# Configuration Management for IT Systems
# Example Policy

Auto-generated from `puppet` policy files
20 Mar 2015

FACIMUS EXEMPLA

**DISTRIBUTION STATEMENT A**: Approved for public release. This is an EXAMPLE distribution statement; 314159th Example Group, Orthogonal Bases, CO

**DESTRUCTION NOTICE**: Destroy by any method that will prevent disclosure or reconstruction of the document.

# Contents

# Changelog

| Date | Person | Change Description |
|---|---|---|
| 27 Mar 2013 | Jared Jennings | Pulled unclassified LaTeX prose in as well as unclassified Puppet modules |

# Executive Summary

The following table lists the NIST SP 800-53 Security Controls that are satisfied through this artifact.

| IA Control Number | IA Control Name |
|---|---|
| COBR-1 | Protection of Backup and Restoration Assets |
| CODB-1 | Data Backup Procedures |
| COSW-1 | Backup Copies of Critical Software |
| DCCS-1 | Configuration Specifications |
| DCCT-1 | Compliance Testing |
| DCHW-1 | Hardware Baseline |
| DCNR-1 | Non-repudiation |
| DCPP-1 | Ports, Protocols, and Services |
| DCSL-1 | System Library Management Controls |
| DCSS-1 | System State Changes |
| DCSW-1 | Software Baseline |
| EBRU-1 | Remote Access for User Functions |
| ECAN-1 | Access for Need-to-Know |
| ECAR-2 | Audit Record Content—Sensitive Systems |
| ECAT-1 | Audit Trail, Monitoring, Analysis and Reporting |
| ECCD-1 | Changes to Data |
| ECLO-1 | Logon |
| ECLP-1 | Least Privilege |
| ECML-1 | Marking and Labeling |
| ECPA-1 | Privileged Account Control |
| ECRC-1 | Resource Control |
| ECRR-1 | Audit Record Retention |
| ECSC-1 | Security Configuration Compliance |
| ECTB-1 | Audit Trail Backup |
| ECTP-1 | Audit Trail Protection |
| ECWM-1 | Warning Message |
| IAAC-1 | Account Control |
| IAIA-1 | Individual Identification and Authentication |
| IATS-1 | Token and Certificate Standards |
| PESL-1 | Screen Lock |

# Chapter 1

# Introduction

This document is a record of how a number of computers are configured and maintained.

Many of the elements of this policy are motivated by requirements in higher-level policies, such as Department of Defense (DoD) Instruction 8500.2 Information Assurance (IA) Controls [13], the Defense Information Services Agency's (DISA) UNIX Security Requirements Guide (SRG, [6]), or various Air Force Instructions (AFIs). §§2, 3, 5 and 6 show how we meet those requirements.

Under this policy, *hosts* (individual computers, real or virtual) are configured using *Puppet*, an automated, policy-based system configuration tool. §§8 and **??** discuss how administrators can follow this policy to configure systems manually in a contingency situation, or set Puppet in place to enforce the policy automatically, as is usual in production.

The same documents that impose requirements on system configuration also impose requirements on system administrators and users, about what to do and how to do it. §7 initiates users in their responsibilities, and §**??** discusses day-to-day tasks done by administrators.

§§9 and 10 discuss how to maintain the policy and this document, and how properly to automate the installation and removal of software.

Finally, the policy (§11) and its attendant files (§12) follow in all their detail.

## 1.1 Typographical conventions

Much of this document has to do with compliance. Near any statement of status regarding compliance there is a margin note with the name of the specific requirement. For example, in §11.100.4, there is Puppet code which configures hosts for compliance with UNIX SRG PDI GEN000590. Just before that code is a comment explaining what the code does and how that complies with the requirement. Beside the code and discussion, in the margin, is a note, "GEN000590." Where requirements are not applicable, the margin note looks like: "N/A: GEN000590." Where compliance is automated, the margin note looks like: "auto: GEN000590." Where compliance is merely documented, the margin note merely says, "GEN000590." And where we are not yet compliant, the margin note is red.

Section numbers are denoted with §.

## 1.2 Navigational aids

Links between parts of this document abound; if you are viewing it as a PDF file, you can click on any section number you see in the text to visit that section. Your PDF reader may have a list of bookmarks in a sidebar, by which you can easily skip around the document. Even if it doesn't, you should find the entire table of contents to be clickable. You may also find the numerous indices, with their clickable page numbers, to be a useful resource.

## 1.3 Colophon

This document was automatically constructed on the date shown on the title page, from the complete set of Puppet policy files (the *manifest*, in Puppet parlance), which contain the policy as enforced on that date. The motivating values behind this are that accuracy, completeness and currency are more important than readability, editability, and approachability.

Put another way, if this were a Word document written by hand, anybody would know how to open it and edit it. But such a document would likely not be up-to-date, because, while updating the policy would fix immediate user problems, updating the Word document would only have vague future benefits. That task could be easily skipped entirely in the near term, or administrators might easily jot down changes without supplying enough context or detail for their writings to be useful six months down the road. As it is, when administrators edit this policy, the source of the documentation is in the same file, on the same screenful of letters. If an administrator updates part of the policy, and the comment right above it becomes wrong, the juxtaposition makes that fact obvious, and more likely to be rectified. The administrator is also more likely to write the documentation in the first place, because no additional files or programs must be opened in order to begin writing it.

See §9 for more information about how to generate this finished document from its pieces, and how to manage and document changes you make to the policy herein.

# Chapter 2

# Compliance by IA control

This chapter summarizes measures taken to implement IA controls within this policy. Full details are to be found in the referenced sections.

So that future expansion will not disrupt the section numbering, there is a section for each IA control set forth in DoDI 8500.2, even though this Configuration Management for IT Systems Example Policy does not implement every IA control. Sections reserved for future expansion have shorter titles so you can easily skip them.

## 2.1 (COAS-1)

This section is reserved for future expansion.

## 2.2 (COAS-2)

This section is reserved for future expansion.

## 2.3 COBR-1: Protection of Backup and Restoration Assets

*From §6.1 (Database STIG compliance under PostgreSQL):*

Make sure that "DBMS files critical for DBMS recovery" are "stored on RAID or other high-availability storage devices," by specifying a RAID hard drive setup when procuring any server on which a PostgreSQL database will reside.

admins do COBR-1
admins do DG0114

*From §11.100.10 (STIG-required SSH configuration):*

Disallow root login over `ssh`: admins must use `su` (§11.101.16) or `sudo` after logging in as themselves.

auto: GEN001020
auto: GEN001100
auto: GEN001120
auto: OSX00165 M6
auto: OSX8-00-00565

## 2.4 CODB-1: Data Backup Procedures

*From §4.1 (Manual Mac compliance):*

Maintain "system recovery backups" for Macs as required by the STIG. admins do OSX00675 M6

*From §11.21.4 (unix2dos):*

This Configuration Management for IT Systems Example Policy comprises a great deal of what is needed to accomplish "recovery of a damaged or compromised [Mac] system in a timely basis." Automated backup of the policy and its dependencies as described in this section is therefore an important part of compliance with this requirement. auto: OSX00675 M6

Lock the fire-rated container which holds the contingency backups. admins do OSX00675 M6

## 2.5 (CODB-2)

This section is reserved for future expansion.

## 2.6 (CODB-3)

This section is reserved for future expansion.

## 2.7 (CODP-1)

This section is reserved for future expansion.

## 2.8 (CODP-2)

This section is reserved for future expansion.

## 2.9 (CODP-3)

This section is reserved for future expansion.

## 2.10 (COEB-1)

This section is reserved for future expansion.

## 2.11 (COEB-2)

This section is reserved for future expansion.

## 2.12 (COED-1)

This section is reserved for future expansion.

## 2.13 (COED-2)

This section is reserved for future expansion.

## 2.14 (COEF-1)

This section is reserved for future expansion.

## 2.15 (COEF-2)

This section is reserved for future expansion.

## 2.16 (COMS-1)

This section is reserved for future expansion.

## 2.17 (COMS-2)

This section is reserved for future expansion.

## 2.18 (COPS-1)

This section is reserved for future expansion.

## 2.19 (COPS-2)

This section is reserved for future expansion.

## 2.20 (COPS-3)

This section is reserved for future expansion.

## 2.21 (COSP-1)

This section is reserved for future expansion.

## 2.22 (COSP-2)

This section is reserved for future expansion.

## 2.23 COSW-1: Backup Copies of Critical Software

*From §11.21.4 (unix2dos):*

Back up this Configuration Management for IT Systems Example Policy, along with organization-specific critical software and documentation, monthly onto read-only media. auto: COSW-1 auto: DCHW-1

Store the contingency backup in a fire-rated container. admins do COSW-1 admins do DCHW-1

## 2.24 (COTR-1)

This section is reserved for future expansion.

## 2.25 (DCAR-1)

This section is reserved for future expansion.

## 2.26 (DCAS-1)

This section is reserved for future expansion.

## 2.27 (DCBP-1)

This section is reserved for future expansion.

## 2.28 (DCCB-1)

This section is reserved for future expansion.

## 2.29 (DCCB-2)

This section is reserved for future expansion.

## 2.30 DCCS-1: Configuration Specifications

As required, “[a] DoD reference document, such as a STIG or SRG, constitutes the primary source for security configuration or implementation guidance.” The policy (§11) configures AFSEO non-Windows hosts, and the procedures (§**??** and §7) guide administrators and users, according to these DISA-released documents:

- the UNIX SRG [6] (see in particular §3 and §**??**)
- the SPAN STIG [2] (see in particular §5 and §**??**)
- the Apache 2.2 Web Server and Web Site STIGs [4] [5] (see §**??**)
- the Generic Database STIG [3] (see in particular §6 and §**??**)

See the Bibliography (§15) for the exact versions of these documents used.

## 2.31 (DCCS-2)

This section is reserved for future expansion.

## 2.32 DCCT-1: Compliance Testing

“A comprehensive set of procedures is implemented that tests all patches, upgrades and new AIS applications prior to deployment,” as required. See §**??** for the procedures.

## 2.33 (DCDS-1)

This section is reserved for future expansion.

## 2.34 (DCFA-1)

This section is reserved for future expansion.

## 2.35 DCHW-1: Hardware Baseline

*From §11.21.4 (unix2dos):*

Back up this Configuration Management for IT Systems Example Policy, along with organization-specific critical software and documentation, monthly onto read-only media. auto: COSW-1 auto: DCHW-1

Store the contingency backup in a fire-rated container. admins do COSW-1 admins do DCHW-1

## 2.36 (DCID-1)

This section is reserved for future expansion.

## 2.37 (DCII-1)

This section is reserved for future expansion.

## 2.38 (DCIT-1)

This section is reserved for future expansion.

## 2.39 (DCMC-1)

This section is reserved for future expansion.

## 2.40 DCNR-1: Non-repudiation

*From §11.33.1 (RHEL 5 FIPS 140-2 guidance):*

Ensure that OpenSSH will operate in a FIPS-compliant fashion, by configuring the OpenSSL cryptographic library to run in FIPS 140-2 compliant mode.

auto: GEN005490
auto: GEN005495

*From §11.75.4 (Passwords on Macs):*

Use a FIPS 140-2 approved algorithm for hashing account passwords.

auto: GEN000590
auto: GEN000595

*From §11.100.3 (Set login banner):*

Configure the SSH server to use only FIPS 140-2 [14] approved ciphers.

auto: GEN005505 M6
auto: GEN005505

Configure the SSH server to use only FIPS 140-2 approved message authentication code (MAC) hash algorithms.

auto: GEN005507 M6
auto: GEN005507

Configure the SSH client to use only FIPS 140-2 approved ciphers.

auto: GEN005510 M6
auto: GEN005510

Configure the SSH client to use only FIPS 140-2 approved MAC hash algorithms.

auto: GEN005512 M6
auto: GEN005512

## 2.41 DCPA-1: Partitioning the Application

*From §6.1 (Database STIG compliance under PostgreSQL):*

To prevent "database tables from unrelated applications" from being "stored in the same database files" under PostgreSQL, ensure that for each "unrelated application" there is a separate database, using the `createdb` utility as appropriate.

DBAs do DCPA-1
DBAs do DG0113

## 2.42 (DCPB-1)

This section is reserved for future expansion.

## 2.43 (DCPD-1)

This section is reserved for future expansion.

## 2.44 DCPP-1: Ports, Protocols, and Services

*From §11.87.1 (Disable rsh, rlogin, and rexec):*

Make sure the telnet daemon is not running. auto: GEN003850 M6

Make sure the finger daemon is not running. auto: OSX8-00-00040
auto: GEN003860 M6
auto: OSX8-00-01115

*From §11.87.1 (Disable rsh, rlogin, and rexec under Mac OS X):*

Under RHEL, to ensure that rsh and rlogin are disabled, uninstall them. auto: GEN003820
auto: GEN003825
auto: GEN003830
auto: GEN003835
auto: GEN003840
auto: GEN003845

*From §11.100.3 (Set login banner):*

Configure the SSH server to reject SSH protocol version 1, which is no longer secure. auto: GEN005500
auto: OSX00175 M6

Configure the SSH client not to use SSH protocol version 1, which is no longer secure. auto: OSX8-00-00570
auto: OSX8-00-00575
auto: GEN005501

*From §11.100.14 (Enable useful SSH features):*

Remove the finger server. auto: GEN003860

*From §11.107.1 (Disable Telnet):*

Remove the Telnet server. auto: GEN003850

## 2.45 (DCPR-1)

This section is reserved for future expansion.

## 2.46 (DCSD-1)

This section is reserved for future expansion.

## 2.47 DCSL-1: System Library Management Controls

*From §11.5 (Turn off AFP server on Red Hat):*

Check for rootkits. The AIDE tool does this adequately for our needs. auto: GEN008380

*From §11.6 (Host-based intrusion detection with AIDE):*

Configure AIDE to create and monitor a baseline of database "software libraries, related applications and configuration files." auto: DCSL-1 auto: DG0050

Check for unauthorized changes to system files, including setuid files and setgid files, every week. auto: GEN000220 auto: GEN002400 auto: GEN002460

*From §11.10 (The at subsystem):*

Never run a group-writable or world-writable program with `at`. Never run a program using `at` which is in or anywhere under a world-writable directory (such as `/tmp`). Don't change the umask in an `at` job. admins do GEN003360 admins do GEN003380 admins do GEN003440 M6

*From §11.23.3 (Under Red Hat):*

Restrict access to the system `crontab` to only root. admins do GEN003440

Before writing or deploying a cron script, make sure it will not execute group- or world-writable programs, nor execute programs in or under world-writable directories. auto: GEN003040 auto: GEN003050 auto: GEN003080 admins do DCSL-1 admins do GEN003000 admins do GEN003020

*From §11.89.2 (The auth database):*

Ensure that "application software and configuration files" dependent on the database are owned by "the software installation account or the designated owner account," in the context of the AFSEO SBU system. auto: DCSL-1 auto: DG0019

*From §11.101.4 (Disable host-based authentication):*

Lock down permissions for "library files." auto: GEN001300 M6

## 2.48 (DCSP-1)

This section is reserved for future expansion.

## 2.49 (DCSQ-1)

This section is reserved for future expansion.

## 2.50 (DCSR-1)

This section is reserved for future expansion.

## 2.51 (DCSR-2)

This section is reserved for future expansion.

## 2.52 (DCSR-3)

This section is reserved for future expansion.

## 2.53 DCSS-1: System State Changes

*From §11.22.2 (The backup host):*

Ensure that "aborts are configured to ensure that the system remains in a secure state."

auto: DCSS-1

*From §11.27 (STIG-required digihub configuration):*

Ensure that "shutdowns" are "configured to ensure that the system remains in a secure state" by preventing an unauthenticated person at the console from rebooting the system.

auto: GEN000000-LNX00580
auto: DCSS-1

*From §11.51 (Kernel core dumping):*

Disable kernel core dumping to improve the security of the system during aborts: Kernel core dump files will contain sensitive data, and heretofore we have not needed to debug crashed kernels.

auto: GEN003510 M6
auto: OSX8-00-01105
auto: GEN003510
auto: DCSS-1
N/A: GEN003520
N/A: GEN003521
N/A: GEN003522
N/A: GEN003523

*From §11.94.5 (Set default umask):*

Control access to single-user mode, so that "system initialization" and "shutdown... are configured to ensure that the system remains in a secure state."

auto: DCSS-1

Under Mac OS X, single-user mode access is controlled by a boot password, which must be set from a utility which is run from the Mac OS X install disk. This cannot be automated.

admins do DCSS-1

## 2.54 (DCSS-2)

This section is reserved for future expansion.

## 2.55 DCSW-1: Software Baseline

*From §11.5 (Turn off AFP server on Red Hat):*

Install and configure the Advanced Intrusion Detection Environment (AIDE) host-based intrusion detection system (IDS) to check system files against a list of cryptographic hashes (a baseline) created at install time. (See §?? for baseline creation and update procedures.)

auto: GEN000140
auto: GEN006480
auto: GEN000140-2

For DBMSes included with RHEL, maintain the baseline for database software and configuration files along with that of the operating system files. (See also §11.86.1.)

auto: DCSW-1
auto: DG0021

*From §11.6 (Host-based intrusion detection with AIDE):*

Install the prescribed configuration for AIDE, causing it to baseline device files, extended access control lists (ACLs), and extended attributes, using FIPS 140-2 approved cryptographic hashing algorithms.

auto: GEN000140
auto: GEN006570
auto: GEN006571
auto: GEN006575

*From §11.40.6 (STIG-required configuration):*

Do not execute world-writable programs from your local initialization files. If you build programs, make sure they don't end up world-writable.

users do
GEN001940

*From §11.86 (Managing GPG keys in the RPM database):*

Use RPM's verify feature to cryptographically verify the integrity of installed software for DBMSes included with RHEL.

auto: DCSW-1
auto: DG0021

## 2.56 (EBBD-1)

This section is reserved for future expansion.

## 2.57 (EBBD-2)

This section is reserved for future expansion.

## 2.58 (EBBD-3)

This section is reserved for future expansion.

## 2.59 (EBCR-1)

This section is reserved for future expansion.

## 2.60 (EBPW-1)

This section is reserved for future expansion.

## 2.61 (EBRP-1)

This section is reserved for future expansion.

## 2.62 EBRU-1: Remote Access for User Functions

*From §11.87.1 (Disable rsh, rlogin, and rexec):*

Make sure the rsh daemon is not running.

auto: GEN003820 M6<br>
auto: OSX8-00-00050

Make sure the finger daemon is not running.

auto: GEN003860 M6<br>
auto: OSX8-00-01115

*From §11.87.1 (Disable rsh, rlogin, and rexec under Mac OS X):*

Under RHEL, to ensure that rsh and rlogin are disabled, uninstall them.

auto: GEN003820<br>
auto: GEN003825<br>
auto: GEN003830<br>
auto: GEN003835<br>
auto: GEN003840<br>
auto: GEN003845

## 2.63 (EBVC-1)

This section is reserved for future expansion.

## 2.64 (ECAD-1)

This section is reserved for future expansion.

## 2.65 ECAN-1: Access for Need-to-Know

*From §4.1 (Manual Mac compliance):*

Disable guest logon and guest access to shared folders on Macs.

admins do<br>
OSX00295 M6<br>
admins do<br>
OSX00300 M6

*From §11.11.2 (Turn down audio input levels):*

Activate audit logging; configure it in a compliant fashion; and protect and retain audit logs.

auto: ECAN-1<br>
auto: ECRR-1

## 2.66 (ECAR-1)

This section is reserved for future expansion.

## 2.67 ECAR-2: Audit Record Content—Sensitive Systems

*From §11.6 (Host-based intrusion detection with AIDE):*

Check for unauthorized changes to system files, including setuid files and setgid files, every week.

auto: GEN000220<br>
auto: GEN002400<br>
auto: GEN002460

*From §11.11.2 (Turn down audio input levels):*

The auditing rules installed in §11.12 fulfill Database STIG requirements.

auto: ECAR-2
auto: DG0140

*From §11.12.2 (File and directory permissions relating to auditing):*

Install the auditing software.

auto: GEN002660

Configure the auditing subsystem according to the requirements of the UNIX SRG.

auto: GEN002720
auto: GEN002740
auto: GEN002750
auto: GEN002751
auto: GEN002752
auto: GEN002753
auto: GEN002760
auto: GEN002800
auto: GEN002820
auto: GEN002825

*From §11.78.5 (Pattern for application roles and permissions):*

"Enable auditing on the database." Configure the database to log the messages required by the STIG, and to send those log messages out via the system log. Retention, periodic review, access restriction, and backup, then, are handled via the provisions for such requirements against the system log; see §11.55.1.

auto: ECAR-2
auto: ECRR-1
auto: ECCD-1
auto: ECTP-1
auto: ECTB-1

Log all attempts to modify data, if required by "application design requirements;" if not, only log attempts to modify the structure of the database.

auto: DG0029
auto: DG0030
auto: DG0031
auto: DG0032
auto: DG0176

Log all connection attempts, and every statement that results in a message with 'error' or greater urgency. This last includes "failed database object attempts," "attempts to access objects that do not exist," and "other activities that may produce unexpected failures."

auto: ECCD-1
auto: ECAR-2
auto: DG0031
auto: DG0145

Log the name of the acting user for each event. Date and time are taken care of by the system log. "Type of event" and "success or failure" are the text of the log message.

auto: ECAR-2
auto: DG0141
auto: DG0145
auto: ECAR-2
auto: DG0145

## 2.68 (ECAR-3)

This section is reserved for future expansion.

## 2.69 ECAT-1: Audit Trail, Monitoring, Analysis and Reporting

*From §11.5 (Turn off AFP server on Red Hat):*

Notify admins of possible intrusions via syslog. Remote logging ensures timely notification; for details, see §11.55.1.

auto: GEN006560

*From §11.6 (Host-based intrusion detection with AIDE):*

Install the prescribed configuration for AIDE, causing it to baseline device files, extended access control lists (ACLs), and extended attributes, using FIPS 140-2 approved cryptographic hashing algorithms.

auto: GEN000140
auto: GEN006570
auto: GEN006571
auto: GEN006575

*From §11.12.2 (File and directory permissions relating to auditing):*

Send an email to the administrator when disk space reserved for audit logs runs low. Mail for root is set up to go to the right places by §**??**.

auto: GEN002719
auto: GEN002730
auto: RHEL-06-000005

Configure the auditing subsystem according to the requirements of the UNIX

auto: GEN002720
auto: GEN002740
auto: GEN002750
auto: GEN002751
auto: GEN002752
auto: GEN002753
auto: GEN002760
auto: GEN002800
auto: GEN002820
auto: GEN002825

SRG.

*From §11.56.4 (Configuring a loghost):*
"[U]se a remote syslog server (loghost)," so that the remotely collected system log data "can be used as an authoritative log source in the event a system is compromised and its local logs are suspect," and so that it's easier to check logs every day and set up automated alerts.

auto: GEN005450

*From §11.66.13 (STIG-required network configuration under Mac OS X):*
Cause "martian packets" to be logged.

auto: GEN003611

*From §11.86 (Managing GPG keys in the RPM database):*
Use the RPM package manager's verify feature to cryptographically verify the integrity of installed system software monthly.

auto: GEN006565

*From §11.101.13 (System file permissions):*
"Verify system software periodically," including the ACLs of files and their extended attributes.

auto: GEN006565 M6
auto: GEN006570 M6
auto: GEN006571 M6

## 2.70 (ECAT-2)

This section is reserved for future expansion.

## 2.71 ECCD-1: Changes to Data

*From §4.1 (Manual Mac compliance):*
Turn off Screen Sharing, File Sharing, Printer Sharing, Web Sharing, Remote Login, Remote Management (Apple Remote Desktop), Remote Apple Events, and Xgrid Sharing on Macs.

*From §11.10 (The at subsystem):*
Never run a group-writable or world-writable program with `at`. Never run a program using `at` which is in or anywhere under a world-writable directory (such as `/tmp`). Don't change the umask in an `at` job.

*From §11.23 (Core dumps):*
Turn off core dumps because we do not need them.

*From §11.23.3 (Under Red Hat):*
Don't write a cron script that changes the umask.

*From §11.25.1 (Set system default printer):*
On hosts which do not need to print, disable CUPS entirely. This triv-

admins do OSX00475 M6
admins do OSX00480 M6
admins do OSX00485 M6
admins do OSX00490 M6
admins do OSX00495 M6
admins do OSX00500 M6
admins do OSX00505 M6
admins do OSX00510 M6
admins do GEN003360
admins do GEN003380
admins do GEN003440 M6
admins do GEN003440
auto: GEN003500
admins do GEN003220
auto: GEN003900

ially complies with this requirement not to “allow all hosts to use local print resources.”

*From §11.26 (Digihub: automatic action when media inserted):*
Disable automatic actions when blank DVDs are inserted.

auto: OSX00341 M6
auto: OSX8-00-00090

*From §11.40.6 (STIG-required configuration):*
Do not add an entry to your `PATH` which is not an absolute path. This prohibition includes `.`, the current directory.

users do GEN001900

*From §11.41.2 (User guidance about home directories):*
Remove `.rhosts` and `.shosts` files from home directories.
Remove `.netrc` files from home directories.

auto: GEN001980
auto: GEN002040
N/A: GEN002020
N/A: GEN002060

*From §11.41.3 (Quick-to-enforce home policies):*
Control ownership and permissions on files contained in home directories.

auto: GEN002000 M6
auto: OSX8-00-00600
auto: GEN002000

*From §11.69.4 (Turn off NFS server on Red Hat machines):*
Control ownership and permissions of the `exports` file.

auto: GEN001540 M6
auto: GEN001550 M6
auto: GEN001540
auto: GEN001550
auto: GEN001560

*From §11.70 (NIS (Network Information System)):*
Make sure there are no pluses in system authentication data files, causing possibly insecure NIS lookups.

auto: GEN005740
auto: GEN005750
auto: GEN005760
auto: GEN001980

*From §11.74.3 (Limit maximum logins):*
Make sure the `.rhosts` file is not supported in PAM.

auto: GEN002100

*From §11.78.5 (Pattern for application roles and permissions):*
“Enable auditing on the database.” Configure the database to log the messages required by the STIG, and to send those log messages out via the system log. Retention, periodic review, access restriction, and backup, then, are handled via the provisions for such requirements against the system log; see §11.55.1.

auto: ECAR-2
auto: ECRR-1
auto: ECCD-1
auto: ECTP-1
auto: ECTB-1
auto: DG0029
auto: DG0030
auto: DG0031
auto: DG0032
auto: DG0176

Log all attempts to modify data, if required by “application design requirements;” if not, only log attempts to modify the structure of the database.

auto: ECCD-1
auto: ECAR-2
auto: DG0031
auto: DG0145

*From §11.84.4 (Ensure only root has user id 0):*
Make sure the root user’s home directory is not `/`.
Secure ownership and permissions of root’s home directory.
Make sure that root’s `PATH`, `LD_LIBRARY_PATH`, and `LD_PRELOAD` environment variables are secure, and that no world-writable directories are on root’s `PATH`.

auto: GEN000900
auto: GEN000920
auto: GEN000940
auto: GEN000945
auto: GEN000950
auto: GEN000960

*From §11.93 (Serial port console support):*
Do not effect any policy which puts a relative path in the `PATH`, `LD_LIBRARY_PATH` or `LD_PRELOAD` environment variables.

admins do GEN001840
admins do GEN001845
admins do GEN001850

*From §11.94.4 (STIG-required shell configuration):*
Set the system default umask to `077`, so that by default files are only accessible by the user who created them. auto: GEN002560

*From §11.100.10 (STIG-required SSH configuration):*
Ignore per-user `.rhosts` and `.shosts` files. auto: GEN002040
Make sure host-based authentication is not used. auto: GEN002040

*From §11.101.1 (Device files):*
Check for system files and directories having "uneven access permissions." auto: GEN001140
auto: GEN001140 M6

*From §11.101.2 (Uneven access permissions):*
Check for files and directories with unknown owners. auto: GEN001160
auto: GEN001170
auto: GEN001160 M6

*From §11.101.3 ("Unowned" files):* auto: GEN001170 M6
Remove `hosts.equiv` and `shosts.equiv` files. auto: GEN002040

*From §11.101.6 (At the GDM login):*
Lock down permissions for manual page files. auto: GEN001280
auto: GEN001280 M6

*From §11.101.7 (Manual page file permissions):*
Make sure unprivileged users cannot remove devices. Device file permissions are "as configured by the vendor:" only "device files specifically intended to be world-writable" are world-writable. auto: GEN002280 M6

*From §11.101.8 (Miscellaneous STIG-required file permission policies):*
Do not deploy any run control script that contains a relative path or empty entry in a PATH variable setting. You should never need to change the `PATH` in a run control script anyway. Similarly, never set `LD_PRELOAD` and never put a relative or empty entry into the `LD_LIBRARY_PATH` used in a run control script. Never deploy a run control script that executes a world-writable program or script. Any run control script that runs a program or script stored on an NFS share should be documented in §3.4.

admins do GEN001600
admins do GEN001605
admins do GEN001610
admins do GEN001640

*From §11.101.14 (Force permissions specified by vendors):*
Find and warn administrators about world-writable directories without the sticky bit set. auto: GEN002500 M6
auto: OSX8-01120

*From §11.110.3 (STIG-required settings):*
Set the system default umask to `077`, so that by default files are only accessible by the user who created them. auto: GEN002560

*From §11.110.3 (STIG-required settings):*
Fix *unowned* files and directories, defined as those whose numerical owner UID or group-owner GID do not map to a known user or group. auto: GEN001160 M6
auto: GEN001170 M6

## 2.72 (ECCD-2)

This section is reserved for future expansion.

## 2.73 (ECCM-1)

This section is reserved for future expansion.

## 2.74 (ECCR-1)

This section is reserved for future expansion.

## 2.75 (ECCR-2)

This section is reserved for future expansion.

## 2.76 (ECCR-3)

This section is reserved for future expansion.

## 2.77 (ECCT-1)

This section is reserved for future expansion.

## 2.78 (ECCT-2)

This section is reserved for future expansion.

## 2.79 (ECDC-1)

This section is reserved for future expansion.

## 2.80 (ECIC-1)

This section is reserved for future expansion.

## 2.81 (ECID-1)

This section is reserved for future expansion.

## 2.82 (ECIM-1)

This section is reserved for future expansion.

## 2.83 (ECLC-1)

This section is reserved for future expansion.

## 2.84 ECLO-1: Logon

*From §11.74.2 (pam_limits):*
Configure the system to limit the maximum number of logins. auto: ECLO-1

*From §11.74.5 (securetty):*
Lock users out after three bad login attempts. auto: GEN000460

*From §11.75.4 (STIG-required password configuration):*
Set the maximum number of failed login attempts on the Mac. auto: OSX00050 M6

*From §11.78.5 (Pattern for application roles and permissions):*
Limit concurrent connections to the database. The vendor recommends 100 concurrent connections as a starting limit. auto: ECLO-1 auto: DG0134

*From §11.100.14 (Enable useful SSH features):*
Make the system delay at least 4 seconds following a failed login. auto: GEN000480

## 2.85 (ECLO-2)

This section is reserved for future expansion.

## 2.86 ECLP-1: Least Privilege

*From §6.1 (Database STIG compliance under PostgreSQL):*

Do not grant "DDL (Data Definition Language) and/or system configuration" privileges to non-privileged DBMS users. To obtain a "list of privileged role assignments" in an installation of PostgreSQL as included in RHEL, perform the following commands as root on the server in question: DBAs do ECLP-1 DBAs do DG0116

Do not use a privileged database account for non-administrative purposes. For each application in the database, create a per-application object owner user and/or per-application administrator user; use one of these, and not a DBA account, to create the objects necessary for the application and to maintain the DBAs do ECLP-1 DBAs do DG0004 DBAs do DG0124 application. Disable this account “when not performing installation or maintenance actions.”

Do not grant “privileges to restore database data, objects, or other configuration or features” to unauthorized DBMS accounts.

DBAs do ECLP-1
DBAs do DG0063

*From §11.6 (Host-based intrusion detection with AIDE):*

Use mode `0700` for the daily log rotation script, as required.

auto: GEN003100
auto: GEN003120
auto: GEN003140

*From §11.10.2 (Guidance for admins about the at subsystem):*

Control ownership and permissions of `at.deny`.

auto: GEN003480 M6

*From §11.10.3 (STIG-required at subsystem configuration for Mac OS X):*

Remove `at.deny`, in order to specify access by who is allowed, not by who is denied.

auto: GEN003252
auto: GEN003300
auto: GEN003480

Control contents and permissions of `at.allow`.

auto: GEN003490

Control permissions of “the ‘at’ directory.”

auto: GEN003280
auto: GEN003320

Remove extended ACL on `at.allow`.

auto: GEN003460
auto: GEN003470

Remove extended ACL on `at.deny`.

auto: GEN003340

Remove extended ACLs in “the ‘at’ directory.”

auto: GEN003400
auto: GEN003420
auto: GEN003430

*From §11.12.1 (Auditing under Mac OS X):*

Fix permissions of audit log files.

auto: GEN003245
auto: GEN003255
auto: GEN003410

*From §11.12.1 (Mac OS X audit log permissions):*

auto: GEN002680 M6

Ensure proper ownership and permissions on audit logs.

auto: GEN002690 M6
auto: GEN002700 M6

Ensure proper ownership and permissions on audit tool executables.

auto: OSX8-00-00205

Remove extended access control lists (ACLs) on audit tool executables.

auto: OSX8-00-00335
auto: OSX8-00-00350

*From §11.12.2 (File and directory permissions relating to auditing):*

Use mode `0700` for the auditd daily cron script, as required.

auto: GEN002680
auto: GEN002690
auto: GEN002700
auto: GEN002715
auto: GEN002716
auto: GEN002717

*From §11.17.2 (NFS mounts):*

Ensure the `nosuid` option is used when mounting an NFS filesystem.

auto: GEN002718 M6
auto: GEN002718

Ensure the `nosuid` option is used when mounting an NFS filesystem.

auto: GEN003100
auto: GEN003120
auto: GEN003140

*From §11.17.2 (NFS mounts):*

Ensure the `nosuid` option is used when mounting an NFS filesystem.

auto: GEN002420
auto: GEN005900
auto: GEN002420
auto: GEN005900

*From §11.17.3 (Adding an automount entry under Mavericks):*

Ensure the `nosuid` option is used when mounting an NFS filesystem.

auto: GEN002420
auto: GEN005900
auto: GEN002420
auto: GEN005900

*From §11.17.7 (NFS mounts in subdirectories):*

Ensure the `nosuid` option is used when mounting an NFS filesystem.

auto: GEN002420
auto: GEN005900

*From §11.23.2 (STIG-required core dump configuration):*

Control ownership and permissions for core-dump-related files written by

auto: GEN003501
auto: GEN003502
auto: GEN003503
auto: GEN003504

the Automated Bug Reporting Tool (ABRT).

Remove extended ACLs on ABRT directories.
auto: GEN003505

*From §11.23.3 (Under Red Hat):*

Make sure only root can edit the `cron.allow` file.
auto: GEN003250

Make sure only root can edit the `cron.deny` file.
auto: GEN003270 M6

Restrict access to the system `crontab` to only root.
auto: GEN003270

Control ownership and permissions of the “at” directory, which under Mac OS X is the same as the “cron” directory.
auto: GEN003040
auto: GEN003050
auto: GEN003080

Under RHEL, restrict access to directories used by `run-parts`, which contain scripts to be run periodically, to only root. Also restrict access to the files in these directories.
auto: GEN003400 M6
auto: GEN003420 M6
auto: GEN003100
auto: GEN003120
auto: GEN003140

Remove extended ACLs on `cron.allow`. Remove extended ACLs on `cron.allow`.
auto: GEN003080-2

Remove extended ACLs on `crontab`.
auto: GEN002990 M6

Remove extended ACLs on directories used by `run-parts`.
auto: GEN002990

Remove extended ACLs on `cron.deny`.
auto: GEN003245

Under RHEL, control usage of the `cron` utility.
auto: GEN003090

Under RHEL, remove the `cron.deny` file if it exists.
auto: GEN003110
auto: GEN003210
auto: GEN002960
auto: GEN002980

*From §11.25.1 (Set system default printer):*

Remove CUPS and the “hosts.lpd (or equivalent) file,” which in the case of CUPS is `/etc/cups/cupsd.conf`. This trivially prevents “unauthorized modifications” or “unauthorized remote access.”
auto: GEN003060
auto: GEN003240
auto: GEN003200
auto: GEN003260
auto: GEN003270

*From §11.25.3 (Define a printer):*

Control ownership and permissions of the “hosts.lpd (or equivalent) file,” in our case `cupsd.conf`.
auto: GEN003920
auto: GEN003930
auto: GEN003940
auto: GEN003950
auto: GEN003920
auto: GEN003930
auto: GEN003940

Remove extended ACLs on the same file.
auto: GEN003950

*From §11.40.5 (Enable serial console):*

Make sure the configuration file `/boot/grub/menu.lst` is owned by root, group-owned by root, has permissions `0600`, and has no extended ACL.
admins do GEN008720
admins do GEN008740

*From §11.41.2 (User guidance about home directories):*

Secure home directories.
admins do GEN008760
admins do GEN008780

Secure local initialization files.
auto: GEN001480

Remove extended ACLs for local initialization files.
auto: GEN001500
auto: GEN001520

*From §11.41.3 (Quick-to-enforce home policies):*
auto: GEN001860 M6

Control ownership and permissions on files contained in home directories.
auto: GEN001860
auto: GEN001870
auto: GEN001880

Remove extended ACLs on home directories, and all files and directories therein.
auto: GEN001890
auto: GEN001540 M6
auto: GEN001550 M6

*From §11.53 (LDAP):*
auto: GEN001540
auto: GEN001550
auto: GEN001560

Control ownership and permissions of `ldap.conf`.
auto: GEN001490 M6
auto: GEN001570 M6

Remove extended ACLs on `ldap.conf`.
auto: GEN001490
auto: GEN001570
auto: GEN008060 M6
auto: GEN008080 M6
auto: GEN008100 M6
auto: GEN008060
auto: GEN008080
auto: GEN008100
auto: GEN008120 M6

*From §11.56.1 (Backing up logs using NFS):*

Control ownership and permissions of the `rsyslog` configuration. auto: GEN005390
auto: GEN005400
Remove extended ACLs on the `rsyslog` configuration.
auto: GEN005420
auto: GEN005395

*From §11.56.5 (Sending log messages to a loghost):*

Secure `cron` logs. Secure SMTP logs. auto: GEN003180
auto: GEN004500
Remove extended ACLs on system log files (including SMTP and `cron` logs).
auto: GEN001270
auto: GEN003190

*From §11.56.7 (Log rules for Macs):* auto: GEN004510

auto: GEN001270 M6
Control ownership and permissions of the `syslog.conf` file.
auto: OSX8-00-00825
Remove extended ACLs from the `syslog.conf` file.
auto: GEN005400 M6
auto: GEN005420 M6

*From §11.66.11 (Non-routers):* auto: GEN005395 M6

Control ownership and permissions of the `services` file. auto: GEN003760 M6
auto: GEN003770 M6
Remove extended ACLs on the `services` file.
auto: GEN003780 M6
auto: GEN003760
auto: GEN003770

*From §11.66.13 (STIG-required network configuration under Mac OS X):*

auto: GEN003780
auto: GEN003790
auto: GEN000000-LNX00480
auto: GEN000000-LNX00500
auto: GEN000000-LNX00520

*From §11.67 (Network tools):* auto: GEN000000-LNX00530

Make the `traceroute` utility executable only by root. auto: GEN003960 M6
auto: GEN003980 M6
Remove extended ACLs on the `traceroute` executable.
auto: GEN004000 M6
auto: GEN003960
auto: GEN003980

*From §11.67.2 (Remove network analysis tools):*

auto: GEN004000
Make the `traceroute` utility executable only by root.
auto: GEN004010 M6
Remove extended ACLs on the `traceroute` executable.
auto: GEN004010
auto: GEN003960 M6
auto: GEN003980 M6

*From §11.68 (NetworkManager):*

auto: GEN004000 M6

Don't let users configure network interfaces: require authentication of an administrator to do this.

auto: GEN003960
auto: GEN003980
auto: GEN004000
auto: GEN004010 M6

*From §11.69.4 (Turn off NFS server on Red Hat machines):* auto: GEN004010

Control ownership and permissions of the `exports` file. auto: GEN003581
auto: GEN005740
Remove extended ACLs on the `exports` file.
auto: GEN005750
auto: GEN005760

*From §11.71 (NTP):* auto: GEN005770

Control ownership and permissions of the `ntp.conf` file. auto: GEN000250
auto: GEN000251
Remove extended ACLs on the `ntp.conf` file.
auto: GEN000252
auto: GEN000253

*From §11.75.1 (Admin guidance about passwords):*

Disable group passwords. auto: GEN000000-LNX001476

*From §11.75.2 (Remove passwords from gshadow):*

Make sure the passwd file does not contain password hashes. auto: GEN001470
Make sure the group file does not contain password hashes. auto: GEN001475

*From §11.78.3 (One-time PostgreSQL initialization):*

Ensure that “the DBMS software installation account” (we take this to mean `postgres`, because while that user does not install the DBMS, it owns the files in which the DBMS data is stored) “is only used when performing software installation and upgrades or other DBMS maintenance,” and not for “DBA activities,” by creating a separate user for automatically enforcing policies inside the DBMS.

auto: ECLP-1
auto: DG0042

*From §11.78.4 (Administering PostgreSQL using Puppet):*

Grant database administrative privileges to database administrators using DBMS roles.

auto: ECLP-1
auto: ECPA-1
auto: DG0116
auto: DG0117

A database administrator `fnord`, to whom the `dba` role below has been granted, must `SET ROLE dba` before doing any database administration. Such a user should `RESET ROLE` when done with the database administration.

DBAs do ECLP-1
DBAs do DG0124

Administrators must not use the `postgres` user to do anything with the database: each, being provided with his own database user, must use that instead.

admins do ECLP-1
admins do DG0042

Avoid granting “excessive or unauthorized” privileges to DBAs, by preventing them from being superusers in the database. “Although DBAs may assign themselves privileges,” that action is logged when it happens, and privileges are reported monthly. See §11.78.6 for details.

auto: ECLP-1
auto: DG0085

*From §11.78.5 (Pattern for application roles and permissions):*

Provide for “monthly... review of privilege assignments,” including DBA roles, within the PostgreSQL database by causing a report of roles and privileges to be sent to the administrators for review.

auto: ECLP-1
auto: ECPA-1
auto: DG0080
auto: DG0086
auto: DG0116
auto: DG0118

*From §11.83.1 (Use System Security Services (SSS)):*

Do not run a web browser under an administrative account, “except as needed for local service administration.”

admins do
GEN004220

*From §11.84.1 (Admin guidance regarding the root user):*

Control ownership and permissions on the `securetty` file.

auto: GEN000000-LNX00620
auto: GEN000000-LNX00640
auto: GEN000000-LNX00660

*From §11.84.4 (Ensure only root has user id 0):*

Ensure that only root has user id 0.

auto: GEN000880 M6
auto: OSX8-00-01065

*From §11.84.4 (Ensure only root has user id 0):*

Make sure root is the only user with a user id of 0.

auto: GEN000880

*From §11.84.4 (Ensure only root has user id 0):*

Remove extended ACLs from root’s home directory.

auto: GEN000930

*From §11.88.2 (STIG-required Samba configuration):*

Control ownership and permissions of `smb.conf`.

auto: GEN006100 M6
auto: GEN006140 M6

Remove extended ACLs on `smb.conf`.

auto: GEN006150 M6

*From §11.88.3 (STIG-required Samba configuration under Mac OS X):*

Control ownership and permissions of `smb.conf`. auto: GEN006100

Remove extended ACLs on `smb.conf`. auto: GEN006120 auto: GEN006140

Control ownership and permissions of `smbpasswd`. auto: GEN006150

Remove extended ACLs on `smbpasswd`. auto: GEN006160 auto: GEN006180 auto: GEN006200 auto: GEN006210

*From §11.89.1 (Unimplemented Apache STIG requirements):*

Prevent the misuse of DBA accounts for non-administrative purposes by creating an object owner user. auto: ECLP-1 auto: DG0124

Disable the application object owner user "when not performing installation or maintenance actions." auto: ECLP-1 auto: DG0004

*From §11.92.1 (Unimplemented Apache STIG requirements):*

Prevent the misuse of DBA accounts for non-administrative purposes by creating an object owner user. auto: ECLP-1 auto: DG0124

Disable the application object owner user "when not performing installation or maintenance actions." auto: ECLP-1 auto: DG0004

*From §11.94.2 (Env modules under RHEL):*

Make sure that no one can influence the environment variables set when the shell starts, except for root. auto: GEN001720 auto: GEN001740 auto: GEN001760

auto: GEN001720 M6 auto: GEN001740 M6 auto: GEN001760 M6

*From §11.94.3 (profile.d permissions):* auto: GEN001730

Control ownership and permissions of shell executables. auto: GEN002200 M6

Remove extended ACLs on shell executables. auto: GEN002220 M6 auto: GEN002200 auto: GEN002210

*From §11.96 (Smartcards):* auto: GEN002220

Control ownership of the SMTP log. (Permissions and ACLs are controlled by §11.56.6.) auto: GEN002230 M6 auto: GEN002230

Do not add any entries to the aliases file which execute programs. auto: GEN004480 admins do GEN004400 admins do GEN004410

*From §11.97.5 (SMTP smarthosts):*

Control ownership and permissions of the `aliases` file. admins do GEN004420

Remove extended ACLs on the `aliases` file. admins do GEN004430

*From §11.100.1 (Limit SSH connections by host IP):* auto: GEN004360

Restrict login via SSH to members of certain groups. auto: GEN004370 auto: GEN004380 auto: GEN004390

*From §11.100.10 (STIG-required SSH configuration):* auto: GEN005521

Cause the SSH server to ignore any user-specific files (*e.g.*, `known_hosts`, `authorized_keys`) that are not under the strict control of that user. auto: GEN005536

Use OpenSSH's privilege separation feature for better security. auto: GEN005537

Restrict write permissions on the public SSH host keys. auto: GEN005522

Restrict reading and writing permissions on the private SSH host keys. auto: GEN005523

*From §11.101.4 (Disable host-based authentication):*

Remove any extended ACLs from library files.

auto: GEN001310 M6
auto: GEN001310

*From §11.101.6 (At the GDM login):*

Remove any extended ACLs from manual page files.

auto: GEN001290
auto: GEN001290 M6

*From §11.101.7 (Manual page file permissions):*

Control ownership and permissions of `resolv.conf`.

auto: GEN001362 M6
auto: GEN001363 M6

Remove extended ACLs on `resolv.conf`.

auto: GEN001364 M6

Control ownership and permissions of the `hosts` file.

auto: GEN001362
auto: GEN001363

Remove extended ACLs on the `hosts` file.

auto: GEN001364

Control ownership and permissions of `nsswitch.conf`.

auto: GEN001365 M6

Remove extended ACLs on `nsswitch.conf`.

auto: GEN001365

Control ownership and permissions of the `passwd` file.

auto: GEN001366 M6
auto: GEN001367 M6

Remove extended ACLs on the `passwd` file.

auto: GEN001368 M6

Control ownership and permissions of the `group` file.

auto: GEN001366
auto: GEN001367

Remove extended ACLs on the `group` file.

auto: GEN001368

Control ownership and permissions of the `shadow` file.

auto: GEN001369 M6

Remove extended ACLs on the `shadow` file.

auto: GEN001369

Remove extended ACLs on sound device files.

auto: GEN001371
auto: GEN001372

Make sure unprivileged users cannot remove devices. Device file permissions are "as configured by the vendor:" only "device files specifically intended to be world-writable" are world-writable.

auto: GEN001373
auto: GEN001374
auto: GEN001378 M6
auto: GEN001379 M6
auto: GEN001380 M6
auto: GEN001378
auto: GEN001379
auto: GEN001380
auto: GEN001390 M6
auto: GEN001390
auto: GEN001391 M6

*From §11.101.8 (Miscellaneous STIG-required file permission policies):*

auto: GEN001392 M6

Restrict permissions on the run control scripts.

auto: GEN001393 M6

Restrict ownership on "system start-up files."

auto: GEN001391
auto: GEN001392

Remove extended ACLs on run control scripts.

auto: GEN001393
auto: GEN001394 M6

*From §11.101.10 (Admin guidance about run control scripts):*

auto: GEN001394

Control ownership and permissions of skeleton files.

auto: GEN001400
auto: GEN001410

Remove extended ACLs from skeleton files.

auto: GEN001420
auto: GEN001430
auto: GEN002330

*From §11.101.12 (Startup file permissions):*

auto: GEN002280 M6

Make sure all "network services daemon files" are not group- or world-writable.

auto: GEN000000-LNX001431
auto: GEN000000-LNX001432

Make sure all "system command files" are not group- or world-writable.

auto: GEN000000-LNX001433
auto: GEN000000-LNX001434

Make sure all "system files, programs, and directories" are owned by "a system account."

auto: GEN000000-LNX00400
auto: GEN000000-LNX00420
auto: GEN000000-LNX00440

Make sure all "system files, programs, and directories" are group-owned by "a system group."

auto: GEN000000-LNX00450
auto: GEN001580 M6

Remove extended ACLs on "network services daemon files."

auto: GEN001580
auto: GEN001660
auto: GEN001680
auto: GEN001590 M6
auto: GEN001590
auto: GEN001800
auto: GEN001820
auto: GEN001830
auto: GEN001810
auto: GEN001180
auto: GEN001180 M6

Remove extended ACLs on “system command files.” auto: GEN001210 M6

*From §11.101.13 (System file permissions):*
To make sure all “system start-up files” are properly owned and group-owned on the Mac, run the disk utility to “reset the ownership to the original installation settings.”

auto: GEN001660 M6
auto: GEN001680 M6

*From §11.101.14 (Force permissions specified by vendors):*
Find and warn administrators about public directories not owned by root. auto: GEN002520 M6
auto: OSX8-00-01110

*From §11.101.15 (World-writable directories):*
Control ownership and permissions of the `xinetd` configuration.
Remove extended ACLs on `xinetd` configuration.

auto: GEN003720
auto: GEN003730
auto: GEN003740
auto: GEN003750
auto: GEN003745
auto: GEN003755

## 2.87 ECML-1: Marking and Labeling

*From §11.89.3 (Server deployment):*
Configure Trac instances on the SBU server to show a banner with a security label at the top of each page. auto: ECML-1

*From §11.92.3 (Server deployment):*
Configure Trac instances on the SBU server to show a banner with a security label at the top of each page. auto: ECML-1

## 2.88 (ECMT-1)

This section is reserved for future expansion.

## 2.89 (ECMT-2)

This section is reserved for future expansion.

## 2.90 (ECND-1)

This section is reserved for future expansion.

## 2.91 (ECND-2)

This section is reserved for future expansion.

## 2.92 (ECNK-1)

This section is reserved for future expansion.

## 2.93 (ECNK-2)

This section is reserved for future expansion.

## 2.94 ECPA-1: Privileged Account Control

*From §11.5 (Turn off AFP server on Red Hat):*

Document setuid and setgid files, by including them in the baseline of system files.

auto: GEN002380
auto: GEN002440

*From §11.10.3 (STIG-required at subsystem configuration for Mac OS X):*

Control contents and permissions of `at.allow`.

auto: GEN003280
auto: GEN003320
auto: GEN003460
auto: GEN003470
auto: GEN003340

*From §11.17.2 (NFS mounts):*

Ensure the `nosuid` option is used when mounting an NFS filesystem.

auto: GEN002420
auto: GEN005900

Ensure the `nosuid` option is used when mounting an NFS filesystem.

auto: GEN002420
auto: GEN005900

*From §11.17.2 (NFS mounts):*

Ensure the `nosuid` option is used when mounting an NFS filesystem.

auto: GEN002420
auto: GEN005900

*From §11.17.3 (Adding an automount entry under Mavericks):*

Ensure the `nosuid` option is used when mounting an NFS filesystem.

auto: GEN002420
auto: GEN005900

*From §11.17.7 (NFS mounts in subdirectories):*

Ensure the `nosuid` option is used when mounting an NFS filesystem.

auto: GEN002420
auto: GEN005900

*From §11.23.3 (Under Red Hat):*

Under RHEL, control usage of the `cron` utility.

auto: GEN002960
auto: GEN002980
auto: GEN003060
auto: GEN003240

*From §11.78.4 (Administering PostgreSQL using Puppet):*

Grant database administrative privileges to database administrators using DBMS roles.

auto: ECLP-1
auto: ECPA-1
auto: DG0116
auto: DG0117

Grant administrative privileges solely via roles.

auto: ECPA-1
auto: DG0117

*From §11.78.5 (Pattern for application roles and permissions):*

Provide for “monthly... review of privilege assignments,” including DBA roles, within the PostgreSQL database by causing a report of roles and privileges to be sent to the administrators for review.

auto: ECLP-1
auto: ECPA-1
auto: DG0080
auto: DG0086
auto: DG0116
auto: DG0118

*From §11.83.1 (Use System Security Services (SSS)):*

Never log in as root, except for "emergency maintenance, the use of single-user mode for maintenance, and situations where individual administrator accounts are not available." admins do GEN001020

*From §11.84.1 (Admin guidance regarding the root user):*
Make sure root can only log in from the console. auto: GEN000980 auto: GEN001020

*From §11.91.1 (Require authentication to exit screensaver):*
Disable administrative accounts from unlocking other users' screens. auto: OSX00200 M6 auto: OSX8-00-00935

*From §11.100.10 (STIG-required SSH configuration):*
Disallow root login over `ssh`: admins must use `su` (§11.101.16) or `sudo` after logging in as themselves. auto: GEN001020 auto: GEN001100 auto: GEN001120 auto: OSX00165 M6 auto: OSX8-00-00565

## 2.95 (ECPC-1)

This section is reserved for future expansion.

## 2.96 (ECPC-2)

This section is reserved for future expansion.

## 2.97 ECRC-1: Resource Control

*From §11.105.1 (Encrypt swap on Macs):*
"Use secure virtual memory," or in other words, make Macs encrypt their swap space. auto: OSX00440 M6

## 2.98 ECRG-1: Audit Reduction and Report Generation

*From §11.12.2 (File and directory permissions relating to auditing):*

admins do ECRG-1

## 2.99 ECRR-1: Audit Record Retention

*From §11.11.2 (Turn down audio input levels):*
Activate audit logging; configure it in a compliant fashion; and protect and retain audit logs. auto: ECAN-1 auto: ECRR-1

*From §11.12.1 (Mac OS X audit log permissions):*

Let only admins access audit data.

auto: ECRR-1

*From §11.12.2 (File and directory permissions relating to auditing):*

"[E]nsure that audit logs that have reached maximum length are not overwritten," by suspending the system if space for audit logs runs out or disk errors prevent the writing of audit logs.

auto: ECRR-1

*From §11.56 (Logging):*

Back up audit logs and other logs to archival media. Retain them for one year, or five years for systems containing sources and methods intelligence (SAMI).

auto: ECRR-1

*From §11.78.5 (Pattern for application roles and permissions):*

"Enable auditing on the database." Configure the database to log the messages required by the STIG, and to send those log messages out via the system log. Retention, periodic review, access restriction, and backup, then, are handled via the provisions for such requirements against the system log; see §11.55.1.

auto: ECAR-2
auto: ECRR-1
auto: ECCD-1
auto: ECTP-1
auto: ECTB-1
auto: DG0029
auto: DG0030
auto: DG0031
auto: DG0032
auto: DG0176

## 2.100 ECSC-1: Security Configuration Compliance

*From §4.1 (Manual Mac compliance):*

Do not install unnecessary packages on a Mac.

Disable guest logon and guest access to shared folders on Macs.

Make Macs require administrator authentication to unlock each System Preference pane.

admins do OSX00010 M6
admins do OSX8-00-01165
admins do OSX00295 M6
admins do OSX00300 M6
admins do OSX00430 M6

*From §11.6 (Host-based intrusion detection with AIDE):*

Install the prescribed configuration for AIDE, causing it to baseline device files, extended access control lists (ACLs), and extended attributes, using FIPS 140-2 approved cryptographic hashing algorithms.

auto: GEN000140
auto: GEN006570
auto: GEN006571
auto: GEN006575

*From §11.11.1 (Disable audio):*

Disable audio support where necessary to "protect the organization's privacy."

auto: OSX00070 M6
auto: OSX8-00-01225

*From §11.12.2 (File and directory permissions relating to auditing):*

Rotate audit logs daily. auto: GEN002860

Rotate audit log files based on time, not their size. auto: GEN002860

*From §11.16.1 (Disable automatic logout):*

Disable “automatic logout due to inactivity” on Macs. auto: OSX00435 M6 auto: OSX8-00-01085

*From §11.16.1 (Disable automatic logout on Macs):*
“Automated file system mounting tools must not be enabled unless needed,” because they “may provide unprivileged users with the ability to access local media and network shares.” This automount configuration does not enable access to local media, and constricts network share access to filers designated for the purpose of serving unprivileged users. auto: GEN008440

*From §11.17.2 (NFS mounts):*
Ensure the `nodev` option is used when mounting an NFS filesystem. auto: GEN002430
Ensure the `nodev` option is used when mounting an NFS filesystem. auto: GEN002430

*From §11.17.2 (NFS mounts):*
Ensure the `nodev` option is used when mounting an NFS filesystem. auto: GEN002430

*From §11.17.3 (Adding an automount entry under Mavericks):*
Ensure the `nodev` option is used when mounting an NFS filesystem. auto: GEN002430

*From §11.17.7 (NFS mounts in subdirectories):*
Ensure the `nodev` option is used when mounting an NFS filesystem. auto: GEN002430

*From §11.19 (Cameras):*
Disable cameras where necessary to “protect the organization’s privacy.” auto: OSX00075 M6

*From §11.26 (Digihub: automatic action when media inserted):*
Disable automatic actions when blank CDs are inserted. auto: OSX00340 M6 auto: OSX8-00-00085
Disable automatic actions when picture CDs are inserted. auto: OSX00350 M6 auto: OSX8-00-00100
Disable automatic actions when video DVDs are inserted. auto: OSX00355 M6 auto: OSX8-00-00105

*From §11.27 (STIG-required digihub configuration):*
Ensure that “shutdowns” are “configured to ensure that the system remains in a secure state” by preventing an unauthenticated person at the console from rebooting the system. auto: GEN000000-LNX00580 auto: DCSS-1

*From §11.34 (File Transfer Protocol (FTP)):*
Remove FTP server software wherever possible. auto: GEN004800 auto: GEN004820 auto: GEN004840

*From §11.36.3 (STIG-required configuration):*
Set the right X server options (`-s` [screensaver timeout], `-audit` [audit level], and `-auth` [authorization record file], which “gdm always automatically uses”), and don’t set the wrong ones (`-ac` [disable host-based access control], `-core` [dump core on fatal errors], and `-nolock` [unknown, not in man page]). (The `-br` option merely makes the screen black by default when the server starts up, instead of the gray weave pattern.) auto: GEN000000-LNX00360 auto: GEN000000-LNX00380

*From §11.40.5 (Enable serial console):*

Turn on auditing in time to audit the actions of startup scripts. auto: GEN000000-LNX00720

*From §11.40.6 (STIG-required configuration):*

Administrators, "educate users about the danger of having terminal messaging set on." admins do GEN001960

Do not add an entry to your `LD_LIBRARY_PATH` which is not an absolute path. users do GEN001901

Do not set the `LD_PRELOAD` environment variable. users do GEN001902

Do not place the command `mesg y` in your startup files. users do GEN001960

*From §11.41.2 (User guidance about home directories):*

Prevent use of the `.forward` file by removing it. auto: GEN004580 M6<br>auto: OSX8-00-01040<br>auto: GEN004580

*From §11.45.1 (Under the Mac OS):*

### Under Red Hat

Disable Firewire "unless needed." We do not need it. auto: GEN008500

*From §11.46 (Infrared):*

Disable infrared support "to prevent unauthorized users from controlling a computer through the infrared receiver." auto: OSX00090 M6<br>auto: OSX8-00-00075

*From §11.46.1 (Disable infrared under Mac OS X):*

Employ a local firewall for IPv6, using `ip6tables`. auto: GEN008520

Configure the local firewall to reject all source-routed IPv6 packets, even those generated locally. auto: GEN003605<br>auto: GEN003606

Configure the local firewall to reject all IPv6 packets by default, allowing only by exception. auto: GEN008540

Configure the local firewall to reject ICMPv6 timestamp requests, including those sent to a broadcast address. auto: GEN003602<br>auto: GEN003604

*From §11.47 (ip6tables):*

Employ a local firewall, using `iptables`. auto: GEN008520

Configure the local firewall to reject all packets by default, allowing only by exception. auto: GEN008540

Configure the local firewall to reject ICMP timestamp requests, including those sent to a broadcast address. auto: GEN003602<br>auto: GEN003604

*From §11.49 (iTunes):*

Disable iTunes Store and other network features of iTunes on Macs. auto: OSX00530 M6<br>auto: OSX8-00-01140<br>auto: OSX8-00-01150<br>auto: OSX8-00-01155

*From §11.51 (Kernel core dumping):*

Disable kernel core dumping to improve the security of the system during auto: GEN003510 M6<br>auto: OSX8-00-01105<br>auto: GEN003510<br>auto: DCSS-1<br>N/A: GEN003520<br>N/A: GEN003521<br>N/A: GEN003522<br>N/A: GEN003523 aborts: Kernel core dump files will contain sensitive data, and heretofore we have not needed to debug crashed kernels.

*From §11.56.3 (Configuring remote logging clients):*

The "site-defined procedure" for setting up and documenting a loghost is this: admins do GEN005460

RHEL5 does not receive syslog messages by default (see `/etc/sysconfig/syslog`). RHEL5: GEN005480 RHEL6 does not receive syslog messages by default (see `/etc/rsyslog.conf`). RHEL6: GEN005480 To prevent inadvertent disclosure of sensitive information, do not configure any host to listen for log messages over the network by any other means than the above procedure. admins do GEN005480

*From §11.56.5 (Sending log messages to a loghost):*

Do not cause unencrypted log traffic to cross enclave boundaries. admins do GEN005440

*From §11.57 (Login window):*

Configure the Mac login window to show username and password prompts, not a "list of local user names available for logon." auto: OSX00310 M6

*From §11.65.1 (Prerequisites for wrapping 32-bit Mozilla plugins):*

Don't configure any IP tunnels. admins do GEN007820

*From §11.66.3 (Disable Bluetooth):*

Disable and/or uninstall Bluetooth protocol on Macs. auto: OSX00065 M6

*From §11.66.3 (Disable Bluetooth under Mac OS X):*

auto: OSX8-00-00060
auto: OSX8-00-00065
auto: OSX8-00-00080
auto: GEN007660

Disable and/or uninstall Bluetooth protocols. (Notably, this requirement does not say, "unless needed.")

*From §11.66.3 (Turn off the IKE daemon on Macs):*

Remove routing protocol daemons from non-routing systems. auto: GEN005590

*From §11.66.4 (Interfaces):*

Turn off IPv4 forwarding for non-router Red Hat hosts. auto: GEN005600

Turn off IPv4 forwarding for non-router Macs. auto: GEN005600 M6 auto: OSX8-00-01205

*From §11.66.4 (IPv4 non-routers):*

Turn on IPv4 forwarding for Red Hat hosts designated as routers. auto: GEN005600

Turn on IPv4 forwarding for Macs designated as routers. auto: GEN005600 M6 auto: OSX8-00-01205

*From §11.66.4 (IPv4 routers):*

"The IPv6 protocol handler must not be bound to the network stack unless needed," and "must be prevented from dynamic loading unless needed." Hosts which include this class need IPv6. auto: GEN007700 auto: GEN007720

*From §11.66.5 (Turn off IPv6 under Mac OS X):*

Unbind the IPv6 protocol from all network interfaces at boot time. auto: GEN007700 auto: GEN007720

*From §11.66.5 (Turn off IPv6 under RHEL):*

Disable 6to4. auto: GEN007780

*From §11.66.5 (Disable 6to4):*

Remove IPv6 routing protocol daemons from non-routing systems. auto: GEN005590

Turn off IPv6 forwarding for non-routers. auto: GEN005610

*From §11.66.5 (IPv6 non-routers):*

Do not configure network bridging. auto: GEN003619

*From §11.66.6 (Avoid Ethernet bridging):*

Disable the Datagram Congestion Control Protocol (DCCP) "unless required." We do not need it. auto: GEN007080

*From §11.66.8 (Don't send ICMP echo replies):*

Disable and/or uninstall the Reliable Datagram Sockets (RDS) protocol "unless required." auto: GEN007480

*From §11.66.9 (Disable RDS):*

Disable the Stream Control Transmission Protocol (SCTP) "unless required." We do not need it. auto: GEN007020

*From §11.66.12 (Platform-specific implementations of compliance):*

Configure the system to block ICMP timestamp requests. auto: GEN003602 M6 auto: OSX8-00-01220

Configure the system to ignore ICMP pings sent to a broadcast address. auto: GEN003603 M6 auto: OSX8-00-01190

Configure the system to "prevent local applications from generating source-routed packets." auto: GEN003606 M6 auto: OSX8-00-01215

Configure the system to "not accept source-routed IPv4 packets." auto: GEN003607 M6 auto: OSX8-00-01195

Configure the system to "ignore ICMPv4 redirect messages." auto: GEN003609 M6 auto: OSX8-00-01200

Prevent the system from sending ICMPv4 redirect messages. auto: GEN003610 M6 auto: OSX8-00-01210

*From §11.66.13 (STIG-required network configuration under Mac OS X):*

Set the TCP backlog queue size appropriately. auto: GEN003601

Configure the system to ignore ICMP pings sent to a broadcast address. auto: GEN003603

Configure the system to ignore source-routed IPv4 packets. auto: GEN003607

Disable Proxy ARP. auto: GEN003608

Cause the system to ignore ICMPv4 redirect messages. auto: GEN003609

Prevent the system from sending ICMPv4 redirect messages. auto: GEN003610

Enable TCP syncookies. auto: GEN003612

Enable the reverse-path filter. auto: GEN003613

Cause the system to ignore ICMPv6 redirect messages. auto: GEN007860

Configure the system to ignore source-routed IPv6 packets. auto: GEN007940

*From §11.66.16 (Disable WiFi):*

Disable Wi-Fi on Macs by removing the driver files that support it. auto: OSX00060 M6

Turn off AirPort power on Macs if "unused." auto: OSX00385 M6

*From §11.69.2 (Disable NFS client):*

Remove the rpcbind or portmap service wherever it is not necessary (it is necessary where NFS is in use). auto: GEN003810 auto: GEN003815

*From §11.69.3 (Remove rpcbind):*

Remove the rpcbind or portmap service wherever it is not necessary (it is necessary where NFS is in use). auto: GEN003810 auto: GEN003815

*From §11.70.1 (Remove NIS lookup directives):*

On all networks where timeservers exist, use `ntpd` to keep continuous synchronization with the timeservers. auto: GEN000241

*From §11.73.1 (Require admin authentication):*

Make sure we don't automatically obtain any updates. auto: GEN008820

*From §11.75.4 (Passwords on Macs):*

Don't let users change passwords more than once a day. auto: GEN000540

*From §11.84.4 (Ensure only root has user id 0):*

Do not change this policy in a manner to cause root to use a shell not located on the root (/) filesystem. admins do GEN001080

Make sure that root's `PATH`, `LD_LIBRARY_PATH`, and `LD_PRELOAD` environment variables are secure, and that no world-writable directories are on root's `PATH`. auto: GEN000940 auto: GEN000945 auto: GEN000950 auto: GEN000960

*From §11.86 (Managing GPG keys in the RPM database):*

Make sure all packages installed have cryptographic signatures. auto: GEN008800

*From §11.87.1 (Disable rsh, rlogin, and rexec under Mac OS X):*

Under RHEL, to ensure that rsh and rlogin are disabled, uninstall them. auto: GEN003820 auto: GEN003825 auto: GEN003830 auto: GEN003835 auto: GEN003840 auto: GEN003845

*From §11.88 (Samba):*

Remove Samba "unless needed." We do not need it here. auto: GEN006060

*From §11.93 (Serial port console support):*

Do not effect any policy which puts a relative path in the `PATH`, `LD_LIBRARY_PATH` or `LD_PRELOAD` environment variables. admins do GEN001840 admins do GEN001845 admins do GEN001850

*From §11.94.3 (profile.d permissions):*

Don't let users `write` each other, because "messaging can be used to cause a denial-of-service attack." auto: GEN001780

Make sure the `/etc/shells` file exists and has controlled contents. auto: GEN002120

Make sure that all shells listed in `/etc/passwd` are listed in `/etc/shells`. auto: GEN002140

*From §11.97.5 (SMTP smarthosts):*

Disable the decode alias. auto: GEN004640

Configure the mail server to ignore `.forward` files. (See also §11.41.3.) auto: GEN004580

*From §11.99.1 (Automatic software updates):*

Disable automatic software updates on the Mac. auto: OSX00290 M6

*From §11.100 (SSH):*

Configure the SSH daemon for IP filtering using TCP wrappers. auto: GEN005540

*From §11.100.3 (Set login banner):*

Configure the SSH server to reject SSH protocol version 1, which is no longer secure. auto: GEN005500 auto: OSX00175 M6

Disable use of the cipher-block chaining (CBC) mode in the SSH server. auto: OSX8-00-00570 auto: OSX8-00-00575

Disable use of CBC mode by the SSH client. auto: GEN005506 M6 auto: GEN005506

*From §11.100.4 (FIPS 140-2-required SSH configuration):* auto: GEN005511 M6

Disable GSSAPI authentication in the SSH server "unless needed." In some cases we need it. auto: GEN005511 auto: GEN005524

Disable GSSAPI authentication in the SSH client "unless needed." In some cases we need it. auto: GEN005525

*From §11.100.6 (Changes required when IPv6 is enabled):*

Disable GSSAPI authentication in the SSH server "unless needed." In some cases we do not need it. auto: GEN005524

Disable GSSAPI authentication in the SSH client "unless needed." In some cases we do not need it. auto: GEN005525

*From §11.100.8 (Changes required when IPv6 is disabled):*

Disallow TCP connection forwarding over SSH, because of the "risk of providing a path to circumvent firewalls and network ACLs." auto: GEN005515

Disallow gateway ports. auto: GEN005517

Disallow X11 forwarding. auto: GEN005519

Disallow `tun(4)` device forwarding. auto: GEN005531

Limit connections to a single session. auto: GEN005533

Disallow TCP forwarding in the client. (See above.) auto: GEN005516

Disallow gateway ports. auto: GEN005518

Disallow X11 forwarding. See above. auto: GEN005520

Disallow `tun(4)` device forwarding. auto: GEN005532

*From §11.100.9 (Disable SSH tunnelling features):*

Configure the SSH daemon to listen on addresses other than management network addresses, because it is "authorized for uses other than management" here. auto: GEN005504

*From §11.100.10 (STIG-required SSH configuration):*

Disable Kerberos authentication in the SSH server "unless needed." We do not need it. auto: GEN005526

Don't accept any environment variables from the client. auto: GEN005528

Disallow environment settings set by the user and applied by the SSH server. auto: GEN005530

auto: GEN005538

auto: GEN005539

*From §11.101 (Miscellaneous STIG requirements):*

Check for extraneous device files at least weekly. auto: GEN002260

*From §11.101.2 (Uneven access permissions):*

Check for files and directories with unknown owners. auto: GEN001160
auto: GEN001170

*From §11.101.5 (Library files):* auto: GEN001160 M6
auto: GEN001170 M6

When a user logs in, show the date and time of the user's last successful login, and the number of unsuccessful login attempts since the last successful login. auto: GEN000452
auto: GEN000454

*From §11.101.8 (Miscellaneous STIG-required file permission policies):*

Do not deploy any run control script that contains a relative path or empty entry in a PATH variable setting. You should never need to change the `PATH` in a run control script anyway. Similarly, never set `LD_PRELOAD` and never put a relative or empty entry into the `LD_LIBRARY_PATH` used in a run control script. Never deploy a run control script that executes a world-writable program or script. Any run control script that runs a program or script stored on an NFS share should be documented in §3.4. admins do GEN001600 admins do GEN001605 admins do GEN001610 admins do GEN001640

*From §11.101.15 (World-writable directories):*

Disable `xinetd` if no services it provides are enabled. auto: GEN003700

*From §11.104.2 (Allow sudo for a user):*

Always ask for passwords when people use sudo. auto: OSX00110 M6

*From §11.105.2 (STIG-required swap configuration):*

Configure `tcp_wrappers` to grant or deny system access to specific hosts. auto: GEN006620

*From §11.110.3 (STIG-required settings):*

Fix *unowned* files and directories, defined as those whose numerical owner UID or group-owner GID do not map to a known user or group. auto: GEN001160 M6
auto: GEN001170 M6

*From §11.111.1 (Unowned system files):*

"The system must have USB disabled unless needed." All of our CAC readers, and most of our keyboards and mice, connect only via USB, so it's fair to say we "need" USB. Do not disable it. auto: GEN008460

*From §11.112 (USB (Universal Serial Bus)):*

Prevent installation of malicious software or exfiltration of data by restricting the use of mass storage to administrators.

auto: GEN008480

*From §11.113.1 (Under RHEL6):*

Make sure that user ids and user names are unique across all accounts, and that every user's primary group is one defined in the group file.

auto: GEN000300
auto: GEN000320
auto: GEN000380

Make sure that all users have a home, and that each user's home exists.

auto: GEN001440
auto: GEN001460

*From §11.114 (Unix-to-Unix Copy (uucp)):*

Make sure that the UUCP service is disabled.

auto: GEN005280 M6
auto: OSX8-00-00550

*From §11.117 (X Window System server):*

Do not deploy any YUM repository configuration with `gpgcheck=0`. Do sign packages. See §10.

admins do GEN008800

## 2.101 ECSD-1: Software Development Change Controls

*From §6.1 (Database STIG compliance under PostgreSQL):*

For each application which uses the database, make sure that the database users which are used in production are not allowed to execute DDL statements (*e.g.* creating and dropping tables, indices, views, etc.).

DBAs do ECSD-1
DBAs do DG0015

## 2.102 (ECSD-2)

This section is reserved for future expansion.

## 2.103 ECTB-1: Audit Trail Backup

*From §11.78.5 (Pattern for application roles and permissions):*

"Enable auditing on the database." Configure the database to log the messages required by the STIG, and to send those log messages out via the system log. Retention, periodic review, access restriction, and backup, then, are handled via the provisions for such requirements against the system log; see §11.55.1.

auto: ECAR-2
auto: ECRR-1
auto: ECCD-1
auto: ECTP-1
auto: ECTB-1
auto: DG0029
auto: DG0030
auto: DG0031
auto: DG0032
auto: DG0176

## 2.104 (ECTC-1)

This section is reserved for future expansion.

## 2.105 (ECTM-1)

This section is reserved for future expansion.

## 2.106 (ECTM-2)

This section is reserved for future expansion.

## 2.107 ECTP-1: Audit Trail Protection

*From §11.12.1 (Auditing under Mac OS X):*
Fix permissions of audit log files.

auto: GEN002680 M6
auto: GEN002690 M6
auto: GEN002700 M6

*From §11.12.1 (Mac OS X audit log permissions):*
Ensure proper ownership and permissions on audit logs.
Remove extended ACLs on audit logs.

auto: OSX8-00-00205
auto: OSX8-00-00335
auto: OSX8-00-00350
auto: GEN002680
auto: GEN002690
auto: GEN002700
auto: GEN002710

*From §11.56.5 (Sending log messages to a loghost):*
Control permissions on all system log files.
Secure `cron` logs. Secure SMTP logs.
Remove extended ACLs on system log files (including SMTP and `cron` logs).

auto: GEN002710 M6
auto: GEN001260
auto: GEN001260 M6
auto: GEN003180
auto: GEN004500
auto: GEN001270
auto: GEN003190
auto: GEN004510
auto: GEN001270 M6
auto: OSX8-00-00825

*From §11.78.5 (Pattern for application roles and permissions):*
"Enable auditing on the database." Configure the database to log the messages required by the STIG, and to send those log messages out via the system log. Retention, periodic review, access restriction, and backup, then, are handled via the provisions for such requirements against the system log; see §11.55.1.

auto: ECAR-2
auto: ECRR-1
auto: ECCD-1
auto: ECTP-1
auto: ECTB-1
auto: DG0029
auto: DG0030
auto: DG0031
auto: DG0032
auto: DG0176

## 2.108 (ECVI-1)

This section is reserved for future expansion.

## 2.109 (ECVP-1)

This section is reserved for future expansion.

## 2.110 ECWM-1: Warning Message

*From §11.29.1 (Turn off MDNS advertisements):*

Display login banners when the user "connects to the computer remotely," via SSH. auto: OSX00105 M6

*From §11.30 (DoD Login Warnings):*
Install notice and consent warnings for tty logins. auto: GEN000400

*From §11.30.1 (Notice of monitoring on the console):*
Show a warning before the login box under GDM. auto: GEN000402

*From §11.30.3 (Notice of monitoring on Macs):*

**Login warnings on Snow Leopard**

Configure the Mac OS Snow Leopard login window to show a login warning. auto: OSX00100 M6

*From §11.30.3 (Login warnings on Mavericks):*
Configure sshd to show a login warning. auto: GEN005550

## 2.111 (ECWN-1)

This section is reserved for future expansion.

## 2.112 IAAC-1: Account Control

*From §4.1 (Manual Mac compliance):*
Disable guest logon and guest access to shared folders on Macs. admins do OSX00295 M6 admins do OSX00300 M6

*From §11.31 (Fast user switching):*
Disable fast user switching on the Mac. auto: OSX00330 M6 auto: OSX8-00-01100

*From §11.57 (Login window):*
Disable password hints in the Mac login window. auto: OSX00325 M6
Disable automatic login on Macs. auto: OSX00425 M6

*From §11.75.4 (Passwords on Macs):*
Disable accounts when passwords expire. auto: GEN000760

*From §11.113 (Users):*
Remove "application accounts for applications not installed on the system." auto: GEN000290

*From §11.113.1 (Remove unnecessary users):*
Remove the `shutdown`, `halt` and `reboot` users. The requirement says to remove "special privilege accounts" but only mentions these three. auto: GEN000000-LNX00320

Remove the `games`, `news`, `gopher` and `ftp` accounts. auto: GEN000290-1 auto: GEN000290-2 auto: GEN000290-3 auto: GEN000290-4

*From §11.113.1 (Under RHEL5):*

Remove the `shutdown`, `halt` and `reboot` user accounts. The requirement says "special privilege accounts" must be removed, but only mentions these three. auto: GEN000000-LNX00320

Some system users are installed by the `setup` package, but not subsequently used. Remove them. auto: GEN000290

## 2.113 (IAGA-1)

This section is reserved for future expansion.

## 2.114 IAIA-1: Individual Identification and Authentication

*From §4.1 (Manual Mac compliance):*

Do not call the administrator account on a Mac something easy to guess, like "Administrator," or the hostname of the Mac. admins do OSX00015 M6

*From §11.40.2 (Disable Nouveau driver in initrd):*

Make sure that authentication is required before changing bootloader settings. auto: GEN008700

*From §11.40.6 (STIG-required configuration):*

Do not set the `PGPASSFILE` environment variable. users do IAIA-1
users do DG0067

*From §11.41.2 (User guidance about home directories):*

Remove `.netrc` files from home directories. auto: GEN002000 M6

Prevent use of the `.pgpass` file, which could contain unencrypted passwords for the PostgreSQL DBMS. auto: OSX8-00-00600
auto: GEN002000
auto: IAIA-1
auto: DG0067

*From §11.69.4 (Turn off NFS server on Red Hat machines):*

Remove the insecure_locks export option wherever it exists. auto: GEN000000-LNX00560

*From §11.74 (Configure PAM):*

Enforce password guessability guidelines using the `pam_cracklib` module. This module first tries to look the password up in a dictionary using `cracklib`, then applies strength checks as directed. auto: GEN000790

Require a minimum password length of 14 characters. auto: GEN000580

Require passwords to contain at least one uppercase letter. auto: GEN000600

Require passwords to contain at least one lowercase letter. auto: GEN000610

Require passwords to contain at least one digit. auto: GEN000620

Require passwords to contain at least one other (special) character. auto: GEN000640

Prohibit the repetition of a single character in a password more than three auto: GEN000680 times in a row.

