chmod 700 /etc/rc.d/init.d/runpuppet;
chkconfig --add runpuppet;
chkconfig --level 35 runpuppet on;
%end
```

4.5.4 SIMP RPMs

Red Hat Enterprise Linux

Name	Source
activemq-5.9.1-2.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/dependencies/x86_64/activemq-5.9.1-2.el

Table 4.1 – continued from previous page

Name	Source
activemq-info-provider-5.9.1-2.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/dependencies/x86_64/activemq-info-prov
apr-util-1.5.2-6.el7.x86_64.rpm	Red Hat Optional Repository
apr-util-ldap-1.5.2-6.el7.x86_64.rpm	Red Hat Optional Repository
boost-regex-1.53.0-23.el7.x86_64.rpm	Red Hat Updates Repository
cfacter-0.3.0-1.el7.x86_64.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/cfacter-0.3.0-1.el7.x86_
chkrootkit-0.50-4el7.x86_64.rpm	https://dl.bintray.com/simp/5.1.X-Ext/chkrootkit-0.50-4el7.x86_64.rpm
clamav-0.98.7-1.el7.x86_64.rpm	http://lug.mtu.edu/epel/7/x86_64/c/clamav-0.98.7-1.el7.x86_64.rpm
clamav-data-0.98.7-1.el7.noarch.rpm	http://lug.mtu.edu/epel/7/x86_64/c/clamav-data-0.98.7-1.el7.noarch.rpm
clamav-data-empty-0.98.7-1.el7.noarch.rpm	http://lug.mtu.edu/epel/7/x86_64/c/clamav-data-empty-0.98.7-1.el7.noarc
clamav-devel-0.98.7-1.el7.x86_64.rpm	http://lug.mtu.edu/epel/7/x86_64/c/clamav-devel-0.98.7-1.el7.x86_64.rpm
clamav-filesystem-0.98.7-1.el7.noarch.rpm	http://lug.mtu.edu/epel/7/x86_64/c/clamav-filesystem-0.98.7-1.el7.noarch
clamav-lib-0.98.7-1.el7.x86_64.rpm	http://lug.mtu.edu/epel/7/x86_64/c/clamav-lib-0.98.7-1.el7.x86_64.rpm
clamav-scanner-0.98.7-1.el7.noarch.rpm	http://lug.mtu.edu/epel/7/x86_64/c/clamav-scanner-0.98.7-1.el7.noarch.rp
clamav-scanner-systemd-0.98.7-1.el7.noarch.rpm	http://lug.mtu.edu/epel/7/x86_64/c/clamav-scanner-systemd-0.98.7-1.el7.
clamav-scanner-sysvinit-0.98.7-1.el7.noarch.rpm	http://lug.mtu.edu/epel/7/x86_64/c/clamav-scanner-sysvinit-0.98.7-1.el7.r
clamav-server-0.98.7-1.el7.x86_64.rpm	http://lug.mtu.edu/epel/7/x86_64/c/clamav-server-0.98.7-1.el7.x86_64.rpm
clamav-server-systemd-0.98.7-1.el7.noarch.rpm	http://lug.mtu.edu/epel/7/x86_64/c/clamav-server-systemd-0.98.7-1.el7.no
clamav-server-sysvinit-0.98.7-1.el7.noarch.rpm	http://lug.mtu.edu/epel/7/x86_64/c/clamav-server-sysvinit-0.98.7-1.el7.no
clamav-update-0.98.7-1.el7.x86_64.rpm	http://lug.mtu.edu/epel/7/x86_64/c/clamav-update-0.98.7-1.el7.x86_64.rp
ctags-5.8-13.el7.x86_64.rpm	Red Hat Updates Repository
dracut-033-241.el7_1.5.x86_64.rpm	Red Hat Updates Repository
dracut-config-rescue-033-241.el7_1.5.x86_64.rpm	Red Hat Updates Repository
dracut-fips-033-241.el7_1.5.x86_64.rpm	Red Hat Updates Repository
dracut-fips-aesni-033-241.el7_1.5.x86_64.rpm	Red Hat Updates Repository
dracut-network-033-241.el7_1.5.x86_64.rpm	Red Hat Updates Repository
elasticsearch-1.3.2.noarch.rpm	https://download.elastic.co/elasticsearch/elasticsearch/elasticsearch-1.3.2.
elasticsearch-curator-1.1.1-0el7.noarch.rpm	https://dl.bintray.com/simp/5.1.X-Ext/elasticsearch-curator-1.1.1-0el7.noa
es2unix-1.6.1-0el7.noarch.rpm	https://dl.bintray.com/simp/5.1.X-Ext/es2unix-1.6.1-0el7.noarch.rpm
etcd-2.0.11-0.SIMP.el7.x86_64.rpm	https://dl.bintray.com/simp/5.1.X-Ext/etcd-2.0.11-0.SIMP.el7.x86_64.rpm
facter-2.4.4-1.el7.x86_64.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/facter-2.4.4-1.el7.x86_6
gweb-2.1.8-1.noarch.rpm	https://dl.bintray.com/simp/5.1.X-Ext/gweb-2.1.8-1.noarch.rpm
haveged-1.9.1-1.el7.x86_64.rpm	http://lug.mtu.edu/epel/7/x86_64/h/haveged-1.9.1-1.el7.x86_64.rpm
hmaccalc-0.9.13-4.el7.x86_64.rpm	Red Hat Updates Repository
incron-0.5.10-8.el7.x86_64.rpm	http://lug.mtu.edu/epel/7/x86_64/i/incron-0.5.10-8.el7.x86_64.rpm
kernel-3.10.0-229.14.1.el7.x86_64.rpm	Red Hat Updates Repository
kibana-3.1.0.SIMP-0.noarch.rpm	https://dl.bintray.com/simp/5.1.X-Ext/kibana-3.1.0.SIMP-0.noarch.rpm
libarchive-devel-3.1.2-7.el7.x86_64.rpm	Red Hat Optional Repository
libconfuse-2.7-7.el7.x86_64.rpm	http://lug.mtu.edu/epel/7/x86_64/l/libconfuse-2.7-7.el7.x86_64.rpm
libev-4.15-3.el7.x86_64.rpm	http://lug.mtu.edu/epel/7/x86_64/l/libev-4.15-3.el7.x86_64.rpm
libselinux-2.2.2-6.el7.x86_64.rpm	Red Hat Updates Repository
libselinux-python-2.2.2-6.el7.x86_64.rpm	Red Hat Updates Repository
libselinux-ruby-2.2.2-6.el7.x86_64.rpm	Red Hat Updates Repository
libselinux-static-2.2.2-6.el7.x86_64.rpm	Red Hat Optional Repository
libselinux-utils-2.2.2-6.el7.x86_64.rpm	Red Hat Updates Repository
libsepol-2.1.9-3.el7.x86_64.rpm	Red Hat Updates Repository
libsepol-static-2.1.9-3.el7.x86_64.rpm	Red Hat Optional Repository
libyaml-0.1.4-11.el7_0.x86_64.rpm	Red Hat Updates Repository
linux-firmware-20140911-0.1.git365e80c.el7.noarch.rpm	Red Hat Updates Repository
logstash-1.4.2-1_2c0f5a1.noarch.rpm	https://download.elasticsearch.org/logstash/logstash/packages/centos/logs

Table 4.1 – continued from previous page

Name	Source
logstash-contrib-1.4.2-1_efd53ef.noarch.rpm	https://download.elastic.co/logstash/logstash/packages/centos/logstash-co
mcollective-2.8.4-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-2.8.4-1.el7.n
mcollective-actionpolicy-auth-2.1.0-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-actionpolicy
mcollective-client-2.8.4-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-client-2.8.4-
mcollective-common-2.8.4-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-common-2.8
mcollective-filemgr-agent-1.0.2-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-filemgr-age
mcollective-filemgr-client-1.0.2-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-filemgr-clie
mcollective-filemgr-common-1.0.2-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-filemgr-com
mcollective-iptables-agent-3.0.2-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-iptables-age
mcollective-iptables-client-3.0.2-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-iptables-clie
mcollective-iptables-common-3.0.2-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-iptables-com
mcollective-nettest-agent-3.0.4-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-nettest-agen
mcollective-nettest-client-3.0.4-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-nettest-clien
mcollective-nettest-common-3.0.4-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-nettest-com
mcollective-nrpe-agent-3.1.0-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-nrpe-agent-
mcollective-nrpe-client-3.1.0-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-nrpe-client-
mcollective-nrpe-common-3.1.0-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-nrpe-comm
mcollective-package-agent-4.4.0-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-package-ag
mcollective-package-client-4.4.0-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-package-cli
mcollective-package-common-4.4.0-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-package-co
mcollective-puppet-agent-1.10.0-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-puppet-ager
mcollective-puppet-client-1.10.0-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-puppet-clien
mcollective-puppet-common-1.10.0-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-puppet-com
mcollective-service-agent-3.1.3-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-service-age
mcollective-service-client-3.1.3-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-service-clie
mcollective-service-common-3.1.3-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-service-com
mcollective-shell-agent-0.0.2-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-shell-agent-
mcollective-shell-client-0.0.2-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-shell-client-
mcollective-shell-common-0.0.2-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-shell-comm
mcollective-sshkey-security-0.5.0-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-sshkey-secu
mcollective-sysctl-data-2.0.1-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-sysctl-data-
mod_ldap-2.4.6-31.el7.x86_64.rpm	Red Hat Optional Repository
openssh-6.6.1p1-12.el7_1.x86_64.rpm	Red Hat Updates Repository
openssh-askpass-6.6.1p1-12.el7_1.x86_64.rpm	Red Hat Updates Repository
openssh-clients-6.6.1p1-12.el7_1.x86_64.rpm	Red Hat Updates Repository
openssh-keycat-6.6.1p1-12.el7_1.x86_64.rpm	Red Hat Updates Repository
openssh-ldap-6.6.1p1-12.el7_1.x86_64.rpm	Red Hat Optional Repository
openssh-server-6.6.1p1-12.el7_1.x86_64.rpm	Red Hat Updates Repository
pdsh-2.29-1el7.x86_64.rpm	https://dl.bintray.com/simp/5.1.X-Ext/pdsh-2.29-1el7.x86_64.rpm
pdsh-debuginfo-2.29-1el7.x86_64.rpm	https://dl.bintray.com/simp/5.1.X-Ext/pdsh-debuginfo-2.29-1el7.x86_64.r
pdsh-mod-dshgroup-2.29-1el7.x86_64.rpm	https://dl.bintray.com/simp/5.1.X-Ext/pdsh-mod-dshgroup-2.29-1el7.x86_
pdsh-mod-machines-2.29-1el7.x86_64.rpm	https://dl.bintray.com/simp/5.1.X-Ext/pdsh-mod-machines-2.29-1el7.x86_
pdsh-mod-netgroup-2.29-1el7.x86_64.rpm	https://dl.bintray.com/simp/5.1.X-Ext/pdsh-mod-netgroup-2.29-1el7.x86_
pdsh-rcmd-exec-2.29-1el7.x86_64.rpm	https://dl.bintray.com/simp/5.1.X-Ext/pdsh-rcmd-exec-2.29-1el7.x86_64.r
pdsh-rcmd-ssh-2.29-1el7.x86_64.rpm	https://dl.bintray.com/simp/5.1.X-Ext/pdsh-rcmd-ssh-2.29-1el7.x86_64.rp
pssh-2.3.1.SIMP-5.el7.noarch.rpm	https://dl.bintray.com/simp/5.1.X-Ext/pssh-2.3.1.SIMP-5.el7.noarch.rpm
puppet-3.8.1-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/puppet-3.8.1-1.el7.noarc
puppet-server-3.8.1-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/puppet-server-3.8.1-1.el
puppetdb-2.3.8-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/puppetdb-2.3.8-1.el7.no

Table 4.1 – continued from previous page

Name	Source
puppetdb-terminus-2.3.8-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/puppetdb-terminus-2.3.8
puppetlabs-release-7-11.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/puppetlabs-release-7-11
puppetserver-1.1.1-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/puppetserver-1.1.1-1.el7
python-elasticsearch-1.2.0-0.el7.centos.noarch.rpm	https://dl.bintray.com/simp/5.1.X-Ext/python-elasticsearch-1.2.0-0.el7.cen
python-redis-2.10.3-1.el7.noarch.rpm	http://lug.mtu.edu/epel/7/x86_64/p/python-redis-2.10.3-1.el7.noarch.rpm
python-simplejson-3.3.3-1.el7.x86_64.rpm	http://lug.mtu.edu/epel/7/x86_64/p/python-simplejson-3.3.3-1.el7.x86_64
python-unittest2-0.5.1-6.el7.noarch.rpm	http://lug.mtu.edu/epel/7/x86_64/p/python-unittest2-0.5.1-6.el7.noarch.rp
razor-server-1.0.1-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/razor-server-1.0.1-1.el7.
razor-torquebox-3.1.1.9-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/razor-torquebox-3.1.1.9-
rrdtool-1.4.8-8.el7.x86_64.rpm	Red Hat Updates Repository
ruby-augeas-0.4.1-3.el7.x86_64.rpm	http://yum.puppetlabs.com/el/7/dependencies/x86_64/ruby-augeas-0.4.1-3
ruby-ldap-0.9.16-1.el7.x86_64.rpm	http://lug.mtu.edu/epel/7/x86_64/r/ruby-ldap-0.9.16-1.el7.x86_64.rpm
ruby-rgen-0.6.5-2.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/dependencies/x86_64/ruby-rgen-0.6.5-2.e
ruby-shadow-2.2.0-2.el7.x86_64.rpm	http://yum.puppetlabs.com/el/7/dependencies/x86_64/ruby-shadow-2.2.0-
rubygem-deep_merge-1.0.0-2.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/dependencies/x86_64/rubygem-deep_mer
rubygem-ffi-1.4.0-2.el7.x86_64.rpm	http://yum.puppetlabs.com/el/7/dependencies/x86_64/rubygem-ffi-1.4.0-2
rubygem-highline-1.6.11-5.el7.noarch.rpm	http://lug.mtu.edu/epel/7/x86_64/r/rubygem-highline-1.6.11-5.el7.noarch
rubygem-net-ping-1.6.2-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/dependencies/x86_64/rubygem-net-ping-
rubygem-puppet-lint-1.1.0-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/dependencies/x86_64/rubygem-puppet-lin
rubygem-rake-0.9.6-25.el7_1.noarch.rpm	Red Hat Optional Repository
rubygem-rake-compiler-0.9.3-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/dependencies/x86_64/rubygem-rake-com
rubygem-stomp-1.3.2-1.el7.noarch.rpm	http://lug.mtu.edu/epel/7/x86_64/r/rubygem-stomp-1.3.2-1.el7.noarch.rpm
rubygem-stomp-doc-1.3.2-1.el7.noarch.rpm	http://lug.mtu.edu/epel/7/x86_64/r/rubygem-stomp-doc-1.3.2-1.el7.noarch
scap-security-guide-0.1.19-2.el7.noarch.rpm	Red Hat Updates Repository
hiera-3.0.2-1.el7.noarch.rpm	https://dl.bintray.com/simp/5.1.X/hiera-3.0.2-1.el7.noarch.rpm
simp-lastbind-2.4.23-0.x86_64.rpm	https://dl.bintray.com/simp/5.1.X-Ext/simp-lastbind-2.4.23-0.x86_64.rpm
simp-ppolicy-check-password-2.4.39-0el7.x86_64.rpm	https://dl.bintray.com/simp/5.1.X-Ext/simp-ppolicy-check-password-2.4.3
source-highlight-3.1.6-6.el7.x86_64.rpm	Red Hat Optional Repository
sudosh2-1.0.2-2el7.x86_64.rpm	https://dl.bintray.com/simp/5.1.X-Ext/sudosh2-1.0.2-2el7.x86_64.rpm
syslinux-tftpboot-4.05-12.el7.x86_64.rpm	Red Hat Optional Repository

Community ENTerprise Operating System

Name	Source
activemq-5.9.1-2.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/dependencies/x86_64/activemq-5.9.1-2.el
activemq-info-provider-5.9.1-2.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/dependencies/x86_64/activemq-info-prov
boost-regex-1.53.0-23.el7.x86_64.rpm	http://mirrors.advancedhosters.com/centos/7.1.1503/os/x86_64/Packages/
cfacter-0.3.0-1.el7.x86_64.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/cfacter-0.3.0-1.el7.x86_
chkrootkit-0.50-4el7.x86_64.rpm	https://dl.bintray.com/simp/5.1.X-Ext/chkrootkit-0.50-4el7.x86_64.rpm
clamav-0.98.7-1.el7.x86_64.rpm	http://lug.mtu.edu/epel/7/x86_64/c/clamav-0.98.7-1.el7.x86_64.rpm
clamav-data-0.98.7-1.el7.noarch.rpm	http://lug.mtu.edu/epel/7/x86_64/c/clamav-data-0.98.7-1.el7.noarch.rpm
clamav-data-empty-0.98.7-1.el7.noarch.rpm	http://lug.mtu.edu/epel/7/x86_64/c/clamav-data-empty-0.98.7-1.el7.noarc
clamav-devel-0.98.7-1.el7.x86_64.rpm	http://lug.mtu.edu/epel/7/x86_64/c/clamav-devel-0.98.7-1.el7.x86_64.rpm
clamav-filesystem-0.98.7-1.el7.noarch.rpm	http://lug.mtu.edu/epel/7/x86_64/c/clamav-filesystem-0.98.7-1.el7.noarch
clamav-lib-0.98.7-1.el7.x86_64.rpm	http://lug.mtu.edu/epel/7/x86_64/c/clamav-lib-0.98.7-1.el7.x86_64.rpm
clamav-scanner-0.98.7-1.el7.noarch.rpm	http://lug.mtu.edu/epel/7/x86_64/c/clamav-scanner-0.98.7-1.el7.noarch.rp
clamav-scanner-systemd-0.98.7-1.el7.noarch.rpm	http://lug.mtu.edu/epel/7/x86_64/c/clamav-scanner-systemd-0.98.7-1.el7.
