Guide to the Secure Configuration of Red Hat Enterprise Linux 8

Current release

Authors	
Owner	
Version Number	0.1.44
Approved by	NOT YET APPROVED
Release Date	2019-06-13
Classification	public
Document ID	RHEL-8

Document History

Version	Author	Date	Description
0.1.44		2019-06-13	draft

Change Log in the Current Version

Table of Contents

Notice

This content has been produced from an import of the SCAP datastream of the

Guide to the Secure Configuration of Red Hat Enterprise Linux 8

as contained in release 0.144 of the ComplianceAsCode Github repository.

into the Scapolite format.

Please refer to the slide set about Scapolite that has been presented at NIST's SCAP v2 workshop for more information about Scapolite.

The copyright holder for the contents of the guide is RedHat, the license of the original SCAP content provided by `ComplianceAsCode` is as follows:

```
SPDX license identifier: BSD-3-Clause
Copyright (c) 2012-2017, Red Hat, Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:
    * Redistributions of source code must retain the above copyright
      notice, this list of conditions and the following disclaimer.
    * Redistributions in binary form must reproduce the above copyright
      notice, this list of conditions and the following disclaimer in the
      documentation and/or other materials provided with the distribution.
    * Neither the name of the <organization> nor the
      names of its contributors may be used to endorse or promote products
      derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

Please refer to ComplianceAsCode for information about the guide and access to the authoritative versions of the guide. This project merely uses the guide as an example of how SCAP content can be expressed and enriched in Scapolite.

For feedback/questions/discussion please use the mailing list

https://list.nist.gov/scap-dev-authoring

Do not attempt to implement any of the settings in this guide without first testing them in a non-operational environment. The creators of this guidance assume no responsibility whatsoever for its use by other parties, and makes no guarantees, expressed or implied, about its quality, reliability, or any other characteristic.

Introduction

Objectives

This guide presents a catalog of security-relevant configuration settings for Red Hat Enterprise Linux 8. It is a rendering of content structured in the eXtensible Configuration Checklist Description Format (XCCDF) in order to support security automation. The SCAP content is is available in the `scap-security-guide` package which is developed at https://www.open-scap.org/security-policies/scap-security-guide.

Providing system administrators with such guidance informs them how to securely configure systems under their control in a variety of network roles. Policy makers and baseline creators can use this catalog of settings, with its associated references to higher-level security control catalogs, in order to assist them in security baseline creation. This guide is a *catalog, not a checklist*, and satisfaction of every item is not likely to be possible or sensible in many operational scenarios. However, the XCCDF format enables granular selection and adjustment of settings, and their association with OVAL and OCIL content provides an automated checking capability. Transformations of this document, and its associated automated checking content, are capable of providing baselines that meet a diverse set of policy objectives. Some example XCCDF *Profiles*, which are selections of items that form checklists and can be used as baselines, are available with this guide. They can be processed, in an automated fashion, with tools that support the Security Content Automation Protocol (SCAP). The DISA STIG, which provides required settings for US

Department of Defense systems, is one example of a baseline created from this guidance.

Introduction

The purpose of this guidance is to provide security configuration recommendations and baselines for the Red Hat Enterprise Linux 8 operating system. Recommended settings for the basic operating system are provided, as well as for many network services that the system can provide to other systems. The guide is intended for system administrators. Readers are assumed to possess basic system administration skills for Unix-like systems, as well as some familiarity with the product's documentation and administration conventions. Some instructions within this guide are complex. All directions should be followed completely and with understanding of their effects in order to avoid serious adverse effects on the system and its security.

How to Use This Guide

Readers should heed the following points when using the guide.

Root Shell Environment Assumed

Most of the actions listed in this document are written with the assumption that they will be executed by the root user running the `/bin/bash` shell. Commands preceded with a hash mark (#) assume that the administrator will execute the commands as root, i.e. apply the command via `sudo` whenever possible, or use `su` to gain root privileges if `sudo` cannot be used. Commands which can be executed as a non-root user are are preceded by a dollar sign ($) prompt.

Test in Non-Production Environment

This guidance should always be tested in a non-production environment before deployment. This test environment should simulate the setup in which the system will be deployed as closely as possible.

Formatting Conventions

Commands intended for shell execution, as well as configuration file text, are featured in a `monospace font`. *Italics* are used to indicate instances where the system administrator must substitute the appropriate information into a command or configuration file.

Read Sections Completely and in Order

Each section may build on information and recommendations discussed in prior sections. Each section should be read and understood completely; instructions should never be blindly applied. Relevant discussion may occur after instructions for an action.

Reboot Required

A system reboot is implicitly required after some actions in order to complete the reconfiguration of the system. In many cases, the changes will not take effect until a reboot is performed. In order to ensure that changes are applied properly and to test functionality, always reboot the system after applying a set of recommendations from this guide.

General Principles

The following general principles motivate much of the advice in this guide and should also influence any configuration decisions that are not explicitly covered.

Configure Security Tools to Improve System Robustness

Several tools exist which can be effectively used to improve a system's resistance to and detection of unknown attacks. These tools can improve robustness against attack at the cost of relatively little configuration effort. In particular, this guide recommends and discusses the use of host-based firewalling, SELinux for protection against vulnerable services, and a logging and auditing infrastructure for detection of problems.

Encrypt Transmitted Data Whenever Possible

Data transmitted over a network, whether wired or wireless, is susceptible to passive monitoring. Whenever practical solutions for encrypting such data exist, they should be applied. Even if data is expected to be transmitted only over a local network, it should still be encrypted. Encrypting authentication data, such as passwords, is particularly important. Networks of Red Hat Enterprise Linux 8 machines can and should be configured so that no unencrypted authentication data is ever transmitted between machines.

Least Privilege

Grant the least privilege necessary for user accounts and software to perform tasks. For example, `sudo` can be implemented to limit authorization to super user accounts on the system only to designated personnel. Another example is to limit logins on server systems to only those administrators who need to log into them in order to perform administration tasks. Using SELinux also follows the principle of least privilege: SELinux policy can confine software to perform only actions on the system that are specifically allowed. This can be far more restrictive than the actions permissible by the traditional Unix permissions model.

Minimize Software to Minimize Vulnerability

The simplest way to avoid vulnerabilities in software is to avoid installing that software. On Red Hat Enterprise Linux 8,the RPM Package Manager (originally Red Hat Package Manager, abbreviated RPM) allows for careful management of the set of software packages installed on a system. Installed software contributes to system vulnerability in several ways. Packages that include setuid programs may provide local attackers a potential path to privilege escalation. Packages that include network services may give this opportunity to network-based attackers. Packages that include programs which are predictably executed by local users (e.g. after graphical login) may provide opportunities for trojan horses or other attack code to be run undetected. The number of software packages installed on a system can almost always be significantly pruned to include only the software for which there is an environmental or operational need.

Run Different Network Services on Separate Systems

Whenever possible, a server should be dedicated to serving exactly one network service. This limits the number of other services that can be compromised in the event that an attacker is able to successfully exploit a software flaw in one network service.

Remediation functions used by the SCAP Security Guide Project

XCCDF form of the various remediation functions as used by remediation scripts from the SCAP Security Guide Project.

System Settings

Contains rules that check correct system settings.

Configure Syslog

The syslog service has been the default Unix logging mechanism for many years. It has a number of downsides, including inconsistent log format, lack of authentication for received messages, and lack of authentication, encryption, or reliable transport for messages sent over a network. However, due to its long history, syslog is a de facto standard which is supported by almost all Unix applications.

In Red Hat Enterprise Linux 8, rsyslog has replaced ksyslogd as the syslog daemon of choice, and it includes some additional security features such as reliable, connection-oriented (i.e. TCP) transmission of logs, the option to log to database formats, and the encryption of log data en route to a central logging server. This section discusses how to configure rsyslog for best effect, and how to use tools provided with the system to maintain and monitor logs.

Rsyslog Logs Sent To Remote Host

If system logs are to be useful in detecting malicious activities, it is necessary to send logs to a remote server. An intruder who has compromised the root account on a system may delete the log entries which indicate that the system was attacked before they are seen by an administrator.

However, it is recommended that logs be stored on the local host in addition to being sent to the loghost, especially if `rsyslog` has been configured to use the UDP protocol to send messages over a network. UDP does not guarantee reliable delivery, and moderately busy sites will lose log messages occasionally, especially in periods of high traffic which may be the result of an attack. In addition, remote `rsyslog` messages are not authenticated in any way by default, so it is easy for an attacker to introduce spurious messages to the central log server. Also, some problems cause loss of network connectivity, which will prevent the sending of messages to the central server. For all of these reasons, it is better to store log messages both centrally and on each host, so that they can be correlated if necessary.

rsyslog_remote_loghost

Ensure Logs Sent To Remote Host

To configure rsyslog to send logs to a remote log server, open `/etc/rsyslog.conf` and read and understand the last section of the file, which describes the multiple directives necessary to activate remote logging. Along with these other directives, the system can be configured to forward its logs to a particular log server by adding or correcting one of the following lines, substituting `loghost.example.com` appropriately. The choice of protocol depends on the environment of the system; although TCP and RELP provide more reliable message delivery, they may not be supported in all environments. To use UDP for log message delivery:

```
*.* @loghost.example.com
```

To use TCP for log message delivery:

```
*.* @@loghost.example.com
```

To use RELP for log message delivery:

```
*.* :omrelp:loghost.example.com
```

There must be a resolvable DNS CNAME or Alias record set to "" for logs to be sent correctly to the centralized logging utility.

Configure rsyslogd to Accept Remote Messages If Acting as a Log Server

By default, `rsyslog` does not listen over the network for log messages. If needed, modules can be enabled to allow the rsyslog daemon to receive messages from other systems and for the system thus to act as a log server. If the system is not a log server, then lines concerning these modules should remain commented out.

rsyslog_nolisten

Ensure rsyslog Does Not Accept Remote Messages Unless Acting As Log Server

The `rsyslog` daemon should not accept remote messages unless the system acts as a log server. To ensure that it is not listening on the network, ensure the following lines are *not* found in `/etc/rsyslog.conf`:

```
$ModLoad imtcp
$InputTCPServerRun port
$ModLoad imudp
$UDPServerRun port
$ModLoad imrelp
$InputRELPServerRun port
```

package_syslogng_installed

Ensure syslog-ng is Installed

syslog-ng can be installed in replacement of rsyslog. The `syslog-ng-core` package can be installed with the following command:

```
$ sudo yum install syslog-ng-core
```

service_syslogng_enabled

Enable syslog-ng Service

The `syslog-ng` service (in replacement of rsyslog) provides syslog-style logging by default on Debian 8. The `syslog-ng` service can be enabled with the following command:

```
$ sudo systemctl enable syslog-ng.service
```

rsyslog_accept_remote_messages_tcp

Enable rsyslog to Accept Messages via TCP, if Acting As Log Server

The `rsyslog` daemon should not accept remote messages unless the system acts as a log server. If the system needs to act as a central log server, add the following lines to `/etc/rsyslog.conf` to enable reception of messages over TCP:

```
$ModLoad imtcp
$InputTCPServerRun 514
```

rsyslog_accept_remote_messages_udp

Enable rsyslog to Accept Messages via UDP, if Acting As Log Server

The `rsyslog` daemon should not accept remote messages unless the system acts as a log server. If the system needs to act as a central log server, add the following lines to `/etc/rsyslog.conf` to enable reception of messages over UDP:

```
$ModLoad imudp
$UDPServerRun 514
```

Ensure Proper Configuration of Log Files

The file `/etc/rsyslog.conf` controls where log message are written. These are controlled by lines called *rules*, which consist of a *selector* and an *action*. These rules are often customized depending on the role of the system, the requirements of the environment, and whatever may enable the administrator to most effectively make use of log data. The default rules in Red Hat Enterprise Linux 8 are:

```
*.info;mail.none;authpriv.none;cron.none                /var/log/messages
authpriv.*                                              /var/log/secure
mail.*                                                  -/var/log/maillog
cron.*                                                  /var/log/cron
*.emerg                                                 *
uucp,news.crit                                          /var/log/spooler
local7.*                                                /var/log/boot.log
```

See the man page `rsyslog.conf(5)` for more information. *Note that the `rsyslog` daemon can be configured to use a timestamp format that some log processing programs may not understand. If this occurs, edit the file `/etc/rsyslog.conf` and add or edit the following line:*

```
$ ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat
```

rsyslog_files_groupownership

Ensure Log Files Are Owned By Appropriate Group

The group-owner of all log files written by `rsyslog` should be `. These log files are determined by the second part of each Rule line in`/etc/rsyslog.conf`and typically all appear in`/var/log`. For each log file *LOGFILE* referenced in`/etc/rsyslog.conf`, run the following command to inspect the file's group owner:

```
$ ls -l LOGFILE
```

If the owner is not ``, run the following command to correct this:

```
$ sudo chgrp  LOGFILE
```

rsyslog_files_ownership

Ensure Log Files Are Owned By Appropriate User

The owner of all log files written by `rsyslog` should be `. These log files are determined by the second part of each Rule line in`/etc/rsyslog.conf`and typically all appear in`/var/log`. For each log file *LOGFILE* referenced in`/etc/rsyslog.conf`, run the following command to inspect the file's owner:

```
$ ls -l LOGFILE
```

If the owner is not ``, run the following command to correct this:

```
$ sudo chown  LOGFILE
```

rsyslog_files_permissions

Ensure System Log Files Have Correct Permissions

The file permissions for all log files written by `rsyslog` should be set to 600, or more restrictive. These log files are determined by the second part of each Rule line in `/etc/rsyslog.conf` and typically all appear in `/var/log`. For each log file *LOGFILE* referenced in `/etc/rsyslog.conf`, run the following command to inspect the file's permissions:

```
$ ls -l LOGFILE
```

If the permissions are not 600 or more restrictive, run the following command to correct this:

```
$ sudo chmod 0600 LOGFILE
```

"

rsyslog_cron_logging

Ensure cron Is Logging To Rsyslog

Cron logging must be implemented to spot intrusions or trace cron job status. If `cron` is not logging to `rsyslog`, it can be implemented by adding the following to the *RULES* section of `/etc/rsyslog.conf`:

```
cron.*                                                  /var/log/cron
```

Ensure All Logs are Rotated by logrotate

Edit the file `/etc/logrotate.d/syslog`. Find the first line, which should look like this (wrapped for clarity):

```
/var/log/messages /var/log/secure /var/log/maillog /var/log/spooler \
  /var/log/boot.log /var/log/cron {
```

Edit this line so that it contains a one-space-separated listing of each log file referenced in `/etc/rsyslog.conf`.

All logs in use on a system must be rotated regularly, or the log files will consume disk space over time, eventually interfering with system operation. The file `/etc/logrotate.d/syslog` is the configuration file used by the `logrotate` program to maintain all log files written by `syslog`. By default, it rotates logs weekly and stores four archival copies of each log. These settings can be modified by editing `/etc/logrotate.conf`, but the defaults are sufficient for purposes of this guide.

Note that `logrotate` is run nightly by the cron job `/etc/cron.daily/logrotate`. If particularly active logs need to be rotated more often than once a day, some other mechanism must be used.

ensure_logrotate_activated

Ensure Logrotate Runs Periodically

The `logrotate` utility allows for the automatic rotation of log files. The frequency of rotation is specified in `/etc/logrotate.conf`, which triggers a cron task. To configure logrotate to run daily, add or correct the following line in `/etc/logrotate.conf`:

```
# rotate log files frequency
daily
```

Configure Logwatch on the Central Log Server

Is this system the central log server? If so, edit the file `/etc/logwatch/conf/logwatch.conf` as shown below.

logwatch_configured_splithosts

Configure Logwatch SplitHosts Line

If `SplitHosts` is set, Logwatch will separate entries by hostname. This makes the report longer but significantly more usable. If it is not set, then Logwatch will not report which host generated a given log entry, and that information is almost always necessary

```
SplitHosts = yes
```

logwatch_configured_hostlimit

Configure Logwatch HostLimit Line

On a central logserver, you want Logwatch to summarize all syslog entries, including those which did not originate on the logserver itself. The `HostLimit` setting tells Logwatch to report on all hosts, not just the one on which it is running.

```
HostLimit = no
```

package_rsyslog_installed

Ensure rsyslog is Installed

Rsyslog is installed by default. The `rsyslog` package can be installed with the following command:

```
$ sudo yum install rsyslog
```

service_rsyslog_enabled

Enable rsyslog Service

The `rsyslog` service provides syslog-style logging by default on Red Hat Enterprise Linux 8. The `rsyslog` service can be enabled with the following command:

```
$ sudo systemctl enable rsyslog.service
```

disable_logwatch_for_logserver

Disable Logwatch on Clients if a Logserver Exists

Does your site have a central logserver which has been configured to report on logs received from all systems? If so:

```
$ sudo rm /etc/cron.daily/0logwatch
```

If no logserver exists, it will be necessary for each system to run Logwatch individually. Using a central logserver provides the security and reliability benefits discussed earlier, and also makes monitoring logs easier and less time-intensive for administrators.

System Accounting with auditd

The audit service provides substantial capabilities for recording system activities. By default, the service audits about SELinux AVC denials and certain types of security-relevant events such as system logins, account modifications, and authentication events performed by programs such as sudo. Under its default configuration, `auditd` has modest disk space requirements, and should not noticeably impact system performance.

NOTE: The Linux Audit daemon `auditd` can be configured to use the `augenrules` program to read audit rules files (`*.rules`) located in `/etc/audit/rules.d` location and compile them to create the resulting form of the `/etc/audit/audit.rules` configuration file during the daemon startup (default configuration). Alternatively, the `auditd` daemon can use the `auditctl` utility to read audit rules from the `/etc/audit/audit.rules` configuration file during daemon startup, and load them into the kernel. The expected behavior is configured via the appropriate `ExecStartPost` directive setting in the `/usr/lib/systemd/system/auditd.service` configuration file. To instruct the `auditd` daemon to use the `augenrules` program to read audit rules (default configuration), use the following setting:

```
ExecStartPost=-/sbin/augenrules --load
```

in the `/usr/lib/systemd/system/auditd.service` configuration file. In order to instruct the `auditd` daemon to use the `auditctl` utility to read audit rules, use the following setting:

```
ExecStartPost=-/sbin/auditctl -R /etc/audit/audit.rules
```

in the `/usr/lib/systemd/system/auditd.service` configuration file. Refer to `[Service]` section of the `/usr/lib/systemd/system/auditd.service` configuration file for further details.

Government networks often have substantial auditing requirements and `auditd` can be configured to meet these requirements. Examining some example audit records demonstrates how the Linux audit system satisfies common requirements. The following example from Fedora Documentation available at `https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/SELinux_Users_and_Administrators_Guide/sect-Security-Enhanced_Linux-Troubleshooting-Fixing_Problems.html#sect-Security-Enhanced_Linux-Fixing_Problems-Raw_Audit_Messages` shows the substantial amount of information captured in a two typical "raw" audit messages, followed by a breakdown of the most important fields. In this example the message is SELinux-related and reports an AVC denial (and the associated system call) that occurred when the Apache HTTP Server attempted to access the `/var/www/html/file1` file (labeled with the `samba_share_t` type):

```
type=AVC msg=audit(1226874073.147:96): avc:  denied  { getattr } for pid=2465 comm="httpd"
path="/var/www/html/file1" dev=dm-0 ino=284133 scontext=unconfined_u:system_r:httpd_t:s0
tcontext=unconfined_u:object_r:samba_share_t:s0 tclass=file

type=SYSCALL msg=audit(1226874073.147:96): arch=40000003 syscall=196 success=no exit=-13
a0=b98df198 a1=bfec85dc a2=54dff4 a3=2008171 items=0 ppid=2463 pid=2465 auid=502 uid=48
gid=48 euid=48 suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=6 comm="httpd"
exe="/usr/sbin/httpd" subj=unconfined_u:system_r:httpd_t:s0 key=(null)
```

- `msg=audit(1226874073.147:96)`
  - The number in parentheses is the unformatted time stamp (Epoch time) for the event, which can be converted to standard time by using the `date` command.
- `{ getattr }`
  - The item in braces indicates the permission that was denied. `getattr` indicates the source process was trying to read the target file's status information. This occurs before reading files. This action is denied due to the file being accessed having the wrong label. Commonly seen permissions include `getattr`, `read`, and `write`.
- `comm="httpd"`
  - The executable that launched the process. The full path of the executable is found in the `exe=` section of the system call (`SYSCALL`) message, which in this case, is `exe="/usr/sbin/httpd"`.
- `path="/var/www/html/file1"`
  - The path to the object (target) the process attempted to access.
- `scontext="unconfined_u:system_r:httpd_t:s0"`
  - The SELinux context of the process that attempted the denied action. In this case, it is the SELinux context of the Apache HTTP Server, which is running in the `httpd_t` domain.
- `tcontext="unconfined_u:object_r:samba_share_t:s0"`
  - The SELinux context of the object (target) the process attempted to access. In this case, it is the SELinux context of `file1`. Note: the `samba_share_t` type is not accessible to processes running in the `httpd_t` domain.

- From the system call (`SYSCALL`) message, two items are of interest:
  - `success=no`: indicates whether the denial (AVC) was enforced or not. `success=no` indicates the system call was not successful (SELinux denied access). `success=yes` indicates the system call was successful - this can be seen for permissive domains or unconfined domains, such as `initrc_t` and `kernel_t`.
  - `exe="/usr/sbin/httpd"`: the full path to the executable that launched the process, which in this case, is `exe="/usr/sbin/httpd"`.

Configure auditd Rules for Comprehensive Auditing

The `auditd` program can perform comprehensive monitoring of system activity. This section describes recommended configuration settings for comprehensive auditing, but a full description of the auditing system's capabilities is beyond the scope of this guide. The mailing list *linux-audit@redhat.com* exists to facilitate community discussion of the auditing system.

The audit subsystem supports extensive collection of events, including:

- Tracing of arbitrary system calls (identified by name or number) on entry or exit.
- Filtering by PID, UID, call success, system call argument (with some limitations), etc.
- Monitoring of specific files for modifications to the file's contents or metadata.

Auditing rules at startup are controlled by the file `/etc/audit/audit.rules`. Add rules to it to meet the auditing requirements for your organization. Each line in `/etc/audit/audit.rules` represents a series of arguments that can be passed to `auditctl` and can be individually tested during runtime. See documentation in `/usr/share/doc/audit-VERSION` and in the related man pages for more details.

If copying any example audit rulesets from `/usr/share/doc/audit-VERSION`, be sure to comment out the lines containing `arch=` which are not appropriate for your system's architecture. Then review and understand the following rules, ensuring rules are activated as needed for the appropriate architecture.

After reviewing all the rules, reading the following sections, and editing as needed, the new rules can be activated as follows:

```
$ sudo service auditd restart
```

Record Information on Kernel Modules Loading and Unloading

To capture kernel module loading and unloading events, use following lines, setting ARCH to either b32 for 32-bit system, or having two lines for both b32 and b64 in case your system is 64-bit:

```
-w /usr/sbin/insmod -p x -k modules
-w /usr/sbin/rmmod -p x -k modules
-w /usr/sbin/modprobe -p x -k modules
-a always,exit -F arch=ARCH -S init_module,delete_module -F key=modules
```

Place to add the lines depends on a way `auditd` daemon is configured. If it is configured to use the `augenrules` program (the default), add the lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility, add the lines to file `/etc/audit/audit.rules`.

audit_rules_kernel_module_loading_modprobe

Ensure auditd Collects Information on Kernel Module Loading and Unloading - modprobe

To capture invocation of modprobe, utility used to insert / remove modules from kernel, add the following line:

```
-w /usr/sbin/modprobe -p x -k modules
```

Place to add the line depends on a way `auditd` daemon is configured. If it is configured to use the `augenrules` program (the default), add the line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility, add the line to file `/etc/audit/audit.rules`.

audit_rules_kernel_module_loading_init

Ensure auditd Collects Information on Kernel Module Loading - init_module

To capture kernel module loading events, use following line, setting ARCH to either b32 for 32-bit system, or having two lines for both b32 and b64 in case your system is 64-bit:

```
-a always,exit -F arch=ARCH -S init_module -F key=modules
```

Place to add the line depends on a way `auditd` daemon is configured. If it is configured to use the `augenrules` program (the default), add the line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility, add the line to file `/etc/audit/audit.rules`.

audit_rules_kernel_module_loading_delete

Ensure auditd Collects Information on Kernel Module Unloading - delete_module

To capture kernel module unloading events, use following line, setting ARCH to either b32 for 32-bit system, or having two lines for both b32 and b64 in case your system is 64-bit:

```
-a always,exit -F arch=ARCH -S delete_module -F key=modules
```

Place to add the line depends on a way `auditd` daemon is configured. If it is configured to use the `augenrules` program (the default), add the line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility, add the line to file `/etc/audit/audit.rules`.

audit_rules_kernel_module_loading

Ensure auditd Collects Information on Kernel Module Loading and Unloading

To capture kernel module loading and unloading events, use following lines, setting ARCH to either b32 for 32-bit system, or having two lines for both b32 and b64 in case your system is 64-bit:

```
-w /usr/sbin/insmod -p x -k modules
-w /usr/sbin/rmmod -p x -k modules
-w /usr/sbin/modprobe -p x -k modules

-a always,exit -F arch=ARCH -S init_module,finit_module,create_module,delete_module -F key=modules
```

The place to add the lines depends on a way `auditd` daemon is configured. If it is configured to use the `augenrules` program (the default), add the lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility, add the lines to file `/etc/audit/audit.rules`.

audit_rules_kernel_module_loading_insmod

Ensure auditd Collects Information on Kernel Module Loading - insmod

To capture invocation of insmod, utility used to insert modules into kernel, use the following line:

```
-w /usr/sbin/insmod -p x -k modules
```

Place to add the line depends on a way `auditd` daemon is configured. If it is configured to use the `augenrules` program (the default), add the line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility, add the line to file `/etc/audit/audit.rules`.

audit_rules_kernel_module_loading_rmmod

Ensure auditd Collects Information on Kernel Module Unloading - rmmod

To capture invocation of rmmod, utility used to remove modules from kernel, add the following line:

```
-w /usr/sbin/rmmod -p x -k modules
```

Place to add the line depends on a way `auditd` daemon is configured. If it is configured to use the `augenrules` program (the default), add the line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility, add the line to file `/etc/audit/audit.rules`.

audit_rules_kernel_module_loading_finit

Ensure auditd Collects Information on Kernel Module Loading and Unloading - finit_module

If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d` to capture kernel module loading and unloading events, setting ARCH to either b32 or b64 as appropriate for your system:

```
-a always,exit -F arch=ARCH -S finit_module -F key=modules
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file in order to capture kernel module loading and unloading events, setting ARCH to either b32 or b64 as appropriate for your system:

```
-a always,exit -F arch=ARCH -S finit_module -F key=modules
```

audit_rules_kernel_module_loading_create

Ensure auditd Collects Information on Kernel Module Loading - create_module

To capture kernel module loading events, use following line, setting ARCH to either b32 for 32-bit system, or having two lines for both b32 and b64 in case your system is 64-bit:

```
-a always,exit -F arch=ARCH -S create_module -F key=modules
```

Place to add the line depends on a way `auditd` daemon is configured. If it is configured to use the `augenrules` program (the default), add the line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility, add the line to file `/etc/audit/audit.rules`.

Record Unauthorized Access Attempts Events to Files (unsuccessful)

At a minimum, the audit system should collect unauthorized file accesses for all users and root. Note that the "-F arch=b32" lines should be present even on a 64 bit system. These commands identify system calls for auditing. Even if the system is 64 bit it can still execute 32 bit system calls. Additionally, these rules can be configured in a number of ways while still achieving the desired effect. An example of this is that the "-S" calls could be split up and placed on separate lines, however, this is less efficient. Add the following to `/etc/audit/audit.rules`:

```
-a always,exit -F arch=b32 -S creat,open,openat,open_by_handle_at,truncate,ftruncate -F exit=-EACCES -F auid>=1000 -F auid!=unset -F
key=access
    -a always,exit -F arch=b32 -S creat,open,openat,open_by_handle_at,truncate,ftruncate -F exit=-EPERM -F auid>=1000 -F auid!=unset -
F key=access
```

If your system is 64 bit then these lines should be duplicated and the arch=b32 replaced with arch=b64 as follows:

```
-a always,exit -F arch=b64 -S creat,open,openat,open_by_handle_at,truncate,ftruncate -F exit=-EACCES -F auid>=1000 -F auid!=unset -F
key=access
    -a always,exit -F arch=b64 -S creat,open,openat,open_by_handle_at,truncate,ftruncate -F exit=-EPERM -F auid>=1000 -F auid!=unset -
F key=access
```

audit_rules_unsuccessful_file_modification_chmod

Record Unsuccessul Permission Changes to Files - chmod

The audit system should collect unsuccessful file permission change attempts for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file.

```
-a always,exit -F arch=b32 -S chmod -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
-a always,exit -F arch=b32 -S chmod -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S chmod -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
-a always,exit -F arch=b64 -S chmod -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
```

audit_rules_unsuccessful_file_modification_chown

Record Unsuccessul Ownership Changes to Files - chown

The audit system should collect unsuccessful file ownership change attempts for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file.

```
-a always,exit -F arch=b32 -S chown -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
-a always,exit -F arch=b32 -S chown -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S chown -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
-a always,exit -F arch=b64 -S chown -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
```

audit_rules_unsuccessful_file_modification_removexattr

Record Unsuccessul Permission Changes to Files - removexattr

The audit system should collect unsuccessful file permission change attempts for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file.

```
-a always,exit -F arch=b32 -S removexattr -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
-a always,exit -F arch=b32 -S removexattr -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S removexattr -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
-a always,exit -F arch=b64 -S removexattr -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
```

audit_rules_unsuccessful_file_modification_ftruncate

Record Unauthorized Access Attempts to Files (unsuccessful) - ftruncate

At a minimum, the audit system should collect unauthorized file accesses for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S ftruncate -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=access
-a always,exit -F arch=b32 -S ftruncate -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=access
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S ftruncate -F exiu=-EACCES -F auid>=1000 -F auid!=unset -F key=access
-a always,exit -F arch=b64 -S ftruncate -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=access
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S ftruncate -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=access
-a always,exit -F arch=b32 -S ftruncate -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=access
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S ftruncate -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=access
-a always,exit -F arch=b64 -S ftruncate -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=access
```

audit_rules_unsuccessful_file_modification_lchown

Record Unsuccessul Ownership Changes to Files - lchown

The audit system should collect unsuccessful file ownership change attempts for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file.

```
-a always,exit -F arch=b32 -S lchown -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
-a always,exit -F arch=b32 -S lchown -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S lchown -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
-a always,exit -F arch=b64 -S lchown -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
```

audit_rules_unsuccessful_file_modification_fremovexattr

Record Unsuccessul Permission Changes to Files - fremovexattr

The audit system should collect unsuccessful file permission change attempts for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file.

```
-a always,exit -F arch=b32 -S fremovexattr -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
-a always,exit -F arch=b32 -S fremovexattr -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S fremovexattr -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
-a always,exit -F arch=b64 -S fremovexattr -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
```

audit_rules_unsuccessful_file_modification_open_o_trunc_write

Record Unauthorized Modification Attempts to Files - open O_TRUNC

The audit system should collect detailed unauthorized file accesses for all users and root. The `open` syscall can be used to modify files if called for write operation of with O_TRUNC flag. The following auidt rules will asure that unsuccessful attempts to modify a file via `open` syscall are collected. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the rules below to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the rules below to `/etc/audit/audit.rules` file.

```
-a always,exit -F arch=b32 -S open -F a1&01003 -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-modification
-a always,exit -F arch=b32 -S open -F a1&01003 -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-modification
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S open -F a1&01003 -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-modification
-a always,exit -F arch=b64 -S open -F a1&01003 -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-modification
```

audit_rules_unsuccessful_file_modification_fchown

Record Unsuccessul Ownership Changes to Files - fchown

The audit system should collect unsuccessful file ownership change attempts for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file.

```
-a always,exit -F arch=b32 -S fchown -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
-a always,exit -F arch=b32 -S fchown -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S fchown -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
-a always,exit -F arch=b64 -S fchown -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
```

audit_rules_unsuccessful_file_modification_openat_o_trunc_write

Record Unauthorized Modification Attempts to Files - openat O_TRUNC

The audit system should collect detailed unauthorized file accesses for all users and root. The `openat` syscall can be used to modify files if called for write operation of with O_TRUNC flag. The following auidt rules will asure that unsuccessful attempts to modify a file via `openat` syscall are collected. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the rules below to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the rules below to `/etc/audit/audit.rules` file.

```
-a always,exit -F arch=b32 -S openat -F a2&01003 -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-modification
-a always,exit -F arch=b32 -S openat -F a2&01003 -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-modification
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S openat -F a2&01003 -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-modification
-a always,exit -F arch=b64 -S openat -F a2&01003 -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-modification
```

audit_rules_unsuccessful_file_modification_fsetxattr

Record Unsuccessul Permission Changes to Files - fsetxattr

The audit system should collect unsuccessful file permission change attempts for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file.

```
-a always,exit -F arch=b32 -S fsetxattr -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
-a always,exit -F arch=b32 -S fsetxattr -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S fsetxattr -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
-a always,exit -F arch=b64 -S fsetxattr -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
```

audit_rules_unsuccessful_file_modification_fchownat

Record Unsuccessul Ownership Changes to Files - fchownat

The audit system should collect unsuccessful file ownership change attempts for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file.

```
-a always,exit -F arch=b32 -S fchownat -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
-a always,exit -F arch=b32 -S fchownat -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S fchownat -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
-a always,exit -F arch=b64 -S fchownat -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
```

audit_rules_unsuccessful_file_modification_open_by_handle_at_o_creat

Record Unauthorized Creation Attempts to Files - open_by_handle_at O_CREAT

The audit system should collect unauthorized file accesses for all users and root. The `open_by_handle_at` syscall can be used to create new files when O_CREAT flag is specified. The following auidt rules will asure that unsuccessful attempts to create a file via `open_by_handle_at` syscall are collected. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the rules below to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the rules below to `/etc/audit/audit.rules` file.

```
-a always,exit -F arch=b32 -S open_by_handle_at -F a2&0100 -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-create
-a always,exit -F arch=b32 -S open_by_handle_at -F a2&0100 -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-create
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S open_by_handle_at -F a2&0100 -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-create
-a always,exit -F arch=b64 -S open_by_handle_at -F a2&0100 -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-create
```

audit_rules_unsuccessful_file_modification_creat

Record Unauthorized Access Attempts to Files (unsuccessful) - creat

At a minimum, the audit system should collect unauthorized file accesses for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S creat -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=access
-a always,exit -F arch=b32 -S creat -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=access
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S creat -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=access
-a always,exit -F arch=b64 -S creat -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=access
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S creat -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=access
-a always,exit -F arch=b32 -S creat -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=access
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S creat -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=access
-a always,exit -F arch=b64 -S creat -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=access
```

audit_rules_unsuccessful_file_modification_open_o_creat

Record Unauthorized Creation Attempts to Files - open O_CREAT

The audit system should collect unauthorized file accesses for all users and root. The `open` syscall can be used to create new files when O_CREAT flag is specified. The following auidt rules will asure that unsuccessful attempts to create a file via `open` syscall are collected. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the rules below to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the rules below to `/etc/audit/audit.rules` file.

```
-a always,exit -F arch=b32 -S open -F a1&0100 -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-create
-a always,exit -F arch=b32 -S open -F a1&0100 -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-create
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S open -F a1&0100 -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-create
-a always,exit -F arch=b64 -S open -F a1&0100 -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-create
```

audit_rules_unsuccessful_file_modification_setxattr

Record Unsuccessul Permission Changes to Files - setxattr

The audit system should collect unsuccessful file permission change attempts for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file.

```
-a always,exit -F arch=b32 -S setxattr -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
-a always,exit -F arch=b32 -S setxattr -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S setxattr -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
-a always,exit -F arch=b64 -S setxattr -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
```

audit_rules_unsuccessful_file_modification_openat_o_creat

Record Unauthorized Creation Attempts to Files - openat O_CREAT

The audit system should collect unauthorized file accesses for all users and root. The `openat` syscall can be used to create new files when O_CREAT flag is specified. The following auidt rules will asure that unsuccessful attempts to create a file via `openat` syscall are collected. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the rules below to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the rules below to `/etc/audit/audit.rules` file.

```
-a always,exit -F arch=b32 -S openat -F a2&0100 -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-create
-a always,exit -F arch=b32 -S openat -F a2&0100 -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-create
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S openat -F a2&0100 -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-create
-a always,exit -F arch=b64 -S openat -F a2&0100 -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-create
```

audit_rules_unsuccessful_file_modification_rename

Record Unsuccessul Delete Attempts to Files - rename

The audit system should collect unsuccessful file deletion attempts for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file.

```
-a always,exit -F arch=b32 -S rename -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccessful-delete
-a always,exit -F arch=b32 -S rename -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccessful-delete
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S rename -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccessful-delete
-a always,exit -F arch=b64 -S rename -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccessful-delete
```

audit_rules_unsuccessful_file_modification_truncate

Record Unauthorized Access Attempts to Files (unsuccessful) - truncate

At a minimum, the audit system should collect unauthorized file accesses for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S truncate -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=access
-a always,exit -F arch=b32 -S truncate -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=access
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S truncate -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=access
-a always,exit -F arch=b64 -S truncate -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=access
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S truncate -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=access
-a always,exit -F arch=b32 -S truncate -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=access
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S truncate -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=access
-a always,exit -F arch=b64 -S truncate -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=access
```

audit_rules_unsuccessful_file_modification_fchmodat

Record Unsuccessul Permission Changes to Files - fchmodat

The audit system should collect unsuccessful file permission change attempts for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file.

```
-a always,exit -F arch=b32 -S fchmodat -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
-a always,exit -F arch=b32 -S fchmodat -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S fchmodat -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
-a always,exit -F arch=b64 -S fchmodat -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
```

audit_rules_unsuccessful_file_modification

Ensure auditd Collects Unauthorized Access Attempts to Files (unsuccessful)

At a minimum the audit system should collect unauthorized file accesses for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S creat,open,openat,open_by_handle_at,truncate,ftruncate -F exit=-EACCES -F auid>=1000 -F auid!=unset
-F key=access
-a always,exit -F arch=b32 -S creat,open,openat,open_by_handle_at,truncate,ftruncate -F exit=-EPERM -F auid>=1000 -F auid!=unset
-F key=access
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S creat,open,openat,open_by_handle_at,truncate,ftruncate -F exit=-EACCES -F auid>=1000 -F auid!=unset
-F key=access
-a always,exit -F arch=b64 -S creat,open,openat,open_by_handle_at,truncate,ftruncate -F exit=-EPERM -F auid>=1000 -F auid!=unset
-F key=access
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S creat,open,openat,open_by_handle_at,truncate,ftruncate -F exit=-EACCES -F auid>=1000 -F auid!=unset
-F key=access
-a always,exit -F arch=b32 -S creat,open,openat,open_by_handle_at,truncate,ftruncate -F exit=-EPERM -F auid>=1000 -F auid!=unset
-F key=access
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S creat,open,openat,open_by_handle_at,truncate,ftruncate -F exit=-EACCES -F auid>=1000 -F auid!=unset
-F key=access
-a always,exit -F arch=b64 -S creat,open,openat,open_by_handle_at,truncate,ftruncate -F exit=-EPERM -F auid>=1000 -F auid!=unset
-F key=access
```

audit_rules_unsuccessful_file_modification_lsetxattr

Record Unsuccessul Permission Changes to Files - lsetxattr

The audit system should collect unsuccessful file permission change attempts for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file.

```
-a always,exit -F arch=b32 -S lsetxattr -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
-a always,exit -F arch=b32 -S lsetxattr -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S lsetxattr -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
-a always,exit -F arch=b64 -S lsetxattr -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
```

audit_rules_unsuccessful_file_modification_renameat

Record Unsuccessul Delete Attempts to Files - renameat

The audit system should collect unsuccessful file deletion attempts for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file.

```
-a always,exit -F arch=b32 -S renameat -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccessful-delete
-a always,exit -F arch=b32 -S renameat -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccessful-delete
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S renameat -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccessful-delete
-a always,exit -F arch=b64 -S renameat -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccessful-delete
```

audit_rules_unsuccessful_file_modification_open_by_handle_at_rule_order

Ensure auditd Unauthorized Access Attempts To open_by_handle_at Are Ordered Correctly

The audit system should collect detailed unauthorized file accesses for all users and root. To correctly identify unsuccessful creation, unsuccessful modification and unsuccessful access of files via `open_by_handle_at` syscall the audit rules collecting these events need to be in certain order. The more specific rules need to come before the less specific rules. The reason for that is that more specific rules cover a subset of events covered in the less specific rules, thus, they need to come before to not be overshadowed by less specific rules, which match a bigger set of events. Make sure that rules for unsuccessful calls of `open_by_handle_at` syscall are in the order shown below. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), check the order of rules below in a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, check the order of rules below in `/etc/audit/audit.rules` file.

```
-a always,exit -F arch=b32 -S open_by_handle_at -F a2&0100 -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-create
-a always,exit -F arch=b32 -S open_by_handle_at -F a2&0100 -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-create
-a always,exit -F arch=b32 -S open_by_handle_at -F a2&01003 -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-
modification
-a always,exit -F arch=b32 -S open_by_handle_at -F a2&01003 -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-
modification
-a always,exit -F arch=b32 -S open_by_handle_at -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-access
-a always,exit -F arch=b32 -S open_by_handle_at -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-access
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S open_by_handle_at -F a2&0100 -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-create
-a always,exit -F arch=b64 -S open_by_handle_at -F a2&0100 -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-create
-a always,exit -F arch=b64 -S open_by_handle_at -F a2&01003 -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-
modification
-a always,exit -F arch=b64 -S open_by_handle_at -F a2&01003 -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-
modification
-a always,exit -F arch=b64 -S open_by_handle_at -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-access
-a always,exit -F arch=b64 -S open_by_handle_at -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-access
```

audit_rules_unsuccessful_file_modification_unlinkat

Record Unsuccessul Delete Attempts to Files - unlinkat

The audit system should collect unsuccessful file deletion attempts for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file.

```
-a always,exit -F arch=b32 -S unlinkat -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccessful-delete
-a always,exit -F arch=b32 -S unlinkat -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccessful-delete
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S unlinkat -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccessful-delete
-a always,exit -F arch=b64 -S unlinkat -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccessful-delete
```

audit_rules_unsuccessful_file_modification_open

Record Unauthorized Access Attempts to Files (unsuccessful) - open

At a minimum, the audit system should collect unauthorized file accesses for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S open -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=access
-a always,exit -F arch=b32 -S open -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=access
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S open -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=access
-a always,exit -F arch=b64 -S open -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=access
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S open -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=access
-a always,exit -F arch=b32 -S open -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=access
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S open -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=access
-a always,exit -F arch=b64 -S open -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=access
```

audit_rules_unsuccessful_file_modification_openat

Record Unauthorized Access Attempts to Files (unsuccessful) - openat

At a minimum, the audit system should collect unauthorized file accesses for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S openat -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=access
-a always,exit -F arch=b32 -S openat -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=access
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S openat -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=access
-a always,exit -F arch=b64 -S openat -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=access
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S openat -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=access
-a always,exit -F arch=b32 -S openat -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=access
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S openat -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=access
-a always,exit -F arch=b64 -S openat -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=access
```

audit_rules_unsuccessful_file_modification_lremovexattr

Record Unsuccessul Permission Changes to Files - lremovexattr

The audit system should collect unsuccessful file permission change attempts for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file.

```
-a always,exit -F arch=b32 -S lremovexattr -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
-a always,exit -F arch=b32 -S lremovexattr -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S lremovexattr -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
-a always,exit -F arch=b64 -S lremovexattr -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
```

audit_rules_unsuccessful_file_modification_fchmod

Record Unsuccessul Permission Changes to Files - fchmod

The audit system should collect unsuccessful file permission change attempts for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file.

```
-a always,exit -F arch=b32 -S fchmod -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
-a always,exit -F arch=b32 -S fchmod -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S fchmod -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
-a always,exit -F arch=b64 -S fchmod -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-perm-change
```

audit_rules_unsuccessful_file_modification_unlink

Record Unsuccessul Delete Attempts to Files - unlink

The audit system should collect unsuccessful file deletion attempts for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file.

```
-a always,exit -F arch=b32 -S unlink -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccessful-delete
-a always,exit -F arch=b32 -S unlink -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccessful-delete
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S unlink -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccessful-delete
-a always,exit -F arch=b64 -S unlink -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccessful-delete
```

audit_rules_unsuccessful_file_modification_open_by_handle_at

Record Unauthorized Access Attempts to Files (unsuccessful) - open_by_handle_at

At a minimum, the audit system should collect unauthorized file accesses for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S open_by_handle_at -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=access
-a always,exit -F arch=b32 -S open_by_handle_at -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=access
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S open_by_handle_at -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=access
-a always,exit -F arch=b64 -S open_by_handle_at -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=access
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S open_by_handle_at,truncate,ftruncate -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=access
-a always,exit -F arch=b32 -S open_by_handle_at,truncate,ftruncate -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=access
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S open_by_handle_at,truncate,ftruncate -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=access
-a always,exit -F arch=b64 -S open_by_handle_at,truncate,ftruncate -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=access
```

audit_rules_unsuccessful_file_modification_openat_rule_order

Ensure auditd Rules For Unauthorized Attempts To openat Are Ordered Correctly

The audit system should collect detailed unauthorized file accesses for all users and root. To correctly identify unsuccessful creation, unsuccessful modification and unsuccessful access of files via `openat` syscall the audit rules collecting these events need to be in certain order. The more specific rules need to come before the less specific rules. The reason for that is that more specific rules cover a subset of events covered in the less specific rules, thus, they need to come before to not be overshadowed by less specific rules, which match a bigger set of events. Make sure that rules for unsuccessful calls of `openat` syscall are in the order shown below. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), check the order of rules below in a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, check the order of rules below in `/etc/audit/audit.rules` file.

```
-a always,exit -F arch=b32 -S openat -F a2&0100 -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-create
-a always,exit -F arch=b32 -S openat -F a2&0100 -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-create
-a always,exit -F arch=b32 -S openat -F a2&01003 -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-modification
-a always,exit -F arch=b32 -S openat -F a2&01003 -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-modification
-a always,exit -F arch=b32 -S openat -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-access
-a always,exit -F arch=b32 -S openat -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-access
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S openat -F a2&0100 -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-create
-a always,exit -F arch=b64 -S openat -F a2&0100 -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-create
-a always,exit -F arch=b64 -S openat -F a2&01003 -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-modification
-a always,exit -F arch=b64 -S openat -F a2&01003 -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-modification
-a always,exit -F arch=b64 -S openat -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-access
-a always,exit -F arch=b64 -S openat -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-access
```

audit_rules_unsuccessful_file_modification_open_by_handle_at_o_trunc_write

Record Unauthorized Modification Attempts to Files - open_by_handle_at O_TRUNC

The audit system should collect detailed unauthorized file accesses for all users and root. The `open_by_handle_at` syscall can be used to modify files if called for write operation of with O_TRUNC flag. The following auidt rules will asure that unsuccessful attempts to modify a file via `open_by_handle_at` syscall are collected. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the rules below to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the rules below to `/etc/audit/audit.rules` file.

```
-a always,exit -F arch=b32 -S open_by_handle_at -F a2&01003 -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-
modification
-a always,exit -F arch=b32 -S open_by_handle_at -F a2&01003 -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-
modification
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S open_by_handle_at -F a2&01003 -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-
modification
-a always,exit -F arch=b64 -S open_by_handle_at -F a2&01003 -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-
modification
```

audit_rules_unsuccessful_file_modification_open_rule_order

Ensure auditd Rules For Unauthorized Attempts To open Are Ordered Correctly

The audit system should collect detailed unauthorized file accesses for all users and root. To correctly identify unsuccessful creation, unsuccessful modification and unsuccessful access of files via `open` syscall the audit rules collecting these events need to be in certain order. The more specific rules need to come before the less specific rules. The reason for that is that more specific rules cover a subset of events covered in the less specific rules, thus, they need to come before to not be overshadowed by less specific rules, which match a bigger set of events. Make sure that rules for unsuccessful calls of `open` syscall are in the order shown below. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), check the order of rules below in a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, check the order of rules below in `/etc/audit/audit.rules` file.

```
-a always,exit -F arch=b32 -S open -F a1&0100 -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-create
-a always,exit -F arch=b32 -S open -F a1&0100 -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-create
-a always,exit -F arch=b32 -S open -F a1&01003 -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-modification
-a always,exit -F arch=b32 -S open -F a1&01003 -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-modification
-a always,exit -F arch=b32 -S open -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-access
-a always,exit -F arch=b32 -S open -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-access
```

If the system is 64 bit then also add the following lines:

```
-a always,exit -F arch=b64 -S open -F a1&0100 -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-create
-a always,exit -F arch=b64 -S open -F a1&0100 -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-create
-a always,exit -F arch=b64 -S open -F a1&01003 -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-modification
-a always,exit -F arch=b64 -S open -F a1&01003 -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-modification
-a always,exit -F arch=b64 -S open -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccesful-access
-a always,exit -F arch=b64 -S open -F exit=-EPERM -F auid>=1000 -F auid!=unset -F key=unsuccesful-access
```

Record Execution Attempts to Run SELinux Privileged Commands

At a minimum, the audit system should collect the execution of SELinux privileged commands for all users and root.

audit_rules_execution_semanage

Record Any Attempts to Run semanage

At a minimum, the audit system should collect any execution attempt of the `semanage` command for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F path=/usr/sbin/semanage -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged-priv_change
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file:

```
-a always,exit -F path=/usr/sbin/semanage -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged-priv_change
```

audit_rules_execution_seunshare

Record Any Attempts to Run seunshare

At a minimum, the audit system should collect any execution attempt of the `seunshare` command for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F path=/usr/sbin/seunshare -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged-priv_change
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file:

```
-a always,exit -F path=/usr/sbin/seunshare -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged-priv_change
```

audit_rules_execution_setfiles

Record Any Attempts to Run setfiles

At a minimum, the audit system should collect any execution attempt of the `setfiles` command for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F path=/usr/sbin/setfiles -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged-priv_change
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file:

```
-a always,exit -F path=/usr/sbin/setfiles -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged-priv_change
```

audit_rules_execution_restorecon

Record Any Attempts to Run restorecon

At a minimum, the audit system should collect any execution attempt of the `restorecon` command for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F path=/usr/sbin/restorecon -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged-priv_change
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file:

```
-a always,exit -F path=/usr/sbin/restorecon -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged-priv_change
```

audit_rules_execution_chcon

Record Any Attempts to Run chcon

At a minimum, the audit system should collect any execution attempt of the `chcon` command for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F path=/usr/bin/chcon -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged-priv_change
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file:

```
-a always,exit -F path=/usr/bin/chcon -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged-priv_change
```

audit_rules_execution_setsebool

Record Any Attempts to Run setsebool

At a minimum, the audit system should collect any execution attempt of the `setsebool` command for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F path=/usr/sbin/setsebool -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged-priv_change
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file:

```
-a always,exit -F path=/usr/sbin/setsebool -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged-priv_change
```

Record Attempts to Alter Logon and Logout Events

The audit system already collects login information for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d` in order to watch for attempted manual edits of files involved in storing logon events:

```
-w /var/log/tallylog -p wa -k logins
-w /var/run/faillock/ -p wa -k logins
-w /var/log/lastlog -p wa -k logins
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file in order to watch for unattempted manual edits of files involved in storing logon events:

```
-w /var/log/tallylog -p wa -k logins
-w /var/run/faillock/ -p wa -k logins
-w /var/log/lastlog -p wa -k logins
```

audit_rules_login_events_lastlog

Record Attempts to Alter Logon and Logout Events - lastlog

The audit system already collects login information for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d` in order to watch for attempted manual edits of files involved in storing logon events:

```
-w /var/log/lastlog -p wa -k logins
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file in order to watch for unattempted manual edits of files involved in storing logon events:

```
-w /var/log/lastlog -p wa -k logins
```

audit_rules_login_events_tallylog

Record Attempts to Alter Logon and Logout Events - tallylog

The audit system already collects login information for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d` in order to watch for attempted manual edits of files involved in storing logon events:

```
-w /var/log/tallylog -p wa -k logins
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file in order to watch for unattempted manual edits of files involved in storing logon events:

```
-w /var/log/tallylog -p wa -k logins
```

audit_rules_login_events

Record Attempts to Alter Logon and Logout Events

The audit system already collects login information for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d` in order to watch for attempted manual edits of files involved in storing logon events:

```
-w /var/log/tallylog -p wa -k logins
-w /var/run/faillock -p wa -k logins
-w /var/log/lastlog -p wa -k logins
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file in order to watch for unattempted manual edits of files involved in storing logon events:

```
-w /var/log/tallylog -p wa -k logins
-w /var/run/faillock -p wa -k logins
-w /var/log/lastlog -p wa -k logins
```

audit_rules_login_events_faillock

Record Attempts to Alter Logon and Logout Events - faillock

The audit system already collects login information for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d` in order to watch for attempted manual edits of files involved in storing logon events:

```
-w /var/run/faillock -p wa -k logins
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file in order to watch for unattempted manual edits of files involved in storing logon events:

```
-w /var/run/faillock -p wa -k logins
```

Records Events that Modify Date and Time Information

Arbitrary changes to the system time can be used to obfuscate nefarious activities in log files, as well as to confuse network services that are highly dependent upon an accurate system time. All changes to the system time should be audited.

audit_rules_time_stime

Record Attempts to Alter Time Through stime

If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following line to a file with suffix `.rules` in the directory `/etc/audit/rules.d` for both 32 bit and 64 bit systems:

```
-a always,exit -F arch=b32 -S stime -F key=audit_time_rules
```

Since the 64 bit version of the "stime" system call is not defined in the audit lookup table, the corresponding "-F arch=b64" form of this rule is not expected to be defined on 64 bit systems (the aforementioned "-F arch=b32" stime rule form itself is sufficient for both 32 bit and 64 bit systems). If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following line to `/etc/audit/audit.rules` file for both 32 bit and 64 bit systems:

```
-a always,exit -F arch=b32 -S stime -F key=audit_time_rules
```

Since the 64 bit version of the "stime" system call is not defined in the audit lookup table, the corresponding "-F arch=b64" form of this rule is not expected to be defined on 64 bit systems (the aforementioned "-F arch=b32" stime rule form itself is sufficient for both 32 bit and 64 bit systems). The -k option allows for the specification of a key in string form that can be used for better reporting capability through ausearch and aureport. Multiple system calls can be defined on the same line to save space if desired, but is not required. See an example of multiple combined system calls:

```
-a always,exit -F arch=b64 -S adjtimex,settimeofday -F key=audit_time_rules
```

audit_rules_time_clock_settime

Record Attempts to Alter Time Through clock_settime

If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S clock_settime -F a0=0x0 -F key=time-change
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S clock_settime -F a0=0x0 -F key=time-change
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following line to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S clock_settime -F a0=0x0 -F key=time-change
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S clock_settime -F a0=0x0 -F key=time-change
```

The -k option allows for the specification of a key in string form that can be used for better reporting capability through ausearch and aureport. Multiple system calls can be defined on the same line to save space if desired, but is not required. See an example of multiple combined syscalls:

```
-a always,exit -F arch=b64 -S adjtimex,settimeofday -F key=audit_time_rules
```

audit_rules_time_adjtimex

Record attempts to alter time through adjtimex

If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S adjtimex -F key=audit_time_rules
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S adjtimex -F key=audit_time_rules
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following line to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S adjtimex -F key=audit_time_rules
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S adjtimex -F key=audit_time_rules
```

The -k option allows for the specification of a key in string form that can be used for better reporting capability through ausearch and aureport. Multiple system calls can be defined on the same line to save space if desired, but is not required. See an example of multiple combined syscalls:

```
-a always,exit -F arch=b64 -S adjtimex,settimeofday -F key=audit_time_rules
```

audit_rules_time_settimeofday

Record attempts to alter time through settimeofday

If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S settimeofday -F key=audit_time_rules
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S settimeofday -F key=audit_time_rules
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following line to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S settimeofday -F key=audit_time_rules
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S settimeofday -F key=audit_time_rules
```

The -k option allows for the specification of a key in string form that can be used for better reporting capability through ausearch and aureport. Multiple system calls can be defined on the same line to save space if desired, but is not required. See an example of multiple combined syscalls:

```
-a always,exit -F arch=b64 -S adjtimex,settimeofday -F key=audit_time_rules
```

audit_rules_time_watch_localtime

Record Attempts to Alter the localtime File

If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-w /etc/localtime -p wa -k audit_time_rules
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following line to `/etc/audit/audit.rules` file:

```
-w /etc/localtime -p wa -k audit_time_rules
```

The -k option allows for the specification of a key in string form that can be used for better reporting capability through ausearch and aureport and should always be used.

Record Events that Modify the System's Discretionary Access Controls

At a minimum, the audit system should collect file permission changes for all users and root. Note that the "-F arch=b32" lines should be present even on a 64 bit system. These commands identify system calls for auditing. Even if the system is 64 bit it can still execute 32 bit system calls. Additionally, these rules can be configured in a number of ways while still achieving the desired effect. An example of this is that the "-S" calls could be split up and placed on separate lines, however, this is less efficient. Add the following to `/etc/audit/audit.rules`:

```
-a always,exit -F arch=b32 -S chmod,fchmod,fchmodat -F auid>=1000 -F auid!=unset -F key=perm_mod
    -a always,exit -F arch=b32 -S chown,fchown,fchownat,lchown -F auid>=1000 -F auid!=unset -F key=perm_mod
    -a always,exit -F arch=b32 -S setxattr,lsetxattr,fsetxattr,removexattr,lremovexattr,fremovexattr -F auid>=1000 -F auid!=unset -F
key=perm_mod
```

If your system is 64 bit then these lines should be duplicated and the arch=b32 replaced with arch=b64 as follows:

```
-a always,exit -F arch=b64 -S chmod,fchmod,fchmodat -F auid>=1000 -F auid!=unset -F key=perm_mod
    -a always,exit -F arch=b64 -S chown,fchown,fchownat,lchown -F auid>=1000 -F auid!=unset -F key=perm_mod
    -a always,exit -F arch=b64 -S setxattr,lsetxattr,fsetxattr,removexattr,lremovexattr,fremovexattr -F auid>=1000 -F auid!=unset -F
key=perm_mod
```

audit_rules_dac_modification_lsetxattr

Record Events that Modify the System's Discretionary Access Controls - lsetxattr

At a minimum, the audit system should collect file permission changes for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S lsetxattr -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S lsetxattr -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following line to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S lsetxattr -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S lsetxattr -F auid>=1000 -F auid!=unset -F key=perm_mod
```

audit_rules_dac_modification_fchmod

Record Events that Modify the System's Discretionary Access Controls - fchmod

At a minimum, the audit system should collect file permission changes for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S fchmod -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S fchmod -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following line to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S fchmod -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S fchmod -F auid>=1000 -F auid!=unset -F key=perm_mod
```

audit_rules_dac_modification_fchown

Record Events that Modify the System's Discretionary Access Controls - fchown

At a minimum, the audit system should collect file permission changes for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S fchown -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S fchown -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following line to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S fchown -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S fchown -F auid>=1000 -F auid!=unset -F key=perm_mod
```

audit_rules_dac_modification_fremovexattr

Record Events that Modify the System's Discretionary Access Controls - fremovexattr

At a minimum, the audit system should collect file permission changes for all users and root.

If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S fremovexattr -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S fremovexattr -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following line to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S fremovexattr -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S fremovexattr -F auid>=1000 -F auid!=unset -F key=perm_mod
```

audit_rules_dac_modification_fsetxattr

Record Events that Modify the System's Discretionary Access Controls - fsetxattr

At a minimum, the audit system should collect file permission changes for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S fsetxattr -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S fsetxattr -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following line to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S fsetxattr -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S fsetxattr -F auid>=1000 -F auid!=unset -F key=perm_mod
```

audit_rules_dac_modification_removexattr

Record Events that Modify the System's Discretionary Access Controls - removexattr

At a minimum, the audit system should collect file permission changes for all users and root.