Require that at least four characters be changed between the old and new passwords. auto: GEN000750

*From §11.74.3 (Limit maximum logins):*

Remember the last ten passwords and prohibit their reuse. auto: GEN000800

*From §11.74.5 (securetty):*

Change passwords for non-interactive or automated accounts at least once a year, and whenever anyone who has one is reassigned. admins do GEN000740

*From §11.75.4 (STIG-required password configuration):*

Prohibit the use of any of the last fifteen passwords as the next password on Macs. auto: GEN000800 M6

Set a maximum password age on Macs. auto: OSX00020 M6

Set a minimum password length for Macs. auto: OSX00030 M6

Require alphabetic characters in passwords on Macs. auto: OSX8-00-00590 auto: OSX00036 M6

Require symbols in passwords on Macs. auto: OSX00038 M6

Prohibit names from being used as passwords on Macs. auto: OSX00040 M6

*From §11.75.4 (Passwords on Macs):*

Require users to change their passwords at least every 60 days. auto: GEN000700

Enforce the correctness of the entire password, not just the first eight characters of it. auto: GEN000585

Use a FIPS 140-2 approved algorithm for hashing account passwords. auto: GEN000590 auto: GEN000595

Log an error if any user is known to have an empty password. auto: GEN000560

*From §11.76.1 (/etc/pki/tls):*

On select hosts, configure the Pluggable Authentication Modules (PAM) subsystem to allow CAC login from the console using the `pam_pkcs11` module. auto: IATS-1 auto: GEN009120

*From §11.76.2 (CAC Login):*

You should change the passphrase at least once every year, because it's analogous to a non-interactive account password. admins do GEN000740

*From §11.84.4 (Ensure only root has user id 0):*

Ensure that only root has user id 0. auto: GEN000880 M6 auto: OSX8-00-01065

*From §11.95 (Control access to single-user mode):*

Require authentication for access to single-user mode. auto: GEN000020

Require authentication for access to single-user mode. auto: GEN000020

*From §11.95.1 (Securing single-user mode under RHEL5):*

Require authentication for access to single-user mode. auto: GEN000020

*From §11.100.10 (STIG-required SSH configuration):*

Disallow root login over `ssh`: admins must use `su` (§11.101.16) or `sudo` after logging in as themselves.

auto: GEN001020
auto: GEN001100
auto: GEN001120
auto: OSX00165 M6
auto: OSX8-00-00565

*From §11.100.14 (Enable useful SSH features):*
Prevent unencrypted terminal access by uninstalling `rsh` and `telnet`.

auto: GEN001100

*From §11.113.1 (Under RHEL6):*
Make sure that user ids and user names are unique across all accounts, and that every user's primary group is one defined in the group file.

auto: GEN000300
auto: GEN000320
auto: GEN000380

## 2.115 (IAIA-2)

This section is reserved for future expansion.

## 2.116 (IAKM-1)

This section is reserved for future expansion.

## 2.117 (IAKM-2)

This section is reserved for future expansion.

## 2.118 (IAKM-3)

This section is reserved for future expansion.

## 2.119 IATS-1: Token and Certificate Standards

*From §11.76.1 (/etc/pki/tls):*
On select hosts, configure the Pluggable Authentication Modules (PAM) subsystem to allow CAC login from the console using the `pam_pkcs11` module.

auto: IATS-1
auto: GEN009120

## 2.120 (IATS-2)

This section is reserved for future expansion.

## 2.121 (PECF-1)

This section is reserved for future expansion.

## 2.122 (PECF-2)

This section is reserved for future expansion.

## 2.123 (PECS-1)

This section is reserved for future expansion.

## 2.124 (PECS-2)

This section is reserved for future expansion.

## 2.125 (PEDD-1)

This section is reserved for future expansion.

## 2.126 (PEDI-1)

This section is reserved for future expansion.

## 2.127 (PEEL-1)

This section is reserved for future expansion.

## 2.128 (PEEL-2)

This section is reserved for future expansion.

## 2.129 (PEFD-1)

This section is reserved for future expansion.

## 2.130 (PEFD-2)

This section is reserved for future expansion.

## 2.131 (PEFI-1)

This section is reserved for future expansion.

## 2.132 (PEFS-1)

This section is reserved for future expansion.

## 2.133 (PEFS-2)

This section is reserved for future expansion.

## 2.134 (PEHC-1)

This section is reserved for future expansion.

## 2.135 (PEHC-2)

This section is reserved for future expansion.

## 2.136 (PEMS-1)

This section is reserved for future expansion.

## 2.137 (PEPF-1)

This section is reserved for future expansion.

## 2.138 (PEPF-2)

This section is reserved for future expansion.

## 2.139 (PEPS-1)

This section is reserved for future expansion.

## 2.140 PESL-1: Screen Lock

*From §11.38 (GNOME Screensaver):*

Cause the screen to lock after 15 minutes of inactivity, requiring re-authentication to unlock it. auto: GEN000500

Enable the lock setting of the screensaver. auto: GEN000500-3

Set the screensaver idle delay to 15 minutes. auto: GEN000500-2

*From §11.41.5 (Hot corners):*
Prevent users from configuring a hot corner to disable the screensaver. auto: OSX00375 M6
auto: OSX8-00-01095

*From §11.91 (Screen saver):*
Password-protect Mac screensavers. auto: OSX00360 M6
auto: OSX00420 M6
auto: OSX8-00-00020

*From §11.91.1 (Require authentication to exit screensaver):*
Disable administrative accounts from unlocking other users' screens. auto: OSX00200 M6
auto: OSX8-00-00935

*From §11.91.2 (Disallow admins from unlocking user screens):*
Set the screensaver idle timeout to "15 minutes or less." auto: OSX00360 M6
auto: OSX8-00-00010

## 2.141 (PESP-1)

This section is reserved for future expansion.

## 2.142 (PESS-1)

This section is reserved for future expansion.

## 2.143 (PETC-1)

This section is reserved for future expansion.

## 2.144 (PETC-2)

This section is reserved for future expansion.

## 2.145 (PETN-1)

This section is reserved for future expansion.

## 2.146 (PEVC-1)

This section is reserved for future expansion.

## 2.147 (PEVR-1)

This section is reserved for future expansion.

## 2.148 (PRAS-1)

This section is reserved for future expansion.

## 2.149 (PRAS-2)

This section is reserved for future expansion.

## 2.150 (PRMP-1)

This section is reserved for future expansion.

## 2.151 (PRMP-2)

This section is reserved for future expansion.

## 2.152 (PRNK-1)

This section is reserved for future expansion.

## 2.153 (PRRB-1)

This section is reserved for future expansion.

## 2.154 (PRTN-1)

This section is reserved for future expansion.

## 2.155 (VIIR-1)

This section is reserved for future expansion.

## 2.156 (VIIR-2)

This section is reserved for future expansion.

## 2.157 VIVM-1: Vulnerability Management

*From §4.1 (Manual Mac compliance):*

Keep all application software on Macs current with security patches and hotfixes. For Apple-distributed applications, the Apple Software Updater does this. Other applications must also be kept current. admins do OSX00055 M6

Keep the operating system up to date on Macs, as done with the Apple Software Updater. admins do OSX00670 M6 admins do OSX8-00-01265

# Chapter 3

# UNIX SRG Compliance

This chapter has to do with the compliance of Linux machines controlled by this policy, and administrators performing the procedures written here, with the UNIX SRG [6].

In the indices of this document you can find a UNIX SRG Compliance Index. All requirements directly and completely implemented by automated application of this policy are listed in that index as “implemented.” The default Red Hat Enterprise Linux (RHEL) install satisfies some SRG requirements; a list of those is below. In places where the SRG imposes policy demands on the actions of administrators, those demands are passed on in §??. All other requirements are discussed in another section below.

Where RHEL defaults to the correct behavior, but it is simple to write automated policy that will fix anything that is broken, we do that, in an attempt to ensure that UNIX hosts are not only compliant at rollout, but remain compliant over time, and to ensure that noncompliance is rare enough to draw attention where it is warranted.

## 3.1 Requirements that RHEL implements by default

RHEL5 does not include LLC support.

RHEL logs all logon attempts by default.

RHEL assigns the root user a home directory of `/root`, which is not `/`.

RHEL logs all root logon attempts by default.

RHEL logs all su attempts by default.

RHEL sets the root user’s shell to `/bin/sh` by default.

RHEL ensures by default that all system files, programs and directories are owned and group-owned by system accounts, via its packaging system.

RHEL ensures by default that all system library files have permissions of `0755` or more restrictive, via its packaging system.

RHEL comes with OpenSSH in the default install, and telnet and rlogin/rsh

RHEL5:
GEN000000-LNX007580

RHEL5:
GEN000000-LNX007620

RHEL5, RHEL6:
GEN000440

RHEL5, RHEL6:
GEN000900

RHEL5, RHEL6:
GEN001040

RHEL5, RHEL6:
GEN001060

RHEL5, RHEL6:
GEN001080

RHEL5, RHEL6:
GEN001220

RHEL5, RHEL6:
GEN001240

RHEL5, RHEL6:
DCSL-1

RHEL5, RHEL6:
GEN001300

RHEL5, RHEL6:
GEN001100

not in the default install. A small policy that provides defense in depth is in §11.101.

Neither RHEL5 nor RHEL6 as shipped contain files with more permissions for group or other than for user. But §11.101.2 checks for them anyway.

RHEL5, RHEL6: GEN001140

System files under RHEL are always owned by a valid user, because packages that install those files add the corresponding user as necessary. By the same token, packages under RHEL add any groups necessary to own system files. But §11.101.3 checks for unowned files anyway.

RHEL5, RHEL6: GEN001160
RHEL5, RHEL6: GEN001170

All system files, programs and directories under RHEL are owned and group-owned by system users, and do not have extended ACLs. None have write permission for any user but root, including executables relating to network services.

RHEL5, RHEL6: GEN001180
RHEL5, RHEL6: GEN001190
RHEL5, RHEL6: GEN001200

With that said, see §11.101.8 for defense in depth.

All library files under RHEL are mode 0755 or less permissive by default.

RHEL5, RHEL6: GEN001210

RHEL packages do not install any non-root-owned system startup files.

RHEL5, RHEL6: GEN001220

RHEL packages do not install programs not owned by a system account, so run control scripts cannot run such programs.

RHEL5, RHEL6: GEN001240

We do not install any device files via policy or procedure, so all device files are in the vendor-designated directories as required.

RHEL5, RHEL6: GEN001300

RHEL makes `/dev` writable only by the owner, as required. As above, device files are only in `/dev`. World-writable device files are `/dev/random`, `/dev/urandom`, `ptmx`, `/dev/null`, `/dev/zero`, `/dev/full`, `/dev/fuse`, `/dev/net/tun`, `/dev/tty`; these are all world-writable by design. (Other STIG requirements have to do with tunnelling; see the Unix SRG index for more on how we deal with them.)

RHEL5, RHEL6: GEN001660
RHEL5, RHEL6: GEN001680
RHEL5, RHEL6: DCSL-1
RHEL5, RHEL6: GEN001700

Under RHEL default settings, console devices such as the floppy drive and the microphone are managed by the pam_console PAM module, which ensures that the user who is logged in at the console owns these devices and no one else can access them (mode 0600, no extended ACLs). This does not comply with the letter of the requirement but does address the vulnerability discussed therein.

RHEL5, RHEL6: GEN002240
RHEL5, RHEL6: GEN002280
RHEL5, RHEL6: GEN002320
RHEL5, RHEL6: GEN002330

System accounts are disabled by default under RHEL.

RHEL5, RHEL6: GEN002340

Support for non-executable data has been activated by default since RHEL3.

RHEL5, RHEL6: GEN002360

All Linux kernels since 1996 have improved TCP sequence number randomization, in material compliance with this requirement.

RHEL5, RHEL6: GEN002640

For the root filesystem and all other local filesystems, RHEL5 and RHEL6 use the ext3 filesystem by default, which is a journalling filesystem.

RHEL5, RHEL6: GEN003540

RHEL logs all successful and unsuccessful logins by default.

RHEL5, RHEL6: GEN003580

Neither RHEL5 nor RHEL6 provides an FSP server, nor do we deploy one.

RHEL5, RHEL6: GEN003640

RHEL X servers write `.Xauthority` files by default, with mode 0600 and no extended ACLs, as required, and use them for access restriction.

RHEL5, RHEL6: GEN003650

RHEL X servers do not listen for network connections by default, so users cannot permit X display access to unauthorized hosts.

RHEL5, RHEL6: GEN003660

RHEL does not provide AOL Instant Messenger (AIM), MSN Messenger, or Yahoo! Messenger. It does provide the Pidgin instant messaging client, which is the means by which users connect to the DISA-sponsored Defense

RHEL5, RHEL6: GEN005060
RHEL5, RHEL6: GEN005160
RHEL5, RHEL6: GEN005180
RHEL5, RHEL6: GEN005190
RHEL5, RHEL6: GEN005220
RHEL5, RHEL6: GEN005240
RHEL5, RHEL6: GEN005260
RHEL5, RHEL6:

Connect Online (DCO) instant messaging service from RHEL. According to the discussion of this requirement, “Clients used to access internal or DoD-controlled IM services are permitted.”

RHEL does not provide any peer-to-peer file sharing applications.

RHEL5, RHEL6: GEN006040

RHEL6 does not provide any Usenet news server software.

RHEL6: GEN006240

Upon inspection of the source code of the `ldd` command both under RHEL5 and RHEL6, it does not run the executable in question, but hands it as a parameter to the dynamic linker. This means that according to the vulnerability discussion, the `ldd` command suitably “protects against the execution of untrusted files.”

RHEL5,RHEL6: GEN007960

RHEL has had the Exec Shield technology for address randomization since RHEL3 update 3. See `http://people.redhat.com/drepper/nonselsec.pdf` and `http://www.redhat.com/f/pdf/rhel/WHP0006US_Execshield.pdf`.

RHEL5,RHEL6: GEN008420

RHEL public directories are as follows. All public directories are owned by root, group root, and have the sticky bit set. No other world-writable directories exist on a stock RHEL system.

RHEL5, RHEL6: GEN002480
RHEL5, RHEL6: GEN002500
RHEL5, RHEL6: GEN002520
RHEL5, RHEL6: GEN002540

- `/tmp`
- `/tmp/.ICE-unix`
- `/tmp/.X11-unix`
- `/tmp/.font-unix`
- `/var/tmp`
- `/usr/src/debug/tmp`

When installed, VMware Workstation installs a public directory for drag-and-drop functionality, `/tmp/VMwareDnD`. It also fulfills the SRG requirements.

## 3.2 Requirements that are not applicable

## 3.3 Requirements we may not be meeting

## 3.4 Things required to be documented with the IAO

Several SRG requirements say that this or that thing must be “documented with the IAO” (Information Assurance Officer). This section should point readers to the places where that documentation resides, or in degenerate cases (“We don’t have any of these things that must be documented with the IAO”) just say so.

# Chapter 4

# Mac OS X STIG Compliance

This chapter relates to compliance with the Mac OS X STIG.

## 4.1 Manual Mac compliance

Being a UNIX-based operating system, Mac OS X can be configured for compliance programmatically in many cases, and compliance with many requirements levied by the Mac OS X STIG is in fact automated by this policy. But some requirements are not automatable as written because they require human judgment, and for some settings, the effort it would take to programmatically set them is not worth the return. “Patches gratefully accepted,” as they say.

For these requirements and settings, administrators must comply manually.

* * *

Do not install unnecessary packages on a Mac.

admins do OSX00010 M6

Do not call the administrator account on a Mac something easy to guess, like “Administrator,” or the hostname of the Mac.

admins do OSX8-00-01165

Keep all application software on Macs current with security patches and hotfixes. For Apple-distributed applications, the Apple Software Updater does this. Other applications must also be kept current.

admins do OSX00015 M6

admins do OSX00055 M6

Keep the operating system up to date on Macs, as done with the Apple Software Updater.

admins do OSX00670 M6

Disable guest logon and guest access to shared folders on Macs.

admins do OSX8-00-01265

This is done by unchecking the appropriately labelled checkbox found when the Guest user is chosen in the Accounts section of System Preferences.

admins do OSX00295 M6

admins do OSX00300 M6

Do not create temporary or emergency accounts. (This is a trivial fulfillment of the STIG requirements. If these account types are necessary, admins must create and apply policies to ensure their timely expiration.)

admins do OSX8-00-00110

admins do OSX8-00-00115

Make Macs require administrator authentication to unlock each System Preference pane.

admins do
OSX00430 M6

This is done by checking the appropriately labelled checkbox found in the General tab of the Security section of System Preferences.

Turn off Screen Sharing, File Sharing, Printer Sharing, Web Sharing, Remote Login, Remote Management (Apple Remote Desktop), Remote Apple Events, and Xgrid Sharing on Macs.

admins do
OSX00475 M6
admins do
OSX00480 M6
admins do
OSX00485 M6
admins do
OSX00490 M6
admins do
OSX00495 M6
admins do
OSX00500 M6
admins do
OSX00505 M6
admins do
OSX00510 M6

This is done by unchecking the appropriately labelled checkboxes found in the Sharing section of System Preferences.

Maintain "system recovery backups" for Macs as required by the STIG.

The contingency backups that can be made using §11.21.4 may form a large part of your system recovery backup. If you avoid modifying Macs except using this Configuration Management for IT Systems Example Policy  you can rest assured that your "emergency system recovery data" has "been updated following the last system modification."

admins do
OSX00675 M6

See the following sections for more requirements binding on you as a Mac administrator:

- §11.94.5 (Set default umask)

# Chapter 5

# SPAN STIG Compliance

This chapter relates to compliance with the Sharing Peripherals Across the Network (SPAN) Security Technical Implementation Guide (STIG) [2]. See also §7.

You'll need to address some items of compliance yourself, in a site-specific copy of this chapter.

# Chapter 6

# Database STIG compliance

Some pieces of database software are included in RHEL and supported by Red Hat. Because of this, many items of compliance with the Database STIG are provided by the operating system, and others are controlled by this Configuration Management for IT Systems Example Policy. These items are documented here, rather than in the documents of whatever Automated Information System (AIS) may be using the database, to avoid duplication and ensure accuracy.

## 6.1 Database STIG compliance under PostgreSQL

PostgreSQL is included in RHEL. Some Database STIG requirements are therefore met as part of the requirements placed on operating system configuration and maintenance by other documents, like the UNIX SRG. See §6.3 for details on these.

Some requirements are met by configuring PostgreSQL in a certain way. See §11.77.3 for details on these.

Many other requirements on DBMS configuration are met by the default configuration of PostgreSQL as included in RHEL. These are documented in this section.

*　　*　　*

The Database STIG is the primary document used in securing the PostgreSQL DBMS under RHEL. DCCS-1 DG0007

The DBA account for the PostgreSQL DBMS under RHEL is `postgres`, which does not have any “host system administrator privileges.” RHEL5,RHEL6: ECLP-1

PostgreSQL as distributed in RHEL contains no “demonstration or sample databases or applications.” RHEL5,RHEL6: DG0005 RHEL5,RHEL6: DCFA-1

The “DBMS software installation account” for the PostgreSQL DBMS under RHEL, `postgres`, is not permitted to log in by default. Only system administrators can perform actions using the privileges of this user, by the use of the `su` or `sudo` commands; all uses of the account are logged by default. (See the RHEL5,RHEL6: DG0014 RHEL5,RHEL6: ECLP-1 RHEL5,RHEL6: DG0040 RHEL5,RHEL6: DG0041

UNIX SRG Compliance index on UNIX SRG PDI GEN003660 and UNIX SRG PDI GEN001060.)

PostgreSQL as included in RHEL does not include “job queues managed by the database.”

N/A: ECLP-1
N/A: DG0051

PostgreSQL does not use a “client database connection configuration file.”

N/A: ECAN-1
N/A: DG0053

For PostgreSQL as included in RHEL, the lists of “DBMS database objects, database configuration files, associated scripts and applications defined within or external to the DBMS that access the database, and DBMS / user environment files/settings” are as follows:

IAIA-1
DG0067

The list of system-level DBMS-related files can be obtained by running the commands

```
rpm -ql postgresql-server
rpm -q --configfiles postgresql-server
```

on a server with PostgreSQL installed. “User environment files/settings” are stored in the user’s shell initialization file and `.pgpass` file. See §11.40.6 and §7 for more details.

The `psql` command allows specification of a password on the command line; this practice is strictly prohibited, as required, in §7.

IAIA-1
DG0068

PostgreSQL has an *auto-vacuuming* feature which “clear[s] residual data from storage locations after use.” The default configuration included in RHEL enables auto-vacuuming.

RHEL5,RHEL6: ECRC-1
RHEL5,RHEL6: DG0084

We need to revisit DBA users in light of later checklist requirements.

DG0085

A review of the PostgreSQL 8.4 documentation has shown that PostgreSQL does not support “objects defined within the database, but stored externally to the database.” Thus they are implicitly disabled, which fulfills the requirement.

DCFA-1
DG0098

PostgreSQL as included in RHEL is prevented from running external executables by the SELinux policy.

RHEL5,RHEL6: DCFA-1
RHEL5,RHEL6: DG0099

PostgreSQL as included in RHEL is prevented from running external executables by the SELinux policy; therefore no OS accounts are used to “execute external procedures.”

N/A: DCFA-1
N/A: DG0101

Since PostgreSQL as included in RHEL does not support external objects and cannot run external executables, users inside the database are effectively (if trivially) prevented from accessing “objects stored and/or executed outside of the DBMS security context.”

RHEL5,RHEL6: ECLP-1
RHEL5,RHEL6: DG0120

All “DBMS processes or services” for PostgreSQL as included in RHEL are owned by the `postgres` user, which is a “custom, dedicated OS account.”

RHEL5,RHEL6: DCFA-1
RHEL5,RHEL6: DG0102

All “DBMS data files, transaction logs and audit files” for PostgreSQL as included in RHEL are stored in “dedicated directories... separate from software or other application files.” These are under `/var/lib/pgsql`, and there are separate directories for each of the three kinds of files. Permissions to each are “customized to allow access only by authorized users and processes.”

RHEL5,RHEL6: DCPA-1
RHEL5,RHEL6: DG0111

DBMS system data files for PostgreSQL are “stored in dedicated disk directories.”

RHEL5,RHEL6: DCPA-1
RHEL5,RHEL6: DG0112

To prevent “database tables from unrelated applications” from being “stored

DBAs do DCPA-1
DBAs do DG0113

in the same database files" under PostgreSQL, ensure that for each "unrelated application" there is a separate database, using the `createdb` utility as appropriate.

Make sure that "DBMS files critical for DBMS recovery" are "stored on RAID or other high-availability storage devices," by specifying a RAID hard drive setup when procuring any server on which a PostgreSQL database will reside.

admins do COBR-1
admins do DG0114

Do not grant "DDL (Data Definition Language) and/or system configuration" privileges to non-privileged DBMS users. To obtain a "list of privileged role assignments" in an installation of PostgreSQL as included in RHEL, perform the following commands as root on the server in question:

DBAs do ECLP-1
DBAs do DG0116

```
sudo -u postgres psql
\l
[A list of databases and privileges is output.]
\du
[A list of roles and privileges is output.]
\c foo
\dp
[A list of objects and privileges is output.]
\q
#
```

Replace 'foo' in the above directions with the name of a database from the list output by `\l`. There may be multiple databases. This data is sent to administrators automatically in a monthly report; see §11.78.6.

See §6.7 for the list of IAO-approved DBA role assignments.

Access to "DBMS system tables and other configuration or metadata" is suitably restricted by default. See Chapter 44, "System Catalogs," in [7].

RHEL5,RHEL6: ECAN-1
RHEL5,RHEL6: DG0123

Do not use a privileged database account for non-administrative purposes. For each application in the database, create a per-application object owner user and/or per-application administrator user; use one of these, and not a DBA account, to create the objects necessary for the application and to maintain the application. Disable this account "when not performing installation or maintenance actions."

DBAs do ECLP-1
DBAs do DG0004
DBAs do DG0124

For each application which uses the database, make sure that the database users which are used in production are not allowed to execute DDL statements (*e.g.* creating and dropping tables, indices, views, etc.).

DBAs do ECSD-1
DBAs do DG0015

"Trustworthiness" of "data files and... configuration files" for PostgreSQL as included in RHEL is provided by the underlying operating system. See §2.53 for a summary of measures taken to preserve system state integrity.

DCSS-1
DG0155

According to its documentation [7], PostgreSQL does not appear to provide a means to "restrict the number of failed logins within a specified time period."

N/A: ECLO-1
N/A: DG0160

A review of the PostgreSQL documentation [7] indicates that there is no way to turn off transaction journalling in PostgreSQL; thus it is enabled as required, but the checklist says, "If no configuration settings are available to enable or disable transaction journaling, this check is Not Applicable."

N/A: ECDC-1
N/A: DG0170

Do not grant "privileges to restore database data, objects, or other configuration or features" to unauthorized DBMS accounts.

DBAs do ECLP-1
DBAs do DG0063

Because PostgreSQL as included in RHEL "does not provide the capability to mark or label sensitive data within the DBMS, this check is Not a Finding."

ECML-1
DG0087

PostgreSQL as included in RHEL "does not provide a method or means for configuration of account lock times," so "this check is Not a Finding."

ECLO-1
DG0133

## 6.2 Database STIG compliance under SQLite

SQLite is not a traditional server-based database. It is, quoting from its website, "a software library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine." Because it does not implement a client-server interaction model, it doesn't listen over the network, nor authenticate users. Authorization to operate on the database is based on operating-system-level access to the single file containing the database, so there is no system of user accounts with differing levels of access. SQLite also has no run-time configuration. Consequently many Database STIG requirements cannot be applied to SQLite. Those dealing with control of the files that comprise an SQLite installation, including security patches, are applicable and are covered in §6.3.

The Database STIG is the primary document used in securing the SQLite DBMS, as far as it applies.

DCCS-1
DG0007

SQLite as distributed in RHEL contains no "demonstration or sample databases or applications."

RHEL5,RHEL6: DCFA-1
RHEL5,RHEL6: DG0014

## 6.3 Databases included with RHEL

Some requirements are met by existing policies and procedures written throughout this Configuration Management for IT Systems Example Policy and notated in those existing places. See the Database STIG Compliance index for an exhaustive list of these; look on the referenced pages for phrases like "databases included with RHEL."

For DBMSes included with RHEL, updates and patches are handled as for any RHEL update. See §**??**.

VIVM-1
DG0003
DG0097

Permissions for software libraries relating to DBMSes included with RHEL are controlled by RHEL's packaging system, and are restricted to the fewest accounts requiring access.

DCSL-1
DG0009

Permissions and changes to database executable and configuration files for DBMSes included with RHEL are checked periodically by §11.86.1 and §11.6.

DCSL-1
DG0010

"Software libraries" and "management tools" for DBMSes included with RHEL are managed and patched using the same procedures as the operating system software. See §**??**.

DCPR-1
DG0011

Data and configuration directories for DBMSes included with RHEL, where applicable, are dedicated for those purposes by the operating system. For executables and libraries, SELinux is the "method that provides... separation of

RHEL5,RHEL6: DCPA-1
RHEL5,RHEL6: DG0012

security context.” Access controls are well-defined through the RPM packaging system, mitigating the discussed vulnerability.

DBMSes included with RHEL have separate components in separate RPM packages; unneeded components are not installed.

RHEL5,RHEL6: DCFA-1

For DBMSes included with RHEL, ownership of “DBMS software libraries and configuration files files” is set by the vendor in the RPM package.

RHEL5,RHEL6: DG0016

DBMS system files for DBMSes included in RHEL are provided on the OS media; trusted recovery measures used for the OS apply to the DBMS software as well.

RHEL5,RHEL6: DCSL-1
RHEL5,RHEL6: DG0019
RHEL5,RHEL6: COTR-1
RHEL5,RHEL6: DG0115

“The DBMS warning banner should meet DoD policy requirements,” but “a warning banner displayed as a function of an Operating System or application login for applications that use the database makes this check Not Applicable.” See §2.110 for summaries of where warning banners are installed by this policy; per-application warning banners are covered in per-application documentation.

N/A: ECWM-1
N/A: DG0179

“DBMS software libraries” for DBMSes included in RHEL are part of the operating system distribution, so OS install media contains them. See §**??** for procedures regarding OS install media; see §2.23 for other assurances about software needed for operations continuity.

COSW-1
DG0187

## 6.4 Requirements implemented by each system

(“System” here means an Automated Information System (AIS), not an individual host.) The requirements not covered by this Configuration Management for IT Systems Example Policy, which must be addressed in per-system documentation, are summarized here:

**DG0011** How database configuration files and stored procedures are configuration-managed; how system personnel interface with IT regarding database software patches

**DG0013** How the database is backed up and recovered, and evidence that such procedures have been followed

**DG0017** Whether production and non-production databases reside on the same host, and, if so, who authorized that

**DG0019** Ownership of “application software and configuration files” (this may be covered in a system-specific way by a piece of policy elsewhere in this Configuration Management for IT Systems Example Policy rather than in a system-specific document)

**DG0020** How database backups are verified and backup procedures tested, and evidence that such testing and verification procedures have been followed

**DG0064** How backups are protected during all phases of backup and recovery

**DG0065** How users are authenticated using DoD PKI certificates, or how this requirement is mitigated

**DG0069,DG0076** How exported production data is protected and modified, if or when it is imported into a development database

**DG0066, DG0067, DG0071, DG0072, DG0073, DG0079, DG0125, DG0126, DG0127, DG0128,** Considerations regarding password authentication, if it is used

**DG0078** A list of authorized DBMS accounts, and how each use of those accounts is mapped to a specific person

**DG0088** How periodic and unannounced vulnerability scans of the database are conducted

**DG0090,DG0092,DG0106** How sensitive data are encrypted at rest if required

**DG0093** How remote administrative database access can only happen over encrypted channels

**DG0096** How DBMS IA policies and procedures are reviewed at least once a year

**DG0103** How the DBMS server software is configured to limit access by network address

**DG0104** How DBMS "services/processes" are named in a clearly identifiable fashion. "An example ... [is] `prdinv01`."

**DG0107** Identification of any "sensitive data" such as PII or classified which is stored in the DBMS

**DG0108** "Assignment of the priority to be placed on restoration of the DBMS"

**DG0109** How the DBMS is isolated from other application services

**DG0110** How the DBMS is isolated from "security support structures" such as Windows domain controllers or Kerberos servers

**DG0116** A list of IAO-approved roles "assigned privileges to perform DDL and/or system configuration actions"

**DG0118** How the IAM reviews changes to DBA role assignments

**DG0119** That the "application user" does not have "administrative privileges."

**DG0124** Which DBMS accounts, specific to the application, create and maintain the DBMS objects needed by the application

**DG0151** That the DBMS listens on a static, default port, if the DBMS listens over the network

**DG0156** Who is the IAO for the DBMS, and evidence that the IAM has assigned and authorized that person

**DG0157, DG0158, DG0159, DG0198** How remote database administration is either disabled, or documented, authorized, audited, monitored by the IAO or IAM, and done over an encrypted channel

**DG0167** How sensitive data served by the DBMS is encrypted in transit

**DG0186** How the database is protected from access originating from public or unauthorized networks

**DG0187** (Possibly) How to quickly reinstate operation of the application that talks to the database, in case of contingency

**DG0075,DG0190,DG0191,DG0192** How the database talks to remote databases and applications in a compliant and secure manner, if it does

**DG0089,DG0194,DG0195** How developers are kept from disturbing production DBMS instances

**DG0008** A list of authorized object owner users in the application's database

**DG0060** A list of "non-interactive, n-tier connection, and shared accounts," evidence of approval of these by the IAO, and how each action taken by one of these users can be traced to an individual person

**DG0070,DG0074** User account lifecycle for database users, including deletion

**DG0091** How "custom and GOTS application source code stored in the database" has been "protected with encryption or encoding"

**DG0105** Authorized list of privileges for application users

**DG0119** That application users do not have "administrative privileges" such as creating tables and other DDL

**DG0121** How application user privileges are granted solely by granting membership in roles, not directly to the application user

**DG0122** How access to "sensitive data" (such data as the information owner would deem as requiring encryption) is restricted to authorized users

**DG0138** How "access grants to sensitive data" (such as requires encryption) are "restricted to authorized user roles"

**DG0165,DG0166** How symmetric and asymmetric (resp.) keys are protected in a compliant fashion (if data is encrypted in the database)

**DG0172** How changes to security labels are audited (if sensitive data needing encryption or classified data are stored in the database)

**DG0193** How non-interactive account passwords expire at least every year

See the Database STIG and security checklists for details on these requirements.

## 6.5 Requirements which may become applicable in future

**DG0085** If a database administrator is needed in future, the least privileges needed by that user for day-to-day operation must be determined, and the user must be limited to those privileges.

No PostgreSQL installations under the purview of this policy accept connections across "network or enclave boundaries as defined in the PPS CAL" at `http://iase.disa.mil/ports/index.html`. N/A: DCPP-1 N/A: DG0152

Because no DBMSes containing classified information are presently managed by this policy, there are trivially no interconnections between DBMSes of differing classification levels. ECIC-1 DG0171

Because no DBMSes using replication are presently managed by this policy, no DBMS accounts exist for the purpose of replication. DCFA-1 DG0100

Because no DBMSes containing classified information are presently managed by this policy, DBMS users need not be notified at login regarding previous successful and failed DBMS login attempts. ECLO-2 DG0135

## 6.6 Requirements which need further attention

For PostgreSQL as included in RHEL:

Names of applications that access the database may not be logged in the audit trail. DG0052

Because names of applications may not be logged, DBMS access using unauthorized applications may not be discovered by monitoring the audit logs. DG0054

## 6.7 Things which must be documented with the IAO

The IAO-authorized list of "roles assigned privileges to perform DDL and/or system configuration actions in the database" in PostgreSQL as included in RHEL is this: ECLP-1 DG0087 DG0116

- `postgres`

As the `postgres` user cannot log in, only system administrators can become this user.

Changes to this list must be discussed with the IAO, and changes are of course tracked. Each AIS may also have a list of database administrative roles.

The IAO-authorized list of DBA role assignments in PostgreSQL as included in RHEL is this: DCSD-1 DG0153

- `postgres`

As the `postgres` user cannot log in, only system administrators can become this user.

Changes to this list must be discussed with the IAO, and changes are of course tracked. Each AIS may also have a list of database administrative roles.

# Chapter 7

# Procedures for users

This chapter contains directions for users of hosts covered by this Configuration Management for IT Systems Example Policy.

## 7.1 Security Features User's Guide

This section contains guidance on the security features of information systems under this Configuration Management for IT Systems Example Policy as required by the SPAN STIG.

### 7.1.1 Single-user KVM switches

Single-user keyboard-video-mouse (KVM) switches are used on unclassified systems to reduce clutter due to too many keyboards, mice and monitors on a desk.

Here's what you need to know about how to operate these KVM switches securely: KVM01.002.00

1. Before interacting with a system connected to a KVM switch, make sure it's the system you think it is, and verify its classification. It should have a banner that lets you know this information.

2. Before switching to another system, lock your screen; then verify the identity and classification of the system you've switched to before interacting with it.

Do not connect a keyboard with a smartcard reader to a KVM switch. users do KVM01.004.00

Do not connect a wireless keyboard or mouse to a KVM switch. users do KVM01.005.00 SPAN STIG PDI KVM01.005.00 says that such devices must comply with the current Wireless STIG, and the current Wireless STIG says there are presently no compliant devices. (In order for them to be compliant, they would have to use FIPS 140-2 compliant encryption.)

### 7.1.2 Removable devices: prohibitions and requirements

Here are some DoD-level requirements that you, the user, should know about.

When removing a hot-swappable device such as a USB device from one computer and connecting it to another, you must wait at least 60 seconds in between. users do USB00.001.00

MP3 players, camcorders and digital cameras must not be attached to information systems (ISes) without prior DAA approval. users do USB01.001.00

No USB device may be connected to a DoD IS unless approved by the Information Assurance Officer (IAO). users do USB01.002.00

Thumb drives that look like anything else besides a thumb drive (e.g., a watch, a pen, a piece of sushi, a little teddy bear...) are not permitted and will be confiscated. USB01.003.00

Any USB device with persistent memory (e.g., USB hard drives) must be formatted with a modern filesystem (e.g., NTFS, ext3, HFS; not FAT). users do USB01.008.00

### 7.1.3 USB usage and handling

Existence of this section is required by SPAN STIG PDI USB01.009.00. Discussion within this section of USB devices with persistent nonremovable memory is required by SPAN STIG PDI USB01.010.00. USB01.009.00 USB01.010.00

Under current directives, you should not plug any USB storage device into any host without authorization from the Information Assurance Manager (IAM), authorization that is specific to you, the computer in question, and the storage device in question.

## 7.2 Miscellaneous prohibitions

When using the `psql` client to connect to the PostgreSQL database, do not supply on the command line a conninfo string containing a password. (Conninfo strings are described in the libpq documentation; try this URL: `http://www.postgresql.org/docs/8.4/static/libpq-connect.html`.) This requirement flows from the more general requirement that database passwords must not be stored in clear text. IAIA-1 IAIA-2 DG0068

# Chapter 8

# Contingency

## 8.1 Contingency procedures

A contingency has happened; one or more workstations or servers must be reconstituted. You have these options:

- If you're building up one host in a temporary situation, it may be simplest to go through this policy, manually implementing its requirements on the machine in question. If you're not in the usual production setting (e.g., filers are missing, networking to another building is out), you may not want to follow the policy exactly, and when manually rebuilding, you don't have to.
- If you're setting up more than one host, or running for a while, it's probably easier to set up a *puppetmaster* and maybe a kickstart server; this way, the hosts will implement the policy themselves, which is faster and less error-prone.

We'll discuss the first alternative here; the second is the same as normal production usage, which is detailed in §**??** and §**??**.

If you'll be reading through this policy and manually applying it to a machine, you'll need to know the syntax and semantics of the policy. In general, refer to [17] and [9]. A few salient specifics follow.

Start with `nodes.pp` (§11.1). Find the node declaration for the host you are concerned with. Follow references from there to high-level classes in `templates.pp` (§11.2), thence to the modules, where you will find the details of how the host is configured. Some pieces of the policy act based on *facts* about the host, like `$::hostname` or `$::kernelrelease`. You'll have to deduce the values of these facts yourself and act accordingly.

Whichever way you choose, you must still personally follow the procedures in §**??**.

## 8.2 Contingency preparedness

Some parts of this policy detail the ways that hosts under this Configuration Management for IT Systems Example Policy should prepare for contingency situations: §11.21.4, §11.55.1.

# Chapter 9

# Maintenance

This chapter discusses how to maintain this policy, both as a set of rules for computers to follow and as a document for humans to read. We'll talk about how to build your own copy of this document; a general approach to using policy-based tools to maintain a set of systems; details you need to keep in mind as you maintain the policy and this document; and what you would want to keep in mind if you were to make CMITS over from scratch.

## 9.1 How to build a copy of this document

First, obtain a copy of the document's sources. The Configuration Management for IT Systems Example Policy is frequently stored and tracked in a Subversion repository. We'll say, for example, that everything is under `https://example.com/svn/trunk/myorg-cmits`. Check out a working copy of the directory using your Subversion client. In your working copy folder, you will find `modules-*` and `manifests` directories, which contain the Puppet source code and other attendant files, and you should find a `unified-policy-document` directory. This directory is where you can generate the policy document from the manifest stored in the `modules-*` and `manifests` directories and the documentation stored in the `latex-*` directories. See the README.txt in the `unified-policy-document` directory for how to proceed.

## 9.2 General process

Here, in general, is how to maintain this policy. We'll use the word *problem* here to mean something that needs to be changed. Think of it like a word *problem*, not like a drinking *problem*.

**A problem appears:** A new security requirement comes down, or a user can't run a program. The machine as configured by the policy does not do what is needed.

**Relate the problem to configuration:** How does the configuration of the machine bring up the problem? Is a wrong directory on the path, does a package need to be installed?

**Express the solved state:** With the problem solved, what's different about the system? Is there an extra line in a file? A certain version of a package installed? Does a file have different permissions? That end state is what you will express with Puppet, not so much the exact steps needed to get there.

**Have a policy working copy:** Check out, if necessary, a copy of the policy from the Subversion server.

**Locate the configuration and relate it to the policy:** Think about what subsystem needs to be configured. Each *module* in the `modules` directory deals with a subsystem, e.g. `ssh`, `nfs`. Find or create the module to which your configuration belongs. Each module contains *manifests*, files which contain *classes*, which in turn contain enough *resources* (the individual units of configuration) to express a single goal. For example, `ssh::no_tunnelling` is a class which turns off all tunnelling of network connections and X11 traffic through SSH sessions.

**Change, write, or co-opt classes that change the configuration:** If you write classes, use other modules as examples, and Puppet reference documentation as a resource. When writing, keep in mind that you have four audiences: Puppet, which will be implementing the policy; other administrators, who need to read and understand the policy; your future self, six months down the road; and auditors. See below for more details about how to write for each of these audiences.

The Puppet community has a set of common modules called the Puppet Forge; if you use one, take intellectual rights into consideration, be sure you know what other modules it depends on, and count on re-documenting it: the CMITS documentation scheme, for better and for worse, serves more purposes than puppetdoc does.

If you change a module, be sure you know where in the policy it is used: you may be reconfiguring more hosts than you think.

**Change manifests to include your classes:** On what nodes, or hosts, does the change need to happen? All hosts which are to be compliant with the requirements of a document (like a STIG)? All hosts in a given room? All hosts belonging to a given subgroup of the organization? Find or create a suitable class in `manifests/templates.pp`; modify `manifests/nodes.pp` if necessary to make some hosts include your new class.

**Test:** Use rspec-puppet to test everything about your module that you can. Such tests can be easily automated and are saved with the code. The quickest way to make sure your module does what you want on your own host is to use `puppet apply` something like so:

```
sudo puppet apply <<< 'include mynewclass'
```

Then if it didn't work right, manually restore whatever system settings were changed and try again.

**Manage changes:** Use a software version control system to track changes to the policy. This helps answer questions of why a change was made later on, and ensures that your changes are properly backed up and deployed.

## 9.3 Invariants

As you maintain the policy, there are several important properties of it that you must maintain.

**Self-documenting:** Write everything you know about the aspect of the configuration that your policy changes. See §11.33 (as of revision 4597, 1 Nov 2011) for a good example. This property makes the policy document mean something to human administrators (including your future self), both during production and in a contingency situation. It also makes the policy document a central place for small but important facts about quirks of the subsystems being configured.

**Discoverable:** Not only the policy files themselves, but also the policy document and its history should be easy places to search for needed knowledge. Take the time to write a cross-reference to another section of the policy, a bibliographical citation to another document containing guidance, the official number of a controlled requirement, a revision number in the admin repository when something was fixed or broken. Links like these made the World Wide Web the amazing resource it is.

**Flexible:** To the greatest extent possible, the policy should not write over changes not under its control. For example, §**??** edits Postfix's configuration, rather than copying an entirely new configuration file over the old one. If an updated postfix package is issued because of a security update, and it changes the Postfix configuration slightly in an unrelated area, the policy that edits the file will still work against updated machines, while a policy that copied over the file would miss something.

**Authoritative:** Any change that needs to be made to any system should be part of the policy. This property is what makes contingency recovery using this policy so easy, and what makes this policy document as complete as it is.

**Managed:** Every change you make should be checked into the version control repository, under your name. This eases compliance with audit-related regulations, and plays into the automated policy updating and backup that's part of the policy (§**??**).

**Convergent:** The thing that lifts Puppet above shell scripting is that when you use its elements to write your policy, you gain the guarantee that a managed host will always move toward conformance with the policy. If you write a shell script, you have to make sure it's *idempotent* (running it multiple times has the same effect as running it once), and that it deals with all possible errors and unexpected inputs.

## 9.4 How to write this document

Any line in a `*.pp` file which starts with a pound sign (#) will be fed to LaTeX when the documentation is built. This is by means of `shaney`, which strips the comment characters off, and surrounds uncommented Puppet policy code with verbatim tags so that it will be typeset as code, and so that LaTeX will not search it for markup tags. The outputs of shaney for each file are concatenated in a certain order.

`shaney` also constructs the §2 (Compliance by IA control) chapter.

Here's what this means for you, the documentation writer:

- If you put an underscore (_) in a comment, put a backslash (\) before it so that LaTeX will not barf.
- Comments with whitespace before the # character are typeset as code; comments starting on the first column are treated as discussion. If you comment something out, kindly put spaces before the # characters, so that your commented-out policy won't be treated as text. By the same token, if you write a comment about some nicety of Puppet syntax you used, and not about what the policy is, you may want to indent it.
- In any module, the `init.pp` comes first, then other `*.pp` files in the same directory in alphabetical order, then `*.pp` files in subdirectories in alphabetical order. So you should start the `init.pp` with `# \section{`*`Subsystem name`*`}`; start other `*.pp` files with a subsection directive, and `*.pp` files in subdirectories with a subsubsection directive, so that the structure of the finished document mirrors the directory structure of the module.
- If you write `\S\ref{class_other::class}` in the comments of your file, readers of the raw text of the file will know to look at `modules/other/manifests/class.pp`; when the document is typeset, the reference will turn into a hyperlink to the section number where the class is written.
- When you write an implements tag `\implements{iacontrol}{FOO-1,BAR-1}`, all lines from that line to the next paragraph break or to the next piece of Puppet code will go into the Compliance by IA control chapter. So aim that first paragraph toward auditors: use language familiar to them by quoting the requirement; don't go into detail about the policy, or things you found out while configuring the system properly; and don't say anything funny or offer personal opinions. Write details and opinions in ensuing paragraphs.

- There's a LaTeX cheat sheet at `http://www.stdout.org/~winston/latex/`.
- Changes to SELinux are usually deployed in *policy packages*, which are files whose names end with `.pp`. If you store any of those files within this policy, you must make sure that the name of the file inside the policy ends not just with `.pp` but with `.selinux.pp`. That way, the scripts that build the unified policy document will know that such files do not contain Puppet code and LaTeX comments, but SELinux policy.
- Write only plain text in section or chapter names: no markup, such as `\emph` or `\tt`. Normally LaTeX supports this, but the hack which automatically writes names of pertinent IA controls after section names in the table of contents is brittle, and causes LaTeX to fail when you do this.

## 9.5 How it all works

The building of this document is done by `sourapples`, which is a part of `shaney`. `sourapples` first generates all the generated parts of the document, then calls LaTeX, `makeindex`, and other utilities, to typeset the document.

### 9.5.1 Written LaTeX parts

The main document is `main.tex`. This sets the title of the document, pulls in the LaTeX packages used, and includes each chapter of the document in order.

Prose chapters and document parts are included from the `latex-fouo` (if it exists) and `latex-unclass` directories.

Some chapters are not written, but generated from many smaller files. These are the generated parts.

### 9.5.2 Generated parts

The Puppet policy is stored in the `*.pp` files in the `manifests`, `modules-unclass` and perhaps `modules-fouo` directories. Shaney finds them all, removes comment characters and surrounds Puppet code with verbatim tags, resulting in the `policy.tex` file. During this process it generates the index directives that result in the indices of classes, defined resource types, and files. It also pulls out per-IA-control excerpts using the `\implements` tags. The documentation in the Puppet code is pulled together into the "Policy" chapter; the excerpts comprise the "Compliance by IA control" chapter.

The attendant files are in the `modules-*/*/files` directories. `sourapples` gathers them and converts the ones which seem to be made of readable text into a form suitable for inclusion into the policy document. The result of this is the "Attendant files" chapter.

## 9.6 Document requirements

If you were to transition this document to another document preparation system, you would need to re-engineer it from its requirements, and so you would need to know those requirements.

Guiding principles for the policy are outlined in §9.3. Guiding principles for this document are given in the Colophon (§1.3).

Sources of requirements for this document:

- We are administering computers every day with the contents of this document, and functional requirements on their configuration change every day. To successfully document this, our documentation must be primarily organized in the same way our problems and configuration changes are.
- We are submitting this document to other organizations to back up our claims of compliance with several *requiring documents* (for example, the UNIX SRG). Those other organizations don't have time to read our whole document.
- In case of contingency we may need to read directly how systems should be configured, rather than delegating the task of configuring them to a tool.
- Several *requiring documents* (for example, the UNIX SRG) place *named requirements* on the configuration of our computers or our procedures. We need to know what our expected compliance posture is, *i.e.*, the set of named requirements met when the policy is applied, plus the set of reasons why unmet requirements are unmet. The *requiring documents* may change a few times per year; corresponding changes to our policy may be needed.
- We are making a document inside the DoD.

Requirements:

1. The parts of the document containing the policy must be programmatically constructed from comments written in the policy.
2. Other parts containing prose (such as the part you are reading right now) must also be integrated into the document.
3. Supporting files, such as configuration files copied into place by the policy, should also be included in the document.
4. It must be easy to notate our posture as regards *named requirements*, such as IA controls and requirements from multiple STIGs—both in comments in policy files and in prose sections. The postures regarding compliance at the time of this writing are:
    - this piece of the policy automates compliance

- we are not yet compliant
- compliance comes through the action of some people, like administrators, or users
- the default configuration of an operating system or piece of software we use is compliant
- the requirement is not applicable
- the requirement is to document something, and here is the documentation of that thing

5. It must be easy to see whether a piece of the document has to do with a named requirement, which one, and what the posture is. For example, a compliance notation could result in a margin note in the document, which is red if we are not compliant.

6. It must be easy to find all parts of the document relating to a given requirement, and what posture they put us in. For example, each compliance notation could result in an entry in a per-requiring-document index, notating that the requirement is "automated," or "N/A."

7. It should be easy to find all parts of the policy relating to a given file, class or defined resource type.

8. Where one part of the policy refers to another (*e.g.*, a class includes another class) there should be a quick way to get to the corresponding part of the document, like a clickable link.

9. There must be a way to get quickly to a given section of the document, for example a table of contents, or if the output file is a PDF, PDF bookmarks pointing to each section.

10. Along with the name of each section in the table of contents, there should be a list of the IA controls dealt with in that section.

11. A summary of compliance broken out by requiring document, in CSV (Comma-Separated Value) format or a similarly easy-to-parse format, must be derived from the compliance notations—including short prose reasons for non-compliance. (CSV may not be appropriate for the prose.)

12. There must be a chapter which summarizes compliance with IA controls, sorted by IA control. It must be programmatically generated. It should provide a quick way to get to the detailed parts of the document relating to each IA control.

13. A given compliance posture notated with regard to a STIG requirement, where the STIG details IA controls related to each STIG requirement, must be programmatically interpreted as the same posture with regard to the corresponding IA controls, and summarized in the per-IA-control chapter as such.

14. Security labels must be written at the top of every page.

15. The title page must contain a security label, the title, the date, the organizational logo, a distribution statement and a destruction notice.

# Chapter 10

# Packaging

You should put software in packages where possible. This chapter discusses how and why, in general terms. How this works out in your organization will vary.

## 10.1 Why package?

Packaging software makes it easier to add, remove and upgrade. It also can push the work of satisfying software dependencies off of you, the administrator, and onto the packaging system. Software that's been packaged and installed is on the local hard drive of each machine, so it works just as well when the network is gone (on the laptop of someone who is on a business trip, for example), and runs faster. It's easier to control the interactions between software providing some duplicate functionality (*e.g.*, OpenMPI vs. MPIch2) when it's in packages—if it's not installed it's unavailable, and if you want a per-machine or per-user choice, the *alternatives* subsystem or the *modules* subsystem can help you to make that choice completely and simply.

## 10.2 The RPM package manager

We speak here about packaging in the context of *RPM*, the RPM Package Manager (formerly Red Hat Package Manager). RPM supports installation, removal and upgrade of packages of software, and keeps track of data about packages which eases administration, such as which packages depend on which others, whether a package has been cryptographically signed, what versions of packages are installed, and whether files which have been installed as part of a package have changed since being installed.

Before packaging a piece of software you will want to see if someone else has packaged it already and if that package is suitable. Fedora's EPEL (Extra Packages for Enterprise Linux) project (`http://fedoraproject.org/wiki/EPEL`) packages some software not packaged as part of Red Hat Enterprise Linux. When obtaining and installing RPM packages not from the vendor, you should make sure you trust the packager. Owing to RPM's flexibility and use across several distributions of Linux, random RPMs you find on the Internet will not necessarily install or run properly on Red Hat Enterprise Linux.

If a package is not already extant for the software you need, you can make your own package. The act of packaging software with RPM is usually almost as easy as installing it from source. See the Fedora RPM Guide (`http://docs.fedoraproject.org/drafts/rpm-guide-en/`) for more about the generalities of this topic; specifics will vary by organization.

## 10.3 Organization-specific details

You should write your own organization-specific guidelines for how to package software, how to track and control changes to your organization's custom packages, and how to deploy packaged software.

# Chapter 11

# Policy

Here follows the policy itself, broken into sections by module, and subsections by class. As required, DoD reference documents constitute the primary source for security configuration done by this document. This Configuration Management for IT Systems Example Policy where applied, configures the “DBMS host platform” for “compliance with applicable STIG requirements.” It also “hardens” some “separately configured components that access the database including web servers, application servers, report servers, etc.” See the compliance index (§16 for overviews. When properly installed, this Configuration Management for IT Systems Example Policy also “regularly audit[s] the security configuration” of subsystems under its control “to confirm continued compliance with security requirements.” See §?? for details of how regular policy enforcement is set up.

auto: DCCS-1
auto: ECSC-1
auto: DG0175

## 11.1 site.pp

Here are sitewide defaults.

```
import "templates"
import "nodes"

Exec { path => "/bin:/sbin:/usr/bin:/usr/sbin" }
File { ignore => ".svn" }
```

## 11.2 nodes.pp

Here is the definition of each node known in this policy. (A *node* is any host which runs Puppet, virtual or physical.) Classes included here will be defined in §11.3.

```
import "templates"

node example1 {
    include example_org_workstation                                    §11.3
}
```

## 11.3 templates.pp

Here are the primary units of functionality needed to configure nodes within our administration. Classes referred to with the `include` directive implement smaller units of policy and are covered in the ensuing sections of §11.

```
class unix_stig_compliance_base {
    include aide                                              §11.6
    include ssh::stig                                         §11.100.10
    include stig_misc                                         §11.101
    include user::valid                                       §11.113.2
    include user::unnecessary                                 §11.113.1
    include gnome-screensaver::stig                           §11.38.1
    include shell::stig                                       §11.94.4
    include pam::rhosts                                       §11.74.4
    include at::stig                                          §11.10.1
    include kdump::no                                         §??
    include network::stig                                     §11.66.12
    include ftp::no                                           §11.34.1
    include pki::ca_certs::system_nss                         §11.76.1
    include ldap::stig                                        §11.53.1
    include disable_ctrlaltdel                                §11.28
    include snmp::no                                          §11.98.1
    include network::no_bluetooth                             §??
}

class example_org_workstation
    include automount                                         §11.17
    class { 'gdm::logo':                                      §11.36.1
        source => 'puppet:///gdm/logo/example-org',
    }
    automount::mount { 'apps': from => 'example-data:/vol/apps' }   §11.42.4
    class { 'grub::password':                                 §11.40.3
        md5_password => 'ddce269a1e3d054cae349621c198dd52',
    }
}
```

## 11.4 Adobe Flash Player

Some users may require the Adobe Flash Player. Getting this to work for them is a challenge because Linux is not well supported by Adobe these days: For 64-bit support, as of March 2013, there have been two attempts at an x86_64 Flash plugin from Adobe, and neither was supported by security updates.[1] And Adobe is phasing out even 32-bit Linux support.

[1] There have been 81 vulnerabilities in Flash in the last year, 76 of which were critical (source: `http://www.cvedetails.com/vulnerability-list/vendor_id-53/product_id-6761/Adobe-Flash-Player.html`), so security updates are a must.

The flash-plugin package is in the Supplementary RHN channel, so any host that needs Flash must be subscribed to that channel, or the package will not be visible on the host.

```
class adobe_flash {

    case $::osfamily {
        'RedHat': {
```

Being from Red Hat, the flash-plugin package takes care of its own wrapping, if all the packages it needs are installed. So we needn't actually wrap the plugin ourselves, just get the prerequisites in place.

```
            include mozilla::wrap_32bit::prerequisites
```

§11.65.1

The 64-bit Flash plugin can get in the way, because these days yum detects when a package is installed twice, once each for the i686 and x86_64 architectures, and refuses to upgrade only one architecture-specific package of the pair and leave the other out of date; but Red Hat has stopped releasing new 64-bit flash-plugin packages.

```
            package { 'flash-plugin.x86_64':
                ensure => absent,
            }

            package { 'flash-plugin.i686':
                ensure => present,
                require => Class[Mozilla::Wrap_32bit::Prerequisites],
            }
        }
        'Darwin': {
            warning 'adobe_flash unimplemented on Macs'
        }
    }
}
```

## 11.5 Apple Filing Protocol

### Turn off AFP server

```
class afp::server::no {
    include "afp::server::no::${::osfamily}"
}
```

**Turn off AFP server on Macs**

```
class afp::server::no::darwin {
```

Disable file sharing via AFP. auto: OSX8-00-00140

```
    service { 'com.apple.AppleFileServer':
        enable => false,
        ensure => stopped,
    }
}
```

**Turn off AFP server on Red Hat** (Red Hat does not include an AFP server. This class is here just so you can include afp::server::no on any host without any trouble.)

```
class afp::server::no::redhat {
}
```

## 11.6 Host-based intrusion detection with AIDE

Install and configure the Advanced Intrusion Detection Environment (AIDE) host-based intrusion detection system (IDS) to check system files against a list of cryptographic hashes (a baseline) created at install time. (See §?? for baseline creation and update procedures.)

auto: DCSW-1
auto: GEN000140
auto: GEN006480
auto: DCSW-1
auto: GEN000140-2

For DBMSes included with RHEL, maintain the baseline for database software and configuration files along with that of the operating system files. (See also §11.86.1.)

auto: DCSW-1
auto: DG0021

Document setuid and setgid files, by including them in the baseline of system files.

auto: ECPA-1
auto: GEN002380
auto: GEN002440

Notify admins of possible intrusions via syslog. Remote logging ensures timely notification; for details, see §11.55.1.

auto: ECAT-1
auto: GEN006560

Check for rootkits. The AIDE tool does this adequately for our needs.

auto: DCSL-1
auto: GEN008380

```
class aide {
    include "aide::${::osfamily}"
}
```

We should watch setuid executables on the system. aide is the tool to do this. But we haven't implemented it on the Mac yet.

OSX8-00-01145

```
class aide::darwin {
    warning 'unimplemented for Macs'
}
```

### 11.6.1 AIDE configuration for Red Hat

```
class aide::redhat {
    package { "aide":
        ensure => present,
    }
```

Install the prescribed configuration for AIDE, causing it to baseline device files, extended access control lists (ACLs), and extended attributes, using FIPS 140-2 approved cryptographic hashing algorithms.

auto: DCSW-1
auto: ECAT-1
auto: ECSC-1
auto: GEN000140
auto: GEN006570
auto: GEN006571
auto: GEN006575

Configure AIDE to create and monitor a baseline of database “software libraries, related applications and configuration files.”

auto: DCSL-1
auto: DG0050

```
        file { "/etc/aide.conf":
            owner => root, group => 0, mode => 0600,
            source => "puppet:///modules/aide/aide.conf",
        }
```

Warn if the aide binary changes.

```
        file { "/usr/sbin/aide":
            audit => all,
        }
```

Check for unauthorized changes to system files, including setuid files and setgid files, every week.

auto: DCSL-1
auto: ECAR-2
auto: GEN000220
auto: GEN002400
auto: GEN002460

```
    cron { aide:
        command => "/usr/sbin/aide --check",
        user => root,
        hour => 2,
        minute => 2,
        weekday => 0,
    }
```

Make sure aide's logs are rotated.

```
    augeas { "aide_weekly":
        context => "/files/etc/logrotate.d/aide/rule",
        changes => "set schedule weekly",
    }
```

Since aide is run by logrotate, make sure logrotate is working.

Use mode 0700 for the daily log rotation script, as required. auto: ECLP-1

```
    file { "/etc/cron.daily/logrotate":
        owner => root, group => 0, mode => 0700,
        source => "puppet:///modules/aide/logrotate",
    }
```

auto: GEN003100
auto: GEN003120
auto: GEN003140

Install the baseline backup script for use by administrators. See §??.

```
    file { "/usr/sbin/backup_baseline.sh":
        owner => root, group => 0, mode => 0755,
        source => "puppet:///modules/aide/backup_baseline.sh",
    }
}
```

# 11.7 AMD graphics card support

(AMD bought ATI several years ago, which may be a more familiar company name to you.)

## 11.7.1 Proprietary driver

TODO: Make some code to actually install the driver, à la 11.72.1.

```
class amd_graphics::proprietary($installer_dir) {
    sudo::auditable::command_alias { 'AMD_DRIVERS':
        type => 'exec',
        commands => [
            '${installer_dir}/amd-*.run',
            ],
    }
}
```

§11.104.3

# 11.8 Apache httpd

Configure the Apache web server in accordance with the Apache STIG [4] [5].

Most of the requirements involve the Apache configuration. We don't have enough distinct web servers that imposing the configuration items by means of a Puppet policy would be expedient. So the STIG requirements are noted in each web server's configuration; all those configurations are version-tracked.

Requirements best fulfilled by Puppet policy are written and documented here.

```
class apache($production=true) {
    package { "httpd":
        ensure => present,
    }
    service { 'httpd':
        enable => true,
        ensure => running,
        require => Package['httpd'],
    }
    include apache::fips                                               §11.8.2
    case $production {
        'false', false: { include apache::stig::nonproduction }
        default: { include apache::stig::production }
    }
}
```

### 11.8.1 Apache configuration

This submodule configures Apache by editing its configuration files with Augeas. The reason for doing this is to make it easier to integrate stock Red Hat httpd configuration, configuration required for STIG compliance, and configuration for particular kinds of websites, as all three change. The most readily apparent simpler scheme would be to construct template files for each kind of website, and control changes to them separately from this Configuration Management for IT Systems Example Policy. But then a process for doing so would have to be worked out (whether formally or not); and tweaking settings for compliance rather than replacing them is something already widely done here.

So we edit Apache configuration files using Augeas. The Httpd Augeas lens defines directives and contexts.

*Contexts* correspond to `<Foo>` ... `</Foo>` sort of constructs in the configuration file. They can contain directives.

*Directives* correspond to `Foo bar baz` sort of constructs in the configuration file, like `Options None` or `Listen 80`.

We define here two resource types to manage these two things. In the case where a directive is inside a context, the defined resource types include dependencies among themselves so that the context must exist before the directive can be set.

#### Resource names using context names

The names of defined resources of these two types are written in a peculiar format: *config_file*:*context_name_1*:*context_name_2*:... where *config_file* is the full path name of an Apache configuration file; *context_name_N* are *names* of contexts inside the file (explained below). The rest (...) is specific to the resource type, *q.v.*.

Context names are used to hook up dependencies among directives and contexts, so that if you want a construct of the form

```
<Foo bletch>
  Bar baz
</Foo>
```

and you make two resources

```
apache::config::context { '/etc/bla/httpd.conf:the_foo':
    context_in_file => '',
    type => 'Foo',
    arguments => ['bletch'],
}
apache::config::directive { '/etc/bla/httpd.conf:the_foo:Bar':
    context_in_file => 'Foo[arg="bletch"]',
    arguments => ['baz'],
}
```

the directive resource will depend on the context resource without your saying anything except to connect them by the *context name* `the_foo`. You make up the name; the important thing is it's the same between the resources.

FIXME: There is a great deal of semantic overlap between context names, which are identifiers that are made up, and contexts inside the file, which have special characters but denote a place in the file exactly.

### context_in_file

The value of a `context_in_file` parameter is a piece of an Augeas context argument. It is tacked onto the end of the path in Augeas denoted by the configuration filename (`/files/`*`config_file`* where *config_file* is gotten from the resource name as above) to denote the place in the Augeas tree where a directive or context should be created or controlled. If this context should be in the toplevel of the file, not inside another angle-bracket-tag sort of thing, `context_in_file`'s value should be the empty string.

```
class apache::config($max_request_body=4194304, $nss_database_dir) {
    class { 'apache::config::httpd_conf':                          §11.8.1
        max_request_body => $max_request_body,
    }
    class { 'apache::config::nss_conf':                            §11.8.1
        nss_database_dir => $nss_database_dir,
    }
    file { '/etc/httpd/common':
        ensure => directory,
        owner => root, group => 0, mode => 0600,
        source => 'puppet:///modules/apache/common',
        recurse => true,
        recurselimit => 1,
    }
    # normally this would be a require, but we had to pass some parameters
    Class['apache::config::nss_conf'] -> Class['apache::config']
}
define apache::config::nss_site($content) {
    include apache
    $nss_sites_dir = $apache::config::nss_conf::nss_sites_dir
    file { "${nss_sites_dir}/${name}.conf":
        owner => root, group => 0, mode => 0600,
        content => $content,
        require => [
            Class['apache::config'],
            File['/etc/httpd/nss-site.d'],
            ],
        notify => Service['httpd'],
    }
}
```

§11.8

### Contexts in Apache configuration

The Httpd Augeas lens defines directives and contexts; contexts correspond to `<Foo>` ... `</Foo>` sort of constructs in the configuration file. They can contain directives.

The name of resources of this type begins as discussed above, and ends with a chosen context name, which must be an identifier (starts with a letter, no spaces, no special characters, just letters, numbers and underscores). Directive resources whose directives are inside this context, and context resources whose contexts are inside this context, will include this context name in their resource names, so it should be short.

`context_in_file` is as discussed above.

`type` is what kind of angle-bracket-tag sort of thing this context should be. Common values for `type` are `'Directory'`, `'LimitExcept'`, `'Location'`, and the like.

`arguments` is an array of arguments that are written inside the angle-brackets. For example, for a `Directory` context, the arguments might be `['/var/www']`. The result written in the configuration file would look like

```
<Directory /var/www>
</Directory>
```

* * *

```
define apache::config::context(
    $context_in_file, $type, $arguments) {

        include apache                                              §11.8
        $pieces = split($name, ’:’)
        $config_file = $pieces[0]
        $parent_context_name = inline_template(’<%=@pieces[1..-2].join(":")-%>’)
        $this_context_name = $pieces[-1]
    augeas { "add ${name} subcontext ${type} nicknamed ${this_context_name}":
        incl => $config_file,
        lens => ’Httpd.lns’,
        context => $context_in_file ? {
            ""      => "/files/${config_file}",
            default => "/files/${config_file}/${context_in_file}",
        },
        changes => inline_template("
clear <%=@type-%>[999]
<% @arguments.each_with_index do |a, zi| %>
set <%=@type-%>[last()]/arg[<%=zi+1-%>] ’<%=a-%>’
<% end %>
"),
        onlyif => "match ${type}[arg=’${arguments[0]}’] size == 0",
        require => $parent_context_name ? {
            ’’ => [],
            default => Apache::Config::Context[
                "${config_file}:${parent_context_name}"],
        },
        notify => Service[’httpd’],
    }
}
```

### Directives in Apache configuration

The name of resources of this type begins as discussed above, and ends with the name of a directive, like `Options` or `NSSUserName` or `Listen`.

The `context_in_file` parameter is as discussed above.

`arguments` is an array of arguments that are written after the name of the directive; for example, if you wanted a directive that says `Deny from all`, `arguments` should be set to `[’from’, ’all’]`.

```
define apache::config::directive(
    $context_in_file, $arguments) {

    include apache                                                  §11.8
    $pieces = split($name, ’:’)
    $config_file = $pieces[0]
    $directive = $pieces[-1]
    $context_name = inline_template(’<%=@pieces[1..-2].join(":")-%>’)
    $context_for_d = $context_in_file ? {
        ’’      => "/files/${config_file}",
        default => "/files/${config_file}/${context_in_file}",
    }
    Augeas {
        incl => $config_file,
        lens => ’Httpd.lns’,
        notify => Service[’httpd’]
    }
    $replace_args = inline_template("
rm arg
<% @arguments.each_with_index do |a, zi| %>
set arg[<%=zi+1-%>] ’<%=a-%>’
<% end %>
")
    augeas { "add ${name}":
context => $context_for_d,
changes => "set directive[999] ’${directive}’",
onlyif => "match directive[.=’${directive}’] size == 0",
        require => $context_name ? {
            ’’ => [],
            default    => Apache::Config::Context[
                "${config_file}:${context_name}"],
        },
    } ->
    augeas { "correct ${name}":
context => "${context_for_d}/directive[.=’${directive}’]",
changes => $replace_args,
    }
}
```

### httpd.conf

Assumption: we are starting with a stock RHEL6 httpd configuration.

Parameter max_request_body is given in bytes. If a website supports file uploads via POST requests, the max_request_size must be set a few kilobytes larger than the largest file that should be uploadable.

```
class apache::config::httpd_conf($max_request_body=4194304) {
    if $::osfamily != ’RedHat’ or $operatingsystemrelease !˜ /ˆ6\./ {
        unimplemented()
    }

    include apache
```

```
$abbr_ehchc  = '/etc/httpd/conf/httpd.conf'
$abbr_fehchc = "/files/${abbr_ehchc}"

Augeas {
    incl => $abbr_ehchc,
    lens => 'Httpd.lns',
    notify => Service['httpd'],
}
```

Ensure a directive is in place and set to a given value, in the toplevel of httpd.conf.

```
define toplevel_directive($arguments) {
    $abbr_ehchc = $apache::config::httpd_conf::abbr_ehchc
    directive { "${abbr_ehchc}:${name}":
        context_in_file => "",
        arguments => $arguments,
    }
}
```

Ensure a directive is in place and set to a given value, in `<Directory />` in httpd.conf.

```
define root_dir_directive($arguments) {
    $abbr_ehchc = $apache::config::httpd_conf::abbr_ehchc
    directive { "${abbr_ehchc}:root:${name}":
        context_in_file => "Directory[arg='/']",
        arguments => $arguments,
    }
}
```

Ensure a directive is in place and set to a given value, in `<Directory /var/www>` in httpd.conf.

```
define var_www_dir_directive($arguments) {
    $abbr_ehchc = $apache::config::httpd_conf::abbr_ehchc
    directive {
        "${abbr_ehchc}:varwww:${name}":
            context_in_file => "Directory[arg=’/var/www’]",
            arguments => $arguments;
    }
}

context { "${abbr_ehchc}:root":
    context_in_file => ’’,
    type => ’Directory’,
    arguments => [’/’],
}
context { "${abbr_ehchc}:varwww":
    context_in_file => ’’,
    type => ’Directory’,
    arguments => [’/var/www’],
}
# augeas { ’httpd.conf root directory add’:
#     context => $abbr_fehchc,
#     changes => [
#         ’clear Directory[999]’,
#         ’set Directory[999]/arg "/"’,
#         ],
#     onlyif => ’match Directory[arg="/"] size == 0’,
# }
# augeas { ’httpd.conf varwww directory add’:
#     context => $abbr_fehchc,
#     changes => [
#         ’clear Directory[998]’,
#         ’set Directory[998]/arg "/var/www"’,
#         ],
#     onlyif => ’match Directory[arg="/var/www"] size == 0’,
# }

toplevel_directive {
```

Avoid warnings about not being able to determine ServerName. This will be overridden in the virtual site configuration anyway.

```
    ’ServerName’: arguments => [$::fqdn];
```

Don’t tell visitors what OS we are running.

```
    ’ServerTokens’: arguments => [’ProductOnly’];
```

auto: WA000-WWA020 A22

```
    ’Timeout’: arguments => [120];
```

auto: WA000-WWA022 A22

```
    ’KeepAlive’: arguments => [’on’];
```

Set MaxKeepAliveRequests to 100 “or greater.”

auto: WG110 A22

```
    ’MaxKeepAliveRequests’: arguments => [100];
```

auto: WA000-WWA024 A22

```
        'KeepAliveTimeout': arguments => [15];
```

Limit request body size. The actual limit is not specified by the STIG. auto: WA000-WWA060 A22

```
        'LimitRequestBody': arguments => [$max_request_body];
```

Limit number of HTTP request header fields. auto: WA000-WWA062 A22

```
        'LimitRequestFields': arguments => [50];
```

Limit size of each HTTP request header field, to "8190 or other approved value." auto: WA000-WWA064 A22

```
        'LimitRequestFieldSize': arguments => [8190];
```

Limit HTTP request line length, to "8190 or other approved value." auto: WA000-WWA066 A22

```
        'LimitRequestLine': arguments => [8190];
    }
```

Remove toplevel Listen directive: leave it to per-website configuration to Listen.

```
    augeas { "httpd.conf remove Listen":
        context => $abbr_fehchc,
        changes => 'rm directive[.="Listen"]',
    }
```

Minimize active software modules. auto: WA00500 A22

```
    define remove_module_load() {
        $abbr_fehchc = $apache::config::httpd_conf::abbr_fehchc
        $abbr_ehchc = $apache::config::httpd_conf::abbr_ehchc
        augeas { "httpd.conf remove module ${name}":
            context => $abbr_fehchc,
            changes => "rm \
                directive[.='LoadModule' and arg[1]='${name}']",
        }
    }

    remove_module_load { [
        'auth_digest_module',
        'authn_anon_module',
        'authn_dbm_module',
        'authz_owner_module',
        'authz_dbm_module',
        'include_module',
        'ext_filter_module',
        'expires_module',
        'headers_module',
        'usertrack_module',
```

Disable status module. auto: WA00510 A22

```
        'status_module',
        'info_module',
```

Turn off all we can of DAV. See http://svn.haxx.se/users/archive-2004-12/0709.shtml.

```
        'dav_fs_module',
        'speling_module',
```

Disable user-specific directories. auto: WA00525 A22

```
            'userdir_module',
```

These may break applications that use Apache as a proxy for a web application container that runs its own web server. We would need reverse proxying for Plone—but we don't tend to use Plone anymore. auto: WA00520 A22

```
            'proxy_module',
            'proxy_balancer_module',
            'proxy_ftp_module',
            'proxy_http_module',
            'proxy_ajp_module',
            'proxy_connect_module',
            'cache_module',
            'suexec_module',
            'disk_cache_module',
            'version_module',
            ]: }
```

WebDAV is supposed to be disabled, but Subversion requires it. Autoindexes are supposed to be disabled, but SBU requires them. WA00505 A22 WG170 A22

Disable the FollowSymLinks option; Options None does this. auto: WA000-WWA052 A22

```
    toplevel_directive { 'Options': arguments => ['None'] }
```

Disable all options at the OS root. auto: WA00545 A22

```
    root_dir_directive { 'Options': arguments => ['None'] }
```

Disable access configuration override at the OS root. auto: WA00547 A22

```
    root_dir_directive { 'AllowOverride': arguments => ['None'] }
```

Deny access to the OS root. (Access is allowed by exception in other parts of the web server configuration.) auto: WA00540 A22

```
    root_dir_directive { 'Order': arguments => ['deny,allow'] } ->
    root_dir_directive { 'Deny': arguments => ['from', 'all'] }
```

Disable TRACE method. auto: WA00550 A22

```
    toplevel_directive { 'TraceEnable': arguments => ['off'] }
```

Avoid executing things using server-side includes (SSIs). We don't use SSIs so they are just turned off altogether (see include_module above). auto: WA000-WWA054 A22

Disable MultiViews. auto: WA000-WWA056 A22

Disable auto-indexing by default. auto: WA000-WWA058 A22

```
    var_www_dir_directive { 'Options': arguments => ['None'] }
```

Limit HTTP request methods. auto: WA00565 A22

Other methods than these might be allowed in certain places inside the website.

```
    context {
        "${abbr_ehchc}:varwww:limitexcept":
            context_in_file => "Directory[arg=’/var/www’]",
            type => ’LimitExcept’,
            arguments => [’GET’, ’POST’, ’OPTIONS’];
    }
    directive { "${abbr_ehchc}:varwww:limitexcept:Deny":
        arguments => [’from’, ’all’],
        context_in_file => "Directory[arg=’/var/www’]/LimitExcept",
    }


    toplevel_directive { ’ErrorLog’: arguments => [’syslog’] }
```

Use the “correct format” for logs. auto: WA00612 A22

```
    augeas { ’change log format at toplevel in httpd.conf’:
        context => "${abbr_fehchc}/directive[\
            .=’LogFormat’ and arg[2]=’combined’]",
        changes => "set arg[1] \
’\"%a %A %h %H %l %m %s %t %u %U \\\"%{Referer}i\\\" \"’",
    }

    toplevel_directive { ’ServerSignature’: arguments => [’Email’] }
```

The icons directory doesn’t need any options.

```
    augeas { "httpd.conf icons remove Options":
        context => "${abbr_fehchc}/Directory[arg=’/var/www/icons’]",
        changes => ’rm directive[.="Options"]’,
    }
}
class apache::config::nss_conf($nss_database_dir) {
    include apache                                                      §11.8
    if $::osfamily != ’RedHat’ or $operatingsystemrelease !˜ /ˆ6\./ {
        unimplemented()
    }

    $nss_sites_dir = ’/etc/httpd/nss-site.d’
    $rel_nss_sites_dir = ’nss-site.d’
    $abbr_ehcnc  = ’/etc/httpd/conf.d/nss.conf’
    $abbr_fehcnc = "/files/${abbr_ehcnc}"

    Augeas {
        incl => $abbr_ehcnc,
        lens => ’Httpd.lns’,
        notify => Service[’httpd’],
    }
```

Ensure a directive is in place and set to a given value, in the toplevel of nss.conf.

```
    define toplevel_directive($arguments) {
        $abbr_ehcnc = $apache::config::nss_conf::abbr_ehcnc
        directive { "${abbr_ehcnc}:${name}":
            context_in_file => "",
            arguments => $arguments,
        }
    }

    toplevel_directive {
```

Listen on a specific IP address, so that if interfaces are added in the future we will not accidentally serve web pages on them by default. auto: WA00555 A22

```
        'Listen':
            arguments => ["${::ipaddress}:443"];
        'NSSPassPhraseDialog':
            arguments => ["file:${nss_database_dir}/pwfile"];
    }
```

We are going to move the virtual host section to its own config file.

```
    augeas { 'remove stock virtualhost from nss.conf':
        incl => $abbr_ehcnc,
        lens => 'Httpd.lns',
        context => $abbr_fehcnc,
        changes => 'rm VirtualHost[arg="_default_:8443"]',
    }
    file { $nss_sites_dir:
        ensure => directory,
        owner => root, group => 0, mode => 0600,
    } ->
    toplevel_directive {
        'Include': arguments => ["${rel_nss_sites_dir}/*.conf"];
    }
}
```

### 11.8.2 FIPS-required configuration

Configure Apache httpd in a manner compliant with FIPS 140-2. We do this using mod_nss instead of mod_ssl; see 11.8.3 for more details.

```
class apache::fips {
    include apache                                                    §11.8
    package {
        "mod_nss":
            ensure => present,
            notify => Service['httpd'];
        "mod_ssl":
            ensure => absent,
            notify => Service['httpd'];
    }
```

The NSS security policy [16] may require that the NSS cryptographic module be auditable. To make it so, we must tell it to log what it does, via an environment variable.

I hope it does not require this because the thing is way too verbose - on the order of fifteen or twenty lines of log per HTTPS request. Turning it off for now. To turn back on, change the line below from “set $nea 0” to “set $nea 1”.

```
    $nea = "NSS_ENABLE_AUDIT"
    augeas { "httpd_nss_audit":
        context => "/files/etc/sysconfig/httpd",
        changes => [
            "rm #comment[.=~regexp('$nea:.*')]",
            "set #comment[last()+1] \
             '$nea: maybe necessary for FIPS compliance'",
            "rm $nea",
            "set $nea 0",
            # make the export exist in the tree but have no value
            "clear $nea/export",
        ],
```

Don’t do this before httpd is installed, otherwise the stock /etc/sysconfig/httpd will be installed as a .rpmnew.

```
        require => Package['httpd'],
        notify => Service['httpd'],
    }
}
```

### 11.8.3 On the use of mod_nss

The usual way of configuring SSL/TLS on an Apache server is to use mod_ssl, which uses OpenSSL libraries to do the cryptographic work.

As of 2 May 2011, when using mod_ssl on a FIPS-enabled host, httpd 2.2.15 will not start, citing failure to generate a 512-bit temporary key. An SSL+FIPS patch exists (http://people.apache.org/~wrowe/ssl-fips-2.2.patch). Judging by a reading of this patch, the failure to generate a temporary key is not because of a lack of available entropy for the pseudo-random number generator, as the documentation says, but perhaps because this is the first cryptographic thing that httpd is trying to do, and it hasn’t called OpenSSL’s FIPS_mode_set function first, so OpenSSL fails to do anything. The patch would fix this, but it is not in the vendor-supported httpd package.

Red Hat does provide mod_nss, which uses the NSS libraries to do cryptographic work instead of OpenSSL. FIPS-accredited versions of NSS exist. I found a Red Hat bug from 2008 where someone was talking about having used the NSSFIPS directive in the Apache configuration. So it would appear that this a more vendor-supported path to FIPS-compliant TLS under Apache httpd.

(The quickest and most familiar route would be to turn off OS-wide FIPS mode (see §11.33); but the UNIX SRG requires that to be on.)

### 11.8.4 Private key security under OpenSSL and NSS

Usually, under mod_ssl, private keys are in files owned by root, and accessible only by root; the httpd process starts as root, reads the files during startup, then drops root and becomes the `apache` user for the rest of its life. If someone were to exploit a vulnerability in `httpd`, they could run arbitrary code as the `apache` user; but `apache` cannot read the private key files. This makes me feel good.