clamav-scanner-sysvinit-0.98.7-1.el7.noarch.rpm	http://lug.mtu.edu/epel/7/x86_64/c/clamav-scanner-sysvinit-0.98.7-1.el7.n

Table 4.2 – continued from previous page

Name	Source
clamav-server-0.98.7-1.el7.x86_64.rpm	http://lug.mtu.edu/epel/7/x86_64/c/clamav-server-0.98.7-1.el7.x86_64.rp
clamav-server-systemd-0.98.7-1.el7.noarch.rpm	http://lug.mtu.edu/epel/7/x86_64/c/clamav-server-systemd-0.98.7-1.el7.n
clamav-server-sysvinit-0.98.7-1.el7.noarch.rpm	http://lug.mtu.edu/epel/7/x86_64/c/clamav-server-sysvinit-0.98.7-1.el7.n
clamav-update-0.98.7-1.el7.x86_64.rpm	http://lug.mtu.edu/epel/7/x86_64/c/clamav-update-0.98.7-1.el7.x86_64.rp
ctags-5.8-13.el7.x86_64.rpm	http://mirrors.advancedhosters.com/centos/7.1.1503/os/x86_64/Packages/
dracut-033-241.el7_1.5.x86_64.rpm	http://mirrors.advancedhosters.com/centos/7.1.1503/updates/x86_64/Pack
dracut-config-rescue-033-241.el7_1.5.x86_64.rpm	http://mirrors.advancedhosters.com/centos/7.1.1503/updates/x86_64/Pack
dracut-fips-033-241.el7_1.5.x86_64.rpm	http://mirrors.advancedhosters.com/centos/7.1.1503/updates/x86_64/Pack
dracut-fips-aesni-033-241.el7_1.5.x86_64.rpm	http://mirrors.advancedhosters.com/centos/7.1.1503/updates/x86_64/Pack
dracut-network-033-241.el7_1.5.x86_64.rpm	http://mirrors.advancedhosters.com/centos/7.1.1503/updates/x86_64/Pack
elasticsearch-1.3.2.noarch.rpm	https://download.elastic.co/elasticsearch/elasticsearch/elasticsearch-1.3.2.
elasticsearch-curator-1.1.1-0el7.noarch.rpm	https://dl.bintray.com/simp/5.1.X-Ext/elasticsearch-curator-1.1.1-0el7.noa
es2unix-1.6.1-0el7.noarch.rpm	https://dl.bintray.com/simp/5.1.X-Ext/es2unix-1.6.1-0el7.noarch.rpm
etcd-2.0.11-0.SIMP.el7.x86_64.rpm	https://dl.bintray.com/simp/5.1.X-Ext/etcd-2.0.11-0.SIMP.el7.x86_64.rpm
facter-2.4.4-1.el7.x86_64.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/facter-2.4.4-1.el7.x86_6
gweb-2.1.8-1.noarch.rpm	https://dl.bintray.com/simp/5.1.X-Ext/gweb-2.1.8-1.noarch.rpm
haveged-1.9.1-1.el7.x86_64.rpm	http://lug.mtu.edu/epel/7/x86_64/h/haveged-1.9.1-1.el7.x86_64.rpm
hiera-3.0.2-1.el7.noarch.rpm	https://dl.bintray.com/simp/5.1.X/hiera-3.0.2-1.el7.noarch.rpm
hmaccalc-0.9.13-4.el7.x86_64.rpm	http://mirror.ash.fastserv.com/pub/linux/centos/7.1.1503/os/x86_64/Packa
incron-0.5.10-8.el7.x86_64.rpm	http://lug.mtu.edu/epel/7/x86_64/i/incron-0.5.10-8.el7.x86_64.rpm
kernel-3.10.0-229.14.1.el7.x86_64.rpm	http://mirrors.advancedhosters.com/centos/7.1.1503/updates/x86_64/Pack
kibana-3.1.0.SIMP-0.noarch.rpm	https://dl.bintray.com/simp/5.1.X-Ext/kibana-3.1.0.SIMP-0.noarch.rpm
libarchive-devel-3.1.2-7.el7.x86_64.rpm	http://mirrors.advancedhosters.com/centos/7.1.1503/os/x86_64/Packages/
libconfuse-2.7-7.el7.x86_64.rpm	http://lug.mtu.edu/epel/7/x86_64/l/libconfuse-2.7-7.el7.x86_64.rpm
libev-4.15-3.el7.x86_64.rpm	http://lug.mtu.edu/epel/7/x86_64/l/libev-4.15-3.el7.x86_64.rpm
libselinux-2.2.2-6.el7.x86_64.rpm	http://mirrors.advancedhosters.com/centos/7.1.1503/os/x86_64/Packages/
libselinux-python-2.2.2-6.el7.x86_64.rpm	http://mirrors.advancedhosters.com/centos/7.1.1503/os/x86_64/Packages/
libselinux-ruby-2.2.2-6.el7.x86_64.rpm	http://mirrors.advancedhosters.com/centos/7.1.1503/os/x86_64/Packages/
libselinux-static-2.2.2-6.el7.x86_64.rpm	http://mirrors.advancedhosters.com/centos/7.1.1503/os/x86_64/Packages/
libselinux-utils-2.2.2-6.el7.x86_64.rpm	http://mirrors.advancedhosters.com/centos/7.1.1503/os/x86_64/Packages/
libsepol-2.1.9-3.el7.x86_64.rpm	http://mirrors.advancedhosters.com/centos/7.1.1503/os/x86_64/Packages/
libsepol-static-2.1.9-3.el7.x86_64.rpm	http://mirrors.advancedhosters.com/centos/7.1.1503/os/x86_64/Packages/
libyaml-0.1.4-11.el7_0.x86_64.rpm	http://mirrors.advancedhosters.com/centos/7.1.1503/os/x86_64/Packages/
linux-firmware-20140911-0.1.git365e80c.el7.noarch.rpm	http://mirror.ash.fastserv.com/pub/linux/centos/7.1.1503/os/x86_64/Packa
logstash-1.4.2-1_2c0f5a1.noarch.rpm	https://download.elasticsearch.org/logstash/logstash/packages/centos/logs
logstash-contrib-1.4.2-1_efd53ef.noarch.rpm	https://download.elastic.co/logstash/logstash/packages/centos/logstash-co
mcollective-2.8.4-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-2.8.4-1.el7.n
mcollective-actionpolicy-auth-2.1.0-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-actionpolicy
mcollective-client-2.8.4-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-client-2.8.4-
mcollective-common-2.8.4-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-common-2.8
mcollective-filemgr-agent-1.0.2-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-filemgr-age
mcollective-filemgr-client-1.0.2-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-filemgr-clie
mcollective-filemgr-common-1.0.2-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-filemgr-com
mcollective-iptables-agent-3.0.2-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-iptables-age
mcollective-iptables-client-3.0.2-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-iptables-clie
mcollective-iptables-common-3.0.2-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-iptables-com
mcollective-nettest-agent-3.0.4-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-nettest-agen
mcollective-nettest-client-3.0.4-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-nettest-clien
mcollective-nettest-common-3.0.4-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-nettest-com

Table 4.2 – continued from previous page

Name	Source
mcollective-nrpe-agent-3.1.0-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-nrpe-agent-
mcollective-nrpe-client-3.1.0-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-nrpe-client-
mcollective-nrpe-common-3.1.0-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-nrpe-comm
mcollective-package-agent-4.4.0-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-package-ag
mcollective-package-client-4.4.0-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-package-cli
mcollective-package-common-4.4.0-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-package-co
mcollective-puppet-agent-1.10.0-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-puppet-age
mcollective-puppet-client-1.10.0-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-puppet-clie
mcollective-puppet-common-1.10.0-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-puppet-com
mcollective-service-agent-3.1.3-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-service-age
mcollective-service-client-3.1.3-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-service-clie
mcollective-service-common-3.1.3-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-service-com
mcollective-shell-agent-0.0.2-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-shell-agent-
mcollective-shell-client-0.0.2-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-shell-client-
mcollective-shell-common-0.0.2-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-shell-comm
mcollective-sshkey-security-0.5.0-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-sshkey-secu
mcollective-sysctl-data-2.0.1-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/mcollective-sysctl-data-
pdsh-2.29-1el7.x86_64.rpm	https://dl.bintray.com/simp/5.1.X-Ext/pdsh-2.29-1el7.x86_64.rpm
pdsh-debuginfo-2.29-1el7.x86_64.rpm	https://dl.bintray.com/simp/5.1.X-Ext/pdsh-debuginfo-2.29-1el7.x86_64.r
pdsh-mod-dshgroup-2.29-1el7.x86_64.rpm	https://dl.bintray.com/simp/5.1.X-Ext/pdsh-mod-dshgroup-2.29-1el7.x86_
pdsh-mod-machines-2.29-1el7.x86_64.rpm	https://dl.bintray.com/simp/5.1.X-Ext/pdsh-mod-machines-2.29-1el7.x86_
pdsh-mod-netgroup-2.29-1el7.x86_64.rpm	https://dl.bintray.com/simp/5.1.X-Ext/pdsh-mod-netgroup-2.29-1el7.x86_
pdsh-rcmd-exec-2.29-1el7.x86_64.rpm	https://dl.bintray.com/simp/5.1.X-Ext/pdsh-rcmd-exec-2.29-1el7.x86_64.r
pdsh-rcmd-ssh-2.29-1el7.x86_64.rpm	https://dl.bintray.com/simp/5.1.X-Ext/pdsh-rcmd-ssh-2.29-1el7.x86_64.rp
pssh-2.3.1.SIMP-5.el7.noarch.rpm	https://dl.bintray.com/simp/5.1.X-Ext/pssh-2.3.1.SIMP-5.el7.noarch.rpm
puppet-3.8.1-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/puppet-3.8.1-1.el7.noarc
puppet-server-3.8.1-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/puppet-server-3.8.1-1.el
puppetdb-2.3.8-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/puppetdb-2.3.8-1.el7.no
puppetdb-terminus-2.3.8-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/puppetdb-terminus-2.3.8
puppetlabs-release-7-11.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/puppetlabs-release-7-11
puppetserver-1.1.1-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/puppetserver-1.1.1-1.el7
python-elasticsearch-1.2.0-0.el7.centos.noarch.rpm	https://dl.bintray.com/simp/5.1.X-Ext/python-elasticsearch-1.2.0-0.el7.cen
python-redis-2.10.3-1.el7.noarch.rpm	http://lug.mtu.edu/epel/7/x86_64/p/python-redis-2.10.3-1.el7.noarch.rpm
python-simplejson-3.3.3-1.el7.x86_64.rpm	http://lug.mtu.edu/epel/7/x86_64/p/python-simplejson-3.3.3-1.el7.x86_64
python-unittest2-0.5.1-6.el7.noarch.rpm	http://lug.mtu.edu/epel/7/x86_64/p/python-unittest2-0.5.1-6.el7.noarch.rp
razor-server-1.0.1-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/razor-server-1.0.1-1.el7.
razor-torquebox-3.1.1.9-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/products/x86_64/razor-torquebox-3.1.1.9-
rrdtool-1.4.8-8.el7.x86_64.rpm	http://mirrors.advancedhosters.com/centos/7.1.1503/os/x86_64/Packages/
ruby-augeas-0.4.1-3.el7.x86_64.rpm	http://yum.puppetlabs.com/el/7/dependencies/x86_64/ruby-augeas-0.4.1-3
ruby-ldap-0.9.16-1.el7.x86_64.rpm	http://lug.mtu.edu/epel/7/x86_64/r/ruby-ldap-0.9.16-1.el7.x86_64.rpm
ruby-rgen-0.6.5-2.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/dependencies/x86_64/ruby-rgen-0.6.5-2.e
ruby-shadow-2.2.0-2.el7.x86_64.rpm	http://yum.puppetlabs.com/el/7/dependencies/x86_64/ruby-shadow-2.2.0-
rubygem-deep_merge-1.0.0-2.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/dependencies/x86_64/rubygem-deep_mer
rubygem-ffi-1.4.0-2.el7.x86_64.rpm	http://yum.puppetlabs.com/el/7/dependencies/x86_64/rubygem-ffi-1.4.0-2
rubygem-highline-1.6.11-5.el7.noarch.rpm	http://lug.mtu.edu/epel/7/x86_64/r/rubygem-highline-1.6.11-5.el7.noarch
rubygem-net-ping-1.6.2-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/dependencies/x86_64/rubygem-net-ping-
rubygem-puppet-lint-1.1.0-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/dependencies/x86_64/rubygem-puppet-lin
rubygem-rake-compiler-0.9.3-1.el7.noarch.rpm	http://yum.puppetlabs.com/el/7/dependencies/x86_64/rubygem-rake-com
rubygem-stomp-1.3.4-2.el7.noarch.rpm	http://lug.mtu.edu/epel/7/x86_64/r/rubygem-stomp-1.3.4-2.el7.noarch.rpm

Table 4.2 – continued from previous page

Name	Source
rubygem-stomp-doc-1.3.4-2.el7.noarch.rpm	http://lug.mtu.edu/epel/7/x86_64/r/rubygem-stomp-doc-1.3.4-2.el7.noarch
scap-security-guide-0.1.19-2.el7.noarch.rpm	http://mirrors.advancedhosters.com/centos/7.1.1503/os/x86_64/Packages/
simp-lastbind-2.4.23-0.x86_64.rpm	https://dl.bintray.com/simp/5.1.X-Ext/simp-lastbind-2.4.23-0.x86_64.rpm
simp-ppolicy-check-password-2.4.39-0el7.x86_64.rpm	https://dl.bintray.com/simp/5.1.X-Ext/simp-ppolicy-check-password-2.4.3
source-highlight-3.1.6-6.el7.x86_64.rpm	http://mirrors.advancedhosters.com/centos/7.1.1503/os/x86_64/Packages/
sudosh2-1.0.2-2el7.x86_64.rpm	https://dl.bintray.com/simp/5.1.X-Ext/sudosh2-1.0.2-2el7.x86_64.rpm
syslinux-tftpboot-4.05-12.el7.x86_64.rpm	http://mirrors.advancedhosters.com/centos/7.1.1503/os/x86_64/Packages/

4.5.5 SIMP SCTM

This SCTM was developed based on the National Institute of Standards and Technology (NIST) Specical Publication 800-53 (Revision 3) controls that SIMP currently meets. Empty contents means SIMP does not meet that control. Implementations are free to take these tables and use them as a starting point for any accreditation activities that follow NIST 800-53.

SIMP SCTM Technical Controls

Control ID	Control Name	Control Family	SIMP Implementation Method
AC-1	Access Control Policy and Procedures	Access Control	
AC-2(1)	Account Management (Control Enhancement)	Access Control	LDAP is used to centrally manage accounts. Local accounts can optionally be added and managed by puppet.