If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S removexattr -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S removexattr -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following line to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S removexattr -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S removexattr -F auid>=1000 -F auid!=unset -F key=perm_mod
```

audit_rules_dac_modification_setxattr

Record Events that Modify the System's Discretionary Access Controls - setxattr

At a minimum, the audit system should collect file permission changes for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S setxattr -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S setxattr -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following line to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S setxattr -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S setxattr -F auid>=1000 -F auid!=unset -F key=perm_mod
```

audit_rules_dac_modification_lchown

Record Events that Modify the System's Discretionary Access Controls - lchown

At a minimum, the audit system should collect file permission changes for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S lchown -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S lchown -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following line to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S lchown -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S lchown -F auid>=1000 -F auid!=unset -F key=perm_mod
```

audit_rules_dac_modification_chown

Record Events that Modify the System's Discretionary Access Controls - chown

At a minimum, the audit system should collect file permission changes for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S chown -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S chown -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following line to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S chown -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S chown -F auid>=1000 -F auid!=unset -F key=perm_mod
```

audit_rules_dac_modification_chmod

Record Events that Modify the System's Discretionary Access Controls - chmod

At a minimum, the audit system should collect file permission changes for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S chmod -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S chmod -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following line to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S chmod -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S chmod -F auid>=1000 -F auid!=unset -F key=perm_mod
```

audit_rules_dac_modification_lremovexattr

Record Events that Modify the System's Discretionary Access Controls - lremovexattr

At a minimum, the audit system should collect file permission changes for all users and root.

If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S lremovexattr -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S lremovexattr -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following line to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S lremovexattr -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S lremovexattr -F auid>=1000 -F auid!=unset -F key=perm_mod
```

audit_rules_dac_modification_fchmodat

Record Events that Modify the System's Discretionary Access Controls - fchmodat

At a minimum, the audit system should collect file permission changes for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S fchmodat -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S fchmodat -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following line to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S fchmodat -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S fchmodat -F auid>=1000 -F auid!=unset -F key=perm_mod
```

audit_rules_dac_modification_fchownat

Record Events that Modify the System's Discretionary Access Controls - fchownat

At a minimum, the audit system should collect file permission changes for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S fchownat -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S fchownat -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following line to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S fchownat -F auid>=1000 -F auid!=unset -F key=perm_mod
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S fchownat -F auid>=1000 -F auid!=unset -F key=perm_mod
```

Record File Deletion Events by User

At a minimum, the audit system should collect file deletion events for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`, setting ARCH to either b32 or b64 as appropriate for your system:

```
-a always,exit -F arch=ARCH -S rmdir,unlink,unlinkat,rename,renameat -F auid>=1000 -F auid!=unset -F key=delete
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following line to `/etc/audit/audit.rules` file, setting ARCH to either b32 or b64 as appropriate for your system:

```
-a always,exit -F arch=ARCH -S rmdir,unlink,unlinkat,rename,renameat -F auid>=1000 -F auid!=unset -F key=delete
```

audit_rules_file_deletion_events

Ensure auditd Collects File Deletion Events by User

At a minimum the audit system should collect file deletion events for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`, setting ARCH to either b32 or b64 as appropriate for your system:

```
-a always,exit -F arch=ARCH -S rmdir,unlink,unlinkat,rename,renameat -F auid>=1000 -F auid!=unset -F key=delete
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following line to `/etc/audit/audit.rules` file, setting ARCH to either b32 or b64 as appropriate for your system:

```
-a always,exit -F arch=ARCH -S rmdir,unlink,unlinkat,rename -S renameat -F auid>=1000 -F auid!=unset -F key=delete
```

audit_rules_file_deletion_events_unlink

Ensure auditd Collects File Deletion Events by User - unlink

At a minimum, the audit system should collect file deletion events for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`, setting ARCH to either b32 or b64 as appropriate for your system:

```
-a always,exit -F arch=ARCH -S unlink -F auid>=1000 -F auid!=unset -F key=delete
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following line to `/etc/audit/audit.rules` file, setting ARCH to either b32 or b64 as appropriate for your system:

```
-a always,exit -F arch=ARCH -S unlink -F auid>=1000 -F auid!=unset -F key=delete
```

audit_rules_file_deletion_events_unlinkat

Ensure auditd Collects File Deletion Events by User - unlinkat

At a minimum, the audit system should collect file deletion events for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`, setting ARCH to either b32 or b64 as appropriate for your system:

```
-a always,exit -F arch=ARCH -S unlinkat -F auid>=1000 -F auid!=unset -F key=delete
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following line to `/etc/audit/audit.rules` file, setting ARCH to either b32 or b64 as appropriate for your system:

```
-a always,exit -F arch=ARCH -S unlinkat -F auid>=1000 -F auid!=unset -F key=delete
```

audit_rules_file_deletion_events_rename

Ensure auditd Collects File Deletion Events by User - rename

At a minimum, the audit system should collect file deletion events for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`, setting ARCH to either b32 or b64 as appropriate for your system:

```
-a always,exit -F arch=ARCH -S rename -F auid>=1000 -F auid!=unset -F key=delete
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following line to `/etc/audit/audit.rules` file, setting ARCH to either b32 or b64 as appropriate for your system:

```
-a always,exit -F arch=ARCH -S rename -F auid>=1000 -F auid!=unset -F key=delete
```

audit_rules_file_deletion_events_renameat

Ensure auditd Collects File Deletion Events by User - renameat

At a minimum, the audit system should collect file deletion events for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`, setting ARCH to either b32 or b64 as appropriate for your system:

```
-a always,exit -F arch=ARCH -S renameat -F auid>=1000 -F auid!=unset -F key=delete
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following line to `/etc/audit/audit.rules` file, setting ARCH to either b32 or b64 as appropriate for your system:

```
-a always,exit -F arch=ARCH -S renameat -F auid>=1000 -F auid!=unset -F key=delete
```

audit_rules_file_deletion_events_rmdir

Ensure auditd Collects File Deletion Events by User - rmdir

At a minimum, the audit system should collect file deletion events for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`, setting ARCH to either b32 or b64 as appropriate for your system:

```
-a always,exit -F arch=ARCH -S rmdir -F auid>=1000 -F auid!=unset -F key=delete
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following line to `/etc/audit/audit.rules` file, setting ARCH to either b32 or b64 as appropriate for your system:

```
-a always,exit -F arch=ARCH -S rmdir -F auid>=1000 -F auid!=unset -F key=delete
```

Record Information on the Use of Privileged Commands

At a minimum, the audit system should collect the execution of privileged commands for all users and root.

audit_rules_privileged_commands_umount

Ensure auditd Collects Information on the Use of Privileged Commands - umount

At a minimum, the audit system should collect the execution of privileged commands for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add a line of the following form to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F path=/usr/bin/umount -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add a line of the following form to `/etc/audit/audit.rules`:

```
-a always,exit -F path=/usr/bin/umount -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

audit_rules_privileged_commands_gpasswd

Ensure auditd Collects Information on the Use of Privileged Commands - gpasswd

At a minimum, the audit system should collect the execution of privileged commands for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add a line of the following form to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F path=/usr/bin/gpasswd -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add a line of the following form to `/etc/audit/audit.rules`:

```
-a always,exit -F path=/usr/bin/gpasswd -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

audit_rules_privileged_commands_newgidmap

Ensure auditd Collects Information on the Use of Privileged Commands - newgidmap

At a minimum, the audit system should collect the execution of privileged commands for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add a line of the following form to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F path=/usr/bin/newgidmap -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add a line of the following form to `/etc/audit/audit.rules`:

```
-a always,exit -F path=/usr/bin/newgidmap -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

audit_rules_privileged_commands_postdrop

Ensure auditd Collects Information on the Use of Privileged Commands - postdrop

At a minimum, the audit system should collect the execution of privileged commands for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add a line of the following form to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F path=/usr/sbin/postdrop -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add a line of the following form to `/etc/audit/audit.rules`:

```
-a always,exit -F path=/usr/sbin/postdrop -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

audit_rules_privileged_commands_unix_chkpwd

Ensure auditd Collects Information on the Use of Privileged Commands - unix_chkpwd

At a minimum, the audit system should collect the execution of privileged commands for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add a line of the following form to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F path=/usr/bin/unix_chkpwd -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add a line of the following form to `/etc/audit/audit.rules`:

```
-a always,exit -F path=/usr/bin/unix_chkpwd -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

audit_rules_privileged_commands_sudo

Ensure auditd Collects Information on the Use of Privileged Commands - sudo

At a minimum, the audit system should collect the execution of privileged commands for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add a line of the following form to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F path=/usr/bin/sudo -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add a line of the following form to `/etc/audit/audit.rules`:

```
-a always,exit -F path=/usr/bin/sudo -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

audit_rules_privileged_commands_pam_timestamp_check

Ensure auditd Collects Information on the Use of Privileged Commands - pam_timestamp_check

At a minimum, the audit system should collect the execution of privileged commands for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add a line of the following form to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F path=/usr/sbin/pam_timestamp_check -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add a line of the following form to `/etc/audit/audit.rules`:

```
-a always,exit -F path=/usr/sbin/pam_timestamp_check -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

audit_rules_privileged_commands_passwd

Ensure auditd Collects Information on the Use of Privileged Commands - passwd

At a minimum, the audit system should collect the execution of privileged commands for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add a line of the following form to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F path=/usr/bin/passwd -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add a line of the following form to `/etc/audit/audit.rules`:

```
-a always,exit -F path=/usr/bin/passwd -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

audit_rules_privileged_commands_pt_chown

Ensure auditd Collects Information on the Use of Privileged Commands - pt_chown

At a minimum, the audit system should collect the execution of privileged commands for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add a line of the following form to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F path=/usr/libexec/pt_chown -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add a line of the following form to `/etc/audit/audit.rules`:

```
-a always,exit -F path=/usr/libexec/pt_chown -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

audit_rules_privileged_commands_mount

Ensure auditd Collects Information on the Use of Privileged Commands - mount

At a minimum, the audit system should collect the execution of privileged commands for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add a line of the following form to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F path=/usr/bin/mount -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add a line of the following form to `/etc/audit/audit.rules`:

```
-a always,exit -F path=/usr/bin/mount -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

audit_rules_privileged_commands_newgrp

Ensure auditd Collects Information on the Use of Privileged Commands - newgrp

At a minimum, the audit system should collect the execution of privileged commands for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add a line of the following form to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F path=/usr/bin/newgrp -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add a line of the following form to `/etc/audit/audit.rules`:

```
-a always,exit -F path=/usr/bin/newgrp -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

audit_rules_privileged_commands

Ensure auditd Collects Information on the Use of Privileged Commands

At a minimum, the audit system should collect the execution of privileged commands for all users and root. To find the relevant setuid / setgid programs, run the following command for each local partition *PART*:

```
$ sudo find PART -xdev -type f -perm -4000 -o -type f -perm -2000 2>/dev/null
```

If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add a line of the following form to a file with suffix `.rules` in the directory `/etc/audit/rules.d` for each setuid / setgid program on the system, replacing the *SETUID_PROG_PATH* part with the full path of that setuid / setgid program in the list:

```
-a always,exit -F path=SETUID_PROG_PATH -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add a line of the following form to `/etc/audit/audit.rules` for each setuid / setgid program on the system, replacing the *SETUID_PROG_PATH* part with the full path of that setuid / setgid program in the list:

```
-a always,exit -F path=SETUID_PROG_PATH -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

audit_rules_privileged_commands_newuidmap

Ensure auditd Collects Information on the Use of Privileged Commands - newuidmap

At a minimum, the audit system should collect the execution of privileged commands for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add a line of the following form to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F path=/usr/bin/newuidmap -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add a line of the following form to `/etc/audit/audit.rules`:

```
-a always,exit -F path=/usr/bin/newuidmap -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

audit_rules_privileged_commands_postqueue

Ensure auditd Collects Information on the Use of Privileged Commands - postqueue

At a minimum, the audit system should collect the execution of privileged commands for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add a line of the following form to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F path=/usr/sbin/postqueue -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add a line of the following form to `/etc/audit/audit.rules`:

```
-a always,exit -F path=/usr/sbin/postqueue -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

audit_rules_privileged_commands_su

Ensure auditd Collects Information on the Use of Privileged Commands - su

At a minimum, the audit system should collect the execution of privileged commands for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add a line of the following form to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F path=/usr/bin/su -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add a line of the following form to `/etc/audit/audit.rules`:

```
-a always,exit -F path=/usr/bin/su -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

audit_rules_privileged_commands_userhelper

Ensure auditd Collects Information on the Use of Privileged Commands - userhelper

At a minimum, the audit system should collect the execution of privileged commands for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add a line of the following form to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F path=/usr/bin/userhelper -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add a line of the following form to `/etc/audit/audit.rules`:

```
-a always,exit -F path=/usr/bin/userhelper -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

audit_rules_privileged_commands_chsh

Ensure auditd Collects Information on the Use of Privileged Commands - chsh

At a minimum, the audit system should collect the execution of privileged commands for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add a line of the following form to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F path=/usr/bin/chsh -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add a line of the following form to `/etc/audit/audit.rules`:

```
-a always,exit -F path=/usr/bin/chsh -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

audit_rules_privileged_commands_usernetctl

Ensure auditd Collects Information on the Use of Privileged Commands - usernetctl

At a minimum, the audit system should collect the execution of privileged commands for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add a line of the following form to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F path=/usr/sbin/usernetctl -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add a line of the following form to `/etc/audit/audit.rules`:

```
-a always,exit -F path=/usr/sbin/usernetctl -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

audit_rules_privileged_commands_sudoedit

Ensure auditd Collects Information on the Use of Privileged Commands - sudoedit

At a minimum, the audit system should collect the execution of privileged commands for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add a line of the following form to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F path=/usr/bin/sudoedit -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add a line of the following form to `/etc/audit/audit.rules`:

```
-a always,exit -F path=/usr/bin/sudoedit -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

audit_rules_privileged_commands_ssh_keysign

Ensure auditd Collects Information on the Use of Privileged Commands - ssh-keysign

At a minimum, the audit system should collect the execution of privileged commands for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add a line of the following form to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F path=/usr/libexec/openssh/ssh-keysign -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add a line of the following form to `/etc/audit/audit.rules`:

```
-a always,exit -F path=/usr/libexec/openssh/key-sign -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

audit_rules_privileged_commands_chage

Ensure auditd Collects Information on the Use of Privileged Commands - chage

At a minimum, the audit system should collect the execution of privileged commands for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add a line of the following form to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F path=/usr/bin/chage -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add a line of the following form to `/etc/audit/audit.rules`:

```
-a always,exit -F path=/usr/bin/chage -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

audit_rules_privileged_commands_at

Ensure auditd Collects Information on the Use of Privileged Commands - at

At a minimum, the audit system should collect the execution of privileged commands for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add a line of the following form to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F path=/usr/bin/at -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add a line of the following form to `/etc/audit/audit.rules`:

```
-a always,exit -F path=/usr/bin/at -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

audit_rules_privileged_commands_crontab

Ensure auditd Collects Information on the Use of Privileged Commands - crontab

At a minimum, the audit system should collect the execution of privileged commands for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add a line of the following form to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F path=/usr/bin/crontab -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add a line of the following form to `/etc/audit/audit.rules`:

```
-a always,exit -F path=/usr/bin/crontab -F perm=x -F auid>=1000 -F auid!=unset -F key=privileged
```

audit_rules_networkconfig_modification

Record Events that Modify the System's Network Environment

If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`, setting ARCH to either b32 or b64 as appropriate for your system:

```
-a always,exit -F arch=ARCH -S sethostname,setdomainname -F key=audit_rules_networkconfig_modification
-w /etc/issue -p wa -k audit_rules_networkconfig_modification
-w /etc/issue.net -p wa -k audit_rules_networkconfig_modification
-w /etc/hosts -p wa -k audit_rules_networkconfig_modification
-w /etc/sysconfig/network -p wa -k audit_rules_networkconfig_modification
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file, setting ARCH to either b32 or b64 as appropriate for your system:

```
-a always,exit -F arch=ARCH -S sethostname,setdomainname -F key=audit_rules_networkconfig_modification
-w /etc/issue -p wa -k audit_rules_networkconfig_modification
-w /etc/issue.net -p wa -k audit_rules_networkconfig_modification
-w /etc/hosts -p wa -k audit_rules_networkconfig_modification
-w /etc/sysconfig/network -p wa -k audit_rules_networkconfig_modification
```

file_ownership_var_log_audit

System Audit Logs Must Be Owned By Root

All audit logs must be owned by root user and group. By default, the path for audit log is

```
/var/log/audit/
```

. To properly set the owner of `/var/log/audit`, run the command:

```
$ sudo chown root /var/log/audit
```

To properly set the owner of `/var/log/audit/*`, run the command:

```
$ sudo chown root /var/log/audit/*
```

audit_rules_immutable

Make the auditd Configuration Immutable

If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following line to a file with suffix `.rules` in the directory `/etc/audit/rules.d` in order to make the auditd configuration immutable:

```
-e 2
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following line to `/etc/audit/audit.rules` file in order to make the auditd configuration immutable:

```
-e 2
```

With this setting, a reboot will be required to change any audit rules.

audit_rules_etc_gshadow_open_by_handle_at

Record Events that Modify User/Group Information via open_by_handle_at syscall - /etc/gshadow

The audit system should collect write events to /etc/gshadow file for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S open_by_handle_at -F a2&03 -F path=/etc/gshadow -F auid>=1000 -F auid!=unset -F key=user-modify
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S open_by_handle_at -F a2&03 -F path=/etc/gshadow -F auid>=1000 -F auid!=unset -F key=user-modify
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S open_by_handle_at -F a2&03 -F path=/etc/gshadow -F auid>=1000 -F auid!=unset -F key=user-modify
```

audit_rules_etc_shadow_openat

Record Events that Modify User/Group Information via openat syscall - /etc/shadow

The audit system should collect write events to /etc/shadow file for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S openat -F a2&03 -F path=/etc/shadow -F auid>=1000 -F auid!=unset -F key=user-modify
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S openat -F a2&03 -F path=/etc/shadow -F auid>=1000 -F auid!=unset -F key=user-modify
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S openat -F a2&03 -F path=/etc/shadow -F auid>=1000 -F auid!=unset -F key=user-modify
```

audit_rules_etc_gshadow_open

Record Events that Modify User/Group Information via open syscall - /etc/gshadow

The audit system should collect write events to /etc/gshadow file for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S open -F a1&03 -F path=/etc/gshadow -F auid>=1000 -F auid!=unset -F key=user-modify
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S open -F a1&03 -F path=/etc/gshadow -F auid>=1000 -F auid!=unset -F key=user-modify
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S open -F a1&03 -F path=/etc/gshadow -F auid>=1000 -F auid!=unset -F key=user-modify
```

audit_rules_etc_passwd_open

Record Events that Modify User/Group Information via open syscall - /etc/passwd

The audit system should collect write events to /etc/passwd file for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S open -F a1&03 -F path=/etc/passwd -F auid>=1000 -F auid!=unset -F key=modify
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S open -F a1&03 -F path=/etc/passwd -F auid>=1000 -F auid!=unset -F key=modify
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S open -F a1&03 -F path=/etc/passwd -F auid>=1000 -F auid!=unset -F key=modify
```

audit_rules_session_events

Record Attempts to Alter Process and Session Initiation Information

The audit system already collects process information for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d` in order to watch for attempted manual edits of files involved in storing such process information:

```
-w /var/run/utmp -p wa -k session
-w /var/log/btmp -p wa -k session
-w /var/log/wtmp -p wa -k session
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file in order to watch for attempted manual edits of files involved in storing such process information:

```
-w /var/run/utmp -p wa -k session
-w /var/log/btmp -p wa -k session
-w /var/log/wtmp -p wa -k session
```

directory_access_var_log_audit

Record Access Events to Audit Log directory

The audit system should collect access events to read audit log directory. The following audit rule will assure that access to audit log directory are collected.

```
-a always,exit -F dir=/var/log/audit/ -F perm=r -F auid>=1000 -F auid!=unset -F key=access-audit-trail
```

If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the rule to a file with suffix `.rules` in the directory `/etc/audit/rules.d`. If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the rule to `/etc/audit/audit.rules` file.

audit_rules_etc_group_open

Record Events that Modify User/Group Information via open syscall - /etc/group

The audit system should collect write events to /etc/group file for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S open -F a1&03 -F path=/etc/group -F auid>=1000 -F auid!=unset -F key=modify
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S open -F a1&03 -F path=/etc/group -F auid>=1000 -F auid!=unset -F key=modify
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S open -F a1&03 -F path=/etc/group -F auid>=1000 -F auid!=unset -F key=modify
```

audit_rules_etc_group_open_by_handle_at

Record Events that Modify User/Group Information via open_by_handle_at syscall - /etc/group

The audit system should collect write events to /etc/group file for all group and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S open_by_handle_at -F a2&03 -F path=/etc/group -F auid>=1000 -F auid!=unset -F key=modify
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S open_by_handle_at -F a2&03 -F path=/etc/group -F auid>=1000 -F auid!=unset -F key=modify
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S open_by_handle_at -F a2&03 -F path=/etc/group -F auid>=1000 -F auid!=unset -F key=modify
```

audit_rules_etc_shadow_open

Record Events that Modify User/Group Information via open syscall - /etc/shadow

The audit system should collect write events to /etc/shadow file for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S open -F a1&03 -F path=/etc/shadow -F auid>=1000 -F auid!=unset -F key=user-modify
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S open -F a1&03 -F path=/etc/shadow -F auid>=1000 -F auid!=unset -F key=user-modify
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S open -F a1&03 -F path=/etc/shadow -F auid>=1000 -F auid!=unset -F key=user-modify
```

audit_rules_usergroup_modification_gshadow

Record Events that Modify User/Group Information - /etc/gshadow

If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`, in order to capture events that modify account changes:

```
-w /etc/gshadow -p wa -k audit_rules_usergroup_modification
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file, in order to capture events that modify account changes:

```
-w /etc/gshadow -p wa -k audit_rules_usergroup_modification
```

audit_rules_etc_shadow_open_by_handle_at

Record Events that Modify User/Group Information via open_by_handle_at syscall - /etc/shadow

The audit system should collect write events to /etc/shadow file for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S open_by_handle_at -F a2&03 -F path=/etc/shadow -F auid>=1000 -F auid!=unset -F key=user-modify
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S open_by_handle_at -F a2&03 -F path=/etc/shadow -F auid>=1000 -F auid!=unset -F key=user-modify
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S open_by_handle_at -F a2&03 -F path=/etc/shadow -F auid>=1000 -F auid!=unset -F key=user-modify
```

audit_rules_usergroup_modification_passwd

Record Events that Modify User/Group Information - /etc/passwd

If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`, in order to capture events that modify account changes:

```
-w /etc/passwd -p wa -k audit_rules_usergroup_modification
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file, in order to capture events that modify account changes:

```
-w /etc/passwd -p wa -k audit_rules_usergroup_modification
```

file_permissions_var_log_audit

System Audit Logs Must Have Mode 0640 or Less Permissive

If `log_group` in `/etc/audit/auditd.conf` is set to a group other than the `root` group account, change the mode of the audit log files with the following command:

```
$ sudo chmod 0640 audit_file
```

Otherwise, change the mode of the audit log files with the following command:

```
$ sudo chmod 0600 audit_file
```

audit_rules_mac_modification

Record Events that Modify the System's Mandatory Access Controls

If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-w /etc/selinux/ -p wa -k MAC-policy
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following line to `/etc/audit/audit.rules` file:

```
-w /etc/selinux/ -p wa -k MAC-policy
```

audit_rules_media_export

Ensure auditd Collects Information on Exporting to Media (successful)

At a minimum, the audit system should collect media exportation events for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`, setting ARCH to either b32 or b64 as appropriate for your system:

```
-a always,exit -F arch=ARCH -S mount -F auid>=1000 -F auid!=unset -F key=export
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following line to `/etc/audit/audit.rules` file, setting ARCH to either b32 or b64 as appropriate for your system:

```
-a always,exit -F arch=ARCH -S mount -F auid>=1000 -F auid!=unset -F key=export
```

audit_rules_system_shutdown

Shutdown System When Auditing Failures Occur

If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-f 2
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following line to the top of the `/etc/audit/audit.rules` file:

```
-f 2
```

audit_rules_etc_group_openat

Record Events that Modify User/Group Information via openat syscall - /etc/group

The audit system should collect write events to /etc/group file for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S openat -F a2&03 -F path=/etc/group -F auid>=1000 -F auid!=unset -F key=modify
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S openat -F a2&03 -F path=/etc/group -F auid>=1000 -F auid!=unset -F key=modify
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S openat -F a2&03 -F path=/etc/group -F auid>=1000 -F auid!=unset -F key=modify
```

audit_rules_etc_passwd_openat

Record Events that Modify User/Group Information via openat syscall - /etc/passwd

The audit system should collect write events to /etc/passwd file for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S openat -F a2&03 -F path=/etc/passwd -F auid>=1000 -F auid!=unset -F key=modify
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S openat -F a2&03 -F path=/etc/passwd -F auid>=1000 -F auid!=unset -F key=modify
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S openat -F a2&03 -F path=/etc/passwd -F auid>=1000 -F auid!=unset -F key=modify
```

directory_permissions_var_log_audit

System Audit Logs Must Have Mode 0750 or Less Permissive

If `log_group` in `/etc/audit/auditd.conf` is set to a group other than the `root` group account, change the mode of the audit log files with the following command:

```
$ sudo chmod 0750 /var/log/audit
```

Otherwise, change the mode of the audit log files with the following command:

```
$ sudo chmod 0700 /var/log/audit
```

audit_rules_usergroup_modification_group

Record Events that Modify User/Group Information - /etc/group

If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`, in order to capture events that modify account changes:

```
-w /etc/group -p wa -k audit_rules_usergroup_modification
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file, in order to capture events that modify account changes:

```
-w /etc/group -p wa -k audit_rules_usergroup_modification
```

audit_rules_usergroup_modification_opasswd

Record Events that Modify User/Group Information - /etc/security/opasswd

If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`, in order to capture events that modify account changes:

```
-w /etc/security/opasswd -p wa -k audit_rules_usergroup_modification
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file, in order to capture events that modify account changes:

```
-w /etc/security/opasswd -p wa -k audit_rules_usergroup_modification
```

audit_rules_etc_passwd_open_by_handle_at

Record Events that Modify User/Group Information via open_by_handle_at syscall - /etc/passwd

The audit system should collect write events to /etc/passwd file for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S open_by_handle_at -F a2&03 -F path=/etc/passwd -F auid>=1000 -F auid!=unset -F key=modify
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S open_by_handle_at -F a2&03 -F path=/etc/passwd -F auid>=1000 -F auid!=unset -F key=modify
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S open_by_handle_at -F a2&03 -F path=/etc/passwd -F auid>=1000 -F auid!=unset -F key=modify
```

audit_rules_usergroup_modification_shadow

Record Events that Modify User/Group Information - /etc/shadow

If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`, in order to capture events that modify account changes:

```
-w /etc/shadow -p wa -k audit_rules_usergroup_modification
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file, in order to capture events that modify account changes:

```
-w /etc/shadow -p wa -k audit_rules_usergroup_modification
```

audit_rules_sysadmin_actions

Ensure auditd Collects System Administrator Actions

At a minimum, the audit system should collect administrator actions for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following line to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-w /etc/sudoers -p wa -k actions
-w /etc/sudoers.d/ -p wa -k actions
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following line to `/etc/audit/audit.rules` file:

```
-w /etc/sudoers -p wa -k actions
-w /etc/sudoers.d/ -p wa -k actions
```

audit_rules_usergroup_modification

Record Events that Modify User/Group Information

If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`, in order to capture events that modify account changes:

```
-w /etc/group -p wa -k audit_rules_usergroup_modification
-w /etc/passwd -p wa -k audit_rules_usergroup_modification
-w /etc/gshadow -p wa -k audit_rules_usergroup_modification
-w /etc/shadow -p wa -k audit_rules_usergroup_modification
-w /etc/security/opasswd -p wa -k audit_rules_usergroup_modification
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file, in order to capture events that modify account changes:

```
-w /etc/group -p wa -k audit_rules_usergroup_modification
-w /etc/passwd -p wa -k audit_rules_usergroup_modification
-w /etc/gshadow -p wa -k audit_rules_usergroup_modification
-w /etc/shadow -p wa -k audit_rules_usergroup_modification
-w /etc/security/opasswd -p wa -k audit_rules_usergroup_modification
```

audit_rules_etc_gshadow_openat

Record Events that Modify User/Group Information via openat syscall - /etc/gshadow

The audit system should collect write events to /etc/gshadow file for all users and root. If the `auditd` daemon is configured to use the `augenrules` program to read audit rules during daemon startup (the default), add the following lines to a file with suffix `.rules` in the directory `/etc/audit/rules.d`:

```
-a always,exit -F arch=b32 -S openat -F a2&03 -F path=/etc/gshadow -F auid>=1000 -F auid!=unset -F key=user-modify
```

If the `auditd` daemon is configured to use the `auditctl` utility to read audit rules during daemon startup, add the following lines to `/etc/audit/audit.rules` file:

```
-a always,exit -F arch=b32 -S openat -F a2&03 -F path=/etc/gshadow -F auid>=1000 -F auid!=unset -F key=user-modify
```

If the system is 64 bit then also add the following line:

```
-a always,exit -F arch=b64 -S openat -F a2&03 -F path=/etc/gshadow -F auid>=1000 -F auid!=unset -F key=user-modify
```

Configure auditd Data Retention

The audit system writes data to `/var/log/audit/audit.log`. By default, `auditd` rotates 5 logs by size (6MB), retaining a maximum of 30MB of data in total, and refuses to write entries when the disk is too full. This minimizes the risk of audit data filling its partition and impacting other services. This also minimizes the risk of the audit daemon temporarily disabling the system if it cannot write audit log (which it can be configured to do). For a busy system or a system which is thoroughly auditing system activity, the default settings for data retention may be insufficient. The log file size needed will depend heavily on what types of events are being audited. First configure auditing to log all the events of interest. Then monitor the log size manually for awhile to determine what file size will allow you to keep the required data for the correct time period.

Using a dedicated partition for `/var/log/audit` prevents the `auditd` logs from disrupting system functionality if they fill, and, more importantly, prevents other activity in `/var` from filling the partition and stopping the audit trail. (The audit logs are size-limited and therefore unlikely to grow without bound unless configured to do so.) Some machines may have requirements that no actions occur which cannot be audited. If this is the case, then `auditd` can be configured to halt the machine if it runs out of space. **Note:** Since older logs are rotated, configuring `auditd` this way does not prevent older logs from being rotated away before they can be viewed. *If your system is configured to halt when logging cannot be performed, make sure this can never happen under normal circumstances! Ensure that `/var/log/audit` is on its own partition, and that this partition is larger than the maximum amount of data `auditd` will retain normally.*

auditd_data_retention_space_left

Configure auditd space_left on Low Disk Space

The `auditd` service can be configured to take an action when disk space is running low but prior to running out of space completely. Edit the file `/etc/audit/auditd.conf`. Add or modify the following line, substituting *SIZE_in_MB* appropriately:

```
space_left = SIZE_in_MB
```

Set this value to the appropriate size in Megabytes cause the system to notify the user of an issue.

auditd_audispd_syslog_plugin_activated

Configure auditd to use audispd's syslog plugin

To configure the `auditd` service to use the `syslog` plug-in of the `audispd` audit event multiplexor, set the `active` line in `/etc/audit/plugins.d/syslog.conf` to `yes`. Restart the `auditd` service:

```
$ sudo service auditd restart
```

auditd_data_disk_full_action

Configure auditd Disk Full Action when Disk Space Is Full

The `auditd` service can be configured to take an action when disk space is running low but prior to running out of space completely. Edit the file `/etc/audit/auditd.conf`. Add or modify the following line, substituting *ACTION* appropriately:

```
disk_full_action = ACTION
```

Set this value to `single` to cause the system to switch to single-user mode for corrective action. Acceptable values also include `syslog`, `exec`, `single`, and `halt`. For certain systems, the need for availability outweighs the need to log all actions, and a different setting should be determined. Details regarding all possible values for *ACTION* are described in the `auditd.conf` man page.

auditd_data_retention_max_log_file

Configure auditd Max Log File Size

Determine the amount of audit data (in megabytes) which should be retained in each log file. Edit the file `/etc/audit/auditd.conf`. Add or modify the following line, substituting the correct value of for *STOREMB*:

```
max_log_file = STOREMB
```

Set the value to `6` (MB) or higher for general-purpose systems. Larger values, of course, support retention of even more audit data.

auditd_data_retention_action_mail_acct

Configure auditd mail_acct Action on Low Disk Space

The `auditd` service can be configured to send email to a designated account in certain situations. Add or correct the following line in `/etc/audit/auditd.conf` to ensure that administrators are notified via email for those situations:

```
action_mail_acct =
```

auditd_data_retention_num_logs

Configure auditd Number of Logs Retained

Determine how many log files `auditd` should retain when it rotates logs. Edit the file `/etc/audit/auditd.conf`. Add or modify the following line, substituting *NUMLOGS* with the correct value of :

```
num_logs = NUMLOGS
```

Set the value to 5 for general-purpose systems. Note that values less than 2 result in no log rotation.

auditd_audispd_disk_full_action

Configure audispd's Plugin disk_full_action When Disk Is Full

Configure the action the operating system takes if the disk the audit records are written to becomes full. Edit the file `/etc/audisp/audisp-remote.conf`. Add or modify the following line, substituting *ACTION* appropriately:

```
disk_full_action = ACTION
```

Set this value to `single` to cause the system to switch to single user mode for corrective action. Acceptable values also include `syslog` and `halt`. For certain systems, the need for availability outweighs the need to log all actions, and a different setting should be determined.

auditd_audispd_configure_remote_server

Configure audispd Plugin To Send Logs To Remote Server

Configure the audispd plugin to off-load audit records onto a different system or media from the system being audited. Set the `remote_server` option in

```
/etc/audit/audisp-remote.conf
```

with an IP address or hostname of the system that the audispd plugin should send audit records to. For example replacing *REMOTE_SYSTEM* with an IP address or hostname:

```
remote_server = REMOTE_SYSTEM
```

auditd_audispd_encrypt_sent_records

Encrypt Audit Records Sent With audispd Plugin

Configure the operating system to encrypt the transfer of off-loaded audit records onto a different system or media from the system being audited. Set the `transport` option in

```
/etc/audit/audisp-remote.conf
```

to `KRB5`.

auditd_audispd_network_failure_action

Configure audispd's Plugin network_failure_action On Network Failure

Configure the action the operating system takes if there is an error sending audit records to a remote system. Edit the file `/etc/audisp/audisp-remote.conf`. Add or modify the following line, substituting *ACTION* appropriately:

```
network_failure_action = ACTION
```

Set this value to `single` to cause the system to switch to single user mode for corrective action. Acceptable values also include `syslog` and `halt`. For certain systems, the need for availability outweighs the need to log all actions, and a different setting should be determined.

auditd_data_disk_error_action

Configure auditd Disk Error Action on Disk Error

The `auditd` service can be configured to take an action when there is a disk error. Edit the file `/etc/audit/auditd.conf`. Add or modify the following line, substituting *ACTION* appropriately:

```
disk_error_action = ACTION
```

Set this value to `single` to cause the system to switch to single-user mode for corrective action. Acceptable values also include `syslog`, `exec`, `single`, and `halt`. For certain systems, the need for availability outweighs the need to log all actions, and a different setting should be determined. Details regarding all possible values for *ACTION* are described in the `auditd.conf` man page.

auditd_data_retention_max_log_file_action

Configure auditd max_log_file_action Upon Reaching Maximum Log Size

The default action to take when the logs reach their maximum size is to rotate the log files, discarding the oldest one. To configure the action taken by `auditd`, add or correct the line in `/etc/audit/auditd.conf`:

```
max_log_file_action = ACTION
```

Possible values for *ACTION* are described in the `auditd.conf` man page. These include:

- `syslog`
- `suspend`
- `rotate`
- `keep_logs`

Set the `ACTION` to `rotate` to ensure log rotation occurs. This is the default. The setting is case-insensitive.

auditd_data_retention_space_left_action

Configure auditd space_left Action on Low Disk Space

The `auditd` service can be configured to take an action when disk space *starts* to run low. Edit the file `/etc/audit/auditd.conf`. Modify the following line, substituting *ACTION* appropriately:

```
space_left_action = ACTION
```

Possible values for *ACTION* are described in the `auditd.conf` man page. These include:

- `syslog`
- `email`
- `exec`
- `suspend`
- `single`
- `halt`

Set this to `email` (instead of the default, which is `suspend`) as it is more likely to get prompt attention. Acceptable values also include `suspend`, `single`, and `halt`.

auditd_data_retention_admin_space_left_action

Configure auditd admin_space_left Action on Low Disk Space

The `auditd` service can be configured to take an action when disk space is running low but prior to running out of space completely. Edit the file `/etc/audit/auditd.conf`. Add or modify the following line, substituting *ACTION* appropriately:

```
admin_space_left_action = ACTION
```

Set this value to `single` to cause the system to switch to single user mode for corrective action. Acceptable values also include `suspend` and `halt`. For certain systems, the need for availability outweighs the need to log all actions, and a different setting should be determined. Details regarding all possible values for *ACTION* are described in the `auditd.conf` man page.

auditd_data_retention_flush

Configure auditd flush priority

The `auditd` service can be configured to synchronously write audit event data to disk. Add or correct the following line in `/etc/audit/auditd.conf` to ensure that audit event data is fully synchronized with the log files on the disk:

```
flush =
```

grub2_audit_argument

Enable Auditing for Processes Which Start Prior to the Audit Daemon

To ensure all processes can be audited, even those which start prior to the audit daemon, add the argument `audit=1` to the default GRUB 2 command line for the Linux operating system in `/etc/default/grub`, in the manner below:

```
GRUB_CMDLINE_LINUX="crashkernel=auto rd.lvm.lv=VolGroup/LogVol06 rd.lvm.lv=VolGroup/lv_swap rhgb quiet rd.shell=0 audit=1"
```

grub2_audit_backlog_limit_argument

Extend Audit Backlog Limit for the Audit Daemon

To improve the kernel capacity to queue all log events, even those which occurred prior to the audit daemon, add the argument `audit_backlog_limit=8192` to the default GRUB 2 command line for the Linux operating system in `/etc/default/grub`, in the manner below:

```
GRUB_CMDLINE_LINUX="crashkernel=auto rd.lvm.lv=VolGroup/LogVol06 rd.lvm.lv=VolGroup/lv_swap rhgb quiet rd.shell=0 audit=1
audit_backlog_limit=8192"
```

service_auditd_enabled

Enable auditd Service

The `auditd` service is an essential userspace component of the Linux Auditing System, as it is responsible for writing audit records to disk. The `auditd` service can be enabled with the following command:

```
$ sudo systemctl enable auditd.service
```

package_auditd_installed

install the auditd service

The auditd service should be installed.

Account and Access Control

In traditional Unix security, if an attacker gains shell access to a certain login account, they can perform any action or access any file to which that account has access. Therefore, making it more difficult for unauthorized people to gain shell access to accounts, particularly to privileged accounts, is a necessary part of securing a system. This section introduces mechanisms for restricting access to accounts under Red Hat Enterprise Linux 8.

Secure Session Configuration Files for Login Accounts

When a user logs into a Unix account, the system configures the user's session by reading a number of files. Many of these files are located in the user's home directory, and may have weak permissions as a result of user error or misconfiguration. If an attacker can modify or even read certain types of account configuration information, they can often gain full access to the affected user's account. Therefore, it is important to test and correct configuration file permissions for interactive accounts, particularly those of privileged users such as root or system administrators.

Ensure that No Dangerous Directories Exist in Root's Path

The active path of the root account can be obtained by starting a new root shell and running:

```
# echo $PATH
```

This will produce a colon-separated list of directories in the path.

Certain path elements could be considered dangerous, as they could lead to root executing unknown or untrusted programs, which could contain malicious code. Since root may sometimes work inside untrusted directories, the `.` character, which represents the current directory, should never be in the root path, nor should any directory which can be written to by an unprivileged or semi-privileged (system) user.

It is a good practice for administrators to always execute privileged commands by typing the full path to the command.

accounts_root_path_dirs_no_write

Ensure that Root's Path Does Not Include World or Group-Writable Directories

For each element in root's path, run:

```
# ls -ld DIR
```

and ensure that write permissions are disabled for group and other.

root_path_no_dot

Ensure that Root's Path Does Not Include Relative Paths or Null Directories

Ensure that none of the directories in root's path is equal to a single `.` character, or that it contains any instances that lead to relative path traversal, such as `..` or beginning a path without the slash (`/`) character. Also ensure that there are no "empty" elements in the path, such as in these examples:

```
PATH=:/bin
PATH=/bin:
PATH=/bin::/sbin
```

These empty elements have the same effect as a single `.` character.

Ensure that Users Have Sensible Umask Values

The umask setting controls the default permissions for the creation of new files. With a default `umask` setting of 077, files and directories created by users will not be readable by any other user on the system. Users who wish to make specific files group- or world-readable can accomplish this by using the chmod command. Additionally, users can make all their files readable to their group by default by setting a `umask` of 027 in their shell configuration files. If default per-user groups exist (that is, if every user has a default group whose name is the same as that user's username and whose only member is the user), then it may even be safe for users to select a `umask` of 007, making it very easy to intentionally share files with groups of which the user is a member.

accounts_umask_etc_profile

Ensure the Default Umask is Set Correctly in /etc/profile

To ensure the default umask controlled by `/etc/profile` is set properly, add or correct the `umask` setting in `/etc/profile` to read as follows:

```
umask
```

accounts_umask_interactive_users

Ensure the Default Umask is Set Correctly For Interactive Users

Remove the `UMASK` environment variable from all interactive users initialization files.

accounts_umask_etc_csh_cshrc

Ensure the Default C Shell Umask is Set Correctly

To ensure the default umask for users of the C shell is set properly, add or correct the `umask` setting in `/etc/csh.cshrc` to read as follows:

```
umask
```

accounts_umask_etc_login_defs

Ensure the Default Umask is Set Correctly in login.defs

To ensure the default umask controlled by `/etc/login.defs` is set properly, add or correct the `UMASK` setting in `/etc/login.defs` to read as follows:

```
UMASK
```

accounts_umask_etc_bashrc

Ensure the Default Bash Umask is Set Correctly

To ensure the default umask for users of the Bash shell is set properly, add or correct the `umask` setting in `/etc/bashrc` to read as follows:

```
umask
```

accounts_user_dot_user_ownership

User Initialization Files Must Be Owned By the Primary User

Set the owner of the user initialization files for interactive users to the primary owner with the following command:

```
$ sudo chown USER /home/USER/.*
```

file_ownership_home_directories

All Interactive User Home Directories Must Be Owned By The Primary User

Change the owner of interactive users home directories to that correct owner. To change the owner of a interactive users home directory, use the following command:

```
$ sudo chown USER /home/USER
```

file_permissions_home_directories

All Interactive User Home Directories Must Have mode 0750 Or Less Permissive

Change the mode of interactive users home directories to `0750`. To change the mode of interactive users home directory, use the following command:

```
$ sudo chmod 0750 /home/USER
```

accounts_have_homedir_login_defs

Ensure Home Directories are Created for New Users

All local interactive user accounts, upon creation, should be assigned a home directory.

Configure the operating system to assign home directories to all new local interactive users by setting the `CREATE_HOME` parameter in `/etc/login.defs` to `yes` as follows:

```
CREATE_HOME yes
```

accounts_logon_fail_delay

Ensure the Logon Failure Delay is Set Correctly in login.defs

To ensure the logon failure delay controlled by `/etc/login.defs` is set properly, add or correct the `FAIL_DELAY` setting in `/etc/login.defs` to read as follows:

```
FAIL_DELAY
```

accounts_user_interactive_home_directory_exists

All Interactive Users Home Directories Must Exist

Create home directories to all interactive users that currently do not have a home directory assigned. Use the following commands to create the user home directory assigned in `/etc/passwd`:

```
$ sudo mkdir /home/USER
```

accounts_users_home_files_groupownership

All User Files and Directories In The Home Directory Must Be Group-Owned By The Primary User

Change the group of a local interactive users files and directories to a group that the interactive user is a member of. To change the group owner of a local interactive users files and directories, use the following command:

```
$ sudo chgrp USER_GROUP /home/USER/FILE_DIR
```

accounts_user_dot_no_world_writable_programs

User Initialization Files Must Not Run World-Writable Programs

Set the mode on files being executed by the user initialization files with the following command:

```
$ sudo chmod 0755 FILE
```

accounts_tmout

Set Interactive Session Timeout

Setting the `TMOUT` option in `/etc/profile` ensures that all user sessions will terminate based on inactivity. The `TMOUT` setting in `/etc/profile` should read as follows:

```
TMOUT=
```

accounts_users_home_files_ownership

All User Files and Directories In The Home Directory Must Be Owned By The Primary User

Change the owner of a interactive users files and directories to that owner. To change the of a local interactive users files and directories, use the following command:

```
$ sudo chown -R USER /home/USER
```

file_groupownership_home_directories

All Interactive User Home Directories Must Be Group-Owned By The Primary User

Change the group owner of interactive users home directory to the group found in `/etc/passwd`. To change the group owner of interactive users home directory, use the following command:

```
$ sudo chgrp USER_GROUP /home/USER
```

accounts_user_dot_group_ownership

User Initialization Files Must Be Group-Owned By The Primary User

Change the group owner of interactive users files to the group found in

```
/etc/passwd
```

for the user. To change the group owner of a local interactive user home directory, use the following command:

```
$ sudo chgrp USER_GROUP /home/USER/.INIT_FILE
```

accounts_user_interactive_home_directory_defined

All Interactive Users Must Have A Home Directory Defined

Assign home directories to all interactive users that currently do not have a home directory assigned.

accounts_max_concurrent_login_sessions

Limit the Number of Concurrent Login Sessions Allowed Per User

Limiting the number of allowed users and sessions per user can limit risks related to Denial of Service attacks. This addresses concurrent sessions for a single account and does not address concurrent sessions by a single user via multiple accounts. To set the number of concurrent sessions per user add the following line in `/etc/security/limits.conf`:

```
* hard maxlogins
```

accounts_users_home_files_permissions

All User Files and Directories In The Home Directory Must Have Mode 0750 Or Less Permissive

Set the mode on files and directories in the local interactive user home directory with the following command:

```
$ sudo chmod 0750 /home/USER/FILE_DIR
```

file_permissions_home_dirs

Ensure that User Home Directories are not Group-Writable or World-Readable

For each human user of the system, view the permissions of the user's home directory:

```
# ls -ld /home/USER
```

Ensure that the directory is not group-writable and that it is not world-readable. If necessary, repair the permissions:

```
# chmod g-w /home/USER
# chmod o-rwx /home/USER
```

file_permission_user_init_files

Ensure All User Initialization Files Have Mode 0740 Or Less Permissive

Set the mode of the user initialization files to `0740` with the following command:

```
$ sudo chmod 0740 /home/USER/.INIT_FILE
```

accounts_user_home_paths_only

Ensure that Users Path Contains Only Local Directories

Ensure that all interactive user initialization files executable search path statements do not contain statements that will reference a working directory other than the users home directory.

Warning Banners for System Accesses

Each system should expose as little information about itself as possible.

System banners, which are typically displayed just before a login prompt, give out information about the service or the host's operating system. This might include the distribution name and the system kernel version, and the particular version of a network service. This information can assist intruders in gaining access to the system as it can reveal whether the system is running vulnerable software. Most network services can be configured to limit what information is displayed.

Many organizations implement security policies that require a system banner provide notice of the system's ownership, provide warning to unauthorized users, and remind authorized users of their consent to monitoring.

Implement a GUI Warning Banner

In the default graphical environment, users logging directly into the system are greeted with a login screen provided by the GNOME Display Manager (GDM). The warning banner should be displayed in this graphical environment for these users. The following sections describe how to configure the GDM login banner.

gconf_gdm_enable_warning_gui_banner

Enable GUI Warning Banner

To enable displaying a login warning banner in the GNOME Display Manager's login screen, run the following command:

```
$ sudo gconftool-2 --direct \
  --config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \
  --type bool \
  --set /apps/gdm/simple-greeter/banner_message_enable true
```

To display a banner, this setting must be enabled and then banner text must also be set.

dconf_gnome_login_banner_text

Set the GNOME3 Login Warning Banner Text

In the default graphical environment, configuring the login warning banner text in the GNOME Display Manager's login screen can be configured on the login screen by setting `banner-message-text` to `string 'APPROVED_BANNER'` where *APPROVED_BANNER* is the approved banner for your environment.

To enable, add or edit `banner-message-text` to `/etc/dconf/db/gdm.d/00-security-settings`. For example:

```
[org/gnome/login-screen]
banner-message-text='APPROVED_BANNER'
```

Once the setting has been added, add a lock to `/etc/dconf/db/gdm.d/locks/00-security-settings-lock` to prevent user modification. For example:

```
/org/gnome/login-screen/banner-message-text
```

After the settings have been set, run `dconf update`. When entering a warning banner that spans several lines, remember to begin and end the string with `'` and use `\n` for new lines.

dconf_gnome_banner_enabled

Enable GNOME3 Login Warning Banner

In the default graphical environment, displaying a login warning banner in the GNOME Display Manager's login screen can be enabled on the login screen by setting `banner-message-enable` to `true`.

To enable, add or edit `banner-message-enable` to `/etc/dconf/db/gdm.d/00-security-settings`. For example:

```
[org/gnome/login-screen]
banner-message-enable=true
```

Once the setting has been added, add a lock to `/etc/dconf/db/gdm.d/locks/00-security-settings-lock` to prevent user modification. For example:

```
/org/gnome/login-screen/banner-message-enable
```

After the settings have been set, run `dconf update`. The banner text must also be set.

gconf_gdm_set_login_banner_text

Set GUI Warning Banner Text

To set the text shown by the GNOME Display Manager in the login screen, run the following command:

```
$ sudo gconftool-2 --direct \
  --config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \
  --type string \
  --set /apps/gdm/simple-greeter/banner_message_text \
  "Text of the warning banner here"
```

When entering a warning banner that spans several lines, remember to begin and end the string with `"`. This command writes directly either to the `/etc/gconf/gconf.xml.mandatory/%gconf-tree.xml` if it exists or to the file `/etc/gconf/gconf.xml.mandatory/apps/gdm/simple-greeter/%gconf.xml`. Either of these files can later be edited directly if necessary.

banner_etc_issue

Modify the System Login Banner

To configure the system login banner edit `/etc/issue`. Replace the default text with a message compliant with the local site policy or a legal disclaimer. The DoD required text is either:

```
You are accessing a U.S. Government (USG) Information System (IS) that
is provided for USG-authorized use only. By using this IS (which includes
any device attached to this IS), you consent to the following conditions:

-The USG routinely intercepts and monitors communications on this IS
for purposes including, but not limited to, penetration testing, COMSEC
monitoring, network operations and defense, personnel misconduct (PM), law
enforcement (LE), and counterintelligence (CI) investigations.

-At any time, the USG may inspect and seize data stored on this IS.

-Communications using, or data stored on, this IS are not private,
are subject to routine monitoring, interception, and search, and may be
disclosed or used for any USG-authorized purpose.

-This IS includes security measures (e.g., authentication and access
controls) to protect USG interests -- not for your personal benefit or
privacy.

-Notwithstanding the above, using this IS does not constitute consent
to PM, LE or CI investigative searching or monitoring of the content of
privileged communications, or work product, related to personal
representation or services by attorneys, psychotherapists, or clergy, and
their assistants. Such communications and work product are private and
confidential. See User Agreement for details.
```

OR:

```
I've read & consent to terms in IS user agreem't.
```

Protect Accounts by Restricting Password-Based Login

Conventionally, Unix shell accounts are accessed by providing a username and password to a login program, which tests these values for correctness using the `/etc/passwd` and `/etc/shadow` files. Password-based login is vulnerable to guessing of weak passwords, and to sniffing and man-in-the-middle attacks against passwords entered over a network or at an insecure console. Therefore, mechanisms for accessing accounts by entering usernames and passwords should be restricted to those which are operationally necessary.

Set Account Expiration Parameters

Accounts can be configured to be automatically disabled after a certain time period, meaning that they will require administrator interaction to become usable again. Expiration of accounts after inactivity can be set for all accounts by default and also on a per-account basis, such as for accounts that are known to be temporary. To configure automatic expiration of an account following the expiration of its password (that is, after the password has expired and not been changed), run the following command, substituting `NUM_DAYS` and `USER` appropriately:

```
$ sudo chage -I NUM_DAYS USER
```

Accounts, such as temporary accounts, can also be configured to expire on an explicitly-set date with the `-E` option. The file `/etc/default/useradd` controls default settings for all newly-created accounts created with the system's normal command line utilities.

account_temp_expire_date

Assign Expiration Date to Temporary Accounts

Temporary accounts are established as part of normal account activation procedures when there is a need for short-term accounts. In the event temporary or emergency accounts are required, configure the system to terminate them after a documented time period. For every temporary and emergency account, run the following command to set an expiration date on it, substituting `USER` and `YYYY-MM-DD` appropriately:

```
$ sudo chage -E YYYY-MM-DD USER
```

`YYYY-MM-DD` indicates the documented expiration date for the account. For U.S. Government systems, the operating system must be configured to automatically terminate these types of accounts after a period of 72 hours.

account_unique_name

Ensure All Accounts on the System Have Unique Names

Change usernames, or delete accounts, so each has a unique name.

account_use_centralized_automated_auth

Use Centralized and Automated Authentication

Implement an automated system for managing user accounts that minimizes the risk of errors, either intentional or deliberate. This system should integrate with an existing enterprise user management system, such as one based on Identity Management tools such as Active Directory, Kerberos, Directory Server, etc.

account_disable_post_pw_expiration

Set Account Expiration Following Inactivity

To specify the number of days after a password expires (which signifies inactivity) until an account is permanently disabled, add or correct the following lines in `/etc/default/useradd`, substituting `NUM_DAYS` appropriately:

```
INACTIVE=
```

A value of 35 is recommended; however, this profile expects that the value is set to `. If a password is currently on the verge of expiration, then 35 days remain until the account is automatically disabled. However, if the password will not expire for another 60 days, then 95 days could elapse until the account would be automatically disabled. See theuseradd` man page for more information. Determining the inactivity timeout must be done with careful consideration of the length of a "normal" period of inactivity for users in the particular environment. Setting the timeout too low incurs support costs and also has the potential to impact availability of the system to legitimate users.

Restrict Root Logins

Direct root logins should be allowed only for emergency use. In normal situations, the administrator should access the system via a unique unprivileged account, and then use `su` or `sudo` to execute privileged commands. Discouraging administrators from accessing the root account directly ensures an audit trail in organizations with multiple administrators. Locking down the channels through which root can connect directly also reduces opportunities for password-guessing against the root account. The `login` program uses the file `/etc/securetty` to determine which interfaces should allow root logins. The virtual devices `/dev/console` and `/dev/tty*` represent the system consoles (accessible via the Ctrl-Alt-F1 through Ctrl-Alt-F6 keyboard sequences on a default installation). The default securetty file also contains `/dev/vc/*`. These are likely to be deprecated in most environments, but may be retained for compatibility. Root should also be prohibited from connecting via network protocols. Other sections of this document include guidance describing how to prevent root from logging in via SSH.

no_password_auth_for_systemaccounts

Ensure that System Accounts Are Locked

Some accounts are not associated with a human user of the system, and exist to perform some administrative function. An attacker should not be able to log into these accounts.

System accounts are those user accounts with a user ID less than UID_MIN, where value of the UID_MIN directive is set in `/etc/login.defs` configuration file. In the default configuration UID_MIN is set to 500, thus system accounts are those user accounts with a user ID less than 500. If any system account *SYSACCT* (other than root) has an unlocked password, disable it with the command:

```
$ sudo passwd -l SYSACCT
```

no_root_webbrowsing

Restrict Web Browser Use for Administrative Accounts

Enforce policy requiring administrative accounts use web browsers only for local service administration.

securetty_root_login_console_only

Restrict Virtual Console Root Logins

To restrict root logins through the (deprecated) virtual console devices, ensure lines of this form do not appear in `/etc/securetty`:

```
vc/1
vc/2
vc/3
vc/4
```

restrict_serial_port_logins

Restrict Serial Port Root Logins

To restrict root logins on serial ports, ensure lines of this form do not appear in `/etc/securetty`:

```
ttyS0
ttyS1
```

root_path_default

Root Path Must Be Vendor Default

Assuming root shell is bash, edit the following files:

```
~/.profile
```

```
~/.bashrc
```

Change any `PATH` variables to the vendor default for root and remove any empty `PATH` entries or references to relative paths.

accounts_no_uid_except_zero

Verify Only Root Has UID 0

If any account other than root has a UID of 0, this misconfiguration should be investigated and the accounts other than root should be removed or have their UID changed. If the account is associated with system commands or applications the UID should be changed to one greater than "0" but less than "1000." Otherwise assign a UID greater than "1000" that has not already been assigned.

no_shelllogin_for_systemaccounts

Ensure that System Accounts Do Not Run a Shell Upon Login

Some accounts are not associated with a human user of the system, and exist to perform some administrative function. Should an attacker be able to log into these accounts, they should not be granted access to a shell.

The login shell for each local account is stored in the last field of each line in `/etc/passwd`. System accounts are those user accounts with a user ID less than UID_MIN, where value of UID_MIN directive is set in /etc/login.defs configuration file. In the default configuration UID_MIN is set to 1000, thus system accounts are those user accounts with a user ID less than 1000. The user ID is stored in the third field. If any system account *SYSACCT* (other than root) has a login shell, disable it with the command:

```
$ sudo usermod -s /sbin/nologin SYSACCT
```

no_direct_root_logins

Direct root Logins Not Allowed

To further limit access to the `root` account, administrators can disable root logins at the console by editing the `/etc/securetty` file. This file lists all devices the root user is allowed to login to. If the file does not exist at all, the root user can login through any communication device on the system, whether via the console or via a raw network interface. This is dangerous as user can login to the system as root via Telnet, which sends the password in plain text over the network. By default, Red Hat Enterprise Linux 8's `/etc/securetty` file only allows the root user to login at the console physically attached to the system. To prevent root from logging in, remove the contents of this file. To prevent direct root logins, remove the contents of this file by typing the following command:

```
$ sudo echo > /etc/securetty
```

Verify Proper Storage and Existence of Password Hashes

By default, password hashes for local accounts are stored in the second field (colon-separated) in `/etc/shadow`. This file should be readable only by processes running with root credentials, preventing users from casually accessing others' password hashes and attempting to crack them. However, it remains possible to misconfigure the system and store password hashes in world-readable files such as `/etc/passwd`, or to even store passwords themselves in plaintext on the system. Using system-provided tools for password change/creation should allow administrators to avoid such misconfiguration.

no_empty_passwords

Prevent Login to Accounts With Empty Password

If an account is configured for password authentication but does not have an assigned password, it may be possible to log into the account without authentication. Remove any instances of the `nullok` option in `/etc/pam.d/system-auth` to prevent logins with empty passwords.

gid_passwd_group_same

All GIDs referenced in /etc/passwd must be defined in /etc/group

Add a group to the system for each GID referenced without a corresponding group.

accounts_password_all_shadowed

Verify All Account Password Hashes are Shadowed

If any password hashes are stored in `/etc/passwd` (in the second field, instead of an `x` or `*`), the cause of this misconfiguration should be investigated. The account should have its password reset and the hash should be properly stored, or the account should be deleted entirely.

no_netrc_files

Verify No netrc Files Exist

The `.netrc` files contain login information used to auto-login into FTP servers and reside in the user's home directory. These files may contain unencrypted passwords to remote FTP servers making them susceptible to access by unauthorized users and should not be used. Any `.netrc` files should be removed.

Set Password Expiration Parameters

The file `/etc/login.defs` controls several password-related settings. Programs such as `passwd`, `su`, and `login` consult `/etc/login.defs` to determine behavior with regard to password aging, expiration warnings, and length. See the man page `login.defs(5)` for more information.

Users should be forced to change their passwords, in order to decrease the utility of compromised passwords. However, the need to change passwords often should be balanced against the risk that users will reuse or write down passwords if forced to change them too often. Forcing password changes every 90-360 days, depending on the environment, is recommended. Set the appropriate value as `PASS_MAX_DAYS` and apply it to existing accounts with the `-M` flag.

The `PASS_MIN_DAYS` (`-m`) setting prevents password changes for 7 days after the first change, to discourage password cycling. If you use this setting, train users to contact an administrator for an emergency password change in case a new password becomes compromised. The `PASS_WARN_AGE` (`-W`) setting gives users 7 days of warnings at login time that their passwords are about to expire.

For example, for each existing human user *USER*, expiration parameters could be adjusted to a 180 day maximum password age, 7 day minimum password age, and 7 day warning period with the following command:

```
$ sudo chage -M 180 -m 7 -W 7 USER
```

accounts_maximum_age_login_defs

Set Password Maximum Age

To specify password maximum age for new accounts, edit the file `/etc/login.defs` and add or correct the following line:

```
PASS_MAX_DAYS
```

A value of 180 days is sufficient for many environments. The DoD requirement is 60. The profile requirement is ``.

accounts_password_minlen_login_defs

Set Password Minimum Length in login.defs

To specify password length requirements for new accounts, edit the file `/etc/login.defs` and add or correct the following line:

```
PASS_MIN_LEN
```

The DoD requirement is `15`. The FISMA requirement is `12`. The profile requirement is `. If a program consults/etc/login.defsand also another PAM module (such aspam_pwquality`) during a password change operation, then the most restrictive must be satisfied. See PAM section for more information about enforcing password quality requirements.

accounts_password_set_max_life_existing

Set Existing Passwords Maximum Age

Configure non-compliant accounts to enforce a 60-day maximum password lifetime restriction by running the following command:

```
$ sudo chage -M 60 USER
```

accounts_minimum_age_login_defs

Set Password Minimum Age

To specify password minimum age for new accounts, edit the file `/etc/login.defs` and add or correct the following line:

```
PASS_MIN_DAYS
```

A value of 1 day is considered sufficient for many environments. The DoD requirement is 1. The profile requirement is ``.

accounts_password_set_min_life_existing

Set Existing Passwords Minimum Age

Configure non-compliant accounts to enforce a 24 hours/1 day minimum password lifetime by running the following command:

```
$ sudo chage -m 1 USER
```

accounts_password_warn_age_login_defs

Set Password Warning Age

To specify how many days prior to password expiration that a warning will be issued to users, edit the file `/etc/login.defs` and add or correct the following line:

```
PASS_WARN_AGE
```

The DoD requirement is 7. The profile requirement is ``.

Protect Physical Console Access

It is impossible to fully protect a system from an attacker with physical access, so securing the space in which the system is located should be considered a necessary step. However, there are some steps which, if taken, make it more difficult for an attacker to quickly or undetectably modify a system from its console.

Configure Screen Locking

When a user must temporarily leave an account logged-in, screen locking should be employed to prevent passersby from abusing the account. User education and training is particularly important for screen locking to be effective, and policies can be implemented to reinforce this.

Automatic screen locking is only meant as a safeguard for those cases where a user forgot to lock the screen.

Configure Console Screen Locking

A console screen locking mechanism is a temporary action taken when a user stops work and moves away from the immediate physical vicinity of the information system but does not logout because of the temporary nature of the absence. Rather than relying on the user to manually lock their operation system session prior to vacating the vicinity, operating systems need to be able to identify when a user's session has idled and take action to initiate the session lock.

package_screen_installed

Install the screen Package

To enable console screen locking, install the `screen` package. The `screen` package can be installed with the following command:

```
$ sudo yum install screen
```

Instruct users to begin new terminal sessions with the following command:

```
$ screen
```

The console can now be locked with the following key combination:

```
ctrl+a x
```

configure_tmux_lock_command

Configure the tmux Lock Command

To enable console screen locking in `tmux` terminal multiplexer, the `vlock` command must be configured to be used as a locking mechanism. Add the following line to `/etc/tmux.conf`:

```
set -g lock-command vlock
```

. The console can now be locked with the following key combination:

```
ctrl+b :lock-session
```

package_tmux_installed

Install the tmux Package

To enable console screen locking, install the `tmux` package. The `tmux` package can be installed with the following command:

```
$ sudo yum install tmux
```

Instruct users to begin new terminal sessions with the following command:

```
$ tmux
```

The console can now be locked with the following key combination:

```
ctrl+b :lock-session
```

Hardware Tokens for Authentication

The use of hardware tokens such as smart cards for system login provides stronger, two-factor authentication than using a username and password. In Red Hat Enterprise Linux servers and workstations, hardware token login is not enabled by default and must be enabled in the system settings.

configure_opensc_nss_db

Configure NSS DB To Use opensc

The `opensc` module should be configured for use over the `Coolkey PKCS#11` module in the NSS database. To configure the NSS database ot use the `opensc` module, run the following command:

```
$ sudo pkcs11-switch opensc
```

smartcard_auth

Enable Smart Card Login

To enable smart card authentication, consult the documentation at:

- **https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/System-Level_Authentication_Guide/smartcards.html#authconfig-smartcards**

For guidance on enabling SSH to authenticate against a Common Access Card (CAC), consult documentation at:

- **https://access.redhat.com/solutions/82273**

service_pcscd_enabled

Enable the pcscd Service

The `pcscd` service can be enabled with the following command:

```
$ sudo systemctl enable pcscd.service
```

configure_opensc_card_drivers

Configure opensc Smart Card Drivers

The OpenSC smart card tool can auto-detect smart card drivers; however, setting the smart card drivers in use by your organization helps to prevent users from using unauthorized smart cards. The default smart card driver for this profile is `. To configure the OpenSC driver, edit the`/etc/opensc-ARCH.conf` (where *ARCH* is the architecture of your operating system) file. Look for a line similar to:

```
# card_drivers = old, internal;
```

and change it to:

```
card_drivers = ;
```

force_opensc_card_drivers

Force opensc To Use Defined Smart Card Driver

The OpenSC smart card tool can auto-detect smart card drivers; however by forcing the smart card driver in use by your organization, opensc will no longer autodetect or use other drivers unless specified. This helps to prevent users from using unauthorized smart cards. The default smart card driver for this profile is `. To force the OpenSC driver, edit the/etc/opensc-ARCH.conf` (where *ARCH* is the architecture of your operating system) file. Look for a line similar to:

```
# force_card_driver = customcos;
```

and change it to:

```
force_card_driver = ;
```

package_pcsc-lite_installed

Install the pcsc-lite package

The `pcsc-lite` package can be installed with the following command:

```
$ sudo yum install pcsc-lite
```

smartcard_configure_cert_checking

Configure Smart Card Certificate Status Checking

Configure the operating system to do certificate status checking for PKI authentication. Modify all of the `cert_policy` lines in `/etc/pam_pkcs11/pam_pkcs11.conf` to include `ocsp_on` like so:

```
cert_policy = ca, ocsp_on, signature;
```

install_smartcard_packages

Install Smart Card Packages For Multifactor Authentication

Configure the operating system to implement multifactor authentication by installing the required packages with the following command: The `esc pam_pkcs11` package can be installed with the following command:

```
$ sudo yum install esc pam_pkcs11
```

package_opensc_installed

Install the opensc Package For Multifactor Authentication

The `opensc` package can be installed with the following command:

```
$ sudo yum install opensc
```

require_singleuser_auth

Require Authentication for Single User Mode

Single-user mode is intended as a system recovery method, providing a single user root access to the system by providing a boot option at startup. By default, no authentication is performed if single-user mode is selected.

By default, single-user mode is protected by requiring a password and is set in `/usr/lib/systemd/system/rescue.service`.

grub2_disable_interactive_boot

Verify that Interactive Boot is Disabled

Red Hat Enterprise Linux 8 systems support an "interactive boot" option that can be used to prevent services from being started. On a Red Hat Enterprise Linux 8 system, interactive boot can be enabled by providing a `1`, `yes`, `true`, or `on` value to the `systemd.confirm_spawn` kernel argument in `/etc/default/grub`. Remove any instance of

```
systemd.confirm_spawn=(1|yes|true|on)
```

from the kernel arguments in that file to disable interactive boot. It is also required to change the runtime configuration, run:

```
/sbin/grubby --update-kernel=ALL --remove-args="systemd.confirm_spawn"
```

disable_ctrlaltdel_burstaction

Disable Ctrl-Alt-Del Burst Action

By default, `SystemD` will reboot the system if the `Ctrl-Alt-Del` key sequence is pressed Ctrl-Alt-Delete more than 7 times in 2 seconds.

To configure the system to ignore the `CtrlAltDelBurstAction` setting, add or modify the following to `/etc/systemd/system.conf`:

```
CtrlAltDelBurstAction=none
```

disable_ctrlaltdel_reboot

Disable Ctrl-Alt-Del Reboot Activation

By default, `SystemD` will reboot the system if the `Ctrl-Alt-Del` key sequence is pressed.

To configure the system to ignore the `Ctrl-Alt-Del` key sequence from the command line instead of rebooting the system, do either of the following:

```
ln -sf /dev/null /etc/systemd/system/ctrl-alt-del.target
```

or

```
systemctl mask ctrl-alt-del.target
```

Do not simply delete the `/usr/lib/systemd/system/ctrl-alt-del.service` file, as this file may be restored during future system updates.

service_debug-shell_disabled

Disable debug-shell SystemD Service

SystemD's `debug-shell` service is intended to diagnose SystemD related boot issues with various `systemctl` commands. Once enabled and following a system reboot, the root shell will be available on `tty9` which is access by pressing `CTRL-ALT-F9`. The `debug-shell` service should only be used for SystemD related issues and should otherwise be disabled.

By default, the `debug-shell` SystemD service is disabled. The `debug-shell` service can be disabled with the following command:

```
$ sudo systemctl disable debug-shell.service
```

Protect Accounts by Configuring PAM

PAM, or Pluggable Authentication Modules, is a system which implements modular authentication for Linux programs. PAM provides a flexible and configurable architecture for authentication, and it should be configured to minimize exposure to unnecessary risk. This section contains guidance on how to accomplish that.

PAM is implemented as a set of shared objects which are loaded and invoked whenever an application wishes to authenticate a user. Typically, the application must be running as root in order to take advantage of PAM, because PAM's modules often need to be able to access sensitive stores of account information, such as /etc/shadow. Traditional privileged network listeners (e.g. sshd) or SUID programs (e.g. sudo) already meet this requirement. An SUID root application, userhelper, is provided so that programs which are not SUID or privileged themselves can still take advantage of PAM.

PAM looks in the directory `/etc/pam.d` for application-specific configuration information. For instance, if the program login attempts to authenticate a user, then PAM's libraries follow the instructions in the file `/etc/pam.d/login` to determine what actions should be taken.

One very important file in `/etc/pam.d` is `/etc/pam.d/system-auth`. This file, which is included by many other PAM configuration files, defines 'default' system authentication measures. Modifying this file is a good way to make far-reaching authentication changes, for instance when implementing a centralized authentication service.

Set Lockouts for Failed Password Attempts

The `pam_faillock` PAM module provides the capability to lock out user accounts after a number of failed login attempts. Its documentation is available in `/usr/share/doc/pam-VERSION/txts/README.pam_faillock`.

accounts_passwords_pam_faillock_unlock_time

Set Lockout Time for Failed Password Attempts

To configure the system to lock out accounts after a number of incorrect login attempts and require an administrator to unlock the account using `pam_faillock.so`, modify the content of both `/etc/pam.d/system-auth` and `/etc/pam.d/password-auth` as follows:

- add the following line immediately `before` the `pam_unix.so` statement in the `AUTH` section:

```
auth required pam_faillock.so preauth silent deny= unlock_time= fail_interval=
```

- add the following line immediately `after` the `pam_unix.so` statement in the `AUTH` section:

```
auth [default=die] pam_faillock.so authfail deny= unlock_time= fail_interval=
```

- add the following line immediately `before` the `pam_unix.so` statement in the `ACCOUNT` section:

```
account required pam_faillock.so
```

accounts_password_pam_unix_remember

Limit Password Reuse

Do not allow users to reuse recent passwords. This can be accomplished by using the `remember` option for the `pam_unix` or `pam_pwhistory` PAM modules.

In the file `/etc/pam.d/system-auth`, append `remember=` to the line which refers to the `pam_unix.so` or `pam_pwhistory.so`module, as shown below:

- for the `pam_unix.so` case:

```
password sufficient pam_unix.so ...existing_options... remember=
```

- for the `pam_pwhistory.so` case:

```
password requisite pam_pwhistory.so ...existing_options... remember=
```

The DoD STIG requirement is 5 passwords.

accounts_passwords_pam_faillock_deny

Set Deny For Failed Password Attempts

To configure the system to lock out accounts after a number of incorrect login attempts using `pam_faillock.so`, modify the content of both `/etc/pam.d/system-auth` and `/etc/pam.d/password-auth` as follows:

- add the following line immediately `before` the `pam_unix.so` statement in the `AUTH` section:

```
auth required pam_faillock.so preauth silent deny= unlock_time= fail_interval=
```

- add the following line immediately `after` the `pam_unix.so` statement in the `AUTH` section:

```
auth [default=die] pam_faillock.so authfail deny= unlock_time= fail_interval=
```

- add the following line immediately `before` the `pam_unix.so` statement in the `ACCOUNT` section:

```
account required pam_faillock.so
```

accounts_passwords_pam_faillock_deny_root

Configure the root Account for Failed Password Attempts

To configure the system to lock out the `root` account after a number of incorrect login attempts using `pam_faillock.so`, modify the content of both `/etc/pam.d/system-auth` and `/etc/pam.d/password-auth` as follows:

- Modify the following line in the `AUTH` section to add `even_deny_root`:

```
auth required pam_faillock.so preauth silent even_deny_root deny= unlock_time= fail_interval=
```

- Modify the following line in the `AUTH` section to add `even_deny_root`:

```
auth [default=die] pam_faillock.so authfail even_deny_root deny= unlock_time= fail_interval=
```

accounts_passwords_pam_faillock_interval

Set Interval For Counting Failed Password Attempts

Utilizing `pam_faillock.so`, the `fail_interval` directive configures the system to lock out an account after a number of incorrect login attempts within a specified time period. Modify the content of both `/etc/pam.d/system-auth` and `/etc/pam.d/password-auth` as follows:

- Add the following line immediately `before` the `pam_unix.so` statement in the `AUTH` section:

```
auth required pam_faillock.so preauth silent deny= unlock_time= fail_interval=
```

- Add the following line immediately `after` the `pam_unix.so` statement in the `AUTH` section:

```
auth [default=die] pam_faillock.so authfail deny= unlock_time= fail_interval=
```

- Add the following line immediately `before` the `pam_unix.so` statement in the `ACCOUNT` section:

```
account required pam_faillock.so
```

Set Password Hashing Algorithm

The system's default algorithm for storing password hashes in `/etc/shadow` is SHA-512. This can be configured in several locations.

set_password_hashing_algorithm_logindefs

Set Password Hashing Algorithm in /etc/login.defs

In `/etc/login.defs`, add or correct the following line to ensure the system will use SHA-512 as the hashing algorithm:

```
ENCRYPT_METHOD SHA512
```

set_password_hashing_algorithm_systemauth

Set PAM's Password Hashing Algorithm

The PAM system service can be configured to only store encrypted representations of passwords. In `/etc/pam.d/system-auth`, the `password` section of the file controls which PAM modules execute during a password change. Set the `pam_unix.so` module in the `password` section to include the argument `sha512`, as shown below:

```
password    sufficient    pam_unix.so sha512 other arguments...
```

This will help ensure when local users change their passwords, hashes for the new passwords will be generated using the SHA-512 algorithm. This is the default.

set_password_hashing_algorithm_libuserconf

Set Password Hashing Algorithm in /etc/libuser.conf

In `/etc/libuser.conf`, add or correct the following line in its `[defaults]` section to ensure the system will use the SHA-512 algorithm for password hashing:

```
crypt_style = sha512
```

Set Password Quality Requirements

The default `pam_pwquality` PAM module provides strength checking for passwords. It performs a number of checks, such as making sure passwords are not similar to dictionary words, are of at least a certain length, are not the previous password reversed, and are not simply a change of case from the previous password. It can also require passwords to be in certain character classes. The `pam_pwquality` module is the preferred way of configuring password requirements.

The `pam_cracklib` PAM module can also provide strength checking for passwords as the `pam_pwquality` module. It performs a number of checks, such as making sure passwords are not similar to dictionary words, are of at least a certain length, are not the previous password reversed, and are not simply a change of case from the previous password. It can also require passwords to be in certain character classes.

The man pages `pam_pwquality(8)` and `pam_cracklib(8)` provide information on the capabilities and configuration of each.

Set Password Quality Requirements with pam_pwquality

The `pam_pwquality` PAM module can be configured to meet requirements for a variety of policies.

For example, to configure `pam_pwquality` to require at least one uppercase character, lowercase character, digit, and other (special) character, make sure that `pam_pwquality` exists in `/etc/pam.d/system-auth`:

```
password    requisite     pam_pwquality.so try_first_pass local_users_only retry=3 authtok_type=
```

If no such line exists, add one as the first line of the password section in `/etc/pam.d/system-auth`. Next, modify the settings in `/etc/security/pwquality.conf` to match the following:

```
difok = 4
minlen = 14
dcredit = -1
ucredit = -1
lcredit = -1
ocredit = -1
maxrepeat = 3
```

The arguments can be modified to ensure compliance with your organization's security policy. Discussion of each parameter follows.

accounts_password_pam_difok

Ensure PAM Enforces Password Requirements - Minimum Different Characters

The pam_pwquality module's `difok` parameter sets the number of characters in a password that must not be present in and old password during a password change.

Modify the `difok` setting in `/etc/security/pwquality.conf` to equal to require differing characters when changing passwords.

accounts_password_pam_ucredit

Ensure PAM Enforces Password Requirements - Minimum Uppercase Characters

The pam_pwquality module's `ucredit=` parameter controls requirements for usage of uppercase letters in a password. When set to a negative number, any password will be required to contain that many uppercase characters. When set to a positive number, pam_pwquality will grant +1 additional length credit for each uppercase character. Modify the `ucredit` setting in `/etc/security/pwquality.conf` to require the use of an uppercase character in passwords.

accounts_password_pam_minlen

Ensure PAM Enforces Password Requirements - Minimum Length

The pam_pwquality module's `minlen` parameter controls requirements for minimum characters required in a password. Add `minlen=` after pam_pwquality to set minimum password length requirements.

accounts_password_pam_retry

Ensure PAM Enforces Password Requirements - Authentication Retry Prompts Permitted Per-Session

To configure the number of retry prompts that are permitted per-session: Edit the `pam_pwquality.so` statement in `/etc/pam.d/system-auth` to show `retry=`, or a lower value if site policy is more restrictive. The DoD requirement is a maximum of 3 prompts per session.

accounts_password_pam_minclass

Ensure PAM Enforces Password Requirements - Minimum Different Categories

The pam_pwquality module's `minclass` parameter controls requirements for usage of different character classes, or types, of character that must exist in a password before it is considered valid. For example, setting this value to three (3) requires that any password must have characters from at least three different categories in order to be approved. The default value is zero (0), meaning there are no required classes. There are four categories available:

```
* Upper-case characters
* Lower-case characters
* Digits
* Special characters (for example, punctuation)
```

Modify the `minclass` setting in `/etc/security/pwquality.conf` entry to require differing categories of characters when changing passwords.

accounts_password_pam_maxrepeat

Set Password Maximum Consecutive Repeating Characters

The pam_pwquality module's `maxrepeat` parameter controls requirements for consecutive repeating characters. When set to a positive number, it will reject passwords which contain more than that number of consecutive characters. Modify the `maxrepeat` setting in `/etc/security/pwquality.conf` to equal to prevent a run of (+ 1) or more identical characters.

accounts_password_pam_ocredit

Ensure PAM Enforces Password Requirements - Minimum Special Characters

The pam_pwquality module's `ocredit=` parameter controls requirements for usage of special (or "other") characters in a password. When set to a negative number, any password will be required to contain that many special characters. When set to a positive number, pam_pwquality will grant +1 additional length credit for each special character. Modify the `ocredit` setting in `/etc/security/pwquality.conf` to equal to require use of a special character in passwords.

accounts_password_pam_lcredit

Ensure PAM Enforces Password Requirements - Minimum Lowercase Characters

The pam_pwquality module's `lcredit` parameter controls requirements for usage of lowercase letters in a password. When set to a negative number, any password will be required to contain that many lowercase characters. When set to a positive number, pam_pwquality will grant +1 additional length credit for each lowercase character. Modify the `lcredit` setting in `/etc/security/pwquality.conf` to require the use of a lowercase character in passwords.

accounts_password_pam_dcredit

Ensure PAM Enforces Password Requirements - Minimum Digit Characters

The pam_pwquality module's `dcredit` parameter controls requirements for usage of digits in a password. When set to a negative number, any password will be required to contain that many digits. When set to a positive number, pam_pwquality will grant +1 additional length credit for each digit. Modify the `dcredit` setting in `/etc/security/pwquality.conf` to require the use of a digit in passwords.

accounts_password_pam_maxclassrepeat

Ensure PAM Enforces Password Requirements - Maximum Consecutive Repeating Characters from Same Character Class

The pam_pwquality module's `maxclassrepeat` parameter controls requirements for consecutive repeating characters from the same character class. When set to a positive number, it will reject passwords which contain more than that number of consecutive characters from the same character class. Modify the `maxclassrepeat` setting in `/etc/security/pwquality.conf` to equal to prevent a run of (+ 1) or more identical characters.

Set Password Quality Requirements, if using pam_cracklib

The `pam_cracklib` PAM module can be configured to meet requirements for a variety of policies.

For example, to configure `pam_cracklib` to require at least one uppercase character, lowercase character, digit, and other (special) character, locate the following line in `/etc/pam.d/system-auth`:

```
password requisite pam_cracklib.so try_first_pass retry=3
```

and then alter it to read:

```
password required pam_cracklib.so try_first_pass retry=3 maxrepeat=3 minlen=14 dcredit=-1 ucredit=-1 ocredit=-1 lcredit=-1 difok=4
```

If no such line exists, add one as the first line of the password section in `/etc/pam.d/system-auth`. The arguments can be modified to ensure compliance with your organization's security policy. Discussion of each parameter follows.

cracklib_accounts_password_pam_maxrepeat

Set Password to Maximum of Three Consecutive Repeating Characters

The pam_cracklib module's `maxrepeat` parameter controls requirements for consecutive repeating characters. When set to a positive number, it will reject passwords which contain more than that number of consecutive characters. Add `maxrepeat=` after pam_cracklib.so to prevent a run of (+ 1) or more identical characters:

```
password required pam_cracklib.so maxrepeat=
```

cracklib_accounts_password_pam_ocredit

Set Password Strength Minimum Special Characters

The pam_cracklib module's `ocredit=` parameter controls requirements for usage of special (or ``other'') characters in a password. When set to a negative number, any password will be required to contain that many special characters. When set to a positive number, pam_cracklib will grant +1 additional length credit for each special character. Add `ocredit=` after pam_cracklib.so to require use of a special character in passwords.

cracklib_accounts_password_pam_ucredit

Set Password Strength Minimum Uppercase Characters

The pam_cracklib module's `ucredit=` parameter controls requirements for usage of uppercase letters in a password. When set to a negative number, any password will be required to contain that many uppercase characters. When set to a positive number, pam_cracklib will grant +1 additional length credit for each uppercase character. Add `ucredit=-1` after pam_cracklib.so to require use of an upper case character in passwords.

cracklib_accounts_password_pam_dcredit

Set Password Strength Minimum Digit Characters

The pam_cracklib module's `dcredit` parameter controls requirements for usage of digits in a password. When set to a negative number, any password will be required to contain that many digits. When set to a positive number, pam_cracklib will grant +1 additional length credit for each digit. Add `dcredit=-1` after pam_cracklib.so to require use of a digit in passwords.

cracklib_accounts_password_pam_minlen

Set Password Minimum Length

The pam_cracklib module's `minlen` parameter controls requirements for minimum characters required in a password. Add `minlen=` after pam_pwquality to set minimum password length requirements.

cracklib_accounts_password_pam_difok

Set Password Strength Minimum Different Characters

The pam_cracklib module's `difok` parameter controls requirements for usage of different characters during a password change. Add `difok=` after pam_cracklib.so to require differing characters when changing passwords. The DoD requirement is `4`.

cracklib_accounts_password_pam_minclass

Set Password Strength Minimum Different Categories

The pam_cracklib module's `minclass` parameter controls requirements for usage of different character classes, or types, of character that must exist in a password before it is considered valid. For example, setting this value to three (3) requires that any password must have characters from at least three different categories in order to be approved. The default value is zero (0), meaning there are no required classes. There are four categories available:

```
* Upper-case characters
* Lower-case characters
* Digits
* Special characters (for example, punctuation)
```

Add `minclass=` after pam_cracklib.so entry into the `/etc/pam.d/system-auth` file in order to require differing categories of characters when changing passwords. For example to require at least three character classes to be used in password, use `minclass=3`.

cracklib_accounts_password_pam_lcredit

Set Password Strength Minimum Lowercase Characters

The pam_cracklib module's `lcredit=` parameter controls requirements for usage of lowercase letters in a password. When set to a negative number, any password will be required to contain that many lowercase characters. When set to a positive number, pam_cracklib will grant +1 additional length credit for each lowercase character. Add `lcredit=-1` after pam_cracklib.so to require use of a lowercase character in passwords.

cracklib_accounts_password_pam_retry

Set Password Retry Prompts Permitted Per-Session

To configure the number of retry prompts that are permitted per-session:

Edit the `pam_cracklib.so` statement in `/etc/pam.d/system-auth` to show `retry=`, or a lower value if site policy is more restrictive.

The DoD requirement is a maximum of 3 prompts per session.

display_login_attempts

Ensure PAM Displays Last Logon/Access Notification

To configure the system to notify users of last logon/access using `pam_lastlog`, add or correct the `pam_lastlog` settings in `/etc/pam.d/postlogin` to read as follows:

```
session     [success=1 default=ignore] pam_succeed_if.so service !~ gdm* service !~ su* quiet
session     [default=1]   pam_lastlog.so nowtmp showfailed
session     optional      pam_lastlog.so silent noupdate showfailed
```

Installing and Maintaining Software

The following sections contain information on security-relevant choices during the initial operating system installation process and the setup of software updates.

Sudo

`Sudo`, which stands for \"su 'do'\", provides the ability to delegate authority to certain users, groups of users, or system administrators. When configured for system users and/or groups, `Sudo` can allow a user or group to execute privileged commands that normally only `root` is allowed to execute.

For more information on `Sudo` and addition `Sudo` configuration options, see **https://www.sudo.ws**.

sudo_remove_no_authenticate

Ensure Users Re-Authenticate for Privilege Escalation - sudo !authenticate

The sudo `!authenticate` option, when specified, allows a user to execute commands using sudo without having to authenticate. This should be disabled by making sure that the `!authenticate` option does not exist in `/etc/sudoers` configuration file or any sudo configuration snippets in `/etc/sudoers.d/`.

sudo_vdsm_nopasswd

Only the VDSM User Can Use sudo NOPASSWD

The sudo `NOPASSWD` tag, when specified, allows a user to execute commands using sudo without having to authenticate. Only the `vdsm` user should have this capability in any sudo configuration snippets in `/etc/sudoers.d/`.

sudo_remove_nopasswd

Ensure Users Re-Authenticate for Privilege Escalation - sudo NOPASSWD

The sudo `NOPASSWD` tag, when specified, allows a user to execute commands using sudo without having to authenticate. This should be disabled by making sure that the `NOPASSWD` tag does not exist in `/etc/sudoers` configuration file or any sudo configuration snippets in `/etc/sudoers.d/`.

sudo_require_authentication

Ensure Users Re-Authenticate for Privilege Escalation - sudo

The sudo `NOPASSWD` and `!authenticate` option, when specified, allows a user to execute commands using sudo without having to authenticate. This should be disabled by making sure that `NOPASSWD` and/or `!authenticate` do not exist in `/etc/sudoers` configuration file or any sudo configuration snippets in `/etc/sudoers.d/`."

GNOME Desktop Environment

GNOME is a graphical desktop environment bundled with many Linux distributions that allow users to easily interact with the operating system graphically rather than textually. The GNOME Graphical Display Manager (GDM) provides login, logout, and user switching contexts as well as display server management.

GNOME is developed by the GNOME Project and is considered the default Red Hat Graphical environment.

For more information on GNOME and the GNOME Project, see **https://www.gnome.org**.

Configure GNOME Screen Locking

In the default GNOME3 desktop, the screen can be locked by selecting the user name in the far right corner of the main panel and selecting **Lock**.

The following sections detail commands to enforce idle activation of the screensaver, screen locking, a blank-screen screensaver, and an idle activation time.

Because users should be trained to lock the screen when they step away from the computer, the automatic locking feature is only meant as a backup.

The root account can be screen-locked; however, the root account should *never* be used to log into an X Windows environment and should only be used to for direct login via console in emergency circumstances.

For more information about enforcing preferences in the GNOME3 environment using the DConf configuration system, see **http://wiki.gnome.org/dconf** and the man page `dconf(1)`.

gconf_gnome_screensaver_idle_delay

Set GNOME Login Inactivity Timeout

Run the following command to set the idle time-out value for inactivity in the GNOME desktop to minutes:

```
$ sudo gconftool-2 \
  --direct \
  --config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \
  --type int \
  --set /desktop/gnome/session/idle_delay
```

dconf_gnome_screensaver_mode_blank

Implement Blank Screensaver

To set the screensaver mode in the GNOME3 desktop to a blank screen, add or set `picture-uri` to `string ''` in `/etc/dconf/db/local.d/00-security-settings`. For example:

```
[org/gnome/desktop/screensaver]
picture-uri=''
```

Once the settings have been added, add a lock to `/etc/dconf/db/local.d/locks/00-security-settings-lock` to prevent user modification. For example:

```
/org/gnome/desktop/screensaver/picture-uri
```

After the settings have been set, run `dconf update`.

gconf_gnome_screen_locking_keybindings

Set GNOME Screen Locking Keybindings

Run the following command to prevent changes to the screensaver lock keybindings:

```
$ sudo gconftool-2 --direct \
--config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \
--type string \
--set /apps/gnome_settings_daemon/keybindings/screensaver "<Control><Alt>l"
```

dconf_gnome_screensaver_idle_activation_locked

Ensure Users Cannot Change GNOME3 Screensaver Idle Activation

If not already configured, ensure that users cannot change GNOME3 screensaver lock settings by adding

```
/org/gnome/desktop/screensaver/idle-activation-enabled
```

to `/etc/dconf/db/local.d/00-security-settings`. For example:

```
/org/gnome/desktop/screensaver/idle-activation-enabled
```

After the settings have been set, run `dconf update`.

gconf_gnome_screensaver_idle_activation_enabled

GNOME Desktop Screensaver Mandatory Use

Run the following command to activate the screensaver in the GNOME desktop after a period of inactivity:

```
$ sudo gconftool-2 --direct \
  --config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \
  --type bool \
  --set /apps/gnome-screensaver/idle_activation_enabled true
```

gconf_gnome_screensaver_max_idle_action

Set GNOME Login Maximum Allowed Inactivity Action

Run the following command to set force logout an inactive user when the maximum allowed inactivity period has expired:

```
$ sudo gconftool-2 --direct \
  --config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \
  --type string \
  --set /desktop/gnome/session/max_idle_action "forced-logout"
```

dconf_gnome_screensaver_idle_delay

Set GNOME3 Screensaver Inactivity Timeout

The idle time-out value for inactivity in the GNOME3 desktop is configured via the `idle-delay` setting must be set under an appropriate configuration file(s) in the `/etc/dconf/db/local.d` directory and locked in `/etc/dconf/db/local.d/locks` directory to prevent user modification.

For example, to configure the system for a 15 minute delay, add the following to `/etc/dconf/db/local.d/00-security-settings`:

```
[org/gnome/desktop/session]
idle-delay=uint32 900
```

Once the setting has been added, add a lock to `/etc/dconf/db/local.d/locks/00-security-settings-lock` to prevent user modification. For example:

```
/org/gnome/desktop/session/idle-delay
```

After the settings have been set, run `dconf update`.

dconf_gnome_screensaver_user_locks

Ensure Users Cannot Change GNOME3 Screensaver Settings

If not already configured, ensure that users cannot change GNOME3 screensaver lock settings by adding `/org/gnome/desktop/screensaver/lock-delay` to `/etc/dconf/db/local.d/locks/00-security-settings-lock` to prevent user modification. For example:

```
/org/gnome/desktop/screensaver/lock-delay
```

After the settings have been set, run `dconf update`.

gconf_gnome_screensaver_max_idle_time

Set GNOME Login Maximum Allowed Inactivity

Run the following command to set the maximum allowed period of inactivity for an inactive user in the GNOME desktop to minutes:

```
$ sudo gconftool-2 \
  --direct \
  --config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \
  --type int \
  --set /desktop/gnome/session/max_idle_time
```

dconf_gnome_screensaver_lock_locked

Ensure Users Cannot Change GNOME3 Screensaver Lock After Idle Period

If not already configured, ensure that users cannot change GNOME3 screensaver lock settings by adding

```
/org/gnome/desktop/screensaver/lock-enabled
```

to `/etc/dconf/db/local.d/00-security-settings`. For example:

```
/org/gnome/desktop/screensaver/lock-enabled
```

After the settings have been set, run `dconf update`.

gconf_gnome_screensaver_mode_blank

Implement Blank Screensaver

Run the following command to set the screensaver mode in the GNOME desktop to a blank screen:

```
$ sudo gconftool-2 --direct \
  --config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \
  --type string \
  --set /apps/gnome-screensaver/mode blank-only
```

dconf_gnome_screensaver_lock_enabled

Enable GNOME3 Screensaver Lock After Idle Period

To activate locking of the screensaver in the GNOME3 desktop when it is activated, add or set `lock-enabled` to `true` in `/etc/dconf/db/local.d/00-security-settings`. For example:

```
[org/gnome/desktop/screensaver]
lock-enabled=true
```

Once the settings have been added, add a lock to `/etc/dconf/db/local.d/locks/00-security-settings-lock` to prevent user modification. For example:

```
/org/gnome/desktop/screensaver/lock-enabled
```

After the settings have been set, run `dconf update`.

dconf_gnome_screensaver_idle_activation_enabled

Enable GNOME3 Screensaver Idle Activation

To activate the screensaver in the GNOME3 desktop after a period of inactivity, add or set `idle-activation-enabled` to `true` in `/etc/dconf/db/local.d/00-security-settings`. For example:

```
[org/gnome/desktop/screensaver]
idle-activation-enabled=true
```

Once the setting has been added, add a lock to `/etc/dconf/db/local.d/locks/00-security-settings-lock` to prevent user modification. For example:

```
/org/gnome/desktop/screensaver/idle-activation-enabled
```

After the settings have been set, run `dconf update`.

dconf_gnome_screensaver_user_info

Disable Full User Name on Splash Shield

By default when the screen is locked, the splash shield will show the user's full name. This should be disabled to prevent casual observers from seeing who has access to the system. This can be disabled by adding or setting `show-full-name-in-top-bar` to `false` in `/etc/dconf/db/local.d/00-security-settings`. For example:

```
[org/gnome/desktop/screensaver]
show-full-name-in-top-bar=false
```

Once the settings have been added, add a lock to `/etc/dconf/db/local.d/locks/00-security-settings-lock` to prevent user modification. For example:

```
/org/gnome/desktop/screensaver/show-full-name-in-top-bar
```

After the settings have been set, run `dconf update`.

dconf_gnome_screensaver_lock_delay

Set GNOME3 Screensaver Lock Delay After Activation Period

To activate the locking delay of the screensaver in the GNOME3 desktop when the screensaver is activated, add or set `lock-delay` to `uint32` in `/etc/dconf/db/local.d/00-security-settings`. For example:

```
[org/gnome/desktop/screensaver]
lock-delay=uint32
```

Once the setting has been added, add a lock to `/etc/dconf/db/local.d/locks/00-security-settings-lock` to prevent user modification. For example:

```
/org/gnome/desktop/screensaver/lock-delay
```

After the settings have been set, run `dconf update`.

gconf_gnome_screensaver_lock_enabled

Enable Screen Lock Activation After Idle Period

Run the following command to activate locking of the screensaver in the GNOME desktop when it is activated:

```
$ sudo gconftool-2 --direct \
  --config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \
  --type bool \
  --set /apps/gnome-screensaver/lock_enabled true
```

dconf_gnome_session_idle_user_locks

Ensure Users Cannot Change GNOME3 Session Idle Settings

If not already configured, ensure that users cannot change GNOME3 session idle settings by adding `/org/gnome/desktop/session/idle-delay` to `/etc/dconf/db/local.d/locks/00-security-settings-lock` to prevent user modification. For example:

```
/org/gnome/desktop/session/idle-delay
```

After the settings have been set, run `dconf update`.

GNOME Media Settings

GNOME media settings that apply to the graphical interface.

gconf_gnome_disable_automount

Disable GNOME Automounting

The system's default desktop environment, GNOME, will mount devices and removable media (such as DVDs, CDs and USB flash drives) whenever they are inserted into the system. Disable automount and autorun within GNOME by running the following:

```
$ sudo gconftool-2 --direct \
  --config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \
  --type bool \
  --set /apps/nautilus/preferences/media_automount false
$ sudo gconftool-2 --direct \
  --config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \
  --type bool \
  --set /apps/nautilus/preferences/media_autorun_never true
```

dconf_gnome_disable_automount

Disable GNOME3 Automounting

The system's default desktop environment, GNOME3, will mount devices and removable media (such as DVDs, CDs and USB flash drives) whenever they are inserted into the system. To disable automount and autorun within GNOME3, add or set `automount` to `false`, `automount-open` to `false`, and `autorun-never` to `true` in `/etc/dconf/db/local.d/00-security-settings`. For example:

```
[org/gnome/desktop/media-handling]
automount=false
automount-open=false
autorun-never=true
```

Once the settings have been added, add a lock to `/etc/dconf/db/local.d/locks/00-security-settings-lock` to prevent user modification. For example:

```
/org/gnome/desktop/media-handling/automount
/org/gnome/desktop/media-handling/automount-open
/org/gnome/desktop/media-handling/autorun-never
```

After the settings have been set, run `dconf update`.

dconf_gnome_disable_thumbnailers

Disable All GNOME3 Thumbnailers

The system's default desktop environment, GNOME3, uses a number of different thumbnailer programs to generate thumbnails for any new or modified content in an opened folder. To disable the execution of these thumbnail applications, add or set `disable-all` to `true` in `/etc/dconf/db/local.d/00-security-settings`. For example:

```
[org/gnome/desktop/thumbnailers]
disable-all=true
```

Once the settings have been added, add a lock to `/etc/dconf/db/local.d/locks/00-security-settings-lock` to prevent user modification. For example:

```
/org/gnome/desktop/thumbnailers/disable-all
```

After the settings have been set, run `dconf update`. This effectively prevents an attacker from gaining access to a system through a flaw in GNOME3's Nautilus thumbnail creators.

gconf_gnome_disable_thumbnailers

Disable All GNOME Thumbnailers

The system's default desktop environment, GNOME, uses a number of different thumbnailer programs to generate thumbnails for any new or modified content in an opened folder. The following command can disable the execution of these thumbnail applications:

```
$ sudo gconftool-2 --direct \
  --config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \
  --type bool \
  --set /desktop/gnome/thumbnailers/disable_all true
```

This effectively prevents an attacker from gaining access to a system through a flaw in GNOME's Nautilus thumbnail creators.

GNOME Remote Access Settings

GNOME remote access settings that apply to the graphical interface.

dconf_gnome_remote_access_encryption

Require Encryption for Remote Access in GNOME3

By default, `GNOME` requires encryption when using `Vino` for remote access. To prevent remote access encryption from being disabled, add or set `require-encryption` to `true` in `/etc/dconf/db/local.d/00-security-settings`. For example:

```
[org/gnome/Vino]
require-encryption=true
```

Once the settings have been added, add a lock to `/etc/dconf/db/local.d/locks/00-security-settings-lock` to prevent user modification. For example:

```
/org/gnome/Vino/require-encryption
```

After the settings have been set, run `dconf update`.

dconf_gnome_remote_access_credential_prompt

Require Credential Prompting for Remote Access in GNOME3

By default, `GNOME` does not require credentials when using `Vino` for remote access. To configure the system to require remote credentials, add or set `authentication-methods` to `['vnc']` in `/etc/dconf/db/local.d/00-security-settings`. For example:

```
[org/gnome/Vino]
authentication-methods=['vnc']
```

Once the settings have been added, add a lock to `/etc/dconf/db/local.d/locks/00-security-settings-lock` to prevent user modification. For example:

```
/org/gnome/Vino/authentication-methods
```

After the settings have been set, run `dconf update`.

Configure GNOME Login Screen

In the default GNOME desktop, the login is displayed after system boot and can display user accounts, allow users to reboot the system, and allow users to login automatically and/or with a guest account. The login screen should be configured to prevent such behavior.

For more information about enforcing preferences in the GNOME3 environment using the DConf configuration system, see **https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Desktop_Migration_and_Administration_Guide/index.html/>** and the man page `dconf(1)`.

dconf_gnome_login_retries

Set the GNOME3 Login Number of Failures

In the default graphical environment, the GNOME3 login screen and be configured to restart the authentication process after a configured number of attempts. This can be configured by setting `allowed-failures` to `3` or less.

To enable, add or edit `allowed-failures` to `/etc/dconf/db/gdm.d/00-security-settings`. For example:

```
[org/gnome/login-screen]
allowed-failures=3
```

Once the setting has been added, add a lock to `/etc/dconf/db/gdm.d/locks/00-security-settings-lock` to prevent user modification. For example:

```
/org/gnome/login-screen/allowed-failures
```

After the settings have been set, run `dconf update`.

gconf_gdm_disable_user_list

Disable the User List

In the default graphical environment, users logging directly into the system are greeted with a login screen that displays all known users. This functionality should be disabled.

Run the following command to disable the user list:

```
$ sudo gconftool-2 --direct \
  --config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \
  --type bool \
  --set /apps/gdm/simple-greeter/disable_user_list true
```

gconf_gnome_disable_restart_shutdown

Disable the GNOME Login Restart and Shutdown Buttons

In the default graphical environment, users logging directly into the system are greeted with a login screen that allows any user, known or unknown, the ability shutdown or restart the system. This functionality should be disabled by running the following:

```
$ sudo gconftool-2 --direct \
  --config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \
  --type bool \
  --set /apps/gdm/simple-greeter/disable_restart_buttons true
```

gnome_gdm_disable_guest_login

Disable GDM Guest Login

The GNOME Display Manager (GDM) can allow users to login without credentials which can be useful for public kiosk scenarios. Allowing users to login without credentials or "guest" account access has inherent security risks and should be disabled. To do disable timed logins or guest account access, set the `TimedLoginEnable` to `false` in the `[daemon]` section in `/etc/gdm/custom.conf`. For example:

```
[daemon]
TimedLoginEnable=false
```

dconf_gnome_enable_smartcard_auth

Enable the GNOME3 Login Smartcard Authentication

In the default graphical environment, smart card authentication can be enabled on the login screen by setting `enable-smartcard-authentication` to `true`.

To enable, add or edit `enable-smartcard-authentication` to `/etc/dconf/db/gdm.d/00-security-settings`. For example:

```
[org/gnome/login-screen]
enable-smartcard-authentication=true
```

Once the setting has been added, add a lock to `/etc/dconf/db/gdm.d/locks/00-security-settings-lock` to prevent user modification. For example:

```
/org/gnome/login-screen/enable-smartcard-authentication
```

After the settings have been set, run `dconf update`.

dconf_gnome_disable_restart_shutdown

Disable the GNOME3 Login Restart and Shutdown Buttons

In the default graphical environment, users logging directly into the system are greeted with a login screen that allows any user, known or unknown, the ability the ability to shutdown or restart the system. This functionality should be disabled by setting `disable-restart-buttons` to `true`.

To disable, add or edit `disable-restart-buttons` to `/etc/dconf/db/gdm.d/00-security-settings`. For example:

```
[org/gnome/login-screen]
disable-restart-buttons=true
```

Once the setting has been added, add a lock to `/etc/dconf/db/gdm.d/locks/00-security-settings-lock` to prevent user modification. For example:

```
/org/gnome/login-screen/disable-restart-buttons
```

After the settings have been set, run `dconf update`.

dconf_gnome_disable_user_list

Disable the GNOME3 Login User List

In the default graphical environment, users logging directly into the system are greeted with a login screen that displays all known users. This functionality should be disabled by setting `disable-user-list` to `true`.

To disable, add or edit `disable-user-list` to `/etc/dconf/db/gdm.d/00-security-settings`. For example:

```
[org/gnome/login-screen]
disable-user-list=true
```

Once the setting has been added, add a lock to `/etc/dconf/db/gdm.d/locks/00-security-settings-lock` to prevent user modification. For example:

```
/org/gnome/login-screen/disable-user-list
```

After the settings have been set, run `dconf update`.

gnome_gdm_disable_automatic_login

Disable GDM Automatic Login

The GNOME Display Manager (GDM) can allow users to automatically login without user interaction or credentials. User should always be required to authenticate themselves to the system that they are authorized to use. To disable user ability to automatically login to the system, set the `AutomaticLoginEnable` to `false` in the `[daemon]` section in `/etc/gdm/custom.conf`. For example:

```
[daemon]
AutomaticLoginEnable=false
```

GNOME Network Settings

GNOME network settings that apply to the graphical interface.

dconf_gnome_disable_wifi_notification

Disable WIFI Network Notification in GNOME3

By default, GNOME disables WIFI notification. This should be permanently set so that users do not connect to a wireless network when the system finds one. While useful for mobile devices, this setting should be disabled for all other systems. To configure the system to disable the WIFI notication, add or set `suppress-wireless-networks-available` to `true` in `/etc/dconf/db/local.d/00-security-settings`. For example:

```
[org/gnome/nm-applet]
suppress-wireless-networks-available=true
```

Once the settings have been added, add a lock to `/etc/dconf/db/local.d/locks/00-security-settings-lock` to prevent user modification. For example:

```
/org/gnome/nm-applet/suppress-wireless-networks-available
```

After the settings have been set, run `dconf update`.

gconf_gnome_disable_wifi_create

Disable WIFI Network Connection Creation in GNOME

GNOME allows users to create ad-hoc wireless connections through the `NetworkManager` applet. Wireless connections should be disabled by running the following:

```
$ sudo gconftool-2 --direct \
  --config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \
  --type bool \
  --set /apps/nm-applet/disable-wifi-create true
```

gconf_gnome_disable_wifi_disconnect

Disable WIFI Network Disconnect Notification in GNOME

By default, GNOME disables WIFI notification when disconnecting from a wireless network. This should be permanently set so that users do not connect to a wireless network when the system finds one. While useful for mobile devices, this setting should be disabled for all other systems. To configure the system to disable the WIFI notication, run the following:

```
$ sudo gconftool-2 --direct \
  --config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \
  --type bool \
  --set /apps/nm-applet/disable-disconnected-notifications true
```

gconf_gnome_disable_wifi_notification

Disable WIFI Network Connection Notification in GNOME

By default, GNOME disables WIFI notification when connecting to a wireless network. This should be permanently set so that users do not connect to a wireless network when the system finds one. While useful for mobile devices, this setting should be disabled for all other systems. To configure the system to disable the WIFI notication, run the following:

```
$ sudo gconftool-2 --direct \
  --config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \
  --type bool \
  --set /apps/nm-applet/disable-connected-notifications true
```

dconf_gnome_disable_wifi_create

Disable WIFI Network Connection Creation in GNOME3

GNOME allows users to create ad-hoc wireless connections through the NetworkManager applet. Wireless connections should be disabled by adding or setting disable-wifi-create to true in /etc/dconf/db/local.d/00-security-settings. For example:

```
[org/gnome/nm-applet]
disable-wifi-create=true
```

Once the settings have been added, add a lock to /etc/dconf/db/local.d/locks/00-security-settings-lock to prevent user modification. For example:

```
/org/gnome/nm-applet/disable-wifi-create
```

After the settings have been set, run dconf update.

GNOME System Settings

GNOME provides configuration and functionality to a graphical desktop environment that changes grahical configurations or allow a user to perform actions that users normally would not be able to do in non-graphical mode such as remote access configuration, power policies, Geo-location, etc. Configuring such settings in GNOME will prevent accidential graphical configuration changes by users from taking place.

gconf_gnome_disable_clock_temperature

Disable the GNOME Clock Temperature Feature

Run the following command to activate locking of the screensaver in the GNOME desktop when it is activated:

```
$ sudo gconftool-2 --direct \
  --config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \
  --type bool \
  --set /apps/panel/applets/clock/prefs/show_temperature false
```

gconf_gnome_disable_clock_weather

Disable the GNOME Clock Weather Feature

Run the following command to activate locking of the screensaver in the GNOME desktop when it is activated:

```
$ sudo gconftool-2 --direct \
  --config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \
  --type bool \
  --set /apps/panel/applets/clock/prefs/show_weather false
```

dconf_gnome_disable_geolocation

Disable Geolocation in GNOME3

`GNOME` allows the clock and applications to track and access location information. This setting should be disabled as applications should not track system location. To configure the system to disable location tracking, add or set `enabled` to `false` in `/etc/dconf/db/local.d/00-security-settings`. For example:

```
[org/gnome/system/location]
enabled=false
```

To configure the clock to disable location tracking, add or set `geolocation` to `false` in `/etc/dconf/db/local.d/00-security-settings`. For example:

```
[org/gnome/clocks]
geolocation=false
```

Once the settings have been added, add a lock to `/etc/dconf/db/local.d/locks/00-security-settings-lock` to prevent user modification. For example:

```
/org/gnome/system/location/enabled
/org/gnome/clocks/geolocation
```

After the settings have been set, run `dconf update`.

dconf_gnome_disable_power_settings

Disable Power Settings in GNOME3

By default, `GNOME` enables a power profile designed for mobile devices with battery usage. While useful for mobile devices, this setting should be disabled for all other systems. To configure the system to disable the power setting, add or set `active` to `false` in `/etc/dconf/db/local.d/00-security-settings`. For example:

```
[org/gnome/settings-daemon/plugins/power]
active=false
```

Once the settings have been added, add a lock to `/etc/dconf/db/local.d/locks/00-security-settings-lock` to prevent user modification. For example:

```
/org/gnome/settings-daemon/plugins/power
```

After the settings have been set, run `dconf update`.

gconf_gnome_disable_ctrlaltdel_reboot

Disable Ctrl-Alt-Del Reboot Key Sequence in GNOME

By default, `GNOME` will reboot the system if the `Ctrl-Alt-Del` key sequence is pressed. To configure the system to ignore the `Ctrl-Alt-Del` key sequence from the Graphical User Interface (GUI) instead of rebooting the system, run the following:

```
$ sudo gconftool-2 --direct \
  --config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \
  --type string \
  --set /apps/gnome_settings_daemon/keybindings/power ""
```

dconf_gnome_disable_user_admin

Disable User Administration in GNOME3

By default, `GNOME` will allow all users to have some administratrion capability. This should be disabled so that non-administrative users are not making configuration changes. To configure the system to disable user administration capability in the Graphical User Interface (GUI), add or set `user-administration-disabled` to `true` in `/etc/dconf/db/local.d/00-security-settings`. For example:

```
[org/gnome/desktop/lockdown]
user-administration-disabled=true
```

Once the settings have been added, add a lock to `/etc/dconf/db/local.d/locks/00-security-settings-lock` to prevent user modification. For example:

```
/org/gnome/desktop/lockdown/user-administration-disabled
```

After the settings have been set, run `dconf update`.

dconf_gnome_disable_ctrlaltdel_reboot

Disable Ctrl-Alt-Del Reboot Key Sequence in GNOME3

By default, `GNOME` will reboot the system if the `Ctrl-Alt-Del` key sequence is pressed.

To configure the system to ignore the `Ctrl-Alt-Del` key sequence from the Graphical User Interface (GUI) instead of rebooting the system, add or set `logout` to `string ''` in `/etc/dconf/db/local.d/00-security-settings`. For example:

```
[org/gnome/settings-daemon/plugins/media-keys]
logout=''
```

Once the settings have been added, add a lock to `/etc/dconf/db/local.d/locks/00-security-settings-lock` to prevent user modification. For example:

```
/org/gnome/settings-daemon/plugins/media-keys/logout
```

After the settings have been set, run `dconf update`.

package_gdm_removed

Remove the GDM Package Group

By removing the `gdm` package, the system no longer has GNOME installed installed. If X Windows is not installed then the system cannot boot into graphical user mode. This prevents the system from being accidentally or maliciously booted into a `graphical.target` mode. To do so, run the following command:

```
$ sudo yum remove gdm
```

dconf_db_up_to_date

Make sure that the dconf databases are up-to-date with regards to respective keyfiles

By default, DConf uses a binary database as a data backend. The system-level database is compiled from keyfiles in the /etc/dconf/db/ directory by the

```
dconf update
```

command.

dconf_use_text_backend

Force dconf to use the textfiles instead of a binary DB

By default, DConf uses a binary database as a data backend. The database is compiled from config files by the

```
dconf update
```

command. dconf can be configured to look into those text files directly by inserting the

```
service-db:keyfile/user
```

directive at the beginning of the

```
/etc/dconf/profile/user
```

file.

enable_dconf_user_profile

Configure GNOME3 DConf User Profile

By default, DConf provides a standard user profile. This profile contains a list of DConf configuration databases. The user profile and database always take the highest priority. As such the DConf User profile should always exist and be configured correctly.

To make sure that the user profile is configured correctly, the `/etc/dconf/profile/user` should be set as follows:

```
user-db:user
system-db:local
system-db:site
system-db:distro
```

Disk Partitioning

To ensure separation and protection of data, there are top-level system directories which should be placed on their own physical partition or logical volume. The installer's default partitioning scheme creates separate logical volumes for `/`, `/boot`, and `swap`.

- If starting with any of the default layouts, check the box to \"Review and modify partitioning.\" This allows for the easy creation of additional logical volumes inside the volume group already created, though it may require making `/`'s logical volume smaller to create space. In general, using logical volumes is preferable to using partitions because they can be more easily adjusted later.
- If creating a custom layout, create the partitions mentioned in the previous paragraph (which the installer will require anyway), as well as separate ones described in the following sections.

If a system has already been installed, and the default partitioning scheme was used, it is possible but nontrivial to modify it to create separate logical volumes for the directories listed above. The Logical Volume Manager (LVM) makes this possible. See the LVM HOWTO at http://tldp.org/HOWTO/LVM-HOWTO/ for more detailed information on LVM.

partition_for_var_log_audit

Ensure /var/log/audit Located On Separate Partition

Audit logs are stored in the `/var/log/audit` directory. Ensure that it has its own partition or logical volume at installation time, or migrate it later using LVM. Make absolutely certain that it is large enough to store all audit logs that will be created by the auditing daemon.

partition_for_var_tmp

Ensure /var/tmp Located On Separate Partition

The `/var/tmp` directory is a world-writable directory used for temporary file storage. Ensure it has its own partition or logical volume at installation time, or migrate it using LVM.

partition_for_tmp

Ensure /tmp Located On Separate Partition

The `/tmp` directory is a world-writable directory used for temporary file storage. Ensure it has its own partition or logical volume at installation time, or migrate it using LVM.

partition_for_srv

Ensure /srv Located On Separate Partition

If a file server (FTP, TFTP...) is hosted locally, create a separate partition for `/srv` at installation time (or migrate it later using LVM). If `/srv` will be mounted from another system such as an NFS server, then creating a separate partition is not necessary at installation time, and the mountpoint can instead be configured later.

partition_for_home

Ensure /home Located On Separate Partition

If user home directories will be stored locally, create a separate partition for `/home` at installation time (or migrate it later using LVM). If `/home` will be mounted from another system such as an NFS server, then creating a separate partition is not necessary at installation time, and the mountpoint can instead be configured later.

encrypt_partitions

Encrypt Partitions

Red Hat Enterprise Linux 8 natively supports partition encryption through the Linux Unified Key Setup-on-disk-format (LUKS) technology. The easiest way to encrypt a partition is during installation time.

For manual installations, select the `Encrypt` checkbox during partition creation to encrypt the partition. When this option is selected the system will prompt for a passphrase to use in decrypting the partition. The passphrase will subsequently need to be entered manually every time the system boots.

For automated/unattended installations, it is possible to use Kickstart by adding the `--encrypted` and `--passphrase=` options to the definition of each partition to be encrypted. For example, the following line would encrypt the root partition:

```
part / --fstype=ext4 --size=100 --onpart=hda1 --encrypted --passphrase=PASSPHRASE
```

Any *PASSPHRASE* is stored in the Kickstart in plaintext, and the Kickstart must then be protected accordingly. Omitting the `--passphrase=` option from the partition definition will cause the installer to pause and interactively ask for the passphrase during installation.

By default, the `Anaconda` installer uses `aes-xts-plain64` cipher with a minimum `512` bit key size which should be compatible with FIPS enabled.

Detailed information on encrypting partitions using LUKS or LUKS ciphers can be found on the Red Hat Enterprise Linux 8 Documentation web site: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Security_Guide/sec-Encryption.html.

partition_for_var

Ensure /var Located On Separate Partition

The `/var` directory is used by daemons and other system services to store frequently-changing data. Ensure that `/var` has its own partition or logical volume at installation time, or migrate it using LVM.

partition_for_var_log

Ensure /var/log Located On Separate Partition

System logs are stored in the `/var/log` directory. Ensure that it has its own partition or logical volume at installation time, or migrate it using LVM.

System and Software Integrity

System and software integrity can be gained by installing antivirus, increasing system encryption strength with FIPS, verifying installed software, enabling SELinux, installing an Intrusion Prevention System, etc. However, installing or enabling integrity checking tools cannot *prevent* intrusions, but they can detect that an intrusion may have occurred. Requirements for integrity checking may be highly dependent on the environment in which the system will be used. Snapshot-based approaches such as AIDE may induce considerable overhead in the presence of frequent software updates.

Software Integrity Checking

Both the AIDE (Advanced Intrusion Detection Environment) software and the RPM package management system provide mechanisms for verifying the integrity of installed software. AIDE uses snapshots of file metadata (such as hashes) and compares these to current system files in order to detect changes.

The RPM package management system can conduct integrity checks by comparing information in its metadata database with files installed on the system.

Verify Integrity with RPM

The RPM package management system includes the ability to verify the integrity of installed packages by comparing the installed files with information about the files taken from the package metadata stored in the RPM database. Although an attacker could corrupt the RPM database (analogous to attacking the AIDE database as described above), this check can still reveal modification of important files. To list which files on the system differ from what is expected by the RPM database:

```
$ rpm -qVa
```

See the man page for `rpm` to see a complete explanation of each column.

rpm_verify_ownership

Verify and Correct Ownership with RPM

The RPM package management system can check file ownership permissions of installed software packages, including many that are important to system security. After locating a file with incorrect permissions, which can be found with

```
rpm -Va | awk '{ if (substr($0,6,1)=="U" || substr($0,7,1)=="G") print $NF }'
```

run the following command to determine which package owns it:

```
$ rpm -qf FILENAME
```

Next, run the following command to reset its permissions to the correct values:

```
$ sudo rpm --setugids PACKAGENAME
```

rpm_verify_hashes

Verify File Hashes with RPM

Without cryptographic integrity protections, system executables and files can be altered by unauthorized users without detection. The RPM package management system can check the hashes of installed software packages, including many that are important to system security. To verify that the cryptographic hash of system files and commands match vendor values, run the following command to list which files on the system have hashes that differ from what is expected by the RPM database:

```
$ rpm -Va | grep '^..5'
```

A "c" in the second column indicates that a file is a configuration file, which may appropriately be expected to change. If the file was not expected to change, investigate the cause of the change using audit logs or other means. The package can then be reinstalled to restore the file. Run the following command to determine which package owns the file:

```
$ rpm -qf FILENAME
```

The package can be reinstalled from a yum repository using the command:

```
$ sudo yum reinstall PACKAGENAME
```

Alternatively, the package can be reinstalled from trusted media using the command:

```
$ sudo rpm -Uvh PACKAGENAME
```

rpm_verify_permissions

Verify and Correct File Permissions with RPM

The RPM package management system can check file access permissions of installed software packages, including many that are important to system security. Verify that the file permissions of system files and commands match vendor values. Check the file permissions with the following command:

```
$ sudo rpm -Va | awk '{ if (substr($0,2,1)=="M") print $NF }'
```

Output indicates files that do not match vendor defaults. After locating a file with incorrect permissions, run the following command to determine which package owns it:

```
$ rpm -qf FILENAME
```

Next, run the following command to reset its permissions to the correct values:

```
$ sudo rpm --setperms PACKAGENAME
```

Verify Integrity with AIDE

AIDE conducts integrity checks by comparing information about files with previously-gathered information. Ideally, the AIDE database is created immediately after initial system configuration, and then again after any software update. AIDE is highly configurable, with further configuration information located in `/usr/share/doc/aide-VERSION`.

aide_use_fips_hashes

Configure AIDE to Use FIPS 140-2 for Validating Hashes

By default, the `sha512` option is added to the `NORMAL` ruleset in AIDE. If using a custom ruleset or the `sha512` option is missing, add `sha512` to the appropriate ruleset. For example, add `sha512` to the following line in `/etc/aide.conf`:

```
NORMAL = FIPSR+sha512
```

AIDE rules can be configured in multiple ways; this is merely one example that is already configured by default.

package_aide_installed

Install AIDE

The `aide` package can be installed with the following command:

```
$ sudo yum install aide
```

aide_build_database

Build and Test AIDE Database

Run the following command to generate a new database:

```
$ sudo /usr/sbin/aide --init
```

By default, the database will be written to the file `/var/lib/aide/aide.db.new.gz`. Storing the database, the configuration file `/etc/aide.conf`, and the binary `/usr/sbin/aide` (or hashes of these files), in a secure location (such as on read-only media) provides additional assurance about their integrity. The newly-generated database can be installed as follows:

```
$ sudo cp /var/lib/aide/aide.db.new.gz /var/lib/aide/aide.db.gz
```

To initiate a manual check, run the following command:

```
$ sudo /usr/sbin/aide --check
```

If this check produces any unexpected output, investigate.

aide_verify_acls

Configure AIDE to Verify Access Control Lists (ACLs)

By default, the `acl` option is added to the `FIPSR` ruleset in AIDE. If using a custom ruleset or the `acl` option is missing, add `acl` to the appropriate ruleset. For example, add `acl` to the following line in `/etc/aide.conf`:

```
FIPSR = p+i+n+u+g+s+m+c+acl+selinux+xattrs+sha256
```

AIDE rules can be configured in multiple ways; this is merely one example that is already configured by default.

aide_scan_notification

Configure Notification of Post-AIDE Scan Details

AIDE should notify appropriate personnel of the details of a scan after the scan has been run. If AIDE has already been configured for periodic execution in `/etc/crontab`, append the following line to the existing AIDE line:

```
| /bin/mail -s "$(hostname) - AIDE Integrity Check" root@localhost
```

Otherwise, add the following line to `/etc/crontab`:

```
05 4 * * * root /usr/sbin/aide --check | /bin/mail -s "$(hostname) - AIDE Integrity Check" root@localhost
```

AIDE can be executed periodically through other means; this is merely one example.

aide_periodic_cron_checking

Configure Periodic Execution of AIDE

At a minimum, AIDE should be configured to run a weekly scan. At most, AIDE should be run daily. To implement a daily execution of AIDE at 4:05am using cron, add the following line to `/etc/crontab`:

```
05 4 * * * root /usr/sbin/aide --check
```

To implement a weekly execution of AIDE at 4:05am using cron, add the following line to `/etc/crontab`:

```
05 4 * * 0 root /usr/sbin/aide --check
```

AIDE can be executed periodically through other means; this is merely one example. The usage of cron's special time codes, such as `@daily` and `@weekly` is acceptable.

aide_verify_ext_attributes

Configure AIDE to Verify Extended Attributes

By default, the `xattrs` option is added to the `FIPSR` ruleset in AIDE. If using a custom ruleset or the `xattrs` option is missing, add `xattrs` to the appropriate ruleset. For example, add `xattrs` to the following line in `/etc/aide.conf`:

```
FIPSR = p+i+n+u+g+s+m+c+acl+selinux+xattrs+sha256
```

AIDE rules can be configured in multiple ways; this is merely one example that is already configured by default.

Federal Information Processing Standard (FIPS)

The Federal Information Processing Standard (FIPS) is a computer security standard which is developed by the U.S. Government and industry working groups to validate the quality of cryptographic modules. The FIPS standard provides four security levels to ensure adequate coverage of different industries, implementation of cryptographic modules, and organizational sizes and requirements.

FIPS 140-2 is the current standard for validating that mechanisms used to access cryptographic modules utilize authentication that meets industry and government requirements. For government systems, this allows Security Levels 1, 2, 3, or 4 for use on Red Hat Enterprise Linux 8.

See **http://csrc.nist.gov/publications/PubsFIPS.html** for more information.

enable_dracut_fips_module

Enable Dracut FIPS Module

To enable FIPS mode, run the following command:

```
fips-mode-setup --enable
```

To enable FIPS, the system requires that the `fips` module is added in `dracut` configuration. Check if `/etc/dracut.conf.d/40-fips.conf` contain `add_dracutmodules+=" fips "`

etc_system_fips_exists

Ensure '/etc/system-fips' exists

On a system where FIPS mode is enabled, `/etc/system-fips` must exist. To enable FIPS mode, run the following command:

```
fips-mode-setup --enable
```

sysctl_crypto_fips_enabled

Set kernel parameter 'crypto.fips_enabled' to 1

System running in FIPS mode is indicated by kernel parameter `'crypto.fips_enabled'`. This parameter should be set to `1` in FIPS mode. To enable FIPS mode, run the following command:

```
fips-mode-setup --enable
```

enable_fips_mode

Enable FIPS Mode

To enable FIPS mode, run the following command:

```
fips-mode-setup --enable
```

The `fips-mode-setup` command will configure the system in FIPS mode by automatically configuring the following:

- Setting the kernel FIPS mode flag (`/proc/sys/crypto/fips_enabled`) to `1`
- Creating `/etc/system-fips`
- Setting the system crypto policy in `/etc/crypto-policies/config` to `FIPS`
- Loading the Dracut `fips` module

Furthermore, the system running in FIPS mode should be FIPS certified by NIST.

Endpoint Protection Software

Endpoint protection security software that is not provided or supported by Red Hat can be installed to provide complementary or duplicative security capabilities to those provided by the base platform. Add-on software may not be appropriate for some specialized systems.

McAfee Endpoint Security Software

In DoD environments, McAfee Host-based Security System (HBSS) and VirusScan Enterprise for Linux (VSEL) is required to be installed on all systems.

McAfee Host-Based Intrusion Detection Software (HBSS)

McAfee Host-based Security System (HBSS) is a suite of software applications used to monitor, detect, and defend computer networks and systems.

install_mcafee_hbss_pa

Install the Policy Auditor (PA) Module

Install the Policy Auditor (PA) Module.

install_mcafee_hbss_accm

Install the Asset Configuration Compliance Module (ACCM)

Install the Asset Configuration Compliance Module (ACCM).

install_mcafee_hbss_hips

Install the Host Intrusion Prevention System (HIPS) Module

Install the McAfee Host Intrusion Prevention System (HIPS) Module if it is absolutely necessary. If SELinux is enabled, do not install or enable this module.

mcafee_antivirus_definitions_updated

Virus Scanning Software Definitions Are Updated

Ensure virus definition files are no older than 7 days or their last release.

install_mcafee_antivirus

Install McAfee Virus Scanning Software

Install McAfee VirusScan Enterprise for Linux antivirus software which is provided for DoD systems and uses signatures to search for the presence of viruses on the filesystem.

service_nails_enabled

Enable nails Service

The `nails` service is used to run McAfee VirusScan Enterprise for Linux and McAfee Host-based Security System (HBSS) services. The `nails` service can be enabled with the following command:

```
$ sudo systemctl enable nails.service
```

install_mcafee_cma_rt

Install the McAfee Runtime Libraries and Linux Agent

Install the McAfee Runtime Libraries (MFErt) and Linux Agent (MFEcma).

configure_user_data_backups

Configure Backups of User Data

The operating system must conduct backups of user data contained in the operating system. The operating system provides utilities for automating backups of user data. Commercial and open-source products are also available.

install_hids

Install Intrusion Detection Software

The base Red Hat Enterprise Linux 8 platform already includes a sophisticated auditing system that can detect intruder activity, as well as SELinux, which provides host-based intrusion prevention capabilities by confining privileged programs and user sessions which may become compromised.

install_antivirus

Install Virus Scanning Software

Install virus scanning software, which uses signatures to search for the presence of viruses on the filesystem. Ensure virus definition files are no older than 7 days, or their last release. Configure the virus scanning software to perform scans dynamically on all accessed files. If this is not possible, configure the system to scan all altered files on the system on a daily basis. If the system processes inbound SMTP mail, configure the virus scanner to scan all received mail.

Operating System Vendor Support and Certification

The assurance of a vendor to provide operating system support and maintenance for their product is an important criterion to ensure product stability and security over the life of the product. A certified product that follows the necessary standards and government certification requirements guarantees that known software vulnerabilities will be remediated, and proper guidance for protecting and securing the operating system will be given.

installed_OS_is_FIPS_certified

The Installed Operating System Is FIPS 140-2 Certified

To enable processing of sensitive information the operating system must provide certified cryptographic modules compliant with FIPS 140-2 standard.

installed_OS_is_vendor_supported

The Installed Operating System Is Vendor Supported

The installed operating system must be maintained by a vendor. Red Hat Enterprise Linux is supported by Red Hat, Inc. As the Red Hat Enterprise Linux vendor, Red Hat, Inc. is responsible for providing security patches.

System Cryptographic Policies

Linux has the capability to centrally configure cryptographic polices. The command `update-crypto-policies` is used to set the policy applicable for the various cryptographic back-ends, such as SSL/TLS libraries. The configured cryptographic policies will be the default policy used by these backends unless the application user configures them otherwise. When the system has been configured to use the centralized cryptographic policies, the administrator is assured that any application that utilizes the supported backends will follow a policy that adheres to the configured profile. Currently the supported backends are:

- GnuTLS library
- OpenSSL library
- NSS library
- OpenJDK
- Libkrb5
- BIND
- OpenSSH

Applications and languages which rely on any of these backends will follow the system policies as well. Examples are apache httpd, nginx, php, and others.

configure_crypto_policy

Configure System Cryptography Policy

To configure the system cyptography policy to use ciphers only from the `` policy, run the following command:

```
$ sudo update-crypto-policies --set
```

configure_bind_crypto_policy

Configure BIND to use System Crypto Policy

Crypto Policies provide a centralized control over crypto algorithms usage of many packages. BIND is supported by crypto policy, but the BIND configuration may be set up to ignore it. To check that Crypto Policies settings are configured correctly, ensure that the `/etc/named.conf` includes the appropriate configuration: In the `options` section of `/etc/named.conf`, make sure that the following line is not commented out or superseded by later includes: `include "/etc/crypto-policies/back-ends/bind.config";`

configure_openssl_crypto_policy

Configure OpenSSL library to use System Crypto Policy

Crypto Policies provide a centralized control over crypto algorithms usage of many packages. OpenSSL is supported by crypto policy, but the OpenSSL configuration may be set up to ignore it. To check that Crypto Policies settings are configured correctly, you have to examine the OpenSSL config file available under `/etc/pki/tls/openssl.cnf`. This file has the `ini` format, and it enables crypto policy support if there is a `[ crypto_policy ]` section that contains the `.include /etc/crypto-policies/back-ends/openssl.config` directive.

configure_libreswan_crypto_policy

Configure Libreswan to use System Crypto Policy

Crypto Policies provide a centralized control over crypto algorithms usage of many packages. Libreswan is supported by system crypto policy, but the Libreswan configuration may be set up to ignore it. To check that Crypto Policies settings are configured correctly, ensure that the `/etc/ipsec.conf` includes the appropriate configuration file. In `/etc/ipsec.conf`, make sure that the following line is not commented out or superseded by later includes: `include /etc/crypto-policies/back-ends/libreswan.config`

configure_ssh_crypto_policy

Configure SSH to use System Crypto Policy

Crypto Policies provide a centralized control over crypto algorithms usage of many packages. SSH is supported by crypto policy, but the SSH configuration may be set up to ignore it. To check that Crypto Policies settings are configured correctly, ensure that the `CRYPTO_POLICY` variable is either commented or not set at all in the `/etc/sysconfig/sshd`.

configure_kerberos_crypto_policy

Configure Kerberos to use System Crypto Policy

Crypto Policies provide a centralized control over crypto algorithms usage of many packages. Kerberos is supported by crypto policy, but it's configuration may be set up to ignore it. To check that Crypto Policies settings for Kerberos are configured correctly, examine that there is a symlink at /etc/krb5.conf.d/crypto-policies targeting /etc/cypto-policies/back-ends/krb5.config. If the symlink exists, kerberos is configured to use the system-wide crypto policy settings.

disable_prelink

Disable Prelinking

The prelinking feature changes binaries in an attempt to decrease their startup time. In order to disable it, change or add the following line inside the file `/etc/sysconfig/prelink`:

```
PRELINKING=no
```

Next, run the following command to return binaries to a normal, non-prelinked state:

```
$ sudo /usr/sbin/prelink -ua
```

Updating Software

The `yum` command line tool is used to install and update software packages. The system also provides a graphical software update tool in the **System** menu, in the **Administration** submenu, called **Software Update**.

Red Hat Enterprise Linux 8 systems contain an installed software catalog called the RPM database, which records metadata of installed packages. Consistently using `yum` or the graphical **Software Update** for all software installation allows for insight into the current inventory of installed software on the system.

security_patches_up_to_date

Ensure Software Patches Installed

If the system is joined to the Red Hat Network, a Red Hat Satellite Server, or a yum server, run the following command to install updates:

```
$ sudo yum update
```

If the system is not configured to use one of these sources, updates (in the form of RPM packages) can be manually downloaded from the Red Hat Network and installed using `rpm`.

NOTE: U.S. Defense systems are required to be patched within 30 days or sooner as local policy dictates.

ensure_gpgcheck_local_packages

Ensure gpgcheck Enabled for Local Packages

`yum` should be configured to verify the signature(s) of local packages prior to installation. To configure `yum` to verify signatures of local packages, set the `localpkg_gpgcheck` to `1` in `/etc/yum.conf`.

clean_components_post_updating

Ensure yum Removes Previous Package Versions

`yum` should be configured to remove previous software components after new versions have been installed. To configure `yum` to remove the previous software components after updating, set the `clean_requirements_on_remove` to `1` in `/etc/yum.conf`.

ensure_gpgcheck_repo_metadata

Ensure gpgcheck Enabled for Repository Metadata

Verify the operating system prevents the installation of patches, service packs, device drivers, or operating system components of local packages without verification of the repository metadata. Check that `yum` verifies the repository metadata prior to install with the following command. This should be configured by setting `repo_gpgcheck` to `1` in `/etc/yum.conf`.

ensure_gpgcheck_never_disabled

Ensure gpgcheck Enabled for All yum Package Repositories

To ensure signature checking is not disabled for any repos, remove any lines from files in `/etc/yum.repos.d` of the form:

```
gpgcheck=0
```

ensure_redhat_gpgkey_installed

Ensure Red Hat GPG Key Installed

To ensure the system can cryptographically verify base software packages come from Red Hat (and to connect to the Red Hat Network to receive them), the Red Hat GPG key must properly be installed. To install the Red Hat GPG key, run:

```
$ sudo subscription-manager register
```

If the system is not connected to the Internet or an RHN Satellite, then install the Red Hat GPG key from trusted media such as the Red Hat installation CD-ROM or DVD. Assuming the disc is mounted in `/media/cdrom`, use the following command as the root user to import it into the keyring:

```
$ sudo rpm --import /media/cdrom/RPM-GPG-KEY
```

ensure_gpgcheck_globally_activated

Ensure gpgcheck Enabled In Main yum Configuration

The `gpgcheck` option controls whether RPM packages' signatures are always checked prior to installation. To configure yum to check package signatures before installing them, ensure the following line appears in `/etc/yum.conf` in the `[main]` section:

```
gpgcheck=1
```

SAP Specific Requirement

SAP (Systems, Applications and Products in Data Processing) is enterprise software to manage business operations and customer relations. The following section contains SAP specific requirement that is not part of standard or common OS setting.

Network Configuration and Firewalls

Most systems must be connected to a network of some sort, and this brings with it the substantial risk of network attack. This section discusses the security impact of decisions about networking which must be made when configuring a system.

This section also discusses firewalls, network access controls, and other network security frameworks, which allow system-level rules to be written that can limit an attackers' ability to connect to your system. These rules can specify that network traffic should be allowed or denied from certain IP addresses, hosts, and networks. The rules can also specify which of the system's network services are available to particular hosts or networks.

Disable Unused Interfaces

Network interfaces expand the attack surface of the system. Unused interfaces are not monitored or controlled, and should be disabled.

If the system does not require network communications but still needs to use the loopback interface, remove all files of the form `ifcfg-interface` except for `ifcfg-lo` from `/etc/sysconfig/network-scripts`:

```
$ sudo rm /etc/sysconfig/network-scripts/ifcfg-interface
```

If the system is a standalone machine with no need for network access or even communication over the loopback device, then disable this service. The `network` service can be disabled with the following command:

```
$ sudo systemctl disable network.service
```

IPSec Support

Support for Internet Protocol Security (IPsec)

package_libreswan_installed

Install libreswan Package

The Libreswan package provides an implementation of IPsec and IKE, which permits the creation of secure tunnels over untrusted networks. The `libreswan` package can be installed with the following command:

```
$ sudo yum install libreswan
```

libreswan_approved_tunnels

Verify Any Configured IPSec Tunnel Connections

Libreswan provides an implementation of IPsec and IKE, which permits the creation of secure tunnels over untrusted networks. As such, IPsec can be used to circumvent certain network requirements such as filtering. Verify that if any IPsec connection (`conn`) configured in `/etc/ipsec.conf` and `/etc/ipsec.d` exists is an approved organizational connection.

Uncommon Network Protocols

The system includes support for several network protocols which are not commonly used. Although security vulnerabilities in kernel networking code are not frequently discovered, the consequences can be dramatic. Ensuring uncommon network protocols are disabled reduces the system's risk to attacks targeted at its implementation of those protocols.

kernel_module_tipc_disabled

Disable TIPC Support

The Transparent Inter-Process Communication (TIPC) protocol is designed to provide communications between nodes in a cluster. To configure the system to prevent the `tipc` kernel module from being loaded, add the following line to a file in the directory `/etc/modprobe.d`:

```
install tipc /bin/true
```

kernel_module_sctp_disabled

Disable SCTP Support

The Stream Control Transmission Protocol (SCTP) is a transport layer protocol, designed to support the idea of message-oriented communication, with several streams of messages within one connection. To configure the system to prevent the `sctp` kernel module from being loaded, add the following line to a file in the directory `/etc/modprobe.d`:

```
install sctp /bin/true
```

kernel_module_dccp_disabled

Disable DCCP Support

The Datagram Congestion Control Protocol (DCCP) is a relatively new transport layer protocol, designed to support streaming media and telephony. To configure the system to prevent the `dccp` kernel module from being loaded, add the following line to a file in the directory `/etc/modprobe.d`:

```
install dccp /bin/true
```

kernel_module_rds_disabled

Disable RDS Support

The Reliable Datagram Sockets (RDS) protocol is a transport layer protocol designed to provide reliable high- bandwidth, low-latency communications between nodes in a cluster. To configure the system to prevent the `rds` kernel module from being loaded, add the following line to a file in the directory `/etc/modprobe.d`:

```
install rds /bin/true
```

iptables and ip6tables

A host-based firewall called `netfilter` is included as part of the Linux kernel distributed with the system. It is activated by default. This firewall is controlled by the program `iptables`, and the entire capability is frequently referred to by this name. An analogous program called `ip6tables` handles filtering for IPv6.

Unlike TCP Wrappers, which depends on the network server program to support and respect the rules written, `netfilter` filtering occurs at the kernel level, before a program can even process the data from the network packet. As such, any program on the system is affected by the rules written.

This section provides basic information about strengthening the `iptables` and `ip6tables` configurations included with the system. For more complete information that may allow the construction of a sophisticated ruleset tailored to your environment, please consult the references at the end of this section.

Strengthen the Default Ruleset

The default rules can be strengthened. The system scripts that activate the firewall rules expect them to be defined in the configuration files `iptables` and `ip6tables` in the directory `/etc/sysconfig`. Many of the lines in these files are similar to the command line arguments that would be provided to the programs `/sbin/iptables` or `/sbin/ip6tables` - but some are quite different.

The following recommendations describe how to strengthen the default ruleset configuration file. An alternative to editing this configuration file is to create a shell script that makes calls to the iptables program to load in rules, and then invokes service iptables save to write those loaded rules to `/etc/sysconfig/iptables`.

The following alterations can be made directly to `/etc/sysconfig/iptables` and `/etc/sysconfig/ip6tables`. Instructions apply to both unless otherwise noted. Language and address conventions for regular iptables are used throughout this section; configuration for ip6tables will be either analogous or explicitly covered.

Restrict ICMP Message Types

In `/etc/sysconfig/iptables`, the accepted ICMP messages types can be restricted. To accept only ICMP echo reply, destination unreachable, and time exceeded messages, remove the line:

```
-A INPUT -p icmp --icmp-type any -j ACCEPT
```

and insert the lines:

```
-A INPUT -p icmp --icmp-type echo-reply -j ACCEPT
-A INPUT -p icmp --icmp-type destination-unreachable -j ACCEPT
-A INPUT -p icmp --icmp-type time-exceeded -j ACCEPT
```

To allow the system to respond to pings, also insert the following line:

```
-A INPUT -p icmp --icmp-type echo-request -j ACCEPT
```

Ping responses can also be limited to certain networks or hosts by using the -s option in the previous rule. Because IPv6 depends so heavily on ICMPv6, it is preferable to deny the ICMPv6 packets you know you don't need (e.g. ping requests) in `/etc/sysconfig/ip6tables`, while letting everything else through:

```
-A INPUT -p icmpv6 --icmpv6-type echo-request -j DROP
```

If you are going to statically configure the system's address, it should ignore Router Advertisements which could add another IPv6 address to the interface or alter important network settings:

```
-A INPUT -p icmpv6 --icmpv6-type router-advertisement -j DROP
```

Restricting ICMPv6 message types in `/etc/sysconfig/ip6tables` is not recommended because the operation of IPv6 depends heavily on ICMPv6. Thus, great care must be taken if any other ICMPv6 types are blocked.

Log and Drop Packets with Suspicious Source Addresses

Packets with non-routable source addresses should be rejected, as they may indicate spoofing. Because the modified policy will reject non-matching packets, you only need to add these rules if you are interested in also logging these spoofing or suspicious attempts before they are dropped. If you do choose to log various suspicious traffic, add identical rules with a target of `DROP` after each *LOG*. To log and then drop these IPv4 packets, insert the following rules in `/etc/sysconfig/iptables` (excepting any that are intentionally used):

```
-A INPUT -s 10.0.0.0/8 -j LOG --log-prefix "IP DROP SPOOF A: "
-A INPUT -s 172.16.0.0/12 -j LOG --log-prefix "IP DROP SPOOF B: "
-A INPUT -s 192.168.0.0/16 -j LOG --log-prefix "IP DROP SPOOF C: "
-A INPUT -s 224.0.0.0/4 -j LOG --log-prefix "IP DROP MULTICAST D: "
-A INPUT -s 240.0.0.0/5 -j LOG --log-prefix "IP DROP SPOOF E: "
-A INPUT -d 127.0.0.0/8 -j LOG --log-prefix "IP DROP LOOPBACK: "
```

Similarly, you might wish to log packets containing some IPv6 reserved addresses if they are not expected on your network:

```
-A INPUT -i eth0 -s ::1 -j LOG --log-prefix "IPv6 DROP LOOPBACK: "
-A INPUT -s 2002:E000::/20 -j LOG --log-prefix "IPv6 6to4 TRAFFIC: "
-A INPUT -s 2002:7F00::/24 -j LOG --log-prefix "IPv6 6to4 TRAFFIC: "
-A INPUT -s 2002:0000::/24 -j LOG --log-prefix "IPv6 6to4 TRAFFIC: "
-A INPUT -s 2002:FF00::/24 -j LOG --log-prefix "IPv6 6to4 TRAFFIC: "
-A INPUT -s 2002:0A00::/24 -j LOG --log-prefix "IPv6 6to4 TRAFFIC: "
-A INPUT -s 2002:AC10::/28 -j LOG --log-prefix "IPv6 6to4 TRAFFIC: "
-A INPUT -s 2002:C0A8::/32 -j LOG --log-prefix "IPv6 6to4 TRAFFIC: "
```

If you are not expecting to see site-local multicast or auto-tunneled traffic, you can log those:

```
-A INPUT -s FF05::/16 -j LOG --log-prefix "IPv6 SITE-LOCAL MULTICAST: "
-A INPUT -s ::0.0.0.0/96 -j LOG --log-prefix "IPv4 COMPATIBLE IPv6 ADDR: "
```

If you wish to block multicasts to all link-local nodes (e.g. if you are not using router auto-configuration and do not plan to have any services that multicast to the entire local network), you can block the link-local all-nodes multicast address (before accepting incoming ICMPv6):

```
-A INPUT -d FF02::1 -j LOG --log-prefix "Link-local All-Nodes Multicast: "
```

However, if you're going to allow IPv4 compatible IPv6 addresses (of the form ::0.0.0.0/96), you should then consider logging the non-routable IPv4-compatible addresses:

```
-A INPUT -s ::0.0.0.0/104 -j LOG --log-prefix "IP NON-ROUTABLE ADDR: "
-A INPUT -s ::127.0.0.0/104 -j LOG --log-prefix "IP DROP LOOPBACK: "
-A INPUT -s ::224.0.0.0.0/100 -j LOG --log-prefix "IP DROP MULTICAST D: "
-A INPUT -s ::255.0.0.0/104 -j LOG --log-prefix "IP BROADCAST: "
```

If you are not expecting to see any IPv4 (or IPv4-compatible) traffic on your network, consider logging it before it gets dropped:

```
-A INPUT -s ::FFFF:0.0.0.0/96 -j LOG --log-prefix "IPv4 MAPPED IPv6 ADDR: "
-A INPUT -s 2002::/16 -j LOG --log-prefix "IPv6 6to4 ADDR: "
```

The following rule will log all traffic originating from a site-local address, which is deprecated address space:

```
-A INPUT -s FEC0::/10 -j LOG --log-prefix "SITE-LOCAL ADDRESS TRAFFIC: "
```

set_iptables_default_rule

Set Default iptables Policy for Incoming Packets

To set the default policy to DROP (instead of ACCEPT) for the built-in INPUT chain which processes incoming packets, add or correct the following line in `/etc/sysconfig/iptables`:

```
:INPUT DROP [0:0]
```

set_iptables_default_rule_forward

Set Default iptables Policy for Forwarded Packets

To set the default policy to DROP (instead of ACCEPT) for the built-in FORWARD chain which processes packets that will be forwarded from one interface to another, add or correct the following line in `/etc/sysconfig/iptables`:

```
:FORWARD DROP [0:0]
```

Inspect and Activate Default Rules

View the currently-enforced `iptables` rules by running the command:

```
$ sudo iptables -nL --line-numbers
```

The command is analogous for `ip6tables`.

If the firewall does not appear to be active (i.e., no rules appear), activate it and ensure that it starts at boot by issuing the following commands (and analogously for `ip6tables`):

```
$ sudo service iptables restart
```

The default iptables rules are:

```
Chain INPUT (policy ACCEPT)
num  target     prot opt source       destination
1    ACCEPT     all  --  0.0.0.0/0    0.0.0.0/0    state RELATED,ESTABLISHED
2    ACCEPT     icmp --  0.0.0.0/0    0.0.0.0/0
3    ACCEPT     all  --  0.0.0.0/0    0.0.0.0/0
4    ACCEPT     tcp  --  0.0.0.0/0    0.0.0.0/0    state NEW tcp dpt:22
5    REJECT     all  --  0.0.0.0/0    0.0.0.0/0    reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT)
num  target     prot opt source       destination
1    REJECT     all  --  0.0.0.0/0    0.0.0.0/0    reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source       destination
```

The `ip6tables` default rules are essentially the same.

service_ip6tables_enabled

Verify ip6tables Enabled if Using IPv6

The `ip6tables` service can be enabled with the following command:

```
$ sudo systemctl enable ip6tables.service
```

service_iptables_enabled

Verify iptables Enabled

The `iptables` service can be enabled with the following command:

```
$ sudo systemctl enable iptables.service
```

set_ip6tables_default_rule

Set Default ip6tables Policy for Incoming Packets

To set the default policy to DROP (instead of ACCEPT) for the built-in INPUT chain which processes incoming packets, add or correct the following line in `/etc/sysconfig/ip6tables`:

```
:INPUT DROP [0:0]
```

If changes were required, reload the ip6tables rules:

```
$ sudo service ip6tables reload
```

firewalld

The dynamic firewall daemon `firewalld` provides a dynamically managed firewall with support for network “zones” to assign a level of trust to a network and its associated connections and interfaces. It has support for IPv4 and IPv6 firewall settings. It supports Ethernet bridges and has a separation of runtime and permanent configuration options. It also has an interface for services or applications to add firewall rules directly. A graphical configuration tool, `firewall-config`, is used to configure `firewalld`, which in turn uses `iptables` tool to communicate with `Netfilter` in the kernel which implements packet filtering. The firewall service provided by `firewalld` is dynamic rather than static because changes to the configuration can be made at anytime and are immediately implemented. There is no need to save or apply the changes. No unintended disruption of existing network connections occurs as no part of the firewall has to be reloaded.

Inspect and Activate Default firewalld Rules

Firewalls can be used to separate networks into different zones based on the level of trust the user has decided to place on the devices and traffic within that network. `NetworkManager` informs firewalld to which zone an interface belongs. An interface's assigned zone can be changed by `NetworkManager` or via the `firewall-config` tool. The zone settings in `/etc/firewalld/` are a range of preset settings which can be quickly applied to a network interface. These are the zones provided by firewalld sorted according to the default trust level of the zones from untrusted to trusted:

- `drop`
  Any incoming network packets are dropped, there is no reply. Only outgoing network connections are possible.
- `block`
  Any incoming network connections are rejected with an `icmp-host-prohibited` message for IPv4 and `icmp6-adm-prohibited` for IPv6. Only network connections initiated from within the system are possible.
- `public`
  For use in public areas. You do not trust the other computers on the network to not harm your computer. Only selected incoming connections are accepted.
- `external`
  For use on external networks with masquerading enabled especially for routers. You do not trust the other computers on the network to not harm your computer. Only selected incoming connections are accepted.
- `dmz`
  For computers in your demilitarized zone that are publicly-accessible with limited access to your internal network. Only selected incoming connections are accepted.
- `work`
  For use in work areas. You mostly trust the other computers on networks to not harm your computer. Only selected incoming connections are accepted.
- `home`
  For use in home areas. You mostly trust the other computers on networks to not harm your computer. Only selected incoming connections are accepted.
- `internal`
  For use on internal networks. You mostly trust the other computers on the networks to not harm your computer. Only selected incoming connections are accepted.
- `trusted`
  All network connections are accepted.

It is possible to designate one of these zones to be the default zone. When interface connections are added to `NetworkManager`, they are assigned to the default zone. On installation, the default zone in firewalld is set to be the public zone. To find out all the settings of a zone, for example the `public zone`, enter the following command as root:

```
# firewall-cmd --zone=public --list-all
```

Example output of this command might look like the following:

```
# firewall-cmd --zone=public --list-all
public
  interfaces:
  services: mdns dhcpv6-client ssh
  ports:
  forward-ports:
  icmp-blocks: source-quench
```

To view the network zones currently active, enter the following command as root:

```
# firewall-cmd --get-service
```

The following listing displays the result of this command on common Red Hat Enterprise Linux 8 system:

```
# firewall-cmd --get-service
amanda-client bacula bacula-client dhcp dhcpv6 dhcpv6-client dns ftp
high-availability http https imaps ipp ipp-client ipsec kerberos kpasswd
ldap ldaps libvirt libvirt-tls mdns mountd ms-wbt mysql nfs ntp openvpn
pmcd pmproxy pmwebapi pmwebapis pop3s postgresql proxy-dhcp radius rpc-bind
samba samba-client smtp ssh telnet tftp tftp-client transmission-client
vnc-server wbem-https
```

Finally to view the network zones that will be active after the next firewalld service reload, enter the following command as root:

```
# firewall-cmd --get-service --permanent
```

package_firewalld_installed

Install firewalld

The `firewalld` package can be installed with the following command:

```
$ sudo yum install firewalld
```

service_firewalld_enabled

Verify firewalld Enabled

The `firewalld` service can be enabled with the following command:

```
$ sudo systemctl enable firewalld.service
```

Strengthen the Default Ruleset

The default rules can be strengthened. The system scripts that activate the firewall rules expect them to be defined in configuration files under the `/etc/firewalld/services` and `/etc/firewalld/zones` directories.

The following recommendations describe how to strengthen the default ruleset configuration file. An alternative to editing this configuration file is to create a shell script that makes calls to the `firewall-cmd` program to load in rules under the `/etc/firewalld/services` and `/etc/firewalld/zones` directories.

Instructions apply to both unless otherwise noted. Language and address conventions for regular firewalld rules are used throughout this section.

set_firewalld_default_zone

Set Default firewalld Zone for Incoming Packets

To set the default zone to `drop` for the built-in default zone which processes incoming IPv4 and IPv6 packets, modify the following line in `/etc/firewalld/firewalld.conf` to be:

```
DefaultZone=drop
```

configure_firewalld_ports

Configure the Firewalld Ports

Configure the `firewalld` ports to allow approved services to have access to the system. To configure `firewalld` to open ports, run the following command:

```
$ sudo firewall-cmd --permanent --add-port=port_number/tcp
```

or

```
$ sudo firewall-cmd --permanent --add-port=service_name
```

Run the command list above for each of the ports listed below: To configure `firewalld` to allow access, run the following command(s): `firewall-cmd --permanent --add-service=ssh`

configure_firewalld_rate_limiting

Configure firewalld To Rate Limit Connections

Create a direct firewall rule to protect against DoS attacks with the following command:

```
$ sudo firewall-cmd --permanent --direct --add-rule ipv4 filter INPUT_direct 0 -p tcp -m limit --limit 25/minute --limit-burst
100  -j ACCEPT
```

IPv6

The system includes support for Internet Protocol version 6. A major and often-mentioned improvement over IPv4 is its enormous increase in the number of available addresses. Another important feature is its support for automatic configuration of many network settings.

Disable Support for IPv6 Unless Needed

Despite configuration that suggests support for IPv6 has been disabled, link-local IPv6 address auto-configuration occurs even when only an IPv4 address is assigned. The only way to effectively prevent execution of the IPv6 networking stack is to instruct the system not to activate the IPv6 kernel module.

network_ipv6_disable_interfaces

Disable Interface Usage of IPv6

To disable interface usage of IPv6, add or correct the following lines in `/etc/sysconfig/network`:

```
NETWORKING_IPV6=no
IPV6INIT=no
```

kernel_module_ipv6_option_disabled

Disable IPv6 Networking Support Automatic Loading

To prevent the IPv6 kernel module (`ipv6`) from binding to the IPv6 networking stack, add the following line to `/etc/modprobe.d/disabled.conf` (or another file in `/etc/modprobe.d`):

```
options ipv6 disable=1
```

This permits the IPv6 module to be loaded (and thus satisfy other modules that depend on it), while disabling support for the IPv6 protocol.

sysctl_net_ipv6_conf_all_disable_ipv6

Disable IPv6 Networking Support Automatic Loading

To disable support for (`ipv6`) add the following line to `/etc/sysctl.d/ipv6.conf` (or another file in `/etc/sysctl.d`):

```
net.ipv6.conf.all.disable_ipv6 = 1
```

This disables IPv6 on all network interfaces as other services and system functionality require the IPv6 stack loaded to work.

network_ipv6_disable_rpc

Disable Support for RPC IPv6

RPC services for NFSv4 try to load transport modules for `udp6` and `tcp6` by default, even if IPv6 has been disabled in `/etc/modprobe.d`. To prevent RPC services such as `rpc.mountd` from attempting to start IPv6 network listeners, remove or comment out the following two lines in `/etc/netconfig`:

```
udp6       tpi_clts      v     inet6    udp     -       -
tcp6       tpi_cots_ord  v     inet6    tcp     -       -
```

Configure IPv6 Settings if Necessary

A major feature of IPv6 is the extent to which systems implementing it can automatically configure their networking devices using information from the network. From a security perspective, manually configuring important configuration information is preferable to accepting it from the network in an unauthenticated fashion.

Disable Automatic Configuration

Disable the system's acceptance of router advertisements and redirects by adding or correcting the following line in `/etc/sysconfig/network` (note that this does not disable sending router solicitations):

```
IPV6_AUTOCONF=no
```

sysctl_net_ipv6_conf_all_accept_source_route

Configure Kernel Parameter for Accepting IPv6 Source-Routed Packets for All Interfaces

To set the runtime status of the `net.ipv6.conf.all.accept_source_route` kernel parameter, run the following command:

```
$ sudo sysctl -w net.ipv6.conf.all.accept_source_route=0
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
net.ipv6.conf.all.accept_source_route = 0
```

sysctl_net_ipv6_conf_default_accept_ra

Configure Accepting IPv6 Router Advertisements by Default

To set the runtime status of the `net.ipv6.conf.default.accept_ra` kernel parameter, run the following command:

```
$ sudo sysctl -w net.ipv6.conf.default.accept_ra=0
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
net.ipv6.conf.default.accept_ra = 0
```

sysctl_net_ipv6_conf_all_accept_ra

Configure Accepting IPv6 Router Advertisements on All Interfaces

To set the runtime status of the `net.ipv6.conf.all.accept_ra` kernel parameter, run the following command:

```
$ sudo sysctl -w net.ipv6.conf.all.accept_ra=0
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
net.ipv6.conf.all.accept_ra = 0
```

sysctl_net_ipv6_conf_all_accept_redirects

Configure Accepting IPv6 Redirects on All Interfaces

To set the runtime status of the `net.ipv6.conf.all.accept_redirects` kernel parameter, run the following command:

```
$ sudo sysctl -w net.ipv6.conf.all.accept_redirects=0
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
net.ipv6.conf.all.accept_redirects = 0
```

sysctl_net_ipv6_conf_default_accept_redirects

Configure Accepting IPv6 Redirects By Default

To set the runtime status of the `net.ipv6.conf.default.accept_redirects` kernel parameter, run the following command:

```
$ sudo sysctl -w net.ipv6.conf.default.accept_redirects=0
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
net.ipv6.conf.default.accept_redirects = 0
```

sysctl_net_ipv6_conf_default_accept_source_route

Configure Kernel Parameter for Accepting Source-Routed Packets for Interfaces By Default

To set the runtime status of the `net.ipv6.conf.default.accept_source_route` kernel parameter, run the following command:

```
$ sudo sysctl -w net.ipv6.conf.default.accept_source_route=0
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
net.ipv6.conf.default.accept_source_route = 0
```

sysctl_net_ipv6_conf_all_forwarding

Disable Kernel Parameter for IPv6 Forwarding

To set the runtime status of the `net.ipv6.conf.all.forwarding` kernel parameter, run the following command:

```
$ sudo sysctl -w net.ipv6.conf.all.forwarding=0
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
net.ipv6.conf.all.forwarding = 0
```

Limit Network-Transmitted Configuration if Using Static IPv6 Addresses

To limit the configuration information requested from other systems and accepted from the network on a system that uses statically-configured IPv6 addresses, add the following lines to `/etc/sysctl.conf`:

```
net.ipv6.conf.default.router_solicitations = 0
net.ipv6.conf.default.accept_ra_rtr_pref = 0
net.ipv6.conf.default.accept_ra_pinfo = 0
net.ipv6.conf.default.accept_ra_defrtr = 0
net.ipv6.conf.default.autoconf = 0
net.ipv6.conf.default.dad_transmits = 0
net.ipv6.conf.default.max_addresses = 1
```

The `router_solicitations` setting determines how many router solicitations are sent when bringing up the interface. If addresses are statically assigned, there is no need to send any solicitations.

The `accept_ra_pinfo` setting controls whether the system will accept prefix info from the router.

The `accept_ra_defrtr` setting controls whether the system will accept Hop Limit settings from a router advertisement. Setting it to 0 prevents a router from changing your default IPv6 Hop Limit for outgoing packets.

The `autoconf` setting controls whether router advertisements can cause the system to assign a global unicast address to an interface.

The `dad_transmits` setting determines how many neighbor solicitations to send out per address (global and link-local) when bringing up an interface to ensure the desired address is unique on the network.

The `max_addresses` setting determines how many global unicast IPv6 addresses can be assigned to each interface. The default is 16, but it should be set to exactly the number of statically configured global addresses required.

network_ipv6_static_address

Manually Assign Global IPv6 Address

To manually assign an IP address for an interface, edit the file `/etc/sysconfig/network-scripts/ifcfg-interface`. Add or correct the following line (substituting the correct IPv6 address):

```
IPV6ADDR=2001:0DB8::ABCD/64
```

Manually assigning an IP address is preferable to accepting one from routers or from the network otherwise. The example address here is an IPv6 address reserved for documentation purposes, as defined by RFC3849.

network_ipv6_privacy_extensions

Use Privacy Extensions for Address

To introduce randomness into the automatic generation of IPv6 addresses, add or correct the following line in `/etc/sysconfig/network-scripts/ifcfg-interface`:

```
IPV6_PRIVACY=rfc3041
```

Automatically-generated IPv6 addresses are based on the underlying hardware (e.g. Ethernet) address, and so it becomes possible to track a piece of hardware over its lifetime using its traffic. If it is important for a system's IP address to not trivially reveal its hardware address, this setting should be applied.

network_ipv6_default_gateway

Manually Assign IPv6 Router Address

Edit the file `/etc/sysconfig/network-scripts/ifcfg-interface`, and add or correct the following line (substituting your gateway IP as appropriate):

```
IPV6_DEFAULTGW=2001:0DB8::0001
```

Router addresses should be manually set and not accepted via any auto-configuration or router advertisement.

Wireless Networking

Wireless networking, such as 802.11 (WiFi) and Bluetooth, can present a security risk to sensitive or classified systems and networks. Wireless networking hardware is much more likely to be included in laptop or portable systems than in desktops or servers.

Removal of hardware provides the greatest assurance that the wireless capability remains disabled. Acquisition policies often include provisions to prevent the purchase of equipment that will be used in sensitive spaces and includes wireless capabilities. If it is impractical to remove the wireless hardware, and policy permits the device to enter sensitive spaces as long as wireless is disabled, efforts should instead focus on disabling wireless capability via software.

Disable Wireless Through Software Configuration

If it is impossible to remove the wireless hardware from the device in question, disable as much of it as possible through software. The following methods can disable software support for wireless networking, but note that these methods do not prevent malicious software or careless users from re-activating the devices.

service_bluetooth_disabled

Disable Bluetooth Service

The `bluetooth` service can be disabled with the following command:

```
$ sudo systemctl disable bluetooth.service
```

```
$ sudo service bluetooth stop
```

kernel_module_bluetooth_disabled

Disable Bluetooth Kernel Modules

The kernel's module loading system can be configured to prevent loading of the Bluetooth module. Add the following to the appropriate `/etc/modprobe.d` configuration file to prevent the loading of the Bluetooth module:

```
install bluetooth /bin/true
```

wireless_disable_interfaces

Deactivate Wireless Network Interfaces

Deactivating wireless network interfaces should prevent normal usage of the wireless capability.

Configure the system to disable all wireless network interfaces with the following command:

```
$ sudo nmcli radio wifi off
```

wireless_disable_in_bios

Disable WiFi or Bluetooth in BIOS

Some machines that include built-in wireless support offer the ability to disable the device through the BIOS. This is hardware-specific; consult your hardware manual or explore the BIOS setup during boot.

Kernel Parameters Which Affect Networking

The `sysctl` utility is used to set parameters which affect the operation of the Linux kernel. Kernel parameters which affect networking and have security implications are described here.

Network Related Kernel Runtime Parameters for Hosts and Routers

Certain kernel parameters should be set for systems which are acting as either hosts or routers to improve the system's ability defend against certain types of IPv4 protocol attacks.

sysctl_net_ipv4_conf_default_accept_redirects

Configure Kernel Parameter for Accepting ICMP Redirects By Default

To set the runtime status of the `net.ipv4.conf.default.accept_redirects` kernel parameter, run the following command:

```
$ sudo sysctl -w net.ipv4.conf.default.accept_redirects=0
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
net.ipv4.conf.default.accept_redirects = 0
```

sysctl_net_ipv4_conf_default_rp_filter

Configure Kernel Parameter to Use Reverse Path Filtering by Default

To set the runtime status of the `net.ipv4.conf.default.rp_filter` kernel parameter, run the following command:

```
$ sudo sysctl -w net.ipv4.conf.default.rp_filter=1
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
net.ipv4.conf.default.rp_filter = 1
```

sysctl_net_ipv4_conf_all_rp_filter

Configure Kernel Parameter to Use Reverse Path Filtering for All Interfaces

To set the runtime status of the `net.ipv4.conf.all.rp_filter` kernel parameter, run the following command:

```
$ sudo sysctl -w net.ipv4.conf.all.rp_filter=1
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
net.ipv4.conf.all.rp_filter = 1
```

sysctl_net_ipv4_conf_all_secure_redirects

Configure Kernel Parameter for Accepting Secure Redirects for All Interfaces

To set the runtime status of the `net.ipv4.conf.all.secure_redirects` kernel parameter, run the following command:

```
$ sudo sysctl -w net.ipv4.conf.all.secure_redirects=0
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
net.ipv4.conf.all.secure_redirects = 0
```

sysctl_net_ipv4_conf_default_log_martians

Configure Kernel Parameter to Log Martian Packets By Default

To set the runtime status of the `net.ipv4.conf.default.log_martians` kernel parameter, run the following command:

```
$ sudo sysctl -w net.ipv4.conf.default.log_martians=1
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
net.ipv4.conf.default.log_martians = 1
```

sysctl_net_ipv4_conf_default_accept_source_route

Configure Kernel Parameter for Accepting Source-Routed Packets By Default

To set the runtime status of the `net.ipv4.conf.default.accept_source_route` kernel parameter, run the following command:

```
$ sudo sysctl -w net.ipv4.conf.default.accept_source_route=0
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
net.ipv4.conf.default.accept_source_route = 0
```

sysctl_net_ipv4_conf_all_accept_source_route

Configure Kernel Parameter for Accepting IPv4 Source-Routed Packets for All Interfaces

To set the runtime status of the `net.ipv4.conf.all.accept_source_route` kernel parameter, run the following command:

```
$ sudo sysctl -w net.ipv4.conf.all.accept_source_route=0
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
net.ipv4.conf.all.accept_source_route = 0
```

sysctl_net_ipv4_tcp_syncookies

Configure Kernel Parameter to Use TCP Syncookies

To set the runtime status of the `net.ipv4.tcp_syncookies` kernel parameter, run the following command:

```
$ sudo sysctl -w net.ipv4.tcp_syncookies=1
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
net.ipv4.tcp_syncookies = 1
```

sysctl_net_ipv4_conf_all_log_martians

Configure Kernel Parameter to Log Martian Packets

To set the runtime status of the `net.ipv4.conf.all.log_martians` kernel parameter, run the following command:

```
$ sudo sysctl -w net.ipv4.conf.all.log_martians=1
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
net.ipv4.conf.all.log_martians = 1
```

sysctl_net_ipv4_icmp_ignore_bogus_error_responses

Configure Kernel Parameter to Ignore Bogus ICMP Error Responses

To set the runtime status of the `net.ipv4.icmp_ignore_bogus_error_responses` kernel parameter, run the following command:

```
$ sudo sysctl -w net.ipv4.icmp_ignore_bogus_error_responses=1
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
net.ipv4.icmp_ignore_bogus_error_responses = 1
```

sysctl_net_ipv4_conf_default_secure_redirects

Configure Kernel Parameter for Accepting Secure Redirects By Default

To set the runtime status of the `net.ipv4.conf.default.secure_redirects` kernel parameter, run the following command:

```
$ sudo sysctl -w net.ipv4.conf.default.secure_redirects=0
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
net.ipv4.conf.default.secure_redirects = 0
```

sysctl_net_ipv4_icmp_echo_ignore_broadcasts

Configure Kernel Parameter to Ignore ICMP Broadcast Echo Requests

To set the runtime status of the `net.ipv4.icmp_echo_ignore_broadcasts` kernel parameter, run the following command:

```
$ sudo sysctl -w net.ipv4.icmp_echo_ignore_broadcasts=1
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
net.ipv4.icmp_echo_ignore_broadcasts = 1
```

sysctl_net_ipv4_conf_all_accept_redirects

Configure Kernel Parameter for Accepting ICMP Redirects for All Interfaces

To set the runtime status of the `net.ipv4.conf.all.accept_redirects` kernel parameter, run the following command:

```
$ sudo sysctl -w net.ipv4.conf.all.accept_redirects=0
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
net.ipv4.conf.all.accept_redirects = 0
```

Network Parameters for Hosts Only

If the system is not going to be used as a router, then setting certain kernel parameters ensure that the host will not perform routing of network traffic.

sysctl_net_ipv4_conf_default_send_redirects

Disable Kernel Parameter for Sending ICMP Redirects by Default

To set the runtime status of the `net.ipv4.conf.default.send_redirects` kernel parameter, run the following command:

```
$ sudo sysctl -w net.ipv4.conf.default.send_redirects=0
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
net.ipv4.conf.default.send_redirects = 0
```

sysctl_net_ipv4_conf_all_send_redirects

Disable Kernel Parameter for Sending ICMP Redirects for All Interfaces

To set the runtime status of the `net.ipv4.conf.all.send_redirects` kernel parameter, run the following command:

```
$ sudo sysctl -w net.ipv4.conf.all.send_redirects=0
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
net.ipv4.conf.all.send_redirects = 0
```

sysctl_net_ipv4_ip_forward

Disable Kernel Parameter for IP Forwarding

To set the runtime status of the `net.ipv4.ip_forward` kernel parameter, run the following command:

```
$ sudo sysctl -w net.ipv4.ip_forward=0
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
net.ipv4.ip_forward = 0
```

Transport Layer Security Support

Support for Transport Layer Security (TLS), and its predecessor, the Secure Sockets Layer (SSL), is included in Red Hat Enterprise Linux in the OpenSSL software (RPM package `openssl`). TLS provides encrypted and authenticated network communications, and many network services include support for it. TLS or SSL can be leveraged to avoid any plaintext transmission of sensitive data. For information on how to use OpenSSL, see **http://www.openssl.org/docs/**. Information on FIPS validation of OpenSSL is available at **http://www.openssl.org/docs/fips.html** and **http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140val-all.htm**.

network_disable_ddns_interfaces

Disable Client Dynamic DNS Updates

Dynamic DNS allows clients to dynamically update their own DNS records. The updates are transmitted by unencrypted means which can reveal information to a potential malicious user. If the system does not require Dynamic DNS, remove all `DHCP_HOSTNAME` references from the `/etc/sysconfig/network-scripts/ifcfg-interface` scripts. If `dhclient` is used, remove all `send host-name hostname` references from the `/etc/dhclient.conf` configuration file and/or any reference from the `/etc/dhcp` directory.

network_configure_name_resolution

Configure Multiple DNS Servers in /etc/resolv.conf

Multiple Domain Name System (DNS) Servers should be configured in `/etc/resolv.conf`. This provides redundant name resolution services in the event that a domain server crashes. To configure the system to contain as least `2` DNS servers, add a corresponding `nameserver ip_address` entry in `/etc/resolv.conf` for each DNS server where *ip_address* is the IP address of a valid DNS server. For example:

```
search example.com
nameserver 192.168.0.1
nameserver 192.168.0.2
```

network_disable_zeroconf

Disable Zeroconf Networking

Zeroconf networking allows the system to assign itself an IP address and engage in IP communication without a statically-assigned address or even a DHCP server. Automatic address assignment via Zeroconf (or DHCP) is not recommended. To disable Zeroconf automatic route assignment in the 169.254.0.0 subnet, add or correct the following line in `/etc/sysconfig/network`:

```
NOZEROCONF=yes
```

network_sniffer_disabled

Ensure System is Not Acting as a Network Sniffer

The system should not be acting as a network sniffer, which can capture all traffic on the network to which it is connected. Run the following to determine if any interface is running in promiscuous mode:

```
$ ip link | grep PROMISC
```

Set Boot Loader Password

During the boot process, the boot loader is responsible for starting the execution of the kernel and passing options to it. The boot loader allows for the selection of different kernels - possibly on different partitions or media. The default Red Hat Enterprise Linux boot loader for x86 systems is called GRUB. Options it can pass to the kernel include *single-user mode*, which provides root access without any authentication, and the ability to disable SELinux. To prevent local users from modifying the boot parameters and endangering security, protect the boot loader configuration with a password and ensure its configuration file's permissions are set properly.

SELinux

SELinux is a feature of the Linux kernel which can be used to guard against misconfigured or compromised programs. SELinux enforces the idea that programs should be limited in what files they can access and what actions they can take.

The default SELinux policy, as configured on Red Hat Enterprise Linux 8, has been sufficiently developed and debugged that it should be usable on almost any system with minimal configuration and a small amount of system administrator training. This policy prevents system services - including most of the common network-visible services such as mail servers, FTP servers, and DNS servers - from accessing files which those services have no valid reason to access. This action alone prevents a huge amount of possible damage from network attacks against services, from trojaned software, and so forth.

This guide recommends that SELinux be enabled using the default (targeted) policy on every Red Hat Enterprise Linux 8 system, unless that system has unusual requirements which make a stronger policy appropriate.

SELinux - Booleans

Enable or Disable runtime customization of SELinux system policies without having to reload or recompile the SELinux policy.

sebool_virt_use_usb

Disable the virt_use_usb SELinux Boolean

By default, the SELinux boolean `virt_use_usb` is enabled. This setting should be disabled. To disable the `virt_use_usb` SELinux boolean, run the following command:

```
$ sudo setsebool -P virt_use_usb off
```

sebool_ftpd_anon_write

Disable the ftpd_anon_write SELinux Boolean

By default, the SELinux boolean `ftpd_anon_write` is disabled. If this setting is enabled, it should be disabled. To disable the `ftpd_anon_write` SELinux boolean, run the following command:

```
$ sudo setsebool -P ftpd_anon_write off
```

sebool_dbadm_manage_user_files

Disable the dbadm_manage_user_files SELinux Boolean

By default, the SELinux boolean `dbadm_manage_user_files` is disabled. If this setting is enabled, it should be disabled. To disable the `dbadm_manage_user_files` SELinux boolean, run the following command:

```
$ sudo setsebool -P dbadm_manage_user_files off
```

sebool_rsync_anon_write

Disable the rsync_anon_write SELinux Boolean

By default, the SELinux boolean `rsync_anon_write` is disabled. If this setting is enabled, it should be disabled. To disable the `rsync_anon_write` SELinux boolean, run the following command:

```
$ sudo setsebool -P rsync_anon_write off
```

sebool_sanlock_use_samba

Disable the sanlock_use_samba SELinux Boolean

By default, the SELinux boolean `sanlock_use_samba` is disabled. If this setting is enabled, it should be disabled. To disable the `sanlock_use_samba` SELinux boolean, run the following command:

```
$ sudo setsebool -P sanlock_use_samba off
```

sebool_tor_bind_all_unreserved_ports

Disable the tor_bind_all_unreserved_ports SELinux Boolean

By default, the SELinux boolean `tor_bind_all_unreserved_ports` is disabled. If this setting is enabled, it should be disabled. To disable the `tor_bind_all_unreserved_ports` SELinux boolean, run the following command:

```
$ sudo setsebool -P tor_bind_all_unreserved_ports off
```

sebool_virt_sandbox_use_all_caps

Disable the virt_sandbox_use_all_caps SELinux Boolean

By default, the SELinux boolean `virt_sandbox_use_all_caps` is enabled. This setting is disabled as containers should not run with privileges. To disable the `virt_sandbox_use_all_caps` SELinux boolean, run the following command:

```
$ sudo setsebool -P virt_sandbox_use_all_caps off
```

sebool_mplayer_execstack

Disable the mplayer_execstack SELinux Boolean

By default, the SELinux boolean `mplayer_execstack` is disabled. If this setting is enabled, it should be disabled. To disable the `mplayer_execstack` SELinux boolean, run the following command:

```
$ sudo setsebool -P mplayer_execstack off
```

sebool_varnishd_connect_any

Disable the varnishd_connect_any SELinux Boolean

By default, the SELinux boolean `varnishd_connect_any` is disabled. If this setting is enabled, it should be disabled. To disable the `varnishd_connect_any` SELinux boolean, run the following command:

```
$ sudo setsebool -P varnishd_connect_any off
```

sebool_secadm_exec_content

Enable the secadm_exec_content SELinux Boolean

By default, the SELinux boolean `secadm_exec_content` is enabled. If this setting is disabled, it should be enabled. To enable the `secadm_exec_content` SELinux boolean, run the following command:

```
$ sudo setsebool -P secadm_exec_content on
```

sebool_mozilla_read_content

Disable the mozilla_read_content SELinux Boolean

By default, the SELinux boolean `mozilla_read_content` is disabled. If this setting is enabled, it should be disabled. To disable the `mozilla_read_content` SELinux boolean, run the following command:

```
$ sudo setsebool -P mozilla_read_content off
```

sebool_ssh_keysign

Disable the ssh_keysign SELinux Boolean

By default, the SELinux boolean `ssh_keysign` is disabled. If this setting is enabled, it should be disabled. To disable the `ssh_keysign` SELinux boolean, run the following command:

```
$ sudo setsebool -P ssh_keysign off
```

sebool_gpg_web_anon_write

Disable the gpg_web_anon_write SELinux Boolean

By default, the SELinux boolean `gpg_web_anon_write` is disabled. If this setting is enabled, it should be disabled. To disable the `gpg_web_anon_write` SELinux boolean, run the following command:

```
$ sudo setsebool -P gpg_web_anon_write off
```

sebool_mozilla_plugin_use_bluejeans

Disable the mozilla_plugin_use_bluejeans SELinux Boolean

By default, the SELinux boolean `mozilla_plugin_use_bluejeans` is disabled. If this setting is enabled, it should be disabled. To disable the `mozilla_plugin_use_bluejeans` SELinux boolean, run the following command:

```
$ sudo setsebool -P mozilla_plugin_use_bluejeans off
```

sebool_selinuxuser_tcp_server

Disable the selinuxuser_tcp_server SELinux Boolean

By default, the SELinux boolean `selinuxuser_tcp_server` is disabled. If this setting is enabled, it should be disabled. To disable the `selinuxuser_tcp_server` SELinux boolean, run the following command:

```
$ sudo setsebool -P selinuxuser_tcp_server off
```

sebool_openvpn_enable_homedirs

Disable the openvpn_enable_homedirs SELinux Boolean

By default, the SELinux boolean `openvpn_enable_homedirs` is enabled. This setting should be disabled. To disable the `openvpn_enable_homedirs` SELinux boolean, run the following command:

```
$ sudo setsebool -P openvpn_enable_homedirs off
```

sebool_gluster_anon_write

Disable the gluster_anon_write SELinux Boolean

By default, the SELinux boolean `gluster_anon_write` is disabled. If this setting is enabled, it should be disabled. To disable the `gluster_anon_write` SELinux boolean, run the following command:

```
$ sudo setsebool -P gluster_anon_write off
```

sebool_git_system_use_cifs

Disable the git_system_use_cifs SELinux Boolean

By default, the SELinux boolean `git_system_use_cifs` is disabled. If this setting is enabled, it should be disabled. To disable the `git_system_use_cifs` SELinux boolean, run the following command:

```
$ sudo setsebool -P git_system_use_cifs off
```

sebool_tmpreaper_use_samba

Disable the tmpreaper_use_samba SELinux Boolean

By default, the SELinux boolean `tmpreaper_use_samba` is disabled. If this setting is enabled, it should be disabled. To disable the `tmpreaper_use_samba` SELinux boolean, run the following command:

```
$ sudo setsebool -P tmpreaper_use_samba off
```

sebool_postgresql_can_rsync

Disable the postgresql_can_rsync SELinux Boolean

By default, the SELinux boolean `postgresql_can_rsync` is disabled. If this setting is enabled, it should be disabled. To disable the `postgresql_can_rsync` SELinux boolean, run the following command:

```
$ sudo setsebool -P postgresql_can_rsync off
```

sebool_named_tcp_bind_http_port

Disable the named_tcp_bind_http_port SELinux Boolean

By default, the SELinux boolean `named_tcp_bind_http_port` is disabled. If this setting is enabled, it should be disabled. To disable the `named_tcp_bind_http_port` SELinux boolean, run the following command:

```
$ sudo setsebool -P named_tcp_bind_http_port off
```

sebool_selinuxuser_use_ssh_chroot

Disable the selinuxuser_use_ssh_chroot SELinux Boolean

By default, the SELinux boolean `selinuxuser_use_ssh_chroot` is disabled. If this setting is enabled, it should be disabled. To disable the `selinuxuser_use_ssh_chroot` SELinux boolean, run the following command:

```
$ sudo setsebool -P selinuxuser_use_ssh_chroot off
```

sebool_boinc_execmem

Disable the boinc_execmem SELinux Boolean

By default, the SELinux boolean `boinc_execmem` is enabled. This setting should be disabled. To disable the `boinc_execmem` SELinux boolean, run the following command:

```
$ sudo setsebool -P boinc_execmem off
```

sebool_polipo_session_bind_all_unreserved_ports

Disable the polipo_session_bind_all_unreserved_ports SELinux Boolean

By default, the SELinux boolean `polipo_session_bind_all_unreserved_ports` is disabled. If this setting is enabled, it should be disabled. To disable the `polipo_session_bind_all_unreserved_ports` SELinux boolean, run the following command:

```
$ sudo setsebool -P polipo_session_bind_all_unreserved_ports off
```

sebool_httpd_can_check_spam

Disable the httpd_can_check_spam SELinux Boolean

By default, the SELinux boolean `httpd_can_check_spam` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_can_check_spam` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_can_check_spam off
```

sebool_saslauthd_read_shadow

Disable the saslauthd_read_shadow SELinux Boolean

By default, the SELinux boolean `saslauthd_read_shadow` is disabled. If this setting is enabled, it should be disabled. To disable the `saslauthd_read_shadow` SELinux boolean, run the following command:

```
$ sudo setsebool -P saslauthd_read_shadow off
```

sebool_zabbix_can_network

Disable the zabbix_can_network SELinux Boolean

By default, the SELinux boolean `zabbix_can_network` is disabled. If this setting is enabled, it should be disabled. To disable the `zabbix_can_network` SELinux boolean, run the following command:

```
$ sudo setsebool -P zabbix_can_network off
```

sebool_samba_share_nfs

Disable the samba_share_nfs SELinux Boolean

By default, the SELinux boolean `samba_share_nfs` is disabled. If this setting is enabled, it should be disabled. To disable the `samba_share_nfs` SELinux boolean, run the following command:

```
$ sudo setsebool -P samba_share_nfs off
```

sebool_postfix_local_write_mail_spool

Enable the postfix_local_write_mail_spool SELinux Boolean

By default, the SELinux boolean `postfix_local_write_mail_spool` is enabled. If this setting is disabled, it should be enabled as it allows Postfix to write to the mail spool directories. To enable the `postfix_local_write_mail_spool` SELinux boolean, run the following command:

```
$ sudo setsebool -P postfix_local_write_mail_spool on
```

sebool_global_ssp

Disable the global_ssp SELinux Boolean

By default, the SELinux boolean `global_ssp` is disabled. If this setting is enabled, it should be disabled. To disable the `global_ssp` SELinux boolean, run the following command:

```
$ sudo setsebool -P global_ssp off
```

sebool_exim_read_user_files

Disable the exim_read_user_files SELinux Boolean

By default, the SELinux boolean `exim_read_user_files` is disabled. If this setting is enabled, it should be disabled. To disable the `exim_read_user_files` SELinux boolean, run the following command:

```
$ sudo setsebool -P exim_read_user_files off
```

sebool_use_nfs_home_dirs

Disable the use_nfs_home_dirs SELinux Boolean

By default, the SELinux boolean `use_nfs_home_dirs` is disabled. If this setting is enabled, it should be disabled. To disable the `use_nfs_home_dirs` SELinux boolean, run the following command:

```
$ sudo setsebool -P use_nfs_home_dirs off
```

sebool_samba_share_fusefs

Disable the samba_share_fusefs SELinux Boolean

By default, the SELinux boolean `samba_share_fusefs` is disabled. If this setting is enabled, it should be disabled. To disable the `samba_share_fusefs` SELinux boolean, run the following command:

```
$ sudo setsebool -P samba_share_fusefs off
```

sebool_awstats_purge_apache_log_files

Disable the awstats_purge_apache_log_files SELinux Boolean

By default, the SELinux boolean `awstats_purge_apache_log_files` is disabled. If this setting is enabled, it should be disabled. To disable the `awstats_purge_apache_log_files` SELinux boolean, run the following command:

```
$ sudo setsebool -P awstats_purge_apache_log_files off
```

sebool_httpd_mod_auth_ntlm_winbind

Disable the httpd_mod_auth_ntlm_winbind SELinux Boolean

By default, the SELinux boolean `httpd_mod_auth_ntlm_winbind` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_mod_auth_ntlm_winbind` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_mod_auth_ntlm_winbind off
```

sebool_tor_can_network_relay

Disable the tor_can_network_relay SELinux Boolean

By default, the SELinux boolean `tor_can_network_relay` is disabled. If this setting is enabled, it should be disabled. To disable the `tor_can_network_relay` SELinux boolean, run the following command:

```
$ sudo setsebool -P tor_can_network_relay off
```

sebool_selinuxuser_rw_noexattrfile

Disable the selinuxuser_rw_noexattrfile SELinux Boolean

By default, the SELinux boolean `selinuxuser_rw_noexattrfile` is enabled. This setting should be disabled as users should not be able to read/write files on filesystems that do not have extended attributes e.g. FAT, CDROM, FLOPPY, etc. To disable the `selinuxuser_rw_noexattrfile` SELinux boolean, run the following command:

```
$ sudo setsebool -P selinuxuser_rw_noexattrfile off
```

sebool_antivirus_can_scan_system

Enable the antivirus_can_scan_system SELinux Boolean

By default, the SELinux boolean `antivirus_can_scan_system` is disabled. This setting should be enabled as it allows antivirus programs to read non-security files on a system. To enable the `antivirus_can_scan_system` SELinux boolean, run the following command:

```
$ sudo setsebool -P antivirus_can_scan_system on
```

sebool_selinuxuser_ping

Enable the selinuxuser_ping SELinux Boolean

By default, the SELinux boolean `selinuxuser_ping` is enabled. If this setting is disabled, it should be enabled as it allows confined users to use ping and traceroute which is helpful for network troubleshooting. To enable the `selinuxuser_ping` SELinux boolean, run the following command:

```
$ sudo setsebool -P selinuxuser_ping on
```

sebool_httpd_run_ipa

Disable the httpd_run_ipa SELinux Boolean

By default, the SELinux boolean `httpd_run_ipa` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_run_ipa` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_run_ipa off
```

sebool_cluster_can_network_connect

Disable the cluster_can_network_connect SELinux Boolean

By default, the SELinux boolean `cluster_can_network_connect` is disabled. If this setting is enabled, it should be disabled. To disable the `cluster_can_network_connect` SELinux boolean, run the following command:

```
$ sudo setsebool -P cluster_can_network_connect off
```

sebool_lsmd_plugin_connect_any

Disable the lsmd_plugin_connect_any SELinux Boolean

By default, the SELinux boolean `lsmd_plugin_connect_any` is disabled. If this setting is enabled, it should be disabled. To disable the `lsmd_plugin_connect_any` SELinux boolean, run the following command:

```
$ sudo setsebool -P lsmd_plugin_connect_any off
```

sebool_gluster_export_all_rw

Configure the gluster_export_all_rw SELinux Boolean

By default, the SELinux boolean `gluster_export_all_rw` is enabled. If `GlusterFS` is in use, this setting should be enabled. Otherwise, disable it. To disable the `gluster_export_all_rw` SELinux boolean, run the following command:

```
$ sudo setsebool -P gluster_export_all_rw off
```

sebool_tftp_home_dir

Disable the tftp_home_dir SELinux Boolean

By default, the SELinux boolean `tftp_home_dir` is disabled. If this setting is enabled, it should be disabled. To disable the `tftp_home_dir` SELinux boolean, run the following command:

```
$ sudo setsebool -P tftp_home_dir off
```

sebool_selinuxuser_execmod

Enable the selinuxuser_execmod SELinux Boolean

By default, the SELinux boolean `selinuxuser_execmod` is enabled. If this setting is disabled, it should be enabled. To enable the `selinuxuser_execmod` SELinux boolean, run the following command:

```
$ sudo setsebool -P selinuxuser_execmod on
```

sebool_httpd_can_network_connect

Disable the httpd_can_network_connect SELinux Boolean

By default, the SELinux boolean `httpd_can_network_connect` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_can_network_connect` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_can_network_connect off
```

sebool_domain_fd_use

Enable the domain_fd_use SELinux Boolean

By default, the SELinux boolean `domain_fd_use` is enabled. If this setting is disabled, it should be enabled. To enable the `domain_fd_use` SELinux boolean, run the following command:

```
$ sudo setsebool -P domain_fd_use on
```

sebool_httpd_tty_comm

Disable the httpd_tty_comm SELinux Boolean

By default, the SELinux boolean `httpd_tty_comm` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_tty_comm` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_tty_comm off
```

sebool_httpd_enable_ftp_server

Disable the httpd_enable_ftp_server SELinux Boolean

By default, the SELinux boolean `httpd_enable_ftp_server` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_enable_ftp_server` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_enable_ftp_server off
```

sebool_xdm_sysadm_login

Disable the xdm_sysadm_login SELinux Boolean

By default, the SELinux boolean `xdm_sysadm_login` is disabled. If this setting is enabled, it should be disabled. To disable the `xdm_sysadm_login` SELinux boolean, run the following command:

```
$ sudo setsebool -P xdm_sysadm_login off
```

sebool_abrt_anon_write

Disable the abrt_anon_write SELinux Boolean

By default, the SELinux boolean `abrt_anon_write` is disabled. If this setting is enabled, it should be disabled. To disable the `abrt_anon_write` SELinux boolean, run the following command:

```
$ sudo setsebool -P abrt_anon_write off
```

sebool_smartmon_3ware

Disable the smartmon_3ware SELinux Boolean

By default, the SELinux boolean `smartmon_3ware` is disabled. If this setting is enabled, it should be disabled. To disable the `smartmon_3ware` SELinux boolean, run the following command:

```
$ sudo setsebool -P smartmon_3ware off
```

sebool_cron_can_relabel

Disable the cron_can_relabel SELinux Boolean

By default, the SELinux boolean `cron_can_relabel` is disabled. If this setting is enabled, it should be disabled. To disable the `cron_can_relabel` SELinux boolean, run the following command:

```
$ sudo setsebool -P cron_can_relabel off
```

sebool_abrt_upload_watch_anon_write

Disable the abrt_upload_watch_anon_write SELinux Boolean

By default, the SELinux boolean `abrt_upload_watch_anon_write` is enabled. This setting should be disabled as it allows the Automatic Bug Report Tool (ABRT) to modify public files used for public file transfer services. To disable the `abrt_upload_watch_anon_write` SELinux boolean, run the following command:

```
$ sudo setsebool -P abrt_upload_watch_anon_write off
```

sebool_httpd_use_cifs

Disable the httpd_use_cifs SELinux Boolean

By default, the SELinux boolean `httpd_use_cifs` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_use_cifs` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_use_cifs off
```

sebool_entropyd_use_audio

Disable the entropyd_use_audio SELinux Boolean

By default, the SELinux boolean `entropyd_use_audio` is enabled. This setting should be disabled as it uses audit input to generate entropy. To disable the `entropyd_use_audio` SELinux boolean, run the following command:

```
$ sudo setsebool -P entropyd_use_audio off
```

sebool_virt_use_nfs

Disable the virt_use_nfs SELinux Boolean

By default, the SELinux boolean `virt_use_nfs` is disabled. If this setting is enabled, it should be disabled. To disable the `virt_use_nfs` SELinux boolean, run the following command:

```
$ sudo setsebool -P virt_use_nfs off
```

sebool_postgresql_selinux_unconfined_dbadm

Enable the postgresql_selinux_unconfined_dbadm SELinux Boolean

By default, the SELinux boolean `postgresql_selinux_unconfined_dbadm` is enabled. If this setting is disabled, it should be enabled as it allows Database Administrators to execute Data Manipulation Language (DML) statements. To enable the `postgresql_selinux_unconfined_dbadm` SELinux boolean, run the following command:

```
$ sudo setsebool -P postgresql_selinux_unconfined_dbadm on
```

sebool_ftpd_connect_all_unreserved

Disable the ftpd_connect_all_unreserved SELinux Boolean

By default, the SELinux boolean `ftpd_connect_all_unreserved` is disabled. If this setting is enabled, it should be disabled. To disable the `ftpd_connect_all_unreserved` SELinux boolean, run the following command:

```
$ sudo setsebool -P ftpd_connect_all_unreserved off
```

sebool_collectd_tcp_network_connect

Disable the collectd_tcp_network_connect SELinux Boolean

By default, the SELinux boolean `collectd_tcp_network_connect` is disabled. If this setting is enabled, it should be disabled. To disable the `collectd_tcp_network_connect` SELinux boolean, run the following command:

```
$ sudo setsebool -P collectd_tcp_network_connect off
```

sebool_puppetmaster_use_db

Disable the puppetmaster_use_db SELinux Boolean

By default, the SELinux boolean `puppetmaster_use_db` is disabled. If this setting is enabled, it should be disabled. To disable the `puppetmaster_use_db` SELinux boolean, run the following command:

```
$ sudo setsebool -P puppetmaster_use_db off
```

sebool_unconfined_chrome_sandbox_transition

Enable the unconfined_chrome_sandbox_transition SELinux Boolean

By default, the SELinux boolean `unconfined_chrome_sandbox_transition` is enabled. If this setting is disabled, it should be enabled. To enable the `unconfined_chrome_sandbox_transition` SELinux boolean, run the following command:

```
$ sudo setsebool -P unconfined_chrome_sandbox_transition on
```

sebool_httpd_manage_ipa

Disable the httpd_manage_ipa SELinux Boolean

By default, the SELinux boolean `httpd_manage_ipa` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_manage_ipa` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_manage_ipa off
```

sebool_dbadm_read_user_files

Disable the dbadm_read_user_files SELinux Boolean

By default, the SELinux boolean `dbadm_read_user_files` is disabled. If this setting is enabled, it should be disabled. To disable the `dbadm_read_user_files` SELinux boolean, run the following command:

```
$ sudo setsebool -P dbadm_read_user_files off
```

sebool_ftpd_use_nfs

Disable the ftpd_use_nfs SELinux Boolean

By default, the SELinux boolean `ftpd_use_nfs` is disabled. If this setting is enabled, it should be disabled. To disable the `ftpd_use_nfs` SELinux boolean, run the following command:

```
$ sudo setsebool -P ftpd_use_nfs off
```

sebool_use_ecryptfs_home_dirs

Disable the use_ecryptfs_home_dirs SELinux Boolean

By default, the SELinux boolean `use_ecryptfs_home_dirs` is disabled. If this setting is enabled, it should be disabled. To disable the `use_ecryptfs_home_dirs` SELinux boolean, run the following command:

```
$ sudo setsebool -P use_ecryptfs_home_dirs off
```

sebool_condor_tcp_network_connect

Disable the condor_tcp_network_connect SELinux Boolean

By default, the SELinux boolean `condor_tcp_network_connect` is disabled. If this setting is enabled, it should be disabled. To disable the `condor_tcp_network_connect` SELinux boolean, run the following command:

```
$ sudo setsebool -P condor_tcp_network_connect off
```

sebool_samba_domain_controller

Disable the samba_domain_controller SELinux Boolean

By default, the SELinux boolean `samba_domain_controller` is disabled. If this setting is enabled, it should be disabled. To disable the `samba_domain_controller` SELinux boolean, run the following command:

```
$ sudo setsebool -P samba_domain_controller off
```

sebool_staff_use_svirt

Disable the staff_use_svirt SELinux Boolean

By default, the SELinux boolean `staff_use_svirt` is disabled. If this setting is enabled, it should be disabled. To disable the `staff_use_svirt` SELinux boolean, run the following command:

```
$ sudo setsebool -P staff_use_svirt off
```

sebool_dbadm_exec_content

Enable the dbadm_exec_content SELinux Boolean

By default, the SELinux boolean `dbadm_exec_content` is enabled. If this setting is disabled, it should be enabled. To enable the `dbadm_exec_content` SELinux boolean, run the following command:

```
$ sudo setsebool -P dbadm_exec_content on
```

sebool_virt_use_rawip

Disable the virt_use_rawip SELinux Boolean

By default, the SELinux boolean `virt_use_rawip` is disabled. If this setting is enabled, it should be disabled. To disable the `virt_use_rawip` SELinux boolean, run the following command:

```
$ sudo setsebool -P virt_use_rawip off
```

sebool_gitosis_can_sendmail

Disable the gitosis_can_sendmail SELinux Boolean

By default, the SELinux boolean `gitosis_can_sendmail` is disabled. If this setting is enabled, it should be disabled. To disable the `gitosis_can_sendmail` SELinux boolean, run the following command:

```
$ sudo setsebool -P gitosis_can_sendmail off
```

sebool_virt_sandbox_use_sys_admin

Disable the virt_sandbox_use_sys_admin SELinux Boolean

By default, the SELinux boolean `virt_sandbox_use_sys_admin` is disabled. If this setting is enabled, it should be disabled. To disable the `virt_sandbox_use_sys_admin` SELinux boolean, run the following command:

```
$ sudo setsebool -P virt_sandbox_use_sys_admin off
```

sebool_dhcpd_use_ldap

Disable the dhcpd_use_ldap SELinux Boolean

By default, the SELinux boolean `dhcpd_use_ldap` is disabled. If this setting is enabled, it should be disabled. To disable the `dhcpd_use_ldap` SELinux boolean, run the following command:

```
$ sudo setsebool -P dhcpd_use_ldap off
```

sebool_mount_anyfile

Enable the mount_anyfile SELinux Boolean

By default, the SELinux boolean `mount_anyfile` is enabled. If this setting is disabled, it should be enabled to allow any file or directory to be mounted. To enable the `mount_anyfile` SELinux boolean, run the following command:

```
$ sudo setsebool -P mount_anyfile on
```

sebool_glance_api_can_network

Disable the glance_api_can_network SELinux Boolean

By default, the SELinux boolean `glance_api_can_network` is disabled. If this setting is enabled, it should be disabled. To disable the `glance_api_can_network` SELinux boolean, run the following command:

```
$ sudo setsebool -P glance_api_can_network off
```

sebool_squid_connect_any

Disable the squid_connect_any SELinux Boolean

By default, the SELinux boolean `squid_connect_any` is enabled. This setting should be disabled as squid should only connect on specified ports. To disable the `squid_connect_any` SELinux boolean, run the following command:

```
$ sudo setsebool -P squid_connect_any off
```

sebool_spamassassin_can_network

Disable the spamassassin_can_network SELinux Boolean

By default, the SELinux boolean `spamassassin_can_network` is disabled. If this setting is enabled, it should be disabled. To disable the `spamassassin_can_network` SELinux boolean, run the following command:

```
$ sudo setsebool -P spamassassin_can_network off
```

sebool_httpd_unified

Disable the httpd_unified SELinux Boolean

By default, the SELinux boolean `httpd_unified` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_unified` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_unified off
```

sebool_samba_enable_home_dirs

Disable the samba_enable_home_dirs SELinux Boolean

By default, the SELinux boolean `samba_enable_home_dirs` is disabled. If this setting is enabled, it should be disabled. To disable the `samba_enable_home_dirs` SELinux boolean, run the following command:

```
$ sudo setsebool -P samba_enable_home_dirs off
```

sebool_httpd_dbus_avahi

Disable the httpd_dbus_avahi SELinux Boolean

By default, the SELinux boolean `httpd_dbus_avahi` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_dbus_avahi` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_dbus_avahi off
```

sebool_webadm_manage_user_files

Disable the webadm_manage_user_files SELinux Boolean

By default, the SELinux boolean `webadm_manage_user_files` is disabled. If this setting is enabled, it should be disabled. To disable the `webadm_manage_user_files` SELinux boolean, run the following command:

```
$ sudo setsebool -P webadm_manage_user_files off
```

sebool_httpd_verify_dns

Disable the httpd_verify_dns SELinux Boolean

By default, the SELinux boolean `httpd_verify_dns` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_verify_dns` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_verify_dns off
```

sebool_smbd_anon_write

Disable the smbd_anon_write SELinux Boolean

By default, the SELinux boolean `smbd_anon_write` is disabled. If this setting is enabled, it should be disabled. To disable the `smbd_anon_write` SELinux boolean, run the following command:

```
$ sudo setsebool -P smbd_anon_write off
```

sebool_auditadm_exec_content

Enable the auditadm_exec_content SELinux Boolean

By default, the SELinux boolean `auditadm_exec_content` is enabled. If this setting is disabled, it should be enabled. To enable the `auditadm_exec_content` SELinux boolean, run the following command:

```
$ sudo setsebool -P auditadm_exec_content on
```

sebool_logging_syslogd_can_sendmail

Disable the logging_syslogd_can_sendmail SELinux Boolean

By default, the SELinux boolean `logging_syslogd_can_sendmail` is disabled. If this setting is enabled, it should be disabled. To disable the `logging_syslogd_can_sendmail` SELinux boolean, run the following command:

```
$ sudo setsebool -P logging_syslogd_can_sendmail off
```

sebool_cvs_read_shadow

Disable the cvs_read_shadow SELinux Boolean

By default, the SELinux boolean `cvs_read_shadow` is disabled. If this setting is enabled, it should be disabled. To disable the `cvs_read_shadow` SELinux boolean, run the following command:

```
$ sudo setsebool -P cvs_read_shadow off
```

sebool_fenced_can_network_connect

Disable the fenced_can_network_connect SELinux Boolean

By default, the SELinux boolean `fenced_can_network_connect` is disabled. If this setting is enabled, it should be disabled. To disable the `fenced_can_network_connect` SELinux boolean, run the following command:

```
$ sudo setsebool -P fenced_can_network_connect off
```

sebool_httpd_can_network_memcache

Disable the httpd_can_network_memcache SELinux Boolean

By default, the SELinux boolean `httpd_can_network_memcache` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_can_network_memcache` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_can_network_memcache off
```

sebool_httpd_dbus_sssd

Disable the httpd_dbus_sssd SELinux Boolean

By default, the SELinux boolean `httpd_dbus_sssd` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_dbus_sssd` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_dbus_sssd off
```

sebool_cluster_manage_all_files

Disable the cluster_manage_all_files SELinux Boolean

By default, the SELinux boolean `cluster_manage_all_files` is disabled. If this setting is enabled, it should be disabled. To disable the `cluster_manage_all_files` SELinux boolean, run the following command:

```
$ sudo setsebool -P cluster_manage_all_files off
```

sebool_use_lpd_server

Disable the use_lpd_server SELinux Boolean

By default, the SELinux boolean `use_lpd_server` is disabled. If this setting is enabled, it should be disabled. To disable the `use_lpd_server` SELinux boolean, run the following command:

```
$ sudo setsebool -P use_lpd_server off
```

sebool_daemons_use_tcp_wrapper

Disable the daemons_use_tcp_wrapper SELinux Boolean

By default, the SELinux boolean `daemons_use_tcp_wrapper` is disabled. If this setting is enabled, it should be disabled. To disable the `daemons_use_tcp_wrapper` SELinux boolean, run the following command:

```
$ sudo setsebool -P daemons_use_tcp_wrapper off
```

sebool_samba_run_unconfined

Disable the samba_run_unconfined SELinux Boolean

By default, the SELinux boolean `samba_run_unconfined` is disabled. If this setting is enabled, it should be disabled. To disable the `samba_run_unconfined` SELinux boolean, run the following command:

```
$ sudo setsebool -P samba_run_unconfined off
```

sebool_puppetagent_manage_all_files

Disable the puppetagent_manage_all_files SELinux Boolean

By default, the SELinux boolean `puppetagent_manage_all_files` is disabled. If this setting is enabled, it should be disabled. To disable the `puppetagent_manage_all_files` SELinux boolean, run the following command:

```
$ sudo setsebool -P puppetagent_manage_all_files off
```

sebool_nfs_export_all_rw

Enable the nfs_export_all_rw SELinux Boolean

By default, the SELinux boolean `nfs_export_all_rw` is enabled. If this setting is disabled, it should be enabled as it allows NFS to export read/write mounts. To enable the `nfs_export_all_rw` SELinux boolean, run the following command:

```
$ sudo setsebool -P nfs_export_all_rw on
```

sebool_deny_ptrace

Disable the deny_ptrace SELinux Boolean

By default, the SELinux boolean `deny_ptrace` is disabled. If this setting is enabled, it should be disabled. To disable the `deny_ptrace` SELinux boolean, run the following command:

```
$ sudo setsebool -P deny_ptrace off
```

sebool_cron_system_cronjob_use_shares

Disable the cron_system_cronjob_use_shares SELinux Boolean

By default, the SELinux boolean `cron_system_cronjob_use_shares` is disabled. If this setting is enabled, it should be disabled. To disable the `cron_system_cronjob_use_shares` SELinux boolean, run the following command:

```
$ sudo setsebool -P cron_system_cronjob_use_shares off
```

sebool_swift_can_network

Disable the swift_can_network SELinux Boolean

By default, the SELinux boolean `swift_can_network` is disabled. If this setting is enabled, it should be disabled. To disable the `swift_can_network` SELinux boolean, run the following command:

```
$ sudo setsebool -P swift_can_network off
```

sebool_abrt_handle_event

Disable the abrt_handle_event SELinux Boolean

By default, the SELinux boolean `abrt_handle_event` is disabled. If this setting is enabled, it should be disabled. To disable the `abrt_handle_event` SELinux boolean, run the following command:

```
$ sudo setsebool -P abrt_handle_event off
```

sebool_virt_sandbox_use_netlink

Disable the virt_sandbox_use_netlink SELinux Boolean

By default, the SELinux boolean `virt_sandbox_use_netlink` is disabled. If this setting is enabled, it should be disabled. To disable the `virt_sandbox_use_netlink` SELinux boolean, run the following command:

```
$ sudo setsebool -P virt_sandbox_use_netlink off
```

sebool_ftpd_use_fusefs

Disable the ftpd_use_fusefs SELinux Boolean

By default, the SELinux boolean `ftpd_use_fusefs` is disabled. If this setting is enabled, it should be disabled. To disable the `ftpd_use_fusefs` SELinux boolean, run the following command:

```
$ sudo setsebool -P ftpd_use_fusefs off
```

sebool_xen_use_nfs

Disable the xen_use_nfs SELinux Boolean

By default, the SELinux boolean `xen_use_nfs` is disabled. If this setting is enabled, it should be disabled. To disable the `xen_use_nfs` SELinux boolean, run the following command:

```
$ sudo setsebool -P xen_use_nfs off
```

sebool_telepathy_tcp_connect_generic_network_ports

Disable the telepathy_tcp_connect_generic_network_ports SELinux Boolean

By default, the SELinux boolean `telepathy_tcp_connect_generic_network_ports` is enabled. This setting should be disabled as `telepathy` should not connect to any generic network ports. To disable the `telepathy_tcp_connect_generic_network_ports` SELinux boolean, run the following command:

```
$ sudo setsebool -P telepathy_tcp_connect_generic_network_ports off
```

sebool_httpd_ssi_exec

Disable the httpd_ssi_exec SELinux Boolean

By default, the SELinux boolean `httpd_ssi_exec` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_ssi_exec` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_ssi_exec off
```

sebool_httpd_can_connect_zabbix

Disable the httpd_can_connect_zabbix SELinux Boolean

By default, the SELinux boolean `httpd_can_connect_zabbix` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_can_connect_zabbix` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_can_connect_zabbix off
```

sebool_mpd_enable_homedirs

Disable the mpd_enable_homedirs SELinux Boolean

By default, the SELinux boolean `mpd_enable_homedirs` is disabled. If this setting is enabled, it should be disabled. To disable the `mpd_enable_homedirs` SELinux boolean, run the following command:

```
$ sudo setsebool -P mpd_enable_homedirs off
```

sebool_ftpd_use_cifs

Disable the ftpd_use_cifs SELinux Boolean

By default, the SELinux boolean `ftpd_use_cifs` is disabled. If this setting is enabled, it should be disabled. To disable the `ftpd_use_cifs` SELinux boolean, run the following command:

```
$ sudo setsebool -P ftpd_use_cifs off
```

sebool_user_exec_content

Enable the user_exec_content SELinux Boolean

By default, the SELinux boolean `user_exec_content` is enabled. If this setting is disabled, it should be enabled. To enable the `user_exec_content` SELinux boolean, run the following command:

```
$ sudo setsebool -P user_exec_content on
```

sebool_httpd_builtin_scripting

Configure the httpd_builtin_scripting SELinux Boolean

By default, the SELinux boolean `httpd_builtin_scripting` is enabled. This setting should be disabled if `httpd` is not running `php` or some similary scripting language. To disable the `httpd_builtin_scripting` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_builtin_scripting off
```

sebool_cups_execmem

Disable the cups_execmem SELinux Boolean

By default, the SELinux boolean `cups_execmem` is disabled. If this setting is enabled, it should be disabled. To disable the `cups_execmem` SELinux boolean, run the following command:

```
$ sudo setsebool -P cups_execmem off
```

sebool_virt_transition_userdomain

Disable the virt_transition_userdomain SELinux Boolean

By default, the SELinux boolean `virt_transition_userdomain` is disabled. If this setting is enabled, it should be disabled. To disable the `virt_transition_userdomain` SELinux boolean, run the following command:

```
$ sudo setsebool -P virt_transition_userdomain off
```

sebool_conman_can_network

Disable the conman_can_network SELinux Boolean

By default, the SELinux boolean `conman_can_network` is disabled. If this setting is enabled, it should be disabled. To disable the `conman_can_network` SELinux boolean, run the following command:

```
$ sudo setsebool -P conman_can_network off
```

sebool_postgresql_selinux_users_ddl

Enable the postgresql_selinux_users_ddl SELinux Boolean

By default, the SELinux boolean `postgresql_selinux_users_ddl` is enabled. If this setting is disabled, it should be enabled as it allows Database Administrators to execute Data Definition Language (DDL) statements. To enable the `postgresql_selinux_users_ddl` SELinux boolean, run the following command:

```
$ sudo setsebool -P postgresql_selinux_users_ddl on
```

sebool_icecast_use_any_tcp_ports

Disable the icecast_use_any_tcp_ports SELinux Boolean

By default, the SELinux boolean `icecast_use_any_tcp_ports` is disabled. If this setting is enabled, it should be disabled. To disable the `icecast_use_any_tcp_ports` SELinux boolean, run the following command:

```
$ sudo setsebool -P icecast_use_any_tcp_ports off
```

sebool_domain_kernel_load_modules

Disable the domain_kernel_load_modules SELinux Boolean

By default, the SELinux boolean `domain_kernel_load_modules` is disabled. If this setting is enabled, it should be disabled. To disable the `domain_kernel_load_modules` SELinux boolean, run the following command:

```
$ sudo setsebool -P domain_kernel_load_modules off
```

sebool_virt_use_samba

Disable the virt_use_samba SELinux Boolean

By default, the SELinux boolean `virt_use_samba` is disabled. If this setting is enabled, it should be disabled. To disable the `virt_use_samba` SELinux boolean, run the following command:

```
$ sudo setsebool -P virt_use_samba off
```

sebool_antivirus_use_jit

Disable the antivirus_use_jit SELinux Boolean

By default, the SELinux boolean `antivirus_use_jit` is disabled. If this setting is enabled, it should be disabled. To disable the `antivirus_use_jit` SELinux boolean, run the following command:

```
$ sudo setsebool -P antivirus_use_jit off
```

sebool_named_write_master_zones

Disable the named_write_master_zones SELinux Boolean

By default, the SELinux boolean `named_write_master_zones` is disabled. If this setting is enabled, it should be disabled. To disable the `named_write_master_zones` SELinux boolean, run the following command:

```
$ sudo setsebool -P named_write_master_zones off
```

sebool_openvpn_can_network_connect

Disable the openvpn_can_network_connect SELinux Boolean

By default, the SELinux boolean `openvpn_can_network_connect` is enabled. This setting should be disabled. To disable the `openvpn_can_network_connect` SELinux boolean, run the following command:

```
$ sudo setsebool -P openvpn_can_network_connect off
```

sebool_zoneminder_anon_write

Disable the zoneminder_anon_write SELinux Boolean

By default, the SELinux boolean `zoneminder_anon_write` is disabled. If this setting is enabled, it should be disabled. To disable the `zoneminder_anon_write` SELinux boolean, run the following command:

```
$ sudo setsebool -P zoneminder_anon_write off
```

sebool_polipo_session_users

Disable the polipo_session_users SELinux Boolean

By default, the SELinux boolean `polipo_session_users` is disabled. If this setting is enabled, it should be disabled. To disable the `polipo_session_users` SELinux boolean, run the following command:

```
$ sudo setsebool -P polipo_session_users off
```

sebool_nfs_export_all_ro

Enable the nfs_export_all_ro SELinux Boolean

By default, the SELinux boolean `nfs_export_all_ro` is enabled. If this setting is disabled, it should be enabled as it allows NFS to export read-only mounts. To enable the `nfs_export_all_ro` SELinux boolean, run the following command:

```
$ sudo setsebool -P nfs_export_all_ro on
```

sebool_container_connect_any

Disable the container_connect_any SELinux Boolean

By default, the SELinux boolean `container_connect_any` is disabled. If this setting is enabled, it should be disabled. To disable the `container_connect_any` SELinux boolean, run the following command:

```
$ sudo setsebool -P container_connect_any off
```

sebool_polipo_connect_all_unreserved

Disable the polipo_connect_all_unreserved SELinux Boolean

By default, the SELinux boolean `polipo_connect_all_unreserved` is disabled. If this setting is enabled, it should be disabled. To disable the `polipo_connect_all_unreserved` SELinux boolean, run the following command:

```
$ sudo setsebool -P polipo_connect_all_unreserved off
```

sebool_ssh_chroot_rw_homedirs

Disable the ssh_chroot_rw_homedirs SELinux Boolean

By default, the SELinux boolean `ssh_chroot_rw_homedirs` is disabled. If this setting is enabled, it should be disabled. To disable the `ssh_chroot_rw_homedirs` SELinux boolean, run the following command:

```
$ sudo setsebool -P ssh_chroot_rw_homedirs off
```

sebool_xend_run_qemu

Enable the xend_run_qemu SELinux Boolean

By default, the SELinux boolean `xend_run_qemu` is enabled. If this setting is disabled, it should be enabled. To enable the `xend_run_qemu` SELinux boolean, run the following command:

```
$ sudo setsebool -P xend_run_qemu on
```

sebool_httpd_can_sendmail

Disable the httpd_can_sendmail SELinux Boolean

By default, the SELinux boolean `httpd_can_sendmail` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_can_sendmail` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_can_sendmail off
```

sebool_ksmtuned_use_cifs

Disable the ksmtuned_use_cifs SELinux Boolean

By default, the SELinux boolean `ksmtuned_use_cifs` is disabled. If this setting is enabled, it should be disabled. To disable the `ksmtuned_use_cifs` SELinux boolean, run the following command:

```
$ sudo setsebool -P ksmtuned_use_cifs off
```

sebool_zoneminder_run_sudo

Disable the zoneminder_run_sudo SELinux Boolean

By default, the SELinux boolean `zoneminder_run_sudo` is disabled. If this setting is enabled, it should be disabled. To disable the `zoneminder_run_sudo` SELinux boolean, run the following command:

```
$ sudo setsebool -P zoneminder_run_sudo off
```

sebool_authlogin_radius

Disable the authlogin_radius SELinux Boolean

By default, the SELinux boolean `authlogin_radius` is disabled. If this setting is enabled, it should be disabled. To disable the `authlogin_radius` SELinux boolean, run the following command:

```
$ sudo setsebool -P authlogin_radius off
```

sebool_httpd_enable_cgi

Configure the httpd_enable_cgi SELinux Boolean

By default, the SELinux boolean `httpd_enable_cgi` is enabled. This setting should be disabled unless `httpd` is used with `CGI` scripting. To disable the `httpd_enable_cgi` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_enable_cgi off
```

sebool_piranha_lvs_can_network_connect

Disable the piranha_lvs_can_network_connect SELinux Boolean

By default, the SELinux boolean `piranha_lvs_can_network_connect` is disabled. If this setting is enabled, it should be disabled. To disable the `piranha_lvs_can_network_connect` SELinux boolean, run the following command:

```
$ sudo setsebool -P piranha_lvs_can_network_connect off
```

sebool_use_samba_home_dirs

Disable the use_samba_home_dirs SELinux Boolean

By default, the SELinux boolean `use_samba_home_dirs` is disabled. If this setting is enabled, it should be disabled. To disable the `use_samba_home_dirs` SELinux boolean, run the following command:

```
$ sudo setsebool -P use_samba_home_dirs off
```

sebool_httpd_graceful_shutdown

Enable the httpd_graceful_shutdown SELinux Boolean

By default, the SELinux boolean `httpd_graceful_shutdown` is enabled. If this setting is disabled, it should be enabled. To enable the `httpd_graceful_shutdown` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_graceful_shutdown on
```

sebool_httpd_read_user_content

Disable the httpd_read_user_content SELinux Boolean

By default, the SELinux boolean `httpd_read_user_content` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_read_user_content` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_read_user_content off
```

sebool_webadm_read_user_files

Disable the webadm_read_user_files SELinux Boolean

By default, the SELinux boolean `webadm_read_user_files` is disabled. If this setting is enabled, it should be disabled. To disable the `webadm_read_user_files` SELinux boolean, run the following command:

```
$ sudo setsebool -P webadm_read_user_files off
```

sebool_mozilla_plugin_use_gps

Disable the mozilla_plugin_use_gps SELinux Boolean

By default, the SELinux boolean `mozilla_plugin_use_gps` is disabled. If this setting is enabled, it should be disabled. To disable the `mozilla_plugin_use_gps` SELinux boolean, run the following command:

```
$ sudo setsebool -P mozilla_plugin_use_gps off
```

sebool_rsync_full_access

Disable the rsync_full_access SELinux Boolean

By default, the SELinux boolean `rsync_full_access` is disabled. If this setting is enabled, it should be disabled. To disable the `rsync_full_access` SELinux boolean, run the following command:

```
$ sudo setsebool -P rsync_full_access off
```

sebool_openvpn_run_unconfined

Disable the openvpn_run_unconfined SELinux Boolean

By default, the SELinux boolean `openvpn_run_unconfined` is disabled. If this setting is enabled, it should be disabled. To disable the `openvpn_run_unconfined` SELinux boolean, run the following command:

```
$ sudo setsebool -P openvpn_run_unconfined off
```

sebool_ssh_sysadm_login

Disable the ssh_sysadm_login SELinux Boolean

By default, the SELinux boolean `ssh_sysadm_login` is disabled. If this setting is enabled, it should be disabled. To disable the `ssh_sysadm_login` SELinux boolean, run the following command:

```
$ sudo setsebool -P ssh_sysadm_login off
```

sebool_fips_mode

Enable the fips_mode SELinux Boolean

By default, the SELinux boolean `fips_mode` is enabled. This allows all SELinux domains to execute in `fips_mode`. If this setting is disabled, it should be enabled. To enable the `fips_mode` SELinux boolean, run the following command:

```
$ sudo setsebool -P fips_mode on
```

sebool_httpd_use_sasl

Disable the httpd_use_sasl SELinux Boolean

By default, the SELinux boolean `httpd_use_sasl` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_use_sasl` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_use_sasl off
```

sebool_httpd_use_openstack

Disable the httpd_use_openstack SELinux Boolean

By default, the SELinux boolean `httpd_use_openstack` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_use_openstack` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_use_openstack off
```

sebool_git_cgi_use_cifs

Disable the git_cgi_use_cifs SELinux Boolean

By default, the SELinux boolean `git_cgi_use_cifs` is disabled. If this setting is enabled, it should be disabled. To disable the `git_cgi_use_cifs` SELinux boolean, run the following command:

```
$ sudo setsebool -P git_cgi_use_cifs off
```

sebool_guest_exec_content

Disable the guest_exec_content SELinux Boolean

By default, the SELinux boolean `guest_exec_content` is enabled. This setting should be disabled as no guest accounts should be used. To disable the `guest_exec_content` SELinux boolean, run the following command:

```
$ sudo setsebool -P guest_exec_content off
```

sebool_httpd_can_connect_ldap

Disable the httpd_can_connect_ldap SELinux Boolean

By default, the SELinux boolean `httpd_can_connect_ldap` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_can_connect_ldap` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_can_connect_ldap off
```

sebool_postgresql_selinux_transmit_client_label

Disable the postgresql_selinux_transmit_client_label SELinux Boolean

By default, the SELinux boolean `postgresql_selinux_transmit_client_label` is disabled. If this setting is enabled, it should be disabled. To disable the `postgresql_selinux_transmit_client_label` SELinux boolean, run the following command:

```
$ sudo setsebool -P postgresql_selinux_transmit_client_label off
```

sebool_samba_export_all_ro

Disable the samba_export_all_ro SELinux Boolean

By default, the SELinux boolean `samba_export_all_ro` is disabled. If this setting is enabled, it should be disabled. To disable the `samba_export_all_ro` SELinux boolean, run the following command:

```
$ sudo setsebool -P samba_export_all_ro off
```

sebool_haproxy_connect_any

Disable the haproxy_connect_any SELinux Boolean

By default, the SELinux boolean `haproxy_connect_any` is disabled. If this setting is enabled, it should be disabled. To disable the `haproxy_connect_any` SELinux boolean, run the following command:

```
$ sudo setsebool -P haproxy_connect_any off
```

sebool_virt_use_sanlock

Disable the virt_use_sanlock SELinux Boolean

By default, the SELinux boolean `virt_use_sanlock` is disabled. If this setting is enabled, it should be disabled. To disable the `virt_use_sanlock` SELinux boolean, run the following command:

```
$ sudo setsebool -P virt_use_sanlock off
```

sebool_use_fusefs_home_dirs

Disable the use_fusefs_home_dirs SELinux Boolean

By default, the SELinux boolean `use_fusefs_home_dirs` is disabled. If this setting is enabled, it should be disabled. To disable the `use_fusefs_home_dirs` SELinux boolean, run the following command:

```
$ sudo setsebool -P use_fusefs_home_dirs off
```

sebool_authlogin_yubikey

Disable the authlogin_yubikey SELinux Boolean

By default, the SELinux boolean `authlogin_yubikey` is disabled. If this setting is enabled, it should be disabled. To disable the `authlogin_yubikey` SELinux boolean, run the following command:

```
$ sudo setsebool -P authlogin_yubikey off
```

sebool_kerberos_enabled

Enable the kerberos_enabled SELinux Boolean

By default, the SELinux boolean `kerberos_enabled` is enabled. If this setting is disabled, it should be enabled to allow confined applications to run with Kerberos. To enable the `kerberos_enabled` SELinux boolean, run the following command:

```
$ sudo setsebool -P kerberos_enabled on
```

sebool_xguest_connect_network

Disable the xguest_connect_network SELinux Boolean

By default, the SELinux boolean `xguest_connect_network` is enabled. This setting should be disabled as guest users should not be able to configure `NetworkManager`. To disable the `xguest_connect_network` SELinux boolean, run the following command:

```
$ sudo setsebool -P xguest_connect_network off
```

sebool_secure_mode_insmod

Disable the secure_mode_insmod SELinux Boolean

By default, the SELinux boolean `secure_mode_insmod` is disabled. If this setting is enabled, it should be disabled. To disable the `secure_mode_insmod` SELinux boolean, run the following command:

```
$ sudo setsebool -P secure_mode_insmod off
```

sebool_minidlna_read_generic_user_content

Disable the minidlna_read_generic_user_content SELinux Boolean

By default, the SELinux boolean `minidlna_read_generic_user_content` is disabled. If this setting is enabled, it should be disabled. To disable the `minidlna_read_generic_user_content` SELinux boolean, run the following command:

```
$ sudo setsebool -P minidlna_read_generic_user_content off
```

sebool_xdm_exec_bootloader

Disable the xdm_exec_bootloader SELinux Boolean

By default, the SELinux boolean `xdm_exec_bootloader` is disabled. If this setting is enabled, it should be disabled. To disable the `xdm_exec_bootloader` SELinux boolean, run the following command:

```
$ sudo setsebool -P xdm_exec_bootloader off
```

sebool_nis_enabled

Disable the nis_enabled SELinux Boolean

By default, the SELinux boolean `nis_enabled` is disabled. If this setting is enabled, it should be disabled. To disable the `nis_enabled` SELinux boolean, run the following command:

```
$ sudo setsebool -P nis_enabled off
```

sebool_daemons_enable_cluster_mode

Disable the daemons_enable_cluster_mode SELinux Boolean

By default, the SELinux boolean `daemons_enable_cluster_mode` is disabled. If this setting is enabled, it should be disabled. To disable the `daemons_enable_cluster_mode` SELinux boolean, run the following command:

```
$ sudo setsebool -P daemons_enable_cluster_mode off
```

sebool_tmpreaper_use_nfs

Disable the tmpreaper_use_nfs SELinux Boolean

By default, the SELinux boolean `tmpreaper_use_nfs` is disabled. If this setting is enabled, it should be disabled. To disable the `tmpreaper_use_nfs` SELinux boolean, run the following command:

```
$ sudo setsebool -P tmpreaper_use_nfs off
```

sebool_cobbler_can_network_connect

Disable the cobbler_can_network_connect SELinux Boolean

By default, the SELinux boolean `cobbler_can_network_connect` is disabled. If this setting is enabled, it should be disabled. To disable the `cobbler_can_network_connect` SELinux boolean, run the following command:

```
$ sudo setsebool -P cobbler_can_network_connect off
```

sebool_unconfined_login

Enable the unconfined_login SELinux Boolean

By default, the SELinux boolean `unconfined_login` is enabled. If this setting is disabled, it should be enabled. To enable the `unconfined_login` SELinux boolean, run the following command:

```
$ sudo setsebool -P unconfined_login on
```

sebool_virt_use_xserver

Disable the virt_use_xserver SELinux Boolean

By default, the SELinux boolean `virt_use_xserver` is disabled. If this setting is enabled, it should be disabled. To disable the `virt_use_xserver` SELinux boolean, run the following command:

```
$ sudo setsebool -P virt_use_xserver off
```

sebool_samba_load_libgfapi

Disable the samba_load_libgfapi SELinux Boolean

By default, the SELinux boolean `samba_load_libgfapi` is disabled. If this setting is enabled, it should be disabled. To disable the `samba_load_libgfapi` SELinux boolean, run the following command:

```
$ sudo setsebool -P samba_load_libgfapi off
```

sebool_virt_read_qemu_ga_data

Disable the virt_read_qemu_ga_data SELinux Boolean

By default, the SELinux boolean `virt_read_qemu_ga_data` is disabled. If this setting is enabled, it should be disabled. To disable the `virt_read_qemu_ga_data` SELinux boolean, run the following command:

```
$ sudo setsebool -P virt_read_qemu_ga_data off
```

sebool_mozilla_plugin_can_network_connect

Disable the mozilla_plugin_can_network_connect SELinux Boolean

By default, the SELinux boolean `mozilla_plugin_can_network_connect` is disabled. If this setting is enabled, it should be disabled. To disable the `mozilla_plugin_can_network_connect` SELinux boolean, run the following command:

```
$ sudo setsebool -P mozilla_plugin_can_network_connect off
```

sebool_xdm_write_home

Disable the xdm_write_home SELinux Boolean

By default, the SELinux boolean `xdm_write_home` is disabled. If this setting is enabled, it should be disabled. To disable the `xdm_write_home` SELinux boolean, run the following command:

```
$ sudo setsebool -P xdm_write_home off
```

sebool_httpd_can_network_connect_cobbler

Disable the httpd_can_network_connect_cobbler SELinux Boolean

By default, the SELinux boolean `httpd_can_network_connect_cobbler` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_can_network_connect_cobbler` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_can_network_connect_cobbler off
```

sebool_xserver_object_manager

Disable the xserver_object_manager SELinux Boolean

By default, the SELinux boolean `xserver_object_manager` is disabled. If this setting is enabled, it should be disabled. To disable the `xserver_object_manager` SELinux boolean, run the following command:

```
$ sudo setsebool -P xserver_object_manager off
```

sebool_pppd_can_insmod

Disable the pppd_can_insmod SELinux Boolean

By default, the SELinux boolean `pppd_can_insmod` is disabled. If this setting is enabled, it should be disabled. To disable the `pppd_can_insmod` SELinux boolean, run the following command:

```
$ sudo setsebool -P pppd_can_insmod off
```

sebool_exim_can_connect_db

Disable the exim_can_connect_db SELinux Boolean

By default, the SELinux boolean `exim_can_connect_db` is disabled. If this setting is enabled, it should be disabled. To disable the `exim_can_connect_db` SELinux boolean, run the following command:

```
$ sudo setsebool -P exim_can_connect_db off
```

sebool_deny_execmem

Disable the deny_execmem SELinux Boolean

By default, the SELinux boolean `deny_execmem` is disabled. If this setting is enabled, it should be disabled. To disable the `deny_execmem` SELinux boolean, run the following command:

```
$ sudo setsebool -P deny_execmem off
```

sebool_sanlock_use_nfs

Disable the sanlock_use_nfs SELinux Boolean

By default, the SELinux boolean `sanlock_use_nfs` is disabled. If this setting is enabled, it should be disabled. To disable the `sanlock_use_nfs` SELinux boolean, run the following command:

```
$ sudo setsebool -P sanlock_use_nfs off
```

sebool_pcp_read_generic_logs

Disable the pcp_read_generic_logs SELinux Boolean

By default, the SELinux boolean `pcp_read_generic_logs` is disabled. If this setting is enabled, it should be disabled. To disable the `pcp_read_generic_logs` SELinux boolean, run the following command:

```
$ sudo setsebool -P pcp_read_generic_logs off
```

sebool_staff_exec_content

Enable the staff_exec_content SELinux Boolean

By default, the SELinux boolean `staff_exec_content` is enabled. If this setting is disabled, it should be enabled. To enable the `staff_exec_content` SELinux boolean, run the following command:

```
$ sudo setsebool -P staff_exec_content on
```

sebool_sanlock_use_fusefs

Disable the sanlock_use_fusefs SELinux Boolean

By default, the SELinux boolean `sanlock_use_fusefs` is disabled. If this setting is enabled, it should be disabled. To disable the `sanlock_use_fusefs` SELinux boolean, run the following command:

```
$ sudo setsebool -P sanlock_use_fusefs off
```

sebool_xend_run_blktap

Enable the xend_run_blktap SELinux Boolean

By default, the SELinux boolean `xend_run_blktap` is enabled. If this setting is disabled, it should be enabled. To enable the `xend_run_blktap` SELinux boolean, run the following command:

```
$ sudo setsebool -P xend_run_blktap on
```

sebool_httpd_can_network_connect_db

Disable the httpd_can_network_connect_db SELinux Boolean

By default, the SELinux boolean `httpd_can_network_connect_db` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_can_network_connect_db` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_can_network_connect_db off
```

sebool_virt_sandbox_use_audit

Enable the virt_sandbox_use_audit SELinux Boolean

By default, the SELinux boolean `virt_sandbox_use_audit` is enabled. If this setting is disabled, it should be enabled to allow sandboxed containers to send audit messages. To enable the `virt_sandbox_use_audit` SELinux boolean, run the following command:

```
$ sudo setsebool -P virt_sandbox_use_audit on
```

sebool_cobbler_anon_write

Disable the cobbler_anon_write SELinux Boolean

By default, the SELinux boolean `cobbler_anon_write` is disabled. If this setting is enabled, it should be disabled. To disable the `cobbler_anon_write` SELinux boolean, run the following command:

```
$ sudo setsebool -P cobbler_anon_write off
```

sebool_cron_userdomain_transition

Enable the cron_userdomain_transition SELinux Boolean

By default, the SELinux boolean `cron_userdomain_transition` is enabled. This setting should be enabled as end user cron jobs run in their default associated user domain(s) instead of the general cronjob domain. To enable the `cron_userdomain_transition` SELinux boolean, run the following command:

```
$ sudo setsebool -P cron_userdomain_transition on
```

sebool_git_session_users

Disable the git_session_users SELinux Boolean

By default, the SELinux boolean `git_session_users` is disabled. If this setting is enabled, it should be disabled. To disable the `git_session_users` SELinux boolean, run the following command:

```
$ sudo setsebool -P git_session_users off
```

sebool_samba_create_home_dirs

Disable the samba_create_home_dirs SELinux Boolean

By default, the SELinux boolean `samba_create_home_dirs` is disabled. If this setting is enabled, it should be disabled. To disable the `samba_create_home_dirs` SELinux boolean, run the following command:

```
$ sudo setsebool -P samba_create_home_dirs off
```

sebool_kdumpgui_run_bootloader

Disable the kdumpgui_run_bootloader SELinux Boolean

By default, the SELinux boolean `kdumpgui_run_bootloader` is disabled. If this setting is enabled, it should be disabled. To disable the `kdumpgui_run_bootloader` SELinux boolean, run the following command:

```
$ sudo setsebool -P kdumpgui_run_bootloader off
```

sebool_xdm_bind_vnc_tcp_port

Disable the xdm_bind_vnc_tcp_port SELinux Boolean

By default, the SELinux boolean `xdm_bind_vnc_tcp_port` is disabled. If this setting is enabled, it should be disabled. To disable the `xdm_bind_vnc_tcp_port` SELinux boolean, run the following command:

```
$ sudo setsebool -P xdm_bind_vnc_tcp_port off
```

sebool_selinuxuser_execheap

Disable the selinuxuser_execheap SELinux Boolean

By default, the SELinux boolean `selinuxuser_execheap` is disabled. If this setting is enabled, it should be disabled. To disable the `selinuxuser_execheap` SELinux boolean, run the following command:

```
$ sudo setsebool -P selinuxuser_execheap off
```

sebool_git_system_enable_homedirs

Disable the git_system_enable_homedirs SELinux Boolean

By default, the SELinux boolean `git_system_enable_homedirs` is disabled. If this setting is enabled, it should be disabled. To disable the `git_system_enable_homedirs` SELinux boolean, run the following command:

```
$ sudo setsebool -P git_system_enable_homedirs off
```

sebool_irc_use_any_tcp_ports

Disable the irc_use_any_tcp_ports SELinux Boolean

By default, the SELinux boolean `irc_use_any_tcp_ports` is disabled. If this setting is enabled, it should be disabled. To disable the `irc_use_any_tcp_ports` SELinux boolean, run the following command:

```
$ sudo setsebool -P irc_use_any_tcp_ports off
```

sebool_gssd_read_tmp

Enable the gssd_read_tmp SELinux Boolean

By default, the SELinux boolean `gssd_read_tmp` is enabled. This setting allows `gssd` processes to access Kerberos to read TGTs in the temp directory. If this setting is disabled, it should be enabled. To enable the `gssd_read_tmp` SELinux boolean, run the following command:

```
$ sudo setsebool -P gssd_read_tmp on
```

sebool_httpd_sys_script_anon_write

Disable the httpd_sys_script_anon_write SELinux Boolean

By default, the SELinux boolean `httpd_sys_script_anon_write` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_sys_script_anon_write` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_sys_script_anon_write off
```

sebool_telepathy_connect_all_ports

Disable the telepathy_connect_all_ports SELinux Boolean

By default, the SELinux boolean `telepathy_connect_all_ports` is disabled. If this setting is enabled, it should be disabled. To disable the `telepathy_connect_all_ports` SELinux boolean, run the following command:

```
$ sudo setsebool -P telepathy_connect_all_ports off
```

sebool_daemons_dump_core

Disable the daemons_dump_core SELinux Boolean

By default, the SELinux boolean `daemons_dump_core` is disabled. If this setting is enabled, it should be disabled. To disable the `daemons_dump_core` SELinux boolean, run the following command:

```
$ sudo setsebool -P daemons_dump_core off
```

sebool_ksmtuned_use_nfs

Disable the ksmtuned_use_nfs SELinux Boolean

By default, the SELinux boolean `ksmtuned_use_nfs` is disabled. If this setting is enabled, it should be disabled. To disable the `ksmtuned_use_nfs` SELinux boolean, run the following command:

```
$ sudo setsebool -P ksmtuned_use_nfs off
```

sebool_httpd_run_preupgrade

Disable the httpd_run_preupgrade SELinux Boolean

By default, the SELinux boolean `httpd_run_preupgrade` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_run_preupgrade` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_run_preupgrade off
```

sebool_spamd_enable_home_dirs

Enable the spamd_enable_home_dirs SELinux Boolean

By default, the SELinux boolean `spamd_enable_home_dirs` is enabled. If this setting is disabled, it should be enabled. To enable the `spamd_enable_home_dirs` SELinux boolean, run the following command:

```
$ sudo setsebool -P spamd_enable_home_dirs on
```

sebool_authlogin_nsswitch_use_ldap

Disable the authlogin_nsswitch_use_ldap SELinux Boolean

By default, the SELinux boolean `authlogin_nsswitch_use_ldap` is disabled. If this setting is enabled, it should be disabled. To disable the `authlogin_nsswitch_use_ldap` SELinux boolean, run the following command:

```
$ sudo setsebool -P authlogin_nsswitch_use_ldap off
```

sebool_polipo_use_nfs

Disable the polipo_use_nfs SELinux Boolean

By default, the SELinux boolean `polipo_use_nfs` is disabled. If this setting is enabled, it should be disabled. To disable the `polipo_use_nfs` SELinux boolean, run the following command:

```
$ sudo setsebool -P polipo_use_nfs off
```

sebool_rsync_export_all_ro

Disable the rsync_export_all_ro SELinux Boolean

By default, the SELinux boolean `rsync_export_all_ro` is disabled. If this setting is enabled, it should be disabled. To disable the `rsync_export_all_ro` SELinux boolean, run the following command:

```
$ sudo setsebool -P rsync_export_all_ro off
```

sebool_logwatch_can_network_connect_mail

Disable the logwatch_can_network_connect_mail SELinux Boolean

By default, the SELinux boolean `logwatch_can_network_connect_mail` is disabled. If this setting is enabled, it should be disabled. To disable the `logwatch_can_network_connect_mail` SELinux boolean, run the following command:

```
$ sudo setsebool -P logwatch_can_network_connect_mail off
```

sebool_mmap_low_allowed

Disable the mmap_low_allowed SELinux Boolean

By default, the SELinux boolean `mmap_low_allowed` is disabled. If this setting is enabled, it should be disabled. To disable the `mmap_low_allowed` SELinux boolean, run the following command:

```
$ sudo setsebool -P mmap_low_allowed off
```

sebool_httpd_mod_auth_pam

Disable the httpd_mod_auth_pam SELinux Boolean

By default, the SELinux boolean `httpd_mod_auth_pam` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_mod_auth_pam` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_mod_auth_pam off
```

sebool_gluster_export_all_ro

Disable the gluster_export_all_ro SELinux Boolean

By default, the SELinux boolean `gluster_export_all_ro` is disabled. If this setting is enabled, it should be disabled. To disable the `gluster_export_all_ro` SELinux boolean, run the following command:

```
$ sudo setsebool -P gluster_export_all_ro off
```

sebool_nagios_run_pnp4nagios

Disable the nagios_run_pnp4nagios SELinux Boolean

By default, the SELinux boolean `nagios_run_pnp4nagios` is disabled. If this setting is enabled, it should be disabled. To disable the `nagios_run_pnp4nagios` SELinux boolean, run the following command:

```
$ sudo setsebool -P nagios_run_pnp4nagios off
```

sebool_selinuxuser_udp_server

Disable the selinuxuser_udp_server SELinux Boolean

By default, the SELinux boolean `selinuxuser_udp_server` is disabled. If this setting is enabled, it should be disabled. To disable the `selinuxuser_udp_server` SELinux boolean, run the following command:

```
$ sudo setsebool -P selinuxuser_udp_server off
```

sebool_cobbler_use_cifs

Disable the cobbler_use_cifs SELinux Boolean

By default, the SELinux boolean `cobbler_use_cifs` is disabled. If this setting is enabled, it should be disabled. To disable the `cobbler_use_cifs` SELinux boolean, run the following command:

```
$ sudo setsebool -P cobbler_use_cifs off
```

sebool_git_system_use_nfs

Disable the git_system_use_nfs SELinux Boolean

By default, the SELinux boolean `git_system_use_nfs` is disabled. If this setting is enabled, it should be disabled. To disable the `git_system_use_nfs` SELinux boolean, run the following command:

```
$ sudo setsebool -P git_system_use_nfs off
```

sebool_nagios_run_sudo

Disable the nagios_run_sudo SELinux Boolean

By default, the SELinux boolean `nagios_run_sudo` is disabled. If this setting is enabled, it should be disabled. To disable the `nagios_run_sudo` SELinux boolean, run the following command:

```
$ sudo setsebool -P nagios_run_sudo off
```

sebool_pcp_bind_all_unreserved_ports

Disable the pcp_bind_all_unreserved_ports SELinux Boolean

By default, the SELinux boolean `pcp_bind_all_unreserved_ports` is disabled. If this setting is enabled, it should be disabled. To disable the `pcp_bind_all_unreserved_ports` SELinux boolean, run the following command:

```
$ sudo setsebool -P pcp_bind_all_unreserved_ports off
```

sebool_httpd_execmem

Disable the httpd_execmem SELinux Boolean

By default, the SELinux boolean `httpd_execmem` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_execmem` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_execmem off
```

sebool_sysadm_exec_content

Enable the sysadm_exec_content SELinux Boolean

By default, the SELinux boolean `sysadm_exec_content` is enabled. If this setting is disabled, it should be enabled. To enable the `sysadm_exec_content` SELinux boolean, run the following command:

```
$ sudo setsebool -P sysadm_exec_content on
```

sebool_login_console_enabled

Enable the login_console_enabled SELinux Boolean

By default, the SELinux boolean `login_console_enabled` is enabled. If this setting is disabled, it should be enabled as it allows login from `/dev/console` to a console session. To enable the `login_console_enabled` SELinux boolean, run the following command:

```
$ sudo setsebool -P login_console_enabled on
```

sebool_mcelog_exec_scripts

Enable the mcelog_exec_scripts SELinux Boolean

By default, the SELinux boolean `mcelog_exec_scripts` is enabled. If this setting is disabled, it should be enabled. To enable the `mcelog_exec_scripts` SELinux boolean, run the following command:

```
$ sudo setsebool -P mcelog_exec_scripts on
```

sebool_httpd_serve_cobbler_files

Disable the httpd_serve_cobbler_files SELinux Boolean

By default, the SELinux boolean `httpd_serve_cobbler_files` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_serve_cobbler_files` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_serve_cobbler_files off
```

sebool_polyinstantiation_enabled

Disable the polyinstantiation_enabled SELinux Boolean

By default, the SELinux boolean `polyinstantiation_enabled` is disabled. If this setting is enabled, it should be disabled. To disable the `polyinstantiation_enabled` SELinux boolean, run the following command:

```
$ sudo setsebool -P polyinstantiation_enabled off
```

sebool_xguest_mount_media

Disable the xguest_mount_media SELinux Boolean

By default, the SELinux boolean `xguest_mount_media` is enabled. This setting should be disabled as guest users should not be able to mount any media. To disable the `xguest_mount_media` SELinux boolean, run the following command:

```
$ sudo setsebool -P xguest_mount_media off
```

sebool_httpd_use_gpg

Disable the httpd_use_gpg SELinux Boolean

By default, the SELinux boolean `httpd_use_gpg` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_use_gpg` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_use_gpg off
```

sebool_mcelog_client

Disable the mcelog_client SELinux Boolean

By default, the SELinux boolean `mcelog_client` is disabled. If this setting is enabled, it should be disabled. To disable the `mcelog_client` SELinux boolean, run the following command:

```
$ sudo setsebool -P mcelog_client off
```

sebool_zebra_write_config

Disable the zebra_write_config SELinux Boolean

By default, the SELinux boolean `zebra_write_config` is disabled. If this setting is enabled, it should be disabled. To disable the `zebra_write_config` SELinux boolean, run the following command:

```
$ sudo setsebool -P zebra_write_config off
```

sebool_xserver_clients_write_xshm

Disable the xserver_clients_write_xshm SELinux Boolean

By default, the SELinux boolean `xserver_clients_write_xshm` is disabled. If this setting is enabled, it should be disabled. To disable the `xserver_clients_write_xshm` SELinux boolean, run the following command:

```
$ sudo setsebool -P xserver_clients_write_xshm off
```

sebool_mailman_use_fusefs

Disable the mailman_use_fusefs SELinux Boolean

By default, the SELinux boolean `mailman_use_fusefs` is disabled. If this setting is enabled, it should be disabled. To disable the `mailman_use_fusefs` SELinux boolean, run the following command:

```
$ sudo setsebool -P mailman_use_fusefs off
```

sebool_git_cgi_use_nfs

Disable the git_cgi_use_nfs SELinux Boolean

By default, the SELinux boolean `git_cgi_use_nfs` is disabled. If this setting is enabled, it should be disabled. To disable the `git_cgi_use_nfs` SELinux boolean, run the following command:

```
$ sudo setsebool -P git_cgi_use_nfs off
```

sebool_daemons_use_tty

Disable the daemons_use_tty SELinux Boolean

By default, the SELinux boolean `daemons_use_tty` is disabled. If this setting is enabled, it should be disabled. To disable the `daemons_use_tty` SELinux boolean, run the following command:

```
$ sudo setsebool -P daemons_use_tty off
```

sebool_git_cgi_enable_homedirs

Disable the git_cgi_enable_homedirs SELinux Boolean

By default, the SELinux boolean `git_cgi_enable_homedirs` is disabled. If this setting is enabled, it should be disabled. To disable the `git_cgi_enable_homedirs` SELinux boolean, run the following command:

```
$ sudo setsebool -P git_cgi_enable_homedirs off
```

sebool_fcron_crond

Disable the fcron_crond SELinux Boolean

By default, the SELinux boolean `fcron_crond` is disabled. If this setting is enabled, it should be disabled. To disable the `fcron_crond` SELinux boolean, run the following command:

```
$ sudo setsebool -P fcron_crond off
```

sebool_ftpd_full_access

Disable the ftpd_full_access SELinux Boolean

By default, the SELinux boolean `ftpd_full_access` is disabled. If this setting is enabled, it should be disabled. To disable the `ftpd_full_access` SELinux boolean, run the following command:

```
$ sudo setsebool -P ftpd_full_access off
```

sebool_logging_syslogd_use_tty

Enable the logging_syslogd_use_tty SELinux Boolean

By default, the SELinux boolean `logging_syslogd_use_tty` is enabled. If this setting is disabled, it should be enabled as it allows `syslog` the ability to read/write to terminal. To enable the `logging_syslogd_use_tty` SELinux boolean, run the following command:

```
$ sudo setsebool -P logging_syslogd_use_tty on
```

sebool_samba_portmapper

Disable the samba_portmapper SELinux Boolean

By default, the SELinux boolean `samba_portmapper` is disabled. If this setting is enabled, it should be disabled. To disable the `samba_portmapper` SELinux boolean, run the following command:

```
$ sudo setsebool -P samba_portmapper off
```

sebool_xguest_use_bluetooth

Disable the xguest_use_bluetooth SELinux Boolean

By default, the SELinux boolean `xguest_use_bluetooth` is enabled. This setting should be disabled as guests users should not be able to access or use bluetooth. To disable the `xguest_use_bluetooth` SELinux boolean, run the following command:

```
$ sudo setsebool -P xguest_use_bluetooth off
```

sebool_tftp_anon_write

Disable the tftp_anon_write SELinux Boolean

By default, the SELinux boolean `tftp_anon_write` is disabled. If this setting is enabled, it should be disabled. To disable the `tftp_anon_write` SELinux boolean, run the following command:

```
$ sudo setsebool -P tftp_anon_write off
```

sebool_pppd_for_user

Disable the pppd_for_user SELinux Boolean

By default, the SELinux boolean `pppd_for_user` is disabled. If this setting is enabled, it should be disabled. To disable the `pppd_for_user` SELinux boolean, run the following command:

```
$ sudo setsebool -P pppd_for_user off
```

sebool_irssi_use_full_network

Disable the irssi_use_full_network SELinux Boolean

By default, the SELinux boolean `irssi_use_full_network` is disabled. If this setting is enabled, it should be disabled. To disable the `irssi_use_full_network` SELinux boolean, run the following command:

```
$ sudo setsebool -P irssi_use_full_network off
```

sebool_mcelog_server

Disable the mcelog_server SELinux Boolean

By default, the SELinux boolean `mcelog_server` is disabled. If this setting is enabled, it should be disabled. To disable the `mcelog_server` SELinux boolean, run the following command:

```
$ sudo setsebool -P mcelog_server off
```

sebool_samba_export_all_rw

Disable the samba_export_all_rw SELinux Boolean

By default, the SELinux boolean `samba_export_all_rw` is disabled. If this setting is enabled, it should be disabled. To disable the `samba_export_all_rw` SELinux boolean, run the following command:

```
$ sudo setsebool -P samba_export_all_rw off
```

sebool_httpd_run_stickshift

Disable the httpd_run_stickshift SELinux Boolean

By default, the SELinux boolean `httpd_run_stickshift` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_run_stickshift` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_run_stickshift off
```

sebool_virt_use_comm

Disable the virt_use_comm SELinux Boolean

By default, the SELinux boolean `virt_use_comm` is disabled. If this setting is enabled, it should be disabled. To disable the `virt_use_comm` SELinux boolean, run the following command:

```
$ sudo setsebool -P virt_use_comm off
```

sebool_nfsd_anon_write

Disable the nfsd_anon_write SELinux Boolean

By default, the SELinux boolean `nfsd_anon_write` is disabled. If this setting is enabled, it should be disabled. To disable the `nfsd_anon_write` SELinux boolean, run the following command:

```
$ sudo setsebool -P nfsd_anon_write off
```

sebool_selinuxuser_direct_dri_enabled

Configure the selinuxuser_direct_dri_enabled SELinux Boolean

By default, the SELinux boolean `selinuxuser_direct_dri_enabled` is enabled. If XWindows is not installed or used on the system, this setting should be disabled. Otherwise, enable it. To disable the `selinuxuser_direct_dri_enabled` SELinux boolean, run the following command:

```
$ sudo setsebool -P selinuxuser_direct_dri_enabled off
```

sebool_neutron_can_network

Disable the neutron_can_network SELinux Boolean

By default, the SELinux boolean `neutron_can_network` is disabled. If this setting is enabled, it should be disabled. To disable the `neutron_can_network` SELinux boolean, run the following command:

```
$ sudo setsebool -P neutron_can_network off
```

sebool_openshift_use_nfs

Disable the openshift_use_nfs SELinux Boolean

By default, the SELinux boolean `openshift_use_nfs` is disabled. If this setting is enabled, it should be disabled. To disable the `openshift_use_nfs` SELinux boolean, run the following command:

```
$ sudo setsebool -P openshift_use_nfs off
```

sebool_cobbler_use_nfs

Disable the cobbler_use_nfs SELinux Boolean

By default, the SELinux boolean `cobbler_use_nfs` is disabled. If this setting is enabled, it should be disabled. To disable the `cobbler_use_nfs` SELinux boolean, run the following command:

```
$ sudo setsebool -P cobbler_use_nfs off
```

sebool_unconfined_mozilla_plugin_transition

Enable the unconfined_mozilla_plugin_transition SELinux Boolean

By default, the SELinux boolean `unconfined_mozilla_plugin_transition` is enabled. If this setting is disabled, it should be enabled. To enable the `unconfined_mozilla_plugin_transition` SELinux boolean, run the following command:

```
$ sudo setsebool -P unconfined_mozilla_plugin_transition on
```

sebool_mozilla_plugin_use_spice

Disable the mozilla_plugin_use_spice SELinux Boolean

By default, the SELinux boolean `mozilla_plugin_use_spice` is disabled. If this setting is enabled, it should be disabled. To disable the `mozilla_plugin_use_spice` SELinux boolean, run the following command:

```
$ sudo setsebool -P mozilla_plugin_use_spice off
```

sebool_dhcpc_exec_iptables

Disable the dhcpc_exec_iptables SELinux Boolean

By default, the SELinux boolean `dhcpc_exec_iptables` is disabled. If this setting is enabled, it should be disabled. To disable the `dhcpc_exec_iptables` SELinux boolean, run the following command:

```
$ sudo setsebool -P dhcpc_exec_iptables off
```

sebool_ftpd_connect_db

Disable the ftpd_connect_db SELinux Boolean

By default, the SELinux boolean `ftpd_connect_db` is disabled. If this setting is enabled, it should be disabled. To disable the `ftpd_connect_db` SELinux boolean, run the following command:

```
$ sudo setsebool -P ftpd_connect_db off
```

sebool_ftpd_use_passive_mode

Disable the ftpd_use_passive_mode SELinux Boolean

By default, the SELinux boolean `ftpd_use_passive_mode` is disabled. If this setting is enabled, it should be disabled. To disable the `ftpd_use_passive_mode` SELinux boolean, run the following command:

```
$ sudo setsebool -P ftpd_use_passive_mode off
```

sebool_virt_use_fusefs

Disable the virt_use_fusefs SELinux Boolean

By default, the SELinux boolean `virt_use_fusefs` is disabled. If this setting is enabled, it should be disabled. To disable the `virt_use_fusefs` SELinux boolean, run the following command:

```
$ sudo setsebool -P virt_use_fusefs off
```

sebool_httpd_setrlimit

Disable the httpd_setrlimit SELinux Boolean

By default, the SELinux boolean `httpd_setrlimit` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_setrlimit` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_setrlimit off
```

sebool_mozilla_plugin_bind_unreserved_ports

Disable the mozilla_plugin_bind_unreserved_ports SELinux Boolean

By default, the SELinux boolean `mozilla_plugin_bind_unreserved_ports` is disabled. If this setting is enabled, it should be disabled. To disable the `mozilla_plugin_bind_unreserved_ports` SELinux boolean, run the following command:

```
$ sudo setsebool -P mozilla_plugin_bind_unreserved_ports off
```

sebool_logrotate_use_nfs

Disable the logrotate_use_nfs SELinux Boolean

By default, the SELinux boolean `logrotate_use_nfs` is disabled. If this setting is enabled, it should be disabled. To disable the `logrotate_use_nfs` SELinux boolean, run the following command:

```
$ sudo setsebool -P logrotate_use_nfs off
```

sebool_httpd_use_nfs

Disable the httpd_use_nfs SELinux Boolean

By default, the SELinux boolean `httpd_use_nfs` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_use_nfs` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_use_nfs off
```

sebool_unprivuser_use_svirt

Disable the unprivuser_use_svirt SELinux Boolean

By default, the SELinux boolean `unprivuser_use_svirt` is disabled. If this setting is enabled, it should be disabled. To disable the `unprivuser_use_svirt` SELinux boolean, run the following command:

```
$ sudo setsebool -P unprivuser_use_svirt off
```

sebool_virt_rw_qemu_ga_data

Disable the virt_rw_qemu_ga_data SELinux Boolean

By default, the SELinux boolean `virt_rw_qemu_ga_data` is disabled. If this setting is enabled, it should be disabled. To disable the `virt_rw_qemu_ga_data` SELinux boolean, run the following command:

```
$ sudo setsebool -P virt_rw_qemu_ga_data off
```

sebool_secure_mode_policyload

Disable the secure_mode_policyload SELinux Boolean

By default, the SELinux boolean `secure_mode_policyload` is disabled. If this setting is enabled, it should be disabled. To disable the `secure_mode_policyload` SELinux boolean, run the following command:

```
$ sudo setsebool -P secure_mode_policyload off
```

sebool_httpd_tmp_exec

Disable the httpd_tmp_exec SELinux Boolean

By default, the SELinux boolean `httpd_tmp_exec` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_tmp_exec` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_tmp_exec off
```

sebool_mysql_connect_any

Disable the mysql_connect_any SELinux Boolean

By default, the SELinux boolean `mysql_connect_any` is disabled. If this setting is enabled, it should be disabled. To disable the `mysql_connect_any` SELinux boolean, run the following command:

```
$ sudo setsebool -P mysql_connect_any off
```

sebool_mpd_use_nfs

Disable the mpd_use_nfs SELinux Boolean

By default, the SELinux boolean `mpd_use_nfs` is disabled. If this setting is enabled, it should be disabled. To disable the `mpd_use_nfs` SELinux boolean, run the following command:

```
$ sudo setsebool -P mpd_use_nfs off
```

sebool_selinuxuser_execstack

disable the selinuxuser_execstack SELinux Boolean

By default, the SELinux boolean `selinuxuser_execstack` is enabled. This setting should be disabled as unconfined executables should not be able to make their stack executable. To disable the `selinuxuser_execstack` SELinux boolean, run the following command:

```
$ sudo setsebool -P selinuxuser_execstack off
```

sebool_glance_use_execmem

Disable the glance_use_execmem SELinux Boolean

By default, the SELinux boolean `glance_use_execmem` is disabled. If this setting is enabled, it should be disabled. To disable the `glance_use_execmem` SELinux boolean, run the following command:

```
$ sudo setsebool -P glance_use_execmem off
```

sebool_selinuxuser_postgresql_connect_enabled

Disable the selinuxuser_postgresql_connect_enabled SELinux Boolean

By default, the SELinux boolean `selinuxuser_postgresql_connect_enabled` is disabled. If this setting is enabled, it should be disabled. To disable the `selinuxuser_postgresql_connect_enabled` SELinux boolean, run the following command:

```
$ sudo setsebool -P selinuxuser_postgresql_connect_enabled off
```

sebool_virt_use_execmem

Disable the virt_use_execmem SELinux Boolean

By default, the SELinux boolean `virt_use_execmem` is disabled. If this setting is enabled, it should be disabled. To disable the `virt_use_execmem` SELinux boolean, run the following command:

```
$ sudo setsebool -P virt_use_execmem off
```

sebool_wine_mmap_zero_ignore

Disable the wine_mmap_zero_ignore SELinux Boolean

By default, the SELinux boolean `wine_mmap_zero_ignore` is disabled. If this setting is enabled, it should be disabled. To disable the `wine_mmap_zero_ignore` SELinux boolean, run the following command:

```
$ sudo setsebool -P wine_mmap_zero_ignore off
```

sebool_rsync_client

Disable the rsync_client SELinux Boolean

By default, the SELinux boolean `rsync_client` is disabled. If this setting is enabled, it should be disabled. To disable the `rsync_client` SELinux boolean, run the following command:

```
$ sudo setsebool -P rsync_client off
```

sebool_git_session_bind_all_unreserved_ports

Disable the git_session_bind_all_unreserved_ports SELinux Boolean

By default, the SELinux boolean `git_session_bind_all_unreserved_ports` is disabled. If this setting is enabled, it should be disabled. To disable the `git_session_bind_all_unreserved_ports` SELinux boolean, run the following command:

```
$ sudo setsebool -P git_session_bind_all_unreserved_ports off
```

sebool_logging_syslogd_run_nagios_plugins

Disable the logging_syslogd_run_nagios_plugins SELinux Boolean

By default, the SELinux boolean `logging_syslogd_run_nagios_plugins` is disabled. If this setting is enabled, it should be disabled. To disable the `logging_syslogd_run_nagios_plugins` SELinux boolean, run the following command:

```
$ sudo setsebool -P logging_syslogd_run_nagios_plugins off
```

sebool_selinuxuser_mysql_connect_enabled

Disable the selinuxuser_mysql_connect_enabled SELinux Boolean

By default, the SELinux boolean `selinuxuser_mysql_connect_enabled` is disabled. If this setting is enabled, it should be disabled. To disable the `selinuxuser_mysql_connect_enabled` SELinux boolean, run the following command:

```
$ sudo setsebool -P selinuxuser_mysql_connect_enabled off
```

sebool_cdrecord_read_content

Disable the cdrecord_read_content SELinux Boolean

By default, the SELinux boolean `cdrecord_read_content` is disabled. If this setting is enabled, it should be disabled. To disable the `cdrecord_read_content` SELinux boolean, run the following command:

```
$ sudo setsebool -P cdrecord_read_content off
```

sebool_secure_mode

Disable the secure_mode SELinux Boolean

By default, the SELinux boolean `secure_mode` is disabled. If this setting is enabled, it should be disabled. To disable the `secure_mode` SELinux boolean, run the following command:

```
$ sudo setsebool -P secure_mode off
```

sebool_httpd_anon_write

Disable the httpd_anon_write SELinux Boolean

By default, the SELinux boolean `httpd_anon_write` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_anon_write` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_anon_write off
```

sebool_prosody_bind_http_port

Disable the prosody_bind_http_port SELinux Boolean

By default, the SELinux boolean `prosody_bind_http_port` is disabled. If this setting is enabled, it should be disabled. To disable the `prosody_bind_http_port` SELinux boolean, run the following command:

```
$ sudo setsebool -P prosody_bind_http_port off
```

sebool_sge_use_nfs

Disable the sge_use_nfs SELinux Boolean

By default, the SELinux boolean `sge_use_nfs` is disabled. If this setting is enabled, it should be disabled. To disable the `sge_use_nfs` SELinux boolean, run the following command:

```
$ sudo setsebool -P sge_use_nfs off
```

sebool_polipo_use_cifs

Disable the polipo_use_cifs SELinux Boolean

By default, the SELinux boolean `polipo_use_cifs` is disabled. If this setting is enabled, it should be disabled. To disable the `polipo_use_cifs` SELinux boolean, run the following command:

```
$ sudo setsebool -P polipo_use_cifs off
```

sebool_nscd_use_shm

Enable the nscd_use_shm SELinux Boolean

By default, the SELinux boolean `nscd_use_shm` is enabled. If this setting is disabled, it should be enabled to allow `nscd` to use shared memory. To enable the `nscd_use_shm` SELinux boolean, run the following command:

```
$ sudo setsebool -P nscd_use_shm on
```

sebool_httpd_can_network_relay

Disable the httpd_can_network_relay SELinux Boolean

By default, the SELinux boolean `httpd_can_network_relay` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_can_network_relay` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_can_network_relay off
```

sebool_mock_enable_homedirs

Disable the mock_enable_homedirs SELinux Boolean

By default, the SELinux boolean `mock_enable_homedirs` is disabled. If this setting is enabled, it should be disabled. To disable the `mock_enable_homedirs` SELinux boolean, run the following command:

```
$ sudo setsebool -P mock_enable_homedirs off
```

sebool_mcelog_foreground

Disable the mcelog_foreground SELinux Boolean

By default, the SELinux boolean `mcelog_foreground` is disabled. If this setting is enabled, it should be disabled. To disable the `mcelog_foreground` SELinux boolean, run the following command:

```
$ sudo setsebool -P mcelog_foreground off
```

sebool_squid_use_tproxy

Disable the squid_use_tproxy SELinux Boolean

By default, the SELinux boolean `squid_use_tproxy` is disabled. If this setting is enabled, it should be disabled. To disable the `squid_use_tproxy` SELinux boolean, run the following command:

```
$ sudo setsebool -P squid_use_tproxy off
```

sebool_sge_domain_can_network_connect

Disable the sge_domain_can_network_connect SELinux Boolean

By default, the SELinux boolean `sge_domain_can_network_connect` is disabled. If this setting is enabled, it should be disabled. To disable the `sge_domain_can_network_connect` SELinux boolean, run the following command:

```
$ sudo setsebool -P sge_domain_can_network_connect off
```

sebool_selinuxuser_share_music

Disable the selinuxuser_share_music SELinux Boolean

By default, the SELinux boolean `selinuxuser_share_music` is disabled. If this setting is enabled, it should be disabled. To disable the `selinuxuser_share_music` SELinux boolean, run the following command:

```
$ sudo setsebool -P selinuxuser_share_music off
```

sebool_httpd_can_connect_ftp

Disable the httpd_can_connect_ftp SELinux Boolean

By default, the SELinux boolean `httpd_can_connect_ftp` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_can_connect_ftp` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_can_connect_ftp off
```

sebool_xguest_exec_content

Disable the xguest_exec_content SELinux Boolean

By default, the SELinux boolean `xguest_exec_content` is enabled. This setting should be disabled as guest users should not be able to run executables. To disable the `xguest_exec_content` SELinux boolean, run the following command:

```
$ sudo setsebool -P xguest_exec_content off
```

sebool_exim_manage_user_files

Disable the exim_manage_user_files SELinux Boolean

By default, the SELinux boolean `exim_manage_user_files` is disabled. If this setting is enabled, it should be disabled. To disable the `exim_manage_user_files` SELinux boolean, run the following command:

```
$ sudo setsebool -P exim_manage_user_files off
```

sebool_cluster_use_execmem

Disable the cluster_use_execmem SELinux Boolean

By default, the SELinux boolean `cluster_use_execmem` is disabled. If this setting is enabled, it should be disabled. To disable the `cluster_use_execmem` SELinux boolean, run the following command:

```
$ sudo setsebool -P cluster_use_execmem off
```

sebool_mpd_use_cifs

Disable the mpd_use_cifs SELinux Boolean

By default, the SELinux boolean `mpd_use_cifs` is disabled. If this setting is enabled, it should be disabled. To disable the `mpd_use_cifs` SELinux boolean, run the following command:

```
$ sudo setsebool -P mpd_use_cifs off
```

sebool_logadm_exec_content

Enable the logadm_exec_content SELinux Boolean

By default, the SELinux boolean `logadm_exec_content` is enabled. If this setting is disabled, it should be enabled. To enable the `logadm_exec_content` SELinux boolean, run the following command:

```
$ sudo setsebool -P logadm_exec_content on
```

sebool_httpd_can_connect_mythtv

Disable the httpd_can_connect_mythtv SELinux Boolean

By default, the SELinux boolean `httpd_can_connect_mythtv` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_can_connect_mythtv` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_can_connect_mythtv off
```

sebool_racoon_read_shadow

Disable the racoon_read_shadow SELinux Boolean

By default, the SELinux boolean `racoon_read_shadow` is disabled. If this setting is enabled, it should be disabled. To disable the `racoon_read_shadow` SELinux boolean, run the following command:

```
$ sudo setsebool -P racoon_read_shadow off
```

sebool_fenced_can_ssh

Disable the fenced_can_ssh SELinux Boolean

By default, the SELinux boolean `fenced_can_ssh` is disabled. If this setting is enabled, it should be disabled. To disable the `fenced_can_ssh` SELinux boolean, run the following command:

```
$ sudo setsebool -P fenced_can_ssh off
```

sebool_privoxy_connect_any

Disable the privoxy_connect_any SELinux Boolean

By default, the SELinux boolean `privoxy_connect_any` is enabled. This setting should be disabled. To disable the `privoxy_connect_any` SELinux boolean, run the following command:

```
$ sudo setsebool -P privoxy_connect_any off
```

sebool_virt_sandbox_use_mknod

Disable the virt_sandbox_use_mknod SELinux Boolean

By default, the SELinux boolean `virt_sandbox_use_mknod` is disabled. If this setting is enabled, it should be disabled. To disable the `virt_sandbox_use_mknod` SELinux boolean, run the following command:

```
$ sudo setsebool -P virt_sandbox_use_mknod off
```

sebool_httpd_dontaudit_search_dirs

Disable the httpd_dontaudit_search_dirs SELinux Boolean

By default, the SELinux boolean `httpd_dontaudit_search_dirs` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_dontaudit_search_dirs` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_dontaudit_search_dirs off
```

sebool_httpd_enable_homedirs

Disable the httpd_enable_homedirs SELinux Boolean

By default, the SELinux boolean `httpd_enable_homedirs` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_enable_homedirs` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_enable_homedirs off
```

sebool_httpd_use_fusefs

Disable the httpd_use_fusefs SELinux Boolean

By default, the SELinux boolean `httpd_use_fusefs` is disabled. If this setting is enabled, it should be disabled. To disable the `httpd_use_fusefs` SELinux boolean, run the following command:

```
$ sudo setsebool -P httpd_use_fusefs off
```

sebool_zarafa_setrlimit

Disable the zarafa_setrlimit SELinux Boolean

By default, the SELinux boolean `zarafa_setrlimit` is disabled. If this setting is enabled, it should be disabled. To disable the `zarafa_setrlimit` SELinux boolean, run the following command:

```
$ sudo setsebool -P zarafa_setrlimit off
```

sebool_glance_use_fusefs

Disable the glance_use_fusefs SELinux Boolean

By default, the SELinux boolean `glance_use_fusefs` is disabled. If this setting is enabled, it should be disabled. To disable the `glance_use_fusefs` SELinux boolean, run the following command:

```
$ sudo setsebool -P glance_use_fusefs off
```

sebool_xserver_execmem

Disable the xserver_execmem SELinux Boolean

By default, the SELinux boolean `xserver_execmem` is disabled. If this setting is enabled, it should be disabled. To disable the `xserver_execmem` SELinux boolean, run the following command:

```
$ sudo setsebool -P xserver_execmem off
```

grub2_enable_selinux

Ensure SELinux Not Disabled in /etc/default/grub

SELinux can be disabled at boot time by an argument in `/etc/default/grub`. Remove any instances of `selinux=0` from the kernel arguments in that file to prevent SELinux from being disabled at boot.

selinux_all_devicefiles_labeled

Ensure No Device Files are Unlabeled by SELinux

Device files, which are used for communication with important system resources, should be labeled with proper SELinux types. If any device files do not carry the SELinux type `device_t`, report the bug so that policy can be corrected. Supply information about what the device is and what programs use it.

To check for unlabeled device files, run the following command:

```
$ sudo find /dev -context *:device_t:* \( -type c -o -type b \) -printf "%p %Z\n"
```

It should produce no output in a well-configured system.

selinux_confinement_of_daemons

Ensure No Daemons are Unconfined by SELinux

Daemons for which the SELinux policy does not contain rules will inherit the context of the parent process. Because daemons are launched during startup and descend from the `init` process, they inherit the `initrc_t` context.

To check for unconfined daemons, run the following command:

```
$ sudo ps -eZ | egrep "initrc" | egrep -vw "tr|ps|egrep|bash|awk" | tr ':' ' ' | awk '{ print $NF }'
```

It should produce no output in a well-configured system.

package_setroubleshoot_removed

Uninstall setroubleshoot Package

The SETroubleshoot service notifies desktop users of SELinux denials. The service provides information around configuration errors, unauthorized intrusions, and other potential errors. The `setroubleshoot` package can be removed with the following command:

```
$ sudo yum erase setroubleshoot
```

package_mcstrans_removed

Uninstall mcstrans Package

The `mcstransd` daemon provides category label information to client processes requesting information. The label translations are defined in `/etc/selinux/targeted/setrans.conf`. The `mcstrans` package can be removed with the following command:

```
$ sudo yum erase mcstrans
```

selinux_policytype

Configure SELinux Policy

The SELinux `targeted` policy is appropriate for general-purpose desktops and servers, as well as systems in many other roles. To configure the system to use this policy, add or correct the following line in `/etc/selinux/config`:

```
SELINUXTYPE=
```

Other policies, such as `mls`, provide additional security labeling and greater confinement but are not compatible with many general-purpose use cases.

selinux_user_login_roles

Map System Users To The Appropriate SELinux Role

Configure the operating system to prevent non-privileged users from executing privileged functions to include disabling, circumventing, or altering implemented security safeguards/countermeasures. All administrators must be mapped to the `sysadm_u` or `staff_u` users with the appropriate domains (`sysadm_t` and `staff_t`).

```
$ sudo semanage login -m -s sysadm_u USER
```

or

```
$ sudo semanage login -m -s staff_u USER
```

All authorized non-administrative users must be mapped to the `user_u` role or the appropriate domain (user_t).

```
$ sudo semanage login -m -s user_u USER
```

selinux_state

Ensure SELinux State is Enforcing

The SELinux state should be set to `` `at system boot time. In the file`/etc/selinux/config` ``, add or correct the following line to configure the system to boot into enforcing mode:

```
SELINUX=
```

File Permissions and Masks

Traditional Unix security relies heavily on file and directory permissions to prevent unauthorized users from reading or modifying files to which they should not have access.

Several of the commands in this section search filesystems for files or directories with certain characteristics, and are intended to be run on every local partition on a given system. When the variable *PART* appears in one of the commands below, it means that the command is intended to be run repeatedly, with the name of each local partition substituted for *PART* in turn.

The following command prints a list of all xfs partitions on the local system, which is the default filesystem for Red Hat Enterprise Linux 8 installations:

```
$ mount -t xfs | awk '{print $3}'
```

For any systems that use a different local filesystem type, modify this command as appropriate.

Verify Permissions on Important Files and Directories

Permissions for many files on a system must be set restrictively to ensure sensitive information is properly protected. This section discusses important permission restrictions which can be verified to ensure that no harmful discrepancies have arisen.

Verify File Permissions Within Some Important Directories

Some directories contain files whose confidentiality or integrity is notably important and may also be susceptible to misconfiguration over time, particularly if unpackaged software is installed. As such, an argument exists to verify that files' permissions within these directories remain configured correctly and restrictively.

file_permissions_library_dirs

Verify that Shared Library Files Have Restrictive Permissions

System-wide shared library files, which are linked to executables during process load time or run time, are stored in the following directories by default:

```
/lib
/lib64
/usr/lib
/usr/lib64
```

Kernel modules, which can be added to the kernel during runtime, are stored in `/lib/modules`. All files in these directories should not be group-writable or world-writable. If any file in these directories is found to be group-writable or world-writable, correct its permission with the following command:

```
$ sudo chmod go-w FILE
```

file_ownership_library_dirs

Verify that Shared Library Files Have Root Ownership

System-wide shared library files, which are linked to executables during process load time or run time, are stored in the following directories by default:

```
/lib
/lib64
/usr/lib
/usr/lib64
```

Kernel modules, which can be added to the kernel during runtime, are also stored in `/lib/modules`. All files in these directories should be owned by the `root` user. If the directory, or any file in these directories, is found to be owned by a user other than root correct its ownership with the following command:

```
$ sudo chown root FILE
```

file_permissions_binary_dirs

Verify that System Executables Have Restrictive Permissions

System executables are stored in the following directories by default:

```
/bin
/sbin
/usr/bin
/usr/libexec
/usr/local/bin
/usr/local/sbin
/usr/sbin
```

All files in these directories should not be group-writable or world-writable. If any file *FILE* in these directories is found to be group-writable or world-writable, correct its permission with the following command:

```
$ sudo chmod go-w FILE
```

file_ownership_binary_dirs

Verify that System Executables Have Root Ownership

System executables are stored in the following directories by default:

```
/bin
/sbin
/usr/bin
/usr/libexec
/usr/local/bin
/usr/local/sbin
/usr/sbin
```

All files in these directories should be owned by the `root` user. If any file *FILE* in these directories is found to be owned by a user other than root, correct its ownership with the following command:

```
$ sudo chown root FILE
```

Verify Permissions on Files with Local Account Information and Credentials

The default restrictive permissions for files which act as important security databases such as `passwd`, `shadow`, `group`, and `gshadow` files must be maintained. Many utilities need read access to the `passwd` file in order to function properly, but read access to the `shadow` file allows malicious attacks against system passwords, and should never be enabled.

file_owner_etc_gshadow

Verify User Who Owns gshadow File

To properly set the owner of `/etc/gshadow`, run the command:

```
$ sudo chown root /etc/gshadow
```

file_permissions_etc_gshadow

Verify Permissions on gshadow File

To properly set the permissions of `/etc/gshadow`, run the command:

```
$ sudo chmod 0000 /etc/gshadow
```

file_owner_etc_group

Verify User Who Owns group File

To properly set the owner of `/etc/group`, run the command:

```
$ sudo chown root /etc/group
```

file_groupowner_etc_group

Verify Group Who Owns group File

To properly set the group owner of `/etc/group`, run the command:

```
$ sudo chgrp root /etc/group
```

file_groupowner_etc_shadow

Verify Group Who Owns shadow File

To properly set the group owner of `/etc/shadow`, run the command:

```
$ sudo chgrp root /etc/shadow
```

file_permissions_etc_passwd

Verify Permissions on passwd File

To properly set the permissions of `/etc/passwd`, run the command:

```
$ sudo chmod 0644 /etc/passwd
```

file_owner_etc_passwd

Verify User Who Owns passwd File

To properly set the owner of `/etc/passwd`, run the command:

```
$ sudo chown root /etc/passwd
```

file_permissions_etc_shadow

Verify Permissions on shadow File

To properly set the permissions of `/etc/shadow`, run the command:

```
$ sudo chmod 0640 /etc/shadow
```

file_groupowner_etc_passwd

Verify Group Who Owns passwd File

To properly set the group owner of `/etc/passwd`, run the command:

```
$ sudo chgrp root /etc/passwd
```

file_permissions_etc_group

Verify Permissions on group File

To properly set the permissions of `/etc/passwd`, run the command:

```
$ sudo chmod 0644 /etc/passwd
```

file_groupowner_etc_gshadow

Verify Group Who Owns gshadow File

To properly set the group owner of `/etc/gshadow`, run the command:

```
$ sudo chgrp root /etc/gshadow
```

file_owner_etc_shadow

Verify User Who Owns shadow File

To properly set the owner of `/etc/shadow`, run the command:

```
$ sudo chown root /etc/shadow
```

file_permissions_ungroupowned

Ensure All Files Are Owned by a Group

If any files are not owned by a group, then the cause of their lack of group-ownership should be investigated. Following this, the files should be deleted or assigned to an appropriate group.

sysctl_fs_protected_hardlinks

Disallow creating hardlinks to a file you not own

To set the runtime status of the `fs.protected_hardlinks` kernel parameter, run the following command:

```
$ sudo sysctl -w fs.protected_hardlinks=1
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
fs.protected_hardlinks = 1
```

file_permissions_unauthorized_suid

Ensure All SUID Executables Are Authorized

The SUID (set user id) bit should be set only on files that were installed via authorized means. A straightforward means of identifying unauthorized SGID files is determine if any were not installed as part of an RPM package, which is cryptographically verified. Investigate the origin of any unpackaged SUID files.

dir_perms_world_writable_sticky_bits

Verify that All World-Writable Directories Have Sticky Bits Set

When the so-called 'sticky bit' is set on a directory, only the owner of a given file may remove that file from the directory. Without the sticky bit, any user with write access to a directory may remove any file in the directory. Setting the sticky bit prevents users from removing each other's files. In cases where there is no reason for a directory to be world-writable, a better solution is to remove that permission rather than to set the sticky bit. However, if a directory is used by a particular application, consult that application's documentation instead of blindly changing modes. To set the sticky bit on a world-writable directory *DIR*, run the following command:

```
$ sudo chmod +t DIR
```

no_files_unowned_by_user

Ensure All Files Are Owned by a User

If any files are not owned by a user, then the cause of their lack of ownership should be investigated. Following this, the files should be deleted or assigned to an appropriate user.

file_permissions_unauthorized_sgid

Ensure All SGID Executables Are Authorized

The SGID (set group id) bit should be set only on files that were installed via authorized means. A straightforward means of identifying unauthorized SGID files is determine if any were not installed as part of an RPM package, which is cryptographically verified. Investigate the origin of any unpackaged SGID files.

dir_perms_world_writable_system_owned

Ensure All World-Writable Directories Are Owned by a System Account

All directories in local partitions which are world-writable should be owned by root or another system account. If any world-writable directories are not owned by a system account, this should be investigated. Following this, the files should be deleted or assigned to an appropriate group.

file_permissions_unauthorized_world_writable

Ensure No World-Writable Files Exist

It is generally a good idea to remove global (other) write access to a file when it is discovered. However, check with documentation for specific applications before making changes. Also, monitor for recurring world-writable files, as these may be symptoms of a misconfigured application or user account. Finally, this applies to real files and not virtual files that are a part of pseudo file systems such as `sysfs` or `procfs`.

file_permissions_systemmap

Verify that local System.map file (if exists) is readable only by root

Files containing sensitive informations should be protected by restrictive permissions. Most of the time, there is no need that these files need to be read by any non-root user To properly set the permissions of `/boot/System.map-*`, run the command:

```
$ sudo chmod 0600 /boot/System.map-*
```

sysctl_fs_protected_symlinks

Disallow creating symlinks to a file you not own

To set the runtime status of the `fs.protected_symlinks` kernel parameter, run the following command:

```
$ sudo sysctl -w fs.protected_symlinks=1
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
fs.protected_symlinks = 1
```

Restrict Dynamic Mounting and Unmounting of Filesystems

Linux includes a number of facilities for the automated addition and removal of filesystems on a running system. These facilities may be necessary in many environments, but this capability also carries some risk -- whether direct risk from allowing users to introduce arbitrary filesystems, or risk that software flaws in the automated mount facility itself could allow an attacker to compromise the system.

This command can be used to list the types of filesystems that are available to the currently executing kernel:

```
$ find /lib/modules/`uname -r`/kernel/fs -type f -name '*.ko'
```

If these filesystems are not required then they can be explicitly disabled in a configuratio file in `/etc/modprobe.d`.

kernel_module_hfs_disabled

Disable Mounting of hfs

To configure the system to prevent the `hfs` kernel module from being loaded, add the following line to a file in the directory `/etc/modprobe.d`:

```
install hfs /bin/true
```

This effectively prevents usage of this uncommon filesystem.

kernel_module_usb-storage_disabled

Disable Modprobe Loading of USB Storage Driver

To prevent USB storage devices from being used, configure the kernel module loading system to prevent automatic loading of the USB storage driver. To configure the system to prevent the `usb-storage` kernel module from being loaded, add the following line to a file in the directory `/etc/modprobe.d`:

```
install usb-storage /bin/true
```

This will prevent the `modprobe` program from loading the `usb-storage` module, but will not prevent an administrator (or another program) from using the `insmod` program to load the module manually.

kernel_module_freevxfs_disabled

Disable Mounting of freevxfs

To configure the system to prevent the `freevxfs` kernel module from being loaded, add the following line to a file in the directory `/etc/modprobe.d`:

```
install freevxfs /bin/true
```

This effectively prevents usage of this uncommon filesystem.

kernel_module_udf_disabled

Disable Mounting of udf

To configure the system to prevent the `udf` kernel module from being loaded, add the following line to a file in the directory `/etc/modprobe.d`:

```
install udf /bin/true
```

This effectively prevents usage of this uncommon filesystem.

kernel_module_jffs2_disabled

Disable Mounting of jffs2

To configure the system to prevent the `jffs2` kernel module from being loaded, add the following line to a file in the directory `/etc/modprobe.d`:

```
install jffs2 /bin/true
```

This effectively prevents usage of this uncommon filesystem.

kernel_module_squashfs_disabled

Disable Mounting of squashfs

To configure the system to prevent the `squashfs` kernel module from being loaded, add the following line to a file in the directory `/etc/modprobe.d`:

```
install squashfs /bin/true
```

This effectively prevents usage of this uncommon filesystem.

kernel_module_hfsplus_disabled

Disable Mounting of hfsplus

To configure the system to prevent the `hfsplus` kernel module from being loaded, add the following line to a file in the directory `/etc/modprobe.d`:

```
install hfsplus /bin/true
```

This effectively prevents usage of this uncommon filesystem.

bios_assign_password

Assign Password to Prevent Changes to Boot Firmware Configuration

Assign a password to the system boot firmware (historically called BIOS on PC systems) to require a password for any configuration changes.

bios_disable_usb_boot

Disable Booting from USB Devices in Boot Firmware

Configure the system boot firmware (historically called BIOS on PC systems) to disallow booting from USB drives.

kernel_module_cramfs_disabled

Disable Mounting of cramfs

To configure the system to prevent the `cramfs` kernel module from being loaded, add the following line to a file in the directory `/etc/modprobe.d`:

```
install cramfs /bin/true
```

This effectively prevents usage of this uncommon filesystem.

grub2_nousb_argument

Disable Kernel Support for USB via Bootloader Configuration

All USB support can be disabled by adding the `nousb` argument to the kernel's boot loader configuration. To do so, append "nousb" to the kernel line in `/etc/default/grub` as shown:

```
kernel /vmlinuz-VERSION ro vga=ext root=/dev/VolGroup00/LogVol00 rhgb quiet nousb
```

service_autofs_disabled

Disable the Automounter

The `autofs` daemon mounts and unmounts filesystems, such as user home directories shared via NFS, on demand. In addition, autofs can be used to handle removable media, and the default configuration provides the cdrom device as `/misc/cd`. However, this method of providing access to removable media is not common, so autofs can almost always be disabled if NFS is not in use. Even if NFS is required, it may be possible to configure filesystem mounts statically by editing `/etc/fstab` rather than relying on the automounter.

The `autofs` service can be disabled with the following command:

```
$ sudo systemctl disable autofs.service
```

Restrict Programs from Dangerous Execution Patterns

The recommendations in this section are designed to ensure that the system's features to protect against potentially dangerous program execution are activated. These protections are applied at the system initialization or kernel level, and defend against certain types of badly-configured or compromised programs.

Disable Core Dumps

A core dump file is the memory image of an executable program when it was terminated by the operating system due to errant behavior. In most cases, only software developers legitimately need to access these files. The core dump files may also contain sensitive information, or unnecessarily occupy large amounts of disk space.

Once a hard limit is set in `/etc/security/limits.conf`, a user cannot increase that limit within his or her own session. If access to core dumps is required, consider restricting them to only certain users or groups. See the `limits.conf` man page for more information.

The core dumps of setuid programs are further protected. The `sysctl` variable `fs.suid_dumpable` controls whether the kernel allows core dumps from these programs at all. The default value of 0 is recommended.

sysctl_fs_suid_dumpable

Disable Core Dumps for SUID programs

To set the runtime status of the `fs.suid_dumpable` kernel parameter, run the following command:

```
$ sudo sysctl -w fs.suid_dumpable=0
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
fs.suid_dumpable = 0
```

disable_users_coredumps

Disable Core Dumps for All Users

To disable core dumps for all users, add the following line to `/etc/security/limits.conf`:

```
*     hard   core    0
```

Daemon Umask

The umask is a per-process setting which limits the default permissions for creation of new files and directories. The system includes initialization scripts which set the default umask for system daemons.

umask_for_daemons

Set Daemon Umask

The file `/etc/init.d/functions` includes initialization parameters for most or all daemons started at boot time. Many daemons on the system already individually restrict themselves to a umask of `077` in their own init scripts. By default, the umask of `022` is set which prevents creation of group- or world-writable files. To set the umask for daemons expected by the profile, edit the following line:

```
umask
```

Enable ExecShield

ExecShield describes kernel features that provide protection against exploitation of memory corruption errors such as buffer overflows. These features include random placement of the stack and other memory regions, prevention of execution in memory that should only hold data, and special handling of text buffers. These protections are enabled by default on 32-bit systems and controlled through `sysctl` variables `kernel.exec-shield` and `kernel.randomize_va_space`. On the latest 64-bit systems, `kernel.exec-shield` cannot be enabled or disabled with `sysctl`.

sysctl_kernel_randomize_va_space

Enable Randomized Layout of Virtual Address Space

To set the runtime status of the `kernel.randomize_va_space` kernel parameter, run the following command:

```
$ sudo sysctl -w kernel.randomize_va_space=2
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
kernel.randomize_va_space = 2
```

sysctl_kernel_kptr_restrict

Restrict Exposed Kernel Pointer Addresses Access

To set the runtime status of the `kernel.kptr_restrict` kernel parameter, run the following command:

```
$ sudo sysctl -w kernel.kptr_restrict=1
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
kernel.kptr_restrict = 1
```

sysctl_kernel_exec_shield

Enable ExecShield via sysctl

By default on Red Hat Enterprise Linux 7 64-bit systems, ExecShield is enabled and can only be disabled if the hardware does not support ExecShield or is disabled in `/etc/default/grub`. For Red Hat Enterprise Linux 7 32-bit systems, `sysctl` can be used to enable ExecShield.

Enable Execute Disable (XD) or No Execute (NX) Support on x86 Systems

Recent processors in the x86 family support the ability to prevent code execution on a per memory page basis. Generically and on AMD processors, this ability is called No Execute (NX), while on Intel processors it is called Execute Disable (XD). This ability can help prevent exploitation of buffer overflow vulnerabilities and should be activated whenever possible. Extra steps must be taken to ensure that this protection is enabled, particularly on 32-bit x86 systems. Other processors, such as Itanium and POWER, have included such support since inception and the standard kernel for those platforms supports the feature. This is enabled by default on the latest Red Hat and Fedora systems if supported by the hardware.

install_PAE_kernel_on_x86-32

Install PAE Kernel on Supported 32-bit x86 Systems

Systems that are using the 64-bit x86 kernel package do not need to install the kernel-PAE package because the 64-bit x86 kernel already includes this support. However, if the system is 32-bit and also supports the PAE and NX features as determined in the previous section, the kernel-PAE package should be installed to enable XD or NX support. The `kernel-PAE` package can be installed with the following command:

```
$ sudo yum install kernel-PAE
```

The installation process should also have configured the bootloader to load the new kernel at boot. Verify this after reboot and modify `/etc/default/grub` if necessary.

bios_enable_execution_restrictions

Enable NX or XD Support in the BIOS

Reboot the system and enter the BIOS or Setup configuration menu. Navigate the BIOS configuration menu and make sure that the option is enabled. The setting may be located under a Security section. Look for Execute Disable (XD) on Intel-based systems and No Execute (NX) on AMD-based systems.

Memory Poisoning

Memory Poisoning consists of writing a special value to uninitialized or freed memory. Poisoning can be used as a mechanism to prevent leak of information and detection of corrupted memory.

grub2_slub_debug_argument

Enable SLUB/SLAB allocator poisoning

To enable poisoning of SLUB/SLAB objects, add the argument `slub_debug=P` to the default GRUB 2 command line for the Linux operating system in `/etc/default/grub`, in the manner below:

```
GRUB_CMDLINE_LINUX="slub_debug=P"
```

grub2_page_poison_argument

Enable page allocator poisoning

To enable poisoning of free pages, add the argument `page_poison=1` to the default GRUB 2 command line for the Linux operating system in `/etc/default/grub`, in the manner below:

```
GRUB_CMDLINE_LINUX="page_poison=1"
```

grub2_vsyscall_argument

Disable vsyscalls

To disable use of virtual syscalls, add the argument `vsyscall=none` to the default GRUB 2 command line for the Linux operating system in `/etc/default/grub`, in the manner below:

```
GRUB_CMDLINE_LINUX="vsyscall=none"
```

sysctl_kernel_dmesg_restrict

Restrict Access to Kernel Message Buffer

To set the runtime status of the `kernel.dmesg_restrict` kernel parameter, run the following command:

```
$ sudo sysctl -w kernel.dmesg_restrict=1
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
kernel.dmesg_restrict = 1
```

sysctl_kernel_yama_ptrace_scope

Restrict usage of ptrace to descendant processes

To set the runtime status of the `kernel.yama.ptrace_scope` kernel parameter, run the following command:

```
$ sudo sysctl -w kernel.yama.ptrace_scope=1
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
kernel.yama.ptrace_scope = 1
```

sysctl_kernel_kexec_load_disabled

Disable kernel image loading

To set the runtime status of the `kernel.kexec_load_disabled` kernel parameter, run the following command:

```
$ sudo sysctl -w kernel.kexec_load_disabled=1
```

If this is not the system default value, add the following line to a file in the directory `/etc/sysctl.d`:

```
kernel.kexec_load_disabled = 1
```

Restrict Partition Mount Options

System partitions can be mounted with certain options that limit what files on those partitions can do. These options are set in the `/etc/fstab` configuration file, and can be used to make certain types of malicious behavior more difficult.

mount_option_nodev_nonroot_local_partitions

Add nodev Option to Non-Root Local Partitions

The `nodev` mount option prevents files from being interpreted as character or block devices. Legitimate character and block devices should exist only in the `/dev` directory on the root partition or within chroot jails built for system services. Add the `nodev` option to the fourth column of `/etc/fstab` for the line which controls mounting of any non-root local partitions.

mount_option_nosuid_removable_partitions

Add nosuid Option to Removable Media Partitions

The `nosuid` mount option prevents set-user-identifier (SUID) and set-group-identifier (SGID) permissions from taking effect. These permissions allow users to execute binaries with the same permissions as the owner and group of the file respectively. Users should not be allowed to introduce SUID and SGID files into the system via partitions mounted from removeable media. Add the `nosuid` option to the fourth column of `/etc/fstab` for the line which controls mounting of any removable media partitions.

mount_option_noexec_removable_partitions

Add noexec Option to Removable Media Partitions

The `noexec` mount option prevents the direct execution of binaries on the mounted filesystem. Preventing the direct execution of binaries from removable media (such as a USB key) provides a defense against malicious software that may be present on such untrusted media. Add the `noexec` option to the fourth column of `/etc/fstab` for the line which controls mounting of any removable media partitions.

mount_option_nodev_removable_partitions

Add nodev Option to Removable Media Partitions

The `nodev` mount option prevents files from being interpreted as character or block devices. Legitimate character and block devices should exist only in the `/dev` directory on the root partition or within chroot jails built for system services. Add the `nodev` option to the fourth column of `/etc/fstab` for the line which controls mounting of any removable media partitions.

mount_option_dev_shm_noexec

Add noexec Option to /dev/shm

The `noexec` mount option can be used to prevent binaries from being executed out of `/dev/shm`. It can be dangerous to allow the execution of binaries from world-writable temporary storage directories such as `/dev/shm`. Add the `noexec` option to the fourth column of `/etc/fstab` for the line which controls mounting of `/dev/shm`.

mount_option_tmp_nosuid

Add nosuid Option to /tmp

The `nosuid` mount option can be used to prevent execution of setuid programs in `/tmp`. The SUID and SGID permissions should not be required in these world-writable directories. Add the `nosuid` option to the fourth column of `/etc/fstab` for the line which controls mounting of `/tmp`.

mount_option_tmp_nodev

Add nodev Option to /tmp

The `nodev` mount option can be used to prevent device files from being created in `/tmp`. Legitimate character and block devices should not exist within temporary directories like `/tmp`. Add the `nodev` option to the fourth column of `/etc/fstab` for the line which controls mounting of `/tmp`.

mount_option_var_tmp_noexec

Add noexec Option to /var/tmp

The `noexec` mount option can be used to prevent binaries from being executed out of `/var/tmp`. Add the `noexec` option to the fourth column of `/etc/fstab` for the line which controls mounting of `/var/tmp`.

mount_option_home_nosuid

Add nosuid Option to /home

The `nosuid` mount option can be used to prevent execution of setuid programs in `/home`. The SUID and SGID permissions should not be required in these user data directories. Add the `nosuid` option to the fourth column of `/etc/fstab` for the line which controls mounting of `/home`.

mount_option_tmp_noexec

Add noexec Option to /tmp

The `noexec` mount option can be used to prevent binaries from being executed out of `/tmp`. Add the `noexec` option to the fourth column of `/etc/fstab` for the line which controls mounting of `/tmp`.

mount_option_dev_shm_nosuid

Add nosuid Option to /dev/shm

The `nosuid` mount option can be used to prevent execution of setuid programs in `/dev/shm`. The SUID and SGID permissions should not be required in these world-writable directories. Add the `nosuid` option to the fourth column of `/etc/fstab` for the line which controls mounting of `/dev/shm`.

mount_option_var_tmp_nodev

Add nodev Option to /var/tmp

The `nodev` mount option can be used to prevent device files from being created in `/var/tmp`. Legitimate character and block devices should not exist within temporary directories like `/var/tmp`. Add the `nodev` option to the fourth column of `/etc/fstab` for the line which controls mounting of `/var/tmp`.

mount_option_var_tmp_bind

Bind Mount /var/tmp To /tmp

The `/var/tmp` directory is a world-writable directory. Bind-mount it to `/tmp` in order to consolidate temporary storage into one location protected by the same techniques as `/tmp`. To do so, edit `/etc/fstab` and add the following line:

```
/tmp     /var/tmp     none     rw,nodev,noexec,nosuid,bind     0 0
```

See the `mount(8)` man page for further explanation of bind mounting.

mount_option_dev_shm_nodev

Add nodev Option to /dev/shm

The `nodev` mount option can be used to prevent creation of device files in `/dev/shm`. Legitimate character and block devices should not exist within temporary directories like `/dev/shm`. Add the `nodev` option to the fourth column of `/etc/fstab` for the line which controls mounting of `/dev/shm`.

mount_option_home_nodev

Add nodev Option to /home

The `nodev` mount option can be used to prevent device files from being created in `/home`. Legitimate character and block devices should exist only in the `/dev` directory on the root partition or within chroot jails built for system services. Add the `nodev` option to the fourth column of `/etc/fstab` for the line which controls mounting of `/home`.

mount_option_var_tmp_nosuid

Add nosuid Option to /var/tmp

The `nosuid` mount option can be used to prevent execution of setuid programs in `/var/tmp`. The SUID and SGID permissions should not be required in these world-writable directories. Add the `nosuid` option to the fourth column of `/etc/fstab` for the line which controls mounting of `/var/tmp`.

Protect Random-Number Entropy Pool

The I/O operations of the Linux kernel block layer due to their inherently unpredictable execution times have been traditionally considered as a reliable source to contribute to random-number entropy pool of the Linux kernel. This has changed with introduction of solid-state storage devices (SSDs) though.

kernel_disable_entropy_contribution_for_solid_state_drives

Ensure Solid State Drives Do Not Contribute To Random-Number Entropy Pool

For each solid-state drive on the system, run:

```
# echo 0 > /sys/block/DRIVE/queue/add_random
```

Set Boot Loader Password

During the boot process, the boot loader is responsible for starting the execution of the kernel and passing options to it. The boot loader allows for the selection of different kernels - possibly on different partitions or media. The default Red Hat Enterprise Linux 8 boot loader for x86 systems is called GRUB2. Options it can pass to the kernel include *single-user mode*, which provides root access without any authentication, and the ability to disable SELinux. To prevent local users from modifying the boot parameters and endangering security, protect the boot loader configuration with a password and ensure its configuration file's permissions are set properly.

file_groupowner_efi_grub2_cfg

Verify the UEFI Boot Loader grub.cfg Group Ownership

The file `/boot/efi/EFI/redhat/grub.cfg` should be group-owned by the `root` group to prevent destruction or modification of the file. To properly set the group owner of `/boot/efi/EFI/redhat/grub.cfg`, run the command:

```
$ sudo chgrp root /boot/efi/EFI/redhat/grub.cfg
```

grub2_password

Set Boot Loader Password in grub2

The grub2 boot loader should have a superuser account and password protection enabled to protect boot-time settings.

To do so, select a superuser account name and password and and modify the `/etc/grub.d/01_users` configuration file with the new account name.

Since plaintext passwords are a security risk, generate a hash for the pasword by running the following command:

```
$ grub2-setpassword
```

When prompted, enter the password that was selected.

NOTE: It is recommended not to use common administrator account names like root, admin, or administrator for the grub2 superuser account.

Change the superuser to a different username (The default is 'root').

```
$ sed -i s/root/bootuser/g /etc/grub.d/01_users
```

To meet FISMA Moderate, the bootloader superuser account and password MUST differ from the root account and password. Once the superuser account and password have been added, update the `grub.cfg` file by running:

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

NOTE: Do NOT manually add the superuser account and password to the `grub.cfg` file as the grub2-mkconfig command overwrites this file.

file_groupowner_grub2_cfg

Verify /boot/grub2/grub.cfg Group Ownership

The file `/boot/grub2/grub.cfg` should be group-owned by the `root` group to prevent destruction or modification of the file. To properly set the group owner of `/boot/grub2/grub.cfg`, run the command:

```
$ sudo chgrp root /boot/grub2/grub.cfg
```

uefi_no_removeable_media

UEFI Boat Loader Is Not Installed On Removeable Media

The system must not allow removable media to be used as the boot loader. Remove alternate methods of booting the system from removable media. `usb0`, `cd`, `fd0`, etc. are some examples of removeable media which should not exist in the line:

```
set root='hd0,msdos1'
```

file_owner_efi_grub2_cfg

Verify the UEFI Boot Loader grub.cfg User Ownership

The file `/boot/efi/EFI/redhat/grub.cfg` should be owned by the `root` user to prevent destruction or modification of the file. To properly set the owner of `/boot/efi/EFI/redhat/grub.cfg`, run the command:

```
$ sudo chown root /boot/efi/EFI/redhat/grub.cfg
```

grub2_enable_iommu_force

IOMMU configuration directive

On x86 architecture supporting VT-d, the IOMMU manages the access control policy between the hardware devices and some of the system critical units such as the memory.

grub2_no_removeable_media

Boat Loader Is Not Installed On Removeable Media

The system must not allow removable media to be used as the boot loader. Remove alternate methods of booting the system from removable media. `usb0`, `cd`, `fd0`, etc. are some examples of removeable media which should not exist in the line:

```
set root='hd0,msdos1'
```

grub2_uefi_password

Set the UEFI Boot Loader Password

The grub2 boot loader should have a superuser account and password protection enabled to protect boot-time settings.

To do so, select a superuser account name and password and and modify the `/etc/grub.d/01_users` configuration file with the new account name.

Since plaintext passwords are a security risk, generate a hash for the pasword by running the following command:

```
$ grub2-setpassword
```

When prompted, enter the password that was selected.

NOTE: It is recommended not to use common administrator account names like root, admin, or administrator for the grub2 superuser account.