Under `mod_nss`, private keys are in the NSS database, in an encrypted file. The database's use is internal to NSS, so it must be assumed that NSS could access it at any time; there are no privileges that can be dropped. So the NSS database files must be owned not by root, but by `apache`. That means our hypothetical attacker can read them. This makes me feel nervous.

But the private keys are encrypted and can only be decrypted with a password. Perhaps the attacker could read the password out of `httpd`'s memory? But the documentation about NSS written in support of its FIPS certification[2] says, "Passwords are automatically zeroized by the NSS cryptographic module immediately after use." So that can't happen.

In either case, the unencrypted private key is in `httpd`'s memory while it's running, anyway.

As the same document and the NSS security policy [16] both say, "Since password-based encryption such as PKCS #5 is not FIPS Approved, the private and secret keys in the private key database are considered in plaintext form by FIPS 140-2 (see FIPS 140-2 Section 4.7 and FIPS 140-2 IG 7.1);" however, password-based encryption is not considered in plaintext form by attackers until after the application of many CPU-hours of work, so it is not without benefit.

### 11.8.5 Disable the web server

```
class apache::no {
    include "${name}::${::osfamily}"
}
    class apache::no::darwin {
```

Turn off "Web Sharing" on Macs—that is, the Apache httpd web server. auto: OSX8-00-01275

```
        service { 'org.apache.httpd':
            ensure => stopped,
            enable => false,
        }
    }
    class apache::no::redhat {
        service { 'httpd':
            ensure => stopped,
            enable => false,
        }
    }
```

### 11.8.6 STIG-required, Puppetable configuration

```
class apache::stig::common {
        include apache
```

§11.8

Secure the web server PID file. auto: WA00530 A22

[2] https://wiki.mozilla.org/VE_07KeyMgmt

```
        file { "/var/run/httpd/httpd.pid":
            mode => 0644,
        }
```

Fix permissions of Web server system files. auto: WG300 A22

Since we use Apache as shipped by Red Hat, and its files are not under `/usr/local`, but in their proper places throughout the filesystem, not all the permission fixes are here. Also, we don't have a "web user": as the vendor recommends, we start Apache httpd as root, and then it drops all the privileges it doesn't need and becomes the apache user. This means the configuration files, private keys, etc. can be owned by root.

```
        file {
            "/etc/httpd":
                owner => root, group => 0, mode => 0600;
        }
```

`bin` permissions are taken care of by the packaging system, and verified in §11.86.1.

`logs` permissions are covered under §11.56.6 and below.

`htdocs` permissions vary by web server. In the particular case of the AFSEO SBU website, see under §11.88.4.

Prevent Web server administration or file uploads over Telnet, FTP, or rsh. auto: WG230 A22

```
        include telnet::no                                          §11.107.1
        include ftp::no                                             §11.34.1
        include rsh::no                                             §11.87.1
```

Make sure root owns the web server log files. Permissions are taken care of by §11.56.6. auto: WG250 A22

```
        file { "/var/log/httpd":
            owner => root, group => 0,
            recurse => true, recurselimit => 2,
        }
```

Get rid of symbolic links which are installed by default. auto: WG360 A22

```
        file { "/var/www/icons/poweredby.png":
            ensure => absent,
        }
    }
    class apache::stig::nonproduction {
        include apache::stig::common                                §11.8.6
    }
```

### Apache STIG compliance on production web servers

```
class apache::stig::production {

        include apache::stig::common                                §11.8.6
```

Remove compilers from production web servers. auto: WG080 A22

(We do not detect here whether a web server is "production.")

```
    package {
        [
            'gcc',
            'gcc-c++',
            'gcc-gfortran',
            'libtool',
            'systemtap',
```

No one should be building modules on the web server.

```
            'httpd-devel',
        ]:
            ensure => absent,
    }
```

Remove all web server documentation, sample code, example applications and tutorials from production web servers. auto: WG385 A22

As above, we do not detect a production web server here. Since this is the only Category I requirement in this STIG, we'll make sure that it works across `httpd` versions, rather than being a piece of tidy policy.

```
    exec { "rm_httpd_docs":
        command => "/bin/rm -rfv /usr/share/doc/httpd-[0-9]*",
        onlyif  => "/bin/ls      /usr/share/doc/httpd-[0-9]*",
        logoutput => true,
    }
    file {
        '/usr/share/man/man8/apachectl.8.gz':
            ensure => absent;
        '/usr/share/man/man8/htcacheclean.8.gz':
            ensure => absent;
        '/usr/share/man/man8/httpd.8.gz':
            ensure => absent;
        '/usr/share/man/man8/rotatelogs.8.gz':
            ensure => absent;
        '/usr/share/man/man8/suexec.8.gz':
            ensure => absent;
    }
    exec { "rm_mod_nss_docs":
        command => "/bin/rm -rfv /usr/share/doc/mod_nss-[0-9]*",
        onlyif  => "/bin/ls      /usr/share/doc/mod_nss-[0-9]*",
        logoutput => true,
    }
    package {
        "httpd-manual": ensure => absent;
```

The debuginfo package may contain the source, which is the ultimate documentation.

```
        "httpd-debuginfo": ensure => absent;
    }
}
```

## 11.9 Application firewall

```
class app_firewall {
    include "app_firewall::${::osfamily}"
}
   class app_firewall::darwin {
       $version_underscores = regsubst(
           $::macosx_productversion_major,
           '\D', '_', 'G')
       $klassname = "${::osfamily}_${version_underscores}"
       include "app_firewall::${klassname}"
   }
   class app_firewall::darwin_10_6 {}
   class app_firewall::darwin_10_9 {
       $sffw = '/usr/libexec/ApplicationFirewall/socketfilterfw'
       exec { 'turn on application firewall':
           command => "${sffw} --setglobalstate on",
           unless => "${sffw} --getglobalstate | grep enabled",
       }
   }
   class app_firewall::redhat {}
```

## 11.10 The at subsystem

### 11.10.1 STIG-required configuration for the at subsystem

```
class at::stig {
    case $::osfamily {
        'redhat': { include at::stig::redhat }
        'darwin': { include at::stig::darwin }
        default:  { unimplemented() }
    }
}
```

### 11.10.2 Guidance for admins about the at subsystem

Never run a group-writable or world-writable program with `at`. Never run a program using `at` which is in or anywhere under a world-writable directory (such as `/tmp`). Don't change the umask in an `at` job.

admins do GEN003360
admins do GEN003380
admins do GEN003440 M6
admins do GEN003440

### 11.10.3 STIG-required at subsystem configuration for Mac OS X

```
class at::stig::darwin {
    file {
```

Control ownership and permissions of `at.deny`.

auto: ECLP-1
auto: GEN003480 M6

```
           '/var/at/at.deny':
               owner => root, group => 0;
       }
   }
```

### 11.10.4 STIG-required at subsystem configuration for RHEL

Under RHEL and derivatives, only allow root to do at jobs.

```
class at::stig::redhat {
    file {
```

Remove `at.deny`, in order to specify access by who is allowed, not by who is denied.

auto: ECLP-1
auto: GEN003252
auto: GEN003300
auto: GEN003480
auto: GEN003490

```
        "/etc/at.deny":
            ensure => absent;
```

Control contents and permissions of `at.allow`.

auto: ECLP-1
auto: ECPA-1
auto: GEN003280
auto: GEN003320
auto: GEN003460
auto: GEN003470
auto: GEN003340

```
        "/etc/at.allow":
            owner => root, group => 0, mode => 0600,
            content => "root\n";
```

Control permissions of “the ‘at’ directory.”

In the default install, this is owned by `daemon`, group `daemon`, so this change might break `at`.

auto: ECLP-1
auto: GEN003400
auto: GEN003420
auto: GEN003430

```
        "/var/spool/at":
            owner => root, group => 0, mode => 0700;
    }

    no_ext_acl {
```

Remove extended ACL on `at.allow`.

auto: ECLP-1
auto: GEN003245

```
        "/etc/at.allow":;
```

Remove extended ACL on `at.deny`.

auto: ECLP-1
auto: GEN003255

```
        "/etc/at.deny":;
```

Remove extended ACLs in “the ‘at’ directory.”

auto: ECLP-1
auto: GEN003410

```
        "/var/spool/at": recurse => true;
    }
}
```

## 11.11 Audio

Configure audio support.

### 11.11.1 Disable audio

```
class audio::no {
    include "audio::no::${::osfamily}"
}
```

#### Disable audio on Macs

```
class audio::no::darwin {
```

Disable audio support where necessary to “protect the organization’s privacy.”

auto: ECSC-1
auto: OSX00070 M6
auto: OSX8-00-01225

```
        $exts = '/System/Library/Extensions'
        file {
            "${exts}/AppleUSBAudio.kext":
                ensure => absent,
                force => true;
            "${exts}/IOAudioFamily.kext":
                ensure => absent,
                force => true;
        }
    }
```

### 11.11.2 Turn down audio input levels

"If audio output is required for the mission ... ensure the input volume is 0." —Apple OS X 10.8 STIG PDI OSX8-00-01225

```
    class audio::zero_input_volume {
        include "audio::zero_input_volume::${::osfamily}"
    }
    class audio::zero_input_volume::darwin {
        exec { 'turn down input volume':
            command => 'osascript -e "set volume input volume 0"',
            unless => 'osascript -e "get volume settings" | \
                       grep "\\<input volume:0\\>"',
            path => ['/bin', '/usr/bin'],
        }
    }
```

## 11.12 Auditing subsystem

Activate audit logging; configure it in a compliant fashion; and protect and retain audit logs. auto: ECAN-1 auto: ECRR-1

The sense in which we implement ECRR-1, Audit Record Retention, here in this section, is that retention includes making sure the logs are not overwritten, nor modified or deleted by unauthorized users. The narrower sense of retention, "moving audit trails from on-line to archival media," is handled by backing up the audit logs in the same way as the rest of the logs. See §11.55.1.

The SRG requires remote audit logging. It seems that audisp-remote can be used for remote audit logging, but it needs a Kerberos infrastructure first. So we do not yet have a remote audit server. We depend on log backups to preserve a remote audit record. GEN002870

The auditing rules installed in §11.12 fulfill Database STIG requirements. auto: ECAR-2 auto: DG0140

```
    class audit {
        include "audit::${::osfamily}"
        include audit::file_permissions
    }
```

§11.12.2

### 11.12.1 Auditing under Mac OS X

```
class audit::darwin {
    warning 'audit configuration unimplemented on darwin'

    service { 'com.apple.auditd':
        enable => true,
        ensure => running,
    }
}
```

#### Mac OS X audit log permissions

The name of this resource is the directory where audit log files are kept. By default this is /var/audit. This is a defined resource type and not a class so that permissions can be imposed on any audit log directory that may be configured, because the STIG check and fix texts dictate that permissions be checked and fixed on any directory (and files therein) listed in the audit configuration file, not just the usual place.

```
define audit::darwin::permissions() {
    $dir = $name
```

Fix owner and group of audit log files to root:wheel. Fix owner and group of audit log folder to root:wheel.

auto: OSX8-00-00210
auto: OSX8-00-00340
auto: OSX8-00-00355
auto: OSX8-00-00215
auto: OSX8-00-00365

```
    file { $dir:
        owner => root, group => wheel,
        recurse => true,
    }
```

We can't implement the permissions with the file resource type because the required permissions are different for the directory and the files inside it.

```
    Exec {
        path => ['/bin', '/usr/bin'],
    }
```

Fix permissions of audit log files.

auto: ECLP-1
auto: ECTP-1
auto: GEN002680 M6
auto: GEN002690 M6
auto: GEN002700 M6
auto: OSX8-00-00205
auto: OSX8-00-00335
auto: OSX8-00-00350

```
    exec { "chmod ${dir} files":
        command => "find ${dir} -mindepth 1 -print0 | \
        xargs -0 chmod 0440",
        onlyif  => "find ${dir} -mindepth 1 \\! -perm 0440 | \
        grep . >&/dev/null",
    }
```

Fix permissions of audit log directory.

auto: OSX8-00-00220
auto: OSX8-00-00370

```
    exec { "chmod ${dir} directory":
        command => "chmod 700 ${dir}",
        onlyif  => "stat -f '%Lp' ${dir} | grep -v ^700\$",
    }
```

Remove extended ACLs from audit log files.

auto: OSX8-00-00225
auto: OSX8-00-00345
auto: OSX8-00-00375

```
    no_ext_acl { $dir:
        recurse => true,
    }
}
```

### 11.12.2 File and directory permissions relating to auditing

```
class audit::file_permissions {
```

First, establish what *system audit logs* and *audit tool executables* are.

```
    $audit_data = $::osfamily ? {
        'darwin' => '/var/audit',
        'redhat' => '/var/log/audit',
        default  => unimplemented,
    }
    $audit_tools = $::osfamily ? {
```

This list of executables comes from the check content in the Mac OS X STIG.

```
        'darwin' => ['/usr/sbin/audit', '/usr/sbin/auditd',
                     '/usr/sbin/auditreduce',
                     '/usr/sbin/praudit'],
```

This list of executables comes from `rpm -ql audit`.

```
        'redhat' => ['/sbin/audispd', '/sbin/auditctl',
                     '/sbin/auditd', '/sbin/aureport',
                     '/sbin/ausearch', '/sbin/autrace',
                     '/usr/bin/aulast', '/usr/bin/aulastlog',
                     '/usr/bin/ausyscall'],
        default  => unimplemented,
    }
```

Let only admins access audit data.

auto: ECRR-1

```
    case $::osfamily {
        'RedHat': {
```

Ensure proper ownership and permissions on audit logs.

auto: ECLP-1
auto: ECTP-1
auto: GEN002680
auto: GEN002690
auto: GEN002700

```
            file { $audit_data:
                recurse => inf,
                owner => root, group => 0, mode => 0600,
            }
```

Remove extended ACLs on audit logs.

auto: ECTP-1
auto: GEN002710

```
            no_ext_acl { $audit_data: recurse => true }
        }
        'Darwin': {
            audit::darwin::permissions { $audit_data: }
        }
    }
```

§11.12.1

auto: ECTP-1
auto: GEN002710 M6
auto: ECLP-1
auto: GEN002715
auto: GEN002716
auto: GEN002717
auto: OSX8-00-00400
auto: OSX8-00-00405
auto: OSX8-00-00410
auto: OSX8-00-00415

Ensure proper ownership and permissions on audit tool executables.

Make sure `praudit` is the right binary. Make sure `auditreduce` is the right binary. Make sure `audit` is the right binary. Make sure `auditd` is the right binary.

These will be correct by default (RHEL5, RHEL6), so this is defense in depth.

The OSX Mountain Lion STIG lists the exact checksums which the files must match, and this just makes sure the files don't change against the first time they are observed. But the checksums in the STIG are not the correct ones for Mavericks nor Snow Leopard anyway.

```
    file { $audit_tools:
        owner => root, group => 0,
        audit => all,
    }
```

Remove extended access control lists (ACLs) on audit tool executables.

auto: ECLP-1
auto: GEN002718 M6

```
    no_ext_acl { $audit_tools: }
}
```

auto: ECLP-1
auto: GEN002718

```
class audit::redhat {
```

Install the auditing software.

auto: ECAR-2
auto: GEN002660

```
    package { "audit":
        ensure => present,
    }
```

Rotate audit logs daily.

auto: ECSC-1
auto: GEN002860

The example provided with auditd uses cron, not logrotate, and we follow suit.

Use mode 0700 for the auditd daily cron script, as required.

auto: ECLP-1
auto: GEN003100
auto: GEN003120
auto: GEN003140

```
    file { "/etc/cron.daily/auditd.cron":
        owner => root, group => 0, mode => 0700,
        source => "puppet:///modules/audit/auditd.cron",
    }
```

We need a non-stock Augeas lens to edit the auditd.conf.

```
    require augeas

    augeas { "auditd_conf":
        context => "/files/etc/audit/auditd.conf",
        changes => [
```

Rotate audit log files based on time, not their size.

auto: ECSC-1
auto: GEN002860

```
            "set max_log_file_action ignore",
```

Keep a ridiculous number of logs. (Most of our machines have a lot of local free space.)

```
            "set num_logs 30",
```

"[E]nsure that audit logs that have reached maximum length are not overwritten," by suspending the system if space for audit logs runs out or disk errors prevent the writing of audit logs.

auto: ECRR-1

```
            "set admin_space_left 50",
            "set admin_space_left_action SUSPEND",
            "set disk_full_action SUSPEND",
            "set disk_error_action SUSPEND",
```

Send an email to the administrator when disk space reserved for audit logs runs low. Mail for root is set up to go to the right places by §**??**.

auto: ECAT-1
auto: GEN002719
auto: GEN002730
auto: RHEL-06-000005

```
            "set space_left 300",
            "set space_left_action email",
            "set action_mail_acct root",
        ],
        notify => Service["auditd"],
    }
```

Configure the auditing subsystem according to the requirements of the UNIX SRG.

auto: ECAR-2
auto: ECAT-1
auto: GEN002720
auto: GEN002740
auto: GEN002750
auto: GEN002751
auto: GEN002752
auto: GEN002753
auto: GEN002760
auto: GEN002800
auto: GEN002820
auto: GEN002825

```
    file { "/etc/audit/audit.rules":
        owner => root, group => 0, mode => 0640,
        source => "puppet:///modules/audit/\
${architecture}-stig.rules",
        notify => Service["auditd"],
    }
```

Reload auditd, don't restart it.

```
    service { "auditd":
        restart => "/sbin/service auditd reload",
    }
}
```

### 11.12.3 Remote audit logging

Remote audit logging in our environment has the following requirements: admins do ECRG-1

1. Make it harder for an attacker who compromises a server to redact its audit log, by sending auditing data from the subject server off to another server.

2. Make it hard for non-admins to see what audit messages result from a given stimulus to the server, by hiding audit messages from non-admins.

3. Use encryption rather than a separate network to do this hiding: multiple networks connected to one host can cause allergic reactions in some accreditors.

4. Use a different means of encrypting and sending audit messages than the one used for sending system log messages, to avoid a single point of failure. (rsyslogd's SSL remote logging seems a bit flaky in practice.)

5. Be as simple as possible within these constraints.

The Linux auditing subsystem supports encrypted remote audit logging using Kerberos for authentication and encryption. For each host sending its audit data off remotely, there must be a Kerberos principal. In order to avoid imposing the unique security requirements of the auditing subsystem on any organization-wide Kerberos deployment, a Kerberos realm dedicated for remote auditing is set up.

#### Collect remote audit messages

The audit message collector host must include this class.

```
class audit::remote::collect($realm) {
```

These steps come from `http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5/html/Deployment_Guide/s1-kerberos-server.html`. That document regards RHEL5; it appears that RHEL6 documentation does not contain things about Kerberos servers.

Install needed packages on the KDC (Key Distribution Center) host.

```
package { [
        'krb5-libs',
        'krb5-server',
        'krb5-workstation',
    ]:
        ensure => present,
}
```

Configure the KDC service.

§11.13

```
include augeas
$ourrealm = "realms/realm[.='$realm']"
augeas { "audit_kdc_set_realm":
    require => Class['augeas'],
    context => '/files/var/kerberos/krb5kdc/kdc.conf',
    changes => [
        "rm realms/realm[.='EXAMPLE.COM']",
        "rm realms/realm[.='$realm']",
        "set realms/realm[999] $realm",
        "set $ourrealm/acl_file \
             /var/kerberos/krb5kdc/kadm5.acl",
        "set $ourrealm/admin_keytab \
             /var/kerberos/krb5kdc/kadm5.keytab",
        "set $ourrealm/supported_enctypes/type \
             aes256-cts:normal",
    ],
}
```

Configure the `krb5.conf`.

This is done as an exported resource, so that the hosts which generate audit records can configure themselves with the resource as well. Values for variables used inside this resource are figured on the audit collector host, not on the generator hosts.

The krb5.conf lens is part of Augeas, so we need not depend on our Augeas customizations for this resource.

```
@@augeas { "audit_krb5_conf":
    context => '/files/etc/krb5.conf',
    changes => [
        "rm realms/realm[.='$realm']",
        "set realms/realm[999] $realm",
        # $ourrealm is set above the previous augeas resource
        "set $ourrealm/kdc $::fqdn",
        "set $ourrealm/admin_server $::fqdn",

        "set libdefaults/default_realm $realm",
        "set domain_realm/$::domain $realm",
        "set domain_realm/.$::domain $realm",
    ],
    tag => "audit_krb5_for_${::fqdn}",
}
```

The Augeas resource that configures the krb5.conf, both for audit producing hosts and audit collecting hosts, is written in §11.12.3.

Collect that exported resource on the audit collector host.

```
Augeas <<| tag == "audit_krb5_for_${::fqdn}" |>>
```

Configure admin access to the KDC.

```
file { "/var/kerberos/krb5kdc/kadm5.acl":
    owner => root, group => 0, mode => 0600,
    content => "*/admin@$realm\t*\n",
}
```

Create the database.

First, we'll need some passwords.

```
define choose_password($write_to_file) {
    exec { $name:
```

We're basing it on a random number, so we want FIPS compliance in place first.

```
        require => Class['fips'],
        command => "/usr/bin/head -c 50 /dev/random | \
             /usr/bin/sha1sum | \
             /bin/cut -d' ' -f1 > ${write_to_file}",
        creates => $write_to_file,
```

Disable timeout: there's no way to know how long it will take to come up with enough entropy.

```
        timeout => 0,
    }
}
```

Choose a password for the master principal.

```
$masterpass = '/var/kerberos/krb5kdc/.masterpass'
choose_password { 'master':
    write_to_file => $masterpass,
}
```

Armed with the master password, create the database.

```
exec { 'audit_create_krb5_db':
    require => [
        Augeas['audit_kdc_set_realm'],
        File['/var/kerberos/krb5kdc/kadm5.acl'],
    ],
    command => "/bin/cat ${masterpass} ${masterpass} | \
                     /usr/sbin/kdb5_util create -s",
    creates => '/var/kerberos/krb5kdc/principal',
}
```

Now we need a principal for Puppet to use to do all of the admin work specified in this manifest.

Choose a password for that principal.

```
$puppetpass = '/var/kerberos/krb5kdc/.puppetpass'
choose_password { 'puppet':
    write_to_file => $puppetpass,
}
```

Add the principal. Since adding the principal doesn't make anything happen that we can see from Puppet, we have to make a stamp file to avoid doing it twice.

```
    $puppetstamp = '/var/kerberos/krb5kdc/.stamp-puppet'

    exec { 'audit_create_puppet_princ':
        require => [
            Choose_password['puppet'],
            Exec['audit_create_krb5_db'],
        ],
        command => "/bin/cat ${puppetpass} ${puppetpass} | \
                    /usr/sbin/kadmin.local \
                        -q 'addprinc puppet' \
                    > ${puppetcookie}",
        creates => $puppetcookie,
    }
```

Now set the KDC and kadmin running.

```
    service { 'krb5kdc':
        require => [
            Package['krb5-server'],
            Augeas['audit_krb5_conf'],
            Exec['audit_create_krb5_db'],
            Exec['audit_create_puppet_princ'],
        ],
        ensure => running,
        enable => true,
    }

    service { 'kadmin':
        require => [
            Package['krb5-server'],
            Augeas['audit_krb5_conf'],
            Exec['audit_create_krb5_db'],
            Exec['audit_create_puppet_princ'],
        ],
        ensure => running,
        enable => true,
    }
}
```

## 11.13 Augeas config file editor

Many parts of this policy use the Augeas system for editing all sorts of configuration files. Make sure it's properly installed.

```
class augeas {
```

We would normally just need to ensure that ruby-augeas is present; but 0.4.1 has some changes that are important for us in this Puppet manifest. And you have to specify an entire version, I think. But the entire version, with release number, varies between OS releases. Ergo, this big long nest of curly braces:

```
case $osfamily {
    RedHat: {
        package { "augeas":
            ensure => present,
        }
        case $operatingsystemrelease {
            /^6\..*/: {
                package { "ruby-augeas":
                    ensure => ’0.4.1-1.el6’,
                }
            }
            /^5\..*/: {
                package { "apscl-rubygem-ruby-augeas":
                    ensure => ’0.5.0-6’,
                }
            }
            default: { unimplemented() }
        }
    }
    ’Darwin’: {
        case $::macosx_productversion_major {
            ’10.6’: {
                mac_package { ’libxml2-2.9.0-1.pkg’: } ->
                mac_package { ’augeas-1.0.0-1.pkg’: } ->
                mac_package { ’ruby-augeas-0.4.1-1.pkg’: }
            }
            ’10.9’: { warning ’augeas install unimplemented on mavericks’ }
            default: { unimplemented() }
        }
    }
}


$lenses_dir = $::osfamily ? {
    ’RedHat’ => ’/usr/share/augeas/lenses’,
    ’Darwin’ => $::macosx_productversion_major ? {
        ’10.9’  => ’/usr/share/augeas/lenses’,
        ’10.6’  => ’/usr/local/share/augeas/lenses’,
    },
}

$lenses_source = $::augeasversion ? {
    ’0.9.0’ => ’puppet:///modules/augeas/0.9.0/lenses’,
    ’1.0.0’ => ’puppet:///modules/augeas/1.0.0/lenses’,
    ’1.2.0’ => ’puppet:///modules/augeas/1.2.0/lenses’,
    ’’ => ’’,
}
```

Install our custom Augeas lenses.

```
       if $lenses_source != ’’ {
           file { $lenses_dir:
               source => $lenses_source,
               ignore => ".svn",
               recurse => true, recurselimit => 1,
               owner => root, group => 0, mode => 0644,
           }
```

Remove the ones which are no longer valid. (We can’t make the copy remove unknown files because the Augeas lenses distributed in the Augeas package are also under that directory.)

```
           file { [
               "${lenses_dir}/logindefs.aug",
               "${lenses_dir}/tcp_wrappers.aug",
               "${lenses_dir}/ssh_config.aug",
               ]:
                   ensure => absent,
           }
       }
```

Remove lenses no longer valid specifically for Augeas version 1.0.0.

```
       if $::augeasversion =˜ /1\..*/ {
           file { [
```

Auditdconf has been superseded by the distributed Simplevars.

```
               "${lenses_dir}/auditdconf.aug",
               "${lenses_dir}/someautomountmaps.aug",
               ]:
                   ensure => absent,
           }
       }
   }
```

## 11.14 Automatic power on after power failure

### 11.14.1 Disable automatic power on after power failure

```
class auto_power_on::no {
    include "auto_power_on::no::${::osfamily}"
}
   class auto_power_on::no::darwin {
       exec { ’disable auto power on’:
           command => ’systemsetup -setrestartpowerfailure off’,
           unless => ’systemsetup -getrestartpowerfailure | \
                      grep Off\$’,
       }
   }
   class auto_power_on::no::redhat {}
```

# 11.15 Automatic login

## 11.15.1 Disable automatic login

```
class autologin::no {
    include "autologin::no::${::osfamily}"
}
   class autologin::no::darwin {
       $version_underscores = regsubst(
           $::macosx_productversion_major,
           '\D', '_', 'G')
       $klassname = "${::osfamily}_${version_underscores}"
       include "autologin::no::${klassname}"
   }
   class autologin::no::darwin_10_6 {}
   class autologin::no::darwin_10_9 {
```

Disable automatic login on Macs. auto: OSX8-00-00925

Isn't this curious? Such a long key, and with a different reverse-DNS at its beginning than the record name. Oh, Apple.

```
       mcx::set { 'com.apple.loginwindow/com.apple.login.mcx.DisableAutoLoginClient':
           value => true,
       }
   }
   class autologin::no::redhat {
   }
```

§11.6.1.2

# 11.16 Automatic logout

## 11.16.1 Disable automatic logout

```
class autologout::no {
    case $::osfamily {
        'Darwin': { include autologout::no::darwin }
        default:  { unimplemented() }
    }
}
```

### Disable automatic logout on Macs

```
class autologout::no::darwin {
```

Disable "automatic logout due to inactivity" on Macs. auto: ECSC-1

```
       mac_plist_value { "disable autologout":
           file => "/Library/Preferences/.GlobalPreferences.plist",
           key => "com.apple.autologout.AutoLogOutDelay",
           value => 0,
       }
   }
```

auto: OSX00435 M6

auto: OSX8-00-01085

# 11.17 Automount

Mount NFS filesystems via the automounter, under `/net`.

"Automated file system mounting tools must not be enabled unless needed," because they "may provide unprivileged users with the ability to access local media and network shares." This automount configuration does not enable access to local media, and constricts network share access to filers designated for the purpose of serving unprivileged users.

auto: ECSC-1
auto: GEN008440

```
class automount {
```

If we're automounting we're going to be using NFS. Make sure we're prepared for that.

```
    include nfs
    include "automount::${::osfamily}"
}
class automount::darwin {
    service { 'com.apple.autofsd':
        enable => true,
        ensure => running,
    }

    $version_underscores = regsubst(
        $::macosx_productversion_major,
        '\D', '_', 'G')
    $klassname = "${::osfamily}_${version_underscores}"
    include "automount::${klassname}"

}
class automount::darwin_10_6 {}
```

§11.69

### 11.17.1 Automount configuration under Mavericks

```
class automount::darwin_10_9 {
```

To edit automount maps we need Augeas.

```
    require augeas
```

Augeas 1.2.0 does not appear to understand how to edit `/etc/auto_master` on a Mavericks Mac, even if it doesn't contain anything weird. Oh, well; what we need in it is quite fixed anyway.

```
    file { '/etc/auto_master':
        owner => root, group => 0, mode => 0644,
        content => "
/net auto_net
",
    }
```

Make sure the auto.net file exists: otherwise any attempt at editing it will fail, causing errors.

```
    file { "/etc/auto_net":
        owner => root, group => 0, mode => 0644,
        ensure => present,
    }

    augeas { "automount_remove_autonet_script":
        require => File["/etc/auto_net"],
        context => "/files/etc/auto_net",
        changes => "rm script_content",
    }
}
```

### 11.17.2 NFS mounts

To make sure of a certain filesystem being mounted, call this define like so:

`automount::mount` { *`name`*`:  from => "`*`nfs path`*`"` }

For example, `automount::mount` { `"home":  from => "myfiler:/export/home"` } would make sure that `myfiler`'s `/export/home` share is mounted as `/net/home`.

To remove an automount entry:

`automount::mount` { *`name`*`:  from => "`*`nfs path`*`", ensure => absent` }

If you have additional mount options (such as you would give to `mount(1)`'s `-o` switch), give them in an array as the options parameter. For example:

`automount::mount` { `'home':  from => 'myfiler:/export/home', options => ['nolocks','nordirplus'],` }

The options given in the options parameter may be inside multiple levels of arrays; this is so that you can create layers of abstraction above this define. The set of available options varies from platform to platform, and the behavior when an unknown option is supplied may also vary.

* * *

```
define automount::mount($from, $under='', $ensure='present', $options=[]) {
    include automount                                          §11.17
    case $::osfamily {
        'redhat': {
            automount::mount::redhat { $name:                  §11.17.4
                from => $from,
                under => $under,
                ensure => $ensure,
                options => $options,
            }
        }
        'darwin': {
            automount::mount::darwin { $name:                  §11.17.2
```

```
                from => $from,
                under => $under,
                ensure => $ensure,
                options => $options,
            }
        }
        default: { unimplemented() }
    }
}
define automount::mount::darwin($under, $from, $ensure, $options) {
    case $::macosx_productversion_major {
        '10.6': {
            automount::mount::darwin_10_6 { $name:                          §11.17.2
                under => $under,
                from => $from,
                ensure => $ensure,
                options => $options,
            }
        }
        '10.9': {
            automount::mount::darwin_10_9 { $name:                          §11.17.3
                under => $under,
                from => $from,
                ensure => $ensure,
                options => $options,
            }
        }
    }
}
define automount::mount::darwin_10_6($under, $from, $ensure, $options) {
    if $under == '' {
```

Ensure the nosuid option is used when mounting an NFS filesystem. auto: ECLP-1
auto: ECPA-1

Ensure the nodev option is used when mounting an NFS filesystem. auto: GEN002420
auto: GEN005900
auto: ECSC-1
auto: GEN002430

```
        mac_automount { "/net/${name}":
            source => $from,
            ensure => $ensure,
            options => ['nodev', 'nosuid', $options],
            notify => Service['com.apple.autofsd'],
        }
    }
    else {
        if !defined(Automount::Mount[$under]) {
            automount::mount { $under: ensure => absent, from => 'nonce/dontmatter' }    §11.42.4
        }
```

Ensure the nosuid option is used when mounting an NFS filesystem. auto: ECLP-1
auto: ECPA-1

Ensure the nodev option is used when mounting an NFS filesystem. auto: GEN002420
auto: GEN005900
auto: ECSC-1
auto: GEN002430

```
            mac_automount { "/net/${under}/${name}":
                source => $from,
                ensure => $ensure,
                options => ['nodev', 'nosuid', $options],
                notify => Service['com.apple.autofsd'],
            }
        }
    }
```

### 11.17.3 Adding an automount entry under Mavericks

Don't use this directly: use `automount::mount` and let it sort out what platform you are on. Documentation is above.

```
define automount::mount::darwin_10_9($from, $under='', $ensure='present', $options=[]) {

    include augeas
    $hostpath = split($from, ':')
    $host = $hostpath[0]
    $path = $hostpath[1]
```

§11.13

Ensure the `nosuid` option is used when mounting an NFS filesystem.

Ensure the `nodev` option is used when mounting an NFS filesystem.

auto: ECLP-1
auto: ECPA-1
auto: GEN002420
auto: GEN005900
auto: ECSC-1
auto: GEN002430

```
    $stig_required = "
set \$here/opt[last()+1] nosuid
set \$here/opt[last()+1] nodev
"

    $extra = inline_template("
<% @options.flatten.each do |o| %>
set \$here/opt[last()+1] ’<%=o%>’
<% end %>
")

    $set_values_script = "
rm  \$here/opt
${stig_required}
${extra}
rm  \$here/location
set \$here/location/1/host ${host}
set \$here/location/1/path ${path}
"

    Augeas {
        lens => ’Automounter.lns’,
        incl => ’/etc/auto_net’,
        context => "/files/etc/auto_net",
        require => File[’/etc/auto_net’],
        notify => Service[’com.apple.autofsd’],
    }

    case $ensure {
        ’present’: {
            if $under == ’’ {
                augeas { "create mount ${name}":
                    onlyif => "match *[.=’${name}’] size == 0",
                    changes => "
defnode here 999 ’${name}’
${set_values_script}
",
                }
                augeas { "modify mount ${name}":
                    onlyif => "match *[.=’${name}’] size > 0",
                    changes => "
defnode here *[.=’${name}’] ’${name}’
${set_values_script}
",
                }
            }
            else {
                augeas { "fix submount ${name} under ${under}":
                    onlyif => "match *[.=’${under}’][mount/*=’/${name}’] size > 0",
                    changes => "
defvar top *[.=’${under}’]
defvar here \$top/mount/*[.=’/${name}’][last()]
${set_values_script}
",
                }
                augeas { "create toplevel ${under} and submount ${name}":
                    onlyif => "match *[.=’${under}’] size == 0",
                    changes => "
defnode top 999 ’${under}’
```

### 11.17.4 Adding an automount entry under Red Hat

Don’t use this directly: use `automount::mount` and let it sort out what platform you are on. Documentation is above.

```
define automount::mount::redhat($from, $under=’’, $ensure=’present’, $options=[]) {
    include augeas
    $hostpath = split($from, ’:’)
    $host = $hostpath[0]
    $path = $hostpath[1]
```

§11.13

Ensure the `nosuid` option is used when mounting an NFS filesystem.

Ensure the `nodev` option is used when mounting an NFS filesystem.

auto: ECLP-1
auto: ECPA-1
auto: GEN002420
auto: GEN005900
auto: ECSC-1
auto: GEN002430

```
    $stig_required = "
set \$here/opt[last()+1] nosuid
set \$here/opt[last()+1] nodev
"

    $extra = inline_template("
<% @options.flatten.each do |o| %>
set \$here/opt[last()+1] ’<%=o%>’
<% end %>
")
```

(The comments in the stock /etc/auto.master make it seem that these may be defaults under the conditions where we are using the automounter; but, better safe than sorry.)

Under RHEL5, the default was to use TCP for NFS mounts, according to `nfs(5)`; under RHEL6 the default is to try to autonegotiate. Without any deeper investigation, it is apparent that this process does not work, and specifying `proto=tcp` makes it work properly. See `nfs(5)` under RHEL6 for more details.

```
    $base_options = "
set \$here/opt[last()+1]     nfsvers
set \$here/opt[last()]/value 3
set \$here/opt[last()+1]     proto
set \$here/opt[last()]/value tcp
"


    $set_values_script = "
set \$here '${name}'
rm  \$here/opt
${base_options}
${stig_required}
${extra}
rm  \$here/location
set \$here/location/1/host ${host}
set \$here/location/1/path ${path}
"

    if $under == '' {
        $autotable = '/etc/auto.net'
        $requires = []
    }
    else {
        $autotable = "/etc/auto.${under}"
        if !defined(Automount::Mount::Redhat::Subdir[$under]) {
            automount::mount::redhat::subdir { $under:                §11.17.4
                ensure => $ensure,
            }
        }
        if !defined(Automount::Mount::Redhat[$under]) {
            automount::mount::redhat { $under:                        §11.17.4
```

```
                ensure => absent,
                from => ’nonce:/dontmatter’,
            }
        }
        $requires = [Automount::Mount::Redhat::Subdir[$under],
            Automount::Mount::Redhat[$under]]
    }

    Augeas {
        lens => ’Automounter.lns’,
        incl => $autotable,
        context => "/files${autotable}",
        require => [
            File[$autotable],
            Package["autofs"],
            $requires,
            ],
        notify => Service[’autofs’],
    }
    case $ensure {
        ’present’: {
            augeas { "create_mount_${under}_${name}":
                onlyif => "match *[.=’$name’] size == 0",
                changes => "
defnode here 999 ${name}
${set_values_script}
",
            }
            augeas { "modify_mount_${under}_${name}":
                onlyif => "match *[.=’$name’] size > 0",
                changes => "
defnode here *[.=’${name}’] ${name}
${set_values_script}
",
            }
        }
        ’absent’: {
            augeas { "no_mount_${under}_${name}":
                changes => [
                    "rm *[.=’$name’]",
                ],
            }
        }
    }
}
```

This is used only by automount::mount::redhat.

```
define automount::mount::redhat::subdir($ensure=’present’) {
    include automount
```

§11.17

First, make sure we don’t tread on existing configuration.

```
    if $name == 'net' {
        fail('You cannot use automount::subdir to create /net/net')
    }
```

Now, make a subtable in the automount configuration.

```
    file { "/etc/auto.${name}":
        owner => root, group => 0, mode => 0644,
        ensure => $ensure,
    }
    if $ensure == 'present' {
        augeas { "automount_add_master_subdir_${name}":
            context => '/files/etc/auto.master',
            changes => [
                "set map[.='/net/${name}'] /net/${name}",
                "set map[.='/net/${name}']/name /etc/auto.${name}",
                "set map[.='/net/${name}']/options --ghost",
                ],
            require => [],
        }
    } else {
        unimplemented()
    }
}
```

### 11.17.5 Turn off automount

Turn off the automounter, on machines where it should not be on.

```
class automount::no {
    case $::osfamily {
        'redhat': {
            service { "autofs":
                enable => false,
               ensure => stopped,
            }
        }
        'darwin': { warning "unimplemented on Macs" }
        default:  { unimplemented() }
    }
}
```

### 11.17.6 Automount configuration under Red Hat

```
class automount::redhat {
```

To edit automount maps we need Augeas.

```
    require augeas

    package { "autofs": ensure => present}

    augeas { "automount_fixed_net_map":
        context => "/files/etc/auto.master",
        changes => [
            "set map[.='/net'] /net",
            "set map[.='/net']/name /etc/auto.net",
            "set map[.='/net']/options --ghost",
            "rm include",
            "rm map[.='/misc']",
        ],
    }
```

Make sure the auto.net file exists: otherwise any attempt at editing it will fail, causing errors.

```
    file { "/etc/auto.net":
        owner => root, group => 0, mode => 0644,
        ensure => present,
    }

    augeas { "automount_remove_autonet_script":
        require => File["/etc/auto.net"],
        context => "/files/etc/auto.net",
        changes => "rm script_content",
    }

    service { "autofs":
        enable => true,
        ensure => running,
        require => Package["autofs"],
```

For some reason some NFS mounts added did not show up when `autofs` was restarted using the `reload` verb instead of `restart`. So even though `restart` is slower and could screw more things up, it's what we need to use.

```
        restart => "/sbin/service autofs restart",
    }
}
```

### 11.17.7 NFS mounts in subdirectories

In the case where you want a mountpoint like /net/foo/bar, `automount::mount` will not suffice. Use this instead.

Example:

```
automount::subdir { 'flarble': }
automount::submount { 'zart': under => 'flarble', from => 'myserver:/dir' }
```

This will create a directory `/net/flarble`, and mount `myserver:/dir` onto `/net/flarble/zart`. It will also unmount anything that was to be mounted under /net/flarble.

```
define automount::subdir($ensure='present') {
    include automount                                                   §11.17
```

First, make sure we don't tread on existing configuration.

```
    case $name {
        'net': { fail('You cannot use automount::subdir to create /net/net') }
        default: {}
    }
```

Now, make a subtable in the automount configuration.

```
    case $::osfamily {
        'redhat': {
            file { "/etc/auto.${name}":
                owner => root, group => 0, mode => 0644,
                ensure => $ensure,
            }
            if $ensure == 'present' {
                augeas { "automount_add_master_subdir_${name}":
                    context => '/files/etc/auto.master',
                    changes => [
                        "set map[.='/net/${name}'] /net/${name}",
                        "set map[.='/net/${name}']/name /etc/auto.${name}",
                        "set map[.='/net/${name}']/options --ghost",
                        ],
                }
            }
        }
        'darwin': {}
        default: { unimplemented() }
    }
}
```

### 11.17.8 Define mounts under subdirectories

Whatever you give as the under value for this define, you must have an automount::subdir define for. See §11.17.7 and §11.42.4.

```
define automount::submount($under, $from, $ensure='present') {
    include automount                                                   §11.17
    case $::osfamily {
        'redhat': {
            automount::mount::redhat { $name:                           §11.17.4
                from => $from,
                under => $under,
                ensure => $ensure,
            }
        }
        'darwin': {
```

Ensure the nosuid option is used when mounting an NFS filesystem. auto: ECLP-1

Ensure the nodev option is used when mounting an NFS filesystem. auto: ECPA-1

auto: GEN002420
auto: GEN005900
auto: ECSC-1
auto: GEN002430

```
               mac_automount { "/net/${under}/${name}":
                   source => $from,
                   ensure => $ensure,
                   options => ['nodev', 'nosuid', 'nolock'],
                   notify => Service['com.apple.autofsd'],
               }
           }
           default: { unimplemented() }
       }
   }
```

## 11.18 Services that “call home”

### 11.18.1 Disable “call home” services

```
class call_home::no {
    include "call_home::no::${::osfamily}"
}
   class call_home::no::darwin {
       $version_underscores = regsubst(
           $::macosx_productversion_major,
           '\D', '_', 'G')
       $klassname = "${::osfamily}_${version_underscores}"
       include "call_home::no::${klassname}"
   }
   class call_home::no::darwin_10_6 {}
   class call_home::no::darwin_10_9 {
```

Disable “Find My Mac.” auto: OSX8-00-00531

```
       service { 'com.apple.findmymacd':
           ensure => stopped,
           enable => false,
       }
```

Disable the “Find My Mac” messenger. auto: OSX8-00-00532

```
       service { 'com.apple.findmymacmessenger':
           ensure => stopped,
           enable => false,
       }
```

Disable the sending of diagnostic and usage data to Apple. auto: OSX8-00-00530

```
       $lascr = '/Library/Application Support/CrashReporter'
       mac_plist_value { 'turn off AutoSubmit':
           file => "${lascr}/DiagnosticMessagesHistory.plist",
           key => 'AutoSubmit',
           value => false,
       }
   }
   class call_home::no::redhat {}
```

## 11.19 Cameras

Configure support for cameras connected as peripherals (i.e. webcams).

### 11.19.1 Disable cameras

Disable cameras where necessary to “protect the organization’s privacy.” auto: ECSC-1
auto: OSX00075 M6

```
  class camera::no {
      case $::osfamily {
          'darwin': { include camera::no::darwin }
          default:  { unimplemented() }
      }
  }
```

#### Disable cameras under Mac OS X

```
class camera::no::darwin {
    $exts = '/System/Library/Extensions'
    $usbp = "${exts}/IOUSBFamily.kext/Contents/PlugIns"
    file {
```

Disable “support for internal iSight cameras.”

```
            "${exts}/Apple_iSight.kext":
                ensure => absent,
                force => true;
```

Disable “support for external cameras.”

```
            "${usbp}/AppleUSBVideoSupport.kext":
                ensure => absent,
                force => true;
        }
```

Remove the Photo Booth application. auto: OSX8-00-00465

```
        file { '/Applications/Photo Booth.app':
            ensure => absent,
            recurse => true,
        }
```

Remove the FaceTime application. auto: OSX8-00-00475

```
        file { '/Applications/FaceTime.app':
            ensure => absent,
            recurse => true,
        }
```

Remove the Image Capture application. auto: OSX8-00-00495

```
        file { '/Applications/Image Capture.app':
            ensure => absent,
            recurse => true,
        }
    }
```

## 11.20 Citrix Receiver ICA client

Some users may require access to the Citrix XenApp server via the Citrix Receiver ICA client.

The ICAClient package is not part of RHEL: it must be fetched from Citrix. But the package fetched from Citrix is signed using the MD5 digest algorithm, and so will not install on a host configured for FIPS 140-2 compliance (see §11.32.3). So we have a custom package, the same in every salient respect except that it is signed using SHA256.

```
   class citrix_ica {
       case $::osfamily {
           ’RedHat’: {
               package { ’ICAClient’:
                   ensure => ’12.1.0.203066-1SK02’,
               }
               mozilla::wrap_32bit { ’npica.so’:                        §11.65.1
                   require => Package[’ICAClient’],
               }
           }
           ’Darwin’: { warning("citrix_ica not yet implemented on Macs") }
           default: { unimplemented() }
       }
       include pki::ca_certs::citrix_receiver                           §11.76.1
   }
```

## 11.21 Common packages

You only get to declare a package once in the whole manifest. But some packages are depended on by many modules. According to a googling done in Fall 2013, options for this are:

1. Surround every package resource with `if # !defined(Package[bla]) {...}`.
2. Write every possible package resource as a virtual resource in one place; realize packages where they are needed.
3. Wherever class A and class B both want to install package X, write a new class C that installs package X, and make A and B depend on C.

Here we implement the third approach.

### 11.21.1 graphviz

```
class common_packages::graphviz {
    package { ’graphviz’:
        ensure => installed,
    }
}
```

### 11.21.2 LaTeX

```
class common_packages::latex {
    package { ['texlive', 'texlive-latex']:
        ensure => installed,
    }
}
```

### 11.21.3 make

```
class common_packages::make {
    case $::osfamily {
        'RedHat': {
            package { 'make':
                ensure => installed,
            }
        }
        'Darwin': {}
        default: { unimplemented() }
    }
}
```

### 11.21.4 unix2dos

```
class common_packages::unix2dos {
    package { ['unix2dos', 'dos2unix']:
        ensure => installed,
    }
}
   class common_packages::unzip {
       case $::osfamily {
           'RedHat': {
               package { 'unzip':
                   ensure => present,
               }
           }
           'Darwin': {}
           default: { unimplemented() }
       }
   }
   class common_packages::wget {
       package { 'wget':
           ensure => present,
       }
   }
```

## 11.22 Contingency backup

Back up this Configuration Management for IT Systems Example Policy, along with organization-specific critical software and documentation, monthly onto read-only media. auto: COSW-1 auto: DCHW-1

(Regarding provisions for data backup in general, see the backup plan and contingency and business continuity plan [CBCP].)

Because this policy plays such an integral part in the installation and configuration of all sorts of hosts, you, the administrator, need it just as urgently during a contingency as you need the operating system install media. So this policy needs to be written on a CD or DVD, along with any software it installs which cannot be found on the vendor-provided install media—irrespective of other means by which it may also be backed up. And hosts which include this class will do just that.

auto: CODB-1
auto: OSX00675 M6

This Configuration Management for IT Systems Example Policy comprises a great deal of what is needed to accomplish "recovery of a damaged or compromised [Mac] system in a timely basis." Automated backup of the policy and its dependencies as described in this section is therefore an important part of compliance with this requirement.

### 11.22.1 Guidance for admins about contingency backups

admins do COSW-1
admins do DCHW-1
admins do OSX00675 M6

Store the contingency backup in a fire-rated container.

Lock the fire-rated container which holds the contingency backups.

Keep a ready supply of CD labels and DVDs. You must receive and abide by the automated email instructions, which are emailed to root (see §11.97.3). Maintain the automated backup script, so that it continues to correctly obtain and back up critical information for all automated information systems to which it pertains. This critical information is hardware baselines, software baselines, administrative manuals, custom software: everything needed to reconstitute each AIS.

The choices of which content to back up are laid out in `critical-backup`, which lives separately from this Configuration Management for IT Systems Example Policy in a Subversion repository.

### 11.22.2 The backup host

A backup host does the backing up. It needs the ability to send messages via SMTP to administrators, an optical drive capable of writing, and a printer. It should be a machine to which admins have frequent physical access. It must be able to check out the policy from the Subversion server non-interactively. And it must have elevated access to some NFS shares upon which critical system administration data is stored, that it can read some files that only root can read, and so that it can write a backup stamp file.

There can and should be more than one backup host. Machinery is built into the backup script so that between all backup hosts only one backup will be made per month.

Executables necessary to build the CMITS policy must be present and runnable by the `nobody` user.

```
class contingency_backup::host(
    $contingency_backup_url,
    $add_to_path,
    $add_to_pythonpath,
    $stamp_directory,
    ) {
    include common_packages::make                                       §11.21.3
    include common_packages::unix2dos                                   §11.21.4
    include common_packages::latex                                      §11.21.2
    include subversion::pki                                             §11.103.1
    package { [
            'file',
            'dvd+rw-tools',
            'ImageMagick',
            'iadoc',
            'iacic',
```

These two are for the empty-optical-disc-awaiter script.

```
            'pygobject2',
            'dbus-python',
        ]:
        ensure => present,
    }

    file { "/etc/cron.daily/contingency_backup.cron":
        owner => root, group => 0, mode => 0700,
        content => template("contingency_backup/cron.erb"),
    }
}
```

## 11.23 Core dumps

Ensure that "aborts are configured to ensure that the system remains in a secure state." auto: DCSS-1

### 11.23.1 Turn off core dumps

Turn off core dumps because we do not need them. auto: ECCD-1

auto: GEN003500

```
class core::no {
    case $::osfamily {
        'RedHat': {
```

This is done by means of `pam_limits.so`. Make sure it's in place.

```
            include pam::limits                                         §11.74.2
```

Now configure `pam_limits.so`. (See §11.74.3 for another example.)

```
            augeas {
                "limits_insert_core":
                    context => "/files/etc/security/limits.conf",
                    onlyif => "match *[.=’*’ and item=’core’]\
                                     size == 0",
                    changes => [
                        "insert domain after *[last()]",
                        "set domain[last()] ’*’",
                        "set domain[last()]/type hard",
                        "set domain[last()]/item core",
                        "set domain[last()]/value 0",
                    ];
                "limits_set_core":
                    require => Augeas["limits_insert_core"],
                    context => "/files/etc/security/limits.conf",
                    changes => [
                        "set domain[.=’*’ and item=’core’]/type hard",
                        "set domain[.=’*’ and item=’core’]/value 10",
                    ];
            }
        }
        ’Darwin’: {}
        default: { unimplemented() }
    }
}
```

With no core dumps, there is no centralized directory where core dumps are stored, so such a directory need not be secured.

N/A: GEN003501
N/A: GEN003502
N/A: GEN003503
N/A: GEN003504
N/A: GEN003505

### 11.23.2 STIG-required core dump configuration

If core dumps are required, include ”this class to configure them in the” required fashion. If not, include ”core::no.”

```
class core::stig {
    include "core::stig::${::osfamily}"
}
class core::stig::darwin {
    $core_dir = ’/Library/Logs/DiagnosticReports’
    file { $core_dir:
```

Ensure root owns the centralized core dump data directory. Ensure the group admin owns the centralized core dump data directory.

auto: OSX8-00-01175
auto: OSX0-00-01185

```
        owner => root, group => admin,
```

Ensure restrictive permissions on the centralized core dump data directory.

auto: OSX8-00-01180

```
        mode => 0750,
    }
}
```

### 11.23.3 Under Red Hat

If core dumps are required, include this class to configure them in the fashion required by the SRG. If not, include `core::no`.

Our goal here is to protect core dumps as if they contain sensitive data, because they may. The SRG requires things about where they are stored, but RHEL6 is more advanced: it contains SOS and ABRT (Automatic Bug Reporting Tool), both of which can send relevant details of a crash to the vendor (Red Hat) or the upstream maintainer of a package. Both of these give the user a chance to vet the outgoing information, but that's still an unacceptable level of risk to the data. Accordingly, in order to keep the spirit of the SRG in an area where its letter does not speak, we secure these tools.

ABRT sets the kernel's `core_pattern` variable so that core dumps are sent to ABRT, and analyzed and written by that tool.

If this policy is extended to other versions of RHEL, this section will need to be revisited.

```
class core::stig::redhat {
```

The stock ABRT config file has sections that look like `[ Section ]`. This means that Augeas expressions for settings inside those sections contain spaces, and Augeas tends to think that spaces delimit parameters. So we need to go in and take out those spaces.

```
    exec { "sanify_abrt_conf_sections":
        command => "/bin/sed -i -e \
            's/^\\[ \\+\\(.*\\) \\+\\]/[\\1]/g' \
            /etc/abrt/abrt.conf",
        onlyif => "/bin/grep '^\\[ ' /etc/abrt/abrt.conf",
    }
```

non-stock Augeas lens

```
    include augeas
```
§11.13

```
    augeas {
```

Remove vendor-supplied FTP and email addresses for SOS uploading, breaking this feature.

```
        "abrt_remove_sos_uploads":
            context => "/files/etc/sos.conf",
            changes => [
                "set ftp_upload_url ''",
                "set gpg_recipient ''",
            ];
```

Don't activate SOS immediately when a crash occurs.

```
        "abrt_disable_sos":
            require => Exec['sanify_abrt_conf_sections'],
            context => "/files/etc/abrt/abrt.conf/Common",
            changes => "rm ActionsAndReporters[.='SOSreport']";
```

Remove `RHTSupport` plugin from the loop for each crash type.

```
        "abrt_logger_only":
            require => Exec['sanify_abrt_conf_sections'],
            context => "/files/etc/abrt/abrt.conf/\
AnalyzerActionsAndReporters",
            changes => [
                "set Kerneloops Logger",
                "set CCpp Logger",
                "set Python Logger",
            ];
    }
```

Control ownership and permissions for core-dump-related files written by the Automated Bug Reporting Tool (ABRT).

auto: ECLP-1
auto: GEN003501
auto: GEN003502
auto: GEN003503
auto: GEN003504

ABRT uses `/var/spool/abrt` for its core files; files may be uploaded into `/var/spool/abrt-upload`, so it will be protected similarly. ABRT's directories must be owned by the abrt user, not root; this does not fulfill the letter of the SRG requirements, but it also does not violate their spirit.

```
    file {
        "/var/spool/abrt":
            owner => abrt, group => 0, mode => 0600,
            recurse => true, recurselimit => 2;
        "/var/spool/abrt-upload":
            owner => abrt, group => 0, mode => 0600,
            recurse => true, recurselimit => 2;
    }
```

Remove extended ACLs on ABRT directories.

auto: ECLP-1
auto: GEN003505

```
    no_ext_acl {
        "/var/spool/abrt": recurse => true;
        "/var/spool/abrt-upload": recurse => true;
    }
}
```

After all this, ABRT and SOS will not do anything rash automatically, but data from crashes will likely still be saved, can be read only by administrators, and can be sent on to vendors or upstream developers where necessary and appropriate.

## 11.24 Cron

RHEL implements cron logging by default.

RHEL5, RHEL6: GEN003160

### 11.24.1 Automated policy

```
class cron($allowed_users=[]) {

    $crontab = $::osfamily ? {
        'darwin' => '/private/var/at/tabs/root',
        'redhat' => '/etc/crontab',
        default  => unimplemented,
    }
    $cron_allow = $::osfamily ? {
        'darwin' => '/private/var/at/cron.allow',
        'redhat' => '/etc/cron.allow',
        default  => unimplemented,
    }
    $cron_deny = $::osfamily ? {
        'darwin' => '/private/var/at/cron.deny',
        'redhat' => '/etc/cron.deny',
        default  => unimplemented,
    }
```

Under Snow Leopard, /usr/lib/cron is a symlink to ../../var/at, and /var is a symlink to /private/var.

cron usually does daily tasks at 4:00 am or so. Sometimes we have tasks that need to send routine email to real people who may have Blackberries, so that emailing them at four in the morning would be a bad idea. For such tasks, we have cron.morningly.

```
        $cron_dirs = $::osfamily ? {
            'darwin' => [ '/private/var/at' ],
            'redhat' => [ '/etc/cron.d', '/etc/cron.morningly',
                          '/etc/cron.hourly', '/etc/cron.daily',
                          '/etc/cron.weekly', '/etc/cron.monthly' ],
            default  => unimplemented,
        }
        $cron_tools = $::osfamily ? {
            'darwin' => [ '/usr/sbin/cron', '/usr/bin/crontab' ],
            'redhat' => [ '/usr/sbin/crond', '/usr/bin/crontab' ],
            default  => unimplemented,
        }

        file {
```

Make sure only root can edit the cron.allow file.

auto: ECLP-1
auto: GEN003250

```
            $cron_allow:
                owner => root, group => 0, mode => 0600;
```

Make sure only root can edit the cron.deny file.

auto: ECLP-1
auto: GEN003270 M6
auto: ECLP-1
auto: GEN003270

```
            $cron_deny:
                owner => root, group => 0, mode => 0600;
```

Restrict access to the system crontab to only root.

auto: DCSL-1
auto: ECLP-1
auto: GEN003040
auto: GEN003050
auto: GEN003080

```
            $crontab:
                owner => root, group => 0, mode => 0600;
```

Control ownership and permissions of the “at” directory, which under Mac

auto: ECLP-1
auto: GEN003400 M6
auto: GEN003420 M6

OS X is the same as the “cron” directory.

Under RHEL, restrict access to directories used by `run-parts`, which contain scripts to be run periodically, to only root. Also restrict access to the files in these directories.

auto: ECLP-1
auto: GEN003100
auto: GEN003120
auto: GEN003140
auto: ECLP-1
auto: GEN003080-2

```
        $cron_dirs:
            ensure => directory,
            owner => root, group => 0, mode => go-rwx,
            recurse => true, recurselimit => 2;
    }


    no_ext_acl {
```

Remove extended ACLs on `cron.allow`. Remove extended ACLs on `cron.allow`.

auto: ECLP-1
auto: GEN002990 M6
auto: ECLP-1
auto: GEN002990

```
        $cron_allow:;
```

Remove extended ACLs on `crontab`.

auto: ECLP-1
auto: GEN003245

```
        $crontab:;
```

Remove extended ACLs on directories used by `run-parts`.

auto: ECLP-1
auto: GEN003090

```
        $cron_dirs:;
```

Remove extended ACLs on `cron.deny`.

auto: ECLP-1
auto: GEN003110
auto: ECLP-1
auto: GEN003210

```
        "/etc/cron.deny":;
    }

    case $::osfamily {
        'redhat': {
            cron { morningly:
                command => "run-parts /etc/cron.morningly",
                user => root,
                hour => 8,
                minute => 2,
            }
```

Under RHEL, control usage of the `cron` utility.

auto: ECLP-1
auto: ECPA-1
auto: GEN002960
auto: GEN002980
auto: GEN003060
auto: GEN003240

The STIG doesn’t say it has to be only usable by root: merely that its use must be controlled by the use of `cron.allow` and `cron.deny` files.

```
            File[$cron_allow] {
                content +> inline_template("
root
<% @allowed_users.to_a.each {|user| %>
<%=user %>
<% } %>"),
            }
```

Under RHEL, remove the `cron.deny` file if it exists.

auto: ECLP-1
auto: GEN003200
auto: GEN003260
auto: GEN003270

```
            File[$cron_deny] {
                ensure +> absent,
            }
        }
```

Under Mac OS X, it appears we cannot limit cron usage to root only, because some antivirus software may depend on its use with non-root users. Also we don’t yet do anything morningly on Macs, so we needn’t worry about setting it up.

```
        'darwin': {}
        default:  {}
    }
}
```

### 11.24.2 Guidance for administrators about cron

Don't write a cron script that changes the umask.

admins do GEN003220

System administrators who need to accomplish periodic tasks which should not be run as root should write scripts that use su or sudo to become the appropriate user before beginning the task.

Before writing or deploying a cron script, make sure it will not execute group- or world-writable programs, nor execute programs in or under world-writable directories.

admins do DCSL-1
admins do GEN003000
admins do GEN003020

### 11.24.3 Daily cron job

Make sure something happens every day—portably.

On Red Hattish Linux hosts, `/etc/cron.daily` exists and is a directory, and executable files inside it are run once a day. On Mac hosts, this directory does not exist.

```
define cron::daily($source) {
    case $::osfamily {
        'RedHat': {
            file { "/etc/cron.daily/${name}":
                owner => root, group => 0, mode => 0700,
                source => $source,
            }
        }
        'Darwin': {
            warning 'cron::daily unimplemented on Macs'
        }
    }
}
```

## 11.25 CUPS (Common UNIX Printing System)

```
class cups::darwin {
```

CUPS is part of Mac OS X and can't be uninstalled, so we have nothing to install. But we do need to make sure it's running.

```
    service { 'org.cups.cupsd':
        enable => true,
        ensure => running,
    }
}
```

### 11.25.1 Set system default printer

```
class cups::default($printer) {
    exec { "set default printer to ${printer}":
        command => "lpadmin -d ’${printer}’",
        unless => "lpstat -d | grep ’${printer}’ >&/dev/null",
        require => Cups::Printer[$printer],
    }
}
```

### 11.25.2 Disable CUPS

On hosts which do not need to print, disable CUPS entirely. This trivially complies with this requirement not to “allow all hosts to use local print resources.”

auto: ECCD-1
auto: GEN003900

```
    class cups::no {
```

Remove CUPS and the “hosts.lpd (or equivalent) file,” which in the case of CUPS is `/etc/cups/cupsd.conf`. This trivially prevents “unauthorized modifications” or “unauthorized remote access.”

auto: ECLP-1
auto: GEN003920
auto: GEN003930
auto: GEN003940
auto: GEN003950

```
        include "cups::no::${::osfamily}"
        file { ’/etc/cups/cupsd.conf’:
            ensure => absent,
        }
    }
    class cups::no::darwin {
```

You can’t get rid of CUPS on Mac OS X; it’s part of the operating system. But you can make sure it isn’t running.

```
        service { ’org.cups.cupsd’:
            enable => false,
            ensure => stopped,
        }
    }
    class cups::no::redhat {
        package { ’cups’:
            ensure => absent,
        }
    }
```

### 11.25.3 Define a printer

This defined resource type adds or removes CUPS printers, and enables or disables them.

It wraps the `lpadmin(8)` command, *q.v.*

Caveats: Since we’re running commands using the shell here, don’t have any apostrophes in any parameters to this define. Printer names must not include the strings “not accepting requests” or “disabled since.”

Values you can use for the `model` parameter can be listed using the CUPS command `lpinfo -m`.

```
define cups::printer(
    $model,
    $options,
    $uri,
    $description,
    $location,
    $enable=true,
    $ensure=present,
    ) {

    $options_switches = inline_template("<%=
        options.collect {|k,v|
            \"-o ’#{k}=#{v}’\"}.sort.join(’ ’) %>")

    case $ensure {
        ’present’: {
            exec { "create_printer_${name}":
                command => "lpadmin -p ’${name}’ \
                    -m ’${model}’ \
                    ${options_switches} \
                    -u allow:all \
                    -v ’${uri}’ \
                    -D ’${description}’ \
                    -L ’${location}’",
                creates => "/etc/cups/ppd/${name}.ppd",
            }
            if $enable == true {
                exec { "accept_printer_${name}":
                    command => "cupsaccept ’${name}’",
                    require => Exec["create_printer_${name}"],
                    onlyif => "lpstat -a ’${name}’ | \
                        grep ’not accepting requests’",
                }
                exec { "enable_printer_${name}":
                    command => "cupsenable ’${name}’",
                    require => Exec["create_printer_${name}"],
                    onlyif => "lpstat -p ’${name}’ | \
                        grep ’disabled since’",
                }
            } else {
                exec { "reject_printer_${name}":
                    command => "cupsreject ’${name}’",
                    require => Exec["create_printer_${name}"],
                    unless => "lpstat -a ’${name}’ | \
                        grep ’not accepting requests’",
                }
                exec { "disable_printer_${name}":
                    command => "cupsdisable ’${name}’",
                    require => Exec["create_printer_${name}"],
                    unless => "lpstat -p ’${name}’ | \
                        grep ’disabled since’",
                }
            }
        }
        ’absent’: {
            exec { "remove_printer_${name}":
                command => "lpadmin -x ’${name}’",
                onlyif => "lpstat -p ’${name}’",
            }
```

```
class cups::redhat {
```

Since cups::no uninstalls CUPS, and this class already assumes CUPS is installed, we may as well make sure of it, so that if some node switches from including cups::no to including cups::stig, things will work better. But CUPS is not necessarily all that must be installed for printing to work properly in a given situation.

```
    package { 'cups':
        ensure => present,
    }
    service { 'cups':
        enable => true,
        ensure => running,
        require => Package['cups'],
    }
}
```

### 11.25.4 STIG-required printing configuration

The SRG requirements pertain to the hosts.lpd file. CUPS does not use such a file. The means by which the administrator tells CUPS from what hosts to accept print jobs is the file /etc/cups/cupsd.conf.

Under RHEL, the Common UNIX Printing System (CUPS) is configured by default only to listen to localhost.

RHEL5, RHEL6: GEN003900

```
class cups::stig {
```

First, make sure CUPS is installed and running.

```
    include "cups::${::osfamily}"
```

Control ownership and permissions of the "hosts.lpd (or equivalent) file," in our case cupsd.conf.

(This file has mode 0640 by default, which is less permissive than the required 0664.)

auto: ECLP-1
auto: GEN003920
auto: GEN003930
auto: GEN003940

```
    file { "/etc/cups/cupsd.conf":
        owner => root, group => 0, mode => 0640,
    }
```

Remove extended ACLs on the same file.

auto: ECLP-1
auto: GEN003950

```
    no_ext_acl { "/etc/cups/cupsd.conf": }
}
```

## 11.26 Digihub: automatic action when media inserted

Configure the digihub. This is the piece of Mac OS X that does things when you insert media such as CDs or DVDs into a Mac.

## 11.27 STIG-required digihub configuration

```
class digihub::stig {

    $dh = 'com.apple.digihub'
```

Disable automatic actions when blank CDs are inserted.

We don't strictly conform with the check and fix text here, because this is a Category I requirement, but the check and fix may only fix the systemwide default settings, not enforce the settings on everyone.

auto: ECSC-1
auto: OSX00340 M6
auto: OSX8-00-00085

```
        mcx::set { "${dh}/${dh}.blank.cd.appeared":
            value => 1,
        }
```

§11.61.2

Disable automatic actions when blank DVDs are inserted.

Same as above.

auto: ECCD-1
auto: OSX00341 M6
auto: OSX8-00-00090

```
        mcx::set { "${dh}/${dh}.blank.dvd.appeared":
            value => 1,
        }
```

§11.61.2

Disable automatic actions when music CDs are inserted.

Here the STIG check and fix text have to do with setting things in the System Preferences GUI. With our MCX mechanism we are enforcing policies regarding these preferences; this is the only way to be sure because these preferences are stored and changed on a per-user basis, so setting the local admin user's preference to "do nothing" does not influence the value of any other user's preference. But setting the MCX policy forces the values of these preferences for everyone on the computer.

auto: OSX00345
auto: OSX8-00-00095

```
        mcx::set { "${dh}/${dh}.cd.music.appeared":
            value => 1,
        }
```

§11.61.2

Disable automatic actions when picture CDs are inserted.

auto: ECSC-1
auto: OSX00350 M6
auto: OSX8-00-00100

```
        mcx::set { "${dh}/${dh}.cd.picture.appeared":
            value => 1,
        }
```

§11.61.2

Disable automatic actions when video DVDs are inserted.

auto: ECSC-1
auto: OSX00355 M6
auto: OSX8-00-00105

```
        mcx::set { "${dh}/${dh}.dvd.video.appeared":
            value => 1,
        }
    }
```

§11.61.2

## 11.28 Disable Ctrl-Alt-Del at console

Ensure that "shutdowns" are "configured to ensure that the system remains in a secure state" by preventing an unauthenticated person at the console from rebooting the system.

auto: ECSC-1
auto: GEN000000-LNX00580
auto: DCSS-1

```
class disable_ctrlaltdel {
    case $::osfamily {
        ’RedHat’: {
            case $::operatingsystemrelease {
                /^6\..*/: { require disable_ctrlaltdel::rhel6 }
                /^5\..*/: { require disable_ctrlaltdel::rhel5 }
                default:  { unimplemented() }
            }
        }
        default: { unimplemented() }
    }
}
class disable_ctrlaltdel::rhel5 {
    augeas { ’disable_ctrlaltdel’:
        context => ’/files/etc/inittab’,
        changes => [
```

Remove the comment before ca as well as ca itself.

```
            ’rm #comment[following-sibling::*[1][self::ca]]’,
            ’rm ca’,
        ],
        notify => Exec[’reread_init’],
    }
    exec { ’reread_init’:
        command => ’/sbin/telinit q’,
        refreshonly => true,
    }
}
class disable_ctrlaltdel::rhel6 {
    require augeas
    augeas { "disable_ctrlaltdel":
        context => "/files/etc/init/control-alt-delete.conf",
        changes => [
            ’rm exec’,
            "set exec ’/usr/bin/logger \
-t /etc/init/control-alt-delete.conf \
-p daemon.warning Control-Alt-Delete \
typed at console. Doing nothing.’",
        ],
    }
}
```

## 11.29 DNS

### 11.29.1 Turn off MDNS advertisements

```
class dns::no_mdns_ads {
    include "dns::no_mdns_ads::${::osfamily}"
}
```

```
class dns::no_mdns_ads::darwin {
    $version_underscores = regsubst(
        $::macosx_productversion_major,
        '\D', '_', 'G')
    $klassname = "${::osfamily}_${version_underscores}"
    include "dns::no_mdns_ads::${klassname}"
}
class dns::no_mdns_ads::darwin_10_6 {}
class dns::no_mdns_ads::darwin_10_9 {
    $slld = '/System/Library/LaunchDaemons'
```

Turn off Bonjour multicast advertising on Macs. auto: OSX8-00-00545

```
    mac_plist_value { 'add NoMulticastAdvertisements':
        file => "${slld}/com.apple.mDNSResponder.plist",
        key => 'ProgramArguments',
        value => [
            '/usr/sbin/mDNSResponder',
            '-NoMulticastAdvertisements',
            ],
    }
}
```

At this time we don't have the requirement under Red Hat to disable MDNS advertisements.

```
class dns::no_mdns_ads::redhat {
}
```

## 11.30 DoD Login Warnings

Install notice and consent warnings.

```
class dod_login_warnings {
    case $::osfamily {
        'redhat': {
            include dod_login_warnings::console              §11.30.1
            include dod_login_warnings::gdm                  §11.30.2
            include dod_login_warnings::ssh                  §11.30.4
        }
        'darwin': {
            include dod_login_warnings::mac_loginwindow      §11.30.3
```

Display login banners when the user "connects to the computer remotely," via SSH. auto: ECWM-1 auto: OSX00105 M6

"When a user opens a terminal locally," Mac OS X STIG PDI OSX00105 M6 requires that "the user sees the access warning." But opening a terminal on a Mac does not constitute logging in to the Mac: the user has already done that, and has already been warned by the login window before doing so. Because the requirement is to "display the logon banner *prior* to a logon attempt," we deviate from the published check and fix content here in order to fulfill the spirit of compliance.

```
            include dod_login_warnings::ssh                  §11.30.4
        }
        default: {
            include dod_login_warnings::console              §11.30.1
            include dod_login_warnings::gdm                  §11.30.2
            include dod_login_warnings::ssh                          §11.30.4
        }
    }
}
```

## 11.30.1 Notice of monitoring on the console

Install notice and consent warnings for tty logins. auto: ECWM-1 auto: GEN000400

```
class dod_login_warnings::console {
    file { "/etc/issue":
        owner => root, group => 0, mode => 0644,
        source => "puppet:///modules/dod_login_warnings/80col",
    }
}
```

## 11.30.2 Notice of monitoring via graphical login

Show a warning before the login box under GDM. auto: ECWM-1 auto: GEN000402

This would normally go under §**??**, but because the text of the warning is of legal import and we are inspected on it yearly, it's better to keep everything that uses the warning text in one place.

```
class dod_login_warnings::gdm {
```

First, do no harm.

```
    if($gdm_installed == 'true') {
```

RHEL5 and RHEL6 show the banner differently.

```
        case $osfamily {
            'RedHat': {
                case $operatingsystemrelease {
                    /^6\..*/: {
                        include dod_login_warnings::gdm::rhel6       §11.30.2
                    }
                    /^5\..*/: {
                        include dod_login_warnings::gdm::rhel5       §11.30.2
                    }
                    default: { unimplemented() }
                }
            }
            default: { unimplemented() }
        }
    }
}
```

### Under RHEL5

```
class dod_login_warnings::gdm::rhel5 {
       include zenity                                                §11.118.3
       include ::gdm::rhel5                                          §??
       # This one is for zenity to show. zenity can word-wrap.
```

```
    file { "/etc/issue_paragraphs":
        owner => root, group => 0, mode => 0644,
        source => "puppet:///modules/\
dod_login_warnings/paragraphs",
    }

    exec { ’show_gdm_login_warning’:
        command => "sed -i -e ’/^exit 0$/i \
zenity --error --text \"‘cat /etc/issue_paragraphs‘\"
’ /etc/gdm/Init/Default",
        unless => "grep ’zenity.*error.*issue.*’ \
                   /etc/gdm/Init/Default",
        notify => Exec[’restart_gdm’],
        require => Class[’gdm::logo’],
    }
}
```

### Under RHEL6

In RHEL6, banner functionality is inside gdm.

```
class dod_login_warnings::gdm::rhel6 {
    $agsg = ’/apps/gdm/simple-greeter’
    gconf { "$agsg/banner_message_enable":
        config_source => ’/var/lib/gdm/.gconf’,
        type => bool,
        value => true,
    }
    gconf { "$agsg/banner_message_text":
        config_source => ’/var/lib/gdm/.gconf’,
        type => string,
        value => template(’dod_login_warnings/paragraphs’),
    }
```

All those settings probably created root-owned, solely-root-readable files in gdm’s home directory. We need to let the gdm user read those files.

```
    file { ’/var/lib/gdm/.gconf’:
        owner => gdm, group => gdm,
        recurse => true, recurselimit => 5,
    }
}
```

## 11.30.3 Notice of monitoring on Macs

```
class dod_login_warnings::mac_loginwindow {
    $version_underscores = regsubst(
        $::macosx_productversion_major,
        ’\D’, ’_’, ’G’)
    $klassname = "${::osfamily}_${version_underscores}"
    include "dod_login_warnings::mac_loginwindow::${klassname}"
}
```

### Login warnings on Snow Leopard

Configure the Mac OS Snow Leopard login window to show a login warning.

auto: ECWM-1
auto: OSX00100 M6

```
class dod_login_warnings::mac_loginwindow::darwin_10_6 {
    mac_default { 'mac_login_warnings':
        domain => '/Library/Preferences/com.apple.loginwindow',
        key => 'LoginwindowText',
        source => 'puppet:///modules/dod_login_warnings/paragraphs',
    }
}
```

### Login warnings on Mavericks

Configure the Mac OS Mavericks login window to show a login warning.

auto: OSX8-00-00185
auto: OSX8-00-00195

```
class dod_login_warnings::mac_loginwindow::darwin_10_9 {
    file { '/Library/Security/PolicyBanner.rtf':
        ensure => present,
        owner => root, group => 0, mode => 0644,
        source => 'puppet:///modules/dod_login_warnings/paragraphs.rtf',
    }
}
```

### 11.30.4 Notice of monitoring via SSH

Configure sshd to show a login warning.

auto: ECWM-1
auto: GEN005550

```
class dod_login_warnings::ssh {
    $banner_file = '/etc/issue.ssh'

    file { $banner_file:
        owner => root, group => 0, mode => 0644,
        source => "puppet:///modules/dod_login_warnings/80col",
    }
    class { 'ssh::banner':
        file => $banner_file,
    }
}
```

§11.100.3

## 11.31 Fast user switching

Enable fast user switching on the Mac. This contravenes Mac OS X STIG PDI OSX00330 M6.

The `menu_style` parameter can have values “Name,” “Short Name” or “Icon.”

```
class fast_user_switching($menu_style=’Name’)  {
    $fus_domain = ’/Library/Preferences/.GlobalPreferences’
    mac_default { "$fus_domain:MultipleSessionEnabled":
        type => bool,
        value => true,
    }

    mac_default { "$fus_domain:userMenuExtraStyle":
        type => int,
        value => $menu_style ? {
            ’Name’ => 0,
            ’Short Name’ => 1,
            ’Icon’ => 2,
            default => fail("unknown fast user switching \
menu style $menu_style"),
        },
    }
}
```

### 11.31.1 Disable fast user switching

Disable fast user switching on the Mac.

auto: IAAC-1
auto: OSX00330 M6
auto: OSX8-00-01100

```
class fast_user_switching::no {
    $fus_domain = ’/Library/Preferences/.GlobalPreferences’
    mac_default { "$fus_domain:MultipleSessionEnabled":
        type => bool,
        value => true,
    }
}
```

## 11.32 Filer policy

Our filers store files and make them accessible over the network. There is policy which applies to the filers, but they run proprietary operating systems which cannot run Puppet. So some hosts are designated as *filer policy agents*, given elevated access to the filers (e.g. allowed to NFS mount volume `vol0` on Network Appliance filers), and tasked to enforce the policy.

### 11.32.1 Filer policy agent

On different networks there are different filers. Classes in this namespace define what it means to be a filer policy agent on each network.

```
class filers::agent {}
```

### 11.32.2 Remove the old cron script

An earlier version of this code only supported pushing users and groups to one filer. Remove the file it put in.

```
class filers::remove_old_users_from_agent {
    file { ’/etc/cron.hourly/filers_users_and_groups’:
        ensure => absent,
    }
}
```

### 11.32.3 Get filer users from an agent host

With an integration between Active Directory and UNIX hosts such as Centrify, UNIX users need to be populated to the filer. This define gathers non-system users from a host and places them in group and passwd files in the filer’s `etc` directory, which is indicated by the name of the resource.

```
define filers::users_from_agent($etc_dir, $ensure=’present’) {
    include filers::remove_old_users_from_agent                     §11.32.2
    file { "/etc/cron.hourly/${name}_users_and_groups":
        owner => root, group => 0, mode => 0755,
        content => template(’filers/users_to_filer.cron’),
        ensure => $ensure,
    }
}
```

## 11.33 FIPS 140-2 compliance, general

For compliance with Federal Information Processing Standard (FIPS) 140-2, there are two main ingredients: accreditation and configuration. The cryptographic modules used must be accredited, and they must be used in a compliant manner.

(In some places in this document we say “FIPS compliance.” While we are likely to comply with other FIPS standards, 140-2 is the only one that anyone’s asked about so far, so, for the time being, this is what “FIPS compliance” means.)

```
class fips {
    case $::osfamily {
        ’RedHat’: {
            case $::operatingsystemrelease {
                /^6\..*/: {
                    require fips::rhel6
                }
                /^5\..*/: {
                    require fips::rhel5
                }
                default: { unimplemented() }
            }
        }
        ’Darwin’: {
            require fips::darwin
        }
    }
}
```

```
class fips::darwin {
    warning 'fips mostly unimplemented on darwin'

    file { '/usr/libexec/cc_fips_test':
        audit => all,
    }
}
```

### 11.33.1 RHEL 5 FIPS 140-2 guidance

This is just like RHEL 6 but simpler: the knowledge base article https://access.redhat.com/kb/docs/DOC-39230 applies directly.

See http://www.redhat.com/solutions/industry/government/certifications.html for FIPS approval status of crypto modules in RHEL.

```
class fips::rhel5 {
```

Make sure we have fipscheck: FIPS-compliant OpenSSL uses it to check itself during startup.

```
    package {
        "fipscheck": ensure => present;
        "fipscheck-lib": ensure => present;
    }

    include prelink::no                                   §11.79.1
    include grub::fips                                    §11.40.1
    include ssh::fips                                     §11.100.4
}
```

### 11.33.2 RHEL 6 FIPS 140-2 compliance

The crypto modules in RHEL6 are FIPS Certified; see http://www.redhat.com/solutions/industry/government/certifications.html. Enabling FIPS mode in RHEL6 is documented in Section 8.2 of the Security Guide, https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Security_Guide/sect-Security_Guide-Federal_Standards_And_Regulations-Federal_Information_Processing_Standard.html.