AC-2(2)	Account Management (Control Enhancement)	Access Control	
AC-2(3)	Account Management (Control Enhancement)	Access Control	Inactive local accounts expire 35 days after password expiration. LDAP accounts can be set to expire in LDAP and using PAM. There is no automated method (included with SIMP) to check inactive LDAP accounts. Implementations should address inactive LDAP accounts with automated or administrative measures.

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
AC-2(4)	Account Management (Control Enhancement)	Access Control	Local account creation is audited with auditd. (as are all of root's actions). Sudosh logs all commands for someone running sudosh. This will not work if the SIMP implementation uses specific sudo rules. Instead, sudo actions are logged using auditd. Ldap modifications are logged in the ldap logs.
AC-2(5)	Account Management (Control Enhancement)	Access Control	Shell accounts are logged out after 15 minutes of inactivity
AC-2(6)	Account Management (Control Enhancement)	Access Control	
AC-2(7)	Account Management (Control Enhancement)	Access Control	SIMP has a default administrators group (700) that users can be assigned to. Additional roles and groups are up to the implementations. Role changes are logged in the LDAP logs.
AC-3	Access Enforcement	Access Control	
AC-3(2)	Access Enforcement (Control Enhancement)	Access Control	
AC-3(3)	Access Enforcement (Control Enhancement)	Access Control	DAC has been built into Unix for a long time and is expected to work. Implementations may want to check that user assignments to groups properly enforce DAC they way they expect. New as of SIMP 5.0 is the use of MAC. All stock SIMP modules work with MAC enabled. It's up to each implementation to ensure their applications and modules are made to work with MAC enabled.
AC-3(4)	Access Enforcement (Control Enhancement)	Access Control	DAC has been built into Unix for a long time and is expected to work. Implements may want to check that user assignments to groups properly enforce DAC they way they expect.

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
AC-3(5)	Access Enforcement (Control Enhancement)	Access Control	SIMP implements file permissions per the SCAP-Security-Guide (SSG) RHEL7 guidance. There are some exceptions of file permissions being more or less restrictive than the guide. Mitigations and responses to those variances will be published once final RHEL7 SCAP content is available.
AC-3(6)	Access Enforcement (Control Enhancement)	Access Control	
AC-4(1)	Information Flow Enforcement (Control Enhancement)	Access Control	IPTables enforces flow control to the puppet master and clients. The default rules allow the services needed for kick start and puppet (and SSH of course). IPTables is managed by puppet so that any user modifications to /etc/sysconfig/iptables is rewritten with the rules from the manifest. The rules can and should be tailored per implementation.
AC-4(2)	Information Flow Enforcement (Control Enhancement)	Access Control	
AC-4(3)	Information Flow Enforcement (Control Enhancement)	Access Control	
AC-4(4)	Information Flow Enforcement (Control Enhancement)	Access Control	
AC-4(5)	Information Flow Enforcement (Control Enhancement)	Access Control	
AC-4(6)	Information Flow Enforcement (Control Enhancement)	Access Control	
AC-4(7)	Information Flow Enforcement (Control Enhancement)	Access Control	
AC-4(8)	Information Flow Enforcement (Control Enhancement)	Access Control	

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
AC-4(9)	Information Flow Enforcement (Control Enhancement)	Access Control	
AC-4(10)	Information Flow Enforcement (Control Enhancement)	Access Control	
AC-4(11)	Information Flow Enforcement (Control Enhancement)	Access Control	
AC-4(12)	Information Flow Enforcement (Control Enhancement)	Access Control	
AC-4(13)	Information Flow Enforcement (Control Enhancement)	Access Control	
AC-4(14)	Information Flow Enforcement (Control Enhancement)	Access Control	
AC-4(15)	Information Flow Enforcement (Control Enhancement)	Access Control	
AC-4(16)	Information Flow Enforcement (Control Enhancement)	Access Control	
AC-4(17)	Information Flow Enforcement (Control Enhancement)	Access Control	
AC-5	Separation of Duties	Access Control	
AC-6	Least Privilege	Access Control	SIMP was built using a minimalist approach. Only the services, applications (RPMs and their dependencies), and network rules that are needed are implemented. Adding additional services, users, or software are done using built in RedHat/CentOS features or puppet. For example, services cannot be manually added without first registering them with puppet.

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
AC-6(1)	Least Privilege (Control Enhancement)	Access Control	File permissions and administrative functions are denied to users who are not administrators using Unix DAC. Roles can be defined by a implementation. Typically it's done using ldap groups and sudosh. Suoders rules can be set for roles that need a limited set of commands/functions.
AC-6(2)	Least Privilege (Control Enhancement)	Access Control	Direct remote root login is not allowed on SIMP. Users must assume their role first (defined in LDAP or locally). There is a local simp user on the puppet master that has a password assigned. That allows for emergency maintenance via SSH. Single user mode is password protected, but will allow direct access before escalation. Protection of the single user mode and simp user's password is up to the implementation. Privilege escalation is performed using sudosh or sudo. Most implementations will use sudosh for global admins and sudo for roles that need minimal admin ability. Lastly, serial port access is does allow direct root login (/etc/securetty). Implementations may further restrict this at the risk.
AC-6(3)	Least Privilege (Control Enhancement)	Access Control	
AC-6(4)	Least Privilege (Control Enhancement)	Access Control	
AC-6(5)	Least Privilege (Control Enhancement)	Access Control	
AC-6(6)	Least Privilege (Control Enhancement)	Access Control	

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
AC-7	Unsuccessful Login Attempts	Access Control	SIMP locks accounts after 5 invalid attempts over 15 minutes span. It then keeps the account locked for 15 minutes. After that, the account is unlocked automatically.
AC-7(1)	Unsuccessful Login Attempts (Control Enhancement)	Access Control	An account is never locked to a point an admin must unlock it. It will continue to be unlocked after 15 minutes. This should meet most modern policies. It can be further restricted if required by local policies.
AC-7(2)	Unsuccessful Login Attempts (Control Enhancement)	Access Control	
AC-8	System Use Notification	Access Control	SIMP displays a default banner prior to login. Implementations must customize that banner for their use.
AC-9	Previous Logon (Access) Notification	Access Control	SIMP uses the pam_lastlog.so module to display last login information.
AC-9(1)	Previous Logon (Access) Notification (Control Enhancement)	Access Control	SIMP uses the pam_lastlog.so module to display last login information.
AC-9(2)	Previous Logon (Access) Notification (Control Enhancement)	Access Control	SIMP uses the pam_lastlog.so module to display last login information, including the number of failed login attempts since the last logon.
AC-9(3)	Previous Logon (Access) Notification (Control Enhancement)	Access Control	

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
AC-10	Concurrent Session Control	Access Control	The default value for concurrent sessions in SIMP is 10 (/etc/security/limits.conf). Given the variety of system usage to include automated processes, it could impact functionality if this value were set lower. It can be tailored to a lower value if the implementation determines that number will not impact functionality.
AC-11	Session Lock	Access Control	Terminal sessions do not enforce a session lock so this control is technically not implemented. However, it's mitigated by forcing inactive sessions to log out. If the gnome module is applied, SIMP locks a gnome session after 5 minutes.
AC-14	Permitted Actions without Identification or Authentication	Access Control	SIMP provides several services that do not require authentication. Most require some form of identification. These are documented in the SIMP Security Concepts and is kept current for that version. Individual modules are not yet documented.
AC-14(1)	Permitted Actions without Identification or Authentication (Control Enhancement)	Access Control	Justifications to those services that do not require Identification and Authentication can be found in the SIMP Security Concepts document.

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
AC-16	Security Attributes	Access Control	New in SIMP 5.0 is the usage of MAC via SELinux. This is optional for each implementation and can be turned off at any time. All of the stock SIMP modules work with SELinux enabled and have the least restrictive MAC policies enforced. These policies assign each object a SELinux user, role, type, and level. These characteristics are used to define a context for each object.
AC-16(1)	Security Attributes (Control Enhancement)	Access Control	
AC-16(2)	Security Attributes (Control Enhancement)	Access Control	
AC-16(3)	Security Attributes (Control Enhancement)	Access Control	
AC-16(4)	Security Attributes (Control Enhancement)	Access Control	SeLinux user, role, type, and level are the security attributes that are associated with each object with SELinux enabled in SIMP.
AC-16(5)	Security Attributes (Control Enhancement)	Access Control	
AC-17	Remote Access		By default, external connections are not allowed with the exception of SSH. This is documented in the SIMP user manual. Implementations have the ability to override this with the understanding that puppet controls Iptables.
AC-17(1)	Remote Access (Control Enhancement)	Access Control	The extent of monitoring remote connections is done by auditd and syslog. The contents of the remote session is not logged. The keystrokes of users with sudosh shells are all logged.
AC-17(2)	Remote Access (Control Enhancement)	Access Control	Remote access is limited to SSH. SSH (openssh on centos/rhel) provides both confidentiality and integrity of the remote session.

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
AC-17(3)	Remote Access (Control Enhancement)	Access Control	
AC-17(4)	Remote Access (Control Enhancement)	Access Control	This control is enforced via other access control mechanisms already covered in 800-53. Namely, AC-6. By default, SSH in SIMP will allow anyone to connect. Once identification and authentication is performed, access control to privileged commands is enforced as usual.
AC-17(5)	Remote Access (Control Enhancement)	Access Control	Auditd provides logging of failed access attempts. It's up to the implementation to perform a level of inspection of these unauthorized events. Auditd does this by default. Other checks will ensure auditd is running and registered with puppet.
AC-17(6)	Remote Access (Control Enhancement)	Access Control	
AC-17(7)	Remote Access (Control Enhancement)	Access Control	
AC-17(8)	Remote Access (Control Enhancement)	Access Control	This control is only met by defining all connections that SIMP allows internally and externally. For now, since this is a remote access control, it should suffice to continue to note that the only remote access protocol allowed by default is SSH.
AC-18	Wireless Access	Access Control	
AC-18(1)	Wireless Access (Control Enhancement)	Access Control	
AC-18(2)	Wireless Access (Control Enhancement)	Access Control	
AC-18(3)	Wireless Access (Control Enhancement)	Access Control	
AC-18(4)	Wireless Access (Control Enhancement)	Access Control	
AC-18(5)	Wireless Access (Control Enhancement)	Access Control	
AC-19	Access Control for Mobile Devices	Access Control	

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
AC-19(1)	Access Control for Mobile Devices (Control Enhancement)	Access Control	
AC-19(2)	Access Control for Mobile Devices (Control Enhancement)	Access Control	
AC-19(3)	Access Control for Mobile Devices (Control Enhancement)	Access Control	
AC-19(4)	Access Control for Mobile Devices (Control Enhancement)	Access Control	
AC-20	Use of External Information Systems	Access Control	
AC-20(1)	Use of External Information Systems (Control Enhancement)	Access Control	
AC-20(2)	Use of External Information Systems (Control Enhancement)	Access Control	
AC-21	User-Based Collaboration and Information Sharing	Access Control	
AC-21(1)	User-Based Collaboration and Information Sharing (Control Enhancement)	Access Control	
AC-22	Publicly Accessible Content	Access Control	
AU-1	Audit and Accountability Policy and Procedures	Audit and Accountability	

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
AU-2	Auditable Events	Audit and Accountability	1. SIMP audit rules were built by using idustry 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 list of events that audited are by auditd can be found in appendix of the Security Concepts document. b. Implementation Specific c. Rational is for audit setting is provided in SSG. d. Threat information is specific to the implementation. Auditd and syslog facility can always be fine tuned for each implementation.
AU-2(3)	Auditable Events (Control Enhancement)	Audit and Accountability	SIMP is constantly reviewing the audit rules for accuracy, relevance, and performance. Rules are added and in some cases removed as information becomes available.

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
AU-2(4)	Auditable Events (Control Enhancement)	Audit and Accountability	Privileged user commands are logged using sudosh and auditd (sudo actions). By default, users in the administrators group can run sudosh. All of the key strokes (except things that are not echoed back to the screen like passwords) are logged to /var/log/sudosh.log and can be sent to syslog. If an implementation sets up specific sudo actions for other groups or users, those actions are logged with auditd.
AU-3	Content of Audit Records	Audit and Accountability	The linux audit daemon contains event type, date/time, host, and outcome of events by default.
AU-3(1)	Content of Audit Records (Control Enhancement)	Audit and Accountability	There are a number of events that are captured beyond the auditd. The SIMP syslog module captures additional log events from apache, ldap, puppet, messages.log, and secure.log.
AU-3(2)	Content of Audit Records (Control Enhancement)	Audit and Accountability	By default, the SIMP syslog module logs locally. There is an option to send the syslog events to a central location. Instructions for implementing a syslog server are provided in the User Guide. Lastly, a combination of elasticsearch, logstash, and kibana (ELK) can be applied to filter, index, and search logs. Puppet modules are provided for the ELK stack
AU-4	Audit Storage Capacity	Audit and Accountability	The audit partition is configured as a separation partition from the system files, reducing the likelihood of audit interfering with system operations. Implementaions can change this but it's highly discouraged.

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
AU-5	Response to Audit Processing Failures	Audit and Accountability	1. Implementation Specific. b. The audit.conf file configures the system to log to syslog when disk space becomes low. If the disk becomes full, the audit daemon will be suspended, but the system will remain active. This is contrary to some industry guidance to put the system into single user mode when disk space becomes an issue. Implementations may wish to change the default behaviour at the risk of stopping the system from functioning.
AU-5(1)	Response to Audit Processing Failures (Control Enhancement)	Audit and Accountability	SIMP provides a warning (to syslog) when the disk has 75MB free. Each log file can be up to 30MB.
AU-5(2)	Response to Audit Processing Failures (Control Enhancement)	Audit and Accountability	
AU-5(3)	Response to Audit Processing Failures (Control Enhancement)	Audit and Accountability	
AU-5(4)	Response to Audit Processing Failures (Control Enhancement)	Audit and Accountability	SIMP will not shut down a system by default. Implementation can configure this option at the own risk in the auditd.conf file.
AU-6	Audit Review, Analysis, and Reporting	Audit and Accountability	
AU-6(1)	Audit Review, Analysis, and Reporting (Control Enhancement)	Audit and Accountability	

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
AU-6(3)	Audit Review, Analysis, and Reporting (Control Enhancement)	Audit and Accountability	The ELK modules provide implementations with one means to centralize, review, and recognize trends in SIMP logs.
AU-6(4)	Audit Review, Analysis, and Reporting (Control Enhancement)	Audit and Accountability	The ELK modules provide implementations with one means to centralize, review, and recognize trends in SIMP logs.
AU-6(5)	Audit Review, Analysis, and Reporting (Control Enhancement)	Audit and Accountability	The ELK modules provide implementations with one means to centralize, review, and recognize trends in SIMP logs. The logs sent to syslog can be customized to include logs from any application. They would then be in a central place for viewing and aggregation by users of the Kibana interface.
AU-6(6)	Audit Review, Analysis, and Reporting (Control Enhancement)	Audit and Accountability	
AU-6(7)	Audit Review, Analysis, and Reporting (Control Enhancement)	Audit and Accountability	
AU-6(9)	Audit Review, Analysis, and Reporting (Control Enhancement)	Audit and Accountability	
AU-7	Audit Reduction and Report Generation	Audit and Accountability	

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
AU-7(1)	Audit Reduction and Report Generation (Control Enhancement)	Audit and Accountability	While not true audit reduction, RedHat does allow someone with access to audit logs to perform filters using the journald. If audit logs are forwarded to a syslog server, it's not difficult for an admin to security officer to run batch filters against all of the audit records. As of SIMP 4.0.5, an optional Logstash, Kibana, and Elasticsearch modules can be applied. If applied, they provide centralized and indexed logs. An implementation can then perform searches against the logs or provide alerts to other parts of their infrastructure.
AU-8	Time Stamps	Audit and Accountability	Auditd uses the system clock to time stamp audit events.
AU-8(1)	Time Stamps (Control Enhancement)	Audit and Accountability	Time is an essential component of puppet. Therefore, NTPD is used to synchronize puppet clients with the puppet server. That default configuration can be changed to synchronize puppet each server/client with another time source.