Change the superuser to a different username (The default is 'root').

```
$ sed -i s/root/bootuser/g /etc/grub.d/01_users
```

To meet FISMA Moderate, the bootloader superuser account and password MUST differ from the root account and password. Once the superuser account and password have been added, update the `grub.cfg` file by running:

```
grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```

NOTE: Do NOT manually add the superuser account and password to the `grub.cfg` file as the grub2-mkconfig command overwrites this file.

file_permissions_grub2_cfg

Verify /boot/grub2/grub.cfg Permissions

File permissions for `/boot/grub2/grub.cfg` should be set to 600. To properly set the permissions of `/boot/grub2/grub.cfg`, run the command:

```
$ sudo chmod 600 /boot/grub2/grub.cfg
```

file_permissions_efi_grub2_cfg

Verify the UEFI Boot Loader grub.cfg Permissions

File permissions for `/boot/efi/EFI/redhat/grub.cfg` should be set to 700. To properly set the permissions of `/boot/efi/EFI/redhat/grub.cfg`, run the command:

```
$ sudo chmod 700 /boot/efi/EFI/redhat/grub.cfg
```

file_owner_grub2_cfg

Verify /boot/grub2/grub.cfg User Ownership

The file `/boot/grub2/grub.cfg` should be owned by the `root` user to prevent destruction or modification of the file. To properly set the owner of `/boot/grub2/grub.cfg`, run the command:

```
$ sudo chown root /boot/grub2/grub.cfg
```

Services

The best protection against vulnerable software is running less software. This section describes how to review the software which Red Hat Enterprise Linux 8 installs on a system and disable software which is not needed. It then enumerates the software packages installed on a default Red Hat Enterprise Linux 8 system and provides guidance about which ones can be safely disabled.

Red Hat Enterprise Linux 8 provides a convenient minimal install option that essentially installs the bare necessities for a functional system. When building Red Hat Enterprise Linux 8 systems, it is highly recommended to select the minimal packages and then build up the system from there.

Network Routing

A router is a very desirable target for a potential adversary because they fulfill a variety of infrastructure networking roles such as access to network segments, gateways to other networks, filtering, etc. Therefore, if one is required, the system acting as a router should be dedicated to that purpose alone and be stored in a physically secure location. The system's default routing software is Quagga, and provided in an RPM package of the same name.

Disable Quagga if Possible

If Quagga was installed and activated, but the system does not need to act as a router, then it should be disabled and removed.

service_zebra_disabled

Disable Quagga Service

The `zebra` service can be disabled with the following command:

```
$ sudo systemctl disable zebra.service
```

package_quagga_removed

Uninstall quagga Package

The `quagga` package can be removed with the following command:

```
$ sudo yum erase quagga
```

APT service configuration

The apt service manage the package management and update of the whole system. Its configuration need to be properly defined to ensure efficient security updates, packages and repository authentication and proper lifecycle management.

apt_conf_disallow_unauthenticated

Disable unauthenticated repositories in APT configuration

Unauthenticated repositories should not be used for updates.

apt_sources_list_official

Ensure that official distribution repositories are used

Check that official Debian repositories, including security repository, are configured in apt.

DNS Server

Most organizations have an operational need to run at least one nameserver. However, there are many common attacks involving DNS server software, and this server software should be disabled on any system on which it is not needed.

Isolate DNS from Other Services

This section discusses mechanisms for preventing the DNS server from interfering with other services. This is done both to protect the remainder of the network should a nameserver be compromised, and to make direct attacks on nameservers more difficult.

Run DNS Software in a chroot Jail

Install the `bind-chroot` package:

```
$ sudo yum install bind-chroot
```

Place a valid named.conf file inside the chroot jail:

```
$ sudo cp /etc/named.conf /var/named/chroot/etc/named.conf
$ sudo chown root:root /var/named/chroot/etc/named.conf
$ sudo chmod 644 /var/named/chroot/etc/named.conf
```

Create and populate an appropriate zone directory within the jail, based on the options directive. If your `named.conf` includes:

```
options {
directory "/path/to/DIRNAME ";
...
}
```

then copy that directory and its contents from the original zone directory:

```
$ sudo cp -r /path/to/DIRNAME /var/named/chroot/DIRNAME
```

Add or correct the following line within `/etc/sysconfig/named`:

```
ROOTDIR=/var/named/chroot
```

Run DNS Software on Dedicated Servers

Since DNS is a high-risk service which must frequently be made available to the entire Internet, it is strongly recommended that no other services be offered by systems which act as organizational DNS servers.

Protect DNS Data from Tampering or Attack

This section discusses DNS configuration options which make it more difficult for attackers to gain access to private DNS data or to modify DNS data.

Use Views to Partition External and Internal Information

If it is not possible to run external and internal nameservers on separate physical systems, run BIND9 and simulate this feature using views. Edit `/etc/named.conf`. Add or correct the following directives (where SUBNET is the numerical IP representation of your organization in the form xxx.xxx.xxx.xxx/xx):

```
acl internal {
  SUBNET ;
  localhost;
};
view "internal-view" {
  match-clients { internal; };
  zone "." IN {
    type hint;
    file "db.cache";
  };
  zone "internal.example.com " IN {
    ...
  };
};

view "external-view" {
  match-clients { any; };
  recursion no;
  zone "example.com " IN {
    ...
  };
};
```

Run Separate DNS Servers for External and Internal Queries

Is it possible to run external and internal nameservers on separate systems? If so, follow the configuration guidance in this section. On the external nameserver, edit `/etc/named.conf` to add or correct the following directives:

```
options {
  allow-query { any; };
  recursion no;
  ...
};
zone "example.com " IN {
  ...
};
```

On the internal nameserver, edit `/etc/named.conf`. Add or correct the following directives, where SUBNET is the numerical IP representation of your organization in the form xxx.xxx.xxx.xxx/xx:

```
acl internal {
  SUBNET ;
  localhost;
};
options {
  allow-query { internal; };
  ...
};
zone "internal.example.com " IN {
  ...
};
```

dns_server_disable_zone_transfers

Disable Zone Transfers from the Nameserver

Is it necessary for a secondary nameserver to receive zone data via zone transfer from the primary server? If not, follow the instructions in this section. If so, see the next section for instructions on protecting zone transfers. Add or correct the following directive within `/etc/named.conf`:

```
options {
  allow-transfer { none; };
  ...
}
```

dns_server_disable_dynamic_updates

Disable Dynamic Updates

Is there a mission-critical reason to enable the risky dynamic update functionality? If not, edit `/etc/named.conf`. For each zone specification, correct the following directive if necessary:

```
zone "example.com " IN {
  allow-update { none; };
  ...
};
```

dns_server_authenticate_zone_transfers

Authenticate Zone Transfers

If it is necessary for a secondary nameserver to receive zone data via zone transfer from the primary server, follow the instructions here. Use dnssec-keygen to create a symmetric key file in the current directory:

```
$ cd /tmp
$ sudo dnssec-keygen -a HMAC-MD5 -b 128 -n HOST dns.example.com
Kdns.example.com .+aaa +iiiii
```

This output is the name of a file containing the new key. Read the file to find the base64-encoded key string:

```
$ sudo cat Kdns.example.com .+NNN +MMMMM .key
dns.example.com IN KEY 512 3 157 base64-key-string
```

Add the directives to `/etc/named.conf` on the primary server:

```
key zone-transfer-key {
  algorithm hmac-md5;
  secret "base64-key-string ";
};
zone "example.com " IN {
  type master;
  allow-transfer { key zone-transfer-key; };
  ...
};
```

Add the directives below to `/etc/named.conf` on the secondary nameserver:

```
key zone-transfer-key {
  algorithm hmac-md5;
  secret "base64-key-string ";
};

server IP-OF-MASTER {
  keys { zone-transfer-key; };
};

zone "example.com " IN {
  type slave;
  masters { IP-OF-MASTER ; };
  ...
};
```

Disable DNS Server

DNS software should be disabled on any systems which does not need to be a nameserver. Note that the BIND DNS server software is not installed on Red Hat Enterprise Linux 8 by default. The remainder of this section discusses secure configuration of systems which must be nameservers.

service_named_disabled

Disable named Service

The `named` service can be disabled with the following command:

```
$ sudo systemctl disable named.service
```

package_bind_removed

Uninstall bind Package

The `named` service is provided by the `bind` package. The `bind` package can be removed with the following command:

```
$ sudo yum erase bind
```

DHCP

The Dynamic Host Configuration Protocol (DHCP) allows systems to request and obtain an IP address and other configuration parameters from a server.

This guide recommends configuring networking on clients by manually editing the appropriate files under `/etc/sysconfig`. Use of DHCP can make client systems vulnerable to compromise by rogue DHCP servers, and should be avoided unless necessary. If using DHCP is necessary, however, there are best practices that should be followed to minimize security risk.

Disable DHCP Client

DHCP is the default network configuration method provided by the system installer, and common on many networks. Nevertheless, manual management of IP addresses for systems implies a greater degree of management and accountability for network activity.

sysconfig_networking_bootproto_ifcfg

Disable DHCP Client in ifcfg

For each interface on the system (e.g. eth0), edit `/etc/sysconfig/network-scripts/ifcfg-interface` and make the following changes:

- Correct the BOOTPROTO line to read:

```
BOOTPROTO=none
```

- Add or correct the following lines, substituting the appropriate values based on your site's addressing scheme:

```
NETMASK=255.255.255.0
IPADDR=192.168.1.2
GATEWAY=192.168.1.1
```

Disable DHCP Server

The DHCP server `dhcpd` is not installed or activated by default. If the software was installed and activated, but the system does not need to act as a DHCP server, it should be disabled and removed.

service_dhcpd_disabled

Disable DHCP Service

The `dhcpd` service should be disabled on any system that does not need to act as a DHCP server. The `dhcpd` service can be disabled with the following command:

```
$ sudo systemctl disable dhcpd.service
```

package_dhcp_removed

Uninstall DHCP Server Package

If the system does not need to act as a DHCP server, the dhcp package can be uninstalled. The `dhcp` package can be removed with the following command:

```
$ sudo yum erase dhcp
```

Configure DHCP Client if Necessary

If DHCP must be used, then certain configuration changes can minimize the amount of information it receives and applies from the network, and thus the amount of incorrect information a rogue DHCP server could successfully distribute. For more information on configuring dhclient, see the `dhclient(8)` and `dhclient.conf(5)` man pages.

dhcp_client_restrict_options

Minimize the DHCP-Configured Options

Create the file `/etc/dhcp/dhclient.conf`, and add an appropriate setting for each of the ten configuration settings which can be obtained via DHCP. For each setting, do one of the following: If the setting should *not* be configured remotely by the DHCP server, select an appropriate static value, and add the line:

```
supersede setting value;
```

If the setting should be configured remotely by the DHCP server, add the lines:

```
request setting;
require setting;
```

For example, suppose the DHCP server should provide only the IP address itself and the subnet mask. Then the entire file should look like:

```
supersede domain-name "example.com";
supersede domain-name-servers 192.168.1.2;
supersede nis-domain "";
supersede nis-servers "";
supersede ntp-servers "ntp.example.com ";
supersede routers 192.168.1.1;
supersede time-offset -18000;
request subnet-mask;
require subnet-mask;
```

Configure DHCP Server

If the system must act as a DHCP server, the configuration information it serves should be minimized. Also, support for other protocols and DNS-updating schemes should be explicitly disabled unless needed. The configuration file for dhcpd is called `/etc/dhcp/dhcpd.conf`. The file begins with a number of global configuration options. The remainder of the file is divided into sections, one for each block of addresses offered by dhcpd, each of which contains configuration options specific to that address block.

dhcp_server_minimize_served_info

Minimize Served Information

Edit /etc/dhcp/dhcpd.conf. Examine each address range section within the file, and ensure that the following options are not defined unless there is an operational need to provide this information via DHCP:

```
option domain-name
option domain-name-servers
option nis-domain
option nis-servers
option ntp-servers
option routers
option time-offset
```

dhcp_server_deny_bootp

Deny BOOTP Queries

Unless your network needs to support older BOOTP clients, disable support for the bootp protocol by adding or correcting the global option:

```
deny bootp;
```

dhcp_server_configure_logging

Configure Logging

Ensure that the following line exists in `/etc/rsyslog.conf`:

```
daemon.*          /var/log/daemon.log
```

Configure logwatch or other log monitoring tools to summarize error conditions reported by the dhcpd process.

dhcp_server_deny_decline

Deny Decline Messages

Edit `/etc/dhcp/dhcpd.conf` and add or correct the following global option to prevent the DHCP server from responding the DHCPDECLINE messages, if possible:

```
deny declines;
```

dhcp_server_disable_ddns

Do Not Use Dynamic DNS

To prevent the DHCP server from receiving DNS information from clients, edit `/etc/dhcp/dhcpd.conf`, and add or correct the following global option:

```
ddns-update-style none;
```

Base Services

This section addresses the base services that are installed on a Red Hat Enterprise Linux 8 default installation which are not covered in other sections. Some of these services listen on the network and should be treated with particular discretion. Other services are local system utilities that may or may not be extraneous. In general, system services should be disabled if not required.

service_sysstat_disabled

Disable System Statistics Reset Service (sysstat)

The `sysstat` service resets various I/O and CPU performance statistics to zero in order to begin counting from a fresh state at boot time. The `sysstat` service can be disabled with the following command:

```
$ sudo systemctl disable sysstat.service
```

service_mdmonitor_disabled

Disable Software RAID Monitor (mdmonitor)

The `mdmonitor` service is used for monitoring a software RAID array; hardware RAID setups do not use this service. The `mdmonitor` service can be disabled with the following command:

```
$ sudo systemctl disable mdmonitor.service
```

service_certmonger_disabled

Disable Certmonger Service (certmonger)

Certmonger is a D-Bus based service that attempts to simplify interaction with certifying authorities on networks which use public-key infrastructure. It is often combined with Red Hat's IPA (Identity Policy Audit) security information management solution to aid in the management of certificates. The `certmonger` service can be disabled with the following command:

```
$ sudo systemctl disable certmonger.service
```

service_cgconfig_disabled

Disable Control Group Config (cgconfig)

Control groups allow an administrator to allocate system resources (such as CPU, memory, network bandwidth, etc) among a defined group (or groups) of processes executing on a system. The `cgconfig` daemon starts at boot and establishes the predefined control groups. The `cgconfig` service can be disabled with the following command:

```
$ sudo systemctl disable cgconfig.service
```

package_psacct_installed

Install the psacct package

The process accounting service, `psacct`, works with programs including `acct` and `ac` to allow system administrators to view user activity, such as commands issued by users of the system. The `psacct` package can be installed with the following command:

```
$ sudo yum install psacct
```

service_netconsole_disabled

Disable Network Console (netconsole)

The `netconsole` service is responsible for loading the netconsole kernel module, which logs kernel printk messages over UDP to a syslog server. This allows debugging of problems where disk logging fails and serial consoles are impractical. The `netconsole` service can be disabled with the following command:

```
$ sudo systemctl disable netconsole.service
```

service_irqbalance_enabled

Enable IRQ Balance (irqbalance)

The `irqbalance` service optimizes the balance between power savings and performance through distribution of hardware interrupts across multiple processors. The `irqbalance` service can be enabled with the following command:

```
$ sudo systemctl enable irqbalance.service
```

service_kdump_disabled

Disable KDump Kernel Crash Analyzer (kdump)

The `kdump` service provides a kernel crash dump analyzer. It uses the `kexec` system call to boot a secondary kernel ("capture" kernel) following a system crash, which can load information from the crashed kernel for analysis. The `kdump` service can be disabled with the following command:

```
$ sudo systemctl disable kdump.service
```

service_qpidd_disabled

Disable Apache Qpid (qpidd)

The `qpidd` service provides high speed, secure, guaranteed delivery services. It is an implementation of the Advanced Message Queuing Protocol. By default the qpidd service will bind to port 5672 and listen for connection attempts. The `qpidd` service can be disabled with the following command:

```
$ sudo systemctl disable qpidd.service
```

service_quota_nld_disabled

Disable Quota Netlink (quota_nld)

The `quota_nld` service provides notifications to users of disk space quota violations. It listens to the kernel via a netlink socket for disk quota violations and notifies the appropriate user of the violation using D-Bus or by sending a message to the terminal that the user has last accessed. The `quota_nld` service can be disabled with the following command:

```
$ sudo systemctl disable quota_nld.service
```

service_psacct_enabled

Enable Process Accounting (psacct)

The process accounting service, `psacct`, works with programs including `acct` and `ac` to allow system administrators to view user activity, such as commands issued by users of the system. The `psacct` service can be enabled with the following command:

```
$ sudo systemctl enable psacct.service
```

service_ntpdate_disabled

Disable ntpdate Service (ntpdate)

The `ntpdate` service sets the local hardware clock by polling NTP servers when the system boots. It synchronizes to the NTP servers listed in `/etc/ntp/step-tickers` or `/etc/ntp.conf` and then sets the local hardware clock to the newly synchronized system time. The `ntpdate` service can be disabled with the following command:

```
$ sudo systemctl disable ntpdate.service
```

service_saslauthd_disabled

Disable Cyrus SASL Authentication Daemon (saslauthd)

The `saslauthd` service handles plaintext authentication requests on behalf of the SASL library. The service isolates all code requiring superuser privileges for SASL authentication into a single process, and can also be used to provide proxy authentication services to clients that do not understand SASL based authentication. The `saslauthd` service can be disabled with the following command:

```
$ sudo systemctl disable saslauthd.service
```

service_portreserve_disabled

Disable Portreserve (portreserve)

The `portreserve` service is a TCP port reservation utility that can be used to prevent portmap from binding to well known TCP ports that are required for other services. The `portreserve` service can be disabled with the following command:

```
$ sudo systemctl disable portreserve.service
```

service_rhnsd_disabled

Disable Red Hat Network Service (rhnsd)

The Red Hat Network service automatically queries Red Hat Network servers to determine whether there are any actions that should be executed, such as package updates. This only occurs if the system was registered to an RHN server or satellite and managed as such. The `rhnsd` service can be disabled with the following command:

```
$ sudo systemctl disable rhnsd.service
```

service_cpupower_disabled

Disable CPU Speed (cpupower)

The `cpupower` service can adjust the clock speed of supported CPUs based upon the current processing load thereby conserving power and reducing heat. The `cpupower` service can be disabled with the following command:

```
$ sudo systemctl disable cpupower.service
```

service_messagebus_disabled

Disable D-Bus IPC Service (messagebus)

D-Bus provides an IPC mechanism used by a growing list of programs, such as those used for Gnome, Bluetooth, and Avahi. Due to these dependencies, disabling D-Bus may not be practical for many systems. The `messagebus` service can be disabled with the following command:

```
$ sudo systemctl disable messagebus.service
```

service_abrtd_disabled

Disable Automatic Bug Reporting Tool (abrtd)

The Automatic Bug Reporting Tool (`abrtd`) daemon collects and reports crash data when an application crash is detected. Using a variety of plugins, abrtd can email crash reports to system administrators, log crash reports to files, or forward crash reports to a centralized issue tracking system such as RHTSupport. The `abrtd` service can be disabled with the following command:

```
$ sudo systemctl disable abrtd.service
```

service_rhsmcertd_disabled

Disable Red Hat Subscription Manager Daemon (rhsmcertd)

The Red Hat Subscription Manager (rhsmcertd) periodically checks for changes in the entitlement certificates for a registered system and updates it accordingly. The `rhsmcertd` service can be disabled with the following command:

```
$ sudo systemctl disable rhsmcertd.service
```

service_oddjobd_disabled

Disable Odd Job Daemon (oddjobd)

The `oddjobd` service exists to provide an interface and access control mechanism through which specified privileged tasks can run tasks for unprivileged client applications. Communication with `oddjobd` through the system message bus. The `oddjobd` service can be disabled with the following command:

```
$ sudo systemctl disable oddjobd.service
```

service_acpid_disabled

Disable Advanced Configuration and Power Interface (acpid)

The Advanced Configuration and Power Interface Daemon (`acpid`) dispatches ACPI events (such as power/reset button depressed) to userspace programs. The `acpid` service can be disabled with the following command:

```
$ sudo systemctl disable acpid.service
```

service_cgred_disabled

Disable Control Group Rules Engine (cgred)

The `cgred` service moves tasks into control groups according to parameters set in the `/etc/cgrules.conf` configuration file. The `cgred` service can be disabled with the following command:

```
$ sudo systemctl disable cgred.service
```

service_smartd_disabled

Disable SMART Disk Monitoring Service (smartd)

SMART (Self-Monitoring, Analysis, and Reporting Technology) is a feature of hard drives that allows them to detect symptoms of disk failure and relay an appropriate warning. The `smartd` service can be disabled with the following command:

```
$ sudo systemctl disable smartd.service
```

package_abrt_removed

Uninstall Automatic Bug Reporting Tool (abrt)

The Automatic Bug Reporting Tool (`abrt`) collects and reports crash data when an application crash is detected. Using a variety of plugins, abrt can email crash reports to system administrators, log crash reports to files, or forward crash reports to a centralized issue tracking system such as RHTSupport. The `abrt` package can be removed with the following command:

```
$ sudo yum erase abrt
```

service_rdisc_disabled

Disable Network Router Discovery Daemon (rdisc)

The `rdisc` service implements the client side of the ICMP Internet Router Discovery Protocol (IRDP), which allows discovery of routers on the local subnet. If a router is discovered then the local routing table is updated with a corresponding default route. By default this daemon is disabled. The `rdisc` service can be disabled with the following command:

```
$ sudo systemctl disable rdisc.service
```

Obsolete Services

This section discusses a number of network-visible services which have historically caused problems for system security, and for which disabling or severely limiting the service has been the best available guidance for some time. As a result of this, many of these services are not installed as part of Red Hat Enterprise Linux 8 by default.

Organizations which are running these services should switch to more secure equivalents as soon as possible. If it remains absolutely necessary to run one of these services for legacy reasons, care should be taken to restrict the service as much as possible, for instance by configuring host firewall software such as `iptables` to restrict access to the vulnerable service to only those remote hosts which have a known need to use it.

Chat/Messaging Services

The talk software makes it possible for users to send and receive messages across systems through a terminal session.

package_talk-server_removed

Uninstall talk-server Package

The `talk-server` package can be removed with the following command:

```
$ sudo yum erase talk-server
```

package_talk_removed

Uninstall talk Package

The `talk` package contains the client program for the Internet talk protocol, which allows the user to chat with other users on different systems. Talk is a communication program which copies lines from one terminal to the terminal of another user. The `talk` package can be removed with the following command:

```
$ sudo yum erase talk
```

Rlogin, Rsh, and Rexec

The Berkeley r-commands are legacy services which allow cleartext remote access and have an insecure trust model.

package_rsh_removed

Uninstall rsh Package

The `rsh` package contains the client commands for the rsh services

service_rlogin_disabled

Disable rlogin Service

The `rlogin` service, which is available with the `rsh-server` package and runs as a service through xinetd or separately as a systemd socket, should be disabled. If using xinetd, set `disable` to `yes` in `/etc/xinetd.d/rlogin`. The `rlogin` socket can be disabled with the following command:

```
$ sudo systemctl disable rlogin.socket
```

package_rsh-server_removed

Uninstall rsh-server Package

The `rsh-server` package can be removed with the following command:

```
$ sudo yum erase rsh-server
```

no_user_host_based_files

Remove User Host-Based Authentication Files

The `~/.shosts` (in each user's home directory) files list remote hosts and users that are trusted by the local system. To remove these files, run the following command to delete them from any location:

```
$ sudo find / -name '.shosts' -type f -delete
```

service_rexec_disabled

Disable rexec Service

The `rexec` service, which is available with the `rsh-server` package and runs as a service through xinetd or separately as a systemd socket, should be disabled. If using xinetd, set `disable` to `yes` in `/etc/xinetd.d/rexec`. The `rexec` socket can be disabled with the following command:

```
$ sudo systemctl disable rexec.socket
```

service_rsh_disabled

Disable rsh Service

The `rsh` service, which is available with the `rsh-server` package and runs as a service through xinetd or separately as a systemd socket, should be disabled. If using xinetd, set `disable` to `yes` in `/etc/xinetd.d/rsh`. The `rsh` socket can be disabled with the following command:

```
$ sudo systemctl disable rsh.socket
```

no_rsh_trust_files

Remove Rsh Trust Files

The files `/etc/hosts.equiv` and `~/.rhosts` (in each user's home directory) list remote hosts and users that are trusted by the local system when using the rshd daemon. To remove these files, run the following command to delete them from any location:

```
$ sudo rm /etc/hosts.equiv
```

```
$ rm ~/.rhosts
```

no_host_based_files

Remove Host-Based Authentication Files

The `shosts.equiv` file list remote hosts and users that are trusted by the local system. To remove these files, run the following command to delete them from any location:

```
$ sudo rm /[path]/[to]/[file]/shosts.equiv
```

Telnet

The telnet protocol does not provide confidentiality or integrity for information transmitted on the network. This includes authentication information such as passwords. Organizations which use telnet should be actively working to migrate to a more secure protocol.

service_telnet_disabled

Disable telnet Service

The `telnet` service configuration file `/etc/xinetd.d/telnet` is not created automatically. If it was created manually, check the `/etc/xinetd.d/telnet` file and ensure that `disable = no` is changed to read `disable = yes` as follows below:

```
# description: The telnet server serves telnet sessions; it uses \\
#       unencrypted username/password pairs for authentication.
service telnet
{
        flags           = REUSE
        socket_type     = stream

        wait            = no
        user            = root
        server          = /usr/sbin/in.telnetd
        log_on_failure  += USERID
        disable         = yes
}
```

If the `/etc/xinetd.d/telnet` file does not exist, make sure that the activation of the `telnet` service on system boot is disabled via the following command: The `rexec` socket can be disabled with the following command:

```
$ sudo systemctl disable rexec.socket
```

package_telnet_removed

Remove telnet Clients

The telnet client allows users to start connections to other systems via the telnet protocol.

package_telnet-server_removed

Uninstall telnet-server Package

The `telnet-server` package can be removed with the following command:

```
$ sudo yum erase telnet-server
```

NIS

The Network Information Service (NIS), also known as 'Yellow Pages' (YP), and its successor NIS+ have been made obsolete by Kerberos, LDAP, and other modern centralized authentication services. NIS should not be used because it suffers from security problems inherent in its design, such as inadequate protection of important authentication information.

package_ypbind_removed

Remove NIS Client

The Network Information Service (NIS), formerly known as Yellow Pages, is a client-server directory service protocol used to distribute system configuration files. The NIS client (`ypbind`) was used to bind a system to an NIS server and receive the distributed configuration files.

service_ypbind_disabled

Disable ypbind Service

The `ypbind` service, which allows the system to act as a client in a NIS or NIS+ domain, should be disabled. The `ypbind` service can be disabled with the following command:

```
$ sudo systemctl disable ypbind.service
```

package_ypserv_removed

Uninstall ypserv Package

The `ypserv` package can be removed with the following command:

```
$ sudo yum erase ypserv
```

Xinetd

The `xinetd` service acts as a dedicated listener for some network services (mostly, obsolete ones) and can be used to provide access controls and perform some logging. It has been largely obsoleted by other features, and it is not installed by default. The older Inetd service is not even available as part of Red Hat Enterprise Linux 8.

package_tcp_wrappers_installed

Install tcp_wrappers Package

When network services are using the `xinetd` service, the `tcp_wrappers` package should be installed. The `tcp_wrappers` package can be installed with the following command:

```
$ sudo yum install tcp_wrappers
```

package_xinetd_removed

Uninstall xinetd Package

The `xinetd` package can be removed with the following command:

```
$ sudo yum erase xinetd
```

service_xinetd_disabled

Disable xinetd Service

The `xinetd` service can be disabled with the following command:

```
$ sudo systemctl disable xinetd.service
```

TFTP Server

TFTP is a lightweight version of the FTP protocol which has traditionally been used to configure networking equipment. However, TFTP provides little security, and modern versions of networking operating systems frequently support configuration via SSH or other more secure protocols. A TFTP server should be run only if no more secure method of supporting existing equipment can be found.

tftpd_uses_secure_mode

Ensure tftp Daemon Uses Secure Mode

If running the `tftp` service is necessary, it should be configured to change its root directory at startup. To do so, ensure `/etc/xinetd.d/tftp` includes `-s` as a command line argument, as shown in the following example (which is also the default):

```
server_args = -s /var/lib/tftpboot
```

service_tftp_disabled

Disable tftp Service

The `tftp` service should be disabled. The `tftp` service can be disabled with the following command:

```
$ sudo systemctl disable tftp.service
```

package_tftp_removed

Remove tftp Daemon

Trivial File Transfer Protocol (TFTP) is a simple file transfer protocol, typically used to automatically transfer configuration or boot files between systems. TFTP does not support authentication and can be easily hacked. The package `tftp` is a client program that allows for connections to a `tftp` server.

package_tftp-server_removed

Uninstall tftp-server Package

The `tftp-server` package can be removed with the following command:

```
$ sudo yum erase tftp-server
```

System Security Services Daemon

The System Security Services Daemon (SSSD) is a system daemon that provides access to different identity and authentication providers such as Red Hat's IdM, Microsoft's AD, openLDAP, MIT Kerberos, etc. It uses a common framework that can provide caching and offline support to systems utilizing SSSD. SSSD using caching to reduce load on authentication servers permit offline authentication as well as store extended user data.

For more information, see

System Security Services Daemon (SSSD) - LDAP

The System Security Services Daemon (SSSD) is a system daemon that provides access to different identity and authentication providers such as Red Hat's IdM, Microsoft's AD, openLDAP, MIT Kerberos, etc. It uses a common framework that can provide caching and offline support to systems utilizing SSSD. SSSD using caching to reduce load on authentication servers permit offline authentication as well as store extended user data.

SSSD can support many backends including LDAP. The `sssd-ldap` backend allows SSSD to fetch identity information from an LDAP server.

sssd_ldap_configure_tls_ca

Configure SSSD LDAP Backend Client CA Certificate

Configure SSSD to implement cryptography to protect the integrity of LDAP remote access sessions. By setting the

```
ldap_tls_cacert
```

option in

```
/etc/sssd/sssd.conf
```

to point to the path for the X.509 certificates used for peer authentication.

```
ldap_tls_cacert /path/to/tls/ca.cert
```

sssd_ldap_configure_tls_ca_dir

Configure SSSD LDAP Backend Client CA Certificate Location

Configure SSSD to implement cryptography to protect the integrity of LDAP remote access sessions. By setting the

```
ldap_tls_cacertdir
```

option in

```
/etc/sssd/sssd.conf
```

to point to the path for the X.509 certificates used for peer authentication.

```
ldap_tls_cacertdir /path/to/tls/cacert
```

sssd_ldap_start_tls

Configure SSSD LDAP Backend to Use TLS For All Transactions

This check verifies that Red Hat Enterprise Linux 8 implements cryptography to protect the integrity of remote LDAP authentication sessions.

To determine if LDAP is being used for authentication, use the following command:

```
$ sudo grep -i useldapauth /etc/sysconfig/authconfig
```

If `USELDAPAUTH=yes`, then LDAP is being used. To check if LDAP is configured to use TLS, use the following command:

```
$ sudo grep -i ldap_id_use_start_tls /etc/sssd/sssd.conf
```

sssd_memcache_timeout

Configure SSSD's Memory Cache to Expire

SSSD's memory cache should be configured to set to expire records after `seconds. To configure SSSD to expire memory cache, set `memcache_timeout` to` under the `[nss]` section in `/etc/sssd/sssd.conf`. For example:

```
[nss]
memcache_timeout =
```

package_sssd_installed

Install the SSSD Package

The `sssd` package should be installed. The `sssd` package can be installed with the following command:

```
$ sudo yum install sssd
```

sssd_enable_pam_services

Configure PAM in SSSD Services

SSSD should be configured to run SSSD `pam` services. To configure SSSD to known SSH hosts, add `pam` to `services` under the `[sssd]` section in `/etc/sssd/sssd.conf`. For example:

```
[sssd]
services = sudo, autofs, pam
```

service_sssd_enabled

Enable the SSSD Service

The SSSD service should be enabled. The `sssd` service can be enabled with the following command:

```
$ sudo systemctl enable sssd.service
```

sssd_ssh_known_hosts_timeout

Configure SSSD to Expire SSH Known Hosts

SSSD should be configured to expire keys from known SSH hosts after `seconds. To configure SSSD to known SSH hosts, set `ssh_known_hosts_timeout` to` under the `[ssh]` section in `/etc/sssd/sssd.conf`. For example:

```
[ssh]
ssh_known_hosts_timeout =
```

sssd_enable_smartcards

Enable Smartcards in SSSD

SSSD should be configured to authenticate access to the system using smart cards. To enable smart cards in SSSD, set `pam_cert_auth` to `true` under the `[pam]` section in `/etc/sssd/sssd.conf`. For example:

```
[pam]
pam_cert_auth = true
```

sssd_offline_cred_expiration

Configure SSSD to Expire Offline Credentials

SSSD should be configured to expire offline credentials after 1 day. To configure SSSD to expire offline credentials, set `offline_credentials_expiration` to `1` under the `[pam]` section in `/etc/sssd/sssd.conf`. For example:

```
[pam]
offline_credentials_expiration = 1
```

Web Server

The web server is responsible for providing access to content via the HTTP protocol. Web servers represent a significant security risk because:

- The HTTP port is commonly probed by malicious sources
- Web server software is very complex, and includes a long history of vulnerabilities
- The HTTP protocol is unencrypted and vulnerable to passive monitoring

The system's default web server software is Apache 2 and is provided in the RPM package `httpd`.

Disable Apache if Possible

If Apache was installed and activated, but the system does not need to act as a web server, then it should be disabled and removed from the system.

service_httpd_disabled

Disable httpd Service

The `httpd` service can be disabled with the following command:

```
$ sudo systemctl disable httpd.service
```

package_httpd_removed

Uninstall httpd Package

The `httpd` package can be removed with the following command:

```
$ sudo yum erase httpd
```

Secure Apache Configuration

The `httpd` configuration file is `/etc/httpd/conf/httpd.conf`. Apply the recommendations in the remainder of this section to this file.

Configure HTTPD-Served Web Content Securely

Running `httpd` inside a `chroot` jail is designed to isolate the web server process to a small section of the filesystem, limiting the damage if it is compromised. Versions of Apache greater than 2.2.10 (such as the one included with Red Hat Enterprise Linux 7) provide the `ChrootDir` directive. To run Apache inside a chroot jail in `/chroot/apache`, add the following line to `/etc/httpd/conf/httpd.conf`:

```
ChrootDir /chroot/apache
```

This necessitates placing all files required by `httpd` inside `/chroot/apache` , including `httpd`'s binaries, modules, configuration files, and served web pages. The details of this configuration are beyond the scope of this guide. This may also require additional SELinux configuration.

partition_for_web_content

Ensure Web Content Located on Separate partition

The `DocumentRoot` directory is used for storing web content and data. Ensure that the `DocumentRoot` directory exists on a separate logical volume at installation time, or migrate it using LVM.

httpd_encrypt_file_uploads

Encrypt All File Uploads

Use only secure encrypted logons and connections for uploading files to the web site.

httpd_configure_documentroot

Each Web Content Directory Must Contain An index.html File

Every `DocumentRoot` that is configured should have an `index.html` file that exists. Add an `index.html` file to every configured `DocumentRoot`.

httpd_limit_java_files

Remove .java And .jpp Files

`.java` and `.jpp` files should not exist and should be removed from the web server.

httpd_remove_robots_file

The robots.txt Files Must Not Exist

Remove any `robots.txt` files that may exist with any web content. Other methods must be employed if there is information on the web site that needs protection from search engines and public view. Inspect all instances of `DocumentRoot` and `Alias` and remove any `robots.txt` file.

```
$ sudo rm -f path/to/robots.txt
```

httpd_disable_content_symlinks

Disable Web Content Symbolic Links

For each `<Directory>` instance, remove the following:

```
FollowSymLinks
```

If symbolic links are allowed, the following can be added for each `<Directory>` instance:

```
Options SymLinksIfOwnerMatchDisable
```

httpd_configure_banner_page

Configure A Banner Page For Each Website

Configure a login banner for each website when authentication is required for user access.

Minimize Web Server Loadable Modules

A default installation of `httpd` includes a plethora of dynamically shared objects (DSO) that are loaded at run-time. Unlike the aforementioned compiled-in modules, a DSO can be disabled in the configuration file by removing the corresponding LoadModule directive.

Note: A DSO only provides additional functionality if associated directives are included in the `httpd` configuration file. It should also be noted that removing a DSO will produce errors on `httpd` startup if the configuration file contains directives that apply to that module. Refer to `http://httpd.apache.org/docs/` for details on which directives are associated with each DSO.

Following each DSO removal, the configuration can be tested with the following command to check if everything still works:

```
$ sudo service httpd configtest
```

The purpose of each of the modules loaded by default will now be addressed one at a time. If none of a module's directives are being used, remove it.

httpd Core Modules

These modules comprise a basic subset of modules that are likely needed for base `httpd` functionality; ensure they are not commented out in `/etc/httpd/conf/httpd.conf`:

```
LoadModule auth_basic_module modules/mod_auth_basic.so
LoadModule authn_default_module modules/mod_authn_default.so
LoadModule authz_host_module modules/mod_authz_host.so
LoadModule authz_user_module modules/mod_authz_user.so
LoadModule authz_groupfile_module modules/mod_authz_groupfile.so
LoadModule authz_default_module modules/mod_authz_default.so
LoadModule log_config_module modules/mod_log_config.so
LoadModule logio_module modules/mod_logio.so
LoadModule setenvif_module modules/mod_setenvif.so
LoadModule mime_module modules/mod_mome.so
LoadModule autoindex_module modules/mod_autoindex.so
LoadModule negotiation_module modules/mod_negotiation.so
LoadModule dir_module modules/mod_dir.so
LoadModule alias_module modules/mod_alias.so
```

Minimizing the number of loadable modules available to the web server reduces risk by limiting the capabilities allowed by the web server.

Minimize Modules for HTTP Basic Authentication

The following modules are necessary if this web server will provide content that will be restricted by a password.

Authentication can be performed using local plain text password files (`authn_file`), local DBM password files (`authn_dbm`) or an LDAP directory. The only module required by the web server depends on your choice of authentication. Comment out the modules you don't need from the following:

```
LoadModule authn_file_module modules/mod_authn_file.so
LoadModule authn_dbm_module modules/mod_authn_dbm.so
```

`authn_alias` allows for authentication based on aliases. `authn_anon` allows anonymous authentication similar to that of anonymous ftp sites. `authz_owner` allows authorization based on file ownership. `authz_dbm` allows for authorization based on group membership if the web server is using DBM authentication.

If the above functionality is unnecessary, comment out the related module:

```
#LoadModule authn_alias_module modules/mod_authn_alias.so
#LoadModule authn_anon_module modules/mod_authn_anon.so
#LoadModule authz_owner_module modules/mod_authz_owner.so
#LoadModule authz_dbm_module modules/mod_authz_dbm.so
```

Minimize Various Optional Components

The following modules perform very specific tasks, sometimes providing access to just a few additional directives. If such functionality is not required (or if you are not using these directives), comment out the associated module:

- External filtering (response passed through external program prior to client delivery)

```
#LoadModule ext_filter_module modules/mod_ext_filter.so
```

- User-specified Cache Control and Expiration

```
#LoadModule expires_module modules/mod_expires.so
```

- Compression Output Filter (provides content compression prior to client delivery)

```
#LoadModule deflate_module modules/mod_deflate.so
```

- HTTP Response/Request Header Customization

```
#LoadModule headers_module modules/mod_headers.so
```

- User activity monitoring via cookies

```
#LoadModule usertrack_module modules/mod_usertrack.so
```

- Dynamically configured mass virtual hosting

```
#LoadModule vhost_alias_module modules/mod_vhost_alias.so
```

Minimizing the number of loadable modules available to the web server reduces risk by limiting the capabilities allowed by the web server.

Minimize Configuration Files Included

The `Include` directive directs `httpd` to load supplementary configuration files from a provided path. The default configuration loads all files that end in `.conf` from the `/etc/httpd/conf.d` directory.

To restrict excess configuration, the following line should be commented out and replaced with `Include` directives that only reference required configuration files:

```
#Include conf.d/*.conf
```

If the above change was made, ensure that the SSL encryption remains loaded by explicitly including the corresponding configuration file:

```
Include conf.d/ssl.conf
```

If PHP is necessary, a similar alteration must be made:

```
Include conf.d/php.conf
```

Explicitly listing the configuration files to be loaded during web server start-up avoids the possibility of unwanted or malicious configuration files to be automatically included as part of the server's running configuration.

httpd_ldap_support

Disable LDAP Support

The `ldap` module provides HTTP authentication via an LDAP directory. If its functionality is unnecessary, comment out the related modules:

```
#LoadModule ldap_module modules/mod_ldap.so
#LoadModule authnz_ldap_module modules/mod_authnz_ldap.so
```

If LDAP is to be used, SSL encryption should be used as well.

httpd_url_correction

Disable URL Correction on Misspelled Entries

The `speling` module attempts to find a document match by allowing one misspelling in an otherwise failed request. If this functionality is unnecessary, comment out the module:

```
#LoadModule speling_module modules/mod_speling.so
```

This functionality weakens server security by making site enumeration easier.

httpd_cgi_support

Disable CGI Support

The `cgi` module allows HTML to interact with the CGI web programming language.

If this functionality is unnecessary, comment out the module:

```
#LoadModule cgi_module modules/mod_cgi.so
```

If the web server requires the use of CGI, enable `mod_cgi`.

httpd_mime_magic

Disable MIME Magic

The `mime_magic` module provides a second layer of MIME support that in most configurations is likely extraneous. If its functionality is unnecessary, comment out the related module:

```
#LoadModule mime_magic_module modules/mod_mime_magic.so
```

httpd_mod_rewrite

Disable HTTP mod_rewrite

The `mod_rewrite` module is very powerful and can protect against certain classes of web attacks. However, it is also very complex and has a significant history of vulnerabilities itself. If its functionality is unnecessary, comment out the related module:

```
#LoadModule rewrite_module modules/mod_rewrite.so
```

httpd_proxy_support

Disable Proxy Support

The `proxy` module provides proxying support, allowing `httpd` to forward requests and serve as a gateway for other servers. If its functionality is unnecessary, comment out the module:

```
#LoadModule proxy_module modules/mod_proxy.so
```

If proxy support is needed, load `mod_proxy` and the appropriate proxy protocol handler module (one of `mod_proxy_http`, `mod_proxy_ftp`, or `mod_proxy_connect`). Additionally, make certain that a server is secure before enabling proxying, as open proxy servers are a security risk. `mod_proxy_balancer` enables load balancing, but requires that `mod status` be enabled.

httpd_webdav

Disable WebDAV (Distributed Authoring and Versioning)

WebDAV is an extension of the HTTP protocol that provides distributed and collaborative access to web content. If its functionality is unnecessary, comment out the related modules:

```
#LoadModule dav_module modules/mod_dav.so
#LoadModule dav_fs_module modules/mod_dav_fs.so
```

If there is a critical need for WebDAV, extra care should be taken in its configuration. Since DAV access allows remote clients to manipulate server files, any location on the server that is DAV enabled should be protected by access controls.

httpd_enable_log_config

Enable log_config_module For HTTPD Logging

The `log_config_module` should exist and be configured in the `/etc/httpd/conf/httpd.conf` file by adding the following module to configure logging:

```
log_config_module
```

httpd_digest_authentication

Disable HTTP Digest Authentication

The `auth_digest` module provides encrypted authentication sessions. If this functionality is unnecessary, comment out the related module:

```
#LoadModule auth_digest_module modules/mod_auth_digest.so
```

httpd_server_configuration_display

Disable Web Server Configuration Display

The `info` module creates a web page illustrating the configuration of the web server. This can create an unnecessary security leak and should be disabled. If its functionality is unnecessary, comment out the module:

```
#LoadModule info_module modules/mod_info.so
```

If there is a critical need for this module, use the `Location` directive to provide an access control list to restrict access to the information.

httpd_server_side_includes

Disable Server Side Includes

Server Side Includes provide a method of dynamically generating web pages through the insertion of server-side code. However, the technology is also deprecated and introduces significant security concerns. If this functionality is unnecessary, comment out the related module:

```
#LoadModule include_module modules/mod_include.so
```

If there is a critical need for Server Side Includes, they should be enabled with the option `IncludesNoExec` to prevent arbitrary code execution. Additionally, user supplied data should be encoded to prevent cross-site scripting vulnerabilities.

httpd_cache_support

Disable Cache Support

The `cache` module allows `httpd` to cache data, optimizing access to frequently accessed content. However, it introduces potential security flaws such as the possibility of circumventing `Allow` and `Deny` directives.

If this functionality is unnecessary, comment out the module:

```
#LoadModule cache_module modules/mod_cache.so
```

If caching is required, it should not be enabled for any limited-access content.

httpd_server_activity_status

Disable Server Activity Status

The `status` module provides real-time access to statistics on the internal operation of the web server. This may constitute an unnecessary information leak and should be disabled unless necessary. To do so, comment out the related module:

```
#LoadModule status_module modules/mod_status.so
```

If there is a critical need for this module, ensure that access to the status page is properly restricted to a limited set of hosts in the status handler configuration.

Restrict Web Server Information Leakage

The `ServerTokens` and `ServerSignature` directives determine how much information the web server discloses about the configuration of the system.

httpd_servertokens_prod

Set httpd ServerTokens Directive to Prod

`ServerTokens Prod` restricts information in page headers, returning only the word "Apache."

Add or correct the following directive in `/etc/httpd/conf/httpd.conf`:

```
ServerTokens Prod
```

httpd_serversignature_off

Set httpd ServerSignature Directive to Off

`ServerSignature Off` restricts `httpd` from displaying server version number on error pages.

Add or correct the following directive in `/etc/httpd/conf/httpd.conf`:

```
ServerSignature Off
```

Configure Operating System to Protect Web Server

The following configuration steps should be taken on the system which hosts the web server, in order to provide as safe an environment as possible for the web server.

Run httpd in a chroot Jail if Practical

Running `httpd` inside a `chroot` jail is designed to isolate the web server process to a small section of the filesystem, limiting the damage if it is compromised. Versions of Apache greater than 2.2.10 (such as the one included with Red Hat Enterprise Linux 8) provide the `ChrootDir` directive. To run Apache inside a chroot jail in `/chroot/apache`, add the following line to `/etc/httpd/conf/httpd.conf`:

```
ChrootDir /chroot/apache
```

This necessitates placing all files required by `httpd` inside `/chroot/apache` , including `httpd`'s binaries, modules, configuration files, and served web pages. The details of this configuration are beyond the scope of this guide. This may also require additional SELinux configuration.

Restrict File and Directory Access

Minimize access to critical `httpd` files and directories.

file_permissions_httpd_server_conf_files

Set Permissions on All Configuration Files Inside /etc/httpd/conf/

To properly set the permissions of `/etc/http/conf/*`, run the command:

```
$ sudo chmod 0640 /etc/http/conf/*
```

dir_perms_etc_httpd_conf

Set Permissions on the /etc/httpd/conf/ Directory

To properly set the permissions of `/etc/http/conf`, run the command:

```
$ sudo chmod 0750 /etc/http/conf
```

dir_perms_var_log_httpd

Set Permissions on the /var/log/httpd/ Directory

Ensure that the permissions on the web server log directory is set to 700:

```
$ sudo chmod 700 /var/log/httpd/
```

This is its default setting.

file_permissions_httpd_server_modules_files

Set Permissions on All Configuration Files Inside /etc/httpd/conf.modules.d/

To properly set the permissions of `/etc/http/conf.modules.d/*`, run the command:

```
$ sudo chmod 0640 /etc/http/conf.modules.d/*
```

file_permissions_httpd_server_conf_d_files

Set Permissions on All Configuration Files Inside /etc/httpd/conf.d/

To properly set the permissions of `/etc/http/conf.d/*`, run the command:

```
$ sudo chmod 0640 /etc/http/conf.d/*
```

http_configure_log_file_ownership

HTTPD Log Files Must Be Owned By Root

All `httpd` logs must be owned by root user and group. By default, the path for httpd logs is `/var/log/httpd/` To properly set the owner of `/var/log/httpd`, run the command:

```
$ sudo chown root /var/log/httpd
```

To properly set the owner of `/var/log/httpd/*`, run the command:

```
$ sudo chown root /var/log/httpd/*
```

httpd_configure_remote_session_encryption

Ensure Remote Administrative Access Is Encrypted

Ensure that the SSH server service is enabled. The `sshd` service can be enabled with the following command:

```
$ sudo systemctl enable sshd.service
```

httpd_antivirus_scan_uploads

Scan All Uploaded Content for Malicious Software

Install anti-virus software on the system and set it to automatically scan new files that are introduced to the web server.

httpd_configure_firewall

Configure firewall to Allow Access to the Web Server

By default, `iptables` blocks access to the ports used by the web server. To configure `iptables` to allow port 80 traffic, one must edit `/etc/sysconfig/iptables` and `/etc/sysconfig/ip6tables` (if IPv6 is in use). Add the following line, ensuring that it appears before the final LOG and DROP lines for the INPUT chain:

```
-A INPUT -m state --state NEW -p tcp --dport 80 -j ACCEPT
```

To configure `iptables` to allow port 443 traffic, one must edit `/etc/sysconfig/iptables` and `/etc/sysconfig/ip6tables` (if IPv6 is in use). Add the following line, ensuring that it appears before the final LOG and DROP lines for the INPUT chain:

```
-A INPUT -m state --state NEW -p tcp --dport 443 -j ACCEPT
```

Configure PHP Securely

PHP is a widely-used and often misconfigured server-side scripting language. It should be used with caution, but configured appropriately when needed.

Review `/etc/php.ini` and make the following changes if possible:

```
# Do not expose PHP error messages to external users
display_errors = Off

# Enable safe mode
safe_mode = On

# Only allow access to executables in isolated directory
safe_mode_exec_dir = php-required-executables-path

# Limit external access to PHP environment
safe_mode_allowed_env_vars = PHP_

# Restrict PHP information leakage
expose_php = Off

# Log all errors
log_errors = On

# Do not register globals for input data
register_globals = Off

# Minimize allowable PHP post size
post_max_size = 1K

# Ensure PHP redirects appropriately
cgi.force_redirect = 0

# Disallow uploading unless necessary
file_uploads = Off

# Disallow treatment of file requests as fopen calls
allow_url_fopen = Off

# Enable SQL safe mode
sql.safe_mode = On
```

Use Denial-of-Service Protection Modules

Denial-of-service attacks are difficult to detect and prevent while maintaining acceptable access to authorized users. However, some traffic-shaping modules can be used to address the problem. Well-known DoS protection modules include:

```
mod_cband mod_bwshare mod_limitipconn mod_evasive
```

Denial-of-service prevention should be implemented for a web server if such a threat exists. However, specific configuration details are very dependent on the environment and often best left at the discretion of the administrator.

Configure PERL Securely

PERL (Practical Extraction and Report Language) is an interpreted language optimized for scanning arbitrary text files, extracting information from those text files, and printing reports based on that information. The language is often used in shell scripting and is intended to be practical, easy to use, and efficient means of generating interactive web pages for the user.

httpd_configure_perl_taint

Configure HTTP PERL Scripts To Use TAINT Option

If the `mod_perl` module is installed, enable Perl Taint checking in `/etc/httpd/conf/httpd.conf`. To enable Perl Taint checking, add or uncomment the following to `/etc/httpd/conf.d/perl.conf`:

```
PerlSwitches -T
```

Directory Restrictions

The Directory tags in the web server configuration file allow finer grained access control for a specified directory. All web directories should be configured on a case-by-case basis, allowing access only where needed.

httpd_disable_anonymous_ftp_access

Disable Anonymous FTP Access

If any directories that contain dynamic scripts can be accessed via FTP by any group or user that does not require access, remove permissions to such directories that allow anonymous access. Also, ensure that any such access employs an encrypted connection.

httpd_limit_available_methods

Limit Available Methods

Web server methods are defined in section 9 of RFC 2616 (http://www.ietf.org/rfc/rfc2616.txt). If a web server does not require the implementation of all available methods, they should be disabled.

Note: `GET` and `POST` are the most common methods. A majority of the others are limited to the WebDAV protocol.

```
<Directory /var/www/html>
# ...
   # Only allow specific methods (this command is case-sensitive!)
   <LimitExcept GET POST>
      Order allow,deny
   </LimitExcept>
# ...
</Directory>
```

httpd_ignore_htaccess_files

Ignore HTTPD .htaccess Files

Set `AllowOverride` to `none` for each instant of `<Directory>`.

httpd_restrict_root_directory

Restrict Root Directory

The `httpd` root directory should always have the most restrictive configuration enabled.

```
<Directory / >
   Options None
   AllowOverride None
   Order allow,deny
</Directory>
```

httpd_restrict_critical_directories

Restrict Other Critical Directories

All accessible web directories should be configured with similarly restrictive settings. The `Options` directive should be limited to necessary functionality and the `AllowOverride` directive should be used only if needed. The `Order` and `Deny` access control tags should be used to deny access by default, allowing access only where necessary.

httpd_anonymous_content_sharing

Web Content Directories Must Not Be Shared Anonymously

Web content directories should not be shared anonymously over remote filesystems such as `nfs` and `smb`. Remove the shares from the applicable directories.

httpd_restrict_web_directory

Restrict Web Directory

The default configuration for the web (`/var/www/html`) Directory allows directory indexing (`Indexes`) and the following of symbolic links (`FollowSymLinks`). Neither of these is recommended.

The `/var/www/html` directory hierarchy should not be viewable via the web, and symlinks should only be followed if the owner of the symlink also owns the linked file.

Ensure that this policy is adhered to by altering the related section of the configuration:

```
<Directory "/var/www/html">
#  ...
   Options SymLinksIfOwnerMatch
#  ...
</Directory>
```

httpd_configure_script_permissions

Remove Write Permissions From Filesystem Paths And Server Scripts

Configure permissions for each instance of `Alias`, `ScriptAlias`, and `ScriptAliasMatch` that exist.

```
$ sudo find DIR -type d -exec chmod 755 {} \;
$ sudo find DIR -type f -exec chmod 555 {} \;
```

Where *DIR* matches the paths from `Alias`, `ScriptAlias`, and `ScriptAliasMatch`.

Use Appropriate Modules to Improve httpd's Security

Among the modules available for `httpd` are several whose use may improve the security of the web server installation. This section recommends and discusses the deployment of security-relevant modules.

Deploy mod_security

The `security` module provides an application level firewall for `httpd`. Following its installation with the base ruleset, specific configuration advice can be found at http://www.modsecurity.org/ to design a policy that best matches the security needs of the web applications. Usage of `mod_security` is highly recommended for some environments, but it should be noted this module does not ship with Red Hat Enterprise Linux itself, and instead is provided via Extra Packages for Enterprise Linux (EPEL). For more information on EPEL please refer to http://fedoraproject.org/wiki/EPEL.

httpd_install_mod_security

Install mod_security

Install the `security` module: The `mod_security` package can be installed with the following command:

```
$ sudo yum install mod_security
```

Deploy mod_ssl

Because HTTP is a plain text protocol, all traffic is susceptible to passive monitoring. If there is a need for confidentiality, SSL should be configured and enabled to encrypt content.

Note: `mod_nss` is a FIPS 140-2 certified alternative to `mod_ssl`. The modules share a considerable amount of code and should be nearly identical in functionality. If FIPS 140-2 validation is required, then `mod_nss` should be used. If it provides some feature or its greater compatibility is required, then `mod_ssl` should be used.

httpd_configure_valid_server_cert

Configure A Valid Server Certificate

Configure the web site to use a valid organizationally defined certificate. For DoD, this is a DoD server certificate issued by the DoD CA.

httpd_install_mod_ssl

Install mod_ssl

Install the `mod_ssl` module: The `mod_ssl` package can be installed with the following command:

```
$ sudo yum install mod_ssl
```

httpd_require_client_certs

Require Client Certificates

`SSLVerifyClient` should be set and configured to `require` by setting the following in `/etc/httpd/conf/httpd.conf`:

```
SSLVerifyClient require
```

httpd_configure_tls

Enable Transport Layer Security (TLS) Encryption

Disable old SSL and TLS version and enable the latest TLS encryption by setting the following in `/etc/httpd/conf.modules.d/ssl.conf`:

```
SSLProtocol all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1
```

Make sure to also set `SSLEngine` to `on` in `/etc/httpd/conf.modules.d/ssl.conf` like the following:

```
SSLEngine on
```

httpd_nipr_accredited_dmz

A public web server, if hosted on the NIPRNet, must be isolated in an accredited DoD DMZ extension

To minimize exposure of private assets to unnecesarry risk by attackers, public web servers must be isolated from internal systems. Logically relocate public web servers to be isolated from internal systems. In addition, ensure the public web server does not have trusted connections with assets outside the confines of the demilitarizez done (DMZ) other than application and/or database servers that are a part of the same system as the web server.

httpd_private_server_on_separate_subnet

A private web server must be located on a separate controlled access subnet

Private web servers, which host sites that serve controlled access data, must be protected from outside threats in addition to insider threats. Isolate the private web server from the public DMZ and separate it from the internal general population LAN.

httpd_configure_max_keepalive_requests

Configure The Number of Allowed Simultaneous Requests

The `MaxKeepAliveRequests` directive should be set and configured to or greater by setting the following in `/etc/httpd/conf/httpd.conf`:

```
MaxKeepAliveRequests
```

httpd_public_resources_not_shared

Public web server resources must not be shared with private assets

It is important to segregate public web server resources from private resources located behind the DoD DMZ in order to protect private assets.

httpd_enable_loglevel

Enable HTTPD LogLevel

`LogLevel` should be enabled and set to . Add or edit the following in `/etc/httpd/conf/httpd.conf`:

```
LogLevel
```

httpd_entrust_passwords

The web server password(s) must be entrusted to the SA or Web Manager

Normally, a service account is established for the web server. This is because a privileged account is not desirable and the server is designed to run for long uninterrupted periods of time. The SA or Web Manager will need password access to the web server to restart the service in the event or an emergency as the web server is not to restart automatically after an unscheduled interruption.

httpd_enable_error_logging

Enable HTTPD Error Logging

`ErrorLog` should be enabled and set to the following in `/etc/httpd/conf/httpd.conf`:

```
ErrorLog "logs/error_log"
```

httpd_no_compilers_in_prod

Installation of a compiler on production web server is prohibited

The presence of a compiler on a production server facilitates the malicious user's task of creating custom versions of programs and installing Trojan Horses or viruses.

httpd_configure_log_format

Configure Error Log Format

`LogFormat` should be enabled and set to the following in `/etc/httpd/conf/httpd.conf`:

```
LogFormat "a %A %h %H %l %m %s %t %u %U \"%{Referer}i\" \"%{User-Agent}i\"" combined
```

httpd_remove_backups

Backup interactive scripts on the production web server are prohibited

Copies of backup files will not execute on the server, but they can be read by the anonymous user if special precautions are not taken.

httpd_enable_system_logging

Enable HTTPD System Logging

`CustomLog` should be enabled and set to the following in `/etc/httpd/conf/httpd.conf`:

```
CustomLog "logs/access_log" combined
```

httpd_disable_mime_types

MIME types for csh or sh shell programs must be disabled

Users must not be allowed to access the shell programs.

Install Apache if Necessary

If `httpd` was not installed and activated, but the system needs to act as a web server, then it should be installed on the system. Follow these guidelines to install it defensively. The `httpd` package can be installed with the following command:

```
$ sudo yum install httpd
```

This method of installation is recommended over installing the "Web Server" package group during the system installation process. The Web Server package group includes many packages which are likely extraneous, while the command-line method installs only the required `httpd` package itself.

Confirm Minimal Built-in Modules Installed

The default `httpd` installation minimizes the number of modules that are compiled directly into the binary (`core prefork http_core mod_so`). This minimizes risk by limiting the capabilities allowed by the web server. Query the set of compiled-in modules using the following command:

```
$ httpd -l
```

If the number of compiled-in modules is significantly larger than the aforementioned set, this guide recommends re-installing `httpd` with a reduced configuration. Minimizing the number of modules that are compiled into the `httpd` binary, reduces risk by limiting the capabilities allowed by the webserver.

X Window System

The X Window System implementation included with the system is called X.org.

Disable X Windows

Unless there is a mission-critical reason for the system to run a graphical user interface, ensure X is not set to start automatically at boot and remove the X Windows software packages. There is usually no reason to run X Windows on a dedicated server system, as it increases the system's attack surface and consumes system resources. Administrators of server systems should instead login via SSH or on the text console.

package_xorg-x11-server-common_removed

Remove the X Windows Package Group

By removing the xorg-x11-server-common package, the system no longer has X Windows installed. If X Windows is not installed then the system cannot boot into graphical user mode. This prevents the system from being accidentally or maliciously booted into a `graphical.target` mode. To do so, run the following command:

```
$ sudo yum groupremove "X Window System"
```

```
$ sudo yum remove xorg-x11-server-common
```

xwindows_runlevel_target

Disable X Windows Startup By Setting Default Target

Systems that do not require a graphical user interface should only boot by default into `multi-user.target` mode. This prevents accidental booting of the system into a `graphical.target` mode. Setting the system's default target to `multi-user.target` will prevent automatic startup of the X server. To do so, run:

```
$ systemctl set-default multi-user.target
```

You should see the following output:

```
rm '/etc/systemd/system/default.target'
ln -s '/usr/lib/systemd/system/multi-user.target' '/etc/systemd/system/default.target'
```

Docker Service

The docker service is necessary to create containers, which are self-sufficient and self-contained applications using the resource isolation features of the kernel.

docker_selinux_enabled

Ensure SELinux support is enabled in Docker

To enable the SELinux for the Docker service, the Docker service must be configured to run the Docker daemon with `--selinux-enabled` option. In `/etc/sysconfig/docker` configuration file, add or correct the following line to enable SELinux support in the Docker daemon:

```
OPTIONS='--selinux-enabled'
```

SSH Server

The SSH protocol is recommended for remote login and remote file transfer. SSH provides confidentiality and integrity for data exchanged between two systems, as well as server authentication, through the use of public key cryptography. The implementation included with the system is called OpenSSH, and more detailed documentation is available from its website, http://www.openssh.org. Its server program is called `sshd` and provided by the RPM package `openssh-server`.

Configure OpenSSH Server if Necessary

If the system needs to act as an SSH server, then certain changes should be made to the OpenSSH daemon configuration file `/etc/ssh/sshd_config`. The following recommendations can be applied to this file. See the `sshd_config(5)` man page for more detailed information.

Strengthen Firewall Configuration if Possible

If the SSH server is expected to only receive connections from the local network, then strengthen the default firewall rule for the SSH service to only accept connections from the appropriate network segment(s).

Determine an appropriate network block, `netwk`, network mask, `mask`, and network protocol, `ip_protocol`, representing the systems on your network which will be allowed to access this SSH server.

Run the following command:

```
firewall-cmd --permanent --add-rich-rule='rule family="ip_protocol" source address="netwk/mask" service name="ssh" accept'
```

sshd_disable_user_known_hosts

Disable SSH Support for User Known Hosts

SSH can allow system users user host-based authentication to connect to systems if a cache of the remote systems public keys are available. This should be disabled.

To ensure this behavior is disabled, add or correct the following line in `/etc/ssh/sshd_config`:

```
IgnoreUserKnownHosts yes
```

sshd_set_loglevel_info

Set LogLevel to INFO

The INFO parameter specifices that record login and logout activity will be logged. To specify the log level in SSH, add or correct the following line in the `/etc/ssh/sshd_config` file:

```
LogLevel INFO
```

sshd_enable_warning_banner

Enable SSH Warning Banner

To enable the warning banner and ensure it is consistent across the system, add or correct the following line in `/etc/ssh/sshd_config`:

```
Banner /etc/issue
```

Another section contains information on how to create an appropriate system-wide warning banner.

firewalld_sshd_port_enabled

Enable SSH Server firewalld Firewall Exception

By default, inbound connections to SSH's port are allowed. If the SSH server is being used but denied by the firewall, this exception should be added to the firewall configuration.

To configure `firewalld` to allow access, run the following command(s): `firewall-cmd --permanent --add-service=ssh`

sshd_use_priv_separation

Enable Use of Privilege Separation

When enabled, SSH will create an unprivileged child process that has the privilege of the authenticated user. To enable privilege separation in SSH, add or correct the following line in the `/etc/ssh/sshd_config` file:

```
UsePrivilegeSeparation sandbox
```

sshd_do_not_permit_user_env

Do Not Allow SSH Environment Options

To ensure users are not able to override environment options to the SSH daemon, add or correct the following line in `/etc/ssh/sshd_config`:

```
PermitUserEnvironment no
```

sshd_enable_x11_forwarding

Enable Encrypted X11 Forwarding

By default, remote X11 connections are not encrypted when initiated by users. SSH has the capability to encrypt remote X11 connections when SSH's `X11Forwarding` option is enabled.

To enable X11 Forwarding, add or correct the following line in `/etc/ssh/sshd_config`:

```
X11Forwarding yes
```

sshd_limit_user_access

Limit Users' SSH Access

By default, the SSH configuration allows any user with an account to access the system. In order to specify the users that are allowed to login via SSH and deny all other users, add or correct the following line in the `/etc/ssh/sshd_config` file:

```
DenyUsers USER1 USER2
```

Where `USER1` and `USER2` are valid user names.

sshd_disable_kerb_auth

Disable Kerberos Authentication

Unless needed, SSH should not permit extraneous or unnecessary authentication mechanisms like Kerberos. To disable Kerberos authentication, add or correct the following line in the `/etc/ssh/sshd_config` file:

```
KerberosAuthentication no
```

sshd_disable_root_login

Disable SSH Root Login

The root user should never be allowed to login to a system directly over a network. To disable root login via SSH, add or correct the following line in `/etc/ssh/sshd_config`:

```
PermitRootLogin no
```

sshd_disable_gssapi_auth

Disable GSSAPI Authentication

Unless needed, SSH should not permit extraneous or unnecessary authentication mechanisms like GSSAPI. To disable GSSAPI authentication, add or correct the following line in the `/etc/ssh/sshd_config` file:

```
GSSAPIAuthentication no
```

sshd_disable_compression

Disable Compression Or Set Compression to delayed

Compression is useful for slow network connections over long distances but can cause performance issues on local LANs. If use of compression is required, it should be enabled only after a user has authenticated; otherwise , it should be disabled. To disable compression or delay compression until after a user has successfully authenticated, add or correct the following line in the `/etc/ssh/sshd_config` file:

```
Compression no
```

or

```
Compression delayed
```

sshd_print_last_log

Enable SSH Print Last Log

When enabled, SSH will display the date and time of the last successful account logon. To enable LastLog in SSH, add or correct the following line in the `/etc/ssh/sshd_config` file:

```
PrintLastLog yes
```

sshd_disable_rhosts

Disable SSH Support for .rhosts Files

SSH can emulate the behavior of the obsolete rsh command in allowing users to enable insecure access to their accounts via `.rhosts` files.

To ensure this behavior is disabled, add or correct the following line in `/etc/ssh/sshd_config`:

```
IgnoreRhosts yes
```

sshd_disable_empty_passwords

Disable SSH Access via Empty Passwords

To explicitly disallow SSH login from accounts with empty passwords, add or correct the following line in `/etc/ssh/sshd_config`:

```
PermitEmptyPasswords no
```

Any accounts with empty passwords should be disabled immediately, and PAM configuration should prevent users from being able to assign themselves empty passwords.

sshd_set_idle_timeout

Set SSH Idle Timeout Interval

SSH allows administrators to set an idle timeout interval. After this interval has passed, the idle user will be automatically logged out.

To set an idle timeout interval, edit the following line in `/etc/ssh/sshd_config` as follows:

```
ClientAliveInterval
```

The timeout **interval** is given in seconds. To have a timeout of 15 minutes, set **interval** to 900.

If a shorter timeout has already been set for the login shell, that value will preempt any SSH setting made here. Keep in mind that some processes may stop SSH from correctly detecting that the user is idle.

sshd_allow_only_protocol2

Allow Only SSH Protocol 2

Only SSH protocol version 2 connections should be permitted. The default setting in `/etc/ssh/sshd_config` is correct, and can be verified by ensuring that the following line appears:

```
Protocol 2
```

sshd_set_keepalive

Set SSH Client Alive Max Count

To ensure the SSH idle timeout occurs precisely when the `ClientAliveInterval` is set, edit `/etc/ssh/sshd_config` as follows:

```
ClientAliveCountMax
```

sshd_set_max_auth_tries

Set SSH authentication attempt limit

The `MaxAuthTries` parameter specifies the maximum number of authentication attempts permitted per connection. Once the number of failures reaches half this value, additional failures are logged. to set MaxAUthTries edit `/etc/ssh/sshd_config` as follows:

```
MaxAuthTries tries
```

sshd_disable_rhosts_rsa

Disable SSH Support for Rhosts RSA Authentication

SSH can allow authentication through the obsolete rsh command through the use of the authenticating user's SSH keys. This should be disabled.

To ensure this behavior is disabled, add or correct the following line in `/etc/ssh/sshd_config`:

```
RhostsRSAAuthentication no
```

disable_host_auth

Disable Host-Based Authentication

SSH's cryptographic host-based authentication is more secure than `.rhosts` authentication. However, it is not recommended that hosts unilaterally trust one another, even within an organization.

To disable host-based authentication, add or correct the following line in `/etc/ssh/sshd_config`:

```
HostbasedAuthentication no
```

sshd_enable_strictmodes

Enable Use of Strict Mode Checking

SSHs StrictModes option checks file and ownership permissions in the user's home directory `.ssh` folder before accepting login. If world- writable permissions are found, logon is rejected. To enable StrictModes in SSH, add or correct the following line in the `/etc/ssh/sshd_config` file:

```
StrictModes yes
```

service_sshd_disabled

Disable SSH Server If Possible (Unusual)

The SSH server service, sshd, is commonly needed. However, if it can be disabled, do so. The `sshd` service can be disabled with the following command:

```
$ sudo systemctl disable sshd.service
```

This is unusual, as SSH is a common method for encrypted and authenticated remote access.

file_permissions_sshd_pub_key

Verify Permissions on SSH Server Public *.pub Key Files

To properly set the permissions of `/etc/ssh/*.pub`, run the command:

```
$ sudo chmod 0644 /etc/ssh/*.pub
```

service_sshd_enabled

Enable the OpenSSH Service

The SSH server service, sshd, is commonly needed. The `sshd` service can be enabled with the following command:

```
$ sudo systemctl enable sshd.service
```

firewalld_sshd_disabled

Remove SSH Server firewalld Firewall exception (Unusual)

By default, inbound connections to SSH's port are allowed. If the SSH server is not being used, this exception should be removed from the firewall configuration.

To configure `firewalld` to prevent access, run the following command(s): `firewall-cmd --permanent --remove-service=ssh`

iptables_sshd_disabled

Remove SSH Server iptables Firewall exception (Unusual)

By default, inbound connections to SSH's port are allowed. If the SSH server is not being used, this exception should be removed from the firewall configuration.

Edit the files `/etc/sysconfig/iptables` and `/etc/sysconfig/ip6tables` (if IPv6 is in use). In each file, locate and delete the line:

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
```

This is unusual, as SSH is a common method for encrypted and authenticated remote access.

file_permissions_sshd_private_key

Verify Permissions on SSH Server Private *_key Key Files

To properly set the permissions of `/etc/ssh/*_key`, run the command:

```
$ sudo chmod 0640 /etc/ssh/*_key
```

package_openssh-server_installed

Install the OpenSSH Server Package

The `openssh-server` package should be installed. The `openssh-server` package can be installed with the following command:

```
$ sudo yum install openssh-server
```

SNMP Server

The Simple Network Management Protocol allows administrators to monitor the state of network devices, including computers. Older versions of SNMP were well-known for weak security, such as plaintext transmission of the community string (used for authentication) and usage of easily-guessable choices for the community string.