Database management system software included with RHEL uses the cryptographic modules included with RHEL, whose accreditation status is discussed in §11.33. DCNR-1 DG0025

```
class fips::rhel6 {
```

Disable prelinking: it changes the library files, making checksums run against them come out with the wrong results. See the relevant section regarding when the un-prelinking will actually happen.

```
    include prelink::no                                   §11.79.1
```

Make sure we have fipscheck: FIPS-compliant OpenSSL uses it to check itself during startup.

```
    package {
        "fipscheck": ensure => present;
        "fipscheck-lib": ensure => present;
    }
```

Prepare the initramfs for FIPS mode. (The `dracut-fips` package may also be necessary for OpenSSL to successfully initialize in FIPS-compliant mode.)

```
    package { 'dracut-fips':
        ensure => present,
        notify => Exec['recreate initramfs file'],
    }
    exec { 'recreate initramfs file':
        refreshonly => true,
        command => '/sbin/dracut -f',
    }
```

Disable old, unapproved cryptographic algorithms.

```
    include ssh::fips
```
§11.100.4

Ensure that OpenSSH will operate in a FIPS-compliant fashion, by configuring the OpenSSL cryptographic library to run in FIPS 140-2 compliant mode. auto: DCNR-1 auto: GEN005490 auto: GEN005495

Turn on the `fips=1` kernel parameter. This changes how OpenSSL starts up and may effectively disable OpenSSH if you are not properly prepared.

```
    include grub::fips
```
§11.40.1

"Enforced FIPS mode" for gcrypt: Documentation for this mode is in `http://www.gnupg.org/documentation/manuals/gcrypt.pdf`, Appendix B, "Description of the FIPS mode." The reason why not to use it, even though it sounds like a good thing to enable, is written in `https://bugzilla.redhat.com/show_bug.cgi?id=869827`. In short, it breaks *all* SSL/TLS connections. (TLS $\geq 1.2$ could work, but it's only been standardized for four months at this writing. Not practical.)

```
    file { '/etc/gcrypt/fips_enabled':
        ensure => absent,
    }
}
```

The last step in the guide is to reboot the system. From Puppet, we aren't in a position to force this.

In addition to these measures, FIPS mode must also be enabled for each Network Security Services (NSS) database in use; this isn't a useful thing to do for `/etc/pki/nss`, the systemwide NSS database, because it would ask for a password before doing anything interesting, and the password would have to be systemwide. But see §11.7.1 module for how we make sure NSS databases used by Apache httpd's `mod_nss` module are placed into FIPS mode.

# 11.34 File Transfer Protocol (FTP)

The File Transfer Protocol (FTP) is an old, unencrypted protocol, which we do not use.

## 11.34.1 Disable FTP

```
class ftp::no {
```

Remove FTP server software wherever possible.

auto: ECSC-1

```
    package { "vsftpd": ensure => absent }
```

auto: GEN004800
auto: GEN004820
auto: GEN004840

Remove the ftp user so pwck will be happy. Since it's a system uid, chances that it will be reused for a different user are lower; so if ftp happened to own any files they will likely remain secure.

```
    user { "ftp": ensure => absent }

}
```

Where FTP is disabled, the `ftpusers` file likely does not exist, but that isn't a problem.

N/A: GEN004880
N/A: GEN004900
N/A: GEN004920
N/A: GEN004930
N/A: GEN004940
N/A: GEN004950

Where FTP is disabled, the FTP daemon cannot be "configured for logging or verbose mode."

N/A: GEN004980

Since we have no FTP servers, we do no anonymous FTP.

N/A: GEN005000
N/A: GEN005020
N/A: GEN005040

# 11.35 Games

## 11.35.1 Remove fun things

```
class fun::no {
    include "fun::no::${::osfamily}"
}
```

### Remove fun things on Macs

```
class fun::no::darwin {
    $version_underscores = regsubst(
        $::macosx_productversion_major,
        '\D', '_', 'G')
    $klassname = "${::osfamily}_${version_underscores}"
    include "fun::no::${klassname}"
}
   class fun::no::darwin_10_6 {}
   class fun::no::darwin_10_9 {
```

Remove the Chess application from Macs.

auto: OSX8-00-00470

```
    file { '/Applications/Chess.app':
        ensure => absent,
        recurse => true,
        force => true,
    }
```

Remove the Game Center application from Macs. auto: OSX8-00-00480

```
    file { '/Applications/Game Center.app':
        ensure => absent,
        recurse => true,
        force => true,
    }
```

"This requirement is N/A if requirement Apple OS X 10.8 STIG PDI OSX8-00-00480 is met." N/A: OSX8-00-00481

```
}
class fun::no::redhat {}
```

# 11.36 GNOME Display Manager (gdm)

For GDM login warnings, see §11.29.1.

## 11.36.1 Login prompt logos

Configure GDM to show an organization's logo at the login prompt.

The source parameter is used to fetch the image files for the logo. It specifies a Puppet module and directory inside which image files for the logo can be found. As an example, if you write

```
class { 'gdm::logo':
    source => 'puppet:///modules/gdm/logo/afseo',
}
```

then files will be copied from puppet:///modules/gdm/logo/afseo to places under /usr/share/icons. The files placed in the manifest should go in the gdm/files/logo/afseo directory. Inside that directory there should be a logo-48x48.png file and a logo-scalable.png file.

For more details and explanation, consult the governing standards: http://developer.gnome.org/integration-guide/stable/icons.html.en, http://standards.freedesktop.org/icon-naming-spec/latest/, and http://standards.freedesktop.org/icon-theme-spec/icon-theme-spec-latest.html.

```
class gdm::logo($source) {
    if($gdm_installed == 'true') {
        case $osfamily {
            RedHat: {
                case $operatingsystemrelease {
                    /^6\..*/: {
                        class { 'gdm::logo::rhel6':                §11.36.1
                            source => $source,
                        }
                    }
                    /^5\..*/: {
                        class { 'gdm::logo::rhel5':                §11.36.1
```

```
                                source => $source,
                            }
                        }
                        default: { unimplemented() }
                    }
                }
                default: { unimplemented() }
            }
        }
    }
```

### Setting the GDM logo under RHEL5

```
class gdm::logo::rhel5($source) {
    $hic = "/usr/share/icons/hicolor"
    file {
        "$hic/48x48/stock/image/puppet-logo.png":
            owner => root, group => 0, mode => 0644,
            source => "${source}/logo-48x48.png";
        "$hic/scalable/stock/image/puppet-logo.png":
            owner => root, group => 0, mode => 0644,
            source => "${source}/logo-scalable.png";
    }

    $logo = "${hic}/scalable/stock/image/puppet-logo.png"

    require augeas
    augeas { ’gdm_logo’:
        context => ’/files/etc/gdm/custom.conf’,
        changes => [
            ’set daemon/Greeter /usr/libexec/gdmlogin’,
            ’set greeter/DefaultWelcome false’,
```

Don’t “welcome” the user: legalities.

```
                ’set greeter/Welcome "%n"’,
                "set greeter/Logo ${logo}",
                ],
        }
    }
```

### Setting the GDM logo under RHEL6

```
class gdm::logo::rhel6($source) {
    $agsg = ’/apps/gdm/simple-greeter’
    gconf { "$agsg/logo_icon_name":
        config_source => ’/var/lib/gdm/.gconf’,
        type => string,
        value => ’puppet-logo’,
    }

    $hic = "/usr/share/icons/hicolor"
    file {
        "$hic/48x48/stock/image/puppet-logo.png":
            owner => root, group => 0, mode => 0644,
            source => "${source}/logo-48x48.png",
            notify => Exec[’gdm_logo_update_icon_cache’];
        "$hic/scalable/stock/image/puppet-logo.png":
            owner => root, group => 0, mode => 0644,
            source => "${source}/logo-scalable.png",
            notify => Exec[’gdm_logo_update_icon_cache’];
    }

    exec { ’gdm_logo_update_icon_cache’:
        command => "/usr/bin/gtk-update-icon-cache $hic",
        refreshonly => true,
    }

}
```

## 11.36.2 Remove user list

Prevent GDM from showing a list of possible users to log in as.

```
class gdm::no_user_list {
    if($gdm_installed == ’true’) {
        case $osfamily {
            RedHat: {
                case $operatingsystemrelease {
                    /ˆ6\..*/: {
                        include gdm::no_user_list::rhel6                    §11.36.2
                    }
```

GDM 2 (RHEL5) doesn’t do user lists.

```
                    /ˆ5\..*/: { }
                    default: { unimplemented() }
                }
            }
            default: { unimplemented() }
        }
    }
}
```

### Removing GDM user list under RHEL6

```
class gdm::no_user_list::rhel6 {
    $agsg = ’/apps/gdm/simple-greeter’
    gconf { "$agsg/disable_user_list":
        config_source => ’/var/lib/gdm/.gconf’,
        type => bool,
        value => true,
    }
}
   class gdm::rhel5 {
       exec { ’restart_gdm’:
           command => ’/usr/sbin/gdm-safe-restart’,
           refreshonly => true,
       }
   }
```

## 11.36.3 STIG-required configuration

The way to configure GDM and the X servers it starts varies between RHEL5 and RHEL6.

```
   class gdm::stig {
       if($gdm_installed == ’true’) {
           case $osfamily {
               RedHat: {
                   case $operatingsystemrelease {
                       /^6.*/: { include gdm::stig::rhel6 }
                       /^5.*/: { include gdm::stig::rhel5 }
                       default: { unimplemented() }
                   }
               }
               default: { unimplemented() }
           }
       }
   }
```

### Under RHEL5

```
class gdm::stig::rhel5 {
```

Make sure the file we’re about to edit exists: if we have no custom options set yet, it won’t.

```
       file { "/etc/gdm/custom.conf":
           ensure => present,
           owner => root, group => 0, mode => 0644,
       }
```

Set the right X server options (`-s` [screensaver timeout], `-audit` [audit level], and `-auth` [authorization record file], which “gdm always automatically uses”), and don’t set the wrong ones (`-ac` [disable host-based access control], `-core` [dump core on fatal errors], and `-nolock` [unknown, not in man page]). (The `-br` option merely makes the screen black by default when the server starts up, instead of the gray weave pattern.)

auto: ECSC-1
auto: GEN000000-LNX00360
auto: ECSC-1
auto: GEN000000-LNX00380

```
    require augeas
    augeas { "gdm_servers_switches":
        require => File["/etc/gdm/custom.conf"],
        context => "/files/etc/gdm/custom.conf/server-Standard",
```

Copied from Red Hat 5 STIG fix text.

```
        changes => [
            "set command '/usr/bin/Xorg -br -audit 4 -s 15'",
            "set name 'Standard server'",
            "set chooser false",
            "set handled true",
            "set flexible true",
            "set priority 0",
        ],
    }
}
```

**Under RHEL6**

GDM X server startup requirements appear to be unimplementable under RHEL6.

GEN000000-LNX00360
GEN000000-LNX00380

RHEL 6 contains `gdm` 2.30. At `2.22`, GDM was rewritten, and no longer pays attention to the server-startup-related sections of `/etc/gdm/custom.conf`. See `https://bugzilla.redhat.com/show_bug.cgi?id=452528`, `http://live.gnome.org/GDM/2.22/Configuration`. It appears that the command-line switches `-br -verbose` are hard-coded into `/usr/libexec/gdm-simple-slave`.

I have filed RHBZ 773111 about this. `https://bugzilla.redhat.com/show_bug.cgi?id=773111`

```
class gdm::stig::rhel6 {}
```

## 11.37 Gluster

A distributed filesystem.

### 11.37.1 Gluster with Automount

As of 3.6.0.29-2.el6, `glusterfs` when used with automount fails to mount the requested filesystem. If you turn up the debugging on autofs enough, you find this error:

```
/sbin/mount.glusterfs: line 13: /dev/stderr: Permission denied
```

This boils down to an AVC denial. An SELinux module that allows the required behavior is provided here. Include the class to install the SELinux module.

```
class gluster::automount {
    require ::automount
    $selmoduledir = "/usr/share/selinux/targeted"
    file { "${selmoduledir}/gluster_automount.pp":
        owner => root, group => 0, mode => 0644,
        source => "puppet:///modules/gluster/\
gluster_automount.selinux.pp",
    }
    selmodule { "gluster_automount":
       ensure => present,
       syncversion => true,
       notify => Service['autofs'],
    }
}
```

## 11.38 GNOME Screensaver

Configure the GNOME screensaver.

### 11.38.1 STIG-required screensaver configuration

```
class gnome-screensaver::stig {
```

All settings we are about to set should go in the mandatory GConf tree. And that is the default for this resource type.

```
    gconf {
```

Make sure the screensaver will only show something publicly viewable, such as a blank screen. RHEL6 does not ship with any screensavers that could show anything not publicly viewable.

RHEL6: GEN000510

```
        "/apps/gnome-screensaver/mode":
            ensure => absent;
```

Cause the screen to lock after 15 minutes of inactivity, requiring re-authentication to unlock it.

auto: PESL-1
auto: GEN000500

```
        "/apps/gnome-screensaver/idle_activation_enabled":
            type => bool, value => true;
```

Enable the lock setting of the screensaver.

auto: PESL-1
auto: GEN000500-3

```
        "/apps/gnome-screensaver/lock_enabled":
            type => bool, value => true;
```

Set the screensaver idle delay to 15 minutes.

auto: PESL-1
auto: GEN000500-2

```
        "/apps/gnome-screensaver/idle_delay":
            type => int, value => 15;
    }
}
```

## 11.39 Graphical login

Some hosts should have graphical login. Others should not. This class enables or disables that feature.

This class only turns graphical login on or off; it does not apply STIG-related requirements to the mechanism of graphical login. See §11.35.1 for that.

```
class graphical_login {
    case $::osfamily {
        ’RedHat’: {
            package { ’gdm’:
                ensure => installed,
            }
```

Fortunately this is the one thing RHEL5 and RHEL6 have in common between their init systems.

```
            augeas { ’default_runlevel_5’:
                context => ’/files/etc/inittab’,
                changes => ’set id/runlevels 5’,
            }
        }
```

Mac OS X always has graphical login.

```
        ’Darwin’: {}
        default: { unimplemented() }
    }
}
```

### 11.39.1 Disable graphical login

This class is Red Hat-centric.

```
class graphical_login::no {
    augeas { ’default_runlevel_3’:
        context => ’/files/etc/inittab’,
        changes => ’set id/runlevels 3’,
    }
}
```

## 11.40 GRUB

### 11.40.1 Enable FIPS-compliant kernel mode

See §11.33.

```
class grub::fips {
    $g = "/boot/grub/grub.conf"
    exec { "fipsify_kernel_cmdlines":
        path => "/bin:/sbin",
        onlyif => "grep ’^[[:space:]]*kernel’ $g | \
                   grep -v fips=1 >&/dev/null",
        command => "sed -i.fips -e \
            ’/^[[:space:]]*kernel/s/\$/ fips=1/’ $g",
        logoutput => true,
    }
```

Warning: this probably won’t work right with EFI. See https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Security_Guide/chap-Federal_Standards_and_Regulations.html.

```
    exec { "bootify_kernel_cmdlines":
        path => ’/bin:/sbin’,
        onlyif => "grep ’^[[:space:]]*kernel’ $g | \
                   grep -v boot=${::boot_filesystem_device} \
                       >&/dev/null",
        command => "sed -i.fips2 -e \
            ’/^[[:space:]]*kernel/s!\$! boot=${::boot_filesystem_device}!’ $g",
        logoutput => true,
    }
}
```

### 11.40.2 Nouveau

The initrd may load the Nouveau driver on hosts having NVIDIA graphics adapters. Once this driver sets the graphics mode, it cannot be unloaded, because it is ”in use.” But the NVIDIA proprietary drivers will not install or run if the Nouveau driver is active.

#### Disable Nouveau driver in initrd

This action is originally documented in the README for the NVIDIA driver.

```
class grub::nouveau::no {
    $g = "/boot/grub/grub.conf"
    exec { "disable_nouveau_kernel_cmdlines":
        path => "/bin:/sbin",
        onlyif => "grep ’^[[:space:]]*kernel’ $g | \
                   grep -v nouveau >&/dev/null",
        command => "sed -i.disable_nouveau -e \
                    ’/[[:space:]]*kernel/s/\$/ rdblacklist=nouveau /’ $g",
        logoutput => true,
    }
}
```

### 11.40.3 Ensure authentication required

Make sure that authentication is required before changing bootloader settings. auto: IAIA-1

If you follow the procedures in §**??**, you should end up with a bootloader password at OS install time. This, then, is either a failsafe measure, or a means by which you can easily change bootloader passwords across many hosts. auto: GEN008700

Example invocation:

```
class { ’grub::password’:
    md5_password => ’d3b07384d113edec49eaa6238ad5ff00’,
}
```

This results in a line like this in GRUB’s configuration:

```
password --md5 d3b07384d113edec49eaa6238ad5ff00
```

* * *

```
class grub::password($md5_password) {
    case $::osfamily {
        'RedHat': {
            augeas { "ensure_grub_password":
```

Augeas knows how to edit `/etc/grub.conf` but maybe not `/boot/grub/menu.lst` or some such: it goes by filename.

```
                context => '/files/etc/grub.conf',
                changes => [
```

Grub's behavior regarding passwords appears to differ depending on where in the configuration the password directive is written, but the Augeas lens which understands the Grub configuration doesn't make that order information easily available to us.

Previously we just set the password, which would insert a password line at the end of the Grub configuration if there was no password line already. That did the wrong thing. So we get rid of those, if any, and put one at the top of the file.

```
                    "rm password",
                    "ins password before default",
                    "set password '$md5_password'",
                    'clear password/md5',
                ],
            }
        }
```

Mac OS X doesn't have grub.

```
        'Darwin': {}
        default: { unimplemented() }
    }
}
```

## 11.40.4 Red Hat graphical boot

The Red Hat graphical boot is a splash screen that covers the details of the system's boot process. But it may use video drivers, and we may want to change things about video drivers at boot time.

### Disable Red Hat graphical boot

This is so that the video driver will certainly not be in use at boot time, so we can install the NVIDIA driver if necessary.

```
class grub::rhgb::no {
    $g = "/boot/grub/grub.conf"
    exec { "disable_rhgb_kernel_cmdlines":
        path => "/bin:/sbin",
        onlyif => "grep ’^[[:space:]]*kernel’ $g | \
                   grep -v rhgb >&/dev/null",
        command => "sed -i.disable_rhgb -e \
                    ’/[[:space:]]*kernel/s/\\<rhgb\\>//’ $g",
        logoutput => true,
    }
}
```

### 11.40.5 Enable serial console

See §11.93.

```
class grub::serial_console($speed=9600) {
```

First, make all the kernels treat the serial port as the console.

```
    $g = "/boot/grub/grub.conf"
    exec { "serial_console_ify_kernel_cmdlines":
        path => "/bin:/sbin",
        onlyif => "grep ’^[[:space:]]*kernel’ $g | \
                   grep -v console=ttyS0,${speed}n8 >&/dev/null",
        command => "sed -i.serial_console_kernels -e \
            ’/[[:space:]]*kernel/s/\$/ console=tty console=ttyS0,${speed}n8 /’ $g",
        logoutput => true,
    }
```

Then, make grub itself treat the serial port as the console.

Regarding the terminal command: ”When both the serial port and the attached monitor and keyboard are configured they will both ask for a key to be pressed until the timeout expires. If a key is pressed then the boot menu is displayed to that device. Disconcertingly, the other device sees nothing.”

```
    exec { "serial_console_ify_grub":
        path => "/bin:/sbin",
        unless => "grep ^serial $g",
        command => "sed -i.serial_console_grub -e \
            ’/[[:space:]]*default/i \\\n\
serial --unit=0 --speed=${speed} \
 --word=8 --parity=no --stop=1\\\n\
terminal --timeout=10 serial console\n\
’ $g",
        logoutput => true,
    }
}
```

### 11.40.6 STIG-required configuration

```
class grub::stig {
```

Turn on auditing in time to audit the actions of startup scripts.

auto: ECSC-1
auto: GEN000000-LNX00720

```
    $g = "/boot/grub/grub.conf"
    exec { "auditify_kernel_cmdlines":
        path => "/bin:/sbin",
        onlyif => "grep '^[[:space:]]*kernel' $g | \
                   grep -v audit=1 >&/dev/null",
        command => "sed -i.audit -e \
            '/[[:space:]]*kernel/s/\$/ audit=1/' $g",
        logoutput => true,
    }
```

Make sure the configuration file /boot/grub/menu.lst is owned by root, group-owned by root, has permissions 0600, and has no extended ACL.

admins do GEN008720
admins do GEN008740
admins do GEN008760
admins do GEN008780

```
    file { $g:
        owner => root, group => 0, mode => 0600,
    }
    no_ext_acl { $g: }
}
```

## 11.41 Home directories

Apply policies to the home directories of users.

This is harder than it sounds, mostly because the set of home directories varies from host to host, and no policy can be applied to them all as a whole, but they must each be treated separately.

In accordance with UNIX SRG PDI GEN003620, there is a separate file system for user home directories, /home; so the custom fact *home_perms* is the collection of home directories listed in /etc/passwd which are under /home, along with the user ID and primary group ID of its rightful owner.

Since Facter only makes facts which are strings, but we need a list of triples, delimiters are inserted into the *home_perms* fact, and here in the home class we split the fact back up. A further restriction is that when arrays are used to define multiple resources in Puppet, it appears that further parameters unique to each resource cannot be provided; so all of the pieces of data needed must be squished into the resource's name. So the name of a home directory resource looks like /home/user:uid:gid, and any defined resource types must use the split function to take this apart. In this way, each home directory along with its rightful owner and group can make its way from the /etc/passwd file, through Facter, into Puppet as an instance of one or more home::* defined resource types.

```
class home {
    $home_perms_a = split($home_perms, ',')
    home::quick { $home_perms_a: }                    §11.41.3
    home::slow { $home_perms_a: }                     §11.41.4
}
```

We have NFS-mounted home directories on most of our hosts, and all of the normal ones do not have root access to that NFS share (UNIX SRG PDI GEN005880 is related to this issue, but our NFS servers do not run UNIX).

In future a host will be dedicated to applying policies to NFS homes. For now we limit ourselves to enforcing the policies against local homes.

### 11.41.1 Admin guidance about home directories

Administrators, “educate users about the danger of having terminal messaging set on.” admins do GEN001960

### 11.41.2 User guidance about home directories

The SRG imposes requirements on the contents of local initialization files, which cannot be programmatically enforced without an extraordinarily severe uniformity, nor automatically checked for. These files are `$HOME/.bashrc`, `$HOME/.profile` and the like. You are responsible for fulfilling these requirements:

Do not add an entry to your `PATH` which is not an absolute path. This prohibition includes `.`, the current directory. users do GEN001900

Do not add an entry to your `LD_LIBRARY_PATH` which is not an absolute path. users do GEN001901

Do not set the `LD_PRELOAD` environment variable. users do GEN001902

Do not execute world-writable programs from your local initialization files. If you build programs, make sure they don’t end up world-writable. users do GEN001940

Do not place the command `mesg y` in your startup files. users do GEN001960

Do not set the `PGPASSFILE` environment variable. users do IAIA-1

users do DG0067

### 11.41.3 Quick-to-enforce home policies

This defined resource type contains policies regarding the home directory that can likely be enforced in under five seconds per home directory.

```
define home::quick() {
    $s = split($name, ':')
    $dir = $s[0]
    $uid = $s[1]
    $gid = $s[2]

    File {
        owner => $uid, group => $gid, mode => 0640,
    }

    file {
```

Secure home directories. auto: ECLP-1 auto: GEN001480 auto: GEN001500 auto: GEN001520

```
        "${dir}":
            ensure => directory,
            recurse => false,
            mode => 0700;
```

Secure local initialization files. auto: ECLP-1 auto: GEN001860 M6 auto: ECLP-1 auto: GEN001860 auto: GEN001870 auto: GEN001880

```
        "${dir}/.bash_profile":;
        "${dir}/.bash_login":;
        "${dir}/.profile":;
        "${dir}/.bashrc":;
        "${dir}/.bash_logout":;

        "${dir}/.tcshrc":;
        "${dir}/.cshrc":;
        "${dir}/.history":;
        "${dir}/.login":;
        "${dir}/.logout":;
        "${dir}/.cshdirs":;
```

Additional required by Mac OS X STIG.

```
        "${dir}/.env":;
        "${dir}/.dtprofile":;
        "${dir}/.dispatch":;
```

This is likely a directory, but Puppet will do the right thing with the execute bits.

```
        "${dir}/.emacs":;
        "${dir}/.exrc":;
```

Remove `.rhosts` and `.shosts` files from home directories.

auto: ECCD-1
auto: GEN001980
auto: GEN002040
N/A: GEN002020
N/A: GEN002060

```
        "${dir}/.rhosts":
            ensure => absent;
        "${dir}/.shosts":
            ensure => absent;
```

Remove `.netrc` files from home directories.

auto: ECCD-1
auto: IAIA-1
auto: GEN002000 M6
auto: OSX8-00-00600
auto: IAIA-1
auto: GEN002000

```
        "${dir}/.netrc":
            ensure => absent;
    }

    no_ext_acl {
```

Remove extended ACLs for local initialization files.

auto: ECLP-1
auto: GEN001890

```
        "${dir}/.bash_profile":;
        "${dir}/.bash_login":;
        "${dir}/.profile":;
        "${dir}/.bashrc":;
        "${dir}/.bash_logout":;

        "${dir}/.tcshrc":;
        "${dir}/.cshrc":;
        "${dir}/.history":;
        "${dir}/.login":;
        "${dir}/.cshdirs":;
    }
```

Prevent use of the `.forward` file by removing it.

auto: ECSC-1
auto: GEN004580 M6
auto: OSX8-00-01040

```
    file { "${dir}/.forward": ensure => absent }
```

Prevent use of the `.pgpass` file, which could contain unencrypted passwords for the PostgreSQL DBMS.

auto: ECSC-1
auto: GEN004580
auto: IAIA-1
auto: DG0067

```
    file { "${dir}/.pgpass": ensure => absent }
```

Get rid of signed-in Apple IDs for iCloud (previously MobileMe, eh). auto: OSX8-00-01130

```
    $mma = "${dir}/Library/Preferences/MobileMeAccounts.plist"
    exec { "warn of possible signed-in Apple IDs in ${dir}":
        onlyif => "stat ${mma}",
        command => "echo ${mma} exists. \
This may indicate a signed-in Apple ID in violation of the STIG.",
        loglevel => err,
    }
}
```

### 11.41.4 Slow-to-enforce home directory policies

This defined resource type contains policies that will likely take minutes or longer to enforce for a user with many files.

```
define home::slow() {
    $s = split($name, ':')
    $dir = $s[0]
    $uid = $s[1]
    $gid = $s[2]
```

Control ownership and permissions on files contained in home directories.

auto: ECCD-1
auto: ECLP-1
auto: GEN001540 M6
auto: GEN001550 M6
auto: ECCD-1
auto: ECLP-1
auto: GEN001540
auto: GEN001550
auto: GEN001560

It appears that "contained in" is intended to mean *anywhere under* the home directory. File resources seem to run slowly and take a lot of memory in the case of thousands of files; so we use `find`, `xargs`, `chown` and `chmod`. (See 11.101.8 for more details on this phenomenon.)

The `-r` switch to xargs is a GNU extension which does not run the given command if there are no arguments to run it with. According to the man page, "Normally, the command is run once even if there is no input."

Under Mac OS X, the xargs command does not accept the `-r` switch, but it appears that if there are no arguments to consume, xargs will not run the given command. That behavior may be documented by this sentence: "The xargs utility exits immediately... if a command line cannot be assembled..."

```
    $xargs0 = $osfamily ? {
        darwin  => "xargs -0",
        default => "xargs -0 -r",
    }
    exec { "chown_${uid}_home_files":
        path => [’/bin’, ’/usr/bin’],
        command => "find ’${dir}’ -mindepth 1 \\( \
                        \\! -user ${uid} -o \\! -group ${gid} \
                        \\) -print0 | \
                    ${xargs0} chown ${uid}:${gid}",
        onlyif => ["test -d ’${dir}’",
               "find ’${dir}’ -mindepth 1 \
                   \\! -user ${uid} -o \\! -group ${gid} | \
                grep . >&/dev/null"],
    }
    exec { "chmod_${uid}_home_files":
        path => [’/bin’, ’/usr/bin’],
        command => "find ’${dir}’ -mindepth 1 \\
                        \\! -type l -perm +026 -print0 | \
                    ${xargs0} chmod g-w,o-rw",
        onlyif => ["test -d ’${dir}’",
                "find ’${dir}’ -mindepth 1 \\
                     \\! -type l -perm +026 | \
                 grep . >&/dev/null"],
    }
```

Remove extended ACLs on home directories, and all files and directories therein.

auto: ECLP-1
auto: GEN001490 M6
auto: GEN001570 M6

```
    no_ext_acl { "${dir}": recurse => true }
}
```

auto: ECLP-1
auto: GEN001490
auto: GEN001570

### 11.41.5 Hot corners

Configure “hot corners” on Macs, that is, actions that happen when the mouse pointer is moved to a corner of the screen and left there for a couple of seconds.

The `hot_corner` resource defined below makes a computer-wide policy for what action should be attached to one of the corners of the screen.

The name of a `hot_corner` resource is one of the four strings `tl`, `tr`, `bl` or `br`, denoting which corner of the screen we’re talking about. `action` is one of the keys in the settings hash below.

Example:

```
hot_corner { ’tl’:
```

* * *

```
define hot_corner($action) {
```

These settings were derived under Snow Leopard by changing the settings in System Preferences, and reading them out using `defaults(1)`.

```
    $settings = {
        'nothing'             => 1,
        'all-windows'         => 2,
        'application-windows' => 3,
        'desktop'             => 4,
        'dashboard'           => 7,
        'spaces'              => 8,
        'start-screensaver'   => 5,
```

Don't configure any of the corners to disable the screensaver. Don't.

```
#           'disable-screensaver' => 6,
        'sleep-display'       => 10,
    }

    mcx::set { "com.apple.dock/wvous-${name}-corner":                §11.61.2
        value => $settings[$action],
    }
```

Not sure exactly what the modifier means; this is just what showed up in the `defaults(1)` when a corner was set to no action.

```
    mcx::set { "com.apple.dock/wvous-${name}-modifier":              §11.61.2
        value => $action ? {
            'nothing' => 1048576,
            default   => 0,
        },
    }
}
```

### 11.41.6 Prevent users from disabling screensaver

```
class hot_corner::stig {
```

Prevent users from configuring a hot corner to disable the screensaver.

Another way to do this besides disabling all hot corners would be to force the hot corner configuration to something known to be compliant.

auto: PESL-1
auto: OSX00375 M6
auto: OSX8-00-01095

```
    hot_corner {
        'tl': action => 'nothing';
        'tr': action => 'nothing';
        'bl': action => 'nothing';
        'br': action => 'nothing';
    }
}
```

## 11.42 HPC Clustering

Configure HPC clusters with login nodes on the production network, and management nodes behind the login nodes, in a particular style.

Besides offering users on the production network access to the cluster, the login node also forwards the services of the production network inside the cluster network, so that security updates, policy enforcement, and unified authentication and authorization can happen without the cluster management software being exposed on the production network.

We assume here that the cluster's internal network has a subnet `X.Y.0.0/16`. We give the cluster's internal network a DNS subdomain named for the cluster's login node(s); this subdomain is visible only inside the cluster. Inside that network and DNS subdomain, we have the following common subnets, addresses and hostnames:

- Subnet 0: management
  - X.Y.0.1: `head`, the IP address belonging to whichever head is active among the redundant head nodes
  - X.Y.0.2: `head1`, the first head node
  - X.Y.0.3: `head2`, the second head node (and so on)
- Subnet 1: login nodes
  - X.Y.1.1: `login`, the internal IP address beloging to whichever login node is active among the redundant login nodes
  - X.Y.1.2: `login1`, the internal IP address of the first login node
  - X.Y.1.3: `login2`, the internal IP address of the second login node (and so on)
- Subnet 50 (and beyond, if needed): compute nodes

Furthermore, we assume another subnet `X.Z.0.0/16`, where $Z$ is usually $Y + 1$, used for Infiniband.

Settings on the outside of the cluster are not set here. For example, if you have a cluster with two login nodes known on the production network as `fnord1` and `fnord2`, you'll need to set up DNS for each of these outside this class, as well as whatever mechanism makes it possible for them all to show up as host `fnord`, and users who attempt access to be shunted to one login node or another.

### 11.42.1 Login node

To serve its internal network a login node must make available NTP, DNS, HTTP, HTTPS, Puppet, and likely NFS. We do this as far as possible without packet forwarding, because it seems usual in the DoD to avoid configurations that, while easy, may make it less clear which hosts are generating traffic and which forwarding it.

The `cluster_hostname` parameter is used in other resources to identify the cluster we're talking about, so it should be unique across all cluster hostnames in your Puppet manifest. The default value for this is the hostname of the cluster login node. If your cluster login nodes are called `fnord1`, `fnord2`, etc., you'll have to set cluster_hostname to fnord manually, and cluster_fqdn to fnord.example.com.

internal_ipv4_first_two_octets should be set to the first two octets of the cluster's internal network, delimited by a dot, like "10.24".

internal_ipv4_address is the internal IPv4 address of this login node; follow the cluster IP address plan in §11.42.4.

internal_infiniband_ipv4_first_two_octets is the subnet to use for Infiniband; this should normally be one more than internal_ipv4_first_two_octets, such that if the latter is 10.24, the former is 10.25.

```
class hpc_cluster::login_node(
        $internal_ipv4_first_two_octets,
        $internal_ipv4_address,
        $use_infiniband = 'false',
        $internal_infiniband_ipv4_first_two_octets,
        $internal_infiniband_ipv4_address,
        $compute_node_count,
        $compute_node_third_octet = "50",
        $cluster_hostname = $::hostname,
        $cluster_fqdn = $::fqdn,
        $external_interface = 'eth0',
        $internal_interface = 'eth1',
        $infiniband_interface = 'ib0',
        ) {
```

$cluster_hostname is used in the hpc_cluster::node class to collect resources exported by this class, so having multiple clusters with the same hostname in different domains in the same sphere of Puppet management is not supported by this module.

```
    tag $cluster_hostname

    $dnsmasq_hosts_file = '/etc/dnsmasq.hosts'
    $iifto = $internal_ipv4_first_two_octets
    $iibifto = $internal_infiniband_ipv4_first_two_octets
    $internal_ipv4_subnet = "${iifto}.0.0/16"
    $internal_ipv4_with_netmask = "${iifto}.0.0/255.255.0.0"
    $compute_node_first_three_octets = "${iifto}.${compute_node_third_octet}"
    $compute_node_infiniband_first_three_octets = "${iibifto}.${compute_node_third_octet}"
    $login_internal_ipv4 = "${iifto}.1.1"

    $login1_fqdn = inline_template("<%=
        short_name, domain = cluster_fqdn.split('.', 2);
        short_name + '1.' + domain %>")
    $login2_fqdn = inline_template("<%=
        short_name, domain = cluster_fqdn.split('.', 2);
        short_name + '2.' + domain %>")
```

Cheat: we assume RHEL 6 here, because its default squid configuration is very close to what we want. And no one wants an hpc_cluster::login_node that isn’t running RHEL6, yet.

```
    if $::osfamily != ’RedHat’ or $::operatingsystemrelease !˜ /ˆ6\..*/ {
        unimplemented()
    }
```

Make DNS available on the internal network. This will include whatever is written in the `/etc/hosts` file on the login node—which we will get to shortly.

```
    package { ’dnsmasq’:
        ensure => installed,
    }

    augeas { ’dnsmasq for cluster login’:
        context => ’/files/etc/dnsmasq.conf’,
        changes => [
            "set interface         ${internal_interface}",
```

Don’t serve DHCP: the management node will do that.

```
            "set no-dhcp-interface ${internal_interface}",
```

Don’t bind to every interface, but only the ones given above. This seems to resonate with security principles originally espoused in the Apache httpd STIG. (`clear` means don’t set a value, but make sure it exists.)

```
            "clear bind-interfaces",
            "clear expand-hosts",
```

We’ll set up a subdomain by the name of the cluster. This way we get to use generic names inside the subdomain, like `head1`.

```
            "set domain            ${cluster_fqdn}",
        ],
        require => Package[’dnsmasq’],
        notify => Service[’dnsmasq’],
    }
```

We need to know the IPs of compute nodes on the login node, so we can ssh to them, so we can support interactive jobs like debuggers. The management node knows this information, but under Scyld it doesn’t share the information in a way the login node can consume it, so we have to write this in /etc/hosts on the login node.

But dnsmasq usually serves everything in /etc/hosts up using DNS. We don’t want the compute nodes to be able to get their own addresses both from the master node via bproc and from the login node via DNS: grief lies that way. So we need to make dnsmasq serve information from a separate file, not /etc/hosts.

```
augeas { 'dnsmasq hosts file setting':
    context => '/files/etc/dnsmasq.conf',
    changes => [
        $dnsmasq_hosts_file ? {
            '/etc/hosts' => '',
            default      => 'clear no-hosts',
        },
        "set addn-hosts ${dnsmasq_hosts_file}",
        ],
    require => Package['dnsmasq'],
    notify => Service['dnsmasq'],
}

service { 'dnsmasq':
    enable => true,
    ensure => running,
}
```

NTP is taken care of by NTP classes which are specific to the network where the cluster lives. That will include the `ntp` module (11.70.1).

Here's how we share NTP with the inside of the cluster network:

```
@@augeas { "${cluster_hostname} ntp.conf":
    context => "/files/etc/ntp.conf",
    changes => [
```

Remove comments about pool.ntp.org: they are not useful here.

```
        "rm #comment[. =~ regexp('.*pool.ntp.org.*')]",
        "rm server",
        "set server[1] login1",
```

If login2 doesn't exist, ntpd won't mind much.

```
        "set server[2] login2",
    ],
}
```

Set up some addresses inside the cluster.

The entry containing the `cluster_fqdn` is pretty special, because it appears Centrify uses the canonical name on that line as the hostname when joining Active Directory. So if you have

```
 x.y.z.w   flarble the.hosts.fqdn
```

Centrify should rightfully use `the.hosts.fqdn` when joining AD, but instead it uses `flarble`. So the FQDN has to come first on the line.

These host entries should be both in the dnsmasq.hosts and hosts files, so we write them in a variable.

```
    $cluster_login_nodes_gbe_host_entries_script = "
rm *[canonical=’head’]
set 990/ipaddr    ${iifto}.0.1
set 990/canonical head
set 990/alias     head.${cluster_fqdn}
rm *[canonical=’head1’]
set 991/ipaddr    ${iifto}.0.2
set 991/canonical head1
set 991/alias     head1.${cluster_fqdn}
rm *[canonical=’head2’]
set 992/ipaddr    ${iifto}.0.3
set 992/canonical head2
set 992/alias     head2.${cluster_fqdn}
rm *[canonical=’login’]
rm *[canonical=’${cluster_fqdn}’]
set 993/ipaddr    ${login_internal_ipv4}
set 993/canonical ${cluster_fqdn}
set 993/alias[1]  login
set 993/alias[2]  login.${cluster_fqdn}
rm *[canonical=’login1’]
rm *[canonical=’${cluster_fqdn}’]
set 994/ipaddr    ${iifto}.1.2
set 994/canonical ${cluster_fqdn}
set 994/alias[1] login1
set 994/alias[2]  login1.${cluster_fqdn}
set 994/alias[3]  ${login1_fqdn}
rm *[canonical=’login2’]
set 995/ipaddr    ${iifto}.1.3
set 995/canonical login2
set 995/alias[1]  login2.${cluster_fqdn}
set 995/alias[2]  ${login2_fqdn}
"

    $cluster_login_nodes_infiniband_host_entries_script = "
rm *[canonical=’head1-ib’]
set 980/ipaddr    ${iibifto}.0.2
set 980/canonical head1-ib
set 980/alias     head1-ib.${cluster_fqdn}
rm *[canonical=’head2-ib’]
set 981/ipaddr    ${iibifto}.0.3
set 981/canonical head2-ib
set 981/alias     head2-ib.${cluster_fqdn}
"

    $cluster_login_nodes_host_entries_script = $use_infiniband ? {
        ’true’ => "
${cluster_login_nodes_gbe_host_entries_script}
${cluster_login_nodes_infiniband_host_entries_script}
",
        ’false’ => "
${cluster_login_nodes_gbe_host_entries_script}
",
    }
```

Get the node IP addresses in the login node's `/etc/hosts` file. These are needed for a few different things: (a) if you have Grid Engine interactive jobs, qsub needs to ssh to one of these addresses when you submit one of those; and (b) if you are mounting a Gluster volume using the Gluster client, the login node needs to speak to any node that has a brick on it, and for that to happen, both forward and reverse name lookups need to work OK.

Assumption: you don't have 200 hosts already, and you don't have more than 200 compute nodes.

```
    $compute_nodes_host_entries_script = inline_template("
<% 0.upto(@compute_node_count.to_i - 1) do |nodenumber| %>
rm *[canonical='n<%=nodenumber -%>.${cluster_fqdn}']
set <%= nodenumber + 200 -%>/ipaddr ${compute_node_first_three_octets}.<%=nodenumber %>
set <%= nodenumber + 200 -%>/canonical n<%=nodenumber -%>.${cluster_fqdn}
set <%= nodenumber + 200 -%>/alias[1]  n<%=nodenumber %>
<%   if @use_infiniband == 'true' %>
rm *[canonical='n<%=nodenumber -%>-ib.${cluster_fqdn}']
set <%= nodenumber + 400 -%>/ipaddr ${compute_node_infiniband_first_three_octets}.<%=nodenumber %>
set <%= nodenumber + 400 -%>/canonical n<%=nodenumber -%>-ib.${cluster_fqdn}
set <%= nodenumber + 400 -%>/alias[1]  n<%=nodenumber -%>-ib
<%   end %>
<% end %>
")

    $host_entries_on_login_node = "
${cluster_login_nodes_host_entries_script}
${compute_nodes_host_entries_script}
"

    augeas { "${cluster_hostname}_internal_hosts":
        context => "/files/${dnsmasq_hosts_file}",
        incl => $dnsmasq_hosts_file,
        lens => 'Hosts.lns',
        changes => $cluster_login_nodes_host_entries_script,
        notify => Service['dnsmasq'],
    }

    augeas { "${cluster_hostname}_hosts":
        context => '/files/etc/hosts',
        incl => '/etc/hosts',
        lens => 'Hosts.lns',
        changes => $host_entries_on_login_node,
    }
```

Tell nodes inside the cluster to use this node as a DNS server.

Proxy HTTP and HTTPS for the internal network.

```
    class { 'hpc_cluster::login_node::proxy':                        §11.42.1
        internal_ipv4_subnet => $internal_ipv4_subnet,
    }
```

Configure the internal network interfaces.

```
    $augeas_ifcfg = ’/files/etc/sysconfig/network-scripts/ifcfg’
    augeas { "${hostname} ${cluster_hostname} internal":
        context => "${augeas_ifcfg}-${internal_interface}",
        changes => [
            ’set ONBOOT yes’,
            ’set BOOTPROTO static’,
            "set IPADDR ${internal_ipv4_address}",
            ’set NETMASK 255.255.0.0’,
        ],
    }
    if $use_infiniband == ’true’ {
```

To drive the Infiniband card:

```
        package { [’rdma’, ’ibutils’, ’libibverbs’]:
            ensure => present,
        }
        ->
        service { ’rdma’:
            enable => true,
            ensure => running,
        }
        ->
```

Set the InfiniBand network address. (This doesn’t bring up the interface.)

```
        augeas { "${hostname} ${cluster_hostname} infiniband internal":
            context => "${augeas_ifcfg}-${infiniband_interface}",
            changes => [
                ’set ONBOOT yes’,
                ’set BOOTPROTO static’,
                "set IPADDR ${internal_infiniband_ipv4_address}",
                ’set NETMASK 255.255.0.0’,
                ’set NM_CONTROLLED no’,
                ],
        }
    }
```

Prepare the /srv/passwd directory for the below.

```
    file { ’/srv/passwd’:
        ensure => directory,
        owner => root, group => 0, mode => 0644,
    }
```

Pass user and group information to the inside of the cluster.

```
    file { ’/etc/cron.hourly/hpc_cluster_passwd_group’:
        owner => root, group => 0, mode => 0755,
        source => "puppet:///modules/hpc_cluster/gather.cron",
    }
```

Export that information to the nodes inside the cluster.

```
augeas { 'export_passwd_to_cluster':
    context => '/files/etc/exports',
    changes => [
        'rm dir[.="/srv/passwd"]',
        'set dir[999] "/srv/passwd"',
        "set dir[.='/srv/passwd']/client \
                ${internal_ipv4_subnet}",
        'set dir[.="/srv/passwd"]/client/option ro',
    ],
}
include nfs                                                   §11.69
class { 'nfs::allow':                                         §11.69
    from => $internal_ipv4_with_netmask,
}
```

Tell nodes inside the cluster to grab this user and group information.

```
@@automount::mount { 'passwd':
    from => "${cluster_hostname}:/srv/passwd",
    tag => "${cluster_hostname}_passwd",
}
```

Listen inside the cluster for SMTP mail to relay to the outside.

```
include hpc_cluster::login_node::smtp                         §11.42.1
```

Tell nodes inside the cluster to use the login node as proxy.

```
@@proxy::yum { "${cluster_hostname}":
    host => 'login',
    port => 3128,
}
```

Tell nodes inside the cluster to use the login node as DNS server.

```
@@augeas { "${cluster_hostname} dns":
    context => '/files/etc/resolv.conf',
    changes => [
        'rm *',
        "set nameserver ${login_internal_ipv4}",
        "set search/domain[1] ${cluster_fqdn}",
        "set search/domain[2] ${::domain}",
    ],
}
```

Tell nodes inside the cluster to use the login node as gateway.

```
@@augeas { "${cluster_hostname} gateway":
    context => "${augeas_ifcfg}-eth0",
    changes => "set GATEWAY ${login_internal_ipv4}",
}
```

Install the Scyld OpenMPI packages. (Not automated yet.)

We used to make sure the Scyld modulefiles were on the MODULEPATH with an extra `profile.d` script. But now the `shell::env_modules` class (§11.94.2) takes a parameter we can set to include `/opt/scyld/modulefiles`.

```
    file { ’/etc/profile.d/before_modules_2.sh’:
        ensure => absent,
    }

}
class hpc_cluster::login_node::proxy(
    $internal_ipv4_subnet) {
```

Make HTTP and HTTPS available on the internal network.

```
    package { ’squid’:
        ensure => installed,
    }
    augeas { ’squid for cluster login’:
        context => ’/files/etc/squid/squid.conf’,
        changes => [
            ’rm acl[localnet][position() > 1]’,
            ’set acl[localnet][1]/localnet/type src’,
            "set acl[localnet][1]/localnet/setting \
             ’${internal_ipv4_subnet}’",
        ],
        require => Package[’squid’],
        notify => Service[’squid’],
    }
    augeas { ’let cluster nodes use Puppet port’:
        context => ’/files/etc/squid/squid.conf’,
        changes => [
            ’defnode puppet_port acl[999] ""’,
            ’set $puppet_port/SSL_ports/type port’,
            ’set $puppet_port/SSL_ports/setting 8140’,
            ],
        onlyif => "match acl[SSL_ports/type=’port’ and \
                             SSL_ports/setting=’8140’] \
                   size == 0",
    }
    service { ’squid’:
        enable => true,
        ensure => running,
    }
}
```

Use this class when the proxy that the login node offers to the HPC cluster internal network should in turn use a proxy to access the Net.

```
class hpc_cluster::login_node::proxy::upstream(
    $host,
    $port,
    $dontproxy_domain)
{
    include hpc_cluster::login_node::proxy
```

§11.42.1

```
    augeas { 'squid upstream proxy for cluster login':
        context => '/files/etc/squid/squid.conf',
        changes => [
            'rm acl[dontproxy_dns][position() > 1]',
            'set acl[dontproxy_dns]/dontproxy_dns/type dstdomain',
            "set acl[dontproxy_dns]/dontproxy_dns/setting \
             ${dontproxy_domain}",
            'rm acl[dontproxy_ip][position() > 1]',
            'set acl[dontproxy_ip]/dontproxy_ip/type dst',
            "set acl[dontproxy_ip]/dontproxy_ip/setting \
             ${hpc_cluster::login_node::proxy::internal_ipv4_subnet}",
            "set cache_peer \
             '${host} parent ${port} 0 no-query default'",
            "set cache_peer_access \
             '${host} deny dontproxy_dns dontproxy_ip'",
            'rm acl[localnet][position() > 1]',
            'set acl[localnet][1]/localnet/type src',
            "set acl[localnet][1]/localnet/setting \
             '${internal_ipv4_subnet}'",
        ],
        require => Package['squid'],
        notify => Service['squid'],
    }
}

class hpc_cluster::login_node::smtp {
    augeas { 'serve smtp to cluster network':
        context => '/files/etc/postfix/main.cf',
        # note: the $ reference is meant for postfix to read, not puppet
        changes => 'set inet_interfaces "$myhostname, localhost"',
    }
}
```

## 11.42.2 Management nodes

These are the nodes that head up the cluster: running the cluster management and queueing system software.

```
class hpc_cluster::management_node($cluster_hostname) {
  class { 'hpc_cluster::node':                                           §11.42.3
      cluster_hostname => $cluster_hostname,
  }
  Automount::Mount <<| tag == "${cluster_hostname}_passwd" |>>
  # Get user and group information from the login node and write it in
  # my passwd and group files.
  file { '/etc/cron.hourly/hpc_cluster_passwd_group':
      owner => root, group => 0, mode => 0755,
      source => "puppet:///hpc_cluster/integrate.cron",
  }
```

At present there is no puppet on management nodes. Besides the preceding, to get a management node up you must do the following:

1. Add the ClusterWare yum repo. (The exact URL depends on the cluster ID.)
2. Install ClusterWare: `yum groupinstall Scyld-ClusterWare`.
3. Configure (`/etc/beowulf/config`).
4. Obtain the DirectFLOW RPM from Panasas that corresponds to the ClusterWare kernel you're running.
5. Verify internal filer connectivity; set up NFS and Panasas mounts, on management and compute nodes.
6. Choose a place where the SGE_ROOT will go.
7. Build and install GridEngine.
8. Write modulefiles for GridEngine for the login and management nodes.
9. chkconfig GridEngine on.
10. Make sure the management node has /etc/modulefiles on the MODULEPATH.
11. Make sure the management node's internal IP reverse-looks-up to headX.CLUSTER.FQDN.
12. Install Scyld OpenMPI packages on login nodes.
13. Configure HA.
14. Prestage `/etc/profile.d/before_modules.sh` in the /etc/beowulf/config so the MODULEPATH will be right on the compute nodes.
15. Install valgrind.
16. Export /usr/bin, /usr/sbin, /usr/share from the management node to the cluster network.
17. Configure the compute nodes to mount these filesystems.
18. Adapt the Scyld /etc/beowulf/init.d/sshd script to merely configure sshd, not run it.
19. Configure GridEngine to use ssh for its rsh/rlogin, so that interactive jobs can be run with X forwarding.

```
}
```

### 11.42.3 All internal nodes

Any node inside the cluster needs these resources. With cluster management software, perhaps only the management nodes will run Puppet, and will cause the compute nodes to fall in line by other means than Puppet.

```
class hpc_cluster::node($cluster_hostname) {
    Proxy::Yum <<| name == "${cluster_hostname}" |>>
    Augeas <<| name == "${cluster_hostname} dns" |>>
    Augeas <<| name == "${cluster_hostname} gateway" |>>
    Smtp::Use_smarthost <<| tag == $cluster_hostname |>>
    include ::ntp                                                  §??
    Augeas <<| name == "${cluster_hostname} ntp.conf" |>>

    package { [
            'lynx',
            'man',
            'vim-enhanced',
            'wget',
            'bind-utils',
            'ipmitool',
```

panfs install uses bc.

```
            'bc',
```

Infiniband.

```
            'opensm',
            'ibutils',
            'rdma',
            'libibverbs-utils',
            'infiniband-diags',
        ]:
            ensure => installed,
    }

    service {
        'rdma':
            enable => true,
            ensure => running;
        'opensm':
            enable => true,
            ensure => running;
    }
```

This is so when people `module add openmpi`, they will get the PGI version by default, from among the `openmpis` that Scyld has built.

```
        file { '/opt/scyld/modulefiles/openmpi/.modulerc':
            ensure => present,
            owner => root, group => 0, mode => 0644,
            content => "#%Module
module-version pgi default
",
        }

}
```

### 11.42.4 Solitary login node

This is just like `login_node` but is used in the case where the login node is not redundant.

```
class hpc_cluster::solitary_login_node(
        $internal_ipv4_first_two_octets,
        $internal_infiniband_ipv4_first_two_octets,
        $external_interface = 'eth0',
        $internal_interface = 'eth1',
        $compute_node_count,
        $use_infiniband='false',
        ) {

    $iifto = $internal_ipv4_first_two_octets
    $login_internal_ipv4 = "${iifto}.1.1"
    $login1_internal_ipv4 = "${iifto}.1.2"
    $iibifto = $internal_infiniband_ipv4_first_two_octets
    $login1_internal_infiniband_ipv4 = "${iibifto}.1.2"

    class { 'hpc_cluster::login_node':
        internal_ipv4_first_two_octets =>
                $internal_ipv4_first_two_octets,
        internal_infiniband_ipv4_first_two_octets =>
                $internal_infiniband_ipv4_first_two_octets,
        internal_ipv4_address =>
                $login1_internal_ipv4,
        internal_infiniband_ipv4_address =>
                $login1_internal_infiniband_ipv4,
        compute_node_count =>
                $compute_node_count,
        use_infiniband => $use_infiniband,
        internal_interface => $internal_interface,
        external_interface => $external_interface,
    }
```

§11.42.1

Configure the alias on the internal network interface. Redundant login nodes will have heartbeat configuration to pass this IP address between themselves on failure, but solitary login nodes will just always hold the alias.

```
    $augeas_ifcfg = ’/files/etc/sysconfig/network-scripts/ifcfg’
    augeas { "${hostname} ${cluster_hostname} internal solitary":
        context => "${augeas_ifcfg}-${internal_interface}",
        changes => [
            "set IPADDR2 ${login_internal_ipv4}",
            ’set NETMASK2 255.255.0.0’,
        ],
    }
}
define automount::mount($from, $under=’’, $ensure=’present’, $options=[]) {}
class nfs {}

class nfs::allow($from) {}
define proxy::yum($host, $port) {}
```

## 11.43 High-Performance Computing Modernization Program

Configuration necessary to connect to HPCMP clusters.

### 11.43.1 Kerberos

Configuration necessary to get an HPCMP Kerberos ticket.

```
class hpcmp::kerberos {
    include "hpcmp::kerberos::${::osfamily}"
}
class hpcmp::kerberos::darwin {
    notify { ’hpcmp::kerberos unimplemented on Mac OS’:
        loglevel => err,
    }
}
class hpcmp::kerberos::redhat {
    package { ’hpc_krb5’:
        ensure => present,
    }
```

If we’re using some other form of Kerberos, the `/etc/krb5.conf` file may be automatically, repeatedly overwritten with settings not useful to us in getting HPCMP Kerberos tickets. So we want to explicitly use an HPCMP-specific configuration when doing HPCMP Kerberos.

```
    file { ’/etc/profile.d/hpc_krb5.sh’:
        owner => bin, group => 0, mode => 0444,
        content => "\
hpc_krb5=/usr/local/hpc_krb5
export PATH=\$hpc_krb5/bin:\$PATH
alias pkinit=\"KRB5_CONFIG=\$hpc_krb5/etc/krb5.conf \\\n\
                pkinit \"\n\
",
    }
```

We need DoD root and CA certificates. These are off in the pki module so that we can have only one copy of the certificates.

```
        include pki::ca_certs::pkinit                                    §11.76.1
    }
```

### 11.43.2 OpenSSH

Configuration necessary to connect to an HPCMP-administered cluster.

The parameter `hpc_cluster_host_patterns` is one or a list of host patterns as defined in `ssh_config(1)`, to which client-side SSH settings will apply. The host patterns should match any HPCMP cluster login node, but should not match local hosts.

```
    class hpcmp::openssh($hpc_cluster_host_patterns) {
        include hpcmp::kerberos                                          §11.43.1
        include "hpcmp::openssh::${::osfamily}"
```

This define implements for a set of hosts some of the settings Vern Staats set out on 1 May 2012. In the original configuration they are applied to all hosts. But we may need different settings, and so these settings should only apply when connecting to an HPCMP cluster.

Some of the original configurations Vern specified are now part of the `ssh::fips` class, §11.100.4, and so are not written here.

```
    define vrs_settings() {
        require augeas
        augeas { "hpcmp_ssh_config_add_${name}":
            context => "/files${ssh::client_config}",
            onlyif =>
"match Host[.=’${name}’] size == 0",
            changes => [
                "set Host[999] ’${name}’",
            ],
        }

        augeas { "hpcmp_ssh_config_config_${name}":
            require => [
                Augeas["hpcmp_ssh_config_add_${name}"],
                Package[’hpc_ossh’],
                ],
            context =>
"/files${ssh::client_config}/Host[.=’${name}’]",
            changes => [
                ’set GSSAPIAuthentication yes’,
                ’set GSSAPIDelegateCredentials yes’,
                ’set GSSAPIKeyExchange yes’,
                ’set GSSAPIRenewalForcesRekey yes’,
                "set PreferredAuthentications \
gssapi-with-mic,external-keyx,publickey,\
hostbased,keyboard-interactive,password",
                ’set ForwardX11 yes’,
                ’set ForwardX11Trusted no’,
```

The Unix SRG prevents us from using SSH forwarding everywhere (see §11.100.9), but for HPCMP clusters we need it, and apparently the HPCMP has accepted the risk, because their distribution of OpenSSH comes with it enabled. So un-disable it when talking to HPCMP clusters.

```
                ’set ClearAllForwardings no’,
```

Get rid of some settings, which when implemented here cause ssh to groan and fail.

```
                ’rm NoneEnabled’,
                ’rm MaxSessions’,
                ’rm XAuthLocation’,
                ’rm TcpRcvBuf’,
                ’rm TcpRcvBufPoll’,
                ’rm UMask’,
            ],
        }
    }

    vrs_settings { $hpc_cluster_host_patterns: }
}
```

```
   class hpcmp::openssh::darwin {
       notify { 'hpcmp::openssh::darwin unimplemented':
           loglevel => err,
       }
   }
   class hpcmp::openssh::redhat {
       package { 'hpc_ossh':
           ensure => present,
       }
   }
```

# 11.44 iCloud

```
class icloud::no_prompt {
       define cusa_set($value) {
           mcx::set { "com.apple.SetupAssistant/${name}":                §11.61.2
               value => $value,
           }
       }
```

Disable the prompt for Apple ID and iCloud for all users (the requirement only has to do with new users). auto: OSX8-00-01125

```
       cusa_set { 'DidSeeCloudSetup': value => true }
       cusa_set { 'LastSeenCloudProductVersion':
           value => $::macosx_productversion,
       }
   }
   class icloud::stig {
       include icloud::no_prompt                                          §11.44
   }
```

# 11.45 IEEE 1394 (Firewire)

## 11.45.1 Disabling IEEE 1394 (Firewire)

The implementations of this class tend to be rather destructive and not easily undoable.

```
   class ieee1394::no {
       include "ieee1394::no::${::osfamily}"
   }
```

### Under the Mac OS

```
class ieee1394::no::darwin {
    $exts = '/System/Library/Extensions'
```

Remove the Firewire driver on Macs. auto: OSX8-00-00845

```
       file { "${exts}/IOFireWireSerialBusProtocolTransport.kext":
           ensure => absent,
           force => true,
       }
   }
```

### Under Red Hat

Disable Firewire "unless needed." We do not need it.

auto: ECSC-1
auto: GEN008500

```
class ieee1394::no::redhat {
    kernel_module {
        "firewire-core": ensure => absent;
        "firewire-ohci": ensure => absent;
        "firewire-sbp2": ensure => absent;
        "firewire-net": ensure => absent;
    }
    file {
        "/lib/modules/$kernelrelease/kernel/drivers/firewire":
            ensure => absent, recurse => true,
            recurselimit => 1, force => true;
    }
}
```

To reinstate IEEE 1394 support on a host which has previously had it disabled in the above manner, you must reinstall the kernel package and restart the host.

## 11.46 Infrared

Configure support for infrared control.

### 11.46.1 Disable infrared support

Disable infrared support "to prevent unauthorized users from controlling a computer through the infrared receiver."

auto: ECSC-1
auto: OSX00090 M6
auto: OSX8-00-00075

```
class infrared::no {
    case $::osfamily {
        'darwin': { include infrared::no::darwin }
        default:  { unimplemented() }
    }
}
```

### Disable infrared under Mac OS X

```
class infrared::no::darwin {
    $exts = '/System/Library/Extensions'
    file {
        "${exts}/AppleIRController.kext":
            ensure => absent,
            force => true;
    }
}
```

## 11.47 ip6tables

`ip6tables` is the IPv6 packet filter under Linux.

Employ a local firewall for IPv6, using `ip6tables`.

auto: ECSC-1
auto: GEN008520

`ip6tables` rules are constructed in this policy from templates. This lets us group related rules, and include them as a whole; it makes explicit the order of the rules, which is quite important; and it lets us have both sets of rules general to a whole class of host (*e.g.* workstations) and sets of rules specific to a single host (*e.g.* `sumo`).

```
class ip6tables {
    package { 'iptables-ipv6':
        ensure => present,
    }
    service { 'ip6tables':
        ensure => running,
        hasstatus => true,
    }
}
```

The actual firewall rules that implement the following requirements are in the templates for this module, not here; but here is the place where they can be indexed, summarized and prose written about them, so here they are documented.

Configure the local firewall to reject all source-routed IPv6 packets, even those generated locally.

auto: ECSC-1
auto: GEN003605
auto: GEN003606

Source routing in IPv6 is done with Routing Header 0 (RH0); we merely need to drop every packet that has that optional header.

Configure the local firewall to reject all IPv6 packets by default, allowing only by exception.

auto: ECSC-1
auto: GEN008540

Configure the local firewall to reject ICMPv6 timestamp requests, including those sent to a broadcast address. To apply a set of ip6tables rules to a given host (node), first know the network and broadcast addresses of the node, and its default gateway. In this example we'll say the site is allocated a /48 prefix, and the host has IPv6 address 2001:DB8:0:3::16. The subnet's address is 2001:DB8:0:3::/64, and the whole site's address is 2001:DB8:0::/48. (See RFC 3849.) Then you would write:

auto: ECSC-1
auto: GEN003602
auto: GEN003604

```
ip6tables::use { "mytemplate":
    subnet => "2001:DB8:0:3::/64",
    site   => "2001:DB8:0::/48",
}
```

where `mytemplate` is the name of a file in `modules/ip6tables/templates` in this policy. `site` is used for rules which deal with traffic within a site's (possibly multiple) networks, such as SSH connections or pings.

```
define ip6tables::use($subnet, $site) {
    include ip6tables
    $ipt_text = template("ip6tables/${name}")
```

§11.47

```
    file { "/etc/sysconfig/ip6tables":
        owner => root, group => 0, mode => 0600,
        content => $ipt_text,
        notify => Service["ip6tables"],
    }
}
```

## 11.48 iptables

Employ a local firewall, using `iptables`.

auto: ECSC-1
auto: GEN008520

`iptables` rules are constructed in this policy from templates. This lets us group related rules, and include them as a whole; it makes explicit the order of the rules, which is quite important; and it lets us have both sets of rules general to a whole class of host (*e.g.* workstations) and sets of rules specific to a single host (*e.g.* `sumo`).

```
class iptables {
    service { "iptables":
        ensure => running,
        hasstatus => true,
    }
```

The requirement is to drop source-routed IPv4 packets. At SEARDE production go-time, the `xtables-addons` package, which supplies the iptables match code for IPv4 options, including source routing, wasn't working with the rest of iptables. That means source-routed packets are not being specifically dropped at the host firewall. See §11.66.12 for another way that most of the source-routed traffic is being rejected.

GEN003600
GEN003605
GEN003606

Our previous means of compliance here has been deleted; see previous versions of this file in Subversion.

```
}
```

Configure the local firewall to reject all packets by default, allowing only by exception.

auto: ECSC-1
auto: GEN008540

Configure the local firewall to reject ICMP timestamp requests, including those sent to a broadcast address. To apply a set of iptables rules to a given host (node), first know the network and broadcast addresses of the node, and its default gateway. In this example we'll say the host has IPv4 address 192.0.2.45. The network address is 192.0.2.0/25; the corresponding broadcast address is 192.0.2.127 (the address derived by turning on all the bits masked out by the netmask). The gateway in our example is 192.0.2.1. (See RFC 5737.) Then you would write:

auto: ECSC-1
auto: GEN003602
auto: GEN003604

```
iptables::use { "amodule/mytemplate":
    site_subnets => ["192.0.2.0/25"],
    broadcast => "192.0.2.127",
    gateway => "192.0.2.1",
}
```

where mytemplate is the name of a file in amodule/templates, and amodule is somewhere on Puppet's module path (e.g., in modules-unclass or modules-fouo). site_subnets are used for rules which deal with traffic within a site's (possibly multiple) networks, such as SSH connections or pings.

```
define iptables::use($site_subnets, $broadcast, $gateway) {
    include iptables
    file { "/etc/sysconfig/iptables":
        owner => root, group => 0, mode => 0600,
        content => template("${name}"),
        notify => Service["iptables"],
```

§11.48

This previusly required xtables-addons; see Subversion revision 6550.

```
    }
}
```

# 11.49 iTunes

Configure iTunes.

## 11.49.1 STIG-required configuration

Configure iTunes in accordance with the Mac OS X STIG.

```
class itunes::stig {
```

Disable iTunes Store and other network features of iTunes on Macs.

auto: ECSC-1
auto: OSX00530 M6
auto: OSX8-00-01140
auto: OSX8-00-01150
auto: OSX8-00-01155

Note that because this policy uses an MCX object, it imposes this setting on every user at once, obviating any actions that must be "performed for each user."

§11.61.2

```
    mcx::set { [
            'com.apple.iTunes/disableMusicStore',
            'com.apple.iTunes/disablePing',
            'com.apple.iTunes/disablePodcasts',
            'com.apple.iTunes/disableRadio',
            'com.apple.iTunes/disableSharedMusic',
            ]:
        value => true,
    }
}
```

# 11.50 Java Runtime Environment

## 11.50.1 STIG-required JRE configuration

The Java Runtime Environment (JRE) STIG [?, jre-stig]as some DoD-level requirements regarding how the JRE must deal with cryptographically signed code. Here we enforce those requirements.

```
define jre::stig(
        $jre='/usr/lib/jvm/jre-1.6.0') {
```

Make sure the deployment properties file exists.

auto: JRE0080-UX

```
    $dp = "${jre}/lib/deployment.properties"

    file { $dp:
        ensure => present,
        owner => root, group => 0, mode => 0644,
    }
```

Enforce policy regarding the contents of the deployment properties file.

```
    $notinca = "deployment.security.askgrantdialog.notinca"
    $crl     = "deployment.security.validation.crl"
    $ocsp    = "deployment.security.validation.ocsp"

    augeas { "jre_stig_${jre}_deployment_properties":
        lens => 'Properties.lns',
        incl => $dp,
        changes => [
```

"Disable ability to grant permission to untrusted authority." auto: JRE0001-UX

```
            "set ${notinca} false",
```

"Lock out option to grant permission to untrusted." auto: JRE0010-UX

```
            "set ${notinca}.locked true",
```

"Enable revocation check on publisher certificates." auto: JRE0020-UX

```
            "set ${crl} true",
```

"Lock the option to check certificates for revocation." auto: JRE0030-UX

```
            "set ${crl}.locked true",
```

"Enable online certificate validation." auto: JRE0040-UX

```
            "set ${ocsp} true",
```

"Lock online certificate validation." auto: JRE0050-UX

```
            "set ${ocsp}.locked true",
        ],
    }
```

Make sure the deployment configuration file exists. auto: JRE0070-UX

```
    $dc = "${jre}/lib/deployment.config"

    file { $dc:
        ensure => present,
        owner => root, group => 0, mode => 0644,
    }
```

Enforce policy regarding the contents of the deployment configuration file.

Configure the deployment configuration file to point at the deployment properties file. auto: JRE0060-UX

```
        $dsconfig = "deployment.system.config"

        augeas { "jre_stig_${jre}_deployment_config":
            lens => 'Properties.lns',
            incl => $dc,
            changes => "set ${dsconfig} \"file:${dp}\"",
        }
    }
```

## 11.51 Kernel core dumping

### 11.51.1 Disable kernel dumping

Disable kernel core dumping to improve the security of the system during aborts: Kernel core dump files will contain sensitive data, and heretofore we have not needed to debug crashed kernels.

auto: ECSC-1
auto: GEN003510 M6
auto: OSX8-00-01105
auto: ECSC-1
auto: GEN003510
auto: DCSS-1
N/A: GEN003520
N/A: GEN003521
N/A: GEN003522
N/A: GEN003523

```
    class kernel_core::no {
        case $::osfamily {
            'redhat': {
                service { 'kdump':
                    enable => false,
                    ensure => stopped,
                }
            }
            'darwin': {
                augeas { 'sysctl_kern_coredump_off':
                    context => '/files/etc/sysctl.conf',
                    changes => 'set kern.coredump 0',
                }
            }
            default:  { unimplemented() }
        }
    }
```

## 11.52 KVM (Kernel Virtual Machine)

### 11.52.1 Random number generator

When in FIPS-compliant mode, OpenSSL uses /dev/random for its randomness needs. This can be much slower without any decent sources of randomness, such as network packets, console keystrokes, etc., which a virtual machine may lack. The virtio-rng module uses randomness from the host system in the virtual machine, improving the performance of /dev/random.

```
    class kvm::guest_random {
        if $virtual == "kvm" {
```

See [15], §22.6, “Persistent Module Loading.”

```
        file { "/etc/sysconfig/modules/virtio-rng.modules":
            owner => root, group => 0, mode => 0755,
            content => "#!/bin/sh\nmodprobe virtio-rng\n",
        }
    }
}
```

# 11.53 LDAP

We do not presently use the Lightweight Directory Access Protocol (LDAP) for authentication, but if we did, we would have to implement these requirements:

Systems using LDAP for authentication or account information must use FIPS-approved means for constructing a TLS connection, use DoD-signed certificates to authenticate themselves and the server, and check for trust and revocation of the server certificate. Use this PKI-based method or Kerberos, not storage of a password, to authenticate LDAP client hosts.

N/A: GEN007970
N/A: GEN007980
N/A: GEN008000
N/A: GEN008020
N/A: GEN008040
N/A: GEN008050

Macs using LDAP must be “securely configured” in a variety of ways.

N/A: OSX00115 M6
N/A: OSX00120 M6
N/A: OSX00125 M6
N/A: OSX00121 M6
N/A: OSX00122 M6
N/A: OSX00123 M6
N/A: OSX00124 M6

## 11.53.1 STIG-required LDAP configuration

```
class ldap::stig {
```

Control ownership and permissions of `ldap.conf`.

```
    $ldap_conf = $::osfamily ? {
        'redhat' => '/etc/ldap.conf',
        'darwin' => '/etc/openldap/ldap.conf',
        default  => unimplemented,
    }
    file { $ldap_conf:
        owner => root, group => 0, mode => 0644,
    }
```

auto: ECLP-1
auto: GEN008060 M6
auto: GEN008080 M6
auto: GEN008100 M6
auto: ECLP-1
auto: GEN008060
auto: GEN008080
auto: GEN008100

Remove extended ACLs on `ldap.conf`.

```
    no_ext_acl { $ldap_conf: }
```

auto: ECLP-1
auto: GEN008120 M6

This policy presently does not configure an LDAP client.

```
}
```

auto: ECLP-1
auto: GEN008120
N/A: GEN008140
N/A: GEN008160
N/A: GEN008180
N/A: GEN008200
N/A: GEN008220
N/A: GEN008240
N/A: GEN008260
N/A: GEN008280
N/A: GEN008300
N/A: GEN008320
N/A: GEN008340
N/A: GEN008360

# 11.54 libreport

When a crash happens, it appears this library is used to send news of it to someone, somewhere, somehow. For example, an email may be sent.

```
   class libreport {
       case $::osfamily {
           'RedHat': {
               augeas { 'libreport_set_from_address':
                   context => '/files/etc/libreport/plugins/mailx.conf',
                   changes => "set EmailFrom 'root@${::fqdn}'",
               }
           }
           default: {}
       }
   }
```

# 11.55 Location services

## 11.55.1 Disable location services

```
class location::no {
    include "location::no::${::osfamily}"
}
   class location::no::darwin {
       $version_underscores = regsubst(
           $::macosx_productversion_major,
           '\D', '_', 'G')
       $klassname = "${::osfamily}_${version_underscores}"
       include "location::no::${klassname}"
   }
   class location::no::darwin_10_6 {}
   class location::no::darwin_10_9 {
```

Disable Location Services on Macs. auto: OSX8-00-00535

```
       mcx::set { 'com.apple.MCX/DisableLocationServices':
           value => true,
       }
   }
   class location::no::redhat {}
```

§11.61.2

# 11.56 Logging

## 11.56.1 Log backup

rsyslog should log remotely in most cases, and logs can be backed up from the loghost. But limited use in practice indicates that rsyslog may fail to send log messages under some conditions, and its incomplete PKI support means remote logging may become infeasible in our case, given security requirements.

Remotely logged messages are saved in files on the loghost. Log messages are always written to local files, whether they are sent remotely or not. Audit messages are only written to local files: we have no remote audit logging capability at present.

Back up audit logs and other logs to archival media. Retain them for auto: ECRR-1 one year, or five years for systems containing sources and methods intelligence (SAMI).

Exactly how logs are backed up and to where depends on to which network a host is connected. `log::backup::*` classes make various implementations of log backup happen. This Configuration Management for IT Systems Example Policy may not cover the entire journey of log backups to archival media: consult the Backup Policy [**?**] in addition.

### Backing up logs using NFS

If you had a `/net/admin` directory mounted on each host, to which logs could be backed up, this class would do it.

It may not be required to back up logs daily.

```
class log::backup::to_net_admin {
    file { "/etc/cron.daily/backup_logs":
        owner => root, group => 0, mode => 0700,
        source => "file:///puppet/modules/log/backup/to_net_admin.sh",
    }
```

Tell the filer policy agent to make a directory for the logs to land in.

```
    @@log::backup::to_net_admin::for_host { "$::hostname": }
}
```

This is what the filer policy agent (see 11.31.1) must do to enable log backups to `/net/admin`.

```
class log::backup::to_net_admin::filer {
    file { "/net/admin/BACKUPS":
        ensure => directory,
        owner => root, group => skadmin, mode => 2770,
    }
```

Collect the directories each host has requested; implement those policies on the filer policy agent host.

```
    Log::Backup::To_net_admin::For_host <<| |>>
```

Clean out old logs. Keep logs for five years, just in case we have sources and methods intelligence (SAMI) on some host. Disks are cheap, noncompliance expensive.

```
    tidy { "/net/admin/BACKUPS":
        recurse => 2,
        matches => "system_logs-*.tar.gz",
        age => "5y",
    }
}
```

How the filer policy agent can make a directory for me to back up my logs in:

```
define log::backup::to_net_admin::for_host {
    file {
        "/net/admin/BACKUPS/${name}":
            ensure => directory,
            owner => root, group => skadmin, mode => 2755;
        "/net/admin/BACKUPS/${name}/LOGS":
            ensure => directory,
            owner => root, group => skadmin, mode => 2755;
    }
}
```

### 11.56.2 Logging via rsyslog

RHEL6 uses rsyslog as its default logging dæmon. rsyslog supports remote logging over TCP, and TLS encryption using GnuTLS. But it appears not to support CRLs, nor OCSP.[3] Also, it requires that the loghost's certificate and all client certificates be signed by the same CA certificate.[4]

A loghost set up using this scheme will require hosts which connect to have a valid certificate whose common name is a fully qualified DNS name ending in the same domain as the loghost. For example, if the loghost is named loghost.example.com, it will require connecting clients to have certs with common names matching the glob *.example.com.

```
class log::rsyslog {
    package { ["rsyslog", "rsyslog-gnutls"]:
        ensure => present,
    }
    service { "rsyslog":
        enable => true,
        ensure => running,
    }
```

Control ownership and permissions of the rsyslog configuration. Compliance and configuration are mixed here.

auto: ECLP-1
auto: GEN005390
auto: GEN005400
auto: GEN005420

[3] According to the rsyslog Git repository as of 2011 Jun 09, runtime/nsd_gtls.c, line 628, has a comment indicating that as of May 2008 the author, Rainer Gerhards, "doubt[s] we'll ever [use CRLs]. This functionality is considered legacy." The term OCSP is not found in the code.

[4] /usr/share/doc/rsyslog-4.6.2/ns_gtls.html in the rsyslog package: "Even in x509/fingerprint mode, both the client and sever [sic] certificate currently must be signed by the same root CA. This is an artifact of the underlying GnuTLS library and the way we use it. It is expected that we can resolve this issue in the future."

http://www.rsyslog.com/doc/ns_gtls.html says the same thing as of 2011 Jun 09.

As of Jan 2013, we have rsyslog 5.8.10, and it's the same in this respect.

```
    file {
        "/etc/rsyslog.d":
            ensure => directory,
            owner => root, group => 0, mode => 0640,
            recurse => true;
        "/etc/rsyslog.conf":
            owner => root, group => 0, mode => 0640,
            content => "\$IncludeConfig /etc/rsyslog.d/*.conf\n",
            require => File['/etc/rsyslog.d'],
            notify => Service['rsyslog'];
    }
```

Remove extended ACLs on the `rsyslog` configuration. auto: ECLP-1

```
    no_ext_acl { "/etc/rsyslog.conf": }
    no_ext_acl { "/etc/rsyslog.d": recurse => true }
```

auto: GEN005395

```
    define common_conf() {
        file { "/etc/rsyslog.d/${name}":
            owner => root, group => 0, mode => 0640,
            content => template("log/rsyslog/${name}"),
            notify => Service['rsyslog'],
        }
    }
    common_conf {
        "00common-global.conf":;
        "10gnutls-global.conf":;
        "50local.conf":;
    }
}
```

### 11.56.3 Configuring remote logging clients

(This excludes configuration of exactly which log server to use; see §11.56.5.)

```
class log::rsyslog::client($networkname) {
    include log::rsyslog
```

§11.56.2

Install the SELinux rules that let rsyslogd talk to the loghost.

```
    $selmoduledir = "/usr/share/selinux/targeted"
    file { "${selmoduledir}/rsyslog_client.pp":
        owner => root, group => 0, mode => 0644,
        source => "puppet:///modules/log/rsyslog/\
rsyslog_client.selinux.pp",
    }
    selmodule { "rsyslog_client":
       # autorequires above file
       ensure => present,
       syncversion => true,
       notify => Service['rsyslog'],
    }
```

Collect the to_loghost resource exported by the loghost.

```
    Log::Rsyslog::To_loghost <<|
        networkname == $networkname
    |>>
```

The client needs a certificate that the server will recognize in order to connect.

The client needs the CA certificate(s) installed so it can authenticate the server.

Configuration of the rsyslogd (`/etc/rsyslog.conf`) is set in §11.56.5 because it depends on the loghost's address.

```
}
```

### 11.56.4 Configuring a loghost

The "site-defined procedure" for setting up and documenting a loghost is this: admins do GEN005460

1. Write `include log::loghost` in the node declaration in §11.2.
2. Immediately before this, write a comment containing the tag `\documented{unixsrg}{GEN005460}` and the justification for that host to be a loghost.

RHEL5 does not receive syslog messages by default (see `/etc/sysconfig/syslog`). RHEL5: GEN005480 RHEL6 does not receive syslog messages by default (see `/etc/rsyslog.conf`). RHEL6: GEN005480 To prevent inadvertent disclosure of sensitive information, do not configure any host to listen for log messages over the network by any other means than the above procedure. admins do GEN005480

Now, this is how a loghost so documented is configured:

```
class log::rsyslog::loghost($networkname) {
    include log::rsyslog
```

§11.56.2

Install the SELinux rules that let rsyslogd listen to clients.

```
    $selmoduledir = "/usr/share/selinux/targeted"
    file { "${selmoduledir}/rsyslog_loghost.pp":
        owner => root, group => 0, mode => 0644,
        source => "puppet:///modules/log/rsyslog/\
rsyslog_loghost.selinux.pp",
    }
    selmodule { "rsyslog_loghost":
       ensure => present,
       syncversion => true,
       notify => Service['rsyslog'],
    }
```

The loghost needs a certificate, which will also be distributed to each log client.