AU-9	Protection of Audit Information	Audit and Accountability	File system permissions and SELinux protect the content of /var/log/audit and /etc/audit/*
AU-9(1)	Protection of Audit Information (Control Enhancement)	Audit and Accountability	
AU-9(2)	Protection of Audit Information (Control Enhancement)	Audit and Accountability	
AU-9(3)	Protection of Audit Information (Control Enhancement)	Audit and Accountability	
AU-9(4)	Protection of Audit Information (Control Enhancement)	Audit and Accountability	

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
AU-10	Non-repudiation	Audit and Accountability	
AU-10(1)	Non-repudiation (Control Enhancement)	Audit and Accountability	
AU-10(2)	Non-repudiation (Control Enhancement)	Audit and Accountability	
AU-10(3)	Non-repudiation (Control Enhancement)	Audit and Accountability	
AU-10(4)	Non-repudiation (Control Enhancement)	Audit and Accountability	
AU-10(5)	Non-repudiation (Control Enhancement)	Audit and Accountability	
AU-12(1)	Audit Generation (Control Enhancement)	Audit and Accountability	
AU-11	Audit Record Retention	Audit and Accountability	
AU-12	Audit Generation	Audit and Accountability	1. Auditd provides the audit generation capability and is running on all SIMP systems by default.b. The audit.rules files configures events that are audited. c. The audit.rules applies the list of audit rules defined in SIMP Security Concepts document.
AU-12(1)	Audit Generation (Control Enhancement)	Audit and Accountability	Auditd stamps audit records with the system time. The system time is obtained from a central time source and synchronized between SIMP systems.
AU-12(2)	Audit Generation (Control Enhancement)	Audit and Accountability	Auditd provides logging in standard formats. Additionally, logs that are sent through syslog adhere to that standard.
AU-13	Monitoring For Information Disclosure	Audit and Accountability	
AU-14	Session Audit	Audit and Accountability	
AU-14(1)	Session Audit (Control Enhancement)	Audit and Accountability	Sessions that use the sudo shell have all keystrokes recorded. Those sessions can be viewed in text format or replayed to the screen

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
IA-1	Identification and Authentication Policy and Procedures	Identification and Authentication	
IA-2(1)	User Identification and Authentication (Organizational Users) (Control Enhancement)	Identification and Authentication	
IA-2(2)	User Identification and Authentication (Organizational Users) (Control Enhancement)	Identification and Authentication	
IA-2(3)	User Identification and Authentication (Organizational Users) (Control Enhancement)	Identification and Authentication	
IA-2(4)	User Identification and Authentication (Organizational Users) (Control Enhancement)	Identification and Authentication	
IA-2(5)	User Identification and Authentication (Organizational Users) (Control Enhancement)	Identification and Authentication	
IA-2(6)	User Identification and Authentication (Organizational Users) (Control Enhancement)	Identification and Authentication	
IA-2(7)	User Identification and Authentication (Organizational Users) (Control Enhancement)	Identification and Authentication	
IA-2(8)	User Identification and Authentication (Organizational Users) (Control Enhancement)	Identification and Authentication	The authentication mechanisms used within SIMP are all resistant to replay attacks by default. Known vulnerabilities can occur in the protocols. As they are known, vendors release patches, which must them be applied by the implementation. Privileged accounts use the same protocols as unprivileged accounts.

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
IA-2(9)	User Identification and Authentication (Organizational Users) (Control Enhancement)	Identification and Authentication	The authentication mechanisms used within SIMP are all resistant to replay attacks by default. Known vulnerabilities can occur in the protocols. As they are known, vendors release patches, which must them be applied by the implementation.
IA-3	Device Identification and Authentication	Identification and Authentication	Identification of each puppet client occurs before an IP address can be assigned. This is controlled using DHCP (each client must have an address bound by MAC address). Devices identification and authentication with puppet occurs using SSL certificates. The clients must each have a SSL certificate installed to establish a valid session with the puppet master.
IA-3(1)	Device Identification and Authentication (Control Enhancement)	Identification and Authentication	
IA-3(2)	Device Identification and Authentication (Control Enhancement)	Identification and Authentication	
IA-3(3)	Device Identification and Authentication (Control Enhancement)	Identification and Authentication	DHCP is used to statically define the IP addresses of each puppet client.
IA-4	Identifier Management	Identification and Authentication	Local accounts expire 35 days after their passwords expire. There is no mechanism implemented to detect inactive LDAP accounts. Implementations might wish to mitigate this by regularly reviewing and removing unneeded accounts.
IA-4(1)	Identifier Management (Control Enhancement)	Identification and Authentication	
IA-4(2)	Identifier Management (Control Enhancement)	Identification and Authentication	
IA-4(3)	Identifier Management (Control Enhancement)	Identification and Authentication	

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
IA-4(4)	Identifier Management (Control Enhancement)	Identification and Authentication	
IA-4(5)	Identifier Management (Control Enhancement)	Identification and Authentication	
IA-5	Authenticator Management	Identification and Authentication	3. Authenticator strength is enforced using pam_crack_lib.so. This works for user defined passwords on local and LDAP accounts. E. It's up to the implementation to change the values for the various passwords. F. Password history is set to 24 by default in SIMP and enforced with pam.G. For local accounts, password aging is set to 180 days. It's set to the same in LDAP, but enforced at the time of account creation using ldifs. LDAP subsequently uses PAM to enforce the aging. Key based passwordless logins do not enforce aging. Upon generation, server and puppet certificates can also be set to expire.H. Authenticators for local and LDAP account are protected using operating system access controls. The server certificates are also protected using operating system controls.

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
IA-5(1)	Authenticator Management (Control Enhancement)	Identification and Authentication	1. Authenticator strength is enforced using pam_crack_lib.so. This works for user defined passwords on local and LDAP accounts. Administrators can bypass PAM and set weak passwords in LDAP. Under normal circumstances, users would be forced to change their password at login, at which point pam enforced complexity. b. Not enforced c. Hashed passwords are built into linux (/etc/shadow and /etc/pam.d/system-auth pam_unix.so). LDAP password changed by users are done through pam before getting placed in LDAP. Manual LDAP password are created using the slapasswd command.d. Password minimum and maximum lifetimes are enforced through /etc/login.defs and ldap. e. By default, the previous 24 passwords can not be reused.

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
IA-5(2)	Authenticator Management (Control Enhancement)	Identification and Authentication	Puppet comes with a self contained public key infrastructure. Though just used for puppet, it operates as a full PKI. So the certificate path is validated.SSL certificates that are used for SSL and TLS also have certificate path validation built into the protocol.Note: SSH Keys are not considered PKI.
IA-5(3)	Authenticator Management (Control Enhancement)	Identification and Authentication	
IA-5(4)	Authenticator Management (Control Enhancement)	Identification and Authentication	Pam cracklib enforces password complexity rules on Redhat and CentOS. Additional tools to check authenticator strength can be used in operational settings.
IA-5(5)	Authenticator Management (Control Enhancement)	Identification and Authentication	The simp-config utility gives each implementation an opportunity to change default passwords at build time. It's up to the implementation to change the values for the various passwords.
IA-5(6)	Authenticator Management (Control Enhancement)	Identification and Authentication	Authenticators are protected with operating system access control and file permissions.
IA-5(7)	Authenticator Management (Control Enhancement)	Identification and Authentication	Plaintext passwords are only used when application support no other means of providing a password.
IA-5(8)	Authenticator Management (Control Enhancement)	Identification and Authentication	
IA-6	Authenticator Feedback	Identification and Authentication	Plaintext passwords are not echoed back to the screen.

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
IA-7	Cryptographic Module Authentication	Identification and Authentication	Redhat 7 and the several modules are being evaluated for FIPS 140 compliance. Implementations should check the FIPS site for updates on this evaluation. The SIMP team will also continue to evaluate the status and any relevant settings that need to be applied as a result of this evaluation.
IA-8	Identification and Authentication (Non-Organizational Users)	Identification and Authentication	
SC-1	System and Communications Protection Policy and Procedures	System and Communications Protection	
SC-2	Application Partitioning	System and Communications Protection	The spirit of this control is providing logical separation so that users are not able to access administrative functions. There is no notion of partitioning within SIMP. There are access control enforcement that can be proven through tests on those controls. If this control is allocated to SIMP alone, it's unlikely it can be met. Since SIMP is the infrastructure that applications would use, showing that application users cannot access the SIMP environment is a better way to prove this control is met.

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
SC-2(1)	Application Partitioning (Control Enhancement)	System and Communications Protection	The spirit of this control is providing logical separation so that users are not able to access administrative functions. There is no notion of partitioning within SIMP. There are access control enforcement that can be proven through tests on those controls. If this control is allocated to SIMP alone, it's unlikely it can be met. Since SIMP is the infrastructure that applications would use, showing that application users cannot access the SIMP environment is a better way to prove this control is met.
SC-3	Security Function Isolation	System and Communications Protection	The spirit of this control is providing logical separation so that users are not able to access administrative functions. There is no notion of partitioning within SIMP. There are access control enforcement that can be proven through tests on those controls. If this control is allocated to SIMP alone, it's unlikely it can be met. Since SIMP is the infrastructure that applications would use, showing that application users cannot access the SIMP environment is a better way to prove this control is met.
SC-3(1)	Security Function Isolation (Control Enhancement)	System and Communications Protection	
SC-3(2)	Security Function Isolation (Control Enhancement)	System and Communications Protection	
SC-3(3)	Security Function Isolation (Control Enhancement)	System and Communications Protection	
SC-3(4)	Security Function Isolation (Control Enhancement)	System and Communications Protection	
SC-3(5)	Security Function Isolation (Control Enhancement)	System and Communications Protection	

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
SC-4	Information In Shared Resources	System and Communications Protection	While difficult for the SIMP team to prove, object reuse has been part of previous versions of RedHat common criteria testing. That testing focusing on Files system objects, IPC objects and Memory objects. Any issues discovered within the platform that cause object reuse issues are likely to be address in security patches provided by the vendor.
SC-4(1)	Information In Shared Resources (Control Enhancement)	System and Communications Protection	
SC-5	Denial of Service Protection	System and Communications Protection	
SC-5(1)	Denial of Service Protection (Control Enhancement)	System and Communications Protection	
SC-5(2)	Denial of Service Protection (Control Enhancement)	System and Communications Protection	
SC-6	Resource Priority	System and Communications Protection	
SC-7	Boundary Protection	System and Communications Protection	Most of this control deals with a separate boundary interface (FW etc.). There is a part of this control that deals with controlling network access at key internal boundary points. Since SIMP implements IPTables on all hosts (by default), each node might be considered an internal boundary. Note – internal boundaries are more likely implemented via vlans or internal layer 3 devices.
SC-7(1)	Boundary Protection (Control Enhancement)	System and Communications Protection	
SC-7(2)	Boundary Protection (Control Enhancement)	System and Communications Protection	
SC-7(3)	Boundary Protection (Control Enhancement)	System and Communications Protection	
SC-7(4)	Boundary Protection (Control Enhancement)	System and Communications Protection	

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
SC-7(5)	Boundary Protection (Control Enhancement)	System and Communications Protection	Iptables, as configured by default, blocks all incoming traffic except for what is explicitly allowed.
SC-7(6)	Boundary Protection (Control Enhancement)	System and Communications Protection	
SC-7(7)	Boundary Protection (Control Enhancement)	System and Communications Protection	
SC-7(8)	Boundary Protection (Control Enhancement)	System and Communications Protection	
SC-7(9)	Boundary Protection (Control Enhancement)	System and Communications Protection	
SC-7(10)	Boundary Protection (Control Enhancement)	System and Communications Protection	
SC-7(11)	Boundary Protection (Control Enhancement)	System and Communications Protection	
SC-7(12)	Boundary Protection (Control Enhancement)	System and Communications Protection	IPTables is the host based firewall implementation on RedHat/CentOS.
SC-7(13)	Boundary Protection (Control Enhancement)	System and Communications Protection	
SC-7(14)	Boundary Protection (Control Enhancement)	System and Communications Protection	
SC-7(15)	Boundary Protection (Control Enhancement)	System and Communications Protection	
SC-7(16)	Boundary Protection (Control Enhancement)	System and Communications Protection	
SC-7(17)	Boundary Protection (Control Enhancement)	System and Communications Protection	
SC-7(18)	Boundary Protection (Control Enhancement)	System and Communications Protection	

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
SC-8	Transmission Integrity	System and Communications Protection	With the exception of the services needed for kickstart, most communications within SIMP are protected by SSH or SSL. Implementations can add additional services or modules that do not use SSH or SSL. The SIMP Security Concepts document details the default allowed protocols and the mechanisms in place to protect them. It's also worth noting that the SIMP team has taken ever measure possible to remove encryption ciphers available to operating system applications. In the event this breaks an application, implementations might have to add those ciphers back.
SC-8(1)	Transmission Integrity (Control Enhancement)	System and Communications Protection	With the exception of the services needed for kickstart, most communications within SIMP are protected by SSH or SSL. Implementations can add additional services or modules that do not use SSH or SSL. The SIMP Security Concepts document details the default allowed protocols and the mechanisms in place to protect them. It's also worth noting that the SIMP team has taken ever measure possible to remove encryption ciphers available to operating system applications. In the event this breaks an application, implementations might have to add those ciphers back.
SC-8(2)	Transmission Integrity (Control Enhancement)	System and Communications Protection	

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
SC-9	Transmission Confidentiality	System and Communications Protection	With the exception of the services needed for kickstart, most communications within SIMP are protected by SSH or SSL. Implementations can add additional services or modules that do not use SSH or SSL. The SIMP Security Concepts document details the default allowed protocols and the mechanisms in place to protect them. It's also worth noting that the SIMP team has taken ever measure possible to remove encryption ciphers available to operating system applications. In the event this breaks an application, implementations might have to add those ciphers back.
SC-9(1)	Transmission Confidentiality (Control Enhancement)	System and Communications Protection	With the exception of the services needed for kickstart, most communications within SIMP are protected by SSH or SSL. Implementations can add additional services or modules that do not use SSH or SSL. The SIMP Security Concepts document details the default allowed protocols and the mechanisms in place to protect them. It's also worth noting that the SIMP team has taken ever measure possible to remove encryption ciphers available to operating system applications. In the event this breaks an application, implementations might have to add those ciphers back.
SC-9(2)	Transmission Confidentiality (Control Enhancement)	System and Communications Protection	
SC-10	Network Disconnect	System and Communications Protection	

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
SC-11	Trusted Path	System and Communications Protection	
SC-12	Cryptographic Key Establishment and Management	System and Communications Protection	In an operational setting, SIMP does not establish keys. It does come with the ability to create server keys using a custom application know as "FakeCA". SSH keys can also be established using standard Unix command line tools. In an operational settings, both sets of keys should be obtained from valid key infrastructures. There is also a CA that puppet uses to generate and manage keys for puppet only.
SC-12(1)	Cryptographic Key Establishment and Management (Control Enhancement)	System and Communications Protection	
SC-12(2)	Cryptographic Key Establishment and Management (Control Enhancement)	System and Communications Protection	
SC-12(3)	Cryptographic Key Establishment and Management (Control Enhancement)	System and Communications Protection	
SC-12(4)	Cryptographic Key Establishment and Management (Control Enhancement)	System and Communications Protection	
SC-12(5)	Cryptographic Key Establishment and Management (Control Enhancement)	System and Communications Protection	

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
SC-13	Use of Cryptography		The forms of cryptography used are applied through SSH, SSL, and TLS. RedHat FIPs mode enabling is on the near term horizon for SIMP. Once enabled, it will be documented here and should allow implemtations to further explain how this control is being met. There are several unencrypted protocols used on the puppet server (Apache/YUM, DHCPD, TFTP, and DNS). The Security Concepts docucment provides additional details on default services/protocols that are used.
SC-13(1)	Use of Cryptography (Control Enhancement)		The forms of cryptography used are applied through SSH, SSL, and TLS. There are several unencrypted protocols used on the puppet server (Apache/YUM, DHCPD, TFTP, and DNS) that are documented in the Security Concepts document.
SC-13(2)	Use of Cryptography (Control Enhancement)		The forms of cryptography used are applied through SSH, SSL, and TLS. There are several unencrypted protocols used on the puppet server (Apache/YUM, DHCPD, TFTP, and DNS) that are documented in the Security Concepts document.