Configure SNMP Server if Necessary

If it is necessary to run the snmpd agent on the system, some best practices should be followed to minimize the security risk from the installation. The multiple security models implemented by SNMP cannot be fully covered here so only the following general configuration advice can be offered:

- use only SNMP version 3 security models and enable the use of authentication and encryption
- write access to the MIB (Management Information Base) should be allowed only if necessary
- all access to the MIB should be restricted following a principle of least privilege
- network access should be limited to the maximum extent possible including restricting to expected network addresses both in the configuration files and in the system firewall rules
- ensure SNMP agents send traps only to, and accept SNMP queries only from, authorized management stations
- ensure that permissions on the `snmpd.conf` configuration file (by default, in `/etc/snmp`) are 640 or more restrictive
- ensure that any MIB files' permissions are also 640 or more restrictive

snmpd_not_default_password

Ensure Default SNMP Password Is Not Used

Edit `/etc/snmp/snmpd.conf`, remove or change the default community strings of `public` and `private`. Once the default community strings have been changed, restart the SNMP service:

```
$ sudo service snmpd restart
```

snmpd_use_newer_protocol

Configure SNMP Service to Use Only SNMPv3 or Newer

Edit `/etc/snmp/snmpd.conf`, removing any references to `rocommunity`, `rwcommunity`, or `com2sec`. Upon doing that, restart the SNMP service:

```
$ sudo service snmpd restart
```

Disable SNMP Server if Possible

The system includes an SNMP daemon that allows for its remote monitoring, though it not installed by default. If it was installed and activated but is not needed, the software should be disabled and removed.

package_net-snmp_removed

Uninstall net-snmp Package

The `net-snmp` package provides the snmpd service. The `net-snmp` package can be removed with the following command:

```
$ sudo yum erase net-snmp
```

service_snmpd_disabled

Disable snmpd Service

The `snmpd` service can be disabled with the following command:

```
$ sudo systemctl disable snmpd.service
```

Mail Server Software

Mail servers are used to send and receive email over the network. Mail is a very common service, and Mail Transfer Agents (MTAs) are obvious targets of network attack. Ensure that systems are not running MTAs unnecessarily, and configure needed MTAs as defensively as possible.

Very few systems at any site should be configured to directly receive email over the network. Users should instead use mail client programs to retrieve email from a central server that supports protocols such as IMAP or POP3. However, it is normal for most systems to be independently capable of sending email, for instance so that cron jobs can report output to an administrator. Most MTAs, including Postfix, support a submission-only mode in which mail can be sent from the local system to a central site MTA (or directly delivered to a local account), but the system still cannot receive mail directly over a network.

The `alternatives` program in Red Hat Enterprise Linux 8 permits selection of other mail server software (such as Sendmail), but Postfix is the default and is preferred. Postfix was coded with security in mind and can also be more effectively contained by SELinux as its modular design has resulted in separate processes performing specific actions. More information is available on its website, http://www.postfix.org.

Configure SMTP For Mail Clients

This section discusses settings for Postfix in a submission-only e-mail configuration.

postfix_client_configure_mail_alias

Configure System to Forward All Mail For The Root Account

Set up an alias for root that forwards to a monitored email address:

```
$ sudo echo "root: " >> /etc/aliases
$ sudo newaliases
```

postfix_network_listening_disabled

Disable Postfix Network Listening

Edit the file `/etc/postfix/main.cf` to ensure that only the following `inet_interfaces` line appears:

```
inet_interfaces = localhost
```

Configure Operating System to Protect Mail Server

The guidance in this section is appropriate for any host which is operating as a site MTA, whether the mail server runs using Sendmail, Postfix, or some other software.

Configure SSL Certificates for Use with SMTP AUTH

If SMTP AUTH is to be used, the use of SSL to protect credentials in transit is strongly recommended. There are also configurations for which it may be desirable to encrypt all mail in transit from one MTA to another, though such configurations are beyond the scope of this guide. In either event, the steps for creating and installing an SSL certificate are independent of the MTA in use, and are described here.

Ensure Security of Postfix SSL Certificate

Create the PKI directory for mail certificates, if it does not already exist:

```
$ sudo mkdir /etc/pki/tls/mail
$ sudo chown root:root /etc/pki/tls/mail
$ sudo chmod 755 /etc/pki/tls/mail
```

Using removable media or some other secure transmission format, install the files generated in the previous step onto the mail server:

```
/etc/pki/tls/mail/serverkey.pem: the private key mailserverkey.pem
/etc/pki/tls/mail/servercert.pem: the certificate file mailservercert.pem
```

Verify the ownership and permissions of these files:

```
$ sudo chown root:root /etc/pki/tls/mail/serverkey.pem
$ sudo chown root:root /etc/pki/tls/mail/servercert.pem
$ sudo chmod 600 /etc/pki/tls/mail/serverkey.pem
$ sudo chmod 644 /etc/pki/tls/mail/servercert.pem
```

Verify that the CA's public certificate file has been installed as `/etc/pki/tls/CA/cacert.pem`, and has the correct permissions:

```
$ sudo chown root:root /etc/pki/tls/CA/cacert.pem
$ sudo chmod 644 /etc/pki/tls/CA/cacert.pem
```

Configure Postfix if Necessary

Postfix stores its configuration files in the directory /etc/postfix by default. The primary configuration file is `/etc/postfix/main.cf`.

Control Mail Relaying

Postfix's mail relay controls are implemented with the help of the smtpd recipient restrictions option, which controls the restrictions placed on the SMTP dialogue once the sender and recipient envelope addresses are known. The guidance in the following sections should be applied to all systems. If there are systems which must be allowed to relay mail, but which cannot be trusted to relay unconditionally, configure SMTP AUTH with SSL support.

Require SMTP AUTH Before Relaying from Untrusted Clients

SMTP authentication allows remote clients to relay mail safely by requiring them to authenticate before submitting mail. Postfix's SMTP AUTH uses an authentication library called SASL, which is not part of Postfix itself. To enable the use of SASL authentication, see http://www.postfix.org/SASL_README.html

Enact SMTP Recipient Restrictions

To configure Postfix to restrict addresses to which it will send mail, see: http://www.postfix.org/SMTPD_ACCESS_README.html#danger The full contents of `smtpd_recipient_restrictions` will vary by site, since this is a common place to put spam restrictions and other site-specific options. The `permit_mynetworks` option allows all mail to be relayed from the systems in `mynetworks`. Then, the `reject_unauth_destination` option denies all mail whose destination address is not local, preventing any other systems from relaying. These two options should always appear in this order, and should usually follow one another immediately unless SMTP AUTH is used.

Use TLS for SMTP AUTH

Postfix provides options to use TLS for certificate-based authentication and encrypted sessions. An encrypted session protects the information that is transmitted with SMTP mail or with SASL authentication. To configure Postfix to protect all SMTP AUTH transactions using TLS, see http://www.postfix.org/TLS_README.html.

Enact SMTP Relay Restrictions

To configure Postfix to restrict addresses to which it will send mail, see: http://www.postfix.org/SMTPD_ACCESS_README.html#danger The full contents of `smtpd_recipient_restrictions` will vary by site, since this is a common place to put spam restrictions and other site-specific options. The `permit_mynetworks` option allows all mail to be relayed from the systems in `mynetworks`. Then, the `reject_unauth_destination` option denies all mail whose destination address is not local, preventing any other systems from relaying. These two options should always appear in this order, and should usually follow one another immediately unless SMTP AUTH is used.

Configure Trusted Networks and Hosts

Edit `/etc/postfix/main.cf`, and configure the contents of the `mynetworks` variable in one of the following ways:

- If any system in the subnet containing the MTA may be trusted to relay messages, add or correct the following line:

```
mynetworks_style = subnet
```

  This is also the default setting, and is in effect if all `my_networks_style` directives are commented.

- If only the MTA host itself is trusted to relay messages, add or correct the following line:

```
mynetworks_style = host
```

- If the set of systems which can relay is more complicated, manually specify an entry for each netblock or IP address which is trusted to relay by setting the `mynetworks` variable directly:

```
mynetworks = 10.0.0.0/16, 192.168.1.0/24, 127.0.0.1
```

postfix_prevent_unrestricted_relay

Prevent Unrestricted Mail Relaying

Modify the

```
/etc/postfix/main.cf
```

file to restrict client connections to the local network with the following command:

```
$ sudo postconf -e 'smtpd_client_restrictions = permit_mynetworks,reject'
```

Configure Postfix Resource Usage to Limit Denial of Service Attacks

Edit `/etc/postfix/main.cf`. Edit the following lines to configure the amount of system resources Postfix can consume:

```
default_process_limit = 100
smtpd_client_connection_count_limit = 10
smtpd_client_connection_rate_limit = 30
queue_minfree = 20971520
header_size_limit = 51200
message_size_limit = 10485760
smtpd_recipient_limit = 100
```

The values here are examples.

postfix_server_banner

Configure SMTP Greeting Banner

Edit `/etc/postfix/main.cf`, and add or correct the following line, substituting some other wording for the banner information if you prefer:

```
smtpd_banner = $myhostname ESMTP
```

package_sendmail_removed

Uninstall Sendmail Package

Sendmail is not the default mail transfer agent and is not installed by default. The `sendmail` package can be removed with the following command:

```
$ sudo yum erase sendmail
```

service_postfix_enabled

Enable Postfix Service

The Postfix mail transfer agent is used for local mail delivery within the system. The default configuration only listens for connections to the default SMTP port (port 25) on the loopback interface (127.0.0.1). It is recommended to leave this service enabled for local mail delivery. The `postfix` service can be enabled with the following command:

```
$ sudo systemctl enable postfix.service
```

FTP Server

FTP is a common method for allowing remote access to files. Like telnet, the FTP protocol is unencrypted, which means that passwords and other data transmitted during the session can be captured and that the session is vulnerable to hijacking. Therefore, running the FTP server software is not recommended.

However, there are some FTP server configurations which may be appropriate for some environments, particularly those which allow only read-only anonymous access as a means of downloading data available to the public.

Configure vsftpd to Provide FTP Service if Necessary

The primary vsftpd configuration file is `/etc/vsftpd.conf`, if that file exists, or `/etc/vsftpd/vsftpd.conf` if it does not.

Restrict the Set of Users Allowed to Access FTP

This section describes how to disable non-anonymous (password-based) FTP logins, or, if it is not possible to do this entirely due to legacy applications, how to restrict insecure FTP login to only those users who have an identified need for this access.

ftp_restrict_to_anon

Restrict Access to Anonymous Users if Possible

Is there a mission-critical reason for users to transfer files to/from their own accounts using FTP, rather than using a secure protocol like SCP/SFTP? If not, edit the vsftpd configuration file. Add or correct the following configuration option:

```
local_enable=NO
```

If non-anonymous FTP logins are necessary, follow the guidance in the remainder of this section to secure these logins as much as possible.

ftp_limit_users

Limit Users Allowed FTP Access if Necessary

If there is a mission-critical reason for users to access their accounts via the insecure FTP protocol, limit the set of users who are allowed this access. Edit the vsftpd configuration file. Add or correct the following configuration options:

```
userlist_enable=YES
userlist_file=/etc/vsftp.ftpusers
userlist_deny=NO
```

Edit the file `/etc/vsftp.ftpusers`. For each user USERNAME who should be allowed to access the system via FTP, add a line containing that user's name:

```
USERNAME
```

If anonymous access is also required, add the anonymous usernames to `/etc/vsftp.ftpusers` as well.

```
anonymous
ftp
```

ftp_present_banner

Create Warning Banners for All FTP Users

Edit the vsftpd configuration file, which resides at `/etc/vsftpd/vsftpd.conf` by default. Add or correct the following configuration options:

```
banner_file=/etc/issue
```

ftp_log_transactions

Enable Logging of All FTP Transactions

Add or correct the following configuration options within the `vsftpd` configuration file, located at `/etc/vsftpd/vsftpd.conf`:

```
xferlog_enable=YES
xferlog_std_format=NO
log_ftp_protocol=YES
```

ftp_disable_uploads

Disable FTP Uploads if Possible

Is there a mission-critical reason for users to upload files via FTP? If not, edit the vsftpd configuration file to add or correct the following configuration options:

```
write_enable=NO
```

If FTP uploads are necessary, follow the guidance in the remainder of this section to secure these transactions as much as possible.

ftp_home_partition

Place the FTP Home Directory on its Own Partition

By default, the anonymous FTP root is the home directory of the FTP user account. The df command can be used to verify that this directory is on its own partition.

ftp_configure_firewall

Configure Firewalls to Protect the FTP Server

By default, `iptables` blocks access to the ports used by the web server. To configure `iptables` to allow port 21 traffic, one must edit `/etc/sysconfig/iptables` and `/etc/sysconfig/ip6tables` (if IPv6 is in use). Add the following line, ensuring that it appears before the final LOG and DROP lines for the INPUT chain:

```
-A INPUT -m state --state NEW -p tcp --dport 21 -j ACCEPT
```

Edit the file `/etc/sysconfig/iptables-config`. Ensure that the space-separated list of modules contains the FTP connection tracking module:

```
IPTABLES_MODULES="ip_conntrack_ftp"
```

Use vsftpd to Provide FTP Service if Necessary

If your use-case requires FTP service, install and set-up vsftpd to provide it.

package_vsftpd_installed

Install vsftpd Package

If this system must operate as an FTP server, install the `vsftpd` package via the standard channels. The `vsftpd` package can be installed with the following command:

```
$ sudo yum install vsftpd
```

Disable vsftpd if Possible

To minimize attack surface, disable vsftpd if at all possible.

service_vsftpd_disabled

Disable vsftpd Service

The `vsftpd` service can be disabled with the following command:

```
$ sudo systemctl disable vsftpd.service
```

package_vsftpd_removed

Uninstall vsftpd Package

The `vsftpd` package can be removed with the following command:

```
$ sudo yum erase vsftpd
```

Network Time Protocol

The Network Time Protocol is used to manage the system clock over a network. Computer clocks are not very accurate, so time will drift unpredictably on unmanaged systems. Central time protocols can be used both to ensure that time is consistent among a network of systems, and that their time is consistent with the outside world.

If every system on a network reliably reports the same time, then it is much easier to correlate log messages in case of an attack. In addition, a number of cryptographic protocols (such as Kerberos) use timestamps to prevent certain types of attacks. If your network does not have synchronized time, these protocols may be unreliable or even unusable.

Depending on the specifics of the network, global time accuracy may be just as important as local synchronization, or not very important at all. If your network is connected to the Internet, using a public timeserver (or one provided by your enterprise) provides globally accurate timestamps which may be essential in investigating or responding to an attack which originated outside of your network.

A typical network setup involves a small number of internal systems operating as NTP servers, and the remainder obtaining time information from those internal servers.

There is a choice between the daemons `ntpd` and `chronyd`, which are available from the repositories in the `ntp` and `chrony` packages respectively.

The default `chronyd` daemon can work well when external time references are only intermittently accesible, can perform well even when the network is congested for longer periods of time, can usually synchronize the clock faster and with better time accuracy, and quickly adapts to sudden changes in the rate of the clock, for example, due to changes in the temperature of the crystal oscillator. `Chronyd` should be considered for all systems which are frequently suspended or otherwise intermittently disconnected and reconnected to a network. Mobile and virtual systems for example.

The `ntpd` NTP daemon fully supports NTP protocol version 4 (RFC 5905), including broadcast, multicast, manycast clients and servers, and the orphan mode. It also supports extra authentication schemes based on public-key cryptography (RFC 5906). The NTP daemon (`ntpd`) should be considered for systems which are normally kept permanently on. Systems which are required to use broadcast or multicast IP, or to perform authentication of packets with the `Autokey` protocol, should consider using `ntpd`.

Refer to https://docs.fedoraproject.org/en-US/fedora/rawhide/system-administrators-guide/servers/Configuring_NTP_Using_the_chrony_Suite/ for more detailed comparison of features of `chronyd` and `ntpd` daemon features respectively, and for further guidance how to choose between the two NTP daemons.

The upstream manual pages at http://chrony.tuxfamily.org/manual.html for `chronyd` and http://www.ntp.org for `ntpd` provide additional information on the capabilities and configuration of each of the NTP daemons.

package_ntp_installed

Install the ntp service

The ntpd service should be installed.

service_timesyncd_enabled

Enable systemd_timesyncd Service

The `systemd_timesyncd` service can be enabled with the following command:

```
$ sudo systemctl enable systemd_timesyncd.service
```

service_chronyd_or_ntpd_enabled

Enable the NTP Daemon

Run the following command to determine the current status of the `chronyd` service:

```
$ systemctl is-active chronyd
```

If the service is running, it should return the following:

```
active
```

Note: The `chronyd` daemon is enabled by default.

Run the following command to determine the current status of the `ntpd` service:

```
$ systemctl is-active ntpd
```

If the service is running, it should return the following:

```
active
```

Note: The `ntpd` daemon is not enabled by default. Though as mentioned in the previous sections in certain environments the `ntpd` daemon might be preferred to be used rather than the `chronyd` one. Refer to: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/System_Administrators_Guide/ch-Configuring_NTP_Using_the_chrony_Suite.html for guidance which NTP daemon to choose depending on the environment used.

chronyd_or_ntpd_set_maxpoll

Configure Time Service Maxpoll Interval

The `maxpoll` should be configured to in `/etc/ntp.conf` or `/etc/chrony.conf` to continuously poll time servers. To configure `maxpoll` in `/etc/ntp.conf` or `/etc/chrony.conf` add the following:

```
maxpoll
```

ntpd_specify_remote_server

Specify a Remote NTP Server

To specify a remote NTP server for time synchronization, edit the file `/etc/ntp.conf`. Add or correct the following lines, substituting the IP or hostname of a remote NTP server for *ntpserver*:

```
server ntpserver
```

This instructs the NTP software to contact that remote server to obtain time data.

ntpd_specify_multiple_servers

Specify Additional Remote NTP Servers

Additional NTP servers can be specified for time synchronization in the file `/etc/ntp.conf`. To do so, add additional lines of the following form, substituting the IP address or hostname of a remote NTP server for *ntpserver*:

```
server ntpserver
```

service_ntpd_enabled

Enable the NTP Daemon

The `ntpd` service can be enabled with the following command:

```
$ sudo systemctl enable ntpd.service
```

chronyd_or_ntpd_specify_multiple_servers

Specify Additional Remote NTP Servers

Depending on specific functional requirements of a concrete production environment, the Red Hat Enterprise Linux 8 system can be configured to utilize the services of the `chronyd` NTP daemon (the default), or services of the `ntpd` NTP daemon. Refer to https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/System_Administrators_Guide/ch-Configuring_NTP_Using_the_chrony_Suite.html for more detailed comparison of the features of both of the choices, and for further guidance how to choose between the two NTP daemons. Additional NTP servers can be specified for time synchronization. To do so, perform the following:

- if the system is configured to use the `chronyd` as the NTP daemon (the default), edit the file `/etc/chrony.conf` as follows,
- if the system is configured to use the `ntpd` as the NTP daemon, edit the file `/etc/ntp.conf` as documented below.

Add additional lines of the following form, substituting the IP address or hostname of a remote NTP server for *ntpserver*:

```
server ntpserver
```

service_ntp_enabled

Enable the NTP Daemon

The `ntpd` service can be enabled with the following command:

```
$ sudo systemctl enable ntpd.service
```

chronyd_or_ntpd_specify_remote_server

Specify a Remote NTP Server

Depending on specific functional requirements of a concrete production environment, the Red Hat Enterprise Linux 8 system can be configured to utilize the services of the `chronyd` NTP daemon (the default), or services of the `ntpd` NTP daemon. Refer to https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/System_Administrators_Guide/ch-Configuring_NTP_Using_the_chrony_Suite.html for more detailed comparison of the features of both of the choices, and for further guidance how to choose between the two NTP daemons. To specify a remote NTP server for time synchronization, perform the following:

- if the system is configured to use the `chronyd` as the NTP daemon (the default), edit the file `/etc/chrony.conf` as follows,
- if the system is configured to use the `ntpd` as the NTP daemon, edit the file `/etc/ntp.conf` as documented below.

Add or correct the following lines, substituting the IP or hostname of a remote NTP server for *ntpserver*:

```
server ntpserver
```

This instructs the NTP software to contact that remote server to obtain time data.

Proxy Server

A proxy server is a very desirable target for a potential adversary because much (or all) sensitive data for a given infrastructure may flow through it. Therefore, if one is required, the system acting as a proxy server should be dedicated to that purpose alone and be stored in a physically secure location. The system's default proxy server software is Squid, and provided in an RPM package of the same name.

Disable Squid if Possible

If Squid was installed and activated, but the system does not need to act as a proxy server, then it should be disabled and removed.

service_squid_disabled

Disable Squid

The `squid` service can be disabled with the following command:

```
$ sudo systemctl disable squid.service
```

package_squid_removed

Uninstall squid Package

The `squid` package can be removed with the following command:

```
$ sudo yum erase squid
```

Avahi Server

The Avahi daemon implements the DNS Service Discovery and Multicast DNS protocols, which provide service and host discovery on a network. It allows a system to automatically identify resources on the network, such as printers or web servers. This capability is also known as mDNSresponder and is a major part of Zeroconf networking.

Configure Avahi if Necessary

If your system requires the Avahi daemon, its configuration can be restricted to improve security. The Avahi daemon configuration file is `/etc/avahi/avahi-daemon.conf`. The following security recommendations should be applied to this file: See the `avahi-daemon.conf(5)` man page, or documentation at http://www.avahi.org, for more detailed information about the configuration options.

avahi_restrict_published_information

Restrict Information Published by Avahi

If it is necessary to publish some information to the network, it should not be joined by any extraneous information, or by information supplied by a non-trusted source on the system. Prevent user applications from using Avahi to publish services by adding or correcting the following line in the `[publish]` section:

```
disable-user-service-publishing=yes
```

Implement as many of the following lines as possible, to restrict the information published by Avahi.

```
publish-addresses=no
publish-hinfo=no
publish-workstation=no
publish-domain=no
```

Inspect the files in the directory `/etc/avahi/services/`. Unless there is an operational need to publish information about each of these services, delete the corresponding file.

avahi_check_ttl

Check Avahi Responses' TTL Field

To make Avahi ignore packets unless the TTL field is 255, edit `/etc/avahi/avahi-daemon.conf` and ensure the following line appears in the `[server]` section:

```
check-response-ttl=yes
```

avahi_prevent_port_sharing

Prevent Other Programs from Using Avahi's Port

To prevent other mDNS stacks from running, edit `/etc/avahi/avahi-daemon.conf` and ensure the following line appears in the `[server]` section:

```
disallow-other-stacks=yes
```

avahi_disable_publishing

Disable Avahi Publishing

To prevent Avahi from publishing its records, edit `/etc/avahi/avahi-daemon.conf` and ensure the following line appears in the `[publish]` section:

```
disable-publishing=yes
```

avahi_ip_only

Serve Avahi Only via Required Protocol

If you are using only IPv4, edit `/etc/avahi/avahi-daemon.conf` and ensure the following line exists in the `[server]` section:

```
use-ipv6=no
```

Similarly, if you are using only IPv6, disable IPv4 sockets with the line:

```
use-ipv4=no
```

Disable Avahi Server if Possible

Because the Avahi daemon service keeps an open network port, it is subject to network attacks. Disabling it can reduce the system's vulnerability to such attacks.

service_avahi-daemon_disabled

Disable Avahi Server Software

The `avahi-daemon` service can be disabled with the following command:

```
$ sudo systemctl disable avahi-daemon.service
```

LDAP

LDAP is a popular directory service, that is, a standardized way of looking up information from a central database. Red Hat Enterprise Linux 8 includes software that enables a system to act as both an LDAP client and server.

Configure OpenLDAP Server

This section details some security-relevant settings for an OpenLDAP server.

Install and Protect LDAP Certificate Files

Create the PKI directory for LDAP certificates if it does not already exist:

```
$ sudo mkdir /etc/pki/tls/ldap
$ sudo chown root:root /etc/pki/tls/ldap
$ sudo chmod 755 /etc/pki/tls/ldap
```

Using removable media or some other secure transmission format, install the certificate files onto the LDAP server:

- `/etc/pki/tls/ldap/serverkey.pem`: the private key `ldapserverkey.pem`
- `/etc/pki/tls/ldap/servercert.pem`: the certificate file `ldapservercert.pem`

Verify the ownership and permissions of these files:

```
$ sudo chown root:ldap /etc/pki/tls/ldap/serverkey.pem
$ sudo chown root:ldap /etc/pki/tls/ldap/servercert.pem
$ sudo chmod 640 /etc/pki/tls/ldap/serverkey.pem
$ sudo chmod 640 /etc/pki/tls/ldap/servercert.pem
```

Verify that the CA's public certificate file has been installed as `/etc/pki/tls/CA/cacert.pem`, and has the correct permissions:

```
$ sudo mkdir /etc/pki/tls/CA
$ sudo chown root:root /etc/pki/tls/CA/cacert.pem
$ sudo chmod 644 /etc/pki/tls/CA/cacert.pem
```

As a result of these steps, the LDAP server will have access to its own private certificate and the key with which that certificate is encrypted, and to the public certificate file belonging to the CA. Note that it would be possible for the key to be protected further, so that processes running as ldap could not read it. If this were done, the LDAP server process would need to be restarted manually whenever the server rebooted.

package_openldap-servers_removed

Uninstall openldap-servers Package

The `openldap-servers` package should be removed if not in use. Is this system the OpenLDAP server? If not, remove the package. The `openldap-servers` package can be removed with the following command:

```
$ sudo yum erase openldap-servers
```

The openldap-servers RPM is not installed by default on a Red Hat Enterprise Linux 8 system. It is needed only by the OpenLDAP server, not by the clients which use LDAP for authentication. If the system is not intended for use as an LDAP Server it should be removed.

Configure OpenLDAP Clients

This section provides information on which security settings are important to configure in OpenLDAP clients by manually editing the appropriate configuration files. Red Hat Enterprise Linux 8 provides an automated configuration tool called authconfig and a graphical wrapper for authconfig called `system-config-authentication`. However, these tools do not provide as much control over configuration as manual editing of configuration files. The authconfig tools do not allow you to specify locations of SSL certificate files, which is useful when trying to use SSL cleanly across several protocols. Installation and configuration of OpenLDAP on Red Hat Enterprise Linux 8 is available at

enable_ldap_client

Enable the LDAP Client For Use in Authconfig

To determine if LDAP is being used for authentication, use the following command:

```
$ sudo grep -i useldapauth /etc/sysconfig/authconfig
```

If `USELDAPAUTH=yes`, then LDAP is being used. If not, set `USELDAPAUTH` to `yes`.

ldap_client_start_tls

Configure LDAP Client to Use TLS For All Transactions

This check verifies that Red Hat Enterprise Linux 8 implements cryptography to protect the integrity of remote LDAP authentication sessions.

To determine if LDAP is being used for authentication, use the following command:

```
$ sudo grep -i useldapauth /etc/sysconfig/authconfig
```

If `USELDAPAUTH=yes`, then LDAP is being used. To check if LDAP is configured to use TLS, use the following command:

```
$ sudo grep -i ssl /etc/pam_ldap.conf
```

ldap_client_tls_cacertpath

Configure Certificate Directives for LDAP Use of TLS

Ensure a copy of a trusted CA certificate has been placed in the file `/etc/pki/tls/CA/cacert.pem`. Configure LDAP to enforce TLS use and to trust certificates signed by that CA. First, edit the file `/etc/nslcd.conf`, and add or correct either of the following lines:

```
tls_cacertdir /etc/pki/tls/CA
```

or

```
tls_cacertfile /etc/pki/tls/CA/cacert.pem
```

Then review the LDAP server and ensure TLS has been configured.

Print Support

The Common Unix Printing System (CUPS) service provides both local and network printing support. A system running the CUPS service can accept print jobs from other systems, process them, and send them to the appropriate printer. It also provides an interface for remote administration through a web browser. The CUPS service is installed and activated by default. The project homepage and more detailed documentation are available at http://www.cups.org.

Configure the CUPS Service if Necessary

CUPS provides the ability to easily share local printers with other systems over the network. It does this by allowing systems to share lists of available printers. Additionally, each system that runs the CUPS service can potentially act as a print server. Whenever possible, the printer sharing and print server capabilities of CUPS should be limited or disabled. The following recommendations should demonstrate how to do just that.

cups_disable_printserver

Disable Print Server Capabilities

To prevent remote users from potentially connecting to and using locally configured printers, disable the CUPS print server sharing capabilities. To do so, limit how the server will listen for print jobs by removing the more generic port directive from /etc/cups/cupsd.conf:

```
Port 631
```

and replacing it with the `Listen` directive:

```
Listen localhost:631
```

This will prevent remote users from printing to locally configured printers while still allowing local users on the system to print normally.

cups_disable_browsing

Disable Printer Browsing Entirely if Possible

By default, CUPS listens on the network for printer list broadcasts on UDP port 631. This functionality is called printer browsing. To disable printer browsing entirely, edit the CUPS configuration file, located at `/etc/cups/cupsd.conf`, to include the following:

```
Browsing Off
BrowseAllow none
```

service_cups_disabled

Disable the CUPS Service

The `cups` service can be disabled with the following command:

```
$ sudo systemctl disable cups.service
```

Samba(SMB) Microsoft Windows File Sharing Server

When properly configured, the Samba service allows Linux systems to provide file and print sharing to Microsoft Windows systems. There are two software packages that provide Samba support. The first, `samba-client`, provides a series of command line tools that enable a client system to access Samba shares. The second, simply labeled `samba`, provides the Samba service. It is this second package that allows a Linux system to act as an Active Directory server, a domain controller, or as a domain member. Only the `samba-client` package is installed by default.

Configure Samba if Necessary

All settings for the Samba daemon can be found in `/etc/samba/smb.conf`. Settings are divided between a `[global]` configuration section and a series of user created share definition sections meant to describe file or print shares on the system. By default, Samba will operate in user mode and allow client systems to access local home directories and printers. It is recommended that these settings be changed or that additional limitations be set in place.

Restrict SMB File Sharing to Configured Networks

Only users with local user accounts will be able to log in to Samba shares by default. Shares can be limited to particular users or network addresses. Use the `hosts allow` and `hosts deny` directives accordingly, and consider setting the valid users directive to a limited subset of users or to a group of users. Separate each address, user, or user group with a space as follows for a particular *share* or global:

```
[share]
  hosts allow = 192.168.1. 127.0.0.1
  valid users = userone usertwo @usergroup
```

It is also possible to limit read and write access to particular users with the read list and write list options, though the permissions set by the system itself will override these settings. Set the read only attribute for each share to ensure that global settings will not accidentally override the individual share settings. Then, as with the valid users directive, separate each user or group of users with a space:

```
[share]
  read only = yes
  write list = userone usertwo @usergroup
```

Restrict Printer Sharing

By default, Samba utilizes the CUPS printing service to enable printer sharing with Microsoft Windows workstations. If there are no printers on the local system, or if printer sharing with Microsoft Windows is not required, disable the printer sharing capability by commenting out the following lines, found in `/etc/samba/smb.conf`:

```
[global]
  load printers = yes
  cups options = raw
[printers]
  comment = All Printers
  path = /usr/spool/samba
  browseable = no
  guest ok = no
  writable = no
  printable = yes
```

There may be other options present, but these are the only options enabled and uncommented by default. Removing the `[printers]` share should be enough for most users. If the Samba printer sharing capability is needed, consider disabling the Samba network browsing capability or restricting access to a particular set of users or network addresses. Set the `valid users` parameter to a small subset of users or restrict it to a particular group of users with the shorthand `@`. Separate each user or group of users with a space. For example, under the `[printers]` share:

```
[printers]
  valid users = user @printerusers
```

require_smb_client_signing

Require Client SMB Packet Signing, if using smbclient

To require samba clients running `smbclient` to use packet signing, add the following to the `[global]` section of the Samba configuration file, `/etc/samba/smb.conf`:

```
client signing = mandatory
```

Requiring samba clients such as `smbclient` to use packet signing ensures they can only communicate with servers that support packet signing.

smb_server_disable_root

Disable Root Access to SMB Shares

Administrators should not use administrator accounts to access Samba file and printer shares. Disable the root user and the wheel administrator group:

```
[share]
  invalid users = root @wheel
```

If administrator accounts cannot be disabled, ensure that local system passwords and Samba service passwords do not match.

mount_option_smb_client_signing

Require Client SMB Packet Signing, if using mount.cifs

Require packet signing of clients who mount Samba shares using the `mount.cifs` program (e.g., those who specify shares in `/etc/fstab`). To do so, ensure signing options (either `sec=krb5i` or `sec=ntlmv2i`) are used.

See the `mount.cifs(8)` man page for more information. A Samba client should only communicate with servers who can support SMB packet signing.

package_samba-common_installed

Install the Samba Common Package

The `samba-common` package should be installed. The `samba-common` package can be installed with the following command:

```
$ sudo yum install samba-common
```

Disable Samba if Possible

Even after the Samba server package has been installed, it will remain disabled. Do not enable this service unless it is absolutely necessary to provide Microsoft Windows file and print sharing functionality.

package_samba_removed

Uninstall Samba Package

The `samba` package can be removed with the following command:

```
$ sudo yum erase samba
```

service_smb_disabled

Disable Samba

The `smb` service can be disabled with the following command:

```
$ sudo systemctl disable smb.service
```

Cron and At Daemons

The cron and at services are used to allow commands to be executed at a later time. The cron service is required by almost all systems to perform necessary maintenance tasks, while at may or may not be required on a given system. Both daemons should be configured defensively.

Restrict at and cron to Authorized Users if Necessary

The `/etc/cron.allow` and `/etc/at.allow` files contain lists of users who are allowed to use `cron` and at to delay execution of processes. If these files exist and if the corresponding files `/etc/cron.deny` and `/etc/at.deny` do not exist, then only users listed in the relevant allow files can run the crontab and `at` commands to submit jobs to be run at scheduled intervals. On many systems, only the system administrator needs the ability to schedule jobs. Note that even if a given user is not listed in `cron.allow`, cron jobs can still be run as that user. The `cron.allow` file controls only administrative access to the crontab command for scheduling and modifying cron jobs.

To restrict `at` and `cron` to only authorized users:

- Remove the `cron.deny` file:

```
$ sudo rm /etc/cron.deny
```

- Edit `/etc/cron.allow`, adding one line for each user allowed to use the crontab command to create cron jobs.
- Remove the `at.deny` file:

```
$ sudo rm /etc/at.deny
```

- Edit `/etc/at.allow`, adding one line for each user allowed to use the at command to create at jobs.

file_groupowner_cron_allow

Verify Group Who Owns /etc/cron.allow file

If `/etc/cron.allow` exists, it must be group-owned by `root`. To properly set the group owner of `/etc/cron.allow`, run the command:

```
$ sudo chgrp root /etc/cron.allow
```

file_owner_cron_allow

Verify User Who Owns /etc/cron.allow file

If `/etc/cron.allow` exists, it must be owned by `root`. To properly set the owner of `/etc/cron.allow`, run the command:

```
$ sudo chown root /etc/cron.allow
```

service_atd_disabled

Disable At Service (atd)

The `at` and `batch` commands can be used to schedule tasks that are meant to be executed only once. This allows delayed execution in a manner similar to cron, except that it is not recurring. The daemon `atd` keeps track of tasks scheduled via `at` and `batch`, and executes them at the specified time. The `atd` service can be disabled with the following command:

```
$ sudo systemctl disable atd.service
```

service_cron_enabled

Enable cron Service

The `crond` service is used to execute commands at preconfigured times. It is required by almost all systems to perform necessary maintenance tasks, such as notifying root of system activity. The `cron` service can be enabled with the following command:

```
$ sudo systemctl enable cron.service
```

service_crond_enabled

Enable cron Service

The `crond` service is used to execute commands at preconfigured times. It is required by almost all systems to perform necessary maintenance tasks, such as notifying root of system activity. The `crond` service can be enabled with the following command:

```
$ sudo systemctl enable crond.service
```

disable_anacron

Disable anacron Service

The `cronie-anacron` package, which provides `anacron` functionality, is installed by default. The `cronie-anacron` package can be removed with the following command:

```
$ sudo yum erase cronie-anacron
```

package_cron_installed

Install the cron service

The Cron service should be installed.

Deprecated services

Some deprecated software services impact the overall system security due to their behavior (leak of confidentiality in network exchange, usage as uncontrolled communication channel, risk associated with the service due to its old age, etc.

package_ntpdate_removed

Uninstall the ntpdate package

ntpdate is a historical ntp synchronization client for unixes. It sould be uninstalled.

package_nis_removed

Uninstall the nis package

The support for Yellowpages should not be installed unless it is required.

package_inetutils-telnetd_removed

Uninstall the inet-based telnet server

The inet-based telnet daemon should be uninstalled.

package_telnetd_removed

Uninstall the telnet server

The telnet daemon should be uninstalled.

package_telnetd-ssl_removed

Uninstall the ssl compliant telnet server

The telnet daemon, even with ssl support, should be uninstalled.

NFS and RPC

The Network File System is a popular distributed filesystem for the Unix environment, and is very widely deployed. This section discusses the circumstances under which it is possible to disable NFS and its dependencies, and then details steps which should be taken to secure NFS's configuration. This section is relevant to systems operating as NFS clients, as well as to those operating as NFS servers.

Configure NFS Clients

The steps in this section are appropriate for systems which operate as NFS clients.

Mount Remote Filesystems with Restrictive Options

Edit the file `/etc/fstab`. For each filesystem whose type (column 3) is `nfs` or `nfs4`, add the text , `nodev`, `nosuid` to the list of mount options in column 4. If appropriate, also add , `noexec`.

See the section titled "Restrict Partition Mount Options" for a description of the effects of these options. In general, execution of files mounted via NFS should be considered risky because of the possibility that an adversary could intercept the request and substitute a malicious file. Allowing setuid files to be executed from remote servers is particularly risky, both for this reason and because it requires the clients to extend root-level trust to the NFS server.

mount_option_noexec_remote_filesystems

Mount Remote Filesystems with noexec

Add the `noexec` option to the fourth column of `/etc/fstab` for the line which controls mounting of any NFS mounts.

mount_option_nosuid_remote_filesystems

Mount Remote Filesystems with nosuid

Add the `nosuid` option to the fourth column of `/etc/fstab` for the line which controls mounting of any NFS mounts.

mount_option_nodev_remote_filesystems

Mount Remote Filesystems with nodev

Add the `nodev` option to the fourth column of `/etc/fstab` for the line which controls mounting of any NFS mounts.

mount_option_krb_sec_remote_filesystems

Mount Remote Filesystems with Kerberos Security

Add the `sec=krb5:krb5i:krb5p` option to the fourth column of `/etc/fstab` for the line which controls mounting of any NFS mounts.

Disable NFS Server Daemons

There is no need to run the NFS server daemons `nfs` and `rpcsvcgssd` except on a small number of properly secured systems designated as NFS servers. Ensure that these daemons are turned off on clients.

service_rpcsvcgssd_disabled

Disable Secure RPC Server Service (rpcsvcgssd)

The rpcsvcgssd service manages RPCSEC GSS contexts required to secure protocols that use RPC (most often Kerberos and NFS). The rpcsvcgssd service is the server-side of RPCSEC GSS. If the system does not require secure RPC then this service should be disabled. The `rpcsvcgssd` service can be disabled with the following command:

```
$ sudo systemctl disable rpcsvcgssd.service
```

service_nfs_disabled

Disable Network File System (nfs)

The Network File System (NFS) service allows remote hosts to mount and interact with shared filesystems on the local system. If the local system is not designated as a NFS server then this service should be disabled. The `nfs` service can be disabled with the following command:

```
$ sudo systemctl disable nfs.service
```

nfs_no_anonymous

Specify UID and GID for Anonymous NFS Connections

To specify the UID and GID for remote root users, edit the `/etc/exports` file and add the following for each export:

```
anonuid=value greater than UID_MAX from /etc/login.defs
anongid=value greater than GID_MAX from /etc/login.defs
```

Note that a value of "-1" is technically acceptable as this will randomize the `anonuid` and `anongid` values on a Red Hat Enterprise Linux 6 based NFS server. While acceptable from a security perspective, a value of `-1` may cause interoperability issues, particularly with Red Hat Enterprise Linux 7 client systems. Alternatively, functionally equivalent values of 60001, 65534, 65535 may be used.

Configure All Systems which Use NFS

The steps in this section are appropriate for all systems which run NFS, whether they operate as clients or as servers.

Configure NFS Services to Use Fixed Ports (NFSv3 and NFSv2)

Firewalling should be done at each host and at the border firewalls to protect the NFS daemons from remote access, since NFS servers should never be accessible from outside the organization. However, by default for NFSv3 and NFSv2, the RPC Bind service assigns each NFS service to a port dynamically at service startup time. Dynamic ports cannot be protected by port filtering firewalls such as `iptables`.

Therefore, restrict each service to always use a given port, so that firewalling can be done effectively. Note that, because of the way RPC is implemented, it is not possible to disable the RPC Bind service even if ports are assigned statically to all RPC services.

In NFSv4, the mounting and locking protocols have been incorporated into the protocol, and the server listens on the the well-known TCP port 2049. As such, NFSv4 does not need to interact with the `rpcbind, lockd, and rpc.statd` daemons, which can and should be disabled in a pure NFSv4 environment. The `rpc.mountd` daemon is still required on the NFS server to setup exports, but is not involved in any over-the-wire operations.

nfs_fixed_lockd_udp_port

Configure lockd to use static UDP port

Configure the `lockd` daemon to use a static UDP port as opposed to letting the RPC Bind service dynamically assign a port. Edit the file `/etc/sysconfig/nfs`. Add or correct the following line:

```
LOCKD_UDPPORT=lockd-port
```

Where `lockd-port` is a port which is not used by any other service on your network.

nfs_fixed_lockd_tcp_port

Configure lockd to use static TCP port

Configure the `lockd` daemon to use a static TCP port as opposed to letting the RPC Bind service dynamically assign a port. Edit the file `/etc/sysconfig/nfs`. Add or correct the following line:

```
LOCKD_TCPPORT=lockd-port
```

Where `lockd-port` is a port which is not used by any other service on your network.

nfs_fixed_mountd_port

Configure mountd to use static port

Configure the `mountd` daemon to use a static port as opposed to letting the RPC Bind service dynamically assign a port. Edit the file `/etc/sysconfig/nfs`. Add or correct the following line:

```
MOUNTD_PORT=statd-port
```

Where `mountd-port` is a port which is not used by any other service on your network.

nfs_fixed_statd_port

Configure statd to use static port

Configure the `statd` daemon to use a static port as opposed to letting the RPC Bind service dynamically assign a port. Edit the file `/etc/sysconfig/nfs`. Add or correct the following line:

```
STATD_PORT=statd-port
```

Where `statd-port` is a port which is not used by any other service on your network.

Make Each System a Client or a Server, not Both

If NFS must be used, it should be deployed in the simplest configuration possible to avoid maintainability problems which may lead to unnecessary security exposure. Due to the reliability and security problems caused by NFS (specially NFSv3 and NFSv2), it is not a good idea for systems which act as NFS servers to also mount filesystems via NFS. At the least, crossed mounts (the situation in which each of two servers mounts a filesystem from the other) should never be used.

Disable All NFS Services if Possible

If there is not a reason for the system to operate as either an NFS client or an NFS server, follow all instructions in this section to disable subsystems required by NFS.

Disable netfs if Possible

To determine if any network filesystems handled by netfs are currently mounted on the system execute the following command:

```
$ mount -t nfs,nfs4,smbfs,cifs,ncpfs
```

If the command did not return any output then disable netfs.

service_netfs_disabled

Disable Network File Systems (netfs)

The netfs script manages the boot-time mounting of several types of networked filesystems, of which NFS and Samba are the most common. If these filesystem types are not in use, the script can be disabled, protecting the system somewhat against accidental or malicious changes to `/etc/fstab` and against flaws in the netfs script itself. The `netfs` service can be disabled with the following command:

```
$ sudo systemctl disable netfs.service
```

Disable Services Used Only by NFS

If NFS is not needed, disable the NFS client daemons nfslock, rpcgssd, and rpcidmapd.

All of these daemons run with elevated privileges, and many listen for network connections. If they are not needed, they should be disabled to improve system security posture.

service_rpcbind_disabled

Disable rpcbind Service

The rpcbind utility maps RPC services to the ports on which they listen. RPC processes notify rpcbind when they start, registering the ports they are listening on and the RPC program numbers they expect to serve. The rpcbind service redirects the client to the proper port number so it can communicate with the requested service. If the system does not require RPC (such as for NFS servers) then this service should be disabled. The `rpcbind` service can be disabled with the following command:

```
$ sudo systemctl disable rpcbind.service
```

service_rpcgssd_disabled

Disable Secure RPC Client Service (rpcgssd)

The rpcgssd service manages RPCSEC GSS contexts required to secure protocols that use RPC (most often Kerberos and NFS). The rpcgssd service is the client-side of RPCSEC GSS. If the system does not require secure RPC then this service should be disabled. The `rpcgssd` service can be disabled with the following command:

```
$ sudo systemctl disable rpcgssd.service
```

service_nfslock_disabled

Disable Network File System Lock Service (nfslock)

The Network File System Lock (nfslock) service starts the required remote procedure call (RPC) processes which allow clients to lock files on the server. If the local system is not configured to mount NFS filesystems then this service should be disabled. The `nfslock` service can be disabled with the following command:

```
$ sudo systemctl disable nfslock.service
```

service_rpcidmapd_disabled

Disable RPC ID Mapping Service (rpcidmapd)

The rpcidmapd service is used to map user names and groups to UID and GID numbers on NFSv4 mounts. If NFS is not in use on the local system then this service should be disabled. The `rpcidmapd` service can be disabled with the following command:

```
$ sudo systemctl disable rpcidmapd.service
```

Configure NFS Servers

The steps in this section are appropriate for systems which operate as NFS servers.

Use Access Lists to Enforce Authorization Restrictions

When configuring NFS exports, ensure that each export line in `/etc/exports` contains a list of hosts which are allowed to access that export. If no hosts are specified on an export line, then that export is available to any remote host which requests it. All lines of the exports file should specify the hosts (or subnets, if needed) which are allowed to access the exported directory, so that unknown or remote hosts will be denied.

Authorized hosts can be specified in several different formats:

- Name or alias that is recognized by the resolver
- Fully qualified domain name
- IP address
- IP subnets in the format `address/netmask` or `address/CIDR`

Configure the Exports File Restrictively

Linux's NFS implementation uses the file `/etc/exports` to control what filesystems and directories may be accessed via NFS. (See the `exports(5)` manpage for more information about the format of this file.)

The syntax of the `exports` file is not necessarily checked fully on reload, and syntax errors can leave your NFS configuration more open than intended. Therefore, exercise caution when modifying the file.

The syntax of each line in `/etc/exports` is:

```
/DIR    host1(opt1,opt2) host2(opt3)
```

where `/DIR` is a directory or filesystem to export, `hostN` is an IP address, netblock, hostname, domain, or netgroup to which to export, and `optN` is an option.

Export Filesystems Read-Only if Possible

If a filesystem is being exported so that users can view the files in a convenient fashion, but there is no need for users to edit those files, exporting the filesystem read-only removes an attack vector against the server. The default filesystem export mode is `ro`, so do not specify `rw` without a good reason.

use_root_squashing_all_exports

Use Root-Squashing on All Exports

If a filesystem is exported using root squashing, requests from root on the client are considered to be unprivileged (mapped to a user such as nobody). This provides some mild protection against remote abuse of an NFS server. Root squashing is enabled by default, and should not be disabled.

Ensure that no line in `/etc/exports` contains the option `no_root_squash`.

no_insecure_locks_exports

Ensure Insecure File Locking is Not Allowed

By default the NFS server requires secure file-lock requests, which require credentials from the client in order to lock a file. Most NFS clients send credentials with file lock requests, however, there are a few clients that do not send credentials when requesting a file-lock, allowing the client to only be able to lock world-readable files. To get around this, the `insecure_locks` option can be used so these clients can access the desired export. This poses a security risk by potentially allowing the client access to data for which it does not have authorization. Remove any instances of the `insecure_locks` option from the file `/etc/exports`.

restrict_nfs_clients_to_privileged_ports

Restrict NFS Clients to Privileged Ports

By default, the server NFS implementation requires that all client requests be made from ports less than 1024. If your organization has control over systems connected to its network, and if NFS requests are prohibited at the border firewall, this offers some protection against malicious requests from unprivileged users. Therefore, the default should not be changed.

To ensure that the default has not been changed, ensure no line in `/etc/exports` contains the option `insecure`.

use_kerberos_security_all_exports

Use Kerberos Security on All Exports

Using Kerberos on all exported mounts prevents a malicious client or user from impersonating a system user. To cryptography authenticate users to the NFS server, add `sec=krb5:krb5i:krb5p` to each export in `/etc/exports`.

no_all_squash_exports

Ensure All-Squashing Disabled On All Exports

The `all_squash` maps all uids and gids to an anonymous user. This should be disabled by removing any instances of the `all_squash` option from the file `/etc/exports`.

IMAP and POP3 Server

Dovecot provides IMAP and POP3 services. It is not installed by default. The project page at http://www.dovecot.org contains more detailed information about Dovecot configuration.

Configure Dovecot if Necessary

If the system will operate as an IMAP or POP3 server, the dovecot software should be configured securely by following the recommendations below.

Enable SSL Support

SSL should be used to encrypt network traffic between the Dovecot server and its clients. Users must authenticate to the Dovecot server in order to read their mail, and passwords should never be transmitted in clear text. In addition, protecting mail as it is downloaded is a privacy measure, and clients may use SSL certificates to authenticate the server, preventing another system from impersonating the server.

dovecot_configure_ssl_cert

Configure Dovecot to Use the SSL Certificate file

This option tells Dovecot where to find the the mail server's SSL Certificate.

Edit `/etc/dovecot/conf.d/10-ssl.conf` and add or correct the following line (*note: the path below is the default path set by the Dovecot installation. If you are using a different path, ensure you reference the appropriate file*):

```
ssl_cert = </etc/pki/dovecot/certs/dovecot.pem
```

"

dovecot_configure_ssl_key

Configure Dovecot to Use the SSL Key file

This option tells Dovecot where to find the the mail server's SSL Key.

Edit `/etc/dovecot/conf.d/10-ssl.conf` and add or correct the following line (*note: the path below is the default path set by the Dovecot installation. If you are using a different path, ensure you reference the appropriate file*):

```
ssl_key = </etc/pki/dovecot/private/dovecot.pem
```

dovecot_enable_ssl

Enable the SSL flag in /etc/dovecot.conf

To allow clients to make encrypted connections the `ssl` flag in Dovecot's configuration file needs to be set to `yes`.

Edit `/etc/dovecot/conf.d/10-ssl.conf` and add or correct the following line:

```
ssl = yes
```

dovecot_disable_plaintext_auth

Disable Plaintext Authentication

To prevent Dovecot from attempting plaintext authentication of clients, edit `/etc/dovecot/conf.d/10-auth.conf` and add\or correct the following line:

```
disable_plaintext_auth = yes
```

Support Only the Necessary Protocols

Dovecot supports the IMAP and POP3 protocols, as well as SSL-protected versions of those protocols. Configure the Dovecot server to support only the protocols needed by your site. Edit `/etc/dovecot/dovecot.conf`. Add or correct the following lines, replacing `PROTOCOL` with only the subset of protocols (`imap`, `imaps`, `pop3`, `pop3s`) required:

```
protocols = PROTOCOL
```

If possible, require SSL protection for all transactions. The SSL protocol variants listen on alternate ports (995 instead of 110 for pop3s, and 993 instead of 143 for imaps), and require SSL-aware clients. An alternate approach is to listen on the standard port and require the client to use the STARTTLS command before authenticating.

Allow IMAP Clients to Access the Server

The default iptables configuration does not allow inbound access to any services. This modification will allow remote hosts to initiate connections to the IMAP daemon, while keeping all other ports on the server in their default protected state. To configure `iptables` to allow port 143 traffic, one must edit `/etc/sysconfig/iptables` and `/etc/sysconfig/ip6tables` (if IPv6 is in use). Add the following line, ensuring that it appears before the final LOG and DROP lines for the INPUT chain:

```
-A INPUT -m state --state NEW -p tcp --dport 143 -j ACCEPT
```

Disable Dovecot

If the system does not need to operate as an IMAP or POP3 server, the dovecot software should be disabled and removed.

service_dovecot_disabled

Disable Dovecot Service

The `dovecot` service can be disabled with the following command:

```
$ sudo systemctl disable dovecot.service
```

package_dovecot_removed

Uninstall dovecot Package

The `dovecot` package can be removed with the following command:

```
$ sudo yum erase dovecot
```

References