The loghost needs a copy of the CA certificate(s) which have signed the certificates of the log clients.

The locations of these files are written in the `rsyslog.conf` file.

```
    file { '/etc/rsyslog.d/20loghost.conf':
        owner => root, group => 0, mode => 0640,
        content => template(
            'log/rsyslog/loghost-only/20loghost.conf'),
        notify => Service['rsyslog'],
    }
```

Export the to_loghost resource so that clients can pick it up.

```
    @@log::rsyslog::to_loghost { "$::fqdn":
        networkname => $networkname,
        ipaddress => $::ipaddress,
    }
}
```

### 11.56.5 Sending log messages to a loghost

"[U]se a remote syslog server (loghost)," so that the remotely collected system auto: ECAT-1
log data "can be used as an authoritative log source in the event a system is auto: GEN005450
compromised and its local logs are suspect," and so that it's easier to check logs every day and set up automated alerts.

Call this define with the name of the loghost. It must match the common name in the loghost's certificate.

The way this happens is that the loghost exports one of these (the Puppet term here is "exported resources"), and the clients collect it. So the name parameter is given by the loghost, but the contents of the define happen on the clients.

(See §11.2 and §11.1 for places where this defined resource type is used.)

```
define log::rsyslog::to_loghost($networkname, $ipaddress) {
    $loghost = $name
    file { '/var/spool/rsyslog':
        ensure => directory,
        owner => root, group => 0, mode => 0700,
    }
    file { "/etc/rsyslog.d/80send-to-loghost.conf":
        owner => root, group => 0, mode => 0640,
        content => template(
            'log/rsyslog/client-only/80send-to-loghost.conf'),
        notify => Service['rsyslog'],
        require => File['/var/spool/rsyslog'],
    }
    augeas { "add loghost to /etc/hosts":
        context => "/files/etc/hosts",
        changes => [
            "set 999/ipaddr '$ipaddress'",
            "set 999/canonical '$loghost'",
            "set 999/alias[999] loghost",
        ],
        onlyif => "match *[canonical='$loghost'] size == 0",
    }
}
```

### 11.56.6 STIG-required logging configuration

```
class log::stig {
```

Control permissions on all system log files.

auto: ECTP-1

Make all system log files have mode `0640` or less permissive.

auto: GEN001260
auto: ECTP-1
auto: GEN001260 M6

This is a pair of execs and not a file resource type because the file resource type can't set a different mode for a directory versus its contents. (We need to be careful because some files under `/var/log` already have more restrictive permissions than `0640`, so to use a numeric mode would be painting with too wide a brush.)

GNU chmod, when called with `-v`, will “output a diagnostic for every file processed.” The `-c` switch will “report only when a change is made.” Mac (BSD?) chmod `-v`, on the other hand, says it will show filenames “as the mode is modified.” This latter chmod does not recognize the `-c` switch and will fail if it is given.

```
    $verbose_chmod = $::osfamily ? {
        'RedHat' => '/bin/chmod -c',
        'Darwin' => '/bin/chmod -v',
        default  => '/bin/chmod -v',
    }
```

Secure `cron` logs. Secure SMTP logs.

auto: ECLP-1
auto: ECTP-1
auto: GEN003180
auto: ECLP-1
auto: GEN004500

```
    exec { "var_log_contents_other_minus_read":
        command => "${verbose_chmod} -R o-rwx,g-w /var/log",
        logoutput => true,
    }
    exec { "var_log_self_read_ok":
        command => "${verbose_chmod} o+rx /var/log",
        logoutput => true,
        require => Exec["var_log_contents_other_minus_read"],
    }
```

Remove extended ACLs on system log files (including SMTP and `cron` logs).

```
    no_ext_acl { "/var/log": recurse => true }
```

auto: ECLP-1
auto: ECTP-1
auto: GEN001270
auto: GEN003190
auto: GEN004510
auto: ECLP-1
auto: ECTP-1
auto: GEN001270 M6
auto: OSX8-00-00825

Some SRG requirements regard the system logging configuration file. The name of the system logging configuration file depends on which system logger is in use. See the class for the relevant logger for the implementations of those requirements.

Impose platform-specific configurations on log files:

```
    include "log::stig::${::osfamily}"
}
```

### 11.56.7 Admin guidance regarding logging

Do not cause unencrypted log traffic to cross enclave boundaries.

admins do
GEN005440

### Log rules for Macs

```
class log::stig::darwin {
```

Make sure root:wheel owns the system log files listed in the syslog configuration. auto: OSX8-00-00815

```
    exec { ’chown mac logs’:
        command => ’grep ^/ /etc/newsyslog.conf | \
                    awk "{print \$1}" | \
                    xargs chown root:wheel’,
        unless => ’grep ^/ /etc/newsyslog.conf | \
                   awk "{print \$1}" | \
                   xargs stat -f "%Su:%Sg" 2>/dev/null | \
                   grep -v "^root:wheel\$" | \
                   awk "BEGIN{x=0;}{x=1;}END{exit x;}"’,
    }
```

Ensure restrictive permissions for system log files. auto: OSX8-00-00820

```
    exec { ’chmod mac logs’:
        command => ’grep ^/ /etc/newsyslog.conf | \
                    awk "{print $1}" | \
                    xargs chmod g-w,o-rwx’,
        unless => ’grep ^/ /etc/newsyslog.conf | \
                   awk "{print $1}" | \
                   xargs stat -f "%Sp" 2>/dev/null | \
                   grep -v "^.rw..-----\$" | \
                   awk "BEGIN{x=0;}{x=1;}END{exit x;}"’,
    }
```

(On a stock Mavericks system it looks like none of these files actually exist.)

Enable local logging on Macs. auto: OSX8-00-01025

```
    service { ’com.apple.newsyslog’:
        enable => true,
        ensure => running,
    }
```

The default setting for how many logs to keep is 5. This is adequate for this organization at this time. Mavericks: OSX8-00-01030

```
}

class log::stig::redhat {
}
```

## 11.56.8 Logging via syslogd

No provisions for remote logging are made here as they are with rsyslog.

```
class log::syslog {
```

Control ownership and permissions of the `syslog.conf` file. auto: ECLP-1 auto: GEN005400 M6 auto: GEN005420 M6

```
    file { ’/etc/syslog.conf’:
        owner => root, group => 0,
    }
```

Remove extended ACLs from the `syslog.conf` file. auto: ECLP-1 auto: GEN005395 M6

```
    no_ext_acl { ’/etc/syslog.conf’: }
}
```

### 11.56.9 Make logs viewable by the logview user

Concept of operations: A log viewing host has an automatic graphical login to the logview user. This host has no input devices, only monitors. On this host resides logview's private SSH key. Part of the session startup is to start an xterm with an ssh in it; the ssh connects to the loghost and runs a log-tailing command. To mitigate the risk of having a private key with no passphrase protecting it, we make sure that the key is only usable on the loghost to run the log-tailing command, not any arbitrary command. Rather than making the log files available to the unprivileged logview user for reading, we make logview sudo in order to read them.

Apparently, obtaining a pty and using a command-limited SSH key are two things that OpenSSH does not support at the same time. So we have to reconfigure sudo such that for this user it will allow sudoing without a tty. The `sudoers(5)` man page seems to imply that the `requiretty` option exists to make sure that people use sudo and not scripts, by compelling its use from a login session. The stock `/etc/sudoers` file says in its comments that the reason to require a tty is so that sudo can suppress the display of the password as it is typed. In this case we want to enable sudo to be used by a script (limited to one command, tailing the system log), and logview does not use a password to sudo, so a password cannot be accidentally shown. With the risks of not requiring a tty suitably mitigated, we proceed cheerfully.

```
class log::viewable($ssh_public_key) {
    Group <| title == "logview" |>
    User <| title == "logview" |>

    file { "/usr/local/sbin/tail-messages":
        owner => root, group => 0, mode => 0755,
        content => "#!/bin/sh\n\
sudo /usr/bin/tail -f /var/log/messages\n",
    }
```

```
       file { "/etc/sudoers.d/logview":
           owner => root, group => 0, mode => 0440,
           content => "Defaults:logview !requiretty\n\
   logview ALL=(ALL) \
           NOPASSWD:/usr/bin/tail -f /var/log/messages\n",
       }

       ssh_authorized_key {
           "logview":
               require => [
                   File["/home/logview"],
                   File["/usr/local/sbin/tail-messages"],
               ],
               user => "logview",
               type => "ssh-dss",
               name => "logview@bla",
               options => ['command="/usr/local/sbin/tail-messages"'],
               key => $ssh_public_key,
       }
   }
```

## 11.57 Login window

Configure the Mac login window.

### 11.57.1 STIG-required login window configuration

```
class loginwindow::stig {

    $lw_domain = "/Library/Preferences/com.apple.loginwindow"
```

Configure the Mac login window to show username and password prompts, not a “list of local user names available for logon.”

auto: ECSC-1
auto: OSX00310 M6

```
       mac_default { "$lw_domain:SHOWFULLNAME":
           type => int,
           value => 1,
       }
```

Disable password hints in the Mac login window.

auto: IAAC-1
auto: OSX00325 M6

```
       mac_default { "$lw_domain:RetriesUntilHint":
           type => int,
           value => 0,
       }
```

Disable automatic login on Macs.

auto: IAAC-1
auto: OSX00425 M6

```
       mac_default { "$lw_domain:autoLoginUser":
           ensure => absent,
       }
   }
```

## 11.58 Mac launchd service definitions

A defined resource type that creates launchd service files. launchctl uses these to start and stop services; the Puppet service resource type can talk to launchctl. Just as the Puppet service type cannot create `/etc/init.d` files under Linux, it also can't create `/Library/LaunchDaemons` files on a Mac.

Parameters: `name` is the canonical name of the service. It's written in backward DNS, for example `com.example.myservice`. This is the same name you'll need to tell the Puppet service resource type to start and stop the service once you've defined it using this type. `description` is a vernacular description of the service. `environment` is a hash with variable names as keys and values as values. `arguments` is an array of arguments with which to run the program, *starting with argument 0*, the program name.

```
define mac_launchd_file(
         $description,
         $environment,
         $arguments,
         $requires_network='true',
         ) {
    $ld = '/Library/LaunchDaemons'
    $plist = "${ld}/${name}.plist"
    file { $plist:
        ensure => present,
        owner => root, group => 0, mode => 0644,
    }
```

Make the arguments always be an array, because in the property list file they should always be an array. See `http://projects.puppetlabs.com/issues/15813`. We assume here that if no arguments are to be given to the program to start, it's harmless to provide one argument which is an empty string.

```
   $arglength = inline_template("<%=@arguments.length%>")
   if $arglength == 1 {
        $array_args = [$arguments[0], ""]
    } else {
        $array_args = $arguments
    }
```

From an old wiki page, `http://projects.puppetlabs.com/projects/1/wiki/Puppet_With_Launchd`, and `launchd.plist(5)`.

```
        mac_plist_value {
            "${plist}:Label":
                value => $name;
            "${plist}:ServiceDescription":
                value => $description;
            "${plist}:EnvironmentVariables":
                value => $environment;
            "${plist}:ProgramArguments":
                value => $array_args;
        }
        case $requires_network {
            'true': {
                mac_plist_value {
                    "${plist}:RunAtLoad":
                        value => false;
                    "${plist}:KeepAlive":
                        value => {
                            'NetworkState' => true,
                            };
                }
            }
            default: {
                mac_plist_value {
                    "${plist}:RunAtLoad":
                        value => true;
                    "${plist}:KeepAlive":
                        value => true;
                }
            }
        }
    }
```

## 11.59 Mac local groups

### 11.59.1 Remove sharepoint groups

There are some "sharepoint" groups on any given Mac, which have something to do with sharing folders over the network (not with Microsoft Sharepoint). We don't share folders from our Macs, only from our filers, so we don't need membership in these groups. But we do have many other groups. NFSv3 has a sixteen-group limit, and some of our users have nearly sixteen groups that it's important they be in. The sharepoint groups count against that maximum, and they contain the `everyone` group nested inside them, so here we remove that so to free up groups for our users.

```
    class mac_local_groups::remove_sharepoints {
```

```
    define remove_everyone_from() {
        $everyone_uuid = "ABCDEFAB-CDEF-ABCD-EFAB-CDEF0000000C"
        exec { "remove everyone from ${name}":
            onlyif => "/usr/bin/dscl . \
                          -read /Groups/${name} NestedGroups | \
                      /usr/bin/grep ${everyone_uuid} >&/dev/null",
            command => "/usr/bin/dscl . \
                          -delete /Groups/${name} NestedGroups \
                                  ${everyone_uuid}",
        }
    }
    remove_everyone_from { 'com.apple.sharepoint.group.2': }
    remove_everyone_from { 'com.apple.sharepoint.group.3': }
}
```

## 11.60 Mac packages

The apple and pkgdmg providers for the package resource type require that a source parameter be given. Mac packages will be stored on some NFS or HTTP location, but that location is specific to a given network, and modules-unclass is supposed to be generic.

This define exists to gather all of the references to such a location into one place.

```
define mac_package(
    $ensure='installed',
    $sourcedir='',
    ) {
```

We haven't got Hiera installed on our Puppet 2 master server.

```
    if $sourcedir == '' {
        if $::puppet_version !~ /^3\./ {
            $use_source = '/'
        } else {
            $use_source = hiera('mac_package::sourcedir', '/')
        }
    } else {
        $use_source = $sourcedir
    }
```

Attempt to autorequire the network mount that the sourcedir appears to be on.

```
    if $use_source =~ /^(\/net\/[^\/]+)/ {
            if defined(Mac_automount[$1]) {
                $requires = [Mac_automount[$1]]
            } else {
                $requires = []
            }
        } else {
            $requires = []
        }

        package { $name:
            ensure => $ensure,
            source => "${use_source}/${name}",
            require => $requires,
        }
    }
```

## 11.61 MCX

Manage per-user or per-computer settings on Macs using MCX (acronym expansion unknown).

Puppet provides an mcx resource type, which "manages the entire MCXSettings attribute available to some directory services nodes." According to a mailing list message from October 2009, this is because there are "many nested values that would be impossible to neatly specify in the puppet DSL." The best guide so far for how to manage MCX using the mcx resource type is at http://flybyproduct.carlcaum.com/2010/03/managing-mcx-with-puppet-on-snow.html.

With all that said, this module does not use the mcx resource type: here we try to manage in more detail, so that settings needed for one reason or another can be written in the place in this Configuration Management for IT Systems Example Policy where they logically belong, rather than being jumbled together into one big pot of settings.

### 11.61.1 Prepare computer object

Make an object for the computer so that we can set MCX settings on it. See http://projects.puppetlabs.com/issues/5079 for why we would not just use /Computers/localhost.

```
class mcx::prepare {
```

This exec resource is lifted from http://flybyproduct.carlcaum.com/2010/03/managing-mcx-with-puppet-on-snow.html. But we use the -F switch to grep so that it will treat the FQDN as a literal string to search for, not a regular expression. This may never matter but it is more correct.

```
    exec { "System in Local Directory":
        path => ['/bin', '/usr/bin'],
        command => "dscl localhost -create \
                    /Local/Default/Computers/$::fqdn \
                    ENetAddress $::macaddress_en0 \
                    RealName $::fqdn \
                    RecordName $::fqdn",
        unless => "dscl localhost -list \
                    /Local/Default/Computers | \
                    grep -F $::fqdn",
    }
}
```

### 11.61.2 Set MCX values on the computer

The name must be in the format *appDomain/key1*[/*key2*/*key3*...] .

This defined resource type always uses the record /Computers/$::fqdn as the place to set the key.

Example:

```
mcx::set { "com.apple.digihub/com.apple.digihub.cd.music.appeared":
    mcx_domain => 'always',
    value => 1,
}
```

* * *

```
define mcx::set($mcx_domain='always', $value, $ensure='present') {
    require mcx::prepare

    mac_mcx_plist_value { "/Computers/${::fqdn}:${name}":
        mcx_domain => $mcx_domain,
        value => $value,
        ensure => $ensure,
    }
}
```

## 11.62 Menu add-ons

Menu add-ons are the little icons that show in the right side of the Mac menu bar, and let you change the sound volume, the AirPort settings, search for things, switch users, etc.

### 11.62.1 Security menu

The biggest reason to enable this is that the menu it makes available has a “Lock Screen” item on it.

```
class menu_addons::security {
    $kcaccess = '/Applications/Utilities/Keychain Access.app'
    $filename = "${kcaccess}/Contents/Resources/Keychain.menu"
    mcx::set { "com.apple.mcxMenuExtras:${filename}":                §11.61.2
        value => true,
    }
}
```

## 11.63 Add MIME types

Deploy new MIME types. These are necessary on web servers so that Apache can send the right HTTP Content-Type header when serving files, so that the client on the other end can know what to do with the file it's receiving (e.g., show it directly in Word rather than asking what to do with it).

(Stock Apache httpd generally keeps its own MIME types list, but Red Hat has patched it to use the systemwide list, so we only need change it once.)

```
class mimetypes {
```

This define will help us insert MIME types below. It is only useful in the case where there is a single file extension given for the MIME type.

```
    define mimetype($ext) {
        require augeas
        # mimetype_$name may be more correct but too long to be wieldy.
        augeas { "mimetype_for_$ext":
            # incl + lens instead of context "greatly speeds up execution"
            incl => "/etc/mime.types",
            lens => "Mimetypes.lns",
            changes => [
                "set rules[.='$name'] '$name'",
                "set rules[.='$name']/rule '$ext'",
            ],
        }
    }
```

Office 2007 formats: http://blogs.msdn.com/dmahugh/archive/2006/08/08/692600.aspx

```
    $avoxfod = "application/vnd.openxmlformats-officedocument"
    $ms = "application/vnd.ms"
    $me12 = "macroEnabled.12"

    # indentation style altered to look better in print
    mimetype {
"${avoxfod}.wordprocessingml.document":   ext => "docx";
"${avoxfod}.wordprocessingml.template":   ext => "dotx";
"${avoxfod}.presentationml.slideshow":    ext => "ppsx";
"${avoxfod}.presentationml.presentation": ext => "pptx";
"${avoxfod}.spreadsheetml.sheet":         ext => "xlsx";
"${ms}-word.document.$me12":              ext => "docm";
"${ms}-word.template.$me12":              ext => "dotm";
"${ms}-powerpoint.slideshow.$me12":       ext => "ppsm";
"${ms}-powerpoint.presentation.$me12":    ext => "pptm";
"${ms}-excel.sheet.binary.$me12":         ext => "xlsb";
"${ms}-excel.sheet.$me12":                ext => "xlsm";
"${ms}-xpsdocument":                      ext => "xps" ;
    }
}
```

## 11.64 Mobile code

```
class mobile_code::stig {
    include "mobile_code::stig::${::osfamily}"
}
```

```
class mobile_code::stig::darwin {
    $version_underscores = regsubst(
        $::macosx_productversion_major,
        '\D', '_', 'G')
    $klassname = "${::osfamily}_${version_underscores}"
    include "mobile_code::stig::${klassname}"
}
class mobile_code::stig::darwin_10_6 {}
class mobile_code::stig::darwin_10_9 {
```

Make sure Xprotect Update is running on Macs.

auto: OSX8-00-00755

```
    service { 'com.apple.xprotectupdater':
        ensure => running,
        enable => true,
    }
}
class mobile_code::stig::redhat {
}
```

## 11.65 Mozilla

Configure browsers originating from the Mozilla Foundation, such as Firefox.

### 11.65.1 Wrap 32-bit plugins

This defined resource type makes sure a 32-bit Mozilla plugin is wrapped on 64-bit hosts. 32-bit plugins that come from Red Hat (e.g., `flash-plugin`) will do this themselves, but plugins from other vendors may not.

To use this resource type, first get the 32-bit plugin installed, under `/usr/lib/mozilla/plugins`, the place for 32-bit browser plugins under Red Hat-family Linuxen. Then make a resource of this type, whose name is the name of the plugin file.

Example:

```
  mozilla::wrap_32bit { ’npica.so’: }
```

* * *

```
define mozilla::wrap_32bit {
    require mozilla::wrap_32bit::prerequisites
    case $::osfamily {
        ’RedHat’: {
            $thirtytwo_dir = "/usr/lib/mozilla/plugins"
            $wrapped_dir   = "/usr/lib64/mozilla/plugins-wrapped"
            case $::architecture {
                ’x86_64’: {
                    exec { "wrap_32bit_${name}":
                        onlyif => "test -f ${thirtytwo_dir}/${name}",
                        command => "mozilla-plugin-config -i",
                        creates => "${wrapped_dir}/nswrapper_32_64.${name}",
                    }
                }
                ’i386’: {}
                default: { unimplemented() }
            }
        }
        default: { unimplemented() }
    }
}
```

#### Prerequisites for wrapping 32-bit Mozilla plugins

```
class mozilla::wrap_32bit::prerequisites {
    case $::osfamily {
        ’RedHat’: {
            case $::architecture {
                ’x86_64’: {
```

The package containing the plugin may not know about all the prerequisites necessary for it to happen, so it may not pull them in when it’s installed. We list them here so they will certainly be installed.

```
                       package { [
                           ’nspluginwrapper.i686’,
                           ’nspluginwrapper.x86_64’,
                           ’zlib.i686’,
```

Without these, the Flash plugin and Citrix ICA receiver plugin have successfully installed, but failed to actually run under nspluginwrapper.

```
                           'libcanberra-gtk2.i686',
                           'PackageKit-gtk-module.i686',
                           'gtk2-engines.i686',
                       ]:
                           ensure => present,
                       }
                   }
```

No wrapping is necessary for 32-bit plugins on a 32-bit system.

```
                   'i386': {}
                   default: { unimplemented() }
               }
           }
           default: { unimplemented() }
       }
   }
```

# 11.66 Network

```
class network {
```

Support restarting the network: Other parts of the manifest have `notify => Service["network"]`. That refers here.

```
       service { "network": }
```

Anything interested in restarting the network is likely interested in knowing about which interfaces we're using on this host.

```
       include network::interfaces
   }
```

§11.66.4

RHEL6 does not appear to provide any packages or loadable kernel modules relating to the less-widely-used UDP-Lite, IPX, AppleTalk, DECnet, TIPC or NDP protocols.

RHEL does not run the DHCP client for any interfaces not configured for DHCP, i.e. where it is "not needed."

The DHCP client is configured not to send dynamic DNS updates, surprisingly, in §??.

RHEL6: GEN007140
RHEL6: GEN007200
RHEL6: GEN007260
RHEL6: GEN007320
RHEL6: GEN007540
RHEL6: GEN007760
RHEL5, RHEL6: GEN007840

## 11.66.1 Admin guidance regarding networking

Don't configure any IP tunnels.

admins do GEN007820

## 11.66.2 AirDrop

An ad-hoc Wi-Fi technology from Apple.

### Disable AirDrop

```
class network::airdrop::no {
    include "${name}::${::osfamily}"
}
   class network::airdrop::no::darwin {
       $version_underscores = regsubst(
           $::macosx_productversion_major,
           '\D', '_', 'G')
       $klassname = "${::osfamily}_${version_underscores}"
       include "network::airdrop::no::${klassname}"
   }
```

Snow Leopard doesn't have AirDrop.

```
   class network::airdrop::no::darwin_10_6 {}
   class network::airdrop::no::darwin_10_9 {
       mcx::set { 'com.apple.NetworkBrowser/DisableAirDrop':
           value => true,
       }
   }
   class network::airdrop::no::redhat {}
```

§11.61.2

## 11.66.3 Bluetooth

### Disable Bluetooth

```
class network::bluetooth::no {
    case $::osfamily {
        'redhat': { include network::bluetooth::no::redhat }
        'darwin': { include network::bluetooth::no::darwin }
        default:  { unimplemented() }
    }
}
```

**Disable Bluetooth under Mac OS X** Disable and/or uninstall Bluetooth protocol on Macs.

auto: ECSC-1
auto: OSX00065 M6
auto: OSX8-00-00060
auto: OSX8-00-00065
auto: OSX8-00-00080

```
   class network::bluetooth::no::darwin {
       $exts = '/System/Library/Extensions'
       file {
           "${exts}/IOBluetoothFamily.kext":
               ensure => absent,
               force => true;
           "${exts}/IOBluetoothHIDDriver.kext":
               ensure => absent,
               force => true;
       }
   }
```

**Disable Bluetooth under Red Hat** Disable and/or uninstall Bluetooth protocols. (Notably, this requirement does not say, “unless needed.”)

auto: ECSC-1
auto: GEN007660

```
class network::bluetooth::no::redhat {
    package {
        "gnome-bluetooth.x86_64":               ensure => absent;
        "gnome-bluetooth-debuginfo.i686":       ensure => absent;
        "gnome-bluetooth-debuginfo.x86_64":     ensure => absent;
        "gnome-bluetooth-libs-devel.i686":      ensure => absent;
        "gnome-bluetooth-libs-devel.x86_64":    ensure => absent;
        "pulseaudio-module-bluetooth.x86_64":   ensure => absent;
        "bluez.x86_64":                         ensure => absent;
        "bluez-alsa.i686":                      ensure => absent;
        "bluez-alsa.x86_64":                    ensure => absent;
        "bluez-compat.x86_64":                  ensure => absent;
        "bluez-libs-devel.i686":                ensure => absent;
        "bluez-libs-devel.x86_64":              ensure => absent;
        "bluez-cups.x86_64":                    ensure => absent;
        "bluez-gstreamer.i686":                 ensure => absent;
        "bluez-gstreamer.x86_64":               ensure => absent;
        "bluez-utils.i686":                     ensure => absent;
        "bluez-utils.x86_64":                   ensure => absent;
        "gvfs-obexftp.x86_64":                  ensure => absent;
        "obex-data-server.x86_64":              ensure => absent;
        "obexd.x86_64":                         ensure => absent;
    }
    kernel_module {
        "bnep":      ensure => absent;
        "rfcomm":    ensure => absent;
        "hidp":      ensure => absent;
        "bluetooth": ensure => absent;
        "cmtp":      ensure => absent;
        "sco":       ensure => absent;
        "l2cap":     ensure => absent;
    }
```

"Unprivileged local processes may be able to cause the system to dynamically load a protocol handler by opening a socket using the protocol." (SRG discussion) Prevent this by removing related kernel module files.

```
    file {
        "/lib/modules/$kernelrelease/kernel/net/bluetooth":
            ensure => absent,
            recurse => true,
            recurselimit => 2,
            force => true,
    }
}
```

### Turn off IKE service

Turn off Internet Key Exchange dæmon. This is used in the setup of IPsec VPNs.

```
class network::ike::no {
    include "network::ike::no::${::osfamily}"
}
```

**Turn off the IKE daemon on Macs** `class network::ike::no::darwin {`

Turn off the `racoon` daemon.

auto: OSX8-00-00144

```
    service { ’com.apple.racoon’:
        ensure => stopped,
        enable => false,
    }
}
```

There is no requirement in the RHEL STIG to turn off IKE services.

```
class network::ike::no::redhat {
}
```

### Infiniband non-routers

```
class network::infiniband::non_router {
```

Remove routing protocol daemons from non-routing systems.

auto: ECSC-1
auto: GEN005590

```
    package { "opensm":
        ensure => absent,
    }
}
```

## 11.66.4 Interfaces

Use Facter to figure out which interfaces we’re using. Assume the first one is the one we should configure. Facter takes care of filtering out `lo`, the loopback interface.

```
class network::interfaces {
```

The `$interfaces` variable is a string with all the interfaces separated by commas. First turn it into an array...

```
    $all = split($interfaces,",")
```

then pick out the first member.

```
    $first = $all[0]
}
```

### IPv4 non-routers

```
class network::ipv4::non_router {

    case $::osfamily {
        ’redhat’: {
```

Turn off IPv4 forwarding for non-router Red Hat hosts.

auto: ECSC-1
auto: GEN005600

```
            augeas { "no_ipv4_forwarding":
                context => "/files/etc/sysctl.conf",
                changes => "set net.ipv4.ip_forward 0",
            }
        }
        ’darwin’: {
```

Turn off IPv4 forwarding for non-router Macs.

auto: ECSC-1
auto: GEN005600 M6
auto: OSX8-00-01205

```
               augeas { "no_ipv4_forwarding":
                   context => "/files/etc/sysctl.conf",
                   changes => "set net.inet.ip.forwarding 0",
               }
           }
           default:  { unimplemented() }
       }
   }
```

### IPv4 routers

```
class network::ipv4::router {

    case $::osfamily {
        'redhat': {
```

Turn on IPv4 forwarding for Red Hat hosts designated as routers.

auto: ECSC-1
auto: GEN005600

```
               augeas { "ipv4_forwarding":
                   context => "/files/etc/sysctl.conf",
                   changes => "set net.ipv4.ip_forward 1",
               }
           }
           'darwin': {
```

Turn on IPv4 forwarding for Macs designated as routers.

auto: ECSC-1
auto: GEN005600 M6
auto: OSX8-00-01205

```
               augeas { "ipv4_forwarding":
                   context => "/files/etc/sysctl.conf",
                   changes => "set net.inet.ip.forwarding 1",
               }
           }
           default:  { unimplemented() }
       }
   }
```

## 11.66.5 IPv6

On some networks we need IPv6 enabled. This class enables it. See below for a class which disables it.

```
   class network::ipv6 {
    define ipv6init_yes() {
        augeas { "${name}_turn_on_ipv6":
            changes => "set IPV6INIT yes",
            context =>
        "/files/etc/sysconfig/network-scripts/ifcfg-${name}",
            onlyif => "match \
        /files/etc/sysconfig/network-scripts/ifcfg-${name} \
                size == 1",
        }
    }
    ipv6init_yes {
        "eth0":;
        "eth1":;
        "lo":;
    }
```

Even when IPv6 is enabled, we still must disable 6to4.

```
    include network::ipv6::no_6to4
```

§11.66.5

The localhost6 hosts entry may have been removed. Put it back.

```
    augeas { "hosts_ensure_localhost6":
        context => ’/files/etc/hosts’,
        onlyif => ’match *[ipaddr="::1"] size == 0’,
        changes => [
            ’set 999/ipaddr "::1"’,
            ’set 999/canonical "localhost6"’,
            ’set 999/alias     "localhost6.localdomain6"’,
        ],
    }
```

“The IPv6 protocol handler must not be bound to the network stack unless needed,” and “must be prevented from dynamic loading unless needed.” Hosts which include this class need IPv6. auto: ECSC-1 auto: GEN007700 auto: GEN007720

```
    $n6c = "net.ipv6.conf"
    augeas { "sysctl_disable_ipv6":
        context => "/files/etc/sysctl.conf",
        changes => [
            "set $n6c.all.disable_ipv6 0",
            "set $n6c.default.disable_ipv6 0",
        ],
    }
```

By the same token, the “IPv6 protocol handler” is needed, so we do not uninstall it. N/A: GEN007740

Undo any SSH-specific IPv6 disabling which may have been done.

```
    include ssh::ipv6
}
```

§11.100.6

Non-gateway, IPv6-supporting systems will be configured with a default IPv6 gateway by means of DHCPv6. The DHCPv6 server and its configuration may run on Windows servers, and thus may be outside the scope of this document. N/A: GEN005570

RHEL6 provides no packages or loadable kernel modules that support Teredo. RHEL6: GEN007800

### Turn off IPv6

Air Force TCNO 2008-011-301 requires disabling IPv6. The UNIX SRG requires disabling it “unless needed.”

```
class network::ipv6::no {
    case $::osfamily {
        'redhat': { include network::ipv6::no::redhat }
        'darwin': { include network::ipv6::no::darwin }
        default:  { unimplemented() }
    }
}
```

### Turn off IPv6 under Mac OS X

```
class network::ipv6::no::darwin {
```

Turn off IPv6 “if not being used.” auto: OSX8-00-01240

```
    define on_interface() {
        exec { "turn off IPv6 on ${name}":
            command => "networksetup -setv6off ${name}",
            unless => "networksetup -getinfo ${name} | \
                       grep '^IPv6: Off\$'",
        }
    }
    on_interface { 'Ethernet': }
}
```

### Turn off IPv6 under RHEL

```
class network::ipv6::no::redhat {
    define ipv6init_no() {
        augeas { "${name}_turn_off_ipv6":
            changes => "set IPV6INIT no",
            context =>
        "/files/etc/sysconfig/network-scripts/ifcfg-${name}",
            onlyif => "match \
        /files/etc/sysconfig/network-scripts/ifcfg-${name} \
                size == 1",
        }
    }
    ipv6init_no {
        "eth0":;
        "eth1":;
        "lo":;
    }

    include network::ipv6::no_6to4
```

§11.66.5

When postfix tries to listen on localhost, if it finds an IPv6 address in `/etc/hosts` it will try to listen on it. If we’ve disabled IPv6, it will fail, and then it will quit. So we need to remove that IPv6 address for localhost.

```
        augeas { "hosts_remove_localhost6":
                context => "/files/etc/hosts",
                changes => "rm *[ipaddr='::1']",
        }
```

Unbind the IPv6 protocol from all network interfaces at boot time. auto: ECSC-1

Testing has shown that this also prevents dynamic loading of IPv6 modules by means of attempting to use IPv6. auto: GEN007700 auto: GEN007720

```
        $n6c = "net.ipv6.conf"
        augeas { "sysctl_disable_ipv6":
            context => "/files/etc/sysctl.conf",
            changes => [
                "set $n6c.all.disable_ipv6 1",
                "set $n6c.default.disable_ipv6 1",
            ],
        }
```

This requirement says that the IPv6 protocol handler "must not be installed unless needed." But it could be needed in the future, and its removal is not easily reversible because it isn't in a separate package. So, because it will be "needed" in the future, we settle for disabling it here. N/A: GEN007740

Disabling IPv6 entirely as just above causes an obscure problem with X forwarding in ssh. Not that I would know about that, because we disabled X forwarding.

```
        include ssh::no_ipv6
    }
```

§11.100.8

No hosts on the Eglin network use IPv6, so they are not configured for an IPv6 default gateway. N/A: GEN005570

RHEL6 provides no packages or loadable kernel modules that support Teredo. RHEL6: GEN007800

### Disable 6to4

Disable 6to4. auto: ECSC-1

See /usr/share/doc/initscripts-9.03.17/ipv6-6to4.howto. auto: GEN007780

```
    class network::ipv6::no_6to4 {
        define ipv6to4init_no() {
            augeas { "${name}_turn_off_6to4":
                changes => "set IPV6TO4INIT no",
                context =>
            "/files/etc/sysconfig/network-scripts/ifcfg-${name}",
                onlyif => "match \
            /files/etc/sysconfig/network-scripts/ifcfg-${name} \
                    size == 1",
            }
        }
        ipv6to4init_no {
            "eth0":;
            "eth1":;
            "lo":;
        }
        augeas {
            "network_turn_off_6to4":
                context => "/files/etc/sysconfig/network",
                changes => "rm IPV6_DEFAULTDEV",
                onlyif => "get IPV6_DEFAULTDEV == 'tun6to4'";
        }
    }
```

### IPv6 non-routers

```
class network::ipv6::non_router {

    case $::osfamily {
        'redhat': {
```

Remove IPv6 routing protocol daemons from non-routing systems. auto: ECSC-1 auto: GEN005590

```
                package {
                    "quagga": ensure => absent;
                    "radvd": ensure => absent;
                }
```

Turn off IPv6 forwarding for non-routers. auto: ECSC-1 auto: GEN005610

```
                augeas { "no_ipv6_forwarding":
                    context => "/files/etc/sysctl.conf",
                    changes => "set ipv6.conf.all.forwarding 0",
                }
            }
            'darwin': {
```

The Mac OS X STIG appears to have no requirements for us to do anything here.

```
            }
            default:  { unimplemented() }
        }
    }
```

### 11.66.6 Avoid Ethernet bridging

Do not configure network bridging. auto: ECSC-1 auto: GEN003619

Warn if the system is configured for network bridging. (Removal of the bridge probably can't happen programmatically: it needs too much knowledge of the entire network configuration of a host.)

```
class network::no_bridge {
    include "network::no_bridge::${::osfamily}"
}
class network::no_bridge::darwin {}
class network::no_bridge::redhat {
```

Make sure we have `brctl`.

```
    package { "bridge-utils":
        ensure => present,
    }
```

Use it to make sure there are no bridges in operation.

```
    exec { "no_bridges":
        path => "/bin:/sbin:/usr/bin:/usr/sbin",
```

`brctl show` always shows a header; skip it. After that, if there are any lines of output, we have a situation.

```
        onlyif => "test ‘brctl show | tail -n +2 | wc -l‘ -ne 0",
        command => "echo ETHERNET BRIDGING CONFIGURED; \
                    brctl show",
        logoutput => true,
        loglevel => err,
    }
}
```

### 11.66.7 Disable DCCP

Disable the Datagram Congestion Control Protocol (DCCP) "unless required." We do not need it.

auto: ECSC-1
auto: GEN007080

```
class network::no_dccp {
    kernel_module {
        "dccp_diag": ensure => absent;
        "dccp_ipv4": ensure => absent;
        "dccp_ipv6": ensure => absent;
        "dccp_probe": ensure => absent;
        "dccp": ensure => absent;
    }
```

"Unprivileged local processes may be able to cause the system to dynamically load a protocol handler by opening a socket using the protocol." (SRG discussion) Prevent this by removing related kernel module files.

```
    file {
        "/lib/modules/$kernelrelease/kernel/net/dccp":
            ensure => absent,
            recurse => true,
            recurselimit => 1,
            force => true,
    }
}
```

### 11.66.8 Don't send ICMP echo replies

This is known as “stealth mode” on Macs. Oo, stealthy.

```
class network::no_icmp_echo {
    include "network::no_icmp_echo::${::osfamily}"
}
class network::no_icmp_echo::darwin {
    $version_underscores = regsubst(
        $::macosx_productversion_major,
        '\D', '_', 'G')
    $klassname = "${::osfamily}_${version_underscores}"
    include "network::no_icmp_echo::${klassname}"
}
class network::no_icmp_echo::darwin_10_6 {}
class network::no_icmp_echo::darwin_10_9 {
```

Enable “Stealth Mode” on the OSX firewall auto: OSX8-00-01245

```
    $sffw = '/usr/libexec/ApplicationFirewall/socketfilterfw'
    exec { 'turn on stealth mode':
        command => "${sffw} --setstealthmode on",
        unless => "${sffw} --getstealthmode | grep enabled",
    }
}
```

### 11.66.9 Disable RDS

Disable and/or uninstall the Reliable Datagram Sockets (RDS) protocol “unless required.”

auto: ECSC-1
auto: GEN007480

```
class network::no_rds {
    package {
        "rds-tools": ensure => absent;
        "rds-tools-debuginfo": ensure => absent;
    }
    kernel_module {
        "rds": ensure => absent;
        "rds_rdma": ensure => absent;
        "rds_tcp": ensure => absent;
    }
```

“Unprivileged local processes may be able to cause the system to dynamically load a protocol handler by opening a socket using the protocol.” (SRG discussion) Prevent this by removing related kernel module files.

```
    file {
        "/lib/modules/$kernelrelease/kernel/net/rds":
            ensure => absent,
            recurse => true,
            recurselimit => 1,
            force => true,
    }
}
```

### 11.66.10 Disable SCTP

Disable the Stream Control Transmission Protocol (SCTP) "unless required." We do not need it.

auto: ECSC-1
auto: GEN007020

```
class network::no_sctp {
    package {
        "lksctp-tools": ensure => absent;
        "lksctp-tools-debuginfo": ensure => absent;
        "lksctp-tools-devel": ensure => absent;
        "lksctp-tools-doc": ensure => absent;
    }
    kernel_module { "sctp": ensure => absent }
```

"Unprivileged local processes may be able to cause the system to dynamically load a protocol handler by opening a socket using the protocol." (SRG discussion) Prevent this by removing related kernel module files.

```
    file {
        "/lib/modules/$kernelrelease/kernel/net/sctp":
            ensure => absent,
            recurse => true,
            recurselimit => 1,
            force => true,
    }
}
class network::no_sharing {
    include "${name}::${::osfamily}"
}
class network::no_sharing::darwin {
    service { 'com.apple.InternetSharing':
        ensure => stopped,
        enable => false,
    }
}
```

### 11.66.11 Non-routers

A host may be designated as a router for any of several protocols. This class is for use on hosts which do not route at all.

```
class network::non_router {
    include network::ipv4::non_router                    §11.66.4
    include network::ipv6::non_router                    §11.66.5
    include network::infiniband::non_router              §11.66.3
}
```

### 11.66.12 STIG-required network configuration

```
class network::stig {
```

#### Common implementations of compliance

Control ownership and permissions of the `services` file.

auto: ECLP-1
auto: GEN003760 M6
auto: GEN003770 M6
auto: GEN003780 M6
auto: ECLP-1
auto: GEN003760
auto: GEN003770
auto: GEN003780

```
        file { "/etc/services":
            owner => root, group => 0, mode => 0644,
        }
```

Remove extended ACLs on the `services` file.

auto: ECLP-1
auto: GEN003790

```
        no_ext_acl { "/etc/services": }
```

**Platform-specific implementations of compliance**

```
    case $::osfamily {
        'RedHat': { include network::stig::redhat }
        'Darwin': { include network::stig::darwin }
        default:  { unimplemented() }
    }
}
```

### 11.66.13 STIG-required network configuration under Mac OS X

```
class network::stig::darwin {
```

First ensure that sysctl.conf exists; the STIG implies that it may not.

For least surprise for policy maintainers, this should probably go in a more generic module than “network.”

```
        file { '/etc/sysctl.conf':
            ensure => present,
            owner => root, group => 0, mode => 0644,
        }
```

All of our edits will be to sysctl.conf.

```
        Augeas {
            context => "/files/etc/sysctl.conf",
        }


        augeas {
```

Configure the system to block ICMP timestamp requests.

auto: ECSC-1
auto: GEN003602 M6
auto: OSX8-00-01220

```
            "block_icmp_timestamp_requests":
                changes => "set net.inet.icmp.timestamp 1";
```

Configure the system to ignore ICMP pings sent to a broadcast address.

auto: ECSC-1
auto: GEN003603 M6
auto: OSX8-00-01190

```
            "ignore_icmpv4_broadcast_echoreq":
                changes => "set net.inet.icmp.bmcastecho 1";
```

Configure the system to “prevent local applications from generating source-routed packets.”

auto: ECSC-1
auto: GEN003606 M6
auto: OSX8-00-01215

```
            "prevent_outgoing_source_routing":
                changes => "set net.inet.ip.sourceroute 0";
```

Configure the system to “not accept source-routed IPv4 packets.”

auto: ECSC-1
auto: GEN003607 M6
auto: OSX8-00-01195

```
            "reject_ipv4_source_routed":
                changes => "set net.inet.ip.accept_sourceroute 0";
```

Configure the system to “ignore ICMPv4 redirect messages.”

auto: ECSC-1
auto: GEN003609 M6
auto: OSX8-00-01200

A typo in the earlier Mac OS X stig said to make this 0.

```
            "ignore_icmpv4_redirects":
                changes => "set net.inet.icmp.drop_redirect 1";
```

Prevent the system from sending ICMPv4 redirect messages.

auto: ECSC-1
auto: GEN003610 M6
auto: OSX8-00-01210

```
        "dont_send_icmpv4_redirects":
            changes => "set net.inet.ip.redirect 0";
    }
    include network::ike::no
}
```

§11.66.3

### 11.66.14 STIG-required network configuration under Red Hat

```
class network::stig::redhat {
```

All of our edits will be to sysctl.conf.

```
    Augeas {
        context => "/files/etc/sysctl.conf",
    }
```

Abbreviations used below:

```
    $n4 = "net.ipv4"
    $n4ca = "net.ipv4.conf.all"
    $n6ca = "net.ipv6.conf.all"
    augeas {
```

Set the TCP backlog queue size appropriately. auto: ECSC-1

```
        "increase_tcp_syn_backlog":
            changes => "set $n4.tcp_max_syn_backlog 1280";
```

auto: GEN003601

Configure the system to ignore ICMP pings sent to a broadcast address. auto: ECSC-1

```
        "ignore_icmpv4_broadcast_echoreq":
            changes => "set $n4.icmp_echo_ignore_broadcasts 1";
```

auto: GEN003603

Configure the system to ignore source-routed IPv4 packets. auto: ECSC-1

Note that this setting is not enough to satisfy all of the STIG requirements regarding IPv4 source-routed packets. See §11.48. auto: GEN003607

```
        "reject_ipv4_source_routed":
            changes => "set $n4ca.accept_source_route 0";
```

Disable Proxy ARP. auto: ECSC-1

```
        "disable_proxy_arp":
            changes => "set $n4ca.proxy_arp 0";
```

auto: GEN003608

Cause the system to ignore ICMPv4 redirect messages. auto: ECSC-1

```
        "ignore_icmpv4_redirects":
            changes => "set $n4ca.accept_redirects 0";
```

auto: GEN003609

Prevent the system from sending ICMPv4 redirect messages. auto: ECSC-1

```
        "dont_send_icmpv4_redirects":
            changes => "set $n4ca.send_redirects 0";
```

auto: GEN003610

Cause "martian packets" to be logged. auto: ECAT-1

```
        "log_martian_packets":
            changes => "set $n4ca.log_martians 1";
```

auto: GEN003611

Enable TCP syncookies. auto: ECSC-1

```
        "tcp_syncookies":
            changes => "set $n4.tcp_syncookies 1";
```

auto: GEN003612

Enable the reverse-path filter. auto: ECSC-1

Note: according to https://access.redhat.com/knowledge/solutions/53031, the meaning of "1" differs between RHEL5 and RHEL6; in RHEL5 it means "do source validation by reversed path" (versus not doing it) and in RHEL6 it means "Strict mode as defined in RFC3704 Strict Reverse Path" (rather than no validation or "loose mode"). In both cases this is the setting we want. auto: GEN003613

```
        "reverse_path_filter":
            changes => "set $n4ca.rp_filter 1";
```

Cause the system to ignore ICMPv6 redirect messages.

auto: ECSC-1
auto: GEN007860

```
        "ignore_icmpv6_redirects":
            changes => "set $n6ca.accept_redirects 0";
```

Configure the system to ignore source-routed IPv6 packets.

auto: ECSC-1
auto: GEN007940

```
        "reject_ipv6_source_routed":
            changes => "set $n6ca.accept_source_route 0";
    }
```

Some IPv6 requirements would be implemented with `ip6tables`, as their corresponding IPv4 requirements are with `iptables`.

N/A: GEN007880
N/A: GEN007920
N/A: GEN007950

Someone made an IPv6 `rp_filter` patch for the Linux kernel in 2006. It appears that that patch is not in the RHEL kernel. More investigation is needed, but not warranted at this time because we are not deploying IPv6 yet.

N/A: GEN007900

```
    file { "/etc/sysctl.conf":
        owner => root, group => 0, mode => 0600,
    }
    no_ext_acl { "/etc/sysctl.conf": }
```

auto: ECLP-1
auto: GEN000000-LNX00480
auto: GEN000000-LNX00500
auto: GEN000000-LNX00520
auto: ECLP-1
auto: GEN000000-LNX00530

```
    include network::no_dccp
    include network::no_rds
    include network::no_sctp
```

§11.66.7
§11.66.9
§11.66.10

Any system which is not a router should include the `network::non_router` class for STIG compliance; but this class is generic enough that it may be included on designated routers.

```
    # include network::non_router
```

Any host not using IPv6 should include network::ipv6::no.

```
}
```

### 11.66.15 WiFi (IEEE 802.11)

### 11.66.16 Disable WiFi

```
class network::wifi::no {
    case $::osfamily {
        'darwin': { include network::wifi::no::darwin }
        default:  { unimplemented() }
    }
}
```

### 11.66.17 Disable WiFi on Macs

```
class network::wifi::no::darwin {
```

Disable Wi-Fi on Macs by removing the driver files that support it.

auto: ECSC-1
auto: OSX00060 M6

```
    $exts = '/System/Library/Extensions'
```

```
file { "${exts}/IO80211Family.kext":
    ensure => absent,
    force => true,
}

$nse = 'networkserviceenabled'
exec { 'disable AirPort network service':
    command => 'networksetup -set${nse} AirPort off',
    onlyif => 'networksetup -get${nse} | grep Enabled',
}
exec { 'disable Wi-Fi network service':
    command => 'networksetup -set${nse} Wi-Fi off',
    onlyif => 'networksetup -get${nse} | grep Enabled',
}
```

Turn off AirPort power on Macs if “unused.” auto: ECSC-1

This one is a little tricky because you have to give a network interface name, not a network service name. And it's theoretically possible for a network service to own multiple interfaces. auto: OSX00385 M6

```
exec { 'turn off AirPort power':
```

So—if any Wi-Fi or AirPort devices have power On...

```
    onlyif => "\
          networksetup -listnetworkserviceorder | \
          grep -A1 'Wi-Fi\\|AirPort' | \
          grep -o 'Device: [a-z0-9]\\+' | \
          cut -d: -f2 | \
          xargs -n 1 networksetup -getairportpower | \
          grep 'On\$'",
```

...turn off power to all Wi-Fi or AirPort devices.

```
        command => "\
              networksetup -listnetworkserviceorder | \
              grep -A1 'Wi-Fi\\|AirPort' | \
              grep -o 'Device: [a-z0-9]\\+' | \
              cut -d: -f2 | \
              xargs -I % networksetup -setairportpower % Off",
    }

This is done using System Preferences. Open the Network section;
for each active AirPort interface in the pane on the left, click the
interface, and click ``Turn AirPort Off.'' After all of this, click
``Apply.''

This is done using System Preferences.
\doneby{admins}{macosxstig}{OSX00400 M6}%
Turn off IPv6 on Macs ``if not being used.''

This is done using System Preferences. Open the Network section;
for each active interface in the pane on the left, click the interface,
click the ``Advanced...'' button toward the lower right, and in the TCP/IP
tab, change the ``Configure IPv6'' setting to ``Off.'' After all of this,
click ``Apply.''
}
```

## 11.67 Network tools

Policies relating to software used for network analysis and debugging.

### 11.67.1 Remove network analysis tools

Remove tools used for packet capture and analysis. auto: GEN003865

```
class stig_misc::network_tools {
    package {
        "iptraf": ensure => absent;
        "mtr-gtk": ensure => absent;
        "mtr": ensure => absent, require => Package['mtr-gtk'];
        "nmap": ensure => absent;
        "wireshark-gnome": ensure => absent;
        "wireshark": ensure => absent, require => Package['wireshark-gnome'];
```

This one may be innocuous—but once I had it installed and it made a log message about root logging in, *every five seconds*. Kill it with fire!

```
        "mrtg": ensure => absent;
        "tcpdump": ensure => absent;
    }
```

Make the `traceroute` utility executable only by root. auto: ECLP-1

```
$traceroute = $::osfamily ? {
```

auto: GEN003960 M6
auto: GEN003980 M6
auto: GEN004000 M6

We'll throw in `traceroute6` for free.

auto: ECLP-1
auto: GEN003960
auto: GEN003980
auto: GEN004000

```
        'redhat' => [ '/bin/traceroute', '/bin/traceroute6' ],
        'darwin' => '/usr/sbin/traceroute',
        default  => unimplemented,
    }
    file { $traceroute:
        owner => root, group => 0, mode => 0700;
    }
```

Remove extended ACLs on the traceroute executable.

auto: ECLP-1
auto: GEN004010 M6

```
    no_ext_acl { $traceroute: }
}
```

auto: ECLP-1
auto: GEN004010

### 11.67.2 Remove network analysis tools

Remove tools used for packet capture and analysis.

auto: GEN003865

```
class network_tools::remove {
    package {
        "iptraf": ensure => absent;
        "mtr-gtk": ensure => absent;
        "mtr": ensure => absent, require => Package['mtr-gtk'];
        "nmap": ensure => absent;
        "wireshark-gnome": ensure => absent;
        "wireshark": ensure => absent, require => Package['wireshark-gnome'];
```

This one may be innocuous—but once I had it installed and it made a log message about root logging in, *every five seconds.* Kill it with fire!

```
        "mrtg": ensure => absent;
        "tcpdump": ensure => absent;
    }
}
```

### 11.67.3 Lock down essential network analysis tools

For network tools that can't or shouldn't be removed, lock down access to them.

```
class network_tools::stig_essential {
```

Make the traceroute utility executable only by root.

auto: ECLP-1
auto: GEN003960 M6
auto: GEN003980 M6
auto: GEN004000 M6

```
    $traceroute = $::osfamily ? {
```

We'll throw in traceroute6 for free.

auto: ECLP-1
auto: GEN003960
auto: GEN003980
auto: GEN004000

```
        'redhat' => [ '/bin/traceroute', '/bin/traceroute6' ],
        'darwin' => '/usr/sbin/traceroute',
        default  => unimplemented,
    }
    file { $traceroute:
        owner => root, group => 0, mode => 0700;
    }
```

Remove extended ACLs on the traceroute executable.

auto: ECLP-1
auto: GEN004010 M6

```
    no_ext_acl { $traceroute: }
}
```

auto: ECLP-1
auto: GEN004010

```
class network_tools::tcpdump {
    package { "tcpdump":
        ensure => present,
    }
}
```

```
class network_tools::wireshark {
    package { ["wireshark-gnome", "wireshark"]:
        ensure => present,
    }
}
```

# 11.68 NetworkManager

## 11.68.1 Restrict network changes to admins

Don’t let users configure network interfaces: require authentication of an administrator to do this.

auto: ECLP-1
auto: GEN003581

*N.B.* This will cause trouble on any host which may change networks in the normal course of duty—like a laptop.

```
class networkmanager::admin_auth {
    case $osfamily {
        RedHat: {
            case $operatingsystemrelease {
```

RHEL6 comes with NetworkManager, and it works and lets users do things to configure the network unless it’s configured otherwise. Here we configure it to require admin authentication for any changes.

```
                /^6\..*/: {
```

Get rid of the pre-policykit::rule file.

```
                    file {
"/etc/polkit-1/localauthority/90-mandatory.d/\
50-mil.af.eglin.afseo.admin-network.pkla":
                        ensure => absent,
                    }

                    policykit::rule { ’admin-auth-network’:        §11.77.3
                        description =>
’only admins can change network settings’,
                        identity => ’*’,
                        action =>
"org.freedesktop.NetworkManager.*;\
org.freedesktop.network-manager-settings.*",
                    }
                }
```

While RHEL5 comes with NetworkManager, it appears that it doesn’t come with PolicyKit, and it also doesn’t appear that you can do anything with the network settings without being an admin, as required.

```
                /^5\..*/: {}

                default: { unimplemented() }
            }
        }
```

Darwin doesn’t have NetworkManager.

```
        Darwin: {}
        default: { unimplemented() }
    }
}
```

## 11.69 NFS version 3

Most NFS filesystems are mounted using the automounter; see 11.42.4 and look in the Defined Resource Types index.

To use NFSv3 we must do remote procedure calls (RPC). This requires a portmapper or binder; under RHEL5 this is called `portmap` and under RHEL6 it is `rpcbind`.

There's also a statd and maybe a lockd which need to be installed and running, which are contacted via RPC.

```
class nfs {
```

In §11.35.1, the pieces of policy for each OS and version are split out into separate files. Here they are all written in two big case statements. For further implementations, decide which is simpler and better.

```
    case $osfamily {
        RedHat: {
            $portmap = $operatingsystemrelease ? {
                /^6.*/ => "rpcbind",
                /^5.*/ => "portmap",
                default => unimplemented(),
            }
            package { $portmap: ensure => present }
            tcp_wrappers::allow { $portmap:                  §11.106.1
```

```
                from => "127.0.0.1",
            }
            service { $portmap:
                require => [
                    Package[$portmap],
                    Tcp_wrappers::Allow[$portmap],
                ],
                enable => true,
                ensure => running,
            }

            package { "nfs-utils":
                require => Package[$portmap],
                ensure => present,
            }
            service { "nfslock":
                require => [
                    Service[$portmap],
                    Package["nfs-utils"],
                ],
                enable => true,
                ensure => running,
            }
        }
```

Mac OS X Snow Leopard is rather more monolithically installed, and comes with NFS support.

```
        darwin: {}
        default: { unimplemented() }
    }
}
class nfs::allow($from) inherits nfs {
    case $::osfamily {
        ’RedHat’: {
            case $::operatingsystemrelease {
                /^6\..*/: {
                    Tcp_wrappers::Allow[’rpcbind’] {
                        from +> $from,
                    }
                    tcp_wrappers::allow { ’mountd’:                §11.106.1
                        from => $from,
                    }
                    tcp_wrappers::allow { ’nfs’:                   §11.106.1
```

```
                        from => $from,
                    }
                    service { 'nfs':
                        enable => true,
                        ensure => running,
                    }
                }
                default: { unimplemented() }
            }
        }
        default: { unimplemented() }
    }
}
```

### 11.69.1 ARX workaround

According to `http://support.f5.com/kb/en-us/solutions/public/14000/400/sol14478.html?sr=35037786`, a change was made in RHEL 6.3 to enable more remote procedure calls to be in-flight between the client system and an NFS server. The ARX is ill-equipped to handle many in-flight RPCs, though, so we must limit the RHEL systems back to previous behavior to avoid flooding the ARX.

```
class nfs::arx {
    case $::osfamily {
        'RedHat': {
            file { '/etc/modprobe.d/sunrpc.conf':
                owner => root, group => 0, mode => 0644,
                content => "
options sunrpc tcp_max_slot_table_entries=16
",
            }
        }
        default: {}
    }
}
class nfs::client::no {
    include "nfs::client::no::${::osfamily}"
}
```

### 11.69.2 Disable NFS client

This class disables services that are needed both for NFS servers and for NFS clients.

If you need your Macs to be NFS clients, do not include this class.

```
class nfs::client::no::darwin {
```

Disable the NFS lock dæmon. auto: OSX8-00-00142

```
    service { 'com.apple.lockd':
        enable => false,
        ensure => stopped,
    }
```

Disable the NFS stat dæmon. auto: OSX8-00-00143

```
    service { 'com.apple.statd':
        enable => false,
        ensure => stopped,
    }
}
```

### 11.69.3 Remove rpcbind

Remove the rpcbind or portmap service wherever it is not necessary (it is necessary where NFS is in use).

auto: ECSC-1
auto: GEN003810
auto: GEN003815

```
class nfs::client::no::redhat {
    case $operatingsystemrelease {
        /6\..*/: {
```

We have to do this using an exec because the package type can only remove one package at a time, but nfs-utils and nfs-utils-lib each depend on the other, so neither can be successfully removed by itself. See `http://projects.puppetlabs.com/issues/2198`.

```
            exec { 'remove NFS client packages':
                command => "/usr/bin/yum -y remove \
                rpcbind \
                nfs-utils \
                nfs-utils-lib",
                onlyif => "/bin/rpm -q \
                rpcbind \
                nfs-utils \
                nfs-utils-lib",
            }
        }
        /5\..*/: {
            package {
                "portmap": ensure => absent;
                "ypbind": ensure => absent;
                "nfs-utils": ensure => absent;
            }
        }
        default: { unimplemented() }
    }
}
```

### 11.69.4 Remove rpcbind

Remove the rpcbind or portmap service wherever it is not necessary (it is necessary where NFS is in use).

auto: ECSC-1
auto: GEN003810
auto: GEN003815

```
class nfs::no {
    case $osfamily {
        RedHat: {
            case $operatingsystemrelease {
                /6\..*/: {
```

We have to do this using an exec because the package type can only remove one package at a time, but nfs-utils and nfs-utils-lib each depend on the other, so neither can be successfully removed by itself. See http://projects.puppetlabs.com/issues/2198.

```
                    exec { 'remove NFS client packages':
                        command => "/usr/bin/yum -y remove \
                            rpcbind \
                            nfs-utils \
                            nfs-utils-lib",
                        onlyif => "/bin/rpm -q \
                            rpcbind \
                            nfs-utils \
                            nfs-utils-lib",
                    }
                }
                /5\..*/: {
                    package {
                        "portmap": ensure => absent;
                        "ypbind": ensure => absent;
                        "nfs-utils": ensure => absent;
                    }
                }
                default: { unimplemented() }
            }
        }
        default: { unimplemented() }
    }
}
class nfs::server::no {
    include "nfs::server::no::${::osfamily}"
}
```

**Disable NFS file sharing on Macs**

```
class nfs::server::no::darwin {
```

Disable file sharing via NFS. auto: OSX8-00-00141

```
    service { 'com.apple.nfsd':
        enable => false,
        ensure => stopped,
    }
}
```

**Turn off NFS server on Red Hat machines** We can't remove the NFS server software on Red Hat because it comes in the same package as the NFS client software. But we can stop the services.

```
class nfs::server::no::redhat {
    service { 'nfs':
        ensure => stopped,
        enable => false,
    }
}
```

### 11.69.5 STIG-required NFS configuration

```
class nfs::stig {
       include nfs
```

§11.69

Control ownership and permissions of the `exports` file.

auto: ECCD-1

```
       file { "/etc/exports":
           owner => root, group => 0, mode => 0644,
       }
```

auto: ECLP-1
auto: GEN005740
auto: GEN005750
auto: GEN005760

Remove extended ACLs on the `exports` file.

auto: ECLP-1

```
       no_ext_acl { "/etc/exports": }
```

auto: GEN005770

Remove the insecure_locks export option wherever it exists.

auto: IAIA-1

```
       augeas { 'remove_insecure_locks_in_exports':
           context => '/files/etc/exports',
           changes => 'rm dir/client/option[.="insecure_locks"]',
       }
   }
```

auto: GEN000000-LNX00560

## 11.70 NIS (Network Information System)

We don't use NIS.

### 11.70.1 Remove NIS lookup directives

A plus (+) when found alone in any of several system files means to use NIS to look up additional entries for that file. We don't use NIS, so this should not be the case anywhere.

```
   class nis::no_pluses {
       define no_pluses_in() {
           exec { "no_pluses_in_${name}":
               command => "/bin/echo \
                   ---- FOUND A PLUS CHARACTER IN ${name} ----",
               onlyif => [
                   "test -f ${name}",
                   "grep '^+:*' ${name} >&/dev/null",
               ],
               logoutput => true,
               loglevel => err,
           }
       }
```

Make sure there are no pluses in system authentication data files, causing possibly insecure NIS lookups.

auto: ECCD-1
auto: GEN001980

Note that this does not remove pluses from files in home directories as required by this PDI, *i.e.*, `.rhosts` and `.shosts`. Note further, though, that the `.rhosts` file is supposed to be read by `rsh`, `rlogin`, `rexec` and the like, which tools §11.101 uninstalls; and the `.shosts` file is supposed to be read by `ssh`, but §11.100.10 tells the SSH server not to pay any attention to it. Note even further that §11.41.3 removes `.rhosts` and `.shosts` files from home directories, which effectively ensures that they don't contain pluses.

```
    no_pluses_in {
        "/etc/passwd":;
        "/etc/shadow":;
        "/etc/group":;
        "/etc/hosts.equiv":;
        "/etc/shosts.equiv":;
    }
}
```

## 11.71 NTP

Configure the Network Time Protocol (NTP) service.

On all networks where timeservers exist, use **ntpd** to keep continuous synchronization with the timeservers.

auto: ECSC-1
auto: GEN000241

Here is some background regarding NTP implementation interoperability as it relates to cryptographic authentication of time data:

According to [1], §1, time services on Windows support a subset of NTPv3 ( [12]), not NTPv4 ( [11], [8]), and §3.2.5.1 says, "[T]he authentication mechanism defined in RFC 1305 Appendix C.1 is not supported." This means that Windows time services support neither the symmetric key authentication of NTPv3 nor the Autokey of NTPv4 as cryptographic means of authenticating time data, but only support the Microsoft-proprietary means of time data authentication within the context of an Active Directory domain. These proprietary extensions to NTP are not supported by the NTP software used in RHEL 5 and 6, which is the reference implementation of NTPv4 from the University of Delaware.

```
class ntp {
    include "ntp::${::osfamily}"
}
class ntp::darwin {
```

Make sure the Mac is using NTP.

auto: OSX8-00-00325

```
    exec { 'enable NTP':
        path => ['/bin', '/sbin', '/usr/bin', '/usr/sbin'],
        command => 'systemsetup -setusingnetworktime on',
        unless => 'systemsetup -getusingnetworktime | grep On',
    }
```

The network time server must also be set; this is site-specific.

```
}
class ntp::redhat {
    $major_release = regsubst(
        $operatingsystemrelease,
        '[^0-9].*', '', 'G')

    include "ntp::redhat_${major_release}"
}
```

```
class ntp::redhat_5 {
    package { ’ntp’:
        ensure => present,
    }
    service { ’ntpd’:
        enable => true,
        ensure => running,
    }
```

Control ownership and permissions of the `ntp.conf` file. auto: ECLP-1

```
    file { "/etc/ntp.conf":
        owner => root, group => 0, mode => 0640,
    }
```

auto: GEN000250
auto: GEN000251
auto: GEN000252

Remove extended ACLs on the `ntp.conf` file. auto: ECLP-1

```
    no_ext_acl { "/etc/ntp.conf": }
}
```

auto: GEN000253

```
class ntp::redhat_6 {
    include ntp::redhat_5
}
```

§11.71

## 11.72 NVIDIA

Deal with NVIDIA hardware.

### 11.72.1 Proprietary drivers

Install proprietary NVIDIA graphics drivers for best graphics performance. The original documentation for this process is the README for the NVIDIA driver.

Assumptions that we are running Red Hat Enterprise Linux or a derivative are common in this class.

The installer_dir should be a directory that always exists, and contains at least two shell scripts `latest-x86_64` and `latest-i386`. Most likely it will be a directory containing a bunch of Linux NVIDIA driver installers, with one symlinked as `latest-x86_64` and one symlinked as `latest-i386`. For desktops this may be a networked directory; for laptops it should be a cached copy on the local hard drive, because if someone takes the laptop off the network, and a new kernel has been installed, but never yet booted, the video driver will need to be reinstalled without reference to the network.

```
class nvidia::proprietary($installer_dir) {

    if $::has_nvidia_graphics_card == ’true’ {
```

The NVIDIA driver must be installed when X is not running. Rather than figure out how to safely kill the X server and boot the console user off, we just install an init script that will install the driver at boot time.

Nowadays, graphical boot is common because it looks slick, but for this purpose it gets in our way. Turn it off:

```
        include grub::rhgb::no
```

§11.40.4

The driver builds some adapter code, then links it with the proprietary driver code to arrive at a kernel module. To do this, it needs the C compiler, and the kernel development files.

```
package { [
        ’gcc’,
        ’kernel-devel’,
    ]:
    ensure => present,
}
require common_packages::make
```

Now install the init script.

```
file { "/etc/rc.d/init.d/nvidia-rebuild":
    owner => root, group => 0, mode => 0755,
    content => template(’nvidia/nvidia-rebuild.sh.erb’),
```

If the X server is not installed before the proprietary NVIDIA driver, the driver won’t install all of its files properly.

```
    require => Package[’xorg-x11-server-Xorg’],
}
```

With the script installed the service can be added.

```
exec { ’add_nvidia_rebuild_service’:
    command => ’/sbin/chkconfig --add nvidia-rebuild’,
    refreshonly => true,
    subscribe => File[’/etc/rc.d/init.d/nvidia-rebuild’],
}
```

The init script defines an `nvidia-rebuild` service; enable it so it will be started at boot. We don’t want to start it immediately: if this isn’t boot time, there’s most likely an X server running, so it would fail.

```
service { ’nvidia-rebuild’:
    enable => true,
    require => [
        File[’/etc/rc.d/init.d/nvidia-rebuild’],
        Exec[’add_nvidia_rebuild_service’],
    ],
}
```

Place an X configuration file so that X will use the nvidia driver. In order to allow further configuration, like TwinView or rotated displays, we won’t replace the configuration if it’s already there.

```
file { ’/etc/X11/xorg.conf.d/01-nvidia.conf’:
    owner => root, group => 0, mode => 0644,
    replace => false,
    source => "puppet:///modules/nvidia/01-nvidia.conf",
```

The xorg.conf.d directory is provided by this X server package. (And maybe others.)

```
    require => Class[’xserver’],
}
```

The NVIDIA proprietary driver will not install if the Nouveau driver is in use. So to install the proprietary driver we must disable the Nouveau driver:

```
        if $::using_nouveau_driver == ’true’ {
```

Change the GRUB config to prevent the initrd from loading the Nouveau driver.

```
            include grub::nouveau::no                              §11.40.2
```

Prevent the system after boot from automatically loading Nouveau.

```
            file { ’/etc/modprobe.d/disable-nouveau.conf’:
                owner => root, group => 0, mode => 0644,
                content => "blacklist nouveau\n\
options nouveau modeset=0\n",
            }
        }
```

Let admins sudo to run the driver installer manually if need be.

```
        sudo::auditable::command_alias { ’NVIDIA_DRIVERS’:         §11.104.3
            type => ’exec’,
            commands => [
                "${installer_dir}/NVIDIA*.run",
                ],
        }
    }
}
```

## 11.73 PackageKit

PackageKit helps normal users install packages. It’s intended to enable security and bugfix updates on computers where there is no real administrator—like home desktops. In general, any environment where we are running Puppet is an environment with a real administrator, and where there are admins, users should not be making decisions about software updates.

Some parts of PackageKit look useful: for example, its service pack functionality. Admins can use `pkcon`, `pkgenpack`, or `gpk-application` to access these parts; meanwhile, users should not be bothered with anything relating to software packages.

```
class packagekit {
    include packagekit::no_icon                                §11.73.3
    include packagekit::admin_auth                             §11.73.1
    include packagekit::no_auto                                §11.73.2
    include packagekit::no_notify                              §11.73.4
}
```

### 11.73.1 Require admin authentication

Keep normal users from installing or removing software.

```
class packagekit::admin_auth {
    case $osfamily {
        RedHat: {
            case $operatingsystemrelease {

                /^6\..*/: {
```

Get rid of the pre-policykit::rule file.

```
                        file {
"/etc/polkit-1/localauthority/90-mandatory.d/\
50-mil.af.eglin.afseo.admin-packagekit.pkla":
                            ensure => absent,
                        }

                        policykit::rule { 'admin-packagekit':
```

§11.77.3

```
                            description =>
'require admin authn for package actions',
                            identity => '*',
                            action =>
'org.freedesktop.packagekit.*',
                        }
                }
```

RHEL5 includes neither PackageKit nor PolicyKit, so users already can't install or remove software without admin privileges.

```
                /^5\..*/: {}

                default: { unimplemented() }
            }
        }
    }
}
```

### 11.73.2 Turn off automatic updates

Make sure we don't automatically obtain any updates. auto: ECSC-1

auto: GEN008820

```
class packagekit::no_auto {
    gconf {
        "/apps/gnome-packagekit/update-icon/auto_update":
            type => string, value => "none";
    }
}
```

### 11.73.3 Remove package update icon

Users can't usefully install package updates. Don't bother showing them the icon.

```
class packagekit::no_icon {
```

This works for RHEL6.

```
    file { "/etc/xdg/autostart/gpk-update-icon.desktop":
        ensure => absent,
    }
```

This works for RHEL5.

```
    file { "/etc/xdg/autostart/puplet.desktop":
        ensure => absent,
    }
}
```

### 11.73.4 Turn off notifications

For users who somehow have the `gpk-update-icon` running, turn off notifications to them about things which, after all, they can't control.

```
class packagekit::no_notify {
    Gconf {
        type => bool, value => false,
    }
    $agpui = "/apps/gnome-packagekit/update-icon"
    gconf {
        "$agpui/notify_update_failed":;
        "$agpui/notify_critical":;
        "$agpui/notify_available":;
        "$agpui/notify_distro_upgrades":;
        "$agpui/notify_complete":;
        "$agpui/notify_update_started":;
        "$agpui/notify_update_complete_restart":;
        "$agpui/notify_update_complete":;
        "$agpui/notify_message":;
        "$agpui/notify_errors":;
        "$agpui/notify_update_not_battery":;
    }
}
```

## 11.74 Configure PAM

As of this writing, most PAM configuration happens outside this section, but at some point it will be brought together.

This requirement deserves a hard look. It appears from a reading of the manual pages that the pam_console PAM module has little, if anything, to do with the asserted vulnerability. If that is true, disabling it would not result in the security outcome claimed; meanwhile, disabling it would have serious usability consequences. GEN000000-LNX00600

```
class pam::cracklib {
```

Enforce password guessability guidelines using the `pam_cracklib` module. This module first tries to look the password up in a dictionary using `cracklib`, then applies strength checks as directed. auto: IAIA-1 auto: GEN000790

```
        augeas { "system_auth_cracklib":
            context => "/files/etc/pam.d/system-auth",
            changes => [
                "rm *[type='password'][module='pam_cracklib.so']",
                "ins 100 before *[type='password' and module!='pam_centrifydc.so'][1]",
                "set 100/type password",
                "set 100/control requisite",
                "set 100/module pam_cracklib.so",
```

Require a minimum password length of 14 characters. auto: IAIA-1 auto: GEN000580

```
                "set 100/argument[1] minlen=14",
```

Require passwords to contain at least one uppercase letter. auto: IAIA-1 auto: GEN000600

```
                "set 100/argument[2] ucredit=-1",
```

Require passwords to contain at least one lowercase letter. auto: IAIA-1 auto: GEN000610

```
                "set 100/argument[3] lcredit=-1",
```

Require passwords to contain at least one digit. auto: IAIA-1 auto: GEN000620

```
                "set 100/argument[4] dcredit=-1",
```

Require passwords to contain at least one other (special) character. auto: IAIA-1 auto: GEN000640

```
                "set 100/argument[5] ocredit=-1",
```

Prevent users from using parts of their usernames in their passwords.
(This and a few other things were GEN000660 in the 2006 UNIX STIG.)

```
                "set 100/argument[6] reject_username",
```

Prohibit the repetition of a single character in a password more than three times in a row. auto: IAIA-1 auto: GEN000680

```
                "set 100/argument[7] maxrepeat=3",
```

Let the user have three attempts at entering a strong password.

```
                "set 100/argument[8] retry=3",
```

Require that at least four characters be changed between the old and new passwords. auto: IAIA-1 auto: GEN000750

(When changing this setting, see the man page for `pam_cracklib`: the exact semantics of the difok parameter are slightly different from the semantics of the STIG requirement.)

```
                "set 100/argument[9] difok=4",
                ],
        }
    }
```

### 11.74.1 Set login failure delay

```
class pam::faildelay($seconds) {
```

The delay argument is in microseconds, so we convert.

```
        $microseconds = $seconds * 1000000

        augeas { "pam_faildelay":
            context => "/files/etc/pam.d/system-auth",
            changes => [
                "rm *[type='auth'][module='pam_faildelay.so']",
                "insert 999 before *[type='auth' and module!='pam_centrifydc.so'][1]",
                "set 999/type auth",
                "set 999/control required",
                "set 999/module pam_faildelay.so",
                "set 999/argument delay=$microseconds",
            ],
        }
    }
```

### 11.74.2 pam_limits

Make sure that `pam_limits.so` is called by the PAM configuration.

```
    class pam::limits {
        augeas {
            "pam_limits_insert":
                context => "/files/etc/pam.d/system-auth",
                onlyif => "match *[type='session' and \
                                   module='pam_limits.so'] \
                           size == 0",
                changes => [
                    "insert 999 before *[type='session' and module!='pam_centrifydc.so'][1]",
                    "set 999/type session",
                    "set 999/control required",
                    "set 999/module pam_limits.so",
                ];
            "pam_limits_require":
                require => Augeas["pam_limits_insert"],
                context => "/files/etc/pam.d/system-auth",
                changes => "set *[\
                        type='session' and \
                        module='pam_limits.so']/control \
                    required";
        }
    }
```

### 11.74.3 Limit maximum logins

Configure the system to limit the maximum number of logins. auto: ECLO-1

Note that each terminal window opened by a user may consume a login, so if you have more than `$limit` terminal windows open, and then you go to another host, and try to `ssh` to your workstation, you could be denied.

```
    class pam::max_logins($limit=10) {
```

This is done by means of `pam_limits.so`. Make sure it's in place.

```
        include pam::limits
```

§11.74.2

Now—pam_limits.so gets its list of limits from a configuration file. Make sure that file says that everyone has a maxlogins of 10.

```
    augeas {
        "limits_insert_maxlogins":
            context => "/files/etc/security/limits.conf",
            onlyif => "match *[.='*' and item='maxlogins']\
                             size == 0",
            changes => [
                "insert domain after *[last()]",
                "set domain[last()] '*'",
                "set domain[last()]/type hard",
                "set domain[last()]/item maxlogins",
                "set domain[last()]/value ${limit}",
            ];
        "limits_set_maxlogins":
            require => Augeas["limits_insert_maxlogins"],
            context => "/files/etc/security/limits.conf",
            changes => [
                "set domain[.='*' and item='maxlogins']/type hard",
                "set domain[.='*' and item='maxlogins']/value ${limit}",
            ];
    }
}
class pam::pwhistory {
```

Use the pam_pwhistory module to make sure passwords are not reused within the last ten changes. First, make sure there is a line in the right place calling pam_pwhistory:

```
    augeas { "system_auth_pwhistory":
        require => Augeas["system_auth_cracklib"],
        context => "/files/etc/pam.d/system-auth",
        changes => [
            "rm *[type='password'][module='pam_pwhistory.so']",
            "ins 100 after *[type='password']\
[module='pam_cracklib.so' or module='pam_centrifydc.so'][last()]",
            "set 100/type password",
            "set 100/control requisite",
            "set 100/module pam_pwhistory.so",
```

Remember the last ten passwords and prohibit their reuse. auto: IAIA-1

```
            "set 100/argument[1] remember=10",
```

auto: GEN000800

Do this even for root.

```
            "set 100/argument[2] enforce_for_root",
```

Don't prompt for another password: use the one from the module above this one.

```
            "set 100/argument[3] use_authtok",
            ],
    }
}
```

### 11.74.4 Disable rhosts in PAM

```
class pam::rhosts {
```

Make sure the .rhosts file is not supported in PAM.

auto: ECCD-1

auto: GEN002100

```
       augeas { "system_auth_no_rhosts":
           context => "/files/etc/pam.d/system-auth",
           changes => "rm *[module='pam_rhosts.so']",
       }
   }
```

### 11.74.5 securetty

Install the pam_securetty module which prevents root from logging in from a tty not explicitly considered secure. See also §11.84.2.

```
   class pam::securetty {
       augeas { "system_auth_securetty":
           context => "/files/etc/pam.d/system-auth",
           changes => [
               "rm *[type='auth'][module='pam_securetty.so']",
```

The pam::faildelay class (§11.74.1 inserts an auth module at the beginning of the list, and so does this one. Without loss of generality, we will put this one second, so they don't both always think the file needs to be edited.

```
               "ins 100 before *[type='auth' and module!='pam_centrifydc.so'][2]",
               "set 100/type auth",
               "set 100/control required",
               "set 100/module pam_securetty.so",
           ]
       }
   }
   class pam::tally2 {
```

Lock users out after three bad login attempts.

auto: ECLO-1

auto: GEN000460

We use the pam_tally2 module for this. It's noteworthy that due to where we put this module in the stack, if smartcard login is enabled and the user presents a valid smartcard and PIN, she is logged in regardless of tally count. The reason for this is that the pam_tally2 module needs to know a username, but in the smartcard case, the pam_pkcs11 module is finding that username out—and if it succeeds, the rest of the stack is bypassed, including pam_tally2. If pam_tally2 were put first, the user would have to enter a username before being prompted for a PIN. In terms of total system risk, the requirement to lock out users after three bad attempts is made in the context of passwords, and this policy works in the context of passwords; in the context of smartcards, the card itself will lock after three bad PIN attempts. Either of these taken alone meets the security requirement; there should not be many hosts accepting both passwords and CACs for authentication of normal users.

```
        augeas { "system_auth_tally2":
            context => "/files/etc/pam.d/system-auth",
            changes => [
                "rm *[module='pam_tally2.so'][type='auth']",
                "ins 100 before *[module='pam_deny.so' and type='auth']",
                "set 100/type auth",
                "set 100/control required",
                "set 100/module pam_tally2.so",
                "set 100/argument deny=3",
                "set 100/argument[2] audit",
                ],
        }
    }
```

## 11.75 Passwords

Implement guidelines regarding passwords.

### 11.75.1 Admin guidance about passwords

The 2006 UNIX STIG required these things: (GEN000720) Change the root password at least every 90 days. (GEN000840) Don't give the root password to anyone besides security and administrative users requiring access. Such users must be listed under §3.4. (GEN000860) Change the root password whenever anyone who has it is reassigned.