SC-13(3)	Use of Cryptography (Control Enhancement)		
SC-13(4)	Use of Cryptography (Control Enhancement)		
SC-14	Public Access Protections	System and Communications Protection	
SC-15	Collaborative Computing Devices	System and Communications Protection	

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
SC-15(1)	Collaborative Computing Devices (Control Enhancement)	System and Communications Protection	
SC-15(2)	Collaborative Computing Devices (Control Enhancement)	System and Communications Protection	
SC-15(3)	Collaborative Computing Devices (Control Enhancement)	System and Communications Protection	
SC-16	Transmission of Security Attributes	System and Communications Protection	
SC-16(1)	Transmission of Security Attributes (Control Enhancement)	System and Communications Protection	
SC-17	Public Key Infrastructure Certificates	System and Communications Protection	In an operational setting, SIMP does not establish keys. It does come with the ability to create server keys using a custom application know as "FakeCA". SSH keys can also be established using standard unix command line tools. In an operational settings, both sets of keys should be obtained from valid key infrastructures.There is also a CA that puppet uses to generate and manage keys for puppet only.
SC-18	Mobile Code	System and Communications Protection	
SC-18(1)	Mobile Code (Control Enhancement)	System and Communications Protection	
SC-18(2)	Mobile Code (Control Enhancement)	System and Communications Protection	
SC-18(3)	Mobile Code (Control Enhancement)	System and Communications Protection	
SC-18(4)	Mobile Code (Control Enhancement)	System and Communications Protection	
SC-19	Voice Over Internet Protocol	System and Communications Protection	
SC-20	Secure Name /Address Resolution Service (Authoritative Source)	System and Communications Protection	
SC-20(1)	Secure Name /Address Resolution Service (Authoritative Source) (Control Enhancement)	System and Communications Protection	

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
SC-21	Secure Name /Address Resolution Service (Recursive or Caching Resolver)	System and Communications Protection	
SC-21(1)	Secure Name /Address Resolution Service (Recursive or Caching Resolver) (Control Enhancement)	System and Communications Protection	
SC-22	Architecture and Provisioning for Name/Address Resolution Service	System and Communications Protection	
SC-23	Session Authenticity	System and Communications Protection	The forms of cryptography used are applied through SSH, SSL, and TLS. There are several unencrypted protocols used on the puppet server (Apache/YUM, DHCPD, TFTP, and DNS) that are documented in the Security Concepts document.
SC-23(1)	Session Authenticity (Control Enhancement)	System and Communications Protection	The forms of cryptography used are applied through SSH, SSL, and TLS. There are several unencrypted protocols used on the puppet server (Apache/YUM, DHCPD, TFTP, and DNS) that are documented in the Security Concepts document.
SC-23(2)	Session Authenticity (Control Enhancement)	System and Communications Protection	
SC-23(3)	Session Authenticity (Control Enhancement)	System and Communications Protection	The forms of cryptography used are applied through SSH, SSL, and TLS. There are several unencrypted protocols used on the puppet server (Apache/YUM, DHCPD, TFTP, and DNS) that are documented in the Security Concepts document.
SC-23(4)	Session Authenticity (Control Enhancement)	System and Communications Protection	

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
SC-24	Fail in Known State	System and Communications Protection	The forms of cryptography used are applied through SSH, SSL, and TLS. There are several unencrypted protocols used on the puppet server (Apache/YUM, DHCPD, TFTP, and DNS) that are documented in the Security Concepts document.
SC-25	Thin Nodes	System and Communications Protection	
SC-26	Honeypots	System and Communications Protection	
SC-26(1)	Honeypots (Control Enhancement)	System and Communications Protection	
SC-27	Operating System-Independent Applications	System and Communications Protection	
SC-28	Protection of Information at Rest	System and Communications Protection	Confidentiality of data at rest is achieved using the operating system access control. Integrity is only checked for critical operating system files. Implementations have the ability to extend the integrity checking of AIDE to include additional files that are not frequently changed.
SC-28	Protection of Information at Rest (Control Enhancement)	System and Communications Protection	
SC-29	Heterogeneity	System and Communications Protection	
SC-30	Virtualization Techniques	System and Communications Protection	
SC-30(1)	Virtualization Techniques (Control Enhancement)	System and Communications Protection	
SC-30(2)	Virtualization Techniques (Control Enhancement)	System and Communications Protection	
SC-31	Covert Channel Analysis	System and Communications Protection	
SC-31(1)	Covert Channel Analysis (Control Enhancement)	System and Communications Protection	
SC-32	Information System Partitioning	System and Communications Protection	
SC-33	Transmission Preparation Integrity	System and Communications Protection	

Continued on next page

Table 4.3 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
SC-34	Non-modifiable Executable Programs	System and Communications Protection	
SC-34(1)	Non-modifiable Executable Programs (Control Enhancement)	System and Communications Protection	
SC-34(2)	Non-modifiable Executable Programs (Control Enhancement)	System and Communications Protection	

Table: SIMP SCTM

SIMP SCTM Operational Controls

Control ID	Control Name	Control Family	SIMP Implementation Method
AT-1	Security Awareness and Training Policy and Procedures	Awareness and Training	
AT-2(1)	Security Awareness (Control Enhancement)	Awareness and Training	
AT-3	Security Training	Awareness and Training	
AT-3(1)	Security Training (Control Enhancement)	Awareness and Training	
AT-3(2)	Security Training (Control Enhancement)	Awareness and Training	
AT-4	Security Training Records	Awareness and Training	
AT-5	Contacts with Security Groups and Associations	Awareness and Training	
CM-1	Configuration Management Policy and Procedures	Configuration Management	
CM-2	Baseline Configuration	Configuration Management	SIMP has strictly enforced version control during development. The baseline files for SIMP are kept and maintained in a git repository. Files are packaged and a series of auto tests are performed on each release. Once released, there is a version number associated for distribution. Additionally, custom puppet modules are in the form of RPMs and have version numbers associated with them. All documentation is also built with source code.

Continued on next page

Table 4.4 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
CM-2(1)	Baseline Configuration (Control Enhancement)	Configuration Management	
CM-2(2)	Baseline Configuration (Control Enhancement)	Configuration Management	SIMP has strictly enforced version control during development. The baseline files for SIMP are kept and maintained in a git repository. Files are packaged and a series of auto tests are performed on the release. Once released, there is a version number associated for distribution. All documentation is also built with source code.
CM-2(3)	Baseline Configuration (Control Enhancement)	Configuration Management	All old versions of SIMP remain in the code repository.
CM-2(4)	Baseline Configuration (Control Enhancement)	Configuration Management	
CM-2(5)	Baseline Configuration (Control Enhancement)	Configuration Management	1. SIMP provides a minimal list of packages and services installed. The minimal list of packages can be found in kickstart files and the appendix of this document. Additional packages are installed by each implementation or as SIMP modules are applied. b. It's not feasible to technically deny additional applications from being installed. There is nothing in SIMP that can stop and RPM from being applied. Applications that require network access to service activation must be registered with puppet.

Continued on next page

Table 4.4 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
CM-2(6)	Baseline Configuration (Control Enhancement)	Configuration Management	As a project, SIMP is developmental only. The environments where it is tested is up to the implementation. Development testing is performed on SIMP in environments that have a code base frozen.
CM-3	Configuration Change Control	Configuration Management	
CM-3(1)	Configuration Change Control (Control Enhancement)	Configuration Management	
CM-3(2)	Configuration Change Control (Control Enhancement)	Configuration Management	
CM-3(3)	Configuration Change Control (Control Enhancement)	Configuration Management	Configuration changes in SIMP are automated using a combination of puppet, yum, and rsync. While not all files on an operating system are managed by those mechanisms, many are. Changes to critical files that are managed by puppet, revert back to their original state. These mechanisms were not meant to defeat an attack by a malicious insider.
CM-3(4)	Configuration Change Control (Control Enhancement)	Configuration Management	
CM-4	Security Impact Analysis	Configuration Management	All features or bugs in SIMP are vetted through the development process by being placed on the product backlog and discussed with the entire team. There is a security representative on the SIMP team that is part of that vetting process.
CM-4(1)	Security Impact Analysis (Control Enhancement)	Configuration Management	
CM-4(2)	Security Impact Analysis (Control Enhancement)	Configuration Management	

Continued on next page

Table 4.4 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
CM-5	Access Restrictions for Change	Configuration Management	SIMP can only meet the enforcement part of this control. The remainder must be met by the environment that SIMP is implemented in. Changes to a SIMP based systems are enforced with built in Unix/LDAP groups. Only someone with sudo or sudosh access (usually an admin group) can apply changes to the environment
CM-5(1)	Access Restrictions for Change (Control Enhancement)	Configuration Management	SIMP can only meet the enforcement part of this control. The remainder must be met by the environment that SIMP is implemented in. Changes to a SIMP based systems are enforced with built in Unix/LDAP groups. Only someone with sudo or sudosh access (usually an admin group) can apply changes to the environment
CM-5(2)	Access Restrictions for Change (Control Enhancement)	Configuration Management	
CM-5(3)	Access Restrictions for Change (Control Enhancement)	Configuration Management	Redhat and Centos packages are signed with gpg keys. Those keys are vendor specific. Package installation occurs only when those gpgkeys are validate using the installed gpg public keys for the operating system. SIMP specific RPMS that were developed are signed using keys generate by the development team.
CM-5(4)	Access Restrictions for Change (Control Enhancement)	Configuration Management	
CM-5(5)	Access Restrictions for Change (Control Enhancement)	Configuration Management	

Continued on next page

Table 4.4 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
CM-5(6)	Access Restrictions for Change (Control Enhancement)	Configuration Management	
CM-5(7)	Access Restrictions for Change (Control Enhancement)	Configuration Management	Most of the critical files that are managed by puppet cannot be permanently changed on a puppet client without disabling puppet and rsync. If they are changed, puppet will revert them back to their original state.
CM-6	Configuration Settings	Configuration Management	Part "d" of this control is met my SIMP. The others are not. SIMP uses puppet to monitor changes to configuration settings. If changes to puppet controlled settings are manually made, they revert back to their original state.
CM-6(1)	Configuration Settings (Control Enhancement)	Configuration Management	The puppet master is the central point of management for a SIMP system. While not required, the puppet master usually hosts a kickstart server so that clients are built the same every time.
CM-6(2)	Configuration Settings (Control Enhancement)	Configuration Management	Puppet is not intended to be a security mechanism to prevent unauthorized changes to files. For files that are managed by puppet that changed, they will revert back to their original state. This control is really about protecting from unauthorized changes so access control to the puppet master should suffice to meet it. Changes to files are audited using auditd. Puppet changes are also audited. It's up to the implementation to perform altering on those changes.

Continued on next page

Table 4.4 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
CM-6(3)	Configuration Settings (Control Enhancement)	Configuration Management	This control is not fully met by SIMP. It's important to point out that SIMP does provide logging of events to syslog. It's currently up to the implementation to alert on those events.
CM-7	Least Functionality	Configuration Management	There isn't an explicit list of services that SIMP denies. Instead, it was built to provide only the essential functionality. Additional services get added only as needed.
CM-7(1)	Least Functionality (Control Enhancement)	Configuration Management	
CM-7(2)	Least Functionality (Control Enhancement)	Configuration Management	Applications can be installed, but new services will not run unless first registered with puppet. Additionally, puppet modules must be modified to ensure that IPtables opens up the necessary services. Minimally, for a service to remain active, it must be registered with puppet or the svckill.rb script will stop them.To be clear, there is nothing in SIMP that prevents the installation of RPMs (from the command line or YUM).
CM-7(3)	Least Functionality (Control Enhancement)	Configuration Management	The registration process for ports, protocols, and services are handled via puppet.
CM-8	Information System Component Inventory	Configuration Management	
CM-8(1)	Information System Component Inventory (Control Enhancement)	Configuration Management	
CM-8(2)	Information System Component Inventory (Control Enhancement)	Configuration Management	To the extent possible, puppet tracks clients that are within it's control. It's not meant to be a true inventory mechanism.

Continued on next page

Table 4.4 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
CM-8(3)	Information System Component Inventory (Control Enhancement)	Configuration Management	
CM-8(4)	Information System Component Inventory (Control Enhancement)	Configuration Management	
CM-8(5)	Information System Component Inventory (Control Enhancement)	Configuration Management	
CM-8(6)	Information System Component Inventory (Control Enhancement)	Configuration Management	
CM-9	Configuration Management Plan	Configuration Management	
CM-9(1)	Configuration Management Plan (Control Enhancement)	Configuration Management	
CP-1	Contingency Planning Policy and Procedures	Contingency Planning	
CP-2	Contingency Plan	Contingency Planning	
CP-2(1)	Contingency Plan (Control Enhancement)	Contingency Planning	
CP-2(2)	Contingency Plan (Control Enhancement)	Contingency Planning	
CP-2(3)	Contingency Plan (Control Enhancement)	Contingency Planning	
CP-2(4)	Contingency Plan (Control Enhancement)	Contingency Planning	
CP-2(5)	Contingency Plan (Control Enhancement)	Contingency Planning	
CP-2(6)	Contingency Plan (Control Enhancement)	Contingency Planning	
CP-3	Contingency Training	Contingency Planning	
CP-3(1)	Contingency Training (Control Enhancement)	Contingency Planning	
CP-3(2)	Contingency Training (Control Enhancement)	Contingency Planning	
CP-4	Contingency Plan Testing and Exercises	Contingency Planning	
CP-4(1)	Contingency Plan Testing and Exercises (Control Enhancement)	Contingency Planning	
CP-4(2)	Contingency Plan Testing and Exercises (Control Enhancement)	Contingency Planning	
CP-4(3)	Contingency Plan Testing and Exercises (Control Enhancement)	Contingency Planning	
CP-6	Alternate Storage Site	Contingency Planning	

Continued on next page

Table 4.4 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
CP-6(1)	Alternate Storage Site (Control Enhancement)	Contingency Planning	
CP-6(2)	Alternate Storage Site (Control Enhancement)	Contingency Planning	
CP-6(3)	Alternate Storage Site (Control Enhancement)	Contingency Planning	
CP-7	Alternate Processing Site	Contingency Planning	
CP-7(1)	Alternate Processing Site (Control Enhancement)	Contingency Planning	
CP-7(2)	Alternate Processing Site (Control Enhancement)	Contingency Planning	
CP-7(3)	Alternate Processing Site (Control Enhancement)	Contingency Planning	
CP-7(4)	Alternate Processing Site (Control Enhancement)	Contingency Planning	
CP-7(5)	Alternate Processing Site (Control Enhancement)	Contingency Planning	
CP-8	Telecommunications Services	Contingency Planning	
CP-8(1)	Telecommunications Services (Control Enhancement)	Contingency Planning	
CP-8(2)	Telecommunications Services (Control Enhancement)	Contingency Planning	
CP-8(3)	Telecommunications Services (Control Enhancement)	Contingency Planning	
CP-8(4)	Telecommunications Services (Control Enhancement)	Contingency Planning	
CP-9	Information System Backup	Contingency Planning	The BackupPC module is not currently available in SIMP 5.0.
CP-9(1)	Information System Backup (Control Enhancement)	Contingency Planning	
CP-9(2)	Information System Backup (Control Enhancement)	Contingency Planning	
CP-9(3)	Information System Backup (Control Enhancement)	Contingency Planning	
CP-9(5)	Information System Backup (Control Enhancement)	Contingency Planning	
CP-9(6)	Information System Backup (Control Enhancement)	Contingency Planning	

Continued on next page

Table 4.4 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
CP-10	Information System Recovery and Reconstitution	Contingency Planning	The BackupPC module is not currently available in SIMP 5.0.
CP-10(1)	Information System Recovery and Reconstitution (Control Enhancement)	Contingency Planning	
CP-10(2)	Information System Recovery and Reconstitution (Control Enhancement)	Contingency Planning	
CP-10(3)	Information System Recovery and Reconstitution (Control Enhancement)	Contingency Planning	
CP-10(4)	Information System Recovery and Reconstitution (Control Enhancement)	Contingency Planning	
CP-10(5)	Information System Recovery and Reconstitution (Control Enhancement)	Contingency Planning	
CP-10(6)	Information System Recovery and Reconstitution (Control Enhancement)	Contingency Planning	
IR-1	Incident Response Policy and Procedures	Incident Response	
IR-2	Incident Response Training	Incident Response	
IR-2(1)	Incident Response Training (Control Enhancement)	Incident Response	
IR-2(2)	Incident Response Training (Control Enhancement)	Incident Response	
IR-3	Incident Response Testing and Exercises	Incident Response	
IR-3(1)	Incident Response Testing and Exercises (Control Enhancement)	Incident Response	
IR-4	Incident Handling	Incident Response	
IR-4(1)	Incident Handling (Control Enhancement)	Incident Response	
IR-4(2)	Incident Handling (Control Enhancement)	Incident Response	If an implementation chooses, they can leverage puppet's ability to reconfigure systems as part of incident response. While puppet is not intended to be a security product, its features can help provide security functionality such as dynamic reconfigurations.