Change passwords for non-interactive or automated accounts at least once a year, and whenever anyone who has one is reassigned. admins do GEN000740

### 11.75.2 Remove passwords from gshadow

```
class passwords::no_gshadow {
```

We require a custom lens.

```
    include augeas
```

§11.13

Disable group passwords. auto: ECLP-1 auto: GEN000000-LNX001476

Although gshadow(5) says that a password only needs to start with a single exclamation point to be invalid, the check listed for this requirement only matches double exclamation points. So that the check will succeed, we set everything to double exclamation points.

```
    case $::osfamily {
        RedHat: {
            augeas { 'disable_gshadow_passwords':
                context => '/files/etc/gshadow',
                changes => [
                    'set */password "!!"',
                ],
            }
        }
        default: { unimplemented() }
    }
}
```

### 11.75.3 Guard hashed passwords

Make sure that password hashes are not stored in the /etc/passwd or /etc/group files, which are readable to everyone: if everyone can read a hashed password, someone can take it somewhere else and figure out the password by brute computational force.

```
class passwords::only_shadow {
```

Make sure the passwd file does not contain password hashes. auto: ECLP-1

(A side effect of this command is to warn if anyone has an empty password in /etc/passwd.) auto: GEN001470

```
    exec { "passwd_no_hashes":
        command => "/bin/grep -v '^[^:]\\+:x:' /etc/passwd",
        onlyif  => "/bin/grep -v '^[^:]\\+:x:' /etc/passwd",
        logoutput => true,
        loglevel => err,
    }
```

Make sure the group file does not contain password hashes. auto: ECLP-1

(A side effect of this command is to warn if any group has an empty password in /etc/group.) auto: GEN001475

```
    exec { "group_no_hashes":
        command => "/bin/grep -v '^[^:]\\+:x:' /etc/group",
        onlyif  => "/bin/grep -v '^[^:]\\+:x:' /etc/group",
        logoutput => true,
        loglevel => err,
    }
}
```

### 11.75.4 STIG-required password configuration

Implement guidelines regarding password length, strength, and age, and prevent password guessing.

```
class passwords::stig {
```

The way to do these things properly varies by platform.

```
    case $osfamily {
        'RedHat': { include passwords::stig::redhat }
        'Darwin': { include passwords::stig::darwin }
        default: { unimplemented() }
    }
}
```

### Passwords on Macs

```
class passwords::stig::darwin {
```

Prohibit the use of any of the last fifteen passwords as the next password on Macs.

auto: IAIA-1
auto: GEN000800 M6

```
    global_pwpolicy { 'usingHistory': value => 15 }
```

Set a maximum password age on Macs.

86400 minutes is 60 days.

auto: IAIA-1
auto: OSX00020 M6

```
    global_pwpolicy { 'maxMinutesUntilChangePassword':
        value => 86399,
    }
```

Set a minimum password length for Macs.

auto: IAIA-1
auto: OSX00030 M6
auto: OSX8-00-00590

```
    global_pwpolicy { 'minChars': value => 15 }
```

Require alphabetic characters in passwords on Macs.

auto: IAIA-1
auto: OSX00036 M6

```
    global_pwpolicy { 'requiresAlpha': value => true }
```

Require symbols in passwords on Macs.

auto: IAIA-1
auto: OSX00038 M6

```
    global_pwpolicy { 'requiresSymbol': value => true }
```

Prohibit names from being used as passwords on Macs.

auto: IAIA-1
auto: OSX00040 M6

```
    global_pwpolicy { 'passwordCannotBeName': value => true }
```

Unlock users after 15 minutes when they have locked themselves out with bad password attempts.

auto: OSX8-00-001325

Note that this contravenes the earlier Snow Leopard requirement Mac OS X STIG PDI OSX00045 M6.

```
    global_pwpolicy { 'minutesUntilFailedLoginReset': value => 15 }
```

Set the maximum number of failed login attempts on the Mac.

auto: ECLO-1
auto: OSX00050 M6

```
    global_pwpolicy { 'maxFailedLoginAttempts': value => 3 }
```

Disable the password hint field.

auto: OSX8-00-00630

§11.61.2

```
    mcx::set { 'com.apple.loginwindow:RetriesUntilHint':
        value => 0,
    }
}
```

### Passwords under Red Hattish Linuxen

```
class passwords::stig::redhat {
```

We need the augeas class because it teaches Augeas the format of the `login.defs` file.

```
    include pam::tally2
    include pam::cracklib
    include pam::pwhistory
    require augeas
    augeas {
```

§11.74.5
§11.74
§11.74.3

Enforce minimum and maximum password ages.

```
        "passwords_stig_login_defs":
            context => "/files/etc/login.defs",
            changes => [
```

Don't let users change passwords more than once a day.

auto: ECSC-1
auto: GEN000540

```
                "set PASS_MIN_DAYS 1",
```

Require users to change their passwords at least every 60 days.

auto: IAIA-1
auto: GEN000700

```
                "set PASS_MAX_DAYS 60",
```

Enforce the correctness of the entire password, not just the first eight characters of it.

auto: IAIA-1
auto: GEN000585

The man page says that the PASS_MIN_LEN and PASS_MAX_LEN in `/etc/login.defs` are ignored when MD5 passwords are enabled—meaning that none of the password is thrown away when hashing or applying length and strength rules. The operative minimum password length is specified above in section configuring cracklib; for any decent hashing function there is no maximum length, because it all gets hashed to the same length.

Use a FIPS 140-2 approved algorithm for hashing account passwords.

auto: DCNR-1
auto: IAIA-1
auto: GEN000590
auto: GEN000595

The man page further says that the MD5_CRYPT_ENAB variable is superseded by ENCRYPT_METHOD. That's good, because MD5 is broken and SHA1 is almost. The discussion on this PDI requires specifically something in the SHA-2 family of algorithms; we'll use the SHA-256 variant.

Red Hat Enterprise Linux 6 hashes passwords using only FIPS-approved hashing algorithms, performed by approved cryptographic modules running in FIPS-compliant mode.

RHEL6:
GEN000588

According to `https://bugzilla.redhat.com/show_bug.cgi?id=504949#c37` and a check of the dependencies of the glibc RPM package in RHEL6, glibc's libcrypt, used by pam_unix to hash passwords, uses NSS for cryptographic hashing. See 11.33 for details on FIPS accreditation status of NSS. RHEL5 may or may not be compliant with this requirement.

```
                "set ENCRYPT_METHOD SHA256",
            ];
```

Disable accounts when passwords expire.

auto: IAAC-1
auto: GEN000760

The requirement is after 35 days of inactivity, but I can't find anywhere where that this can be configured other than as an interval after password expiration.

```
        "expire_on_password_expire":
            context => "/files/etc/default/useradd",
            changes => "set INACTIVE 0";
    }
```

Log an error if any user is known to have an empty password.

auto: IAIA-1
auto: GEN000560

This will only detect empty passwords for users whose passwords are stored locally.

```
    exec { "no_empty_passwords":
        path => [’/bin’],
        command =>
            "echo ---- USERS WITH EMPTY PASSWORDS ----; \
             grep ’^[^:]\\+::’ /etc/shadow",
        onlyif  => "grep ’^[^:]\\+::’ /etc/shadow",
        loglevel => err,
        logoutput => true,
    }
    include passwords::only_shadow                                      §11.75.3
    include passwords::no_gshadow                                       §11.75.2
}
```

## 11.76 PKI (Public Key Infrastructure)

Configure PKI-related parts of the system. These have to do with certification authority (CA) certificates, certificate revocation lists (CRLs) and the like.

```
class pki {
    file { ’/etc/pki’:
        ensure => directory,
        owner => root, group => 0, mode => 0644,
    }
}
```

### 11.76.1 CA certificates

Install and maintain CA certificates in various places.

#### HPC Kerberos pkinit

Install CA certs into the /etc/pki directory, where they will be used by the pkinit utility from the HPCMP Kerberos distribution.

pkinit wants the root certificates and the CA certificates in different directories, so we put the root certificates in a root subdirectory beside the CA certificates, in /etc/pki/dod.

```
class pki::ca_certs::pkinit {
    include pki                                                         §11.76
    file { "/etc/pki/pkinit":
        ensure => directory,
        owner => root, group => 0, mode => 0644,
        source => "puppet:///modules/pki/pkinit",
        recurse => true,
```

We are copying files in a subdirectory—increase recurselimit.

```
        recurselimit => 2,
        ignore => ".svn",
        purge => true,
    }
}
```

### Citrix Receiver ICA clients

Install CA certs into the proper directory where they can be used by the Citrix Receiver ICA client.

It appears that the ICA client only needs the root certificate.

```
class pki::ca_certs::citrix_receiver {
    define install($cacerts) {
        file { "$cacerts/$name":
            owner => root, group => 0, mode => 0444,
            source => "puppet:///modules/pki/all-ca-certs/$name",
        }
    }
    case $::osfamily {
        'RedHat': {
            install { 'DoD-Root2-Root.crt':
                cacerts => '/opt/Citrix/ICAClient/keystore/cacerts',
            }
        }
        default: {
            notify { "unimplemented on $::osfamily": }
        }
    }
}
```

### libpurple (Pidgin)

Install CA certs into the /usr/share/purple/ca-certs directory, where they will be used by instant messaging clients that use the `libpurple` library.

```
class pki::ca_certs::libpurple {
    # This method seems janky.
    define install() {
        $cacerts = $::osfamily ? {
            'RedHat' => '/usr/share/purple/ca-certs',
            default  => unimplemented(),
        }
        file { "$cacerts/$name":
            owner => root, group => 0, mode => 0444,
            source => "puppet:///modules/pki/all-ca-certs/$name",
            require => Package['libpurple'],
        }
    }
    define remove() {
        $cacerts = $::osfamily ? {
            'RedHat' => '/usr/share/purple/ca-certs',
            default  => unimplemented(),
        }
```

```
            file { "$cacerts/$name":
                ensure => absent,
                require => Package['libpurple'],
            }
        }
        install { [
                'DoD-email-Root2-CA21.crt',
                'DoD-email-Root2-CA22.crt',
                'DoD-email-Root2-CA23.crt',
                'DoD-email-Root2-CA24.crt',
                'DoD-email-Root2-CA25.crt',
                'DoD-email-Root2-CA26.crt',
                'DoD-email-Root2-CA27.crt',
                'DoD-email-Root2-CA28.crt',
                'DoD-email-Root2-CA29.crt',
                'DoD-email-Root2-CA30.crt',
                'DoD-Root2-CA21.crt',
                'DoD-Root2-CA22.crt',
                'DoD-Root2-CA23.crt',
                'DoD-Root2-CA24.crt',
                'DoD-Root2-CA25.crt',
                'DoD-Root2-CA26.crt',
                'DoD-Root2-CA27.crt',
                'DoD-Root2-CA28.crt',
                'DoD-Root2-CA29.crt',
                'DoD-Root2-CA30.crt',
                'DoD-Root2-Root.crt',
                'ECA-IdenTrust3.crt',
                'ECA-ORC-HW4.crt',
                'ECA-ORC-SW4.crt',
                'ECA-Root2.crt',
                'ECA-Root.crt',
                'ECA-Verisign-G3.crt',
            ]: }
        remove { [
                'DoD-Class3-Root.crt',
                'DoD-email-Root2-CA15.crt',
                'DoD-email-Root2-CA16.crt',
                'DoD-email-Root2-CA17.crt',
                'DoD-email-Root2-CA18.crt',
                'DoD-email-Root2-CA19.crt',
                'DoD-email-Root2-CA20.crt',
                'DoD-Root2-CA15.crt',
                'DoD-Root2-CA16.crt',
                'DoD-Root2-CA17.crt',
                'DoD-Root2-CA18.crt',
                'DoD-Root2-CA19.crt',
                'DoD-Root2-CA20.crt',
                'ECA-IdenTrust2.crt',
                'ECA-Verisign-G2.crt',
                'ECA-ORC-HW3.crt',
                'ECA-ORC-SW3.crt',
            ]: }
    }
```

### /etc/pki/pam_pkcs11

Install selected CA certs into an NSS database just for pam_pkcs11. This is because we only want to trust the DoD identity CAs for local CAC logins, not (for example) the ECAs.

```
class pki::ca_certs::pam_pkcs11 {
    pki::nss::db { "/etc/pki/pam_pkcs11":                              §11.76.5
        owner => root, group => 0, mode => 0644,
    }
    Nss_cert {
        dbdir => "/etc/pki/pam_pkcs11",
        source => "puppet:///modules/pki/all-ca-certs/",
        require => Pki::Nss::Db["/etc/pki/pam_pkcs11"],
    }
    nss_cert {
        "DoD-Root2-CA19":;
        "DoD-Root2-CA20":;
        "DoD-Root2-CA21":;
        "DoD-Root2-CA22":;
        "DoD-Root2-CA23":;
        "DoD-Root2-CA24":;
        "DoD-Root2-CA25":;
        "DoD-Root2-CA26":;
        "DoD-Root2-CA27":;
        "DoD-Root2-CA28":;
        "DoD-Root2-CA29":;
        "DoD-Root2-CA30":;
        "DoD-Root2-CA31":;
        "DoD-Root2-CA32":;
        "DoD-Root2-Root":;
    }
    nss_cert {
        "DoD-Root2-CA11": ensure => absent;
        "DoD-Root2-CA12": ensure => absent;
        "DoD-Root2-CA13": ensure => absent;
        "DoD-Root2-CA14": ensure => absent;
        "DoD-Root2-CA15": ensure => absent;
        "DoD-Root2-CA16": ensure => absent;
        "DoD-Root2-CA17": ensure => absent;
        "DoD-Root2-CA18": ensure => absent;
    }
}
```

### Systemwide NSS (/etc/pki/nssdb)

Install CA certs into the systemwide NSS database.

```
class pki::ca_certs::system_nss {
    $db = "/etc/pki/nssdb"
    pki::nss::db { $db:                                                §11.76.5
```

```
            owner => root, group => 0, mode => 0644,
        }
        pki::nss::dod_roots { $db: }                          §11.76.10
        pki::nss::dod_cas { $db: }                            §11.76.6
        pki::nss::dod_email_cas { $db: }                      §11.76.8
        pki::nss::eca_roots { $db: }                          §11.76.12
        pki::nss::eca_cas { $db: }                            §11.76.11
    }
```

### Systemwide NSS (/etc/pki/nssdb) using SQLite

Install CA certs into the systemwide Berkeley DB-based NSS database.

```
    class pki::ca_certs::system_nss_berkeleydb {
        $db = "/etc/pki/nssdb"
        pki::nss::db { $db:                                   §11.76.5
            owner => root, group => 0, mode => 0644,
            sqlite => false,
        }
        pki::nss::dod_roots { $db: sqlite => false }          §11.76.10
        pki::nss::dod_cas { $db: sqlite => false }            §11.76.6
        pki::nss::dod_email_cas { $db: sqlite => false }      §11.76.8
        pki::nss::eca_roots { $db: sqlite => false }          §11.76.12
        pki::nss::eca_cas { $db: sqlite => false }            §11.76.11
    }
```

### /etc/pki/tls

Trust only DoD PKI CAs. auto: WG355 A22

These CA certificates will be used by web servers. Web servers should let ECA people in as well as CAC people.

```
    class pki::ca_certs::tls {
        include pki                                           §11.76
        file { "/etc/pki/tls":
            ensure => directory,
            owner => root, group => 0, mode => 0644,
        }
        file { "/etc/pki/tls/cacerts":
            ensure => directory,
            source => "puppet:///modules/pki/tls",
            recurse => true,
```

We are copying files in a subdirectory—increase recurselimit.

```
            recurselimit => 2,
            ignore => ".svn",
            purge => true,
            owner => root, group => 0, mode => 0644,
        }
    }
```

### 11.76.2 CAC Login

On select hosts, configure the Pluggable Authentication Modules (PAM) subsystem to allow CAC login from the console using the `pam_pkcs11` module. auto: IAIA-1 auto: IATS-1 auto: GEN009120

These changes are quite similar to what the command

```
authconfig --enablesmartcard --update
```

would do.

Note that as of early 2011, RHEL cannot reliably use Alternate Logon Tokens (ALTs) because of a shortcoming in CoolKey; see `https://bugzilla.redhat.com/show_bug.cgi?id=574953`.

```
class pki::cac_login {
    augeas {
        "pam_pkcs11_sa":
            context => "/files/etc/pam.d/system-auth-ac",
            changes => [
```

Add the pam_pkcs11 module to the configuration.

```
                "ins 100 before \
                     *[module='pam_unix.so'][type='auth']",
                "set 100/type auth",
                "set 100/control '[success=done \
authinfo_unavail=ignore ignore=ignore default=die]'",
                "set 100/module pam_pkcs11.so",
            ],
            onlyif => [
                "match *[module='pam_pkcs11.so'][type='auth'] \
                 size == 0",
            ];
        "pam_pkcs11_arguments_sa":
            require => Augeas["pam_pkcs11_sa"],
            context => "/files/etc/pam.d/system-auth-ac/\
*[module='pam_pkcs11.so'][type='auth']",
            changes => [
                'rm argument',
            ];
```

Just before it, skip pam_pkcs11 for all but a few services trying to authenticate the user.

```
        "pam_ignore_pkcs11_sa":
            require => Augeas["pam_pkcs11_sa"],
            context => "/files/etc/pam.d/system-auth-ac",
            changes => [
                "ins 99 before \
                    *[module='pam_pkcs11.so'][type='auth']",
                "set 99/type auth",
                "set 99/control '[success=1 default=ignore]'",
                "set 99/module pam_succeed_if.so",
            ],
            onlyif => [
                "match *[module='pam_succeed_if.so'][type='auth'] \
                 size == 0",
            ];

        "pam_ignore_pkcs11_arguments_sa":
            require => Augeas["pam_ignore_pkcs11_sa"],
            context => "/files/etc/pam.d/system-auth-ac/\
*[module='pam_succeed_if.so'][type='auth']\
[control='[success=1 default=ignore]']",
            changes => [
                "rm argument",
                "set argument[1] service",
                "set argument[2] notin",
```

authconfig does not enable smartcards for use with sudo, but this policy does, by putting sudo in the following list of services.

```
                "set argument[3] \
login:sudo:gdm:xdm:kdm:xscreensaver:\
gnome-screensaver:kscreensaver",
                "set argument[4] quiet",
                "set argument[5] use_uid",
            ];
    }
```

Make sure the CA certs are in place for pam_pkcs11 to use.

```
    include pki::ca_certs::pam_pkcs11                          §11.76.1
```

Configure pam_pkcs11 to look for certificate common names in the GECOS field. The pam_pkcs11 configuration file format is complicated enough that I couldn't write an Augeas lens for it within a couple of hours, so we just copy the file over.

```
    file { "/etc/pam_pkcs11/pam_pkcs11.conf":
        owner => root, group => 0, mode => 0644,
        source => "puppet:///modules/pki/pam_pkcs11.conf",
    }
}
```

### 11.76.3 NSS and FIPS

Each NSS database has a FIPS-compliance switch, which can be on or off. The most visible effect of FIPS compliance is that a passphrase is required before any cryptographical work can be done using the contents of the NSS database. Some programs (e.g., Apache with `mod_nss`) have their own FIPS compliance setting, which may use the database in FIPS mode even if its FIPS setting is off.

In order for the FIPS mode to work, a passphrase must be set. The above defined resource type does not set a passphrase, so any freshly made database will be unusable in FIPS mode.

To make it usable:

1. Turn off FIPS mode if necessary: `modutil -fips false -dbdir` *`directory`*.
2. Set a passphrase on it: `modutil -changepw "NSS Certificate DB" -dbdir` *`directory`*.
3. Turn on FIPS mode if necessary: `modutil -fips true -dbdir` *`directory`*.
4. You will need to type that passphrase every time you start the server.
5. Do not write the passphrase in a file. This would enable services that need to use NSS for encryption, like Apache with `mod_nss`, to do so without prompting for the passphrase. It would also enable a remote attacker who compromised such a service to get at the private keys immediately, without needing to brute-force the passphrase.
6. Such a file has the following format: Each line of the file should look like *`module`*`:`*`password`*. The modules of interest are "internal", "NSS Certificate DB" and "NSS FIPS 140-2 Certificate DB".

You should change the passphrase at least once every year, because it's analogous to a non-interactive account password.

admins do
GEN000740

### 11.76.4 Let Australian DoD certs in

This defined resource type will install DoD CCEB interoperability root CA certificates into the named database. These offer a trust path to some certificates issued outside the DoD. See `http://iase.disa.mil/pki-pke/interoperability/` for more details, and for rules under which you must operate when trusting this CA from a DoD server.

It also will install Australian Defence Organisation (sp?) certs.

```
define pki::nss::australia($pwfile=’’, $sqlite=true) {
    Nss_cert {
        source => "puppet:///modules/pki/all-ca-certs/",
        pwfile => $pwfile,
        sqlite => $sqlite,
        require => Pki::Nss::Db[$name],
    }
    nss_cert { "${name}:DoD-CCEB-Interop-Root-CA1":
        trustargs => ’CT,C,C’,
    }
    nss_cert {
        "${name}:Bridge-DoDCCEBIRCA1-ADOCA03": ;
        "${name}:ADO-CA014": ;
        "${name}:ADO-CA016": ;
    }
}
```

### Maintain CRLs for NSS database

Keep certificate revocation lists (CRLs) up to date. auto: WG145 A22

```
define pki::nss::crl($dbdir, $pwfile, $http_proxy=’’, $sqlite=true) {
    file { "/usr/sbin/refresh_crls_nss.py":
        owner => root, group => 0, mode => 0755,
        source => "puppet:///modules/pki/\
get_crl/refresh_crls_nss.py",
    }

    $berkeley_switch = $sqlite ? {
        true  => ’’,
        false => ’-B’,
    }
    file { "/etc/cron.daily/refresh_nss_crls_${name}":
        owner => root, group => 0, mode => 0700,
        content => "#!/bin/sh
export http_proxy=${http_proxy}

/usr/sbin/refresh_crls_nss.py \
        ${berkeley_switch} ${dbdir} ${pwfile}
",
    }
}
```

## 11.76.5 NSS databases

Some subsystems store their CA certificates in an NSS database rather than a directory. Here is how to ensure that such an NSS database exists and is ready to have certificates imported into it.

The `pwfile` parameter dictates whether to create a password file along with the database. For specific services this may be necessary; for managing the systemwide NSS database it should be false.

```
define pki::nss::db($owner, $group, $mode, $sqlite=true, $pwfile=false) {
    $dbdir = $sqlite ? {
        true  => "sql:${name}",
        false => $name,
    }
    $creates = $sqlite ? {
        true  => "${name}/cert9.db",
        false => "${name}/cert8.db",
    }
```

Every NSS database is a directory containing several `.db` files, and is referred to using the name of the directory. First, make sure the directory exists.

```
    file { "$name":
        ensure => directory,
        owner => $owner, group => $group, mode => $mode,
        recurse => true,
        recurselimit => 1,
    }
```

Then, if there is no certificate database file in the directory, create it.

```
    case $pwfile {
        true: {
```

*certutil* needs the password file, and other automated NSS management by Puppet needs the password file; but on production servers the password should be saved somewhere and the password file should be deleted, so that using the NSS database set up here will require the passphrase to be entered.

```
            pki::nss::pwfile { "${name}":                          §11.76.12
                require => File["${name}"],
            } ->
            exec { "create_nssdb_${name}_with_pwfile":
                command => "/usr/bin/certutil \
                    -N -d ${dbdir} -f ${name}/pwfile",
                creates => $creates,
            } ˜> # squiggle not dash
            exec { "enable_fips_${name}_with_pwfile":
                refreshonly => true,
                command => "/usr/bin/modutil \
                    -dbdir ${dbdir} \
                    -fips true",
            }
        }
        default: {
```

We use `modutil` to create the database. `certutil` would work too, but it needs a passphrase.

```
            exec { "create_nssdb_${name}":
                command => "/usr/bin/modutil \
                    -create \
                    -dbdir ${dbdir} \
                    </dev/null >&/dev/null",
```

The redirections get rid of `modutil`'s warning about modifying the database while "the browser is running." In a systemwide context this doesn't matter.

```
                require => File["$name"],
                creates => $creates,
            }
```

We don't turn on FIPS mode because that would require a password before the database could be used, and we didn't set up a password file.

```
        }
    }
}
```

In other PKI subsections the above define is used to automate these checks.

### 11.76.6 Install DoD CA certs

This defined resource type will install DoD CA certificates (not email CAs, not ECAs) into the named NSS database.

```
define pki::nss::dod_cas($pwfile='', $sqlite=true) {
    Nss_cert {
        source => "puppet:///modules/pki/all-ca-certs/",
        pwfile => $pwfile,
        sqlite => $sqlite,
        require => [
            Pki::Nss::Db[$name],
            Nss_cert["${name}:DoD-Root2-Root"],
        ],
    }

    nss_cert {
        "${name}:DoD-Root2-CA21":;
        "${name}:DoD-Root2-CA22":;
        "${name}:DoD-Root2-CA23":;
        "${name}:DoD-Root2-CA24":;
        "${name}:DoD-Root2-CA25":;
        "${name}:DoD-Root2-CA26":;
        "${name}:DoD-Root2-CA27":;
        "${name}:DoD-Root2-CA28":;
        "${name}:DoD-Root2-CA29":;
        "${name}:DoD-Root2-CA30":;
        "${name}:DoD-Root2-CA31":;
        "${name}:DoD-Root2-CA32":;
    }
```

Remove expired CA certs.

```
    nss_cert {
        "${name}:DoD-Root2-CA11": ensure => absent;
        "${name}:DoD-Root2-CA12": ensure => absent;
        "${name}:DoD-Root2-CA13": ensure => absent;
        "${name}:DoD-Root2-CA14": ensure => absent;
        "${name}:DoD-Root2-CA15": ensure => absent;
        "${name}:DoD-Root2-CA16": ensure => absent;
        "${name}:DoD-Root2-CA17": ensure => absent;
        "${name}:DoD-Root2-CA18": ensure => absent;
        "${name}:DoD-Root2-CA19": ensure => absent;
        "${name}:DoD-Root2-CA20": ensure => absent;
    }
}
```

### 11.76.7 Install DoD CCEB interoperability root cert(s)

This defined resource type will install DoD CCEB interoperability root CA certificates into the named database. These offer a trust path to some certificates issued outside the DoD. See http://iase.disa.mil/pki-pke/interoperability/ for more details, and for rules under which you must operate when trusting this CA from a DoD server.

```
define pki::nss::dod_cceb_interop($pwfile='', $sqlite=true) {
    nss_cert { "${name}:DoD-CCEB-Interop-Root-CA1":
        source => "puppet:///modules/pki/all-ca-certs/",
        trustargs => 'CT,C,C',
        pwfile => $pwfile,
        require => Pki::Nss::Db[$name],
        sqlite => $sqlite,
    }
}
```

### 11.76.8 Install DoD email CA certs

This defined resource type will install DoD email CA certificates (not identity CAs, not ECAs) into the named NSS database.

```
define pki::nss::dod_email_cas($pwfile=’’, $sqlite=true) {
    Nss_cert {
        source => "puppet:///modules/pki/all-ca-certs/",
        pwfile => $pwfile,
        sqlite => $sqlite,
        require => [
            Pki::Nss::Db[$name],
            Nss_cert["${name}:DoD-Root2-Root"],
        ],
    }

    nss_cert {
        "${name}:DoD-email-Root2-CA21":;
        "${name}:DoD-email-Root2-CA22":;
        "${name}:DoD-email-Root2-CA23":;
        "${name}:DoD-email-Root2-CA24":;
        "${name}:DoD-email-Root2-CA25":;
        "${name}:DoD-email-Root2-CA26":;
        "${name}:DoD-email-Root2-CA27":;
        "${name}:DoD-email-Root2-CA28":;
        "${name}:DoD-email-Root2-CA29":;
        "${name}:DoD-email-Root2-CA30":;
    }
```

Remove expired CA certs.

```
    nss_cert {
        "${name}:DoD-email-Root2-CA11": ensure => absent;
        "${name}:DoD-email-Root2-CA12": ensure => absent;
        "${name}:DoD-email-Root2-CA13": ensure => absent;
        "${name}:DoD-email-Root2-CA14": ensure => absent;
        "${name}:DoD-email-Root2-CA15": ensure => absent;
        "${name}:DoD-email-Root2-CA16": ensure => absent;
        "${name}:DoD-email-Root2-CA17": ensure => absent;
        "${name}:DoD-email-Root2-CA18": ensure => absent;
        "${name}:DoD-email-Root2-CA19": ensure => absent;
        "${name}:DoD-email-Root2-CA20": ensure => absent;
}
}
```

### 11.76.9 Install DoD interoperability root cert(s)

This defined resource type will install DoD interoperability root CA certificates into the named database. These offer a trust path to certificates issued outside the DoD. See `http://iase.disa.mil/pki-pke/interoperability/` for more details, and for rules under which you must operate when trusting this CA from a DoD server.

```
define pki::nss::dod_interop_roots($pwfile=’’, $sqlite=true) {
    nss_cert { "${name}:DoD-Interop-Root-CA1":
        source => "puppet:///modules/pki/all-ca-certs/",
        trustargs => ’CT,C,C’,
        pwfile => $pwfile,
        require => Pki::Nss::Db[$name],
        sqlite => $sqlite,
    }
}
```

### 11.76.10 Install DoD root cert(s)

This defined resource type will install DoD root CA certificates (no intermediate CAs, no ECAs) into the named database.

```
define pki::nss::dod_roots($pwfile=’’, $sqlite=true) {
    nss_cert { "${name}:DoD-Root2-Root":
        source => "puppet:///modules/pki/all-ca-certs/",
        trustargs => ’CT,C,C’,
        pwfile => $pwfile,
        require => Pki::Nss::Db[$name],
        sqlite => $sqlite,
    }
}
```

### 11.76.11 Install ECA CA cert(s)

This defined resource type will install CA certificates for External Certification Authorities (ECAs) into the named NSS database.

```
define pki::nss::eca_cas($pwfile=’’, $sqlite=true) {
    Nss_cert {
        source => "puppet:///modules/pki/all-ca-certs/",
        pwfile => $pwfile,
        sqlite => $sqlite,
        require => [
            Pki::Nss::Db[$name],
            Nss_cert["${name}:ECA-Root2"],
        ],
    }
    nss_cert {
```

CA certs issued by the ECA Root CA: None seem to exist any more.

```
        "${name}:ECA-ORC2":
            ensure => absent;
        "${name}:ECA-Identitrust1":
            ensure => absent;
```

CA certs issued by ECA Root CA 2:

```
        "${name}:ECA-Verisign-G2":
            ensure => absent;
        "${name}:ECA-IdenTrust2":
            ensure => absent;
        "${name}:ECA-ORC-HW3":
            ensure => absent;
        "${name}:ECA-ORC-SW3":
            ensure => absent;
        "${name}:ECA-ORC-HW4":;
        "${name}:ECA-ORC-SW4":;
        "${name}:ECA-IdenTrust3":;
        "${name}:ECA-IdenTrust4":;
        "${name}:ECA-Verisign-G3":;
    }
}
```

### 11.76.12 Install ECA root cert(s)

This defined resource type will install External Certification Authority (ECA) root CA certificates into the named database.

```
define pki::nss::eca_roots($pwfile=’’, $sqlite=true) {
    Nss_cert {
        source => "puppet:///modules/pki/all-ca-certs/",
        trustargs => ’CT,C,C’,
        pwfile => $pwfile,
        sqlite => $sqlite,
        require => Pki::Nss::Db[$name],
    }
    nss_cert {
        "${name}:ECA-Root":;
        "${name}:ECA-Root2":;
    }
}
```

#### Insecure NSS password files

This defined resource type generates an NSS password file in the named database directory containing a random password. It’s for use on development servers, which we want to be able to set up with less hands-on administration.

This code does not deal with changing the password every year.

```
define pki::nss::pwfile($filename=’pwfile’) {
    exec { "create ${name}/${filename}":
        command => "bash -c \"\
            PW=$(head -c 24 /dev/random | base64 -); \
            for m in internal ’NSS Certificate DB’ \
                    ’NSS FIPS 140-2 Certificate DB’; do
                echo \\\"\\\$m:\\\$PW\\\"; done > ${name}/${filename}\"",
        path => [’/bin’, ’/usr/bin’],
        creates => "${name}/${filename}",
    }
}
```

### Creating self-signed certs in an NSS database

Imitating the nss_cert custom resource type, the name of this resource is of the form dbdir:nickname. This defined resource type will create a self-signed certificate in the name of the given subject, with the given nickname, if none exists in the database. The subject should not contain double-quotes, backslashes, or other such; PKIX standards do not impose these limitations but we do here.

The noise file must be a file of length at least 2048 bytes, containing random bits. /dev/random is such a file, but could take an hour or more to cough up the required bits. /dev/urandom appears not to work. So, if you want your self-signed certificate to be generated in less than an hour, make your own file containing random bits, and provide it as the value of the noise_file parameter.

A password file called pwfile is required to be in the NSS directory being used in order for the certificate generation to work.

```
define pki::nss::self_signed(
        $subject="cn=${::fqdn}",
        $sqlite=true,
        $noise_file='/dev/random') {
    $pieces = split($name, ':')
    $dir  = $pieces[0]
    $nick = $pieces[1]
    $dbdir = $sqlite ? {
        true => "sql:${dir}",
        false => $dir,
    }
    case $noise_file {
        '/dev/random': {
            $timeout = 7200
            notify { '${name} slow cert warning':
                message => 'Generating this certificate could take hours.',
                loglevel => warning,
            }
        }
        default: {
            $timeout = 30
        }
    }
```

Under virtual machine environments without mature means to pass host entropy to guest machines (I'm looking at you, VirtualBox circa 2013), /dev/random is *glacially slow.* NSS reads 2048 bytes from the given noise file; the entropy pool on a Vagrant virtual machine using VirtualBox fills at something like 5 bits per second. That's an hour or two to generate a certificate. So if security isn't a big priority—and if we're making a self-signed certificate it's not—any file with at least 2048 bytes of stuff in it will do.

```
        exec { "create_self_signed_${nick}_in_${dbdir}":
            command => "certutil -S -d ${dbdir} \
                -x -s \"${subject}\" -n \"${nick}\" \
                -t ,, -f ${dir}/pwfile -z ${noise_file}",
            unless => "certutil -L -d ${dbdir} -n \"${nick}\"",
            timeout => $timeout,
            require => [
                Pki::Nss::Db[$dir],
                Pki::Nss::Pwfile[$dir],
                ],
            path => ['/bin', '/usr/bin'],
        }
    }
```

### 11.76.13 TLS

Maintain certificates, keys, and CRLs needed for TLS (Transport Layer Security). These are used by web servers.

```
    class pki::tls($http_proxy='') {
```

Make sure the private TLS directory is actually private.

```
        file { "/etc/pki/tls/private":
            owner => root, group => 0, mode => 0600,
            recurse => true, recurselimit => 3,
        }
```

This one has to be executable

```
        file { "/etc/pki/tls/private/.startup":
            owner => root, group => 0, mode => 0700,
        }

        include pki::ca_certs::tls                          §11.76.1
        class { 'pki::tls::crl':                            §11.76.13
            http_proxy => $http_proxy,
        }
    }
```

#### Maintain CRLs for TLS CA certificates

Keep certificate revocation lists (CRLs) up to date. auto: WG145 A22

```
    class pki::tls::crl($http_proxy='') {
```

The CRL updating script needs this.

```
        package { "python-ldap": ensure => present }

        file { "/etc/pki/tls/crls":
            ensure => directory,
            owner => root, group => 0, mode => 0644,
            recurse => true, recurselimit => 1,
        }
```

```
    file { "/usr/sbin/refresh_crls.py":
        owner => root, group => 0, mode => 0755,
        source => "puppet:///modules/pki/\
get_crl/refresh_crls.py",
    }

    file { "/etc/cron.daily/refresh_crls":
        owner => root, group => 0, mode => 0700,
        content => "#!/bin/sh\n\
export http_proxy=${http_proxy}\n\
/usr/sbin/refresh_crls.py \
  /etc/pki/tls/cacerts \
  /etc/pki/tls/crls\n",
    }
}
```

## 11.77 PolicyKit

### 11.77.1 Introduction

I took a couple hours finding the following out from the PolicyKit documentation; hopefully my summary makes it quicker for you, the reader.

PolicyKit finds answers to fine-grained permission questions needed for normal desktop operation, like, ”Can I mount this USB disk?” or ”Can I set the WiFi card to use this network?” or ”Can I make the computer go to sleep?” It does this in a secure fashion. Software authors identify things their software needs to do that admins may want to prohibit or restrict, or that malware writers may want to trick users into doing. These are defined by XML files stored (under RHEL6) in `/usr/share/polkit-1/actions`, one per application. These XML files contain defaults given by the software author regarding what the policy should be. For example, ”by default, users should be able to plug in USB disks and have them work.”

The PolicyKit local authority listens on the D-Bus for policy questions from applications. It consults files under `/etc/polkit-1`, `/var/lib/polkit-1` and `/usr/share/polkit-1/actions`. The intent is that admins put pieces of overriding policy in `/etc/polkit-1`, packagers put pieces of distro-specific overriding policy in `/var/lib/polkit-1`, and only software authors mess with what's in `/usr/share/polkit-1/actions`. Then the local authority consults these files to find the answer to whether someone's allowed to do something. Variables include who the user is (user id, group ids), whether the user is in possession of the active console session (if the user Switched User rather than logging in, there are other users in possession of inactive console sessions), and what the action is. Answers may be yes, no, you must type your password, or you must authenticate as an admin; part of the answer is how long the answer is valid for (this process, this whole session, or forever).

Since PolicyKit policy is split out into separate files, all PolicyKit policy is not centralized in this section; different sections of this policy deploy bits of

PolicyKit policy as needed. Look in the Files index for files with `polkit-1` in their pathnames to locate these.

### 11.77.2 Policy regarding PolicyKit as a whole

Make it harder for non-admins to find out what PolicyKit will let them do. The SRG does not require this, but it probably would if they had thought about it.

```
class policykit {
    file { "/etc/polkit-1":
        owner => root, group => 0, mode => 0600,
        recurse => true, recurselimit => 3,
    }
    no_ext_acl { "/etc/polkit-1": recurse => true }
}
```

### 11.77.3 Install a PolicyKit rule

This defined resource type is for system mandatory rules for PolicyKit 0.96 as used in RHEL6.

As an example, one of the things PolicyKit enables is for non-root users to change network settings, so that desktop users, who are not computer administrators by trade, can connect to wireless networks without the security risks involved in becoming root. But (see §11.67.3) as a matter of compliance we may want to get rid of that ability. You could do so like this:

```
    policykit::mandatory_rule { 'network-admin-auth':
        description => "only admins can change network",
        identity => '*',
        action => "org.freedesktop.NetworkManager.*;\
org.freedesktop.network-manager-settings.*",
    }
```

The values you provide are written directly in a PolicyKit local authority file; the syntax is written in `pklocalauthority(8)`. The default result provided by this type is `auth_admin`, because that's what security documents are most likely to require.

There is also much in `pklocalauthority(8)` about how rules combine, and which rules win. Go read it.

```
define policykit::rule(
        $description,
        $identity,
        $action,
        $result_any="auth_admin",
        $result_active="auth_admin",
        $result_inactive="auth_admin",
        $order="50",
        $rule_directory="/etc/polkit-1/\
localauthority/90-mandatory.d",
        ) {

    if     ($::osfamily == 'RedHat') and
           ($::operatingsystemrelease =~ /^6\..*/) {
```

RHEL6 uses PolicyKit.

```
        if $::policykit_installed == 'true' {
            file { "${rule_directory}/\
${order}-cmits-${name}.pkla":
                owner => root,
                group => 0,
                mode => 0600,
                content => "\
[$description]\n\
Identity=$identity\n\
Action=$action\n\
ResultAny=$result_any\n\
ResultActive=$result_active\n\
ResultInactive=$result_inactive\n",
            }
        } else {
```

If PolicyKit is not installed (e.g., on a server), the directory tree where this file belongs will not exist—and there won't be any point installing the file, either, because without PolicyKit, normal users cannot do whatever this rule is limiting. So we do nothing, with no error.

```
        }
    } else {
```

Other operating systems besides RHEL6 may not come with PolicyKit, or may come with a much different version of it. The details above don't make sense for any other OS than RHEL6, so we won't bother dealing with other OSes on a case-by-case basis here.

```
        unimplemented()
    }
}
```

## 11.78 PostgreSQL database server

Being a client-server Database Management System (DBMS), PostgreSQL is subject to the General Database STIG [3]. As with any STIG, some requirements can be automatically enforced by this policy and some are up to database administrators (DBAs), system administrators (SAs) and users to fulfill on an ongoing basis.

This class has to do with PostgreSQL servers. Policy-based PostgreSQL client configuration will be under postgresql::client; this is not yet written.

```
class postgresql($audit_data_changes = false) {

    require postgresql::initialize
    service { "postgresql":
        enable => true,
        ensure => running,
        require => Class['postgresql::initialize'],
```

Don't interrupt service when settings change. If postgresql.conf changes and the server needs to be restarted, not reloaded, that should happen during some planned downtime or something.

```
        restart => "/sbin/service postgresql reload",
    }
```

Get rid of the wide-open initially installed connection permissions (and any wide-open permissions that follow).

```
    augeas { 'remove_hba_wideopen_defaults':
        context => '/files/var/lib/pgsql/data/pg_hba.conf',
        changes => [
            'rm *[database="all"]',
        ],
        require => Exec['postgresql_initdb'],
        notify => Service['postgresql'],
    }
```

But make sure postgres can still connect to the postgres database.

```
    postgresql::allow_local { 'postgres':                §11.78.1
        database => 'postgres'
    }
```

Now apply STIG-based policies regarding the server configuration, and add users for Puppet and for admins.

```
    class { 'postgresql::stig':                          §11.78.6
        audit_data_changes => $audit_data_changes,
    }
    include postgresql::puppet_dba                       §11.78.4
    include postgresql::roles                            §11.78.5
}
```

### 11.78.1 Allow a local PostgreSQL user

This defined resource type is a shortcut to let a given user local to the DBMS server connect to a given database with the same username between the OS and database. Real people should connect this way.

```
define postgresql::allow_local($database) {
    require postgresql::initialize
    include postgresql                                   §11.78
```

This depends on the postgresql class, but since it will most likely be used from inside that class, notating such a dependency would result in a dependency cycle.

```
    augeas { "pg_hba_${name}_into_${database}":
        context => '/files/var/lib/pgsql/data/pg_hba.conf',
        changes => [
            'set 999/type      local',
            "set 999/database  '${database}'",
            "set 999/user      '${name}'",
            'set 999/method    ident',
        ],
        onlyif => "match *[user='${name}'] size < 1",
        require => Class['postgresql::initialize'],
        notify => Service['postgresql'],
    }
}
```

### 11.78.2 Allow an OS user into PostgreSQL as any of several users

This defined resource type is a shortcut to let a given OS user local to the DBMS server connect to a given database with any of several databse usernames, in order to make use of different sets of privileges while complying with the principle of least privilege. Services which may connect for several different reasons should connect this way. For example,a web server may connect to authenticate or authorize users, and web applications it serves may also connect for different reasons.

Example use:

```
postgresql::identmap { "auth":
  os_user => 'foozy',
  ensure => present,
  db_users => ['foozy', 'foozy_dba'],
}
```

The title of the resource is the database to which to grant access. The os_user is the operating system user who should be able to get in (or not). db_users specifies the users as which this operating system user should be able to access the database. As with many other resource types, there is an `ensure` parameter which defaults to present but can be set to absent. If you write `ensure => absent`, the operating system user will be denied all access to the database.

* * *

```
define postgresql::identmap(
        $ensure = 'present',
        $os_user,
        $db_users = [$os_user]) {
```

Least surprise: if someone hands an empty list for db_users, they probably mean that this os_user should not be able to get into the database at all.

```
if $db_users == [] {
    $ensure = ’absent’
}

$database = $name
```

This is part of an Augeas context, thus the /files.

```
$pgconfs = "/files/var/lib/pgsql/data"

include postgresql
```

§11.78

All the changes we make to the PostgreSQL configuration require that the configuration exists first, and cause the service to be restarted.

```
Augeas {
    require => Exec[’postgresql_initdb’],
    notify => Service[’postgresql’],
}

augeas { "pg_hba_identmap_for_${database}":
    context => "$pgconfs/pg_hba.conf",
    changes => [
```

This ident map will be the only way to get into this database, at least locally.

```
        "rm  *[type=’local’ and database=’${database}’]",
        "set 999/type      local",
        "set 999/database  ’${database}’",
```

This ident map will apply to all users trying to get into this database.

```
        "set 999/user      ’all’",
        "set 999/method    ident",
        "set 999/method/option ’map=${database}’",
    ],
}

case $ensure {
    ’present’: {
```

When we create many `postgresql::identmap::entry` resources below, each instance of the defined resource contains an Augeas resource, and none of those Augeas resources know about each other. So we cannot use the same pattern as above, where we tell Augeas to remove everything and then add what we want, because each Augeas resource would remove the changes wrought by all the others. Consequently `identmap_entry` does not remove anything from the `pg_ident.conf`, it only adds things.

So let’s remove everything which wasn’t specified in the manifest.

```
        $not_our_os_user = "os_user != ’${os_user}’"
```

This is going to look like `db_user␣!=␣’foo’␣and␣db_user␣!=␣’bar’`*.

```
        $not_any_of_our_db_users = inline_template(
            ’<%= db_users.map {|x|
                    "db_user != \’#{x}\’"
                 }.join(" and ") -%>’)
        include augeas     # non-stock lens required               §11.13
        augeas { "pg_ident_restrict_for_${database}":
            context => "$pgconfs/pg_ident.conf",
            changes => [
                "rm *[map='${database}' and \
                      os_user='${os_user}' and \
                      ${not_any_of_our_db_users}]",
            ],
        }
```

Now, we add everything which is specified.

```
            postgresql::identmap::entry { $db_users:                   §11.78.2
                os_user => $os_user,
                database => $database,
            }
        }
```

Support removing an OS user from ability to connect to a database.

```
        'absent': {
            include augeas                                             §11.13
            augeas { "pg_ident_remove_${os_user}_into_${database}":
                context => "$pgconfs/pg_ident.conf",
                changes => [
                    "rm *[map='${database}' and \
                          os_user='${os_user}']",
                ],
            }
        }
    }
}
```

### Add PostgreSQL ident map entries

This define is used by the `postgresql::identmap` define, *q.v.*

Since there's likely more than one database user in question, our strategy is to define a resource type pertaining to one database user, and pass an array of database users in as the name parameter in order to construct an array of these defined resources. Search for "puppet for loop" to find out more on this strategy.

```
define postgresql::identmap::entry($os_user, $database) {
    $db_user = $name

    include postgresql                                                 §11.78
```

Yes, this is a long name, but it must be unique across the entire manifest.

```
    augeas { "pg_ident_${os_user}_as_${db_user}_into_${database}":
        context => ’/files/var/lib/pgsql/data/pg_ident.conf’,
        changes => [
            "set 999/map     ’${database}’",
            "set 999/os_user ’${os_user}’",
            "set 999/db_user ’${db_user}’",
        ],
        onlyif => "match *[map=’${database}’ and \
                           os_user=’${os_user}’ and \
                           db_user=’${db_user}’] \
                           size < 1",
        require => Exec[’postgresql_initdb’],
        notify => Service[’postgresql’],
    }
}
```

### 11.78.3 One-time PostgreSQL initialization

```
class postgresql::initialize {
    # First, make sure PostgreSQL is installed and the database is initialized.
    package { "postgresql-server":
        ensure => present,
    }
    exec { "postgresql_initdb":
        command => ’/sbin/service postgresql initdb’,
        creates => ’/var/lib/pgsql/data/base’,
        require => Package[’postgresql-server’],
    }
}
```

### 11.78.4 Administering PostgreSQL using Puppet

Ensure that “the DBMS software installation account” (we take this to mean auto: ECLP-1 `postgres`, because while that user does not install the DBMS, it owns the files auto: DG0042 in which the DBMS data is stored) “is only used when performing software installation and upgrades or other DBMS maintenance,” and not for “DBA activities,” by creating a separate user for automatically enforcing policies inside the DBMS.

The `postgres` user must be used to create this user, of course, but that should only need to happen once.

```
class postgresql::puppet_dba {
```

Install the Ruby pg module so that `pgsql_role` and `pgsql_database` can work.

```
    package { ’ruby-pg’: ensure => present }
```

Make a `puppet_dba` OS user and group.

```
    include user::virtual                                    §11.113.2
    Group <| tag == ’puppet_dba’ |>
    User <| tag == ’puppet_dba’ |>
```

Make a corresponding puppet_dba database user.

```
    pgsql_role { 'puppet_dba':
        os_user => 'postgres',
        db_user => 'postgres',
        database => 'postgres',
        login => true,
        inherit => true,
        superuser => true,
        createdb => true,
        createrole => true,
        require => User['puppet_dba'],
    }
```

Make a database for that user to connect to.

```
    pgsql_database { 'puppet_dba':
        os_user => 'postgres',
        db_user => 'postgres',
        database => 'postgres',
        owner => 'puppet_dba',
    }
```

Allow the user to connect to the database.

```
    postgresql::allow_local { 'puppet_dba':
        database => 'puppet_dba',
    }
}
```

§11.78.1

### 11.78.5 Roles inside PostgreSQL

This section sets out the roles in a PostgreSQL database.

Administrative roles are the same across databases, because they do the same things; per-application roles are set out in per-application documents, but a pattern for them is set here.

Grant database administrative privileges to database administrators using DBMS roles.

auto: ECLP-1
auto: ECPA-1
auto: DG0116
auto: DG0117

```
class postgresql::roles {
```

Do all DBA work as the puppet_dba user.

```
    Pgsql_role {
        db_user => 'puppet_dba',
        os_user => 'puppet_dba',
        database => 'puppet_dba',
    }
```

Grant administrative privileges solely via roles.

auto: ECPA-1
auto: DG0117

"The role attributes LOGIN, SUPERUSER, CREATEDB and CREATEROLE ... are never inherited as ordinary privileges on database objects are. You must actually SET ROLE to a specific role having one of these attributes in order to make use of the attribute." [7, §20.4] So—

A database administrator fnord, to whom the dba role below has been

DBAs do ECLP-1
DBAs do DG0124

granted, must SET ROLE dba before doing any database administration. Such a user should RESET ROLE when done with the database administration.

So, now, the roles with administrative privileges:

DBA users create developer users on development database servers, and create application object owner users, application users, and per-application databases on test and production database servers.

```
  pgsql_role { 'dba':
    login => false,
    inherit => false,
    superuser => false,
    createdb => true,
    createrole => true,
  }
```

Developer users create application object owner users, application users, and per-application databases on development database servers.

Assignments

```
  pgsql_role { 'developer':
    login => false,
    inherit => false,
    superuser => false,
    createdb => true,
    createrole => true,
  }
```

Administrators must not use the postgres user to do anything with the database: each, being provided with his own database user, must use that instead.

admins do ECLP-1
admins do DG0042

```
  pgsql_role { [
      'jenninjl_dba',
      'adamsgd_dba',
      'graymx_dba',
      'shawfra_dba',
      'cookch_dba',
      'queener_dba',
      'chappell_dba',
'coulter_dba',
    ]:
    login => true,
    inherit => true,
```

Avoid granting "excessive or unauthorized" privileges to DBAs, by preventing them from being superusers in the database. "Although DBAs may assign themselves privileges," that action is logged when it happens, and privileges are reported monthly. See §11.78.6 for details.

auto: ECLP-1
auto: DG0085

```
    superuser => false,
    grant_roles => ['dba'],
  }
}
```

**Pattern for application roles and permissions**

This section should become a guide to what application-specific DBMS users should exist and what privileges they must have and must not have (mostly not). But it isn't written yet. Until it is, see §6.4 for a more general list of what an application needs to do to comply with the Database STIG. (Given, of course, that it's running against a database server managed by this Configuration Management for IT Systems Example Policy.)

### 11.78.6 STIG-required configuration for the PostgreSQL DBMS

```
class postgresql::stig($audit_data_changes = false) {
    require augeas
```

"Enable auditing on the database." Configure the database to log the messages required by the STIG, and to send those log messages out via the system log. Retention, periodic review, access restriction, and backup, then, are handled via the provisions for such requirements against the system log; see §11.55.1.

auto: ECAR-2
auto: ECRR-1
auto: ECCD-1
auto: ECTP-1
auto: ECTB-1
auto: DG0029
auto: DG0030
auto: DG0031
auto: DG0032
auto: DG0176

Because the logging implementation is not yet complete, these requirements are not yet met:

- Automated notification of suspicious activity detected in the audit trail is not implemented. DG0083
- Audit trail data is not reviewed daily or more frequently. DG0095
- An automated tool that monitors DBMS audit data and immediately reports suspicious activity is not deployed. DG0161

"Changes to security labels or markings" are not audited; PostgreSQL "does not support the use of security labels or sensitivity markings," so "this check is Not Applicable."

N/A: ECAR-3
N/A: DG0142

Log all attempts to modify data, if required by "application design requirements;" if not, only log attempts to modify the structure of the database.

auto: ECCD-1
auto: ECAR-2
auto: DG0031
auto: DG0145

For example, the PostgreSQL database used in the SBU system contains user and group information used in authorization decisions. That makes everything in the database a "security file," most likely, so all changes to data should be audited in this case. But data about flight tests would not be "security files," and so a flight test database application may not require auditing of all data changes; the server hosting such a database would only log DDL statements.

```
    $log_statement = $audit_data_changes ? {
        true    => ’mod’,
        default => ’ddl’,
    }

    augeas { "postgresql_logging":
        context => "/files/var/lib/pgsql/data/postgresql.conf",
        changes => [
            "set log_destination syslog",
            "set logging_collector off",
            "set syslog_facility LOCAL0",
            "set syslog_ident postgres",
```

Log all connection attempts, and every statement that results in a message with ‘error’ or greater urgency. This last includes “failed database object attempts,” “attempts to access objects that do not exist,” and “other activities that may produce unexpected failures.” auto: ECAR-2 auto: DG0141 auto: DG0145

```
            "set log_connections on",
            "set log_disconnections on",
            "set log_min_error_statement error",
```

Log the name of the acting user for each event. Date and time are taken care of by the system log. “Type of event” and “success or failure” are the text of the log message. auto: ECAR-2 auto: DG0145

Any serious authentication scheme we would implement would be based on Kerberos or LDAP; “blocking or blacklisting a user ID...” would be logged on the authentication server, not by PostgreSQL. N/A: DG0146

```
            "set log_line_prefix \"’%q%r %u @ db %d ’\"",
            "set log_statement ’${log_statement}’",
        ],
        require => Exec[’postgresql_initdb’],
        notify => Service[’postgresql’],
    }
```

Limit concurrent connections to the database. The vendor recommends 100 concurrent connections as a starting limit. auto: ECLO-1 auto: DG0134

```
    augeas { "postgresql_connections":
        context => "/files/var/lib/pgsql/data/postgresql.conf",
        changes => [
            "set max_connections 100",
        ],
        require => Package[’postgresql-server’],
    }
```

The postgres database account is the only default account for PostgreSQL. Upon investigation, PostgreSQL as included in RHEL “does not support changes to” this “default account name” so “this check is Not Applicable only for those accounts that cannot be altered.” N/A: IAIA-1 N/A: DG0131

In terms of real security, the postgres database user can only be used by the local postgres operating system user, which is not allowed to log in, so in order to do anything as the postgres database user, an attacker would first have to become root; in any such scenario, all bets are off anyway. See on Database

STIG PDI DG0041.

Because PostgreSQL and RHEL are open-source software, changing the name of the `postgres` user is possible, but it would require making a custom PostgreSQL package, which would unacceptably slow down and complicate security patch testing and installation. It would be entirely true to say that such a thing is “unsupported.”

Provide for “monthly... review of privilege assignments,” including DBA roles, within the PostgreSQL database by causing a report of roles and privileges to be sent to the administrators for review.

auto: ECLP-1
auto: ECPA-1
auto: DG0080
auto: DG0086
auto: DG0116
auto: DG0118

```
       file { "/etc/cron.monthly/postgresql-privileges-report":
           owner => root, group => 0, mode => 0700,
           source => "puppet:///modules/postgresql/privs-report.sh",
       }
   }
```

## 11.79 Prelinking

Prelinking makes it faster to execute programs that use shared libraries, which means nearly every program under RHEL. Prelinking must be disabled for FIPS 140-2 compliance (see 11.33.

### 11.79.1 Disabling prelinking

```
class prelink::no {
    package { ’prelink’:
        ensure => installed,
    }
```

The `/etc/sysconfig/prelink` file says that `prelink -ua` will be run the next night if `PRELINKING` is set to no. This happens by means of `/etc/cron.daily/prelink`.

But in between now and then, if a reboot happens, we’ll be running in FIPS mode without having un-prelinked the libraries. This will cause familiar and important parts of the system such as yum and ssh to break. So if and only if we’ve changed the above, we should go ahead and run `prelink -ua` now.

```
       augeas { "disable_prelinking":
           context => "/files/etc/sysconfig/prelink",
           changes => "set PRELINKING no",
           notify => Exec[’unprelink now’],
       }
       exec { ’unprelink now’:
           command => ’/usr/sbin/prelink -ua’,
           refreshonly => true,
           require => Package[’prelink’],
       }
   }
```

## 11.80 Proxy configuration

Configure the HTTP and HTTPS proxies, for all applications on the system which use them.

```
class proxy::pac($url) {
    $safe_osrelease = regsubst(
        $operatingsystemrelease,
        '[^A-Za-z0-9]', '_', 'G')

    class { "proxy::pac::${osfamily}_${safe_osrelease}":           §??
        url => $url,
    }
}
class proxy::pac::darwin_10_8_0($url) { class { 'proxy::pac::mac_networksetup': url => $url } }   §11.80
class proxy::pac::darwin_13_4_0($url) {
```

Configure DoD proxies on all active network interfaces. auto: OSX8-00-00810

```
    class { 'proxy::pac::mac_networksetup':                         §11.80
        url => $url,
    }
}
```

### Set proxy autoconfiguration URL in gconf

```
class proxy::pac::gconf($url) {
```

Set the system proxy settings for everyone mandatorily.

```
    gconf { '/system/proxy/autoconfig_url':
        config_source => 'mandatory',
        type => string,
        value => $url,
    }
    gconf { '/system/proxy/mode':
        config_source => 'mandatory',
        type => string,
        value => 'auto',
    }
}
```

### Set proxy autoconfiguration URL on Macs using networksetup

```
class proxy::pac::mac_networksetup($url) {
```

Examples of network services are `Ethernet` and `AirPort`.

```
    $networkservice = 'Ethernet'

    exec { 'set Mac autoproxyurl':
        unless => "networksetup -getautoproxyurl ${networkservice} | \
                   grep \"URL: ${url}\"",
        command => "networksetup -setautoproxyurl ${networkservice} ${url}",
    }

    exec { 'enable Mac autoproxy':
        onlyif => "networksetup -getautoproxyurl ${networkservice} | \
                   grep \"Enabled: no\"",
        command => "networksetup -setautoproxystate ${networkservice} on",
    }
}
```

oughta work

```
class proxy::pac::redhat_5_9($url) { class { 'proxy::pac::gconf': url => $url } }   §11.80
class proxy::pac::redhat_6_1($url) { class { 'proxy::pac::gconf': url => $url } }   §11.80
class proxy::pac::redhat_6_2($url) { class { 'proxy::pac::gconf': url => $url } }   §11.80
class proxy::pac::redhat_6_3($url) { class { 'proxy::pac::gconf': url => $url } }   §11.80
class proxy::pac::redhat_6_4($url) { class { 'proxy::pac::gconf': url => $url } }   §11.80
class proxy::pac::redhat_6_5($url) { class { 'proxy::pac::gconf': url => $url } }   §11.80
class proxy::pac::redhat_6_6($url) { class { 'proxy::pac::gconf': url => $url } }   §11.80
```

### 11.80.1 RHN (Red Hat Network)

The RHN client plugin to yum, somewhat confusingly, uses a different proxy setting than yum as a whole. Set that one too. Set the proxy for use in the shell and programs it starts.

```
class proxy::shell($server, $port) {
    shell::profile_d::sh_entry { 'proxy':                                §11.94.3
        content => "
export http_proxy=http://${server}:${port}
export https_proxy=http://${server}:${port}
export ftp_proxy=http://${server}:${port}
",
    }
}
```

### 11.80.2 YUM

(See 11.117 for everything else about YUM besides proxy settings.)

```
define proxy::yum($host, $port) {
    augeas { 'yum proxy':
        context => '/files/etc/yum.conf/main',
        changes => "set proxy 'http://${server}:${port}'",
    }
}
```

**None**

```
class proxy::yum::no {
    augeas { "proxy_yum_no":
        context => "/files/etc/yum.conf/main",
        changes => "rm proxy",
    }
}
```

## 11.81 Puppet

Configure Puppet itself.

### 11.81.1 Run Puppet client automatically

Arrange for the Puppet client to be run automatically.

```
class puppet::client {

    case $::osfamily {
        'RedHat': {
            case $::operatingsystemrelease {
                /^6\..*/: {
                    package { 'puppet':
                        ensure => installed,
                    }
```

If the Puppet agent is running on a host, we can assume that the Puppet package is installed, which defines the service named above. If the agent is not running on a host, that host will not be paying attention to this:

```
                    service { 'puppet':
                        enable => true,
                        ensure => running,
                    }
                }
                /^5\..*/: {
                    package { 'apscl':
                        ensure => installed,
                    }
                    service { 'apscl-puppet':
                        enable => true,
                        ensure => running,
                    }
                    file { '/usr/bin/puppet':
                        ensure => present,
                        owner => root, group => 0, mode => 0755,
                        content => "#!/bin/sh
scl enable apscl \"puppet \$*\"
",
                    }
                    file { '/usr/bin/facter':
                        ensure => present,
                        owner => root, group => 0, mode => 0755,
                        content => "#!/bin/sh
scl enable apscl \"facter \$*\"
",
                    }
                }
            }
        }
        'Darwin': {
            $service_name = 'mil.hpc.eglin.puppet'
            mac_launchd_file { $service_name:
                description => 'Puppet client daemon',
                environment => {
                    'PATH' =>    '/sbin:/usr/sbin:/bin:/usr/bin',
                    'RUBYLIB' => '/usr/lib/ruby/site_ruby/1.8',
                },
                arguments => [
                    '/usr/bin/puppet',
                    'agent',
                    '--verbose',
                    '--no-daemonize',
                ],
            } ~>
            service { $service_name:
                enable => true,
                ensure => running,
                require => Mac_launchd_file[$service_name],
            }
        }
        default: { unimplemented() }
    }
```

It may be better to run the agent with cron rather than have it hanging about and growing in size. We'll see if that becomes a problem.

Let admins run the Puppet commands with environment variables set.

```
    sudo::auditable::command_alias { 'PUPPET_BINARIES':            §11.104.3
        type => 'setenv_exec',
        commands => [
            '/usr/bin/puppet',
            '/usr/bin/facter',
            ],
    }
}
```

### 11.81.2 Development box

A host where Puppet manifests are to be developed.

```
class puppet::devel {
    include puppet::client                        §11.81.1
    include common_packages::graphviz             §11.21.1
    include common_packages::latex                §11.21.2
    package { [
        "puppet-server",
    ]:
        ensure => installed,
    }
```

Stored configs depend on Rails, which RHEL does not provide as RPMs, so we must install the gems. Passenger involves some manual stuff that may not be automatable just yet.

```
    package { [
        'rspec',
        'rspec-puppet',
    ]:
        provider => gem,
        ensure => installed,
        source => "",
    }
}
```

### 11.81.3 Puppetmaster

```
class puppet::master {
```

This class is not (yet?) portable among Linux flavors or other OSes.

```
    if $::osfamily != "RedHat" {
        unimplemented()
    }

    include puppet::client                        §11.81.1
    package { [
        "puppet-server",
```

Stored configs depend on Rails, which RHEL does not provide as RPMs, so we must install the gems.

```
        "rubygems",
        "ruby-pg",
        "ruby-devel",
        "postgresql-server",
    ]:
        ensure => installed,
    }

    package { "rails":
        provider => gem,
        ensure => installed,
        source => "",
    }
    file { "/etc/sysconfig/puppetmaster":
        owner => root, group => 0, mode => 0644,
        content => "\
PUPPETMASTER_LOG=syslog\n\
PUPPETMASTER_MANIFEST=/etc/puppet/manifests/site.pp\n",
        notify => Service['puppetmaster'],
    }
```

Install the SELinux rules that let puppetmaster do its job.

```
    $selmoduledir = "/usr/share/selinux/targeted"
    file { "${selmoduledir}/puppetmaster.pp":
        owner => root, group => 0, mode => 0644,
        source => "puppet:///modules/puppet/\
puppetmaster.selinux.pp",
    }
    selmodule { "puppetmaster":
       ensure => present,
       syncversion => true,
       notify => Service['puppetmaster'],
    }

    selboolean { "puppetmaster_use_db":
        value => on,
        persistent => true,
        notify => Service['puppetmaster'],
    }
```

We are no longer using the WEBrick based puppetmaster server.

```
    service { 'puppetmaster':
        enable => false,
        ensure => stopped,
    }
```

We're now using the one based on Apache and Passenger.

```
    service { 'httpd':
        enable => true,
        ensure => running,
    }
```

Fix some permissions roiled by other parts of the policy. If these are not fixed, the puppetmaster will try to fix them by chmodding files; and the SELinux policy says that things that httpd runs have no business chmodding anything. This results in 500 Internal Server Errors, rather than catalogs being served to clients.

Furthermore, these cannot be written as file resources, because then they become part of the very catalog that the puppetmaster is incapable of serving—even to itself—so the problem must be fixed outside Puppet.

The `/var/lib/puppet/lib` files must be readable by the `puppet` user because they contain Ruby code required by custom types; the Puppet master must import this code to compile manifests.

```
    $abbr_mhl = '/var/log/puppet/masterhttp.log'
    $abbr_cas = '/var/lib/puppet/ssl/ca/serial'
    $abbr_lib = '/var/lib/puppet/lib'
    cron { 'fix_puppetmaster_perms':
        command => "chown puppet:puppet $abbr_mhl; \
                    chmod 0660 $abbr_mhl; \
                    chown puppet:puppet $abbr_cas; \
                    chmod 0644 $abbr_cas; \
                    chown -R root:puppet $abbr_lib; \
                    chmod -R g+rX $abbr_lib; \
                   ",
        user => root,
        minute => '*/5',
    }
```

Some other permissions don't get in the way of the puppetmaster serving itself a catalog, but do get in the way of manifests being compiled into catalogs for other nodes.

```
    file { '/var/lib/puppet/lib':
        owner => root, group => puppet, mode => 0640,
        recurse => true, recurselimit => 9,
    }
```

Copy the `expect_and_sign` scripts into place. These adapt between Puppet's workflow, where certs are signed immediately after CSRs are generated on the client, and AFSEO's workflow, where we want to do most of the work when we receive notification that a new system will be coming online, not just after the system does come online.

These can't go in `/usr/local/sbin` because of the settings in root's `.bashrc`; see §11.84.5.

```
    file { '/usr/sbin/sign_expected':
        owner => root, group => 0, mode => 0755,
        source => 'puppet:///modules/puppet/sign_expected',
    }
    file { '/usr/sbin/expect_host':
        owner => root, group => 0, mode => 0755,
        source => 'puppet:///modules/puppet/expect_host',
    }
    file { '/usr/sbin/unexpect_host':
        owner => root, group => 0, mode => 0755,
        ensure => symlink,
        target => 'expect_host',
    }
    file { '/var/spool/sign_expected':
        ensure => directory,
        owner => root, group => 0, mode => 0700,
    }
    exec { 'run sign_expected at boot':
        unless => 'grep sign_expected /etc/rc.local',
        command => 'sed -i "/^touch/i \
/usr/sbin/sign_expected >&/var/log/sign_expected.log &" \
                    /etc/rc.local',
    }
```

Let admins run these scripts.

```
    sudo::auditable::command_alias { 'CMITS_PUPPET_SIGN_SCRIPTS':       §11.104.3
        type => 'exec',
        commands => [
            '/usr/sbin/expect_host',
            '/usr/sbin/unexpect_host',
            '/usr/sbin/sign_expected',
            ],
    }


    include subversion::pki::trust_cas                                  §11.103.1
```

Provide for admins to easily manually update the policy.

```
    file { '/usr/sbin/sudo_update_cmits_policy':
        owner => root, group => 0, mode => 0755,
        content => "#!/bin/sh
/usr/bin/sudo /usr/bin/svn --non-interactive up /etc/puppet
/usr/bin/sudo /sbin/restorecon -R /etc/puppet
/usr/bin/sudo /bin/chown -R puppet /etc/puppet
",
    }
```

Update the policy every hour.

```
    file { '/usr/sbin/update_cmits_policy':
        owner => root, group => 0, mode => 0700,
        content => "#!/bin/sh
/usr/bin/svn --non-interactive -q up /etc/puppet
/sbin/restorecon -R /etc/puppet
/bin/chown -R puppet /etc/puppet
",
    }
    cron { 'update_cmits_policy':
        hour => absent,
        minute => '*/10',
        command => '/usr/sbin/update_cmits_policy',
        require => File['/usr/sbin/update_cmits_policy'],
        user => root,
    }
```

Remove old reports, to avoid filling up the filesystem used for logs.

GNU-ism: `xargs -r`.

```
    cron { 'remove_old_logs':
        hour => 3,
        command => "/usr/bin/find /var/lib/puppet/reports \
                       -mtime +10 -type f -name \\*.yaml | \
                    /usr/bin/xargs -r -n 100 /bin/rm",
        user => root,
    }
}
```

## 11.82 Python

Install Python and whatever is necessary to use eggs.

```
class python {
    case $osfamily {
        "RedHat": {
            case $operatingsystemrelease {
                /6\..*/: { include python::rhel6 }
                /5\..*/: { include python::rhel5 }
                default: { unimplemented() }
            }
        }
```

Python not yet implemented under Darwin.

```
        'Darwin': {}
        default: {
            unimplemented()
        }
    }
}
```

### 11.82.1 Python for RHEL5

In AFSEO we tend to use Python 2.6 or later, and eggs. RHEL5 comes with Python 2.5. At some point we need to reconcile this gap. For now we just make a warning.

```
class python::rhel5 {
    warning "Not really implemented."
}
```

### 11.82.2 For RHEL6

RHEL6 comes with Python 2.6.6, a fine version of Python. It just needs the setuptools, which are also (finally!) part of the distro.

```
class python::rhel6 {
    package {
        "python":
            ensure => present;
        "python-setuptools":
            ensure => present;
    }
}
```

## 11.83 Red Hat Network Satellite

Red Hat Network (RHN) Satellite servers are manually set up, entirely according to Red Hat's fine documentation. (Seriously, it's well-written and complete.) Any exceptions will be noted and/or controlled here.

```
class rhn_satellite {
```

The RHN Satellite services are not managed by the service subsystem; there is a separate rhn-satellite executable which takes parameters stop, start, restart, status, etc.

```
    exec { 'rhn_satellite_restart':
        refreshonly => true,
        command => '/usr/sbin/rhn-satellite restart',
    }
}
```

### 11.83.1 Satellite authentication using PAM

This is in direct accordance with section 8.10 of the RHN Satellite Installation Guide [10].

To achieve Active Directory authentication, obtain and install a PAM module on the Satellite server. Centrify works at AFSEO; SSS (part of RHEL) may work for this purpose; other products are also available.

```
class rhn_satellite::pam {
```

In order to "create a PAM service file for RHN Satellite" and "edit the file with the following information: [...]," include one of the ensuing classes. The `sss` class does exactly what the Installation Guide says to.

"Instruct the satellite to use the PAM service file..." `rhn.conf` is a Java properties file.

```
    augeas { 'rhn_satellite_use_pam':
        require => Augeas['rhn_satellite_pam_d'],
        context => '/files/etc/rhn/rhn.conf',
        changes => 'set pam_auth_service rhn-satellite',
```

"Restart the service to pick up the changes."

```
        notify => Exec['rhn_satellite_restart'],
    }
}
```

### Use Centrify DirectControl

```
class rhn_satellite::pam::centrifydc {
    augeas { "rhn_satellite_pam_d":
        require => Package['CentrifyDC'],
        context => "/files/etc/pam.d/rhn-satellite",
        changes => [
            "rm *",
            "set 1/type     auth",
            "set 1/control  required",
            "set 1/module   pam_env.so",
            "set 2/type     auth",
            "set 2/control  sufficient",
            "set 2/module   pam_centrifydc.so",
            "set 3/type     auth",
            "set 3/control  requisite",
            "set 3/module   pam_centrifydc.so",
            "set 3/argument deny",
            "set 4/type     account",
            "set 4/control  sufficient",
            "set 4/module   pam_centrifydc.so",
            "set 5/type     account",
            "set 5/control  required",
            "set 5/module   pam_centrifydc.so",
        ],
    }
}
```

**Use System Security Services (SSS)**

```
class rhn_satellite::pam::sss {
    augeas { "rhn_satellite_pam_d":
        context => "/files/etc/pam.d/rhn-satellite",
        changes => [
            "rm *",
            "set 1/type    auth",
            "set 1/control required",
            "set 1/module  pam_env.so",
            "set 2/type    auth",
            "set 2/control sufficient",
            "set 2/module  pam_sss.so",
            "set 3/type    auth",
            "set 3/control required",
            "set 3/module  pam_deny.so",
            "set 4/type    account",
            "set 4/control sufficient",
            "set 4/module  pam_sss.so",
            "set 5/type    account",
            "set 5/control required",
            "set 5/module  pam_deny.so",
        ],
    }
}
```

## 11.84 The root user

### 11.84.1 Admin guidance regarding the root user

Never log in as root, except for "emergency maintenance, the use of single-user mode for maintenance, and situations where individual administrator accounts are not available."

admins do
GEN001020

Do not run a web browser under an administrative account, "except as needed for local service administration."

admins do
GEN004220

### 11.84.2 Where root can log in

Make sure root can only log in from the console.

auto: ECPA-1

"Console" means any tty listed in /etc/securetty. It's likely that some setting in /etc/login.defs could be set to ensure this property; but we can be more general by using PAM to enforce it instead.

auto: GEN000980
auto: GEN001020

```
class root::login {
    case $::osfamily {
        'RedHat': {
            include pam::securetty
```

§11.74.5

Make sure the /etc/securetty file contains exactly what it should.

Control ownership and permissions on the securetty file.

auto: ECLP-1
auto: GEN000000-LNX00620
auto: GEN000000-LNX00640
auto: GEN000000-LNX00660

```
            file { "/etc/securetty":
                owner => root, group => 0, mode => 0600,
                source => "puppet:///modules/root/login/securetty",
            }
```

Interestingly, there appears to be no STIG requirement to remove extended ACLs from this file. But we do it anyway.

```
            no_ext_acl { "/etc/securetty": }
        }
```

Mac OS X doesn't support root logins at all by default.

```
        'Darwin': {}
        default: { unimplemented() }
    }
}
```

### 11.84.3 Ask those logging in as root who they are

In order to preserve auditability even though `root` is a group authenticator, ask users logging in as root who they are.

Note that this has to be portable across all the platforms we use bash on.

```
class root::manual_audit {
    $bashrc = '/root/.bashrc'
    exec { 'add challenge 1 to root .bashrc':
        command => "sed -i.before_manual_audit -e '\$a \\
trap '\\'\\'' SIGINT\\
echo\\
echo \"Who are you and what are you doing?\"\\
echo \"Press Ctrl-D on an empty line when finished explaining.\"\\
sed '\\''s/[[:cntrl:]]/(CONTROL CHAR)/g'\\'' | \\\\\\
    logger -t \"ROOT LOGIN, user said\"\\
echo \"What you typed has been logged. Continuing.\"\\
trap - SIGINT\\
' ${bashrc}",
        unless => "grep 'root::manual_audit 1 ' ${bashrc}",
        path => '/bin:/sbin',
    }
}
```

### 11.84.4 Ensure only root has user id 0

```
class root::only_uid_0 {
    include "root::only_uid_0::${::osfamily}"
}
```

```
class root::only_uid_0::darwin {
```

Ensure that only root has user id 0.

auto: ECLP-1
auto: IAIA-1
auto: GEN000880 M6
auto: OSX8-00-01065

If the final grep exits without error, it found something. Then we run the command and log its output as errors. Because of the onlyif, we get no log messages if everything is OK.

```
    exec { 'warn if other users have uid 0':
        onlyif => 'dscl . -list /Users UniqueID | \
                    grep -w 0 | \
                    grep -v -w ^root',
        command => 'dscl . -list /Users UniqueID | \
                    grep -w 0 | \
                    grep -v -w ^root',
        loglevel => err,
    }
}
class root::only_uid_0::redhat {
```

Make sure root is the only user with a user id of 0. auto: ECLP-1

Log an error if any account besides root has a user id of 0. Do this by finding all users with a uid of 0, ignoring root (using `grep -v`). If any results remain to be printed, `grep` will exit with 0 (success). Then the command will be executed and its output logged as errors. N.B. `augtool match` does not reliably exit with any given exit code, so we must rely on grep here. See `http://www.redhat.com/archives/augeas-devel/2010-January/msg00100.html`. auto: GEN000880

```
    exec { "only_root_uid_0":
        onlyif =>
            "augtool match \
             /files/etc/passwd/\\*/uid[.=\\'0\\'] \
             | grep -v '^/files/etc/passwd/root/uid = 0'",
        command =>
            "augtool match \
             /files/etc/passwd/\\*/uid[.=\\'0\\'] \
             | grep -v '^/files/etc/passwd/root/uid = 0'",
        logoutput => true,
        loglevel => err,
        require => Class['augeas'],
    }
}
```

### 11.84.5 STIG-required configuration regarding the root user

Parameter `bashrc_variant` lets you choose what bashrc to use for root. This is needed because on most hosts it's necessary to find out which person is using a shared authenticator (i.e., the root account) and why, but on some hosts (e.g. Vagrant boxes) it's necessary to support automated root logins, without questions. In this case, give `'no_questions'` as the value of this parameter.

```
class root::stig($bashrc_variant='default') {
```

Make sure root can only login where root should.

```
    include root::login
```
§11.84.2

Make sure augeas is installed, so we can run `augtool`.

```
    include augeas
```
§11.13

Make sure only root has a UID of 0.

```
    include root::only_uid_0
```
§11.84.4

Make sure the root user's home directory is not `/`. auto: ECCD-1

We have a custom fact for root's home because we'll need it a bit farther down. auto: GEN000900

```
    case $::root_home {
        '/': {
            err("Root's home is /!")
        }
        '': {
            warning("Don't know root's home")
            file { "/root":
                owner => root,
                group => 0,
                mode => 0700,
            }
            no_ext_acl { "/root": }
        }
        default: {
```

Secure ownership and permissions of root's home directory. auto: ECCD-1

We only want to do this if root's home is not `/`. auto: GEN000920

```
            file { "$::root_home":
                owner => root,
                group => 0,
                mode => 0700,
            }
```

Remove extended ACLs from root's home directory. auto: ECLP-1

auto: GEN000930

```
            no_ext_acl { "$::root_home": }
        }
    }
```

Make sure root uses bash, so that root's `.bashrc` will happen when someone becomes root. If the same code in the bashrc were ported to csh, we would not need to force root to use bash; but bash for root is already a vendor default.

Do not change this policy in a manner to cause root to use a shell not located on the root (/) filesystem. admins do GEN001080

```
    augeas { "root_use_bash":
        context => "/files/etc/passwd/*[name='root']",
        changes => "set shell /bin/bash",
    }
```

Make sure that root's `PATH`, `LD_LIBRARY_PATH`, and `LD_PRELOAD` environment variables are secure, and that no world-writable directories are on root's `PATH`. auto: ECCD-1 auto: ECSC-1 auto: GEN000940 auto: GEN000945 auto: GEN000950 auto: GEN000960

```
    file { "${::root_home}/.bashrc":
        owner => root, group => 0, mode => 0640,
        source => "puppet:///modules/root/bashrc.${bashrc_variant}",
    }

    include "root::stig::${::osfamily}"
}
class root::stig::darwin {
```

Make sure the root account is disabled for interactive use. auto: OSX8-00-01230

```
  exec { ’disable root interactive login’:
    command => ’dsenableroot -d’,
```

dscl should say, “No such key: AuthenticationAuthority.” If it says anything else, we want to run the command.

```
    onlyif => ’dscl . -read /Users/root \
                AuthenticationAuthority \
          2>&1 | grep -v "^No such key:"’,
  }
}
class root::stig::redhat {
}
```

# 11.85 RPM Package Manager

# 11.86 Managing GPG keys in the RPM database

This defined resource type can manage GPG keys used to sign RPM packages.

Example:

```
rpm::gpgkey { ’d3adb33f’: source => ’http://myserver/pub/d3adb33f.key’ }
```

The name should be an eight-digit hexadecimal number, the key identifier; the source can be anything that `rpm --import` understands, like an http URL, or an absolute path to a file that exists and contains the GPG public key. For the optional ensure parameter you can give values ‘present’ or ‘absent’; it defaults to ‘present’.

```
define rpm::gpgkey($source, $ensure=’present’) {
  case $ensure {
    ’present’: {
      exec { "import rpm gpg key ${name}":
        command => "rpm --import ${source}",
        unless => "rpm -q gpg-pubkey-${name}",
      }
    }
    ’absent’: {
      exec { "remove rpm gpg key ${name}":
        command => "rpm -e gpg-pubkey-${name}",
        onlyif => "rpm -q gpg-pubkey-${name}",
      }
    }
  }
}
```

## 11.86.1 STIG-required RPM package manager configuration

```
class rpm::stig($known_unsigned_packages=[]) {
```

Use the RPM package manager’s verify feature to cryptographically verify auto: ECAT-1 auto: GEN006565 the integrity of installed system software monthly.