IR-4(3)	Incident Handling (Control Enhancement)	Incident Response	

Continued on next page

Table 4.4 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
IR-4(4)	Incident Handling (Control Enhancement)	Incident Response	
IR-4(5)	Incident Handling (Control Enhancement)	Incident Response	
IR-5	Incident Monitoring	Incident Response	
IR-5(1)	Incident Monitoring (Control Enhancement)	Incident Response	
IR-6	Incident Reporting	Incident Response	
IR-6(1)	Incident Reporting (Control Enhancement)	Incident Response	
IR-6(2)	Incident Reporting (Control Enhancement)	Incident Response	
IR-7	Incident Response Assistance	Incident Response	
IR-7(1)	Incident Response Assistance (Control Enhancement)	Incident Response	
IR-8	Incident Response Plan	Incident Response	
MA-1	System Maintenance Policy and Procedures	Maintenance	
MA-2	Controlled Maintenance	Maintenance	
MA-2(1)	Controlled Maintenance (Control Enhancement)	Maintenance	
MA-2(2)	Controlled Maintenance (Control Enhancement)	Maintenance	
MA-3	Maintenance Tools	Maintenance	
MA-3(1)	Maintenance Tools (Control Enhancement)	Maintenance	
MA-3(2)	Maintenance Tools (Control Enhancement)	Maintenance	
MA-3(3)	Maintenance Tools (Control Enhancement)	Maintenance	
MA-3(4)	Maintenance Tools (Control Enhancement)	Maintenance	

Continued on next page

Table 4.4 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
MA-4	Non-Local Maintenance	Maintenance	Remote maintenance can be performed on SIMP using SSH or direct console access. 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's up to the implementation to decide how to distribute authentication information for remote maintenance.
MA-4(1)	Non-Local Maintenance (Control Enhancement)	Maintenance	Remote maintenance can be performed on SIMP using SSH or direct console access. 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. Audting of those actions also occurs in accordance with the configured audit policy. It's up to the implementation to decide how to distribute authentication information for remote maintenance
MA-4(2)	Non-Local Maintenance (Control Enhancement)	Maintenance	
MA-4(3)	Non-Local Maintenance (Control Enhancement)	Maintenance	
MA-4(4)	Non-Local Maintenance (Control Enhancement)	Maintenance	
MA-4(5)	Non-Local Maintenance (Control Enhancement)	Maintenance	
MA-4(6)	Non-Local Maintenance (Control Enhancement)	Maintenance	Remote maintenance is performed using SSH. SSH inherently provides confidentiality and integrity of data while in transit.

Continued on next page

Table 4.4 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
MA-4(7)	Non-Local Maintenance (Control Enhancement)	Maintenance	
MA-5	Maintenance Personnel	Maintenance	
MA-5(1)	Maintenance Personnel (Control Enhancement)	Maintenance	
MA-5(2)	Maintenance Personnel (Control Enhancement)	Maintenance	
MA-5(3)	Maintenance Personnel (Control Enhancement)	Maintenance	
MA-5(4)	Maintenance Personnel (Control Enhancement)	Maintenance	
MA-6	Timely Maintenance	Maintenance	
MP-1	Media Protection Policy and Procedures	Media Protection	
MP-2	Media Access	Media Protection	
MP-2(1)	Media Access (Control Enhancement)	Media Protection	
MP-2(2)	Media Access (Control Enhancement)	Media Protection	
MP-4	Media Storage	Media Protection	
MP-5	Media Transport	Media Protection	
MP-5(1)	Media Transport (Control Enhancement)	Media Protection	
MP-5(2)	Media Transport (Control Enhancement)	Media Protection	
MP-5(3)	Media Transport (Control Enhancement)	Media Protection	
MP-5(4)	Media Transport (Control Enhancement)	Media Protection	
MP-6	Media Sanitization	Media Protection	
MP-6(1)	Media Sanitization (Control Enhancement)	Media Protection	
MP-6(2)	Media Sanitization (Control Enhancement)	Media Protection	
MP-6(3)	Media Sanitization (Control Enhancement)	Media Protection	
MP-6(4)	Media Sanitization (Control Enhancement)	Media Protection	
MP-6(5)	Media Sanitization (Control Enhancement)	Media Protection	
MP-6(6)	Media Sanitization (Control Enhancement)	Media Protection	
PE-1	Physical and Environmental Protection Policy and Procedures	Physical and Environmental Protection	
PE-2	Physical Access Authorizations	Physical and Environmental Protection	

Continued on next page

Table 4.4 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
PE-2(1)	Physical Access Authorizations (Control Enhancement)	Physical and Environmental Protection	
PE-2(2)	Physical Access Authorizations (Control Enhancement)	Physical and Environmental Protection	
PE-2(3)	Physical Access Authorizations (Control Enhancement)	Physical and Environmental Protection	
PE-3	Physical Access Control	Physical and Environmental Protection	
PE-3(1)	Physical Access Control (Control Enhancement)	Physical and Environmental Protection	
PE-3(2)	Physical Access Control (Control Enhancement)	Physical and Environmental Protection	
PE-3(3)	Physical Access Control (Control Enhancement)	Physical and Environmental Protection	
PE-3(4)	Physical Access Control (Control Enhancement)	Physical and Environmental Protection	
PE-3(5)	Physical Access Control (Control Enhancement)	Physical and Environmental Protection	
PE-3(6)	Physical Access Control (Control Enhancement)	Physical and Environmental Protection	
PE-4	Access Control for Transmission Medium	Physical and Environmental Protection	
PE-5	Access Control for Output Devices	Physical and Environmental Protection	
PE-6	Monitoring Physical Access	Physical and Environmental Protection	
PE-6(1)	Monitoring Physical Access (Control Enhancement)	Physical and Environmental Protection	
PE-6(2)	Monitoring Physical Access (Control Enhancement)	Physical and Environmental Protection	
PE-7	Visitor Control	Physical and Environmental Protection	
PE-7(1)	Visitor Control (Control Enhancement)	Physical and Environmental Protection	
PE-7(2)	Visitor Control (Control Enhancement)	Physical and Environmental Protection	
PE-8	Access Records	Physical and Environmental Protection	
PE-8(1)	Access Records (Control Enhancement)	Physical and Environmental Protection	
PE-8(2)	Access Records (Control Enhancement)	Physical and Environmental Protection	
PE-9	Power Equipment and Power Cabling	Physical and Environmental Protection	

Continued on next page

Table 4.4 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
PE-9(1)	Power Equipment and Power Cabling (Control Enhancement)	Physical and Environmental Protection	
PE-9(2)	Power Equipment and Power Cabling (Control Enhancement)	Physical and Environmental Protection	
PE-10	Emergency Shutoff	Physical and Environmental Protection	
PE-10(1)	Emergency Shutoff (Control Enhancement)	Physical and Environmental Protection	
PE-11	Emergence Power	Physical and Environmental Protection	
PE-11(1)	Emergence Power (Control Enhancement)	Physical and Environmental Protection	
PE-11(2)	Emergence Power (Control Enhancement)	Physical and Environmental Protection	
PE-12	Emergency Lighting	Physical and Environmental Protection	
PE-12(1)	Emergency Lighting (Control Enhancement)	Physical and Environmental Protection	
PE-13	Fire Protection	Physical and Environmental Protection	
PE-13(1)	Fire Protection (Control Enhancement)	Physical and Environmental Protection	
PE-13(2)	Fire Protection (Control Enhancement)	Physical and Environmental Protection	
PE-13(3)	Fire Protection (Control Enhancement)	Physical and Environmental Protection	
PE-13(4)	Fire Protection (Control Enhancement)	Physical and Environmental Protection	
PE-14	Temperature and Humidity Controls	Physical and Environmental Protection	
PE-14(1)	Temperature and Humidity Controls (Control Enhancement)	Physical and Environmental Protection	
PE-14(2)	Temperature and Humidity Controls (Control Enhancement)	Physical and Environmental Protection	
PE-15	Water Damage Protection	Physical and Environmental Protection	
PE-15(1)	Water Damage Protection (Control Enhancement)	Physical and Environmental Protection	
PE-16	Delivery and Removal	Physical and Environmental Protection	
PE-17	Alternate Work Site	Physical and Environmental Protection	
PE-18	Location of Information System Components	Physical and Environmental Protection	

Continued on next page

Table 4.4 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
PE-18(1)	Location of Information System Components (Control Enhancement)	Physical and Environmental Protection	
PE-19	Information Leakage	Physical and Environmental Protection	
SI-1	System and Information Integrity Policy and Procedures	System and Information Integrity	
SI-2(1)	Flaw Remediation (Control Enhancement)	System and Information Integrity	Patches that are part of the software base for SIMP are tested within the development environment. There is automated testing that is constantly being extended to test more features. There are times that patches to the base operating system (Centos or RedHat) are needed to resolve issues in SIMP. Those are also tested at build time, but require additional testing by implementations as patches are released from vendors. It's also important to note that SIMP is packaged and delivered decoupled with the operating system source files. It's up to the implementation to test vendor specific patches that are not part of the SIMP code base. Flaws are tracked using the software project management tool Redmine.
SI-2(2)	Flaw Remediation (Control Enhancement)	System and Information Integrity	
SI-2(3)	Flaw Remediation (Control Enhancement)	System and Information Integrity	

Continued on next page

Table 4.4 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
SI-2(4)	Flaw Remediation (Control Enhancement)	System and Information Integrity	SIMP uses the yellowdog update manager (YUM) to deliver software patches to clients. Each installation usually has at least one YUM repository. There is also a cronjob running that runs once per day. It's the responsibility of the implementation to get patches to the yum server. Once they are there, the cron job will perform a yum update and the patches will be applied.
SI-3	Malicious Code Protection	System and Information Integrity	SIMP has modules available for mcafee and ClamAV. The ClamAV. Implementations need need to provide their own version of the mcafee software for the module to work. That module comes with the ability to sync dat updates to clients via rsync. The modulde does NOT specify how often and what files systems should be scanned. SIMP also implements the open source tool chkrootkit that comes installed by default.
SI-3(1)	Malicious Code Protection (Control Enhancement)	System and Information Integrity	The provided anti-virus modules are installed via puppet modules. Those modules include the ability to sycn data file updates via rsync. Therefore, all management of malicious code detection is done centrally.
SI-3(2)	Malicious Code Protection (Control Enhancement)	System and Information Integrity	
SI-3(3)	Malicious Code Protection (Control Enhancement)	System and Information Integrity	
SI-3(4)	Malicious Code Protection (Control Enhancement)	System and Information Integrity	
SI-3(5)	Malicious Code Protection (Control Enhancement)	System and Information Integrity	
SI-3(6)	Malicious Code Protection (Control Enhancement)	System and Information Integrity	

Continued on next page

Table 4.4 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
SI-4	Information System Monitoring Tools and Techniques	System and Information Integrity	
SI-4(1)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-4(2)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-4(3)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-4(4)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-4(5)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-4(6)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-4(7)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-4(8)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-4(9)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-4(10)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-4(11)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	

Continued on next page

Table 4.4 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
SI-4(12)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-4(13)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-4(14)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-4(15)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-4(16)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-4(17)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-5	System Alerts, Advisories, and Directives	System and Information Integrity	The only part of the control (a) that is met by SIMP, is the tracking of security alerts for products that are part of the code base. The development team subscribes to message boards for the main products (puppet) that are part of the packaging. RedHat/Centos advisories are also tracked out of necessity but since ALL the OS files are not part of SIMP delivery, patches are not our direct responsibility.
SI-5(1)	System Alerts, Advisories, and Directives (Control Enhancement)	System and Information Integrity	

Continued on next page

Table 4.4 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
SI-6	Security Functionality Verification	System and Information Integrity	SIMP comes with an optional module to install and perform regular runs of the SCAP-Security-Guide (the checks for RHEL 7 are not yet complete/finalized). Doing so will report (for a user defined frequency) OVAL results of security settings of a host against SSG recommendations.
SI-6(1)	Security Functionality Verification (Control Enhancement)	System and Information Integrity	SIMP comes with an optional module to install and perform regular runs of the SCAP-Security-Guide. Doing so will report (for a user defined frequency) OVAL results of security settings of a host against SSG recommendations.
SI-6(2)	Security Functionality Verification (Control Enhancement)	System and Information Integrity	SIMP comes with an optional module to install and perform regular runs of the SCAP-Security-Guide. Doing so will report (for a user defined frequency) OVAL results of security settings of a host against SSG recommendations.
SI-6(3)	Security Functionality Verification (Control Enhancement)	System and Information Integrity	SIMP comes with an optional module to install and perform regular runs of the SCAP-Security-Guide. Doing so will report (for a user defined frequency) OVAL results of security settings of a host against SSG recommendations.
SI-7	Software and Information Integrity	System and Information Integrity	SIMP comes with AIDE installed. Puppet also serves the purpose of checking the integrity of files. During each client run, a change in file integrity means the file needs to be restored to it's original state.

Continued on next page

Table 4.4 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
SI-7(1)	Software and Information Integrity (Control Enhancement)	System and Information Integrity	AIDE baselines are not performed beyond initial install unless otherwise configured. Implementations can re-baseline the database.
SI-7(2)	Software and Information Integrity (Control Enhancement)	System and Information Integrity	
SI-7(3)	Software and Information Integrity (Control Enhancement)	System and Information Integrity	AIDE is managed by puppet and is therefore centrally managed.
SI-7(4)	Software and Information Integrity (Control Enhancement)	System and Information Integrity	
SI-8	Spam Protection	System and Information Integrity	
SI-8(1)	Spam Protection (Control Enhancement)	System and Information Integrity	
SI-8(2)	Spam Protection (Control Enhancement)	System and Information Integrity	
SI-9	Information Input Restrictions	System and Information Integrity	
SI-10	Information Input Validation	System and Information Integrity	
SI-11	Error Handling	System and Information Integrity	
SI-13	Predictable Failure Prevention	System and Information Integrity	
SI-13(1)	Predictable Failure Prevention (Control Enhancement)	System and Information Integrity	
SI-13(2)	Predictable Failure Prevention (Control Enhancement)	System and Information Integrity	
SI-13(3)	Predictable Failure Prevention (Control Enhancement)	System and Information Integrity	
SI-13(4)	Predictable Failure Prevention (Control Enhancement)	System and Information Integrity	

Table: SIMP SCTM

SIMP SCTM Management Controls

Control ID	Control Name	Control Family	SIMP Implementation Method
AT-1	Security Awareness and Training Policy and Procedures	Awareness and Training	

Continued on next page

Table 4.5 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
AT-2(1)	Security Awareness (Control Enhancement)	Awareness and Training	
AT-3	Security Training	Awareness and Training	
AT-3(1)	Security Training (Control Enhancement)	Awareness and Training	
AT-3(2)	Security Training (Control Enhancement)	Awareness and Training	
AT-4	Security Training Records	Awareness and Training	
AT-5	Contacts with Security Groups and Associations	Awareness and Training	
CM-1	Configuration Management Policy and Procedures	Configuration Management	
CM-2	Baseline Configuration	Configuration Management	SIMP has strictly enforced version control during development. The baseline files for SIMP are kept and maintained in a git repository. Files are packaged and a series of auto tests are performed on each release. Once released, there is a version number associated for distribution. Additionally, custom puppet modules are in the form of RPMs and have version numbers associated with them. All documentation is also built with source code.
CM-2(1)	Baseline Configuration (Control Enhancement)	Configuration Management	
CM-2(2)	Baseline Configuration (Control Enhancement)	Configuration Management	SIMP has strictly enforced version control during development. The baseline files for SIMP are kept and maintained in a git repository. Files are packaged and a series of auto tests are performed on the release. Once released, there is a version number associated for distribution. All documentation is also built with source code.
CM-2(3)	Baseline Configuration (Control Enhancement)	Configuration Management	All old versions of SIMP remain in the code repository.
CM-2(4)	Baseline Configuration (Control Enhancement)	Configuration Management	

Continued on next page

Table 4.5 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
CM-2(5)	Baseline Configuration (Control Enhancement)	Configuration Management	1. SIMP provides a minimal list of packages and services installed. The minimal list of packages can be found in kickstart files and the appendix of this document. Additional packages are installed by each implementation or as SIMP modules are applied. b. It's not feasible to technically deny additional applications from being installed. There is nothing in SIMP that can stop and RPM from being applied. Applications that require network access to service activation must be registered with puppet.
CM-2(6)	Baseline Configuration (Control Enhancement)	Configuration Management	As a project, SIMP is developmental only. The environments where it is tested is up to the implementation. Development testing is performed on SIMP in environments that have a code base frozen.
CM-3	Configuration Change Control	Configuration Management	
CM-3(1)	Configuration Change Control (Control Enhancement)	Configuration Management	
CM-3(2)	Configuration Change Control (Control Enhancement)	Configuration Management	

Continued on next page

Table 4.5 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
CM-3(3)	Configuration Change Control (Control Enhancement)	Configuration Management	Configuration changes in SIMP are automated using a combination of puppet, yum, and rsync. While not all files on an operating system are managed by those mechanisms, many are. Changes to critical files that are managed by puppet, revert back to their original state. These mechanisms were not meant to defeat an attack by a malicious insider.
CM-3(4)	Configuration Change Control (Control Enhancement)	Configuration Management	
CM-4	Security Impact Analysis	Configuration Management	All features or bugs in SIMP are vetted through the development process by being placed on the product backlog and discussed with the entire team. There is a security representative on the SIMP team that is part of that vetting process.
CM-4(1)	Security Impact Analysis (Control Enhancement)	Configuration Management	
CM-4(2)	Security Impact Analysis (Control Enhancement)	Configuration Management	
CM-5	Access Restrictions for Change	Configuration Management	SIMP can only meet the enforcement part of this control. The remainder must be met by the environment that SIMP is implemented in. Changes to a SIMP based systems are enforced with built in Unix/LDAP groups. Only someone with sudo or sudosh access (usually an admin group) can apply changes to the environment

Continued on next page

Table 4.5 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
CM-5(1)	Access Restrictions for Change (Control Enhancement)	Configuration Management	SIMP can only meet the enforcement part of this control. The remainder must be met by the environment that SIMP is implemented in. Changes to a SIMP based systems are enforced with built in Unix/LDAP groups. Only someone with sudo or sudosh access (usually an admin group) can apply changes to the environment
CM-5(2)	Access Restrictions for Change (Control Enhancement)	Configuration Management	
CM-5(3)	Access Restrictions for Change (Control Enhancement)	Configuration Management	Redhat and Centos packages are signed with gpg keys. Those keys are vendor specific. Package installation occurs only when those gpgkeys are validate using the installed gpg public keys for the operating system. SIMP specific RPMS that were developed are signed using keys generate by the development team.
CM-5(4)	Access Restrictions for Change (Control Enhancement)	Configuration Management	
CM-5(5)	Access Restrictions for Change (Control Enhancement)	Configuration Management	
CM-5(6)	Access Restrictions for Change (Control Enhancement)	Configuration Management	
CM-5(7)	Access Restrictions for Change (Control Enhancement)	Configuration Management	Most of the critical files that are managed by puppet cannot be permanently changed on a puppet client without disabling puppet and rsync. If they are changed, puppet will revert them back to their original state.

Continued on next page

Table 4.5 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
CM-6	Configuration Settings	Configuration Management	Part "d" of this control is met my SIMP. The others are not. SIMP uses puppet to monitor changes to configuration settings. If changes to puppet controlled settings are manually made, they revert back to their original state.
CM-6(1)	Configuration Settings (Control Enhancement)	Configuration Management	The puppet master is the central point of management for a SIMP system. While not required, the puppet master usually hosts a kickstart server so that clients are built the same every time.
CM-6(2)	Configuration Settings (Control Enhancement)	Configuration Management	Puppet is not intended to be a security mechanism to prevent unauthorized changes to files. For files that are managed by puppet that changed, they will revert back to their original state. This control is really about protecting from unauthorized changes so access control to the puppet master should suffice to meet it. Changes to files are audited using auditd. Puppet changes are also audited. It's up to the implementation to perform altering on those changes.
CM-6(3)	Configuration Settings (Control Enhancement)	Configuration Management	This control is not fully met by SIMP. It's important to point out that SIMP does provide logging of events to syslog. It's currently up to the implementation to alert on those events.
CM-7	Least Functionality	Configuration Management	There isn't an explicit list of services that SIMP denies. Instead, it was built to provide only the essential functionality. Additional services get added only as needed.

Continued on next page

Table 4.5 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
CM-7(1)	Least Functionality (Control Enhancement)	Configuration Management	
CM-7(2)	Least Functionality (Control Enhancement)	Configuration Management	Applications can be installed, but new services will not run unless first registered with puppet. Additionally, puppet modules must be modified to ensure that IPtables opens up the necessary services. Minimally, for a service to remain active, it must be registered with puppet or the svckill.rb script will stop them.To be clear, there is nothing in SIMP that prevents the installation of RPMs (from the command line or YUM).
CM-7(3)	Least Functionality (Control Enhancement)	Configuration Management	The registration process for ports, protocols, and services are handled via puppet.
CM-8	Information System Component Inventory	Configuration Management	
CM-8(1)	Information System Component Inventory (Control Enhancement)	Configuration Management	
CM-8(2)	Information System Component Inventory (Control Enhancement)	Configuration Management	To the extent possible, puppet tracks clients that are within it's control. It's not meant to be a true inventory mechanism.
CM-8(3)	Information System Component Inventory (Control Enhancement)	Configuration Management	
CM-8(4)	Information System Component Inventory (Control Enhancement)	Configuration Management	
CM-8(5)	Information System Component Inventory (Control Enhancement)	Configuration Management	
CM-8(6)	Information System Component Inventory (Control Enhancement)	Configuration Management	
CM-9	Configuration Management Plan	Configuration Management	

Continued on next page

Table 4.5 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
CM-9(1)	Configuration Management Plan (Control Enhancement)	Configuration Management	
CP-1	Contingency Planning Policy and Procedures	Contingency Planning	
CP-2	Contingency Plan	Contingency Planning	
CP-2(1)	Contingency Plan (Control Enhancement)	Contingency Planning	
CP-2(2)	Contingency Plan (Control Enhancement)	Contingency Planning	
CP-2(3)	Contingency Plan (Control Enhancement)	Contingency Planning	
CP-2(4)	Contingency Plan (Control Enhancement)	Contingency Planning	
CP-2(5)	Contingency Plan (Control Enhancement)	Contingency Planning	
CP-2(6)	Contingency Plan (Control Enhancement)	Contingency Planning	
CP-3	Contingency Training	Contingency Planning	
CP-3(1)	Contingency Training (Control Enhancement)	Contingency Planning	
CP-3(2)	Contingency Training (Control Enhancement)	Contingency Planning	
CP-4	Contingency Plan Testing and Exercises	Contingency Planning	
CP-4(1)	Contingency Plan Testing and Exercises (Control Enhancement)	Contingency Planning	
CP-4(2)	Contingency Plan Testing and Exercises (Control Enhancement)	Contingency Planning	
CP-4(3)	Contingency Plan Testing and Exercises (Control Enhancement)	Contingency Planning	
CP-6	Alternate Storage Site	Contingency Planning	
CP-6(1)	Alternate Storage Site (Control Enhancement)	Contingency Planning	
CP-6(2)	Alternate Storage Site (Control Enhancement)	Contingency Planning	
CP-6(3)	Alternate Storage Site (Control Enhancement)	Contingency Planning	
CP-7	Alternate Processing Site	Contingency Planning	
CP-7(1)	Alternate Processing Site (Control Enhancement)	Contingency Planning	
CP-7(2)	Alternate Processing Site (Control Enhancement)	Contingency Planning	
CP-7(3)	Alternate Processing Site (Control Enhancement)	Contingency Planning	

Continued on next page

Table 4.5 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
CP-7(4)	Alternate Processing Site (Control Enhancement)	Contingency Planning	
CP-7(5)	Alternate Processing Site (Control Enhancement)	Contingency Planning	
CP-8	Telecommunications Services	Contingency Planning	
CP-8(1)	Telecommunications Services (Control Enhancement)	Contingency Planning	
CP-8(2)	Telecommunications Services (Control Enhancement)	Contingency Planning	
CP-8(3)	Telecommunications Services (Control Enhancement)	Contingency Planning	
CP-8(4)	Telecommunications Services (Control Enhancement)	Contingency Planning	
CP-9	Information System Backup	Contingency Planning	The BackupPC module is not currently available in SIMP 5.0.
CP-9(1)	Information System Backup (Control Enhancement)	Contingency Planning	
CP-9(2)	Information System Backup (Control Enhancement)	Contingency Planning	
CP-9(3)	Information System Backup (Control Enhancement)	Contingency Planning	
CP-9(5)	Information System Backup (Control Enhancement)	Contingency Planning	
CP-9(6)	Information System Backup (Control Enhancement)	Contingency Planning	
CP-10	Information System Recovery and Reconstitution	Contingency Planning	The BackupPC module is not currently available in SIMP 5.0.
CP-10(1)	Information System Recovery and Reconstitution (Control Enhancement)	Contingency Planning	
CP-10(2)	Information System Recovery and Reconstitution (Control Enhancement)	Contingency Planning	
CP-10(3)	Information System Recovery and Reconstitution (Control Enhancement)	Contingency Planning	

Continued on next page

Table 4.5 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
CP-10(4)	Information System Recovery and Reconstitution (Control Enhancement)	Contingency Planning	
CP-10(5)	Information System Recovery and Reconstitution (Control Enhancement)	Contingency Planning	
CP-10(6)	Information System Recovery and Reconstitution (Control Enhancement)	Contingency Planning	
IR-1	Incident Response Policy and Procedures	Incident Response	
IR-2	Incident Response Training	Incident Response	
IR-2(1)	Incident Response Training (Control Enhancement)	Incident Response	
IR-2(2)	Incident Response Training (Control Enhancement)	Incident Response	
IR-3	Incident Response Testing and Exercises	Incident Response	
IR-3(1)	Incident Response Testing and Exercises (Control Enhancement)	Incident Response	
IR-4	Incident Handling	Incident Response	
IR-4(1)	Incident Handling (Control Enhancement)	Incident Response	
IR-4(2)	Incident Handling (Control Enhancement)	Incident Response	If an implementation chooses, they can leverage puppet's ability to reconfigure systems as part of incident response. While puppet is not intended to be a security product, its features can help provide security functionality such as dynamic reconfigurations.
IR-4(3)	Incident Handling (Control Enhancement)	Incident Response	
IR-4(4)	Incident Handling (Control Enhancement)	Incident Response	
IR-4(5)	Incident Handling (Control Enhancement)	Incident Response	
IR-5	Incident Monitoring	Incident Response	
IR-5(1)	Incident Monitoring (Control Enhancement)	Incident Response	
IR-6	Incident Reporting	Incident Response	
IR-6(1)	Incident Reporting (Control Enhancement)	Incident Response	
IR-6(2)	Incident Reporting (Control Enhancement)	Incident Response	

Continued on next page

Table 4.5 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
IR-7	Incident Response Assistance	Incident Response	
IR-7(1)	Incident Response Assistance (Control Enhancement)	Incident Response	
IR-8	Incident Response Plan	Incident Response	
MA-1	System Maintenance Policy and Procedures	Maintenance	
MA-2	Controlled Maintenance	Maintenance	
MA-2(1)	Controlled Maintenance (Control Enhancement)	Maintenance	
MA-2(2)	Controlled Maintenance (Control Enhancement)	Maintenance	
MA-3	Maintenance Tools	Maintenance	
MA-3(1)	Maintenance Tools (Control Enhancement)	Maintenance	
MA-3(2)	Maintenance Tools (Control Enhancement)	Maintenance	
MA-3(3)	Maintenance Tools (Control Enhancement)	Maintenance	
MA-3(4)	Maintenance Tools (Control Enhancement)	Maintenance	
MA-4	Non-Local Maintenance	Maintenance	Remote maintenance can be performed on SIMP using SSH or direct console access. 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's up to the implementation to decide how to distribute authentication information for remote maintenance.

Continued on next page

Table 4.5 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
MA-4(1)	Non-Local Maintenance (Control Enhancement)	Maintenance	Remote maintenance can be performed on SIMP using SSH or direct console access. 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. Audting of those actions also occurs in accordance with the configured audit policy. It's up to the implementation to decide how to distribute authentication information for remote maintenance
MA-4(2)	Non-Local Maintenance (Control Enhancement)	Maintenance	
MA-4(3)	Non-Local Maintenance (Control Enhancement)	Maintenance	
MA-4(4)	Non-Local Maintenance (Control Enhancement)	Maintenance	
MA-4(5)	Non-Local Maintenance (Control Enhancement)	Maintenance	
MA-4(6)	Non-Local Maintenance (Control Enhancement)	Maintenance	Remote maintenance is performed using SSH. SSH inherently provides confidentiality and integrity of data while in transit.
MA-4(7)	Non-Local Maintenance (Control Enhancement)	Maintenance	
MA-5	Maintenance Personnel	Maintenance	
MA-5(1)	Maintenance Personnel (Control Enhancement)	Maintenance	
MA-5(2)	Maintenance Personnel (Control Enhancement)	Maintenance	
MA-5(3)	Maintenance Personnel (Control Enhancement)	Maintenance	
MA-5(4)	Maintenance Personnel (Control Enhancement)	Maintenance	
MA-6	Timely Maintenance	Maintenance	
MP-1	Media Protection Policy and Procedures	Media Protection	
MP-2	Media Access	Media Protection	
MP-2(1)	Media Access (Control Enhancement)	Media Protection	

Continued on next page

Table 4.5 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
MP-2(2)	Media Access (Control Enhancement)	Media Protection	
MP-4	Media Storage	Media Protection	
MP-5	Media Transport	Media Protection	
MP-5(1)	Media Transport (Control Enhancement)	Media Protection	
MP-5(2)	Media Transport (Control Enhancement)	Media Protection	
MP-5(3)	Media Transport (Control Enhancement)	Media Protection	
MP-5(4)	Media Transport (Control Enhancement)	Media Protection	
MP-6	Media Sanitization	Media Protection	
MP-6(1)	Media Sanitization (Control Enhancement)	Media Protection	
MP-6(2)	Media Sanitization (Control Enhancement)	Media Protection	
MP-6(3)	Media Sanitization (Control Enhancement)	Media Protection	
MP-6(4)	Media Sanitization (Control Enhancement)	Media Protection	
MP-6(5)	Media Sanitization (Control Enhancement)	Media Protection	
MP-6(6)	Media Sanitization (Control Enhancement)	Media Protection	
PE-1	Physical and Environmental Protection Policy and Procedures	Physical and Environmental Protection	
PE-2	Physical Access Authorizations	Physical and Environmental Protection	
PE-2(1)	Physical Access Authorizations (Control Enhancement)	Physical and Environmental Protection	
PE-2(2)	Physical Access Authorizations (Control Enhancement)	Physical and Environmental Protection	
PE-2(3)	Physical Access Authorizations (Control Enhancement)	Physical and Environmental Protection	
PE-3	Physical Access Control	Physical and Environmental Protection	
PE-3(1)	Physical Access Control (Control Enhancement)	Physical and Environmental Protection	
PE-3(2)	Physical Access Control (Control Enhancement)	Physical and Environmental Protection	
PE-3(3)	Physical Access Control (Control Enhancement)	Physical and Environmental Protection	
PE-3(4)	Physical Access Control (Control Enhancement)	Physical and Environmental Protection	

Continued on next page

Table 4.5 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
PE-3(5)	Physical Access Control (Control Enhancement)	Physical and Environmental Protection	
PE-3(6)	Physical Access Control (Control Enhancement)	Physical and Environmental Protection	
PE-4	Access Control for Transmission Medium	Physical and Environmental Protection	
PE-5	Access Control for Output Devices	Physical and Environmental Protection	
PE-6	Monitoring Physical Access	Physical and Environmental Protection	
PE-6(1)	Monitoring Physical Access (Control Enhancement)	Physical and Environmental Protection	
PE-6(2)	Monitoring Physical Access (Control Enhancement)	Physical and Environmental Protection	
PE-7	Visitor Control	Physical and Environmental Protection	
PE-7(1)	Visitor Control (Control Enhancement)	Physical and Environmental Protection	
PE-7(2)	Visitor Control (Control Enhancement)	Physical and Environmental Protection	
PE-8	Access Records	Physical and Environmental Protection	
PE-8(1)	Access Records (Control Enhancement)	Physical and Environmental Protection	
PE-8(2)	Access Records (Control Enhancement)	Physical and Environmental Protection	
PE-9	Power Equipment and Power Cabling	Physical and Environmental Protection	
PE-9(1)	Power Equipment and Power Cabling (Control Enhancement)	Physical and Environmental Protection	
PE-9(2)	Power Equipment and Power Cabling (Control Enhancement)	Physical and Environmental Protection	
PE-10	Emergency Shutoff	Physical and Environmental Protection	
PE-10(1)	Emergency Shutoff (Control Enhancement)	Physical and Environmental Protection	
PE-11	Emergence Power	Physical and Environmental Protection	
PE-11(1)	Emergence Power (Control Enhancement)	Physical and Environmental Protection	
PE-11(2)	Emergence Power (Control Enhancement)	Physical and Environmental Protection	
PE-12	Emergency Lighting	Physical and Environmental Protection	

Continued on next page

Table 4.5 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
PE-12(1)	Emergency Lighting (Control Enhancement)	Physical and Environmental Protection	
PE-13	Fire Protection	Physical and Environmental Protection	
PE-13(1)	Fire Protection (Control Enhancement)	Physical and Environmental Protection	
PE-13(2)	Fire Protection (Control Enhancement)	Physical and Environmental Protection	
PE-13(3)	Fire Protection (Control Enhancement)	Physical and Environmental Protection	
PE-13(4)	Fire Protection (Control Enhancement)	Physical and Environmental Protection	
PE-14	Temperature and Humidity Controls	Physical and Environmental Protection	
PE-14(1)	Temperature and Humidity Controls (Control Enhancement)	Physical and Environmental Protection	
PE-14(2)	Temperature and Humidity Controls (Control Enhancement)	Physical and Environmental Protection	
PE-15	Water Damage Protection	Physical and Environmental Protection	
PE-15(1)	Water Damage Protection (Control Enhancement)	Physical and Environmental Protection	
PE-16	Delivery and Removal	Physical and Environmental Protection	
PE-17	Alternate Work Site	Physical and Environmental Protection	
PE-18	Location of Information System Components	Physical and Environmental Protection	
PE-18(1)	Location of Information System Components (Control Enhancement)	Physical and Environmental Protection	
PE-19	Information Leakage	Physical and Environmental Protection	
SI-1	System and Information Integrity Policy and Procedures	System and Information Integrity	

Continued on next page

Table 4.5 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
SI-2(1)	Flaw Remediation (Control Enhancement)	System and Information Integrity	Patches that are part of the software base for SIMP are tested within the development environment. There is automated testing that is constantly being extended to test more features. There are times that patches to the base operating system (Centos or RedHat) are needed to resolve issues in SIMP. Those are also tested at build time, but require additional testing by implementations as patches are released from vendors. It's also important to note that SIMP is packaged and delivered decoupled with the operating system source files. It's up to the implementation to test vendor specific patches that are not part of the SIMP code base. Flaws are tracked using the software project management tool Redmine.