Use RPM's verify feature to cryptographically verify the integrity of installed software for DBMSes included with RHEL. auto: DCSW-1 auto: DG0021

```
    file { "/etc/cron.monthly/rpmV.cron":
        owner => root, group => 0, mode => 0700,
        source => "puppet:///modules/rpm/rpmV.cron",
    }
```

Make sure all packages installed have cryptographic signatures. auto: ECSC-1

(rpm -V as above will warn about files which have been changed since they were installed, but if the installed package is not signed, files from an untrusted source could have been installed via the package system.) auto: GEN008800

Some packages may not be signable. If so, list them in the known_unsigned_packages parameter to this class. You should not share the list of these with the world, because it is a list of weaknesses.

```
    file { "/etc/cron.weekly/rpm-signatures.cron":
        owner => root, group => 0, mode => 0700,
        content => template("rpm/rpm-signatures.cron.erb"),
    }
}
```

## 11.87 rsh, rlogin, rexec

Unencrypted command execution and terminal access. Old, unused, and prohibited by the UNIX SRG.

### 11.87.1 Disable rsh, rlogin, and rexec

```
class rsh::no {
    include "rsh::no::${::osfamily}"
}
```

**Disable rsh, rlogin, and rexec under Mac OS X**

```
class rsh::no::darwin {
```

Make sure the rsh daemon is not running. auto: EBRU-1 auto: GEN003820 M6 auto: OSX8-00-00050

```
    service { 'com.apple.rshd':
        enable => false,
        ensure => stopped,
    }
```

Make sure the rexec daemon is not running. auto: GEN003840 M6 auto: OSX8-00-00035

```
    service { 'com.apple.rexecd':
        enable => false,
        ensure => stopped,
    }
```

Make sure the telnet daemon is not running. auto: DCPP-1 auto: GEN003850 M6 auto: OSX8-00-00040

```
    service { 'com.apple.telnetd':
        enable => false,
        ensure => stopped,
    }
```

Make sure the finger daemon is not running. auto: DCPP-1 auto: EBRU-1 auto: GEN003860 M6 auto: OSX8-00-01115

```
        service { 'com.apple.fingerd':
            enable => false,
            ensure => stopped,
        }
    }
```

**Disable rsh, rlogin, and rexec under Red Hat**

```
class rsh::no::redhat {
```

Under RHEL, to ensure that rsh and rlogin are disabled, uninstall them.

auto: DCPP-1
auto: EBRU-1
auto: ECSC-1
auto: GEN003820
auto: GEN003825
auto: GEN003830
auto: GEN003835
auto: GEN003840
auto: GEN003845

(Under RHEL, `rsh`, `rlogin`, `rexec` and `rcp` and their respective servers all come in two packages.)

```
        package {
            "rsh": ensure => absent;
            "rsh-server": ensure => absent;
        }
    }
```

## 11.88 Samba

The SRG imposes some important requirements on how Samba is to be configured (e.g., do not allow guest access), which are not merely a matter of switching things on and off but impact deployment planning. We do not implement any of these because we do not run any Samba servers. Any implementation of Samba servers in the future needs to take these into account.

N/A: GEN006080
N/A: GEN006220
N/A: GEN006225
N/A: GEN006230
N/A: GEN006235

### 11.88.1 Remove Samba

Remove Samba “unless needed.” We do not need it here.

auto: ECSC-1
auto: GEN006060

```
    class samba::no {
        package {
            "samba-swat": ensure => absent;
            "samba": ensure => absent;
            "samba4": ensure => absent;
        }
    }
```

### 11.88.2 STIG-required Samba configuration

Even though we aren't using Samba, any remaining configuration files are subject to STIG requirements.

```
    class samba::stig {
        case $::osfamily {
            'redhat': { include samba::stig::redhat }
            'darwin': { include samba::stig::darwin }
            default:  { unimplemented() }
        }
    }
```

### 11.88.3 STIG-required Samba configuration under Mac OS X

```
class samba::stig::darwin {
```

Control ownership and permissions of `smb.conf`.

auto: ECLP-1
auto: GEN006100 M6
auto: GEN006140 M6

```
    file { "/etc/smb.conf":
        owner => root, group => 0, mode => 0644,
    }
```

Remove extended ACLs on `smb.conf`.

auto: ECLP-1
auto: GEN006150 M6

```
    no_ext_acl { "/etc/smb.conf": }
}
```

### 11.88.4 STIG-required Samba configuration under Red Hat

```
class samba::stig::redhat {
```

Control ownership and permissions of `smb.conf`.

Under RHEL, all Samba configuration goes under `/etc/samba`, so we secure `/etc/samba/smb.conf` not `/etc/smb.conf`.

auto: ECLP-1
auto: GEN006100
auto: GEN006120
auto: GEN006140

```
    file { "/etc/samba/smb.conf":
        owner => root, group => 0, mode => 0644,
    }
```

Remove extended ACLs on `smb.conf`.

auto: ECLP-1
auto: GEN006150

```
    no_ext_acl { "/etc/samba/smb.conf": }
```

Control ownership and permissions of `smbpasswd`.

auto: ECLP-1
auto: GEN006160
auto: GEN006180
auto: GEN006200

```
    file { "/etc/samba/smbpasswd":
        owner => root, group => 0, mode=> 0600,
    }
```

Remove extended ACLs on `smbpasswd`.

auto: ECLP-1
auto: GEN006210

```
    no_ext_acl { "/etc/samba/smbpasswd": }

}
```

## 11.89 AFSEO Sensitive but Unclassified (SBU) Website

### 11.89.1 Unimplemented Apache STIG requirements

(Some unimplemented requirements, having to do with the Apache server configuration, are listed therein.)

We grant write access to web clients for Incoming directories on the SBU. WG290 A22

### 11.89.2 The auth database

The `auth` database on an SBU server contains the list of users and groups, which the web server consults when making authentication and authorization decisions.

Requirements marked implemented in this section are only implemented in the context of the SBU system. See `https://afseo.eglin.af.mil/projects/ihaaa/ticket/375`.

The mode parameter must be one of ‘production’, ‘installation’ or ‘development’. If installation or development, the builder must be specified. This is the OS user who will be allowed to (re)build the auth database.

```
class sbu::auth_db(
        $mode = 'production',
        $builder = 'jenninjl') {
```

Data in the auth database is security information, so all changes to it should be audited.

```
    class { 'postgresql':
        audit_data_changes => true,
    }
```

§11.78

Do all database administration as `puppet_dba`.

```
    Pgsql_role {
        os_user =>  'puppet_dba',
        db_user =>  'puppet_dba',
        database => 'puppet_dba',
    }
    Pgsql_database {
        os_user =>  'puppet_dba',
        db_user =>  'puppet_dba',
        database => 'puppet_dba',
    }
```

Prevent the misuse of DBA accounts for non-administrative purposes by creating an object owner user.

auto: ECLP-1
auto: DG0124

Disable the application object owner user “when not performing installation or maintenance actions.”

auto: ECLP-1
auto: DG0004

```
    pgsql_role { "sbu_aoou":
        login => $mode ? {
            'installation' => true,
            'development'  => true,
            default        => false,
        },
        inherit => true,
    }

    pgsql_database { "auth":
        owner => "sbu_aoou",
    }
```

SBU-specific roles. Permissions regarding database objects are granted to these roles by the SQL scripts which create the database objects.

```
pgsql_role {
    ’sbu_mod_auth_pgsql_access_log_r’:;
    ’sbu_mod_auth_pgsql_authnz_r’:;
    ’sbu_authapp_r’:;
    ’sbu_authapp_auto_testing_r’:;
    ’sbu_authorization_finder_r’:;
```

Now, SBU-specific users.

```
    ’sbu_authapp’:
        login => true,
        inherit => true,
        grant_roles => $mode ? {
            ’development’ => [
                ’sbu_authapp_r’,
                ’sbu_authapp_auto_testing_r’,
            ],
            default => [
                ’sbu_authapp_r’,
            ],
        };
    ’sbu_mod_auth_pgsql’:
        login => true,
        inherit => true,
        grant_roles => [
            ’sbu_mod_auth_pgsql_access_log_r’,
            ’sbu_mod_auth_pgsql_authnz_r’,
        ];
    ’sbu_upload’:
        login => true,
        inherit => true,
        grant_roles => ’sbu_authorization_finder_r’;
}

case $mode {
    ’development’, ’installation’: {
        pgsql_role { $builder:
            grant_roles => [’sbu_aoou’],
            createdb => true,
            login => true,
            inherit => true,
        }
    }
}
```

Configure `pg_hba.conf` and `pg_ident.conf` to let people connect to auth using an ident map. This is not yet automated.

| **OS user** | **can connect with DB username** |
|---|---|
| apache | `sbu_mod_auth_pgsql` |
| apache | `sbu_authapp` |
| apache | `sbu_upload` |
| developers | `sbu_authapp` |
| developers and installers | `sbu_aoou` |

```
}
```

### 11.89.3 Server deployment

The mode parameter must be one of ‘production’, ‘installation’ or ‘development’. If installation or development, the builder must be specified. This is the OS user who will be allowed to (re)build the auth database.

```
class sbu::server(
        $mode = 'production',
        $builder = 'jenninjl',
        $cert_nickname = $::hostname,
        $http_proxy,
        $admin_email_address,
        $web_fqdn=$::fqdn,
    ) {
    $dbdir = "/etc/pki/mod_nss"
    class { 'apache':                                                  §11.8
        production => $mode ? {
            'production' => true,
            default      => false,
        }
    }
    class { 'apache::config':                                          §11.8.1
        nss_database_dir => $dbdir,
```

Max request body size is 8 gigabytes.

```
        max_request_body => 8589934592,
    }
    apache::config::nss_site { 'sbu':                                  §11.8.1
        content => template('sbu/sbu.conf'),
    }
    # there are some configurations that aren't included in the nss
    # site config file
    file { '/etc/httpd/conf.d':
        owner => root, group => 0, mode => 0600,
        source => 'puppet:///modules/sbu/etc-httpd-conf.d',
    }

    if $mode == 'production' {
        include sbu_fouo::data_structure                               §??
    }

    include python                                                     §11.82
    class { 'sbu::auth_db':                                            §11.92.2
```

```
        mode      => $mode,
        builder   => $builder,
    }

    package {
        [
            "mod_perl",
            "mod_wsgi",
            "mod_dav_svn",
            "mod_authz_ldap",
            "mod_auth_pgsql",
            "python-coverage",
            "python-nose",
            "python-cheetah",
            "python-formencode",
            "python-psycopg2",
            "python-ldap",
            "pyOpenSSL",
            "make",
        ]:
            ensure => present,
    }

    pki::nss::db { $dbdir:                                           §11.76.5
        owner => apache, group => 0, mode => 0600,
        sqlite => false,
        pwfile => true,
    }
    pki::nss::dod_roots { $dbdir:                                    §11.76.10
        pwfile => "$dbdir/pwfile",
        sqlite => false,
    }
    pki::nss::dod_cas { $dbdir:                                      §11.76.6
        pwfile => "$dbdir/pwfile",
        sqlite => false,
    }
```

No e-mail CAs: we want TortoiseSVN not to ask the user whether to use identity or email signing cert *ad nauseam*.

```
    pki::nss::eca_roots { $dbdir:                                    §11.76.12
        pwfile => "$dbdir/pwfile",
        sqlite => false,
    }
    pki::nss::eca_cas { $dbdir:                                      §11.76.11
        pwfile => "$dbdir/pwfile",
        sqlite => false,
    }
```

Follow approved trust path from DoD CAs to Australian Defence Organization (ADO) CAs.

```
    pki::nss::australia { $dbdir:                                    §11.76.4
        pwfile => "$dbdir/pwfile",
        sqlite => false,
    }
```

```
pki::nss::crl { "mod_nss":
    dbdir => $dbdir,
    pwfile => "${dbdir}/pwfile",
    http_proxy => $http_proxy,
    sqlite => false,
}
```

§11.76.4

```
include sbu::trac
```

§11.92.4

We can't put things under `/var/www` if `/var/www` doesn't exist. That directory is put in place by the `httpd` package. When we depend on the whole `apache` class, dependency cycles happen, so we have to depend on the package.

```
file { "/var/www/virus-checkpoint":
    ensure => directory,
    owner => apache, group => 0, mode => 0700,
    require => Package['httpd'],
}
```

Make sure everyone can read the public things.

```
file { "/var/www/html/styles":
    ensure => directory,
    owner => root, group => 0, mode => 0644,
    require => Package['httpd'],
}
file { "/var/www/html/pages":
    ensure => directory,
    owner => root, group => 0, mode => 0644,
    require => Package['httpd'],
}
```

"Protect access to authentication data by restricting access to authorized users and services." auto: APP3360

No authentication data is hardcoded in the application, of course (APP3350), only written in the configuration; but this is also where we control access to the files that make up the application.

Ensure that "application software and configuration files" dependent on the database are owned by "the software installation account or the designated owner account," in the context of the AFSEO SBU system. auto: DCSL-1 auto: DG0019

It is possible that APP3360 does not regard file permissions. But they still need to be set.

On development systems and those undergoing installation, the builder of the database should own the code, and not be prevented from writing to it.

```
$os_app_owner = $mode ? {
    'development'  => $builder,
    'installation' => $builder,
    default        => root,
}
$os_exec_perms = $mode ? {
    'development'  => 0750,
    'installation' => 0550,
    default        => 0550,
}
$os_noexec_perms = $mode ? {
    'development'  => 0640,
    'installation' => 0440,
    default        => 0440,
}

file { [
        '/var/www/sbu-apps',
        '/var/www/sbu-apps/authapp',
        '/var/www/sbu-apps/authapp/config',
        '/var/www/sbu-apps/upload',
        '/var/www/sbu-apps/upload/config',
       ]:
    ensure => directory,
    owner => $os_app_owner,
    group => apache,
    mode => $os_noexec_perms,
    recurse => true,
    recurselimit => 4,
    require => Package['httpd'],
}

# things that should be executable
file { [
        '/var/www/sbu-apps/authapp/public/go.py',
        '/var/www/sbu-apps/upload/public/go.py',
        '/var/www/sbu-apps/authapp/script/approve_cron.py',
        '/var/www/sbu-apps/authapp/script/expire_cron.py',
        '/var/www/sbu-apps/authapp/script/expiringSoon_cron.py',
        '/var/www/sbu-apps/authapp/script/inactivity_cron.py',
        ]:
    owner => $os_app_owner,
    group => apache,
    mode => $os_exec_perms,
    require => Package['httpd'],
}
```

Put symlinks in place for things that need to happen every morning.

```
    file {
        '/etc/cron.morningly/sbu_approve_cron':
            ensure => present,
            owner => root, group => 0, mode => 0700,
            content => "#!/bin/sh\n\
/sbin/runuser apache -s /bin/sh -c \
  /var/www/sbu-apps/authapp/script/approve_cron.py\n";

        '/etc/cron.morningly/sbu_expire_cron':
            ensure => present,
            owner => root, group => 0, mode => 0700,
            content => "#!/bin/sh\n\
/sbin/runuser apache -s /bin/sh -c \
  /var/www/sbu-apps/authapp/script/expire_cron.py\n";

        '/etc/cron.morningly/sbu_expiringSoon_cron.py':
            ensure => present,
            owner => root, group => 0, mode => 0700,
            content => "#!/bin/sh\n\
/sbin/runuser apache -s /bin/sh -c \
  /var/www/sbu-apps/authapp/script/expiringSoon_cron.py\n";

 # FIXME: we name the ssl activity log both here and in the templated
 # httpd config. Come up with a variable for this.

        '/etc/cron.morningly/sbu_inactivity_cron.py':
            ensure => present,
            owner => root, group => 0, mode => 0700,
            content => "#!/bin/sh
cat /var/log/httpd/ssl_activity_log | \\
  /sbin/runuser apache -s /bin/sh -c \\
    /var/www/sbu-apps/authapp/script/inactivity_cron.py
";
    }
```

The DocumentRoot for the password-based Subversion virtual site needs to exist. Nothing needs to be in it, because the only thing served is the Subversion repositories, which mod_dav_svn takes care of.

```
    file { '/var/www/svn-html':
        ensure => directory,
        owner => root, group => apache, mode => 0755,
        require => Package['httpd'],
    }
```

Install the SELinux rules that let SBU apps log errors through the syslog.

```
    $selmoduledir = "/usr/share/selinux/targeted"
    file { "${selmoduledir}/sbu_apps.pp":
        owner => root, group => 0, mode => 0644,
        source => "puppet:///modules/sbu/selinux/\
sbu_apps.selinux.pp",
    }
    selmodule { "sbu_apps":
       ensure => present,
       syncversion => true,
    }
```

Install some convenience scripts. These would work for any web server where the Apache log messages are directed to the system log; but at present there is no policy-based means by which Apache is configured to do this, so it's up to the (SBU-specific) Apache configuration.

```
    file {

        "/usr/local/bin/tail_httpd_access":
            ensure => present,
            owner => root, group => 0, mode => 0755,
            content => "#!/bin/sh\n\
/usr/bin/tail -f /var/log/messages | \
grep --line-buffered httpd__access\n";

        "/usr/local/bin/tail_httpd_error":
            ensure => present,
            owner => root, group => 0, mode => 0755,
            content => "#!/bin/sh\n\
/usr/bin/tail -f /var/log/messages | \
grep --line-buffered ’httpd[^_]’\n";

        "/usr/local/bin/tail_httpd":
            ensure => present,
            owner => root, group => 0, mode => 0755,
            content => "#!/bin/sh\n\
/usr/bin/tail -f /var/log/messages | \
grep --line-buffered httpd\n";

        "/usr/local/bin/HR":
            ensure => present,
            owner => root, group => 0, mode => 0755,
            content => "#!/bin/sh\n\
/sbin/service httpd restart\n";

    }
```

Let the authapp send mail. The `httpd_can_sendmail` sebool appears to allow `httpd_sys_script_t` to run an MTA user agent (like `mail(1)`, perhaps), but not to open a TCP socket itself to talk to the MTA. For that we need two things:

1. The sebool `httpd_can_network_connect`

2. The SELinux contexts of the authapp CGI executable files to be set properly.

```
    selboolean { 'httpd_can_network_connect':
        value => on,
        persistent => true,
    }
}
```

### 11.89.4 Trac

Deploy Trac in such a way as to support multiple instances.

The installation of Trac is documented in the SBU administrator's guide [?]. Here we just take care of the multi-project part.

```
class sbu::trac {
    file { "/var/www/wsgi-bin":
        ensure => directory,
        owner => root, group => 0, mode => 0755,
    }
    file { "/var/www/wsgi-bin/trac.wsgi":
        ensure => file,
        owner => root, group => 0, mode => 0755,
        source => "puppet:///modules/sbu/trac/trac.wsgi",
    }
```

Configure Trac instances on the SBU server to show a banner with a security label at the top of each page. auto: ECML-1

Install the requisite templates in a directory common to all Trac instances.

```
    $tracs = '/var/www/tracs'
    $trac_common = "${tracs}/_common"

    file {
        "$tracs":
            ensure => directory,
            owner => root, group => 0, mode => 0755;
        "$trac_common":
            ensure => directory,
            owner => root, group => 0, mode => 0755;
        "$trac_common/templates":
            ensure => directory,
            owner => root, group => 0, mode => 0755;
        "$trac_common/templates/site.html":
            owner => root, group => 0, mode => 0644,
            source => 'puppet:///modules/sbu/trac/site.html';
        "$trac_common/templates/classbar.html":
            owner => root, group => 0, mode => 0644,
            source => 'puppet:///modules/sbu/trac/classbar.html';
    }
```

Configure all Trac instances to inherit templates from the sitewide directory set up above.

Specifically, in each trac.ini, add an inherit section if there isn't one, and set the `templates_dir` setting in that section to the common templates directory.

```
       augeas { 'trac_inherit_common_templates':
           context => '/files/var/www/tracs/*/conf/trac.ini',
           changes => [
               "setm . inherit '' ",
               "setm inherit templates_dir '$trac_common/templates'",
           ],
       }
   }
   class sbu::vagrant(
       $mode = 'development',
       $builder = 'vagrant')
   {
       class { 'sbu::server':                                      §11.89.3
           mode => $mode,
           builder => $builder,
           cert_nickname => $::hostname,
       }
       pki::nss::self_signed { "${sbu::server::dbdir}:${::hostname}":     §11.76.12
           sqlite => false,
           noise_file => '/vagrant/insecure_noisefile',
       }
       file { '/etc/httpd/conf.d/Data.perms':
           ensure => present,
           owner => root, group => 0, mode => 0600,
           content => '',
       }
   }
```

# 11.90 Screen sharing

## 11.90.1 Disable screen sharing

```
class screen_sharing::no {
    include "screen_sharing::no::${::osfamily}"
}
   class screen_sharing::no::darwin {
       $version_underscores = regsubst(
           $::macosx_productversion_major,
           '\D', '_', 'G')
       $klassname = "${::osfamily}_${version_underscores}"
       include "screen_sharing::no::${klassname}"
   }
   class screen_sharing::no::darwin_10_6 {}
   class screen_sharing::no::darwin_10_9 {
       service { 'com.apple.screensharing':
           ensure => stopped,
           enable => false,
       }
   }
   class screen_sharing::no::redhat {}
```

## 11.91 Screen saver

Configure screen saver.

### 11.91.1 Require authentication to exit screensaver

```
class screensaver::authenticate {
```

Password-protect Mac screensavers.

This requirement is in the rule title of Mac OS X STIG PDI OSX00360 M6, but not in the check or fix content. Mac OS X STIG PDI OSX00420 M6 directly requires it.

auto: PESL-1
auto: OSX00360 M6
auto: OSX00420 M6
auto: OSX8-00-00020

```
       mcx::set {
           'com.apple.screensaver/askForPassword':
               value => 1;
           'com.apple.screensaver/askForPasswordDelay':
               value => 0;
       }
   }
```

§11.61.2

### 11.91.2 Disallow admins from unlocking user screens

Disable administrative accounts from unlocking other users' screens.

Mac OS X has a setting which when turned on lets not only the user who locked the screen unlock it, but also any admin. The STIG requires that this setting be turned off. Admins are still able to unlock their own screens, just not those of other users.

auto: ECPA-1
auto: PESL-1
auto: OSX00200 M6
auto: OSX8-00-00935

```
class screensaver::no_admin_unlock {
    case $::macosx_productversion_major {
        "10.6": {
            mac_plist_value { 'disable_admin_screensaver_unlock':
                file => '/etc/authorization',
                key => ['rights', 'system.login.screensaver', 'rule'],
                value => 'authenticate-session-owner',
            }
        }
        "10.9": {
            mac_authz_plist_value { 'no admin unlock screensaver':
                right => 'system.login.screensaver',
                key => ['rule'],
                value => ['authenticate-session-owner', ''],
            }
        }
        default: { unimplemented() }
    }
}
```

```
class screensaver::public_pattern {
```

Ensure that the screensaver shows a publicly viewable pattern.

auto: OSX8-00-00005

```
    mcx::set { ’com.apple.screensaver/moduleName’:      §11.61.2
        value => ’Flurry’,
    }
}
```

### 11.91.3 STIG-required configuration

Configure the Mac screensaver as required by the Mac OS X STIG.

```
class screensaver::stig {
    include screensaver::public_pattern                 §11.91.2
    include screensaver::no_admin_unlock                §11.91.2
```

Set the screensaver idle timeout to “15 minutes or less.” auto: PESL-1

```
    class { ’screensaver::timeout’:                     auto: OSX00360 M6
        seconds => 900,                                 auto: OSX8-00-00010
    }                                                   §11.91.4
```

Implied by the rule title of Mac OS X STIG PDI OSX00360 M6 but not covered by the check and fix content is that the screensaver must require authentication to unlock.

```
    include screensaver::authenticate                   §11.91.1
}
```

### 11.91.4 Set screensaver timeout

Set a mandatory screensaver timeout for everyone.

```
class screensaver::timeout($seconds) {
    mcx::set { ’com.apple.screensaver/idleTime’:        §11.61.2
        value => $seconds,
    }
}
```

## 11.92 AFSEO Sensitive but Unclassified (SBU) Website

### 11.92.1 Unimplemented Apache STIG requirements

(Some unimplemented requirements, having to do with the Apache server configuration, are listed therein.)

We grant write access to web clients for Incoming directories on the SBU. WG290 A22

### 11.92.2 The auth database

The `auth` database on an SBU server contains the list of users and groups, which the web server consults when making authentication and authorization decisions.

Requirements marked implemented in this section are only implemented in the context of the SBU system. See `https://afseo.eglin.af.mil/projects/ihaaa/ticket/375`.

The mode parameter must be one of ‘production’, ‘installation’ or ‘development’. If installation or development, the builder must be specified. This is the OS user who will be allowed to (re)build the auth database.

```
class sbu::auth_db(
        $mode = 'production',
        $builder = 'jenninjl') {
```

Data in the auth database is security information, so all changes to it should be audited.

§11.78

```
    class { 'postgresql':
        audit_data_changes => true,
    }
```

Do all database administration as `puppet_dba`.

```
    Pgsql_role {
        os_user =>  'puppet_dba',
        db_user =>  'puppet_dba',
        database => 'puppet_dba',
    }
    Pgsql_database {
        os_user =>  'puppet_dba',
        db_user =>  'puppet_dba',
        database => 'puppet_dba',
    }
```

Prevent the misuse of DBA accounts for non-administrative purposes by creating an object owner user.

auto: ECLP-1
auto: DG0124

Disable the application object owner user “when not performing installation or maintenance actions.”

auto: ECLP-1
auto: DG0004

```
    pgsql_role { "sbu_aoou":
        login => $mode ? {
            'installation' => true,
            'development'  => true,
            default        => false,
        },
        inherit => true,
    }

    pgsql_database { "auth":
        owner => "sbu_aoou",
    }
```

SBU-specific roles. Permissions regarding database objects are granted to these roles by the SQL scripts which create the database objects.

```
    pgsql_role {
        'sbu_mod_auth_pgsql_access_log_r':;
        'sbu_mod_auth_pgsql_authnz_r':;
        'sbu_authapp_r':;
        'sbu_authapp_auto_testing_r':;
        'sbu_authorization_finder_r':;
```

Now, SBU-specific users.

```
    'sbu_authapp':
        login => true,
        inherit => true,
        grant_roles => $mode ? {
            'development' => [
                'sbu_authapp_r',
                'sbu_authapp_auto_testing_r',
            ],
            default => [
                'sbu_authapp_r',
            ],
        };
    'sbu_mod_auth_pgsql':
        login => true,
        inherit => true,
        grant_roles => [
            'sbu_mod_auth_pgsql_access_log_r',
            'sbu_mod_auth_pgsql_authnz_r',
        ];
    'sbu_upload':
        login => true,
        inherit => true,
        grant_roles => 'sbu_authorization_finder_r';
}

case $mode {
    'development', 'installation': {
        pgsql_role { $builder:
            grant_roles => ['sbu_aoou'],
            createdb => true,
            login => true,
            inherit => true,
        }
    }
}
```

Configure `pg_hba.conf` and `pg_ident.conf` to let people connect to auth using an ident map. This is not yet automated.

| **OS user** | **can connect with DB username** |
|---|---|
| apache | `sbu_mod_auth_pgsql` |
| apache | `sbu_authapp` |
| apache | `sbu_upload` |
| developers | `sbu_authapp` |
| developers and installers | `sbu_aoou` |

```
}
```

### 11.92.3 Server deployment

The mode parameter must be one of 'production', 'installation' or 'development'. If installation or development, the builder must be specified. This is the OS user who will be allowed to (re)build the auth database.

```
class searde_svn::server(
        $mode = 'production',
        $builder = 'jenninjl',
        $cert_nickname = $::hostname,
        $http_proxy,
        $admin_email_address,
        $web_fqdn=$::fqdn,
    ) {
    $dbdir = "/etc/pki/mod_nss"
    class { 'apache':                                                   §11.8
        production => $mode ? {
            'production' => true,
            default      => false,
        }
    }
    class { 'apache::config':                                           §11.8.1
        nss_database_dir => $dbdir,
```

Max request body size is 8 gigabytes.

```
        max_request_body => 8589934592,
    }
    apache::config::nss_site { 'searde_svn':                            §11.8.1
        content => template('searde_svn/searde_svn.conf'),
    }
    # there are some configurations that aren't included in the nss
    # site config file
    file { '/etc/httpd/conf.d':
        owner => root, group => 0, mode => 0600,
        source => 'puppet:///modules/searde_svn/etc-httpd-conf.d',
    }


    include python                                                      §11.82
    package {
        [
            "mod_wsgi",
            "mod_dav_svn",
        ]:
            ensure => present,
    }

    pki::nss::db { $dbdir:                                              §11.76.5
        owner => apache, group => 0, mode => 0600,
        sqlite => false,
        pwfile => true,
    }
    pki::nss::dod_roots { $dbdir:                                       §11.76.10
        pwfile => "$dbdir/pwfile",
        sqlite => false,
    }
    pki::nss::dod_cas { $dbdir:                                         §11.76.6
```

```
        pwfile => "$dbdir/pwfile",
        sqlite => false,
    }
```

No e-mail CAs: we want TortoiseSVN not to ask the user whether to use identity or email signing cert *ad nauseam*.

```
    pki::nss::eca_roots { $dbdir:                              §11.76.12
        pwfile => "$dbdir/pwfile",
        sqlite => false,
    }
    pki::nss::eca_cas { $dbdir:                                §11.76.11
        pwfile => "$dbdir/pwfile",
        sqlite => false,
    }
    pki::nss::crl { "mod_nss":                                 §11.76.4
        dbdir => $dbdir,
        pwfile => "${dbdir}/pwfile",
        http_proxy => $http_proxy,
        sqlite => false,
    }

    include searde_svn::trac                                   §??
```

We can't put things under `/var/www` if `/var/www` doesn't exist. That directory is put in place by the `httpd` package. When we depend on the whole `apache` class, dependency cycles happen, so we have to depend on the package.

```
    file { "/var/www/virus-checkpoint":
        ensure => directory,
        owner => apache, group => 0, mode => 0700,
        require => Package['httpd'],
    }
```

Make sure everyone can read the public things.

```
    file { "/var/www/html/styles":
        ensure => directory,
        owner => root, group => 0, mode => 0644,
        require => Package['httpd'],
    }
    file { "/var/www/html/pages":
        ensure => directory,
        owner => root, group => 0, mode => 0644,
        require => Package['httpd'],
    }
```

TODO THESE COULD BE MOVED TO APACHE MODULE Install some convenience scripts. These would work for any web server where the Apache log messages are directed to the system log; but at present there is no policy-based means by which Apache is configured to do this, so it's up to the (SBU-specific) Apache configuration.

```
    file {

        "/usr/local/bin/tail_httpd_access":
            ensure => present,
            owner => root, group => 0, mode => 0755,
            content => "#!/bin/sh\n\
/usr/bin/tail -f /var/log/messages | \
grep --line-buffered httpd__access\n";

        "/usr/local/bin/tail_httpd_error":
            ensure => present,
            owner => root, group => 0, mode => 0755,
            content => "#!/bin/sh\n\
/usr/bin/tail -f /var/log/messages | \
grep --line-buffered ’httpd[^_]’\n";

        "/usr/local/bin/tail_httpd":
            ensure => present,
            owner => root, group => 0, mode => 0755,
            content => "#!/bin/sh\n\
/usr/bin/tail -f /var/log/messages | \
grep --line-buffered httpd\n";

        "/usr/local/bin/HR":
            ensure => present,
            owner => root, group => 0, mode => 0755,
            content => "#!/bin/sh\n\
/sbin/service httpd restart\n";

    }
```

TODO Factor this into sub-class in apache module Let the authapp send mail. The `httpd_can_sendmail` sebool appears to allow `httpd_sys_script_t` to run an MTA user agent (like `mail(1)`, perhaps), but not to open a TCP socket itself to talk to the MTA. For that we need two things:

1. The sebool `httpd_can_network_connect`
2. The SELinux contexts of the authapp CGI executable files to be set properly.

```
    selboolean { ’httpd_can_network_connect’:
        value => on,
        persistent => true,
    }
}
```

### 11.92.4 Trac

Deploy Trac in such a way as to support multiple instances.

The installation of Trac is documented in the SBU administrator's guide [**?**]. Here we just take care of the multi-project part.

```
class sbu::trac {
    file { "/var/www/wsgi-bin":
        ensure => directory,
        owner => root, group => 0, mode => 0755,
    }
    file { "/var/www/wsgi-bin/trac.wsgi":
        ensure => file,
        owner => root, group => 0, mode => 0755,
        source => "puppet:///modules/sbu/trac/trac.wsgi",
    }
```

Configure Trac instances on the SBU server to show a banner with a security label at the top of each page. auto: ECML-1

Install the requisite templates in a directory common to all Trac instances.

```
    $tracs = '/var/www/tracs'
    $trac_common = "${tracs}/_common"

    file {
        "$tracs":
            ensure => directory,
            owner => root, group => 0, mode => 0755;
        "$trac_common":
            ensure => directory,
            owner => root, group => 0, mode => 0755;
        "$trac_common/templates":
            ensure => directory,
            owner => root, group => 0, mode => 0755;
        "$trac_common/templates/site.html":
            owner => root, group => 0, mode => 0644,
            source => 'puppet:///modules/sbu/trac/site.html';
        "$trac_common/templates/classbar.html":
            owner => root, group => 0, mode => 0644,
            source => 'puppet:///modules/sbu/trac/classbar.html';
    }
```

Configure all Trac instances to inherit templates from the sitewide directory set up above.

Specifically, in each trac.ini, add an inherit section if there isn't one, and set the `templates_dir` setting in that section to the common templates directory.

```
    augeas { 'trac_inherit_common_templates':
        context => '/files/var/www/tracs/*/conf/trac.ini',
        changes => [
            "setm . inherit '' ",
            "setm inherit templates_dir '$trac_common/templates'",
        ],
    }
}
```

```
class sbu::vagrant(
    $mode = 'development',
    $builder = 'vagrant')
{
    class { 'sbu::server':                                          §11.89.3
        mode => $mode,
        builder => $builder,
        cert_nickname => $::hostname,
    }
    pki::nss::self_signed { "${sbu::server::dbdir}:${::hostname}":     §11.76.12
        sqlite => false,
        noise_file => '/vagrant/insecure_noisefile',
    }
    file { '/etc/httpd/conf.d/Data.perms':
        ensure => present,
        owner => root, group => 0, mode => 0600,
        content => '',
    }
}
```

## 11.93 Serial port console support

This is the stuff necessary to make the system console go over the serial port instead of the video card and keyboard.

I've got this Cyclades ACS48 48-port *terminal server*, meaning a solid-state, special-purpose device with 48 serial ports which hooks up to a master serial port and/or a network, and serves access to the serial ports via these latter two means. Consoles of the switch and RAID shelves are already available via this terminal server; consoles of Linux servers may as well be available by the same means.

There are important security implications: access to the terminal server is mostly equivalent to physical access to the hardware in question, just like a KVM switch. The Network Infrastructure STIG may or may not effectively relegate the terminal server device to a separate management network.

There are roughly three places to set this: grub, the kernel, and the inittab. Both the grub setting and the kernel setting happen in grub's menu.lst file; see §11.40.5. The inittab would usually be set to run a getty on the serial port, so that people can log in by that means. Under RHEL6, the default configuration appears to figure out whether the kernel's console is a serial port, and if so, start a getty on it. So we needn't worry about the getty part under RHEL6.

Source: `http://tldp.org/HOWTO/Remote-Serial-Console-HOWTO/configure-boot-loader-grub.html`.

```
class serial_console($speed=9600) {
    class { 'grub::serial_console':                                  §11.40.5
        speed => $speed,
    }
```

There may be some changes necessary to the /etc/securetty file. If so they have not happened yet. See §11.84.2.

```
}
```

```
class sge::execd($sge_root, $cluster_name) {
    class { "sge::execd::${::osfamily}":                             §??
        sge_root => $sge_root,
        cluster_name => $cluster_name,
    }
}
class sge::execd::darwin($sge_root, $cluster_name) {

    mac_launchd_file { 'net.sunsource.gridengine.sgeexecd':
        description => "The GridEngine execute daemon \
runs jobs submitted by users to GridEngine.",
        environment => {
            'SGE_ROOT'          => $sge_root,
            'SGE_CELL'          => 'default',
            'SGE_ND'            => 1,
            'DYLD_LIBRARY_PATH' => "$sge_root/lib/darwin-x86",
        },
        arguments => ["$sge_root/bin/darwin-x86/sge_execd"],
    }

    service { 'net.sunsource.gridengine.sgeexecd':
        enable => true,
        ensure => running,
        require => Mac_launchd_file['net.sunsource.gridengine.sgeexecd'],
        subscribe => Mac_launchd_file['net.sunsource.gridengine.sgeexecd'],
    }

    include shell::profile_d                                         §11.94.3
    file { '/etc/profile.d/sge.sh':
        owner => root, group => 0, mode => 0644,
        content => "
export SGE_ROOT=${sge_root}
export SGE_CLUSTER_NAME=${cluster_name}
export PATH=\$SGE_ROOT/bin/darwin-x86:\$PATH
export DYLD_LIBRARY_PATH=\$SGE_ROOT/lib/darwin-x86\${DYLD_LIBRARY_PATH:+:\$DYLD_LIBRARY_PATH}
export MANPATH=\$MANPATH:\$SGE_ROOT/man
",
    }
}
```

## 11.94 Configure shells

All shell configuration is in subclasses. Keep reading.

### 11.94.1 Admin guidance regarding shells

Do not effect any policy which puts a relative path in the `PATH`, `LD_LIBRARY_PATH` or `LD_PRELOAD` environment variables. admins do GEN001840 admins do GEN001845 admins do GEN001850

## 11.94.2 Environment modules

Install environment modules, as found at `http://modules.sourceforge.net/`.

```
class shell::env_modules($initial_modulepath) {
    include "shell::env_modules::${::osfamily}"
    shell::profile_d::sh_entry { 'before_modules':
        content => inline_template("export MODULEPATH=\
<%= @initial_modulepath.join(':')%>
"),
    }
}
```

§11.94.3

### Env modules under Mac OS X

```
class shell::env_modules::darwin {
    warning 'unimplemented on darwin'
}
```

### Env modules under RHEL

```
class shell::env_modules::redhat {
    package { 'environment-modules':
        ensure => present,
    }
}
```

## 11.94.3 profile.d permissions

Set permissions for "global initialization files" according to the UNIX SRG.

```
class shell::global_init_files {
```

Make sure that no one can influence the environment variables set when the shell starts, except for root.

auto: ECLP-1
auto: GEN001720
auto: GEN001740
auto: GEN001760

On the Mac, `/etc/profile.d` is not a usual place for global initialization files, but we put it there.

auto: ECLP-1
auto: GEN001720 M6
auto: GEN001740 M6
auto: GEN001760 M6

```
    $glif_owner = $::osfamily ? {
        ’redhat’  => bin,
        ’darwin’  => root,
        default => root,
    }
    File {
        owner => $glif_owner,
        group => 0,
        mode => 0444,
    }
    file {
        "/etc/profile.d":
            ensure => directory,
            recurse => true, recurselimit => 2;
        "/etc/profile": ensure => present;
        "/etc/bashrc":;
        "/etc/csh.login":;
        "/etc/csh.logout":;
        "/etc/csh.cshrc":;
    }
```

auto: ECLP-1
auto: GEN001730

Remove extended ACLs on shell startup files.

```
    no_ext_acl {
        "/etc/profile.d": recurse => true;
        "/etc/profile":;
        "/etc/bashrc":;
        "/etc/csh.login":;
        "/etc/csh.logout":;
        "/etc/csh.cshrc":;
    }
}
class shell::profile_d {
```

Make sure the `profile.d` directory exists.

```
    require shell::global_init_files
    exec { "use profile.d":
        path => [’/bin’, ’/usr/bin’],
        command => "sed -i .before_profile_d -e ’\$a\\
for i in /etc/profile.d/*.sh; do\\
\\    if [ -r \"\$i\" ]; then\\
\\        . \"\$i\"\\
\\    fi\\
done\\
’ /etc/profile",
        unless => "grep -- ’if \\[ -r \"\\\$i’ /etc/profile",
    }
}
define shell::profile_d::csh_entry($content) {
    include shell::profile_d
```

§11.94.3

```
       file { "/etc/profile.d/${name}.csh":
           owner => root, group => 0, mode => 0444,
           content => $content,
       }
   }
   define shell::profile_d::sh_entry($content) {
       include shell::profile_d                                          §11.94.3
       file { "/etc/profile.d/${name}.sh":
           owner => root, group => 0, mode => 0444,
           content => $content,
       }
   }
```

### 11.94.4 STIG-required shell configuration

```
class shell::stig {
    File {
        ensure => present,
        owner => root, group => 0, mode => 0644,
    }
```

Don't let users `write` each other, because "messaging can be used to cause a denial-of-service attack." auto: ECSC-1 auto: GEN001780

```
       file { "/etc/profile.d/mesg.sh":
           content => "mesg n\n",
       }
       file { "/etc/profile.d/mesg.csh":
           content => "mesg n\n",
       }
```

Make sure the `/etc/shells` file exists and has controlled contents. auto: ECSC-1 auto: GEN002120

The contents are specified here, because the ensuing requirements apply to each shell. If you add a shell to the contents of `/etc/shells` here, you must add corresponding policy below.

```
       $valid_shells = $::osfamily ? {
           'redhat' => "/bin/sh
   /bin/bash
   /sbin/nologin
   /bin/tcsh
   /bin/csh
   /bin/zsh
   ",
           'darwin' => "/bin/bash
   /bin/csh
   /bin/ksh
   /bin/sh
   /bin/tcsh
   /bin/zsh
   ",
           default  => unimplemented,
       }
```

```
    file { "/etc/shells":
        ensure => present,
        owner => root, group => 0, mode => 0644,
        content => $valid_shells,
    }
```

Make sure that all shells listed in /etc/passwd are listed in /etc/shells. (This script will change any which are not listed to /sbin/nologin.)

auto: ECSC-1
auto: GEN002140

```
    cron::daily { 'valid-shells':
        source => "puppet:///modules/shell/valid-shells",
    }
```

§11.24.3

Control ownership and permissions of shell executables.

The STIGs say 0755, but we use 0555 here; it is more restrictive and comports with default Mac configuration.

auto: ECLP-1
auto: GEN002200 M6
auto: GEN002220 M6
auto: ECLP-1
auto: GEN002200
auto: GEN002210
auto: GEN002220

```
    file {
        "/bin/sh": owner => root, group => 0, mode => 0555;
        "/bin/bash": owner => root, group => 0, mode => 0555;
        "/sbin/nologin": owner => root, group => 0, mode => 0555;
        "/bin/tcsh": owner => root, group => 0, mode => 0555;
        "/bin/csh": owner => root, group => 0, mode => 0555;
        "/bin/ksh": owner => root, group => 0, mode => 0555;
        "/bin/zsh": owner => root, group => 0, mode => 0555;
    }
```

Remove extended ACLs on shell executables.

auto: ECLP-1
auto: GEN002230 M6
auto: ECLP-1
auto: GEN002230

```
    no_ext_acl {
        "/bin/sh":;
        "/bin/bash":;
        "/sbin/nologin":;
        "/bin/tcsh":;
        "/bin/csh":;
        "/bin/ksh":;
        "/bin/zsh":;
    }

    include shell::global_init_files
}
```

§11.94.3

### 11.94.5 Set default umask

Set the system default umask to 077, so that by default files are only accessible by the user who created them.

auto: ECCD-1
auto: GEN002560

```
class shell::umask {
    define make_umasks_077_in() {
        exec { "umask_077_in_${name}":
            command => "sed -i -e \
                ’s/\\(^[[:space:]]*umask\\>\\).*/\\1 077/’ \
                ${name}",
            onlyif => "grep ’^[[:space:]]*umask’ ${name} | \
                       grep -v ’umask 077\$’",
        }
    }
    make_umasks_077_in {
        ’/etc/profile’:;
        ’/etc/bashrc’:;
        ’/etc/csh.cshrc’:;
    }
}
```

## 11.95 Control access to single-user mode

Different operating systems do this differently; so first we must pick an implementation.

Control access to single-user mode, so that “system initialization” and “shutdown... are configured to ensure that the system remains in a secure state.” auto: DCSS-1

```
class single_user {
    case $osfamily {
        RedHat: {
            case $operatingsystemrelease {
                /^6.*/: {
                    include single_user::rhel6                §11.95.2
                }
                /^5.*/: {
                    include single_user::rhel5                §11.95.1
                }
                default: { unimplemented() }
            }
        }
```

Under Mac OS X, single-user mode access is controlled by a boot password, which must be set from a utility which is run from the Mac OS X install disk. This cannot be automated. admins do DCSS-1

```
        Darwin: {}
        default: { unimplemented() }
    }
}
```

### 11.95.1 Securing single-user mode under RHEL5

Require authentication for access to single-user mode. auto: IAIA-1 auto: GEN000020

```
class single_user::rhel5 {
```

Require authentication for access to single-user mode. auto: IAIA-1 auto: GEN000020

```
    augeas { "single_user":
        context => "/files/etc/inittab",
        changes => [
            "set ~/runlevels S",
            "set ~/action wait",
            "set ~/process /sbin/sulogin",
        ],
    }
```

Also disallow hotkey interactive startup, where the user at the console gets to say which services start or not.

```
    augeas { "single_user_stepwise_init":
        context => "/files/etc/sysconfig/init",
        changes => "set PROMPT no",
    }

}
```

### 11.95.2 Securing single-user mode under RHEL6

RHEL6 uses Upstart as its init.

```
class single_user::rhel6 {
    augeas { "single_user":
        context => "/files/etc/sysconfig/init",
        changes => [
```

Require authentication for access to single-user mode. auto: IAIA-1

```
            "set SINGLE /sbin/sulogin",
```

auto: GEN000020

As interactive startup (opportunity to say whether each service will start) seems like a “maintenance mode,” we’ll disable it here.

```
                "set PROMPT no",
        ],
    }
}
```

## 11.96 Smartcards

Configure smart card drivers and support.

Application-specific settings may also be necessary.

```
class smartcard {
    case $::osfamily {
        'RedHat': {
            package { ['pcsc-lite', 'coolkey']:
                ensure => present,
            }
        }
        'Darwin': {
            case $::macosx_productversion_major {
                '10.6': {
                    mac_package { 'OpenSC-0.12.2-10.6-1.dmg':
                        ensure => installed,
                    }
                }
                '10.9': {
                    mac_package { 'OpenSC-0.12.2-10.9hack.dmg':
                        ensure => installed,
                    }
                }
                default: { unimplemented() }
            }
        }
        default: { unimplemented() }
    }
}
```

## 11.97 SMTP

Configure SMTP properly. This whole module is presently RHEL6-specific and Postfix-specific.

The default RHEL aliases file does not contain any entries which execute programs.

We use postfix, not sendmail.

RHEL6 logs all mail server messages by default.

Postfix does not recognize the SMTP HELP command.

Postfix under default RHEL6 settings does not divulge its version in its greeting.

Red Hat provides up-to-date SMTP servers.

We use postfix, not sendmail.

Postfix does not recognize the SMTP EXPN command.

Postfix does not provide any information in response to an SMTP VRFY request.

Postfix does not recognize the SMTP WIZ command.

Postfix under default RHEL6 settings accepts email only from the local system. This policy does not change this default.

RHEL5, RHEL6: GEN004400
RHEL5, RHEL6: GEN004410
RHEL5, RHEL6: GEN004420
RHEL5, RHEL6: GEN004430
N/A: GEN004440
RHEL6: GEN004460
RHEL6: GEN004540
RHEL6: GEN004560
RHEL6: GEN004600
N/A: GEN004620
RHEL6: GEN004660
RHEL6: GEN004680
RHEL6: GEN004700
RHEL6: GEN004710

```
class smtp {
```

When the aliases file has changed, run newaliases. Our edits using Augeas will notify this exec resource.

```
        exec { "newaliases":
            command => "/usr/bin/newaliases",
            refreshonly => true,
        }
```

Control ownership of the SMTP log. (Permissions and ACLs are controlled by §11.56.6.)

auto: ECLP-1
auto: GEN004480

```
        file { "/var/log/maillog": owner => root }
    }
```

### 11.97.1 Admin guidance regarding SMTP

Do not add any entries to the aliases file which execute programs.

admins do GEN004400
admins do GEN004410
admins do GEN004420
admins do GEN004430

### 11.97.2 Postfix

The postfix service should be reloaded when mail configuration is changed.

```
    class smtp::postfix {
        service { "postfix":
            restart => "/sbin/service postfix reload",
        }
    }
```

### 11.97.3 Mail sent to root

Set the place where root's mail goes to. Any service which discovers programmatically something the human administrator should know will email root, so this should point at a real and capable human. (Examples include cron, when output happens, and auditd, when disk space for audit logs runs low.)

Example usage:

```
      smtp::root { "the.real.admin.ctr@example.com": }
```

```
    define smtp::root() {
        include smtp
```

§11.97

In both cases below we are editing the aliases file. If we change it, we need to run newaliases.

```
        Augeas {
            context => "/files/etc/aliases",
            notify => Exec['newaliases'],
        }
        augeas {
```

If there are multiple root entries in the aliases file, delete them: we can't properly edit them.

```
            "aliases_delete_multiple_roots":
                onlyif => "match *[name='root'] size > 1",
                changes => "rm *[name='root']";
```

If there is one root entry in the aliases file, make sure it has the right value.

```
        "aliases_set_root":
            onlyif => "match *[name='root'] size == 1",
            changes => "set *[name='root']/value '${name}'";
```

If there is no root entry in the aliases file, add one with the right value.

```
        "aliases_add_root":
            onlyif => "match *[name='root'] size == 0",
            changes => [
                "ins 100000 after *[last()]",
                "set 100000/name root",
                "set 100000/value '${name}'",
            ];
    }
}
```

### 11.97.4 Sendmail

When sendmail configuration changes, we must regenerate the real configuration, then reload sendmail.

```
class smtp::sendmail {
    service { "sendmail":
        restart => "/sbin/service sendmail reload",
    }

    package { 'sendmail-cf':
        ensure => installed,
    }
    require common_packages::make

    exec { 'update_sendmail_config':
        command => 'make -C /etc/mail',
        require => [
            Package['sendmail-cf'],
            Package['make'],
            ],
        refreshonly => true,
        notify => Service['sendmail'],
    }
}
```

### 11.97.5 SMTP smarthosts

Configure a host to be a "smarthost," that is, to take on all SMTP delivery duties for some other hosts.

```
class smtp::smarthost {
    case $::osfamily {
        'RedHat': {
            case $::operatingsystemrelease {
                /^6\..*/: {
                    class { "${name}::postfix":
```

§??

```
                    }
                }
                default: { unimplemented() }
            }
        }
        default: { unimplemented() }
    }
}
```

### 11.97.6 STIG-required mail configuration

```
class smtp::stig {
    include smtp
```

§11.97

Disable the decode alias. auto: ECSC-1

Even though the comment that comes above this in the stock configuration ("trap decode to catch security attacks") indicates that it may be positive to leave it uncommented, the STIG specifies that it must be deleted or commented out, and does not discuss further. auto: GEN004640

```
    augeas { "remove_decode_alias":
        context => "/files/etc/aliases",
        changes => [
            "rm *[name='decode']",
```

Go ahead and remove that comment too.

```
            "rm #comment[. =~ regexp('trap decode.*')]",
        ],
        notify => Exec['newaliases'],
    }
```

Control ownership and permissions of the aliases file. auto: ECLP-1

```
    file { "/etc/aliases":
        owner => root, group => 0, mode => 0644,
    }
```

auto: GEN004360
auto: GEN004370
auto: GEN004380

Remove extended ACLs on the aliases file. auto: ECLP-1

```
    no_ext_acl { "/etc/aliases": }
```

auto: GEN004390

```
    case $::osfamily {
        'RedHat': {
            case $::operatingsystemrelease {
                /^6\..*/: {
```

Configure the mail server to ignore .forward files. (See also §11.41.3.) auto: ECSC-1

The forward_path should really be empty, but the Augeas lens for the Postfix configuration doesn't support empty values, and it looks difficult to make it do so, and it's difficult to modify the configuration by other means. This will do. auto: GEN004580

```
                    include smtp::postfix
                    augeas { "ignore_forward_files":
                        context => "/files/etc/postfix/main.cf",
                        notify => Service['postfix'],
                        changes => "set forward_path /dev/null",
                    }
                }
                /^5\..*/: {
```

§11.97.2

```
                    include smtp::sendmail                            §11.97.4
                    $smmc = '/etc/mail/sendmail.mc'
                    $def  = "'define(`confFORWARD_PATH'\\'', `'\\'')dnl'"
                    exec { 'ignore_forward_files':
                        command => "sed -i -e '\$a '${def}  ${smmc}",
                        unless =>  "grep      '^'${def}'\$' ${smmc}",
                        notify =>  Exec['update_sendmail_config'],
                    }
                }
                default: { unimplemented() }
            }
        }
```

I don't think Mac OS X runs an SMTP server.

```
        'Darwin': {}
        default: { unimplemented() }
    }
}
```

## 11.97.7 Smart hosts

A *smart host*, or *relay host*, is a mail server through which all outgoing mail should be routed. The smart host, then, is the host that connects to a destination mail server to deliver the mail, not the host where the mail originated. This is useful in cases where the originating host is behind some sort of firewall and cannot connect to destination mail servers itself.

This is a defined resource type so that it can be exported and collected.

```
define smtp::use_smarthost() {
    case $::osfamily {
        'RedHat': {
            case $::operatingsystemrelease {
                /^6\..*/: {
                    smtp::use_smarthost::postfix { $name: }           §11.97.7
                }
                default: { unimplemented() }
            }
        }
        default: { unimplemented() }
    }
}
```

### Setting the smart host when using Postfix

```
define smtp::use_smarthost::postfix() {
       include smtp::postfix                                          §11.97.2
       augeas { "postfix use smarthost":
           context => '/files/etc/postfix/main.cf',
           changes => "set relayhost '${name}'",
           notify => Service['postfix'],
       }
   }
```

# 11.98 SNMP

## 11.98.1 Disable SNMP

We don't use SNMP on UNIX hosts (yet?). It's not merely inactive, it's not installed, so there are no default communities, users or passphrases. N/A: GEN005300

If and when SNMP is ever deployed, do not use versions 1 or 2, but only version 3 or later. N/A: GEN005305

Use FIPS 140-2 approved algorithms for SNMP. N/A: GEN005306

Being as we don't run SNMP, none of its configuration files exist. N/A: GEN005307

N/A: GEN005320
N/A: GEN005340
N/A: GEN005350
N/A: GEN005360
N/A: GEN005365
N/A: GEN005375

```
class snmp::no {
```

tog-pegasus depends on net-snmp, so it must be removed also.

```
    package { [
            'net-snmp',
            'tog-pegasus',
        ]:
        ensure => absent,
    }
}
```

# 11.99 Mac Software Update

Configure the Mac OS software updater.

## 11.99.1 Automatic software updates

### Disable automatic updates

Disable automatic software updates on the Mac. auto: ECSC-1

auto: OSX00290 M6

```
class softwareupdate::auto::no {
    mac_autoupdate { "auto": enabled => false }
}
```

# 11.100 SSH

See §11.43.2 for other SSH client-side configuration which may apply to some hosts.

```
class ssh {
    $configdir = $::osfamily ? {
        'RedHat' => '/etc/ssh',
        'Darwin' => '/etc',
        default  => unimplemented(),
    }
    $server_config = "${configdir}/sshd_config"
    $client_config = "${configdir}/ssh_config"

    $service_name = $::osfamily ? {
        'redhat' => 'sshd',
        'darwin' => 'com.openssh.sshd',
        default  => unimplemented(),
    }

    service { 'sshd':
        name => $service_name,
    }
}
```

### 11.100.1 Limit SSH connections by host IP

Configure the SSH daemon for IP filtering using TCP wrappers. auto: ECSC-1

Example: auto: GEN005540

```
ssh::allow_connect { "127.0.0.1, 192.168.0.": }
```

This is just a wrapper for `tcp_wrappers::allow`, *q.v.* (§11.106.1)

```
define ssh::allow_connect {
    tcp_wrappers::allow { "sshd":                §11.106.1
        from => $name,
    }
}
```

### 11.100.2 Limit SSH login by group membership

Restrict login via SSH to members of certain groups. auto: ECLP-1

(If any groups are listed in the `AllowGroups` directive of the `sshd` configuration, all other groups are denied login.) auto: GEN005521

Note that while this define can add a group to the AllowGroups directive, it cannot take one away. Taking some away would require knowing the entire set of them, but each `ssh::allow_group` only knows about itself. Perhaps some cunning artificer could use virtual resources to make this work right, but I'm not that person right now.

```
define ssh::allow_group() {
    include ssh                                  §11.100
    include ssh::allow_group::ins                §11.100.2
    augeas {
        "sshd_allow_group_${name}":
            require => Augeas["sshd_ins_allow_group"],
            context => "/files${ssh::server_config}",
            changes => [
                "set AllowGroups/10000 '${name}'",
            ],
            onlyif => "match AllowGroups/*[.='${name}'] \
                       size == 0";
    }
}
```

When multiple `ssh::allow_group` resources are defined, they all need this, and they cannot contain it within themselves, because then it would be repeated; and you only get to have one Augeas named `sshd_ins_allow_group`.

```
class ssh::allow_group::ins {
    augeas { "sshd_ins_allow_group":
        context => "/files${ssh::server_config}",
        changes => "ins AllowGroups after *[last()]",
        onlyif => "match AllowGroups size == 0";
    }
}
```

### 11.100.3 Set login banner

Set a banner that will be seen by people who connect via SSH, before they authenticate.

The `file` parameter must be the absolute path of a file on the client host.

§11.100

```
class ssh::banner($file) {
    include ssh
    augeas { "enable_ssh_banner":
        context => "/files${ssh::server_config}",
        changes => "set Banner /etc/issue.ssh",
        notify => Service[sshd]
    }
}
```

### 11.100.4 FIPS 140-2-required SSH configuration

§11.100

```
class ssh::fips {
    include ssh
    augeas { "sshd_fips":
        context => "/files${ssh::server_config}",
        changes => [
```

Configure the SSH server to reject SSH protocol version 1, which is no longer secure.

```
            "set Protocol 2",
```

Configure the SSH server to use only FIPS 140-2 [14] approved ciphers. According to the SRG, this presently means 3DES and AES.

Disable use of the cipher-block chaining (CBC) mode in the SSH server.

auto: DCPP-1
auto: GEN005500
auto: ECSC-1
auto: OSX00175 M6
auto: OSX8-00-00570
auto: OSX8-00-00575
auto: DCNR-1
auto: GEN005505 M6
auto: DCNR-1
auto: GEN005505
auto: ECSC-1
auto: GEN005506 M6
auto: ECSC-1
auto: GEN005506

(See http://openssh.com/txt/cbc.adv.)

```
            "set Ciphers aes128-ctr,aes192-ctr,aes256-ctr",
```

Configure the SSH server to use only FIPS 140-2 approved message authentication code (MAC) hash algorithms.

auto: DCNR-1
auto: GEN005507 M6
auto: DCNR-1
auto: GEN005507

According to the man page, the only one that looks good is hmac-sha1. Maybe with HMAC MD5 can be OK, but we won't chance it.

```
            "rm MACs",
            "set MACs/1 hmac-sha1",
        ],
        notify => Service["sshd"],
    }
```

The /etc/ssh/ssh_config file is parsed by a non-stock lens.

```
    require augeas

    augeas { "ssh_client_fips":
        context => "/files${ssh::client_config}/Host[.='*']",
        changes => [
```

Configure the SSH client not to use SSH protocol version 1, which is no longer secure.

auto: DCPP-1
auto: GEN005501

```
            "set Protocol 2",
```

Configure the SSH client to use only FIPS 140-2 approved ciphers.

Disable use of CBC mode by the SSH client.

auto: DCNR-1
auto: GEN005510 M6
auto: DCNR-1
auto: GEN005510
auto: ECSC-1
auto: GEN005511 M6
auto: ECSC-1
auto: GEN005511

```
            "rm Ciphers",
            "set Ciphers/1 aes256-ctr",
            "set Ciphers/2 aes192-ctr",
            "set Ciphers/3 aes128-ctr",
```

Configure the SSH client to use only FIPS 140-2 approved MAC hash algorithms.

auto: DCNR-1
auto: GEN005512 M6
auto: DCNR-1
auto: GEN005512

(The sshd_config lens makes the MACs setting a tree; the CMITS-custom ssh_config lens does not treat it specially. That is why this section differs from that above.)

```
            "rm MACs",
            "set MACs/1 hmac-sha1",
        ],
    }
```

If a host has FIPS compatibility configured before the sshd is first started, the sshd init script will try to generate an SSH version 1 RSA host key, and fail. We don't use SSH version 1, so that key need not be made; but the script must be changed in order not to make it, otherwise it will never progress beyond that failure to the part where the sshd actually gets started.

```
    $has_rsa1_keygen_regex =
            '^[[:space:]]*do_rsa1_keygen[[:space:]]*$'
    exec { "sshd_no_version1_keygen":
        path => ['/bin', '/usr/bin'],
        command => "sed -i \
            -e '/${has_rsa1_keygen_regex}/d' \
            /etc/init.d/sshd",
        onlyif => "grep \
            '${has_rsa1_keygen_regex}' \
            /etc/init.d/sshd",
        notify => Service["sshd"],
    }
}
```

### 11.100.5 Enable GSSAPI authentication

Where GSSAPI authentication is needed, enable it.

```
class ssh::gssapi {
```

§11.100

```
    include ssh
    augeas { "sshd_gssapi":
        context => "/files${ssh::server_config}",
        changes => [
```

Disable GSSAPI authentication in the SSH server “unless needed.” In some cases we need it.

auto: ECSC-1
auto: GEN005524

```
            "set GSSAPIAuthentication yes",
        ],
    }
```

The /etc/ssh/ssh_config file is parsed by a non-stock lens.

```
    require augeas

    augeas { "ssh_client_gssapi":
        context => "/files${ssh::client_config}/Host[.='*']",
        changes => [
```

Disable GSSAPI authentication in the SSH client “unless needed.” In some cases we need it.

auto: ECSC-1
auto: GEN005525

```
            "set GSSAPIAuthentication yes",
        ],
    }
}
```

### 11.100.6 Changes required when IPv6 is enabled

Do the opposite of ssh::no_ipv6.

```
class ssh::ipv6 {
```

§11.100

```
    include ssh
    augeas { "ssh_yes_ipv6":
        context => "/files${ssh::server_config}",
        changes => "rm AddressFamily",
    }
}
```

### 11.100.7 Disable GSSAPI authentication

Where GSSAPI authentication is not needed, disable it.

```
class ssh::no_gssapi {
    include ssh                                                    §11.100
    augeas { "sshd_no_gssapi":
        context => "/files${ssh::server_config}",
        changes => [
```

Disable GSSAPI authentication in the SSH server “unless needed.” In some cases we do not need it.

auto: ECSC-1
auto: GEN005524

```
            "set GSSAPIAuthentication no",
        ],
    }
```

The /etc/ssh/ssh_config file is parsed by a non-stock lens.

```
    require augeas

    augeas { "ssh_client_no_gssapi":
        context => "/files${ssh::client_config}/Host[.='*']",
        changes => [
```

Disable GSSAPI authentication in the SSH client “unless needed.” In some cases we do not need it.

auto: ECSC-1
auto: GEN005525

```
            "set GSSAPIAuthentication no",
        ],
    }
}
```

### 11.100.8 Changes required when IPv6 is disabled

http://groups.google.com/group/mailing.unix.openssh-dev/browse_thread/thread/8bc4833f84f05ce3, about halfway down, says that X forwarding isn't working for the person who started the thread because “Sun returns unusable return codes from getaddrinfo(3) when IPv6 is installed on the machine but no interfaces have IPv6 addresses configured. Workaround: put AddressFamily inet in sshd_config.”

```
class ssh::no_ipv6 {
    include ssh                                                    §11.100
    augeas { "ssh_no_ipv6":
        context => "/files${ssh::server_config}",
        changes => "set AddressFamily inet",
    }
}
```

### 11.100.9 Disable SSH tunnelling features

This is the subset of STIG-related SSH configuration that is odious.

```
class ssh::no_tunnelling {
    include ssh                                                    §11.100
    augeas { "sshd_no_tunnelling":
        context => "/files${ssh::server_config}",
        changes => [
```

Disallow TCP connection forwarding over SSH, because of the "risk of providing a path to circumvent firewalls and network ACLs." auto: ECSC-1 auto: GEN005515

Note that under the SRG this can be allowed if mitigated. (The sshd_config man page says, "Note that disabling TCP forwarding does not improve security unless users are also denied shell access, as they can always install their own forwarders." No reply to that from the SRG.)

```
            "set AllowTcpForwarding no",
```

Disallow gateway ports. auto: ECSC-1 auto: GEN005517

```
            "set GatewayPorts no",
```

Disallow X11 forwarding. auto: ECSC-1 auto: GEN005519

This can also be allowed if mitigated.

```
            "set X11Forwarding no",
```

Disallow `tun(4)` device forwarding. auto: ECSC-1 auto: GEN005531

(Wow, I didn't know sshd could do that. Quite cool... except now it's disabled.)

```
            "set PermitTunnel no",
        ],
        notify => Service["sshd"],
    }
```

Limit connections to a single session. auto: ECSC-1 auto: GEN005533

Lower the session limit per connection. A terminal uses a session, and so does a forwarded port or X11 connection. But RHEL5 ssh doesn't understand this directive.

```
    case $::osfamily {
        'RedHat': {
            case $::operatingsystemrelease {
                /^6\./: {
                    augeas { 'sshd_yes_tunnelling_max_sessions':
                        context => "/files${ssh::server_config}",
                        changes => 'set MaxSessions 1',
                        notify => Service['sshd'],
                    }
                }
                /^5\./: {
                    augeas { 'sshd_yes_tunnelling_max_sessions':
                        context => "/files${ssh::server_config}",
                        changes => 'rm MaxSessions',
                        notify => Service['sshd'],
                    }
                }
                default: {}
            }
        }
        default: {}
    }
```

The `/etc/ssh/ssh_config` file is parsed by a non-stock lens.

```
    include augeas
```

§11.13

```
    augeas { "ssh_client_no_tunnelling":
        context => "/files${ssh::server_config}/Host[.=’*’]",
        changes => [
```

Disallow TCP forwarding in the client. (See above.)

auto: ECSC-1
auto: GEN005516

```
            "set ClearAllForwardings yes",
```

Disallow gateway ports.

auto: ECSC-1
auto: GEN005518

```
            "set GatewayPorts no",
```

Disallow X11 forwarding. See above.

auto: ECSC-1
auto: GEN005520

```
            "set ForwardX11 no",
            "set ForwardX11Trusted no",
```

Disallow `tun(4)` device forwarding.

auto: ECSC-1
auto: GEN005532

```
            "set Tunnel no",
        ],
    }
}
```

### 11.100.10 STIG-required SSH configuration

Configure the SSH daemon to listen on addresses other than management network addresses, because it is “authorized for uses other than management” here.

auto: ECSC-1
auto: GEN005504

Either `ssh::gssapi` or `ssh::no_gssapi` must also be included for STIG compliance.

```
class ssh::stig {
    include ssh
    include ssh::fips
    include ssh::no_tunnelling
    include ssh::stig_palatable
}
```

§11.100
§11.100.4
§11.100.9
§11.100.11

### 11.100.11 Palatable STIG-compliant configuration

More than half of these settings are defaults built into OpenSSH, but if they are in the Puppet policy, we gain the guarantee of continuing compliance.

All of these settings are bearable; the unbearable ones are in §11.100.9.

```
class ssh::stig_palatable {
    include ssh
    augeas { "sshd_stig":
        context => "/files${ssh::server_config}",
        changes => [
```

§11.100

Disallow root login over `ssh`: admins must use `su` (§11.101.16) or `sudo` after logging in as themselves.

auto: ECPA-1
auto: IAIA-1
auto: GEN001020
auto: GEN001100
auto: GEN001120

```
            "set PermitRootLogin no",
```

Ignore per-user `.rhosts` and `.shosts` files.

auto: COBR-1
auto: ECPA-1
auto: OSX00165 M6
auto: OSX8-00-00565
auto: ECCD-1
auto: GEN002040

```
            "set IgnoreRhosts yes",
```

Make sure host-based authentication is not used.

(`RhostsRSAAuthentication` would need to be turned off, but it’s only valid for protocol 1 and we just forced protocol 2 above.)

auto: ECCD-1
auto: GEN002040

```
            "set HostbasedAuthentication no",
```

Disable Kerberos authentication in the SSH server "unless needed." We do not need it.

auto: ECSC-1
auto: GEN005526

```
            "set KerberosAuthentication no",
```

Don't accept any environment variables from the client.

auto: ECSC-1
auto: GEN005528

RHEL default settings only accept locale-related environment variables; our policy here is just defense in depth.

```
            "rm AcceptEnv",
```

Disallow environment settings set by the user and applied by the SSH server.

auto: ECSC-1
auto: GEN005530

Don't process requests for environment variables coming from `~/.ssh/environment` or `environment=` sections in `~/.ssh/authorized_keys`, because a malicious user could try to set `LD_PRELOAD`, causing unexpected behavior.

```
            "set PermitUserEnvironment no",
```

Cause the SSH server to ignore any user-specific files (*e.g.*, `known_hosts`, `authorized_keys`) that are not under the strict control of that user.

auto: ECLP-1
auto: GEN005536

```
            "set StrictModes yes",
```

Use OpenSSH's privilege separation feature for better security.

auto: ECLP-1
auto: GEN005537

```
            "set UsePrivilegeSeparation yes",
```

auto: ECSC-1
auto: GEN005538

```
            "set RhostsRSAAuthentication no",
```

auto: ECSC-1
auto: GEN005539

```
            "set Compression delayed",
        ],
        notify => Service["sshd"],
    }
```

The `/etc/ssh/ssh_config` file is parsed by a non-stock lens.

§11.13

```
    include augeas
    augeas { "ssh_client_stig":
        context => "/files${ssh::client_config}/Host[.='*']",
        changes => [
```

No way to disable Kerberos authentication in the stock OpenSSH client is listed in the `man` page.

N/A: GEN005527

RHEL default settings only send locale-related environment variables.

RHEL6:
GEN005529

```
        ],
    }
```

Restrict write permissions on the public SSH host keys.

auto: ECLP-1
auto: GEN005522

```
    file {
        "${ssh::configdir}/ssh_host_key.pub":
            owner => root, group => 0, mode => 0644;
        "${ssh::configdir}/ssh_host_rsa_key.pub":
            owner => root, group => 0, mode => 0644;
        "${ssh::configdir}/ssh_host_dsa_key.pub":
            owner => root, group => 0, mode => 0644;
    }
```

Restrict reading and writing permissions on the private SSH host keys.

auto: ECLP-1
auto: GEN005523

```
    file {
        "${ssh::configdir}/ssh_host_key":
            owner => root, group => 0, mode => 0600;
        "${ssh::configdir}/ssh_host_rsa_key":
            owner => root, group => 0, mode => 0600;
        "${ssh::configdir}/ssh_host_dsa_key":
            owner => root, group => 0, mode => 0600;
    }
}
```

### 11.100.12 Timeout

These settings will have the effect of kicking off clients who haven't sent data within the last ten minutes.

```
class ssh::timeout {
    include ssh
```

§11.100

```
    augeas { "sshd_timeout":
        context => "/files${ssh::server_config}",
        changes => [
```

Set the SSH server ClientAliveInterval to 600. auto: OSX8-00-00715

```
            'set ClientAliveInterval 600',
```

Set the SSH server ClientAliveCountMax to 0. auto: OSX8-00-00720

```
            'set ClientAliveCountMax 0',
            ],
        notify => Service['sshd'],
    }
}
```

### 11.100.13 No timeout

Where the timeout cannot be implemented, include this class.

```
class ssh::timeout::no {
    include ssh
```

§11.100

```
    augeas { "sshd_timeout":
        context => "/files${ssh::server_config}",
        changes => [
            'rm ClientAliveInterval',
            'rm ClientAliveCountMax',
            ],
        notify => Service['sshd'],
    }
}
```

### 11.100.14 Enable useful SSH features

If we wanted to enable useful SSH features, this is how we would do it.

```
class ssh::tunnelling {
    include ssh
```

§11.100

```
    augeas { "sshd_yes_tunnelling":
        context => "/files${ssh::server_config}",
        changes => [
            "set AllowTcpForwarding yes",
```

Still disallow gateway ports.

```
            "set GatewayPorts no",
```

Allow X11 forwarding. UNIX SRG PDI GEN005519 suggests that restrictions be placed on which users can use this feature in order to mitigate the risk of enabling it.

```
            "set X11Forwarding yes",
```

Still disallow `tun(4)` device forwarding. We don’t need it.

```
            "set PermitTunnel no",
        ],
        notify => Service["sshd"],
    }
```

Raise the session limit per connection. A terminal uses a session, and so does a forwarded port or X11 connection. But RHEL5 ssh doesn’t understand this directive.

```
    case $::osfamily {
        ’RedHat’: {
            case $::operatingsystemrelease {
                /ˆ6\./: {
                    augeas { ’sshd_yes_tunnelling_max_sessions’:
                        context => "/files${ssh::server_config}",
                        changes => ’set MaxSessions 10’,
                        notify => Service[’sshd’],
                    }
                }
                /ˆ5\./: {
                    augeas { ’sshd_yes_tunnelling_max_sessions’:
                        context => "/files${ssh::server_config}",
                        changes => ’rm MaxSessions’,
                        notify => Service[’sshd’],
                    }
                }
                default: {}
            }
        }
        default: {}
    }
```

The `/etc/ssh/ssh_config` file is parsed by a non-stock lens.

```
    require augeas

    augeas { "ssh_client_no_tunnelling":
        context => "/files${ssh::client_config}/Host[.=’*’]",
        changes => [
```

Allow TCP forwarding in the client.

```
            "set ClearAllForwardings no",
```

Still disallow gateway ports.

```
            "set GatewayPorts no",
```

Allow X11 forwarding. Trusted is riskier and we don't need it.

```
            "set ForwardX11 yes",
            "set ForwardX11Trusted no",
```

Still disallow tun(4) device forwarding.

```
            "set Tunnel no",
        ],
    }
}
```

## 11.101 Miscellaneous STIG requirements

STIG-related configuration that has to do with sizable subsystems is placed under those subsystems; this section contains policies which are simple, small, and unlikely to interfere with any site-specific configuration.

```
class stig_misc {
    include stig_misc::host_based_authn
    case $::osfamily {
        'RedHat': {
```

§11.101.4

Prevent unencrypted terminal access by uninstalling rsh and telnet.

auto: IAIA-1
auto: GEN001100

```
            include rsh::no
            include telnet::no
```

§11.87.1
§11.107.1

Remove the finger server.

auto: DCPP-1
auto: GEN003860

```
            package {
                "finger-server": ensure => absent;
            }
```

The STIG requires to limit users to 10 simultaneous logins. Many people here, including Jared, run more than 10 xterms routinely, each of which is a "login"; logging in using ssh fails if the maximum logins are not set high enough. GEN000450

```
            class { 'pam::max_logins':
                limit => 30,
            }
```

§11.74.3

Make the system delay at least 4 seconds following a failed login.

auto: ECLO-1
auto: GEN000480

```
            class { 'pam::faildelay':
                seconds => 4,
            }
```

§11.74.1

```
            include stig_misc::login_history
            include stig_misc::permissions
            include stig_misc::startup_files
            include stig_misc::system_files
            include stig_misc::library_files
            include stig_misc::man_page_files
            include stig_misc::skel
            include stig_misc::xinetd
            include stig_misc::run_control_scripts
            include stig_misc::device_files
```

§11.101.6
§11.101.8
§11.101.12
§11.101.13
§11.101.5
§11.101.7
§11.101.11
§11.101.16
§11.101.9
§11.101.1

```
            include stig_misc::find_uneven                    §11.101.2
            include stig_misc::world_writable                 §11.101.15
        }
```

The Mac OS X STIG stuff is all taken care of elsewhere.

```
        'Darwin': {}
        default: { unimplemented() }
    }
}
```

### 11.101.1 Device files

Check for extraneous device files at least weekly. auto: ECSC-1 auto: GEN002260

It appears on RHEL6 that /dev is on a different filesystem from /, so using the -xdev switch, in addition to excluding NFS filesystems, excludes /dev, with the happy result that any device files found by this command are extraneous, so no further filtering is necessary.

```
class stig_misc::device_files {
    file { "/etc/cron.weekly/device-files.cron":
        owner => root, group => 0, mode => 0700,
        source => "puppet:///modules/stig_misc/\
device_files/device-files.cron",
    }
}
```

### 11.101.2 Uneven access permissions

Check for system files and directories having "uneven access permissions." auto: ECCD-1 auto: GEN001140 auto: ECCD-1 auto: GEN001140 M6

```
class stig_misc::find_uneven {

    $system_dirs = "/etc /bin /usr/bin /sbin /usr/sbin"
```

Because the exec to find uneven permissions is long and we need to do it three times, we define a resource type to do it.

Usage:

```
  _log_uneven { 'bla_bla_title':
        bit => '4',
        paths => ['/bin', '/usr/bin', '/etc'],
  }
```

The effect of the above is that if files with uneven read permissions exist (because read is the 4 bit in the mode of a directory entry, see chmod(1)) in /bin, /usr/bin, or /etc, the names of these files will be logged as errors.

```
    define log_uneven($bit, $paths) {
        exec { "log_uneven_permissions_${name}":
            path => ['/bin', '/usr/bin'],
            logoutput => true,
            loglevel => err,
```

The two clauses here find (1) files having the bit for the group but not for the user, and (2) files having the bit for other but not for the user.

```
                command => "find ${paths} \
                    -perm -0${bit}0 \\! -perm -${bit}00 -ls -o \
                    -perm -00${bit} \\! -perm -${bit}00 -ls",
```

In order to avoid having err-level log messages only stating "executed successfully," we only execute the command above if it would produce any output.

```
                onlyif => "find ${paths} \
                    -perm -0${bit}0 \\! -perm -${bit}00 -ls -o \
                    -perm -00${bit} \\! -perm -${bit}00 -ls | \
                    grep . >&/dev/null",
            }
        }
```

And now we use our defined resource type.

```
        log_uneven { 'system_files_read':
            bit => '4',
            paths => $system_dirs,
        }
        log_uneven { 'system_files_write':
            bit => '2',
            paths => $system_dirs,
        }
        log_uneven { 'system_files_execute':
            bit => '1',
            paths => $system_dirs,
        }
    }
```

### 11.101.3 "Unowned" files

Check for files and directories with unknown owners.

auto: ECCD-1
auto: ECSC-1

We assume here that any NFS filesystem which may be mounted will be under `/net`. If that assumption does not hold, we'll end up searching across an NFS filesystem. That could take a while and spit out a bunch of errors.

auto: GEN001160
auto: GEN001170
auto: ECCD-1
auto: ECSC-1
auto: GEN001160 M6
auto: GEN001170 M6

```
    class stig_misc::find_unowned {
        exec { 'files_with_unknown_owner_or_group':
            path => ['/bin', '/usr/bin'],
            command => "find / -path /net -prune -o \
                    -nouser -ls -o \
                    -nogroup -ls",
            logoutput => true,
            loglevel => err,
        }
    }
```

### 11.101.4 Disable host-based authentication

```
class stig_misc::host_based_authn {
```

Remove `hosts.equiv` and `shosts.equiv` files.

auto: ECCD-1
auto: GEN002040

```
    file { "/etc/hosts.equiv": ensure => absent }
    file { "/etc/shosts.equiv": ensure => absent }
}
```

## 11.101.5 Library files

```
class stig_misc::library_files {
```

Lock down permissions for "library files." auto: DCSL-1

```
    $library_dirs = $::osfamily ? {                                  auto: GEN001300 M6
        'darwin' => [ '/System/Library/Frameworks',
                      '/Library/Frameworks',
                      '/usr/lib',
                      '/usr/local/lib' ],
        'redhat' => [ '/lib', '/lib64',
                      '/usr/lib', '/usr/lib64',
                      '/usr/local/lib', '/usr/local/lib64' ],
        default  => [ '/usr/lib', '/usr/local/lib' ],
    }
    file { $library_dirs:
        mode => go-w,
    }
```

Remove any extended ACLs from library files. auto: ECLP-1

```
    no_ext_acl { $library_dirs: recurse => true }                    auto: GEN001310 M6
}                                                                    auto: ECLP-1
                                                                     auto: GEN001310
```

## 11.101.6 Show login history

When a user logs in, show the date and time of the user's last successful login, and the number of unsuccessful login attempts since the last successful login. auto: ECSC-1 auto: GEN000452 auto: GEN000454

It appears that these requirements are also lodged by AFMAN 33-223.

```
class stig_misc::login_history {
    include stig_misc::login_history::console                        §11.101.6
    include stig_misc::login_history::gdm                            §11.101.6
}
```

### At the console

For this we use `pam_lastlog.so`.

```
class stig_misc::login_history::console {
```

First make sure that `pam_lastlog` is called by the PAM configuration.

```
  augeas { "pam_lastlog_insert":
      context => "/files/etc/pam.d/system-auth",
      onlyif => "match *[type='session' and \
                         module='pam_lastlog.so'] \
                 size == 0",
      changes => [
          "insert 999 after *[type='session'][last()]",
          "set 999/type session",
          "set 999/control required",
          "set 999/module pam_lastlog.so",
      ],
  }
```

Now—set its parameters.

```
  augeas { "pam_lastlog_parameters":
      context => "/files/etc/pam.d/system-auth/*[\
               type='session' and \
               module='pam_lastlog.so']",
      changes => [
          "rm argument",
          "set argument showfailed",
      ]
  }
}
```

### At the GDM login

```
class stig_misc::login_history::gdm {
  if($gdm_installed == 'true') {
      include zenity
      package { "loginhistory": ensure => present, }
      file { "/etc/gdm/PostLogin/Default":
          require => Package["zenity"],
          owner => root, group => 0, mode => 0755,
          ensure => present,
          source => "puppet:///modules/stig_misc/\
  login_history/gdm-post-login.sh",
      }
    }
  }
```

§11.118.3

## 11.101.7 Manual page file permissions

```
class stig_misc::man_page_files {
```

Lock down permissions for manual page files.