SI-2(2)	Flaw Remediation (Control Enhancement)	System and Information Integrity	
SI-2(3)	Flaw Remediation (Control Enhancement)	System and Information Integrity	
SI-2(4)	Flaw Remediation (Control Enhancement)	System and Information Integrity	SIMP uses the yellowdog update manager (YUM) to deliver software patches to clients. Each installation usually has at least one YUM repository. There is also a cronjob running that runs once per day. It's the responsibility of the implementation to get patches to the yum server. Once they are there, the cron job will perform a yum update and the patches will be applied.

Continued on next page

Table 4.5 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
SI-3	Malicious Code Protection	System and Information Integrity	SIMP has modules available for mcafee and ClamAV. The ClamAV. Implementations need need to provide their own version of the mcafee software for the module to work. That module comes with the ability to sync dat updates to clients via rsync. The modulde does NOT specify how often and what files systems should be scanned. SIMP also implements the open source tool chkrootkit that comes installed by default.
SI-3(1)	Malicious Code Protection (Control Enhancement)	System and Information Integrity	The provided anti-virus modules are installed via puppet modules. Those modules include the ability to sycn data file updates via rsync. Therefore, all management of malicious code detection is done centrally.
SI-3(2)	Malicious Code Protection (Control Enhancement)	System and Information Integrity	
SI-3(3)	Malicious Code Protection (Control Enhancement)	System and Information Integrity	
SI-3(4)	Malicious Code Protection (Control Enhancement)	System and Information Integrity	
SI-3(5)	Malicious Code Protection (Control Enhancement)	System and Information Integrity	
SI-3(6)	Malicious Code Protection (Control Enhancement)	System and Information Integrity	
SI-4	Information System Monitoring Tools and Techniques	System and Information Integrity	
SI-4(1)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-4(2)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	

Continued on next page

Table 4.5 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
SI-4(3)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-4(4)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-4(5)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-4(6)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-4(7)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-4(8)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-4(9)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-4(10)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-4(11)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-4(12)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-4(13)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-4(14)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	

Continued on next page

Table 4.5 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
SI-4(15)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-4(16)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-4(17)	Information System Monitoring Tools and Techniques (Control Enhancement)	System and Information Integrity	
SI-5	System Alerts, Advisories, and Directives	System and Information Integrity	The only part of the control (a) that is met by SIMP, is the tracking of security alerts for products that are part of the code base. The development team subscribes to message boards for the main products (puppet) that are part of the packaging. RedHat/Centos advisories are also tracked out of necessity but since ALL the OS files are not part of SIMP delivery, patches are not our direct responsibility.
SI-5(1)	System Alerts, Advisories, and Directives (Control Enhancement)	System and Information Integrity	
SI-6	Security Functionality Verification	System and Information Integrity	SIMP comes with an optional module to install and perform regular runs of the SCAP-Security-Guide (the checks for RHEL 7 are not yet complete/finalized). Doing so will report (for a user defined frequency) OVAL results of security settings of a host against SSG recommendations.

Continued on next page

Table 4.5 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
SI-6(1)	Security Functionality Verification (Control Enhancement)	System and Information Integrity	SIMP comes with an optional module to install and perform regular runs of the SCAP-Security-Guide. Doing so will report (for a user defined frequency) OVAL results of security settings of a host against SSG recommendations.
SI-6(2)	Security Functionality Verification (Control Enhancement)	System and Information Integrity	SIMP comes with an optional module to install and perform regular runs of the SCAP-Security-Guide. Doing so will report (for a user defined frequency) OVAL results of security settings of a host against SSG recommendations.
SI-6(3)	Security Functionality Verification (Control Enhancement)	System and Information Integrity	SIMP comes with an optional module to install and perform regular runs of the SCAP-Security-Guide. Doing so will report (for a user defined frequency) OVAL results of security settings of a host against SSG recommendations.
SI-7	Software and Information Integrity	System and Information Integrity	SIMP comes with AIDE installed. Puppet also serves the purpose of checking the integrity of files. During each client run, a change in file integrity means the file needs to be restored to it's original state.
SI-7(1)	Software and Information Integrity (Control Enhancement)	System and Information Integrity	AIDE baselines are not performed beyond initial install unless otherwise configured. Implementations can re-baseline the database.
SI-7(2)	Software and Information Integrity (Control Enhancement)	System and Information Integrity	
SI-7(3)	Software and Information Integrity (Control Enhancement)	System and Information Integrity	AIDE is managed by puppet and is therefore centrally managed.

Continued on next page

Table 4.5 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
SI-7(4)	Software and Information Integrity (Control Enhancement)	System and Information Integrity	
SI-8	Spam Protection	System and Information Integrity	
SI-8(1)	Spam Protection (Control Enhancement)	System and Information Integrity	
SI-8(2)	Spam Protection (Control Enhancement)	System and Information Integrity	
SI-9	Information Input Restrictions	System and Information Integrity	
SI-10	Information Input Validation	System and Information Integrity	
SI-11	Error Handling	System and Information Integrity	
SI-13	Predictable Failure Prevention	System and Information Integrity	
SI-13(1)	Predictable Failure Prevention (Control Enhancement)	System and Information Integrity	
SI-13(2)	Predictable Failure Prevention (Control Enhancement)	System and Information Integrity	
SI-13(3)	Predictable Failure Prevention (Control Enhancement)	System and Information Integrity	
SI-13(4)	Predictable Failure Prevention (Control Enhancement)	System and Information Integrity	
Control ID	Control Name	Control Family	SIMP Implementation Method
Control ID	Control Name	Control Family	SIMP Implementation Method
CA-1	Security Assessment and Authorization Policies	Security Assessment and Authorization	
CA-2	Security Assessments	Security Assessment and Authorization	
CA-2(1)	Security Assessments (Control Enhancement)	Security Assessment and Authorization	
CA-2(2)	Security Assessments (Control Enhancement)	Security Assessment and Authorization	
CA-3	Information System Connections	Security Assessment and Authorization	
CA-3(1)	Information System Connections (Control Enhancement)	Security Assessment and Authorization	
CA-3(2)	Information System Connections (Control Enhancement)	Security Assessment and Authorization	
CA-5	Plan of Action and Milestones	Security Assessment and Authorization	

Continued on next page

Table 4.5 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
CA-5(1)	Plan of Action and Milestones (Control Enhancement)	Security Assessment and Authorization	
CA-6	Security Authorization	Security Assessment and Authorization	
CA-7	Continuous Monitoring	Security Assessment and Authorization	
CA-7(1)	Continuous Monitoring (Control Enhancement)	Security Assessment and Authorization	
CA-7(2)	Continuous Monitoring (Control Enhancement)	Security Assessment and Authorization	
Pl-1	Security Planning Policy and Procedures	Planning	The SIMP installation manual provides instructions for the installation of the product in a manner that is compliant with a multitude of security controls.
PL-2	System Security Plan	Planning	Security Plans are provided for specific implementations. The SIMP team will continue to develop security documentation that can be used as s resource for implementation specific System Security Plans.
PL-2(1)	System Security Plan (Control Enhancement)	Planning	TODO: Develop SIMP specific SSP.
PL-2(2)	System Security Plan (Control Enhancement)	Planning	
PL-4	Rules of Behavior	Planning	
PL-4(1)	Rules of Behavior (Control Enhancement)	Planning	
PL-5	Privacy Impact Assessment	Planning	
PL-6	Security-Related Activity Planning	Planning	
PS-1	Personnel Security Policy and Procedures	Planning	
PS-2	Position Categorization	Planning	
PS-3(2)	Personnel Screening (Control Enhancement)	Planning	
RA-1	Risk Assessment Policy and Procedures	Risk Assessment	
RA-2	Security Categorization	Risk Assessment	
RA-3	Risk Assessment	Risk Assessment	

Continued on next page

Table 4.5 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
RA-5	Vulnerability Scanning	Risk Assessment	The SIMP team performs a variety of security testing as part of the development process. Compliance and configuration checking is done using SSG. SIMP makes every effort to address problems discovered by these tools. Some configuration settings will not align with tools since the product was meant to be used for operational settings where some security features cause a loss in functionality. Implementations have the option of further hardening their system further at the risk of losing some functionality.
RA-5(1)	Vulnerability Scanning (Control Enhancement)	Risk Assessment	SCAP-Security-Guide is the two primary tool used to check for suspected configuration errors. Puppet also continues to protect clients against unwanted changes.
RA-5(2)	Vulnerability Scanning (Control Enhancement)	Risk Assessment	SCAP-Security-Guide is the two primary tool used to check for suspected configuration errors. Puppet also continues to protect clients against unwanted changes.
RA-5(3)	Vulnerability Scanning (Control Enhancement)	Risk Assessment	Regular vulnerability scanning is performed during development of SIMP.
RA-5(4)	Vulnerability Scanning (Control Enhancement)	Risk Assessment	Part of the vulnerability scanning process determines what information can be determined by a malicious outside user.
RA-5(5)	Vulnerability Scanning (Control Enhancement)	Risk Assessment	The compliance tools require that privileged accounts be used to perform testing.
RA-5(6)	Vulnerability Scanning (Control Enhancement)	Risk Assessment	

Continued on next page

Table 4.5 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
RA-5(7)	Vulnerability Scanning (Control Enhancement)	Risk Assessment	Only part of this requirement is met. SIMP can detect when any software is installed via auditd and syslog. Services that are not registered with puppet will not operate without user intervention. Those changes are also audited. SIMP does not provide the ability to alert on those actions, however, Logstash filters or Elasticsearch queries can be applied if needed.
RA-5(8)	Vulnerability Scanning (Control Enhancement)	Risk Assessment	
RA-5(9)	Vulnerability Scanning (Control Enhancement)	Risk Assessment	
SA-1	System and Services Acquisition Policy and Procedures	System and Service Acquisition	
SA-2	Allocation of Resources	System and Service Acquisition	
SA-3	Life Cycle Support	System and Service Acquisition	
SA-4	Acquisitions	System and Service Acquisition	
SA-4(1)	Acquisitions (Control Enhancement)	System and Service Acquisition	
SA-4(2)	Acquisitions (Control Enhancement)	System and Service Acquisition	
SA-4(3)	Acquisitions (Control Enhancement)	System and Service Acquisition	
SA-4(4)	Acquisitions (Control Enhancement)	System and Service Acquisition	
SA-4(5)	Acquisitions (Control Enhancement)	System and Service Acquisition	
SA-4(6)	Acquisitions (Control Enhancement)	System and Service Acquisition	
SA-4(7)	Acquisitions (Control Enhancement)	System and Service Acquisition	
SA-5	Information System Documentation	System and Service Acquisition	
SA-5(1)	Information System Documentation (Control Enhancement)	System and Service Acquisition	
SA-5(2)	Information System Documentation (Control Enhancement)	System and Service Acquisition	

Continued on next page

Table 4.5 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
SA-5(3)	Information System Documentation (Control Enhancement)	System and Service Acquisition	
SA-5(4)	Information System Documentation (Control Enhancement)	System and Service Acquisition	
SA-5(5)	Information System Documentation (Control Enhancement)	System and Service Acquisition	
SA-6	Software Usage Restrictions	System and Service Acquisition	
SA-6 (1)	Software Usage Restrictions	System and Service Acquisition	
SA-7	User Installed Software	System and Service Acquisition	
SA-8	Security Engineering Principles	System and Service Acquisition	
SA-9	External Information System Services	System and Service Acquisition	
SA-9(1)	External Information System Services (Control Enhancement)	System and Service Acquisition	
SA-10	Developer Configuration Management	System and Service Acquisition	
SA-10(1)	Developer Configuration Management (Control Enhancement)	System and Service Acquisition	
SA-10(2)	Developer Configuration Management (Control Enhancement)	System and Service Acquisition	
SA-11	Developer Security Testing	System and Service Acquisition	
SA-11(1)	Developer Security Testing (Control Enhancement)	System and Service Acquisition	
SA-11(2)	Developer Security Testing (Control Enhancement)	System and Service Acquisition	
SA-11(3)	Developer Security Testing (Control Enhancement)	System and Service Acquisition	
SA-12	Supply Chain Protection	System and Service Acquisition	
SA-12(1)	Supply Chain Protection (Control Enhancement)	System and Service Acquisition	
SA-12(2)	Supply Chain Protection (Control Enhancement)	System and Service Acquisition	
SA-12(3)	Supply Chain Protection (Control Enhancement)	System and Service Acquisition	
SA-12(4)	Supply Chain Protection (Control Enhancement)	System and Service Acquisition	

Continued on next page

Table 4.5 – continued from previous page

Control ID	Control Name	Control Family	SIMP Implementation Method
SA-12(5)	Supply Chain Protection (Control Enhancement)	System and Service Acquisition	
SA-12(6)	Supply Chain Protection (Control Enhancement)	System and Service Acquisition	
SA-12(7)	Supply Chain Protection (Control Enhancement)	System and Service Acquisition	
SA-13	Trustworthiness	System and Service Acquisition	
SA-14	Critical Information System Components	System and Service Acquisition	
SA-14(1)	Critical Information System Components (Control Enhancement)	System and Service Acquisition	

Table: Management Controls

4.6 Indices and tables

- genindex
- search

CHAPTER 5

License

5.1 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.

CHAPTER 6

Contact

If you have questions please contact the SIMP team. simp@simp-project.org

CHAPTER 7

Help

If you are looking for assistance please email the SIMP mailing lists. simp@simp-project.org

CHAPTER 8

Indices and tables

- genindex
- search

R

S

T

V

W

X

Y