(There are so many of these that specifying policy for them using the file resource type ran into speed and memory problems.)

```
    $man_page_dirs = ['/usr/share/man']
```

auto: ECCD-1
auto: GEN001280
auto: ECCD-1
auto: GEN001280 M6

We use the `-perm +` syntax for `find` even though it is deprecated by GNU find, because Mac OS X's `find` doesn't understand the recommended `-perm /` syntax.

```
        exec { "chmod_man_pages":
            path => ['/bin', '/usr/bin'],
            command => "chmod -c -R go-w ${man_page_dirs}",
            onlyif  => "find ${man_page_dirs} \
                    \\! -type l -perm +022 | \
                grep . >&/dev/null",
            logoutput => true,
        }
        exec { "chown_man_pages":
            path => ['/bin', '/usr/bin'],
            command => "chown -c -R root:0 ${man_page_dirs}",
            onlyif  => "find ${man_page_dirs} \
                    \\! -user root -o \\! -group 0 | \
                grep . >&/dev/null",
            logoutput => true,
        }
```

Remove any extended ACLs from manual page files.

```
        no_ext_acl { "/usr/share/man": recurse => true }
    }
```

auto: ECLP-1
auto: GEN001290
auto: ECLP-1
auto: GEN001290 M6

### 11.101.8 Miscellaneous STIG-required file permission policies

Set sane permissions in various parts of the system which don't need configuration otherwise.

```
    class stig_misc::permissions {
```

Control ownership and permissions of `resolv.conf`.

```
        file { "/etc/resolv.conf":
            owner => root, group => 0, mode => 0644,
        }
```

Remove extended ACLs on `resolv.conf`.

```
        no_ext_acl { "/etc/resolv.conf": }
```

Control ownership and permissions of the `hosts` file.

```
        file { "/etc/hosts":
            owner => root, group => 0, mode => 0644,
        }
```

Remove extended ACLs on the `hosts` file.

```
        no_ext_acl { "/etc/hosts": }
```

Control ownership and permissions of `nsswitch.conf`.

```
        file { "/etc/nsswitch.conf":
            owner => root, group => 0, mode => 0644,
        }
```

Remove extended ACLs on `nsswitch.conf`.

```
        no_ext_acl { "/etc/nsswitch.conf": }
```

Control ownership and permissions of the `passwd` file.

```
        file { "/etc/passwd":
            owner => root, group => 0, mode => 0644,
        }
```

Remove extended ACLs on the `passwd` file.

```
        no_ext_acl { "/etc/passwd": }
```

auto: ECLP-1
auto: GEN001362 M6
auto: GEN001363 M6
auto: GEN001364 M6
auto: ECLP-1
auto: GEN001362
auto: GEN001363
auto: GEN001364
auto: ECLP-1
auto: GEN001365 M6
auto: ECLP-1
auto: GEN001365
auto: ECLP-1
auto: GEN001366 M6
auto: GEN001367 M6
auto: GEN001368 M6
auto: ECLP-1
auto: GEN001366
auto: GEN001367
auto: GEN001368
auto: ECLP-1
auto: GEN001369 M6
auto: ECLP-1
auto: GEN001369
auto: ECLP-1
auto: GEN001371
auto: GEN001372
auto: GEN001373
auto: ECLP-1
auto: GEN001374
auto: ECLP-1
auto: GEN001378 M6
auto: GEN001379 M6
auto: GEN001380 M6
auto: ECLP-1
auto: GEN001378
auto: GEN001379
auto: GEN001380
auto: ECLP-1

Control ownership and permissions of the `group` file.

auto: ECLP-1

```
file { "/etc/group":
    owner => root, group => 0, mode => 0644,
}
```

auto: GEN001391 M6
auto: GEN001392 M6
auto: GEN001393 M6

Remove extended ACLs on the `group` file.

auto: ECLP-1

```
no_ext_acl { "/etc/group": }
```

auto: GEN001391
auto: GEN001392
auto: GEN001393

Control ownership and permissions of the `shadow` file.

auto: ECLP-1

```
file { "/etc/shadow":
    owner => root, group => 0, mode => 0400,
}
```

auto: GEN001394 M6
auto: ECLP-1
auto: GEN001394

Remove extended ACLs on the `shadow` file.

```
no_ext_acl { "/etc/shadow": }
```

auto: ECLP-1
auto: GEN001400
auto: GEN001410
auto: GEN001420

Remove extended ACLs on sound device files.

```
no_ext_acl {
    "/dev/dsp":;
    "/dev/audio":;
    "/dev/mixer":;
    "/dev/sequencer":;
    "/dev/snd": recurse => true;
}
```

auto: ECLP-1
auto: GEN001430
auto: ECLP-1
auto: GEN002330

Make sure unprivileged users cannot remove devices. Device file permissions are “as configured by the vendor:” only “device files specifically intended to be world-writable” are world-writable.

auto: ECCD-1
auto: ECLP-1
auto: GEN002280 M6

```
file { ’/dev’:
    owner => root, group => 0, mode => o-w,
}
```

auto: ECLP-1
auto: GEN000000-LNX001431
auto: GEN000000-LNX001432
auto: GEN000000-LNX001433

```
file { "/etc/gshadow":
    owner => root, group => 0, mode => 0400,
}
no_ext_acl { "/etc/gshadow": }
```

auto: ECLP-1
auto: GEN000000-LNX001434

auto: ECLP-1
auto: GEN000000-LNX00400
auto: GEN000000-LNX00420
auto: GEN000000-LNX00440

```
file { "/etc/security/access.conf":
    owner => root, group => 0, mode => 0640,
}
no_ext_acl { "/etc/security/access.conf": }
```

auto: ECLP-1
auto: GEN000000-LNX00450

```
}
```

## 11.101.9 Secure run control scripts

```
class stig_misc::run_control_scripts {
```

Restrict permissions on the run control scripts.

auto: ECLP-1
auto: GEN001580 M6

Restrict ownership on “system start-up files.”

auto: ECLP-1
auto: GEN001580

What constitutes a *run control script* is defined by implication in the check content of various STIGs. Confusingly enough, the RHEL 5 STIG check content implies that for that STIG, “run control scripts” and “system start-up files” are the same files.

auto: ECLP-1
auto: GEN001660
auto: GEN001680

```
    $run_control_scripts = $::osfamily ? {
        ’darwin’ => [ ’/System/Library/LaunchDaemons’,
                      ’/System/Library/LaunchAgents’,
                      ’/Library/LaunchDaemons’,
                      ’/Library/LaunchAgents’ ],
        ’redhat’ => [ ’/etc/rc.d’ ],
        default  => unimplemented,
    }
    file { $run_control_scripts:
        owner => root,
```

RHEL default group owner is root for all these files.

```
        group => 0,
        mode => go-w,
        recurse => true,
        recurselimit => 3,
    }
```

Remove extended ACLs on run control scripts.

auto: ECLP-1
auto: GEN001590 M6

```
    no_ext_acl { $run_control_scripts: recurse => true }
}
```

auto: ECLP-1
auto: GEN001590

All run control scripts that come with RHEL contain only absolute paths as entries in their PATH variable settings.

RHEL5, RHEL6: GEN001600

No run control scripts that come with RHEL set the LD_LIBRARY_PATH, and it is empty by default. So, trivially, for all run control scripts, the library search paths contain only absolute paths, as required.

RHEL5, RHEL6: GEN001605

No run control scripts that come with RHEL set the LD_PRELOAD, and it is empty by default. So, trivially, for all run control scripts, the list of preloaded libraries contains only absolute paths.

RHEL5, RHEL6: GEN001610

All executables that come with RHEL are not world-writable, so it is impossible for a stock startup script to execute a world-writable program or script.

RHEL5, RHEL6: GEN001640

### 11.101.10 Admin guidance about run control scripts

Do not deploy any run control script that contains a relative path or empty entry in a PATH variable setting. You should never need to change the PATH in a run control script anyway. Similarly, never set LD_PRELOAD and never put a relative or empty entry into the LD_LIBRARY_PATH used in a run control script. Never deploy a run control script that executes a world-writable program or script. Any run control script that runs a program or script stored on an NFS share should be documented in §3.4.

admins do GEN001600
admins do GEN001605
admins do GEN001610
admins do GEN001640

As noted above, RHEL does not come with any world-writable local programs or scripts. The aide subsystem will detect any adverse permission changes; see §11.6. Do not install any world-writable programs or scripts.

### 11.101.11 Secure skel files

```
class stig_misc::skel {
```

Control ownership and permissions of skeleton files.

auto: ECLP-1
auto: GEN001800
auto: GEN001820
auto: GEN001830

```
       file { "/etc/skel":
           owner => root, group => 0, mode => 0644,
           recurse => true, recurselimit => 8,
       }
```

Remove extended ACLs from skeleton files.

auto: ECLP-1
auto: GEN001810

```
       no_ext_acl { "/etc/skel": recurse => true }
   }
```

### 11.101.12 Startup file permissions

```
class stig_misc::startup_files {
    case $osfamily {
```

The Mac OS X STIG check content and fix text fails to delineate "system start-up files" any more specifically than "every file on the root volume."

```
           'darwin': { include stig_misc::vendor_permissions }
```

The RHEL 5 STIG check content and fix text defines "system start-up files" to be the same set of files as "run control scripts."

```
           'redhat': { include stig_misc::run_control_scripts }
           default:  { unimplemented() }
       }
   }
```

### 11.101.13 System file permissions

```
class stig_misc::system_files {
```

"System accounts" and "system groups" for use in the next couple of requirements.

```
       $system_users = $osfamily ? {
           'darwin' => [ '_uucp', 'root' ],
           'redhat' => [ 'root' ],
           default  => [ 'root' ],
       }
       $system_groups = $osfamily ? {
           'darwin' => [ '_postdrop', 'admin', 'mail', 'procmod',
                        'staff', 'tty', 'wheel' ],
```

Under RHEL, all system files installed by means of RPM packages are owned by system groups—but some system groups are owned by a package, such that if the package isn't installed, the group won't exist. There's no sense creating the group where it doesn't exist, and Puppet can't deal with groups that don't exist. But Puppet doesn't mind using numerical group IDs. So we fall back on the definition of a "system group," which is a group having an identifier less than 500.

N.B. According to Puppet documentation, the default meaning when a list is given as a value for something like group ownership of a file is that any value in the list is a valid value for the group owner of the file, but if the file's group owner is found to be a value not in the list, the group owner will be set to the *first value* in the list. So it's significant that our list starts with 0, which is the `root` group under RHEL.

```
    'redhat' => [
          0,   1,   2,   3,   4,   5,   6,   7,   8,
          9,  10,  11,  12,  13,  14,  15,  16,  17,
         18,  19,  20,  21,  22,  23,  24,  25,  26,
         27,  28,  29,  30,  31,  32,  33,  34,  35,
         36,  37,  38,  39,  40,  41,  42,  43,  44,
         45,  46,  47,  48,  49,  50,  51,  52,  53,
         54,  55,  56,  57,  58,  59,  60,  61,  62,
         63,  64,  65,  66,  67,  68,  69,  70,  71,
         72,  73,  74,  75,  76,  77,  78,  79,  80,
         81,  82,  83,  84,  85,  86,  87,  88,  89,
         90,  91,  92,  93,  94,  95,  96,  97,  98,
         99, 100, 101, 102, 103, 104, 105, 106, 107,
        108, 109, 110, 111, 112, 113, 114, 115, 116,
        117, 118, 119, 120, 121, 122, 123, 124, 125,
        126, 127, 128, 129, 130, 131, 132, 133, 134,
        135, 136, 137, 138, 139, 140, 141, 142, 143,
        144, 145, 146, 147, 148, 149, 150, 151, 152,
        153, 154, 155, 156, 157, 158, 159, 160, 161,
        162, 163, 164, 165, 166, 167, 168, 169, 170,
        171, 172, 173, 174, 175, 176, 177, 178, 179,
        180, 181, 182, 183, 184, 185, 186, 187, 188,
        189, 190, 191, 192, 193, 194, 195, 196, 197,
        198, 199, 200, 201, 202, 203, 204, 205, 206,
        207, 208, 209, 210, 211, 212, 213, 214, 215,
        216, 217, 218, 219, 220, 221, 222, 223, 224,
        225, 226, 227, 228, 229, 230, 231, 232, 233,
        234, 235, 236, 237, 238, 239, 240, 241, 242,
        243, 244, 245, 246, 247, 248, 249, 250, 251,
        252, 253, 254, 255, 256, 257, 258, 259, 260,
        261, 262, 263, 264, 265, 266, 267, 268, 269,
        270, 271, 272, 273, 274, 275, 276, 277, 278,
        279, 280, 281, 282, 283, 284, 285, 286, 287,
        288, 289, 290, 291, 292, 293, 294, 295, 296,
        297, 298, 299, 300, 301, 302, 303, 304, 305,
        306, 307, 308, 309, 310, 311, 312, 313, 314,
        315, 316, 317, 318, 319, 320, 321, 322, 323,
        324, 325, 326, 327, 328, 329, 330, 331, 332,
        333, 334, 335, 336, 337, 338, 339, 340, 341,
        342, 343, 344, 345, 346, 347, 348, 349, 350,
        351, 352, 353, 354, 355, 356, 357, 358, 359,
        360, 361, 362, 363, 364, 365, 366, 367, 368,
        369, 370, 371, 372, 373, 374, 375, 376, 377,
        378, 379, 380, 381, 382, 383, 384, 385, 386,
        387, 388, 389, 390, 391, 392, 393, 394, 395,
        396, 397, 398, 399, 400, 401, 402, 403, 404,
        405, 406, 407, 408, 409, 410, 411, 412, 413,
        414, 415, 416, 417, 418, 419, 420, 421, 422,
        423, 424, 425, 426, 427, 428, 429, 430, 431,
        432, 433, 434, 435, 436, 437, 438, 439, 440,
        441, 442, 443, 444, 445, 446, 447, 448, 449,
        450, 451, 452, 453, 454, 455, 456, 457, 458,
        459, 460, 461, 462, 463, 464, 465, 466, 467,
        468, 469, 470, 471, 472, 473, 474, 475, 476,
        477, 478, 479, 480, 481, 482, 483, 484, 485,
        486, 487, 488, 489, 490, 491, 492, 493, 494,
        495, 496, 497, 498, 499 ],
    default  => [ 0 ],
}
```

Make sure all “network services daemon files” are not group- or world-writable.

auto: ECLP-1
auto: GEN001180

(The check content implies that these files are the ones under `/usr/sbin`.)

auto: ECLP-1
auto: GEN001180 M6

Make sure all “system command files” are not group- or world-writable.

auto: ECLP-1
auto: GEN001200 M6

(The check content implies that these files are the ones under `/bin`, `/sbin` and `/usr/bin`).

Make sure all “system files, programs, and directories” are owned by “a system account.”

auto: ECLP-1
auto: GEN001220 M6

(The check content implies that these files are the ones under `/bin`, `/sbin`, `/usr/bin`, and `/usr/sbin`.)

auto: ECLP-1
auto: GEN001220

Make sure all “system files, programs, and directories” are group-owned by “a system group.”

auto: ECLP-1
auto: GEN001240 M6

(The check content imples that these files, unlike the ones in the previous requirement, are only the ones under `/usr/bin`. We’ll throw the other ones in for free.)

auto: ECLP-1
auto: GEN001240

```
    file { ['/bin', '/sbin', '/usr/bin', '/usr/sbin']:
        owner => $system_users,
        group => $system_groups,
        mode => go-w,
        recurse => true,
    }
```

Remove extended ACLs on “network services daemon files.”

auto: ECLP-1
auto: GEN001190 M6

```
    no_ext_acl { '/usr/sbin':
        recurse => true,
    }
```

Remove extended ACLs on “system command files.”

auto: ECLP-1
auto: GEN001210 M6

```
    no_ext_acl { ['/bin', '/sbin', '/usr/bin']:
        recurse => true,
    }
}
```

### 11.101.14 Force permissions specified by vendors

To make sure all “system start-up files” are properly owned and group-owned on the Mac, run the disk utility to “reset the ownership to the original installation settings.”

auto: ECLP-1
auto: GEN001660 M6
auto: GEN001680 M6

“Verify system software periodically,” including the ACLs of files and their extended attributes.

auto: ECAT-1
auto: GEN006565 M6
auto: GEN006570 M6
auto: GEN006571 M6

```
class stig_misc::vendor_permissions {
    case $osfamily {
        'darwin': {
            exec { 'startup_file_permissions':
                command => "/usr/sbin/diskutil \
                            repairPermissions /",
                loglevel => warning,
            }
        }
        default: { unimplemented() }
    }
}
```

### 11.101.15 World-writable directories

```
class stig_misc::world_writable {
```

FIXME: You can tell Vagrant to use a different directory than `/tmp/vagrant-puppet`; this is just a default; but the code below hardcodes it.

```
    $exceptions = $::vagrant_puppet_provisioning ? {
        'true' => '\! -path /tmp/vagrant-puppet',
        default => '',
    }
```

Find and warn administrators about world-writable directories without the sticky bit set.

auto: ECCD-1
auto: GEN002500 M6
auto: OSX8-01120

We use `xdev` so as not to traverse onto NFS filesystems—indeed, not onto any filesystem other than the root filesystem. On Linux hosts this find may not be large enough in scope, but on Macs it should be.

```
    exec { 'find_non_sticky_world_writable':
        path => ['/bin', '/usr/bin'],
        command => "find / -xdev \
                    -type d -perm -2 \\! -perm -1000 \
                    ${exceptions} \
                    -ls",
        onlyif => "find / -xdev \
                    -type d -perm -2 \\! -perm -1000 \
                    ${exceptions} \
                    -ls  |  grep . >&/dev/null",
        logoutput => true,
        loglevel => err,
    }
```

Find and warn administrators about public directories not owned by root.

auto: ECLP-1
auto: GEN002520 M6
auto: OSX8-00-01110

```
        exec { 'find_public_non_root_owned':
            path => ['/bin', '/usr/bin'],
            command => "find / -xdev \
                        -type d -perm -1002 \\! -user root \
                        -ls",
            onlyif => "find / -xdev \
                        -type d -perm -1002 \\! -user root \
                        -ls  |  grep . >&/dev/null",
            logoutput => true,
            loglevel => err,
        }
    }
```

### 11.101.16 Disable xinetd

Disable xinetd if no services it provides are enabled.

auto: ECSC-1
auto: GEN003700

Note that the SRG does not say that xinetd must always be disabled or uninstalled: but we aren't using it on any hosts controlled by this policy yet, so might as well uninstall it.

```
    class stig_misc::xinetd {
        package { "xinetd": ensure => absent }
     #    service { "xinetd":
     #        ensure => stopped,
     #        enable => false,
     #    }
```

Other packages may install files into /etc/xinetd.d so even if xinetd is not installed we still need to ensure ownership and permissions. Note that if we start using xinetd, we'll have to secure the xinetd.conf file in addition to what's below.

Control ownership and permissions of the xinetd configuration.

auto: ECLP-1
auto: GEN003720
auto: GEN003730
auto: GEN003740
auto: GEN003750

```
        file { "/etc/xinetd.d":
            owner => root, group => 0, mode => 0440,
        }
```

Remove extended ACLs on xinetd configuration.

auto: ECLP-1
auto: GEN003745
auto: GEN003755

```
        no_ext_acl { "/etc/xinetd.d": }
```

If we remove xinetd, it doesn't matter whether it logs or traces because it doesn't do anything.

N/A: GEN003800

```
    }
```

## 11.102 su

### 11.102.1 STIG-required su configuration

UNIX SRG PDI GEN000850 requires that the system "restrict the ability to switch to the root user to members of a defined group." That defined group may vary between sites, and exactly which group it is may be a piece of FOUO information.

## 11.103 The Subversion version control system

### 11.103.1 Hook Subversion to the DoD PKI

This means trusting the DoD PKI certification authorities, and allowing the use of smartcards with Subversion.

```
class subversion::pki {
    include "subversion::pki::${::osfamily}"
    include subversion::pki::trust_cas                              §11.103.1
}
class subversion::pki::darwin {
    mac_package { 'subversion-omnibus-1.7-1.pkg':
        ensure => installed,
    }
}
class subversion::pki::redhat {
```

This part is easy: Red Hat's Subversion packages already support using your smartcard. We just have to get some middleware.

```
    include smartcard                                               §11.96
    package { 'subversion':
        ensure => present,
    }

}
```

#### Make Subversion trust the DoD PKI

```
class subversion::pki::trust_cas {
```

Make sure the CA certs are somewhere we expect.

```
    include pki::ca_certs::tls                                      §11.76.1
    require subversion::servers_config
    augeas { 'subversion_root_ca':
        context => '/files/etc/subversion/servers/global',
        changes => [
```

If you add more `ssl-authority-files`, they should be delimited by semicolons, with no spaces in between them.

```
            "set ssl-authority-files \
/etc/pki/tls/cacerts/DoD-Root2-Root.crt",
        ],
    }
}
```

#### Use CACs with Subversion

Allowing the use of smartcards with Subversion in this way is a systemwide setting, and commits this host to never using soft certificates to access a Subversion repository.

Subversion 1.7 as shipped in RHEL6 looks both in systemwide configuration (`/etc/subversion/servers`) and user-specific configuration (`~/.subversion/servers`) for settings regarding a particular server it's communicating with. The user-specific configuration overrides the systemwide configuration, *but* you can't unset something in user-specific configuration that was set in the systemwide configuration, only set it to a different value. And any value set for the `ssl-pkcs11-provider` setting means soft certificate files will not be used, but instead a PKCS#11 module will be sought. A failure to find a module so named is a failure to authenticate with a certificate. So if there is a systemwide default to use a PKCS#11 provider, there is no setting that can be written in a user's `~/.subversion/servers` that can make that user able to use soft certificates.

A patch to the software could fix this, but such a patch would never enter the upstream software, because the Subversion project has already moved on to 1.8, which does not support PKCS#11 at all. (See `http://subversion.apache.org/docs/release-notes/1.8.html#neon-deleted` and urlhttps://code.google.com/p/serf/issues/detail?id=

```
class subversion::pki::use_smartcard {

    include subversion::pki                                §11.103.1
    $pkcs11_provider = $::osfamily ? {
        'RedHat' => 'coolkey',
        'Darwin' => 'opensc-pkcs11',
        default  => unimplemented(),
    }

    require subversion::servers_config
    augeas { 'subversion_use_smartcard':
```

By using the `[global]` section for these settings, we are telling Subversion that any Subversion server that asks for a client certificate wants the one from the user's CAC. Server groups could be used to make this more specific, but so far anyone who configures a Subversion server to use client certificates has been someone who wanted to use CACs with it.

```
        context => '/files/etc/subversion/servers/global',
        changes => [
            "set ssl-pkcs11-provider ${pkcs11_provider}",
        ],
    }
}
```

### Don't necessarily use CACs with Subversion

This removes the systemwide default to use smartcards with Subversion, to enable a use case where some users on a host have soft certificates. On such a host, users who wish to use their smartcards with Subversion must write a setting for `ssl-pkcs11-provider` in their `~/.subversion/servers` file.

```
class subversion::pki::use_smartcard::no {

    include subversion::pki                                §11.103.1
```

```
        require subversion::servers_config
        augeas { 'subversion_use_smartcard':
            context => '/files/etc/subversion/servers/global',
            changes => [
                "rm ssl-pkcs11-provider",
            ],
        }
    }
```

### 11.103.2 Prepare to edit the systemwide Subversion server configuration file

```
class subversion::servers_config {
        file { '/etc/subversion':
            ensure => directory,
            owner => root, group => 0, mode => 0755,
        }

        file { '/etc/subversion/servers':
            ensure => present,
            owner => root, group => 0, mode => 0644,
        }
```

We require a custom lens because Augeas doesn't ship with one for Subversion.

```
        include augeas
    }
```

§11.13

## 11.104 sudo

The parts of this module you want to use are `sudo::allow_user` and `sudo::allow_group`. See them below. Everything else is machinery to make them happen portably.

```
    class sudo(
        $sudoers=$sudo::params::sudoers,
        $sudoers_d=$sudo::params::sudoers_d)
    inherits sudo::params {
```

As much as possible, we are writing each piece of sudo configuration in its own file. We place these files in the `$sudoers_d`.

```
        file { $sudoers_d:
            ensure => directory,
            owner => root, group => 0, mode => 0750,
        }


        case $::osfamily {
```

RHEL5 and RHEL6 both have sudo newer than 1.7.1, which is when the `#includedir` directive was added. In these cases we can just `#includedir` our `sudoers.d` directory.

```
        'RedHat': {
            augeas { 'consult_sudoers_d':
                context => "/files${sudoers}",
                incl => $sudoers,
                lens => "Sudoers.lns",
                changes => "set '#includedir' '${sudoers_d}'",
            }
        }
```

We deal with Snow Leopard in `sudo::policy_file`.

```
        'Darwin': {}
        default: { unimplemented() }
    }
}
```

### 11.104.1 Allow sudo for a group

Example usage:

```
sudo::allow_group { 'rwwgadm': }
```

```
define sudo::allow_group($run_as='ALL') {
    include sudo::auditable::policy                         §11.104.3
    sudo::auditable::for_group { $name:                     §11.104.3
        run_as => $run_as,
    }
}
```

### 11.104.2 Allow sudo for a user

Example usage:

```
sudo::allow_user { 'jenninjl': }
```

```
define sudo::allow_user($run_as='ALL') {
    include sudo::auditable::policy                         §11.104.3
    sudo::auditable::for { $name:                           §11.104.3
        run_as => $run_as,
    }
}
```

### 11.104.3 Always ask for password when sudoing

```
class sudo::always_ask {
```

The check content in the STIG says to look for these two "Defaults" lines in `/etc/sudoers`; we have written them in a file under `/etc/sudoers.d` instead. So while we are compliant, the check as it stands will fail.

Always ask for passwords when people use sudo. auto: ECSC-1

The Rule Title here does not correctly summarize what the Vulnerability Discussion, Check Content and Fix Text describe. auto: OSX00110 M6

§11.104.4

```
    sudo::policy_file { 'always_ask':
        content => "
Defaults tty_tickets
Defaults timestamp_timeout=0
",
    }
}
```

### Command aliases

This defined resource type sets up a command alias in the sudo configuration. It's quite a thin layer over the `sudoers(5)` syntax. When you see a strange-looking word written in `fixed type` in this section, look for its meaning in the man page.

The `commands` parameter is a list of `Cmnd`s.

The `type` parameter is one of `noexec`, `exec`, `setenv_noexec`, or `setenv_exec`. The meanings of these terms are to be found in `sudoers(5)` by searching for the term `Tag_Spec`.

If enable is false, the command alias will have a bang in front of its name when it is included in the configuration, with the effect that the commands given will be disallowed instead of being allowed. See `Other special characters and reserved words` in the man page.

```
define sudo::auditable::command_alias(
    $commands,
    $type='noexec',
    $enable=true,
    ) {
```

§11.104.4

```
    sudo::policy_file { "30${name}":
        content => inline_template("
Cmnd_Alias <%=@name%> = \\
    <%=[*@commands].join(', ')%>
"),
    }

    $prefixed_type = $enable ? {
        true    => $type,
        default => "DISALLOW_${type}",
    }

    require sudo::auditable::whole
    datacat_fragment { "command_alias ${name}":
        target => "sudoers.d/90auditable_whole",
        data => {
            "$prefixed_type" => [$name,],
        },
    }
}
```

```
define sudo::auditable::for(
    $run_as=’ALL’,
    $no_password=true,
) {
    $user_spec = $name
    $modifiers = $no_password ? {
        true    => ’NOPASSWD:’,
        default => ’’,
    }
    $safe_userspec =     regsubst($user_spec, ’[^a-zA-Z_]’, ’_’)
    require sudo::auditable::whole
    sudo::policy_file { "99${safe_userspec}":                          §11.104.4
        ensure => present,
        content => template("${module_name}/auditable/rule.erb"),
    }
    sudo::remove_direct_sudoers_policy { "${name}": }                  §11.104.4
}
define sudo::auditable::for_group(
    $run_as=’ALL’,
    $no_password=true,
) {
    sudo::auditable::for { "%${name}":                                 §11.104.3
        run_as => $run_as,
        no_password => $no_password,
    }
```

Remove the file that the older version of this policy put in place, if it’s there.

```
    sudo::policy_file { "${name}":                                     §11.104.4
        ensure => absent,
    }
}
```

### Basic auditable policy

The idea here is to make administrators use sudo to run each command they need, because sudo logs each command it’s run with; and prevent administrators from using sudo to run commands that are open-ended, in that they can execute more commands (which would not be logged), or to run commands that are user-written, because these can be anything.

```
class sudo::auditable::policy {
    include sudo::auditable::whole                                     §11.104.3
```

For the noexec type, we allow all local binaries, then disallow problematic ones. It’s quite important that the `LOCAL_BINARIES` directories be only writable by root, and that all their files be only writable by root.

These lists can be rather distro-specific and should be checked and changed whenever using a new distro or updating to a new major version of an existing distro.

```
    sudo::auditable::command_alias { ’LOCAL_BINARIES’:                 §11.104.3
```

```
        commands => [
            ’/bin/’,
            ’/usr/bin/’,
            ’/sbin/’,
            ’/usr/sbin/’,
            ],
    }
```

Disallow `sudo su -`.

```
    sudo::auditable::command_alias { ’SU’:                    §11.104.3
        enable => false,
        commands => [
            ’/usr/bin/su’,
            ],
    }
```

Shells can execute other things, it’s what they do all day long.

```
    sudo::auditable::command_alias { ’SHELLS’:                §11.104.3
        enable => false,
        commands => [
            ’/bin/sh’,
            ’/bin/bash’,
            ’/bin/dash’,
            ’/bin/ksh’,
            ’/bin/tcsh’,
            ’/bin/csh’,
            ’/bin/zsh’,
            ],
    }
```

Just about every editor lets you execute commands. So we disable them all and allow sudoedit instead. `rvim` and friends seem to be OK because they say that when you run them “it will not be possible to start shell commands.”

```
    sudo::auditable::command_alias { ’EDITORS’:               §11.104.3
```

```
        enable => false,
        commands => [
            '/bin/ed',
            '/bin/vi',
            '/usr/bin/ex',
            '/usr/bin/vim',
            '/usr/bin/view',
            '/usr/bin/evim',
            '/usr/bin/eview',
            '/usr/bin/gvim',
            '/usr/bin/gview',
            '/usr/bin/vimdiff',
            '/usr/bin/vimtutor',
            '/usr/bin/emacs',
            '/usr/bin/emacsclient',
            '/usr/bin/gedit',
            '/usr/bin/kwrite',
            '/usr/bin/nano',
            ],
    }
    sudo::auditable::command_alias { 'SUDOEDIT':                §11.104.3
        commands => [
            'sudoedit',
            ],
    }
```

For some reason the noexec doesn't catch this, so we prohibit it expressly.

```
    sudo::auditable::command_alias { 'RUNS_SHELL':              §11.104.3
        enable => false,
        commands => [
            '/usr/bin/tmux',
            '/usr/bin/screen',
            ],
    }
```

For some system files there are special editor wrappers; here we compel their use.

```
    sudo::auditable::command_alias { 'SPECIAL_EDITOR_WRAPPERS':  §11.104.3
        type => 'exec',
        commands => [
            '!sudoedit /etc/sudoers',
            '!sudoedit /etc/sudoers.d/*',
            '!sudoedit /etc/passwd',
            '!sudoedit /etc/group',
            '!sudoedit /etc/shadow',
            '!sudoedit /etc/gshadow',
            '/usr/sbin/visudo',
            '/usr/sbin/vipw',
            '/usr/sbin/vigr',
            ],
    }
```

Now, broadening out, we have scripts and other binaries with a legitimate need to execute subprocesses. Perhaps some of these should be listed elsewhere in this policy. That is what our defined resource type allows.

```
sudo::auditable::command_alias { ’SBIN_SCRIPTS’:          §11.104.3
    type => ’exec’,
    commands => [
        ’/sbin/dracut’,
        ’/sbin/grub-install’,
        ’/sbin/grub-md5-crypt’,
        ’/sbin/grub-terminfo’,
        ’/sbin/ifcfg’,
        ’/sbin/ifdown’,
        ’/sbin/ifup’,
        ’/sbin/mkinitrd’,
        ’/sbin/service’,
        ],
}
sudo::auditable::command_alias { ’BIN_SCRIPTS’:           §11.104.3
    type => ’exec’,
    commands => [
        ’/bin/gunzip’,
        ’/bin/zcat’,
        ’/bin/unicode_start’,
        ’/bin/unicode_stop’,
```

`mount` is not a script, but it may run a more specific mount binary, so it needs to be able to exec.

```
        ’/bin/mount’,
        ],
}
sudo::auditable::command_alias { ’USR_SBIN_SCRIPTS’:      §11.104.3
    type => ’exec’,
    commands => [
        ’/usr/sbin/gdm’,
        ’/usr/sbin/ksmtuned’,
        ’/usr/sbin/virt-what’,
        ],
}
sudo::auditable::command_alias { ’USR_BIN_SCRIPTS’:       §11.104.3
    type => ’exec’,
    commands => [
        ’/usr/bin/batch’,
        ’/usr/bin/ldd’,
        ’/usr/bin/mozilla-plugin-config’,
        ’/usr/bin/startx’,
        ’/usr/bin/reboot’,
        ’/usr/bin/halt’,
        ’/usr/bin/poweroff’,
        ],
}
sudo::auditable::command_alias { ’CRON_SCRIPTS’:          §11.104.3
```

```
        type => 'exec',
        commands => [
            '/etc/cron.hourly/',
            '/etc/cron.daily/',
            '/etc/cron.weekly/',
            '/etc/cron.monthly/',
            ],
    }
    if $::osfamily == 'RedHat' {
        sudo::auditable::command_alias { 'PACKAGE_MANAGEMENT':        §11.104.3
            type => 'exec',
            commands => [
                '/usr/bin/yum',
                '/bin/rpm',
```

/usr/bin/rhn_register is a symlink to consolehelper, “a wrapper that helps console users run system programs” (consolehelper(8)). What this means for the sudoer is that if you run sudo rhn_register, this command alias will not match it, and the one above for local binaries will, and it won't be allowed to execute subprocesses, and it won't work. But if you run sudo /usr/bin/rhn_register, it will work right.

```
                '/usr/bin/rhn_register',
```

rhnreg_ks is not allowed here, because you have to pass it a password on the command line, and that's stored in your history file, and visible to everyone logged in on the host while it's running.

```
                ],
        }
    }
}
class sudo::auditable::whole(
    $sudoers=$sudo::params::sudoers,
    $sudoers_d=$sudo::params::sudoers_d,
    ) inherits sudo::params {
```

It may be possible to use augeas instead of datacat, but as of May 2014 the Augeas sudoers lens couldn't seem to deal with aliases having items starting with bangs (!), which would prevent us from disallowing anything. Whitelisting each possible binary by name would be a sad business.

```
    datacat { "sudoers.d/90auditable_whole":
        path => "${sudoers_d}/90auditable_whole",
        template => "${module_name}/auditable/whole.erb",
        owner => root, group => 0, mode => 0440,
    } ->
    sudo::include_policy_file { "90auditable_whole":        §11.104.4
        sudoers => $sudoers,
        sudoers_d => $sudoers_d,
    }
}
```

### 11.104.4 Including policy files

RHEL 6 has sudo 1.8, which supports #includedir. To make sudo pay attention to a new file in the sudoers.d directory, we need do nothing. But Snow Leopard only has sudo 1.7.0, so we must `#include` each sudo policy file.

This defined resource type does whatever is necessary to make sudo pay attention to a file we've placed in the `sudoers.d`.

```
define sudo::include_policy_file($ensure='present', $sudoers='', $sudoers_d='') {
    require sudo
    include sudo::params                                              §11.104.4
    $d_sudoers = $sudoers ? {
        ’’      => $sudo::params::sudoers,
        default => $sudoers,
    }
    $d_sudoers_d = $sudoers_d ? {
        ’’      => $sudo::params::sudoers_d,
        default => $sudoers_d,
    }

    case $ensure {
        ’absent’: {
            case $osfamily {
                ’RedHat’: {}
                ’Darwin’: {
                    augeas { "sudoers_exclude_${name}":
                        context => "/files/${d_sudoers}",
                        incl => "${d_sudoers}",
                        lens => ’Sudoers.lns’,
                        changes => [
                            "rm #include[.=’${d_sudoers_d}/${name}’]",
                            ],
                    }
                }
                default: { unimplemented() }
            }
        }
        default: {
            case $osfamily {
                ’RedHat’: {}
                ’Darwin’: {
                    augeas { "sudoers_include_${name}":
                        context => "/files/${d_sudoers}",
                        incl => "${d_sudoers}",
                        lens => ’Sudoers.lns’,
                        changes => [
                            "set #include[last()+1] ’${d_sudoers_d}/${name}’",
                            ],
                        onlyif => "match \
                            #include[.=’${d_sudoers_d}/${name}’] size == 0",
                    }
                }
                default: { unimplemented() }
            }
        }
    }
}
class sudo::params(
    $sudoers=’/etc/sudoers’,
    $sudoers_d=’/etc/sudoers.d’,
    ) {}
```

```
define sudo::policy_file($content=’’, $ensure=’present’, $sudoers=’’, $sudoers_d=’’) {
    require sudo
    include sudo::params                                                    §11.104.4
    $d_sudoers = $sudoers ? {
        ’’      => $sudo::params::sudoers,
        default => $sudoers,
    }
    $d_sudoers_d = $sudoers_d ? {
        ’’      => $sudo::params::sudoers_d,
        default => $sudoers_d,
    }

    sudo::include_policy_file { $name:                                      §11.104.4
        ensure => $ensure,
        sudoers => $d_sudoers,
        sudoers_d => $d_sudoers_d,
    }

    file { "${d_sudoers_d}/${name}":
        ensure => $ensure,
        owner => root, group => 0, mode => 0440,
        content => $content,
    }
```

When placing a new file, we should make sure the file is in place before telling sudo to include it. When removing a file, we must make sure sudo isn’t including it before we remove the file. This is because Snow Leopard’s `sudo` segfaults if anything is wrong with its configuration as a whole, with the ... undesirable result that no one can sudo to do anything.

```
    case $ensure {
        ’present’: {
            File["${d_sudoers_d}/${name}"] ->
            Sudo::Include_policy_file[$name]
        }
        default: {
            Sudo::Include_policy_file["$name"] ->
            File["${d_sudoers_d}/${name}"]
        }
    }
}
define sudo::remove_direct_sudoers_policy() {
```

Clean out policies written directly in the sudoers file regarding this user spec.

```
    augeas { "remove_direct_sudoers_${name}":
        context => ’/files/etc/sudoers’,
        changes => "rm spec[user=’${name}’]",
    }
}
define sudo::unlimited($user_spec,$run_as=’ALL’) {
    sudo::policy_file { $name:                                              §11.104.4
        content => template(’sudo/unlimited.erb’),
    } ->
```

```
        sudo::remove_direct_sudoers_policy { $user_spec: }                §11.104.4
    }
    class sudo_user_1 {
        include sudo::auditable::whole                                     §11.104.3
        sudo::auditable::command_alias { 'EDITORS':                        §11.104.3
    commands => ['/usr/bin/vim', '/usr/bin/emacs'],
        }
        sudo::auditable::command_alias { 'SINGLE_MEMBER_ARRAY':            §11.104.3
    commands => ['/bin/true'],
            type => 'setenv_exec',
        }
        sudo::auditable::command_alias { 'SINGLE_ITEM':                    §11.104.3
    commands => '/bin/false',
        }
        sudo::auditable::command_alias { 'BAD_STUFF':                      §11.104.3
            commands => '/sbin/fdisk',
            enable => false,
        }
        sudo::auditable::for { '%luckygroup': }                            §11.104.3
    }
```

## 11.105 Swap space (virtual memory)

### 11.105.1 Encrypt swap

```
class swap::encrypt {
    include "${name}::${::osfamily}"
}
```

#### Encrypt swap on Macs

```
class swap::encrypt::darwin {
    $version_underscores = regsubst(
        $::macosx_productversion_major,
        '\D', '_', 'G')
    $klassname = "${::osfamily}_${version_underscores}"
    include "swap::encrypt::${klassname}"
}
    class swap::encrypt::darwin_10_6 {
```

"Use secure virtual memory," or in other words, make Macs encrypt their swap space. auto: ECRC-1 auto: OSX00440 M6

```
        $vm = "/Library/Preferences/com.apple.virtualMemory.plist"
```

The file may not exist; make sure it has the right ownership and permissions.

```
    file { $vm:
        ensure => present,
        owner => root, group => admin, mode => 0644,
    }
    mac_plist_value { "encrypt swap":
        require => File[$vm],
        file => $vm,
        key => 'UseEncryptedSwap',
        value => true,
    }
```

Use "secure virtual memory" on newer Macs. auto: OSX8-00-01260

```
    mac_plist_value { "un-disable swap encryption":
        require => File[$vm],
        file => $vm,
        key => 'DisableEncryptedSwap',
        value => false,
    }
}
class swap::encrypt::redhat {
    unimplemented()
}
```

### 11.105.2 STIG-required swap configuration

```
class swap::stig {
    include swap::encrypt
}
```

§11.105.1

## 11.106 TCP Wrappers

RHEL comes with TCP wrappers enabled by default. RHEL5, RHEL6: GEN006580

"The system's access control program must log each system access attempt." RHEL logs all access attempts by default. RHEL5, RHEL6: GEN006600

TCP wrappers are used within this policy solely to control SSH access. RHEL's `sshd` logs all successful and failed access attempts. This materially prevents "multiple attempts to log on to the system by an unauthorized user" from "go[ing] undetected." If we were to enable additional services using xinetd, it would also log all connection attempts by default.

Services which are not implemented on a host are not presently booby-trapped using TCP wrappers, so unauthorized users could (for example) attempt to telnet to a host repeatedly, and nothing would be logged by "the system's access control program." That would result in incoming packets which are not explicitly allowed, which would most likely be logged via other means: see §11.48.

Configure `tcp_wrappers` to grant or deny system access to specific hosts. auto: ECSC-1

Use of the `tcp_wrappers::allow` defined resource type below will "configure" TCP wrappers "with appropriate rules." auto: GEN006620

### 11.106.1 Allow some traffic through TCP wrappers

Use this like so:

```
tcp_wrappers::allow { "sshd":
    from => [
        "192.168.122.0/255.255.255.0",
        "172.16.",
    ],
}
```

In keeping with present security guidance regarding TCP wrappers, don't use hostnames in the `from` parameter, because attackers may try to poison DNS.

TCP wrappers do not appear to support CIDR notation (`192.168.122.0/24`) for IPv4 at this time.

```
define tcp_wrappers::allow($from) {
    include tcp_wrappers::default_deny                                    §11.106.2
    require tcp_wrappers::hosts_allow
```

Here follows technical discussion about the specific way we are editing the file.

According to tests in July 2013, if the *single* value of the `changes` parameter to the `augeas` resource type has newlines, each line in the value is treated as a separate command for Augeas. It's not really easy in Puppet 2.7 to take a list of values, turn it into another list of values and concatenate it to another list. But we can easily take a list and turn it into a string containing newlines using `inline_template`.

The reason this is so involved, as compared to some insert-then-change sorts of rules in the `pam` module (§11.73.4), is that an entry with only a process and no clients is not valid under the Augeas lens we are using, so you can't add the process if it doesn't exist, then set up the clients, you have to add the process and setup the clients if there's no entry, or just make sure the clients are set right if there is an entry.

The reason to avoid just nuking the entry if it exists, then recreating it, is that that operation doesn't preserve the order of the entries in the file, and so if we are allowing access to multiple services, we keep deleting and inserting lines, reshuffling the file and never leaving it alone.

If the entry doesn't exist, we need to add it—

```
    $add_entry = inline_template("
set 999/process '<%=@name-%>'
defvar n 999
")
```

If it does, we need to point our n variable at it.

```
    $ref_entry = inline_template("
defvar n *[process=’<%=@name-%>’]
")

    $already_exists_changes = inline_template("
rm \$n/client
<% if @from.is_a? Array;
      @from.each do |client_netmask|
        client, netmask = client_netmask.split(’/’) %>
set \$n/client[last()+1]         ’<%=client-%>’
<%      if netmask %>
set \$n/client[last()  ]/netmask ’<%=netmask-%>’
<%      end %>
<%    end
   else
      client, netmask = @from.split(’/’) %>
set \$n/client[last()+1]         ’<%=client-%>’
<%    if netmask %>
set \$n/client[last()  ]/netmask ’<%=netmask-%>’
<%    end %>
<% end %>
")
```

Non-stock Augeas lens may be required.

```
    require augeas

    Augeas {
        context => ’/files/etc/hosts.allow’,
    }

    augeas { "hosts_allow_add_${name}":
        changes => inline_template("
<%=@add_entry-%>
<%=@already_exists_changes-%>
"),
        onlyif => "match *[process=’${name}’] size == 0",
    }

    augeas { "hosts_allow_modify_${name}":
        changes => inline_template("
<%=@ref_entry-%>
<%=@already_exists_changes-%>
"),
        onlyif => "match *[process=’${name}’] size > 0",
    }
}
```

### 11.106.2 Deny incoming connections by default

Any incoming connections controlled by TCP wrappers, which are not explicitly allowed, should be denied.

```
   class tcp_wrappers::default_deny {
```

We don't need custom Augeas lenses here; but they are needed to write things in the `hosts.allow` file, so if we don't have them, and we write the `hosts.deny`, nothing will be allowed.

```
       require augeas
       file { "/etc/hosts.deny":
           owner => root, group => 0, mode => 0644,
           content => "# Deny by default\nALL: ALL\n";
       }
   }
   class tcp_wrappers::hosts_allow {
       file { '/etc/hosts.allow':
           ensure => present,
           owner => root, group => 0, mode => 0644,
       }
   }
```

## 11.107 Telnet

Old, unencrypted remote terminal protocol. Prohibited by UNIX SRG.

### 11.107.1 Disable Telnet

```
class telnet::no {
    include "telnet::no::${::osfamily}"
}

   class telnet::no::darwin {
```

Disable Telnet on Macs.

auto: OSX8-00-00605
auto: OSX8-00-00690
auto: OSX8-00-00695

```
       service { 'com.apple.telnetd':
           ensure => stopped,
           enable => false,
       }
   }
   class telnet::no::redhat {
       package {
```

Remove the Telnet server.

auto: DCPP-1
auto: GEN003850

```
           "telnet-server": ensure => absent;
       }
   }
```

## 11.108 Trac

This module contains the `trac_permission` custom resource type, *q.v.*, and other means of configuring Trac.

### 11.108.1 Banish a user

This defined resource type removes all special access for a user from a Trac instance. The user will end up being able to do whatever `anonymous` is allowed to do inside that Trac instance.

The name is a directory with a Trac instance in it. Example:

```
trac::banish { '/var/www/tracs/admin':
    users => ['baduser1', 'baduser2', 'baduser3'],
}
```

* * *

```
define trac::banish($users) {
    trac_permission { 'remove $users from $name':
        instance => $name,
        ensure => absent,
        subject => $users,
        action => [
            "BROWSER_VIEW", "CHANGESET_VIEW", "CONFIG_VIEW",
            "EMAIL_VIEW", "FILE_VIEW", "LOG_VIEW",
            "MILESTONE_ADMIN", "MILESTONE_CREATE",
            "MILESTONE_DELETE", "MILESTONE_MODIFY",
            "MILESTONE_VIEW", "PERMISSION_ADMIN",
            "PERMISSION_GRANT", "PERMISSION_REVOKE",
            "REPORT_ADMIN", "REPORT_CREATE", "REPORT_DELETE",
            "REPORT_MODIFY", "REPORT_SQL_VIEW", "REPORT_VIEW",
            "ROADMAP_ADMIN", "ROADMAP_VIEW", "SEARCH_VIEW",
            "TICKET_ADMIN", "TICKET_APPEND", "TICKET_CHGPROP",
            "TICKET_CREATE", "TICKET_EDIT_CC",
            "TICKET_EDIT_COMMENT", "TICKET_EDIT_DESCRIPTION",
            "TICKET_MODIFY", "TICKET_VIEW", "TIMELINE_VIEW",
            "TRAC_ADMIN", "VERSIONCONTROL_ADMIN",
            "WIKI_ADMIN", "WIKI_CREATE", "WIKI_DELETE",
            "WIKI_MODIFY", "WIKI_RENAME", "WIKI_VIEW",
        ],
    }
}
```

## 11.109 Trash

The place where you drag files you want (provisionally, anyway) to remove.

### 11.109.1 STIG-required configuration

```
class trash::stig {
    include "trash::stig::${::osfamily}"
}
```

#### On Macs

```
class trash::stig::darwin {
```

Configure the Finder to empty trash securely. auto: OSX8-00-01075 §11.61.2

```
    mcx::set { ’com.apple.finder/EmptyTrashSecurely’:
        value => true,
    }
}
class trash::stig::redhat {}
```

## 11.110 umask

The *umask* is a set of permissions to *remove* from new files being created. For example, files created by a process running with a umask of 022 will not be writable by their owning group nor everyone else. So the umask acts to provide default file permissions. It is inherited by children of a process, so it’s important to set the umask in shells and process launchers of all sorts to ensure that discretionary access controls act to provide security.

### 11.110.1 Set umasks in shell startup files

This defined resource type can make sure a umask is set properly in a file. It works if the syntax of the umask command is, e.g., `umask 077`, and if lines added to the end of the file will have the proper effect. You have to ensure the file is present yourself.

```
umask::set_in_file { ’/etc/bashrc’: umask => 077 }
```

* * *

```
define umask::set_in_file($umask) {
    $sed_i_umask = $::osfamily ? {
        ’RedHat’ => ’sed -i.before_umask’,
        ’Darwin’ => ’sed -i .before_umask’,
        default  => unimplemented(),
    }
    exec { "add umask ${umask} to ${name}":
        command => "echo ’umask ${umask}’ >> ${name}",
        unless => "grep ’^[[:space:]]*umask’ ${name}",
        path => [’/bin’, ’/usr/bin’],
        require => File[$name],
    }
    exec { "change umask to ${umask} in ${name}":
        command => "${sed_i_umask} -e \
        ’s/\\(^[[:space:]]*umask\\>\\).*/\\1 ${umask}/’ \
        ${name}",
        onlyif => "grep ’^[[:space:]]*umask’ ${name} | \
        grep -v ’umask ${umask}\$’",
        path => [’/bin’, ’/usr/bin’],
    }
}
```

### 11.110.2 Set default umask in shells

```
class umask::shell($umask) {
      Umask::Set_in_file { umask => $umask, }                        §??
      umask::set_in_file {                                           §11.110.1
          '/etc/profile':;
          '/etc/bashrc':;
          '/etc/csh.cshrc':;
      }
   }
```

### 11.110.3 STIG-required settings

```
class umask::stig {
      include umask::stig::shell                                     §11.110.3
      include "umask::stig::${::osfamily}"
   }
   class umask::stig::darwin {
```

Set the default global umask setting for user applications to 027. auto: OSX8-00-01015

```
      file { '/etc/launchd-user.conf':
          ensure => present,
          owner => root, group => 0, mode => 0644,
      }
      umask::set_in_file { '/etc/launchd-user.conf':                 §11.110.1
          umask => 027,
      }
```

Set the default global umask setting for system processes to 022. auto: OSX8-00-01020

```
      file { '/etc/launchd.conf':
          ensure => present,
          owner => root, group => 0, mode => 0644,
      }
      umask::set_in_file { '/etc/launchd.conf':                      §11.110.1
          umask => 022,
      }
   }
   class umask::stig::redhat {}
   class umask::stig::shell {
```

Set the system default umask to 077, so that by default files are only accessible by the user who created them. auto: ECCD-1 auto: GEN002560

```
      class { 'umask::shell': umask => 077 }                         §11.110.2
   }
```

## 11.111 Unowned files and directories

Fix *unowned* files and directories, defined as those whose numerical owner UID or group-owner GID do not map to a known user or group. auto: ECCD-1 auto: ECSC-1 auto: GEN001160 M6 auto: GEN001170 M6

The check content of Mac OS X STIG PDI GEN001160 M6 makes it clear that no unowned files or directories should exist anywhere on the system. But on any given UNIX workstation, some directories may be shared over a network, which makes the potential set of files to check not only uncomfortably large, but also redundant between hosts. Additionally, some of the shared directories may not be mounted in such a fashion that `root` can change the owner or group of files and directories therein, so not all hosts could fix an unowned file or directory should they come across one.

Accordingly the plan for making sure all files and directories are validly owned will vary between networks and between hosts. Classes in this module will take care of different parts of the namespace to provide the tools necessary for a complete defense against this threat.

### 11.111.1 Unowned system files

Unowned system files present the greatest threat. They are likely local to each host, and so each host should include this class.

```
class unowned::system {
    $system_dirs = ['/bin', '/sbin', '/usr/bin', '/usr/sbin']
    unowned { $system_dirs:
        owner => root,
        group => 0,
    }
}
```

## 11.112 USB (Universal Serial Bus)

"The system must have USB disabled unless needed." All of our CAC readers, and most of our keyboards and mice, connect only via USB, so it's fair to say we "need" USB. Do not disable it.

auto: ECSC-1
auto: GEN008460

### 11.112.1 Require admin password for USB storage

Prevent installation of malicious software or exfiltration of data by restricting the use of mass storage to administrators.

auto: ECSC-1
auto: GEN008480

(USB mass storage could be disabled entirely from desktop use, but admins can become root and use the mount command anyway. As long as we trust our vendor to give us correct software, there's no particular advantage in slashing a swath of nonfunctionality through the desktop.)

```
class usb::mass_storage::admin_auth {
    case $osfamily {
        RedHat: {
            case $operatingsystemrelease {

                /^6\..*/: {
    file { "/etc/polkit-1/localauthority/90-mandatory.d/\
50-mil.af.eglin.afseo.admin-udisks.pkla":
        owner => root, group => 0, mode => 0600,
        source => "puppet:///modules/usb/mass_storage/\
admin-udisks.pkla",
    }
                }

                /^5\..*/: {
    unimplemented()
                }

                default: { unimplemented() }
            }
        }
        default: { unimplemented() }
    }
}
```

### 11.112.2 Allow a group to use USB mass storage

Let members of a UNIX group use USB mass storage, without authenticating as admins.

Usage example:

```
usb::mass_storage::allow_group { "accounting": }
```

* * *

```
define usb::mass_storage::allow_group() {
    $group = $name
    case $osfamily {
        RedHat: {
            case $operatingsystemrelease {

                /^6\..*/: {
    file { "/etc/polkit-1/localauthority/90-mandatory.d/\
60-mil.af.eglin.afseo.group-${group}-udisks.pkla":
        owner => root, group => 0, mode => 0600,
        content => template("usb/mass_storage/\
group-udisks.pkla"),
    }
                }

                /^5\..*/: {
    unimplemented()
                }

                default: { unimplemented() }
            }
        }
        default: { unimplemented() }
    }
}
```

### 11.112.3 Use default USB mass storage permissions

Let the console user use USB mass storage, subject to defaults. Whenever the USB mass storage policy for a node or class is made less restrictive, you should replace the `include usb::mass_storage::bla` class with an include for this class in that context.

```
class usb::mass_storage::default {
    file { "/etc/polkit-1/localauthority/90-mandatory.d/\
50-mil.af.eglin.afseo.admin-udisks.pkla":
        ensure => absent,
    }
}
```

#### Disable USB mass storage

```
class usb::mass_storage::no {
    include "usb::mass_storage::no::${::osfamily}"
}
```

**Under the Mac OS** Remove the USB mass storage driver from Macs. auto: OSX8-00-00850

```
class usb::mass_storage::no::darwin {
    $exts = '/System/Library/Extensions'
    file { "${exts}/IOUSBMassStorageClass.kext":
        ensure => absent,
        force => true,
    }
}
class usb::mass_storage::no::redhat {
    unimplemented()
}
```

### 11.112.4 Remove a previous group allowance

Stop letting members of a UNIX group use USB mass storage without authenticating as admins.

Note that this does not explicitly disallow them: it merely undoes what `usb::mass_storage::allow_group` does. That's why this is not called `disallow_group`.

Usage example:

```
usb::mass_storage::unallow_group { "accounting": }
```

* * *

```
define usb::mass_storage::unallow_group() {
    $group = $name
    case $osfamily {
        RedHat: {
            case $operatingsystemrelease {

                /^6\..*/: {
    file { "/etc/polkit-1/localauthority/90-mandatory.d/\
60-mil.af.eglin.afseo.group-${group}-udisks.pkla":
            ensure => absent,
    }
                }

                /^5\..*/: {
    unimplemented()
                }

                default: { unimplemented() }
            }
        }
        default: { unimplemented() }
    }
}
```

## 11.113 Users

### 11.113.1 Remove unnecessary users

Remove "application accounts for applications not installed on the system." auto: IAAC-1 auto: GEN000290

The set of needed system users varies by operating system and release; so, likewise, does the set of unnecessary system users.

```
class user::unnecessary {
    case $osfamily {
        RedHat: {
            case $operatingsystemrelease {
                /^6.*/: { include user::unnecessary::rhel6 }
                /^5.*/: { include user::unnecessary::rhel5 }
                default: { unimplemented() }
            }
        }
        Darwin: {}
        default: { unimplemented() }
    }
}
```

### Under RHEL5

Here we have guidance from the Red Hat 5 STIG—specific, if unclear.

```
class user::unnecessary::rhel5 {
```

Remove the `shutdown`, `halt` and `reboot` users. The requirement says to remove "special privilege accounts" but only mentions these three.

auto: IAAC-1
auto: GEN000000-LNX00320

```
    user { ["shutdown", "halt", "reboot"]:
        ensure => absent,
    }
```

Remove the `games`, `news`, `gopher` and `ftp` accounts.
(The `ftp` account is taken care of in §11.34.1.)

auto: IAAC-1
auto: GEN000290-1
auto: GEN000290-2
auto: GEN000290-3
auto: GEN000290-4

```
    user { ['games', 'news', 'gopher']:
        ensure => absent,
    }

}
```

### Under RHEL6

On a freshly installed RHEL6 system, there exist files owned by the following users:

abrt
apache
avahi
daemon
gdm
haldaemon
lp
ntp
postfix
pulse
puppet
root
rpc
rpcuser
tss
vcsa

The following users, then, do not own any files:

bin
adm
uucp
games
rtkit
saslauth

```
sync        gopher      sshd
shutdown    ftp         tcpdump
halt        nobody      nfsnobody
mail        dbus
```

The system users not owning any files, listed above, are mostly associated with system processes; they are disabled from logging in by default.

RHEL6: GEN000280

The full list of possible system users under RHEL6 can be found in the Deployment Guide [15], §3.3. A user from that list is added when the package requiring the user is installed, so application accounts do not exist for applications not installed on the system. Policy regarding user accounts for people, including ensuring that people who aren't going to use a host are not added as users of that host, is dealt with in other subsections of §11.112.4.

RHEL6: GEN000290

```
class user::unnecessary::rhel6 {
```

Remove the `shutdown`, `halt` and `reboot` user accounts. The requirement says "special privilege accounts" must be removed, but only mentions these three.

auto: IAAC-1
auto: GEN000000-LNX00320

```
    user { ["shutdown", "halt", "reboot"]:
        ensure => absent,
    }
```

Some system users are installed by the `setup` package, but not subsequently used. Remove them.

auto: IAAC-1
auto: GEN000290

Not least to make pwck happy: their home directories seem not to usually exist.

```
    user { ["adm", "uucp", "gopher"]:
        ensure => absent,
    }
```

This user is listed as belonging to the `cyrus-imapd` package; we don't run IMAP servers.

```
    user { "saslauth":
        ensure => absent,
    }

    if($gdm_installed == 'false') {
        user { "gdm":
            ensure => absent,
        }
    }
}
```

### 11.113.2 Ensure validity of password file

```
class user::valid {
```

Make sure that user ids and user names are unique across all accounts, and that every user's primary group is one defined in the group file.

auto: ECSC-1
auto: IAIA-1
auto: GEN000300
auto: GEN000320
auto: GEN000380

Make sure that all users have a home, and that each user's home exists.

auto: ECSC-1
auto: GEN001440
auto: GEN001460

```
    exec { "pwck -r":
        path => "/usr/sbin",
        command => "pwck -r",
        logoutput => on_failure,
        loglevel => err,
        unless => "pwck -r",
    }
```

Resolve some complaints about home directories.

```
    if $::osfamily == 'RedHat' and $::operatingsystemrelease =~ /^6\..*/ {
        $users_array = split($::local_usernames, ' ')
        $has_pulse = inline_template('<%= @users_array.member? "pulse"-%>')
        $has_avahi = inline_template('<%= @users_array.member? "avahi-autoipd"-%>')
        if $has_avahi == 'true' {
            file { '/var/lib/avahi-autoipd':
                ensure => directory,
                owner => 'avahi-autoipd', group => 'root', mode => 0755,
            }
        }
        if $has_pulse == 'true' {
            file { '/var/run/pulse':
                ensure => directory,
                owner => 'pulse', group => 'root', mode => 0755,
            }
        }
    }
}
```

About the `unless` above: Jacob Helwig said on the `puppet-users` mailing list, 7 Jun 2011,

> By doing the "unless =¿ 'pwck -r'", the resource won't even show up as having been run if 'pwck -r' returns 0. Having to run the command twice is a hack, but it's the best I can think of at the moment.

See also `http://projects.puppetlabs.com/issues/7877`.

```
class user::virtual {
    User {
        shell => '/bin/bash',
        ensure => 'present',
        password => '!!',
    }

    @user {
        logview:
            comment => "Log viewing user",
            gid => logview, uid => 49152;
        'puppet_dba':
            comment => "OS user used by Puppet to administer PostgreSQL",
            gid => puppet_dba, uid => 49153;
    }

    @group {
        logview:
            gid => 49152;
        puppet_dba:
            gid => 49153;
    }
}
```

# 11.114 Unix-to-Unix Copy (uucp)

## 11.114.1 Turn off UUCP

UNIX SRG PDI GEN005280 requires that UUCP be disabled.

```
class uucp::no {
    case $::osfamily {
        'redhat': {
            case $::operatingsystemrelease {
```

RHEL6 does not provide a UUCP service.

RHEL6:
GEN005280

```
                /^6\..*$/: {}
                /^5\..*$/: { package { 'uucp': ensure => absent, } }
            }
        }
```

Make sure that the UUCP service is disabled.

auto: ECSC-1
auto: GEN005280 M6
auto: OSX8-00-00550

```
        'darwin': {
            service { 'com.apple.uucp':
                enable => false,
                ensure => stopped,
            }
        }
        default: { unimplemented() }
    }
}
```

## 11.115 VirtualBox

VirtualBox stuff

```
class virtualbox {
package {
'VirtualBox-4.2.x86_64':
ensure => present
}
```

Let admins sudo to run the driver installer manually if need be.

```
        sudo::auditable::command_alias { 'VIRTUALBOX_DRIVERS':
            type => 'exec',
            commands => [
                "/etc/init.d/vboxdrv setup",
                ],
        }
}
```

§11.104.3

## 11.116 Web servers

### 11.116.1 Pylons In SEEK EAGLE (PISE)

RHEL 6 includes Pylons 1.0, and the many other Python packages which it requires and uses. It appears that this forms a good foundation for building new web applications in Python, where 'good' means these things:

- Supported with security updates
- Easy to install on RHEL 6
- Already works for lots of people in the industry
- Good documentation is available
- Training may be available
- Short write–manual test–modify cycle
- It's easy to write and run unit and functional tests
- Debuggable (*i.e.*, runnable using a debugger)
- Deployment is well-defined
- Authentication methods can be changed

"Pylons" is mostly a collective term for many pieces which are bound together into a platform on which to write a web application. PISE denotes all the conventions, common pieces of configuration, and procedures involved in making and deploying Pylons applications under this Configuration Management for IT Systems Example Policy.

Colophon: a *pylon* is the entrance to an Egyptian temple. *Pisé de terre* (pee-ZAY deuh TAIR) is a technique of building walls or large bricks using rammed earth.

### Development machine

PISE developers need Pylons, and may need a database server, but do not need a working web server. Or, at least, not yet.

```
class web::pise::devel {
    include apache                                                    §11.8
    include python                                                    §11.82
    package {
        [
            "mod_wsgi",
            "mod_authz_ldap",
            "mod_auth_pgsql",
            "postgresql-server",
            "python-coverage",
            "python-nose",
            "python-cheetah",
            "python-formencode",
            "python-psycopg2",
            "python-pylons",
            "make",
        ]:
            ensure => present,
    }

}
```

## 11.117 X Window System server

Make sure an X server is installed.

The NVIDIA proprietary drivers need the X server installed, but it may be surprising for the `nvidia::proprietary` class to silently install an X server. So we install it here.

```
class xserver {
    case $::osfamily {
        'RedHat': {
            case $::operatingsystemrelease {
                /^[56]\..*$/: {
                    package { 'xorg-x11-server-Xorg':
                        ensure => present,
                    }
                }
                default: { unimplemented() }
            }
        }
        'Darwin': {}
        default: { unimplemented() }
    }
}
```

## 11.118 YUM (Yellowdog Updater, Modified)

GPG signatures are not checked on package install during kickstart, but they are checked weekly after that (see §11.84.5). The mitigation is that the kickstart network is more trusted than the production network. See §??.

### 11.118.1 Admin guidance about yum

Do not deploy any YUM repository configuration with `gpgcheck=0`. Do sign packages. See §10. admins do GEN008800

### 11.118.2 Turn off the Subscription Manager

Red Hat has moved to certificate-based subscriptions, using the Subscription Manager. But RHN Satellite 5.4.1 does not use these. But the plugin for certificate-based management is enabled by default. So since we don't have the certificates, every time Yum runs, that plugin complains that this system isn't subscribed. This class fixes that.

```
class yum::no_subscription_manager {
    augeas { 'disable_subscription_manager':
        context => "/files/etc/yum/pluginconf.d/\
subscription-manager.conf",
        changes => "set main/enabled 0",
    }
}
```

### 11.118.3 Custom YUM repository on Vagrant machines

On a proper network we may have a Red Hat Satellite server, but on a Vagrant host we may not have any networking, or may not be on the same network as such a server. Installation of most custom packages should be avoided under Vagrant, but some cannot be avoided. This class allows for custom packages distributed with the Vagrant machine to be made available to the virtual machine.

Virtual machines set up with Vagrant are not secure in a networking sense: they have a fixed default root password, a default user with a fixed default password having sudo access, fixed insecure ssh keys, etc. In line with these decisions, we won't perform GPG signature checks on the RPMs in the custom repository, because the provenance of these packages is already exactly as secure as the provenance of the Puppet policy applied at install time: any attacker who could pervert a custom package could just change the Puppet policy. And the virtual machine built from these things is ephemeral and untrusted anyway.

```
class yum::vagrant() {
    yumrepo { "vagrant":
        name => "vagrant",
        baseurl => "file:///vagrant/custom-packages",
        enabled => 1,
        gpgcheck => 0,
    }
}
class zenity {
    package { "zenity": ensure => present, }
}
```

# Chapter 12

# Attendant files

Here follow the files used by the policy.

Wherever you see `[WRAP]` at the end of a line, that line was wrapped in order to fit on the page; if you find yourself in the unfortunate position of typing that line into a computer, do not type [WRAP] and do not start a new line. Lines not ending with `[WRAP]` end with a newline in the original text of the file.

Wherever you see something like `[UNICODE \u5678 MAYBE SOME WORDS]`, the original text of the file contained a Unicode character which could not be reproduced exactly in this document. If the Unicode character database includes a description of the character, it is included; if not, only the character's identity is included.

# 12.1 aide/

For the policy that requires files in this section, see 11.5.

## 12.1.1 aide.conf

```
# Example configuration file for AIDE.

@@define DBDIR /var/lib/aide
@@define LOGDIR /var/log/aide

# The location of the database to be read.
database=file:@@{DBDIR}/aide.db.gz

# The location of the database to be written.
#database_out=sql:host:port:database:login_name:passwd:table
#database_out=file:aide.db.new
database_out=file:@@{DBDIR}/aide.db.new.gz

# Whether to gzip the output to database
gzip_dbout=yes

# Default.
verbose=5

report_url=file:@@{LOGDIR}/aide.log
report_url=stdout
#report_url=stderr
#NOT IMPLEMENTED report_url=mailto:root@foo.com
#NOT IMPLEMENTED report_url=syslog:LOG_AUTH

# These are the default rules.
#
#p:      permissions
#i:      inode:
#n:      number of links
#u:      user
#g:      group
#s:      size
#b:      block count
#m:      mtime
#a:      atime
#c:      ctime
#S:      check for growing size
#acl:           Access Control Lists
#selinux        SELinux security context
#xattrs:        Extended file attributes
#md5:    md5 checksum
#sha1:   sha1 checksum
#sha256:        sha256 checksum
#sha512:        sha512 checksum
#rmd160: rmd160 checksum
#tiger:  tiger checksum

#haval:  haval checksum (MHASH only)
#gost:   gost checksum (MHASH only)
```

```
#crc32:  crc32 checksum (MHASH only)
#whirlpool:     whirlpool checksum (MHASH only)

#R:             p+i+n+u+g+s+m+c+acl+selinux+xattrs+md5
#L:             p+i+n+u+g+acl+selinux+xattrs
#E:             Empty group
#>:             Growing logfile p+u+g+i+n+S+acl+selinux+xattrs

R = p+i+n+u+g+s+m+c+acl+selinux+xattrs+sha256
# You can create custom rules like this.
# With MHASH...
# ALLXTRAHASHES = sha1+rmd160+sha256+sha512+whirlpool+tiger+haval+gost+crc3[WRAP]
2
ALLXTRAHASHES = sha1+sha512
# Everything but access time (Ie. all changes)
EVERYTHING = R+ALLXTRAHASHES

# Sane, with multiple hashes
# NORMAL = R+rmd160+sha256+whirlpool
NORMAL = R+sha512

# For directories, don’t bother doing hashes
DIR = p+i+n+u+g+acl+selinux+xattrs

# Access control only
# UNIX SRG GEN006571: ‘‘The file integrity tool must be configured to verif[WRAP]
y
# extended attributes.’’
PERMS = p+i+u+g+acl+selinux+xattrs

# Logfile are special, in that they often change
LOG = >

# Just do md5 and sha256 hashes
LSPP = R+sha256

# Some files get updated automatically, so the inode/ctime/mtime change
# but we want to know when the data inside them changes
DATAONLY =  p+n+u+g+s+acl+selinux+xattrs+sha256+sha512

# Next decide what directories/files you want in the database.

/boot   NORMAL
/bin    NORMAL
/sbin   NORMAL
/lib    NORMAL
/lib64  NORMAL
/opt    NORMAL
/usr    NORMAL
/root   NORMAL
# These are too volatile
!/usr/src
!/usr/tmp

/etc    PERMS
!/etc/mtab
# Ignore backup files
```

```
!/etc/.*~
/etc/exports  NORMAL
/etc/fstab    NORMAL
/etc/passwd   NORMAL
/etc/group    NORMAL
/etc/gshadow  NORMAL
/etc/shadow   NORMAL
/etc/security/opasswd   NORMAL

/etc/hosts.allow   NORMAL
/etc/hosts.deny    NORMAL

/etc/sudoers NORMAL
/etc/skel NORMAL

/etc/logrotate.d NORMAL

/etc/resolv.conf DATAONLY

/etc/nscd.conf NORMAL
/etc/securetty NORMAL

# Shell/X starting files
/etc/profile NORMAL
/etc/bashrc NORMAL
/etc/bash_completion.d/ NORMAL
/etc/login.defs NORMAL
/etc/zprofile NORMAL
/etc/zshrc NORMAL
/etc/zlogin NORMAL
/etc/zlogout NORMAL
/etc/profile.d/ NORMAL
/etc/X11/ NORMAL

# Pkg manager
/etc/yum.conf NORMAL
/etc/yumex.conf NORMAL
/etc/yumex.profiles.conf NORMAL
/etc/yum/ NORMAL
/etc/yum.repos.d/ NORMAL

/var/log   LOG
/var/run/utmp LOG

# This gets new/removes-old filenames daily
!/var/log/sa
# As we are checking it, we’ve truncated yesterdays size to zero.
!/var/log/aide.log

# LSPP rules...
# AIDE produces an audit record, so this becomes perpetual motion.
# /var/log/audit/ LSPP
/etc/audit/ LSPP
/etc/libaudit.conf LSPP
/usr/sbin/stunnel LSPP
/var/spool/at LSPP
/etc/at.allow LSPP
```

```
/etc/at.deny LSPP
/etc/cron.allow LSPP
/etc/cron.deny LSPP
/etc/cron.d/ LSPP
/etc/cron.daily/ LSPP
/etc/cron.hourly/ LSPP
/etc/cron.monthly/ LSPP
/etc/cron.weekly/ LSPP
/etc/crontab LSPP
/var/spool/cron/root LSPP

/etc/login.defs LSPP
/etc/securetty LSPP
# No, we do not want to try to checksum large sparse files.
#/var/log/faillog LSPP
#/var/log/lastlog LSPP

/etc/hosts LSPP
/etc/sysconfig LSPP

/etc/inittab LSPP
/etc/grub/ LSPP
/etc/rc.d LSPP

/etc/ld.so.conf LSPP

/etc/localtime LSPP

/etc/sysctl.conf LSPP

/etc/modprobe.conf LSPP

/etc/pam.d LSPP
/etc/security LSPP
/etc/aliases LSPP
/etc/postfix LSPP

/etc/ssh/sshd_config LSPP
/etc/ssh/ssh_config LSPP

/etc/stunnel LSPP

/etc/vsftpd.ftpusers LSPP
/etc/vsftpd LSPP

/etc/issue LSPP
/etc/issue.net LSPP

/etc/cups LSPP

# With AIDE's default verbosity level of 5, these would give lots of
# warnings upon tree traversal. It might change with future version.
#
#=/lost\+found    DIR
#=/home           DIR

# Ditto /var/log/sa reason...
```

```
!/var/log/and-httpd

# lastlog and faillog are huge and sparse and change often. Don't checksum
/var/log/lastlog L
/var/log/faillog L

# Admins dot files constantly change, just check perms
/root/\..* PERMS

# STIG GEN000140: baseline device files too. Avoid checksumming, because
# reading a device file means something special and, here, unintended.
/dev/.* L

# Apache STIG WG440: monitor CGI scripts. We'll be non-sticklers and includ[WRAP]
e
# code that the scripts call. Python libraries are covered under /usr.
/var/www/cgi-bin             NORMAL
/var/www/wsgi-bin            NORMAL
/var/www/sbu-apps            NORMAL
# Code is part of the baseline; configuration ... is too.
/var/www/sbu-apps/.*/config  NORMAL

# Database STIG DG0050: monitor database ``configuration files.'' ``Softwar[WRAP]
e
# libraries'' and ``applications'' are covered under /usr, for DBMSes inclu[WRAP]
ded
# with RHEL.
#
# /etc/pam.d/postgresql included under /etc/pam.d
/var/lib/pgsql/.bash_profile    NORMAL
/var/lib/pgsql/data/*.conf      NORMAL
/var/lib/pgsql/data/*.opts      NORMAL
# /etc/sysconfig/pgsql included under /etc/sysconfig
```

### 12.1.2 backup_baseline.sh

```
#!/bin/sh

umask 077
set -e
# come up with a decent directory to compose in
dir=$(mktemp -d)
cd_size=700000000
oldpwd=$(pwd)
cd $dir
mkdir cd-files
cp /var/lib/aide/* cd-files/
cp /var/cfengine/checksum_digests.db cd-files/
mkisofs -RJ -o baseline-backup.iso -V "$(hostname) baseline"
size=$(stat -c %s baseline-backup.iso)
if [ $size -lt $cd_size ]; then
    cdrecord -data baseline-backup.iso
    cd $oldpwd
    rm -rf $dir
else
    # Since this script is intended to be run manually, I don't expect anyo[WRAP]
ne
```

```
    # to be able to deny service by causing the baseline backup to be too b[WRAP]
ig:
    # the admin will be sitting there watching it happen and will clean up.
    echo "Baseline backup $size is bigger than a CD ($cd_size bytes)" >&2
    echo "Not proceeding farther" >&2
    exit 1
fi
```

### 12.1.3 logrotate

```
#!/bin/sh

/usr/sbin/logrotate /etc/logrotate.conf >/dev/null 2>&1
EXITVALUE=$?
if [ $EXITVALUE != 0 ]; then
    /usr/bin/logger -t logrotate "ALERT exited abnormally with [$EXITVALUE][WRAP]
"
fi
exit 0
```

## 12.2 apache/

For the policy that requires files in this section, see 11.7.1.

### 12.2.1 common/nss-site-cac.conf

```
# \implementsapachestig{WG140 A22} Require client certificates from a
# DoD-authorized CA.
NSSVerifyClient require

ErrorDocument 401 /pages/401.html
# Let unauthenticated users actually get that file
<Location /pages/401.html>
Satisfy Any
</Location>

# SSL options
#   o FakeBasicAuth:
#     the user needs this password: ‘xxj31ZMTZzkVA’.
#   o ExportCertData:
#     exports PEM-encoded certificates in environment as SSL_CLIENT_CERT an[WRAP]
d
#     SSL_SERVER_CERT.
#   o StdEnvVars:
#     only use for locations corresponding to scripts, not static pages: it[WRAP]
 is
#     expensive
#   o StrictRequire:
#     This denies access when "NSSRequireSSL" or "NSSRequire" applied even
#     under a "Satisfy any" situation, i.e. when it applies access is denie[WRAP]
d
#     and no other module can change it.
#   o OptRenegotiate:
#     This enables optimized SSL connection renegotiation handling when SSL
#     directives are used in per-directory context.
#NSSOptions +FakeBasicAuth +ExportCertData +CompatEnvVars +StrictRequire

NSSOptions +StrictRequire +FakeBasicAuth
# This username is only given as the REMOTE_USER environment variable visib[WRAP]
le
# to CGI and WSGI: in all authorization checks, ’/’ plus the comma-delimite[WRAP]
d
# certificate distinguished name is used
### See #I332, #I333
NSSUserName SSL_CLIENT_S_DN_CN
```

### 12.2.2 common/nss-site-common.conf

```
# CVE-2007-4465, TCNO 2007-292-002, due Dec 17
# Also this implements APP3530 in the Application Security & Development ST[WRAP]
IG.
AddDefaultCharset utf-8

# \implements{apachestig}{WA00615 A22} Enable ‘‘system logging’’,
# using CustomLog not TransferLog.
CustomLog "|/usr/bin/logger -t httpd__access -i -p local6.info" common
```

```
# \implements{apachestig}{WA00605 A22} Enable error logging.
ErrorLog syslog

# LogLevel is not inherited by virtual hosts from the httpd.conf setting.
# \implements{apachestig}{WA00620 A22} The requirement says we must have a
# LogLevel directive; the check says that if it isn't exactly "warn," that'[WRAP]
s a
# finding.
LogLevel warn

# \implements{apachestig}{WG340} \implements{apachestig}{WG340 A22} Use TLS[WRAP]
. The
# validation procedure listed in the STIG will not work for this and many m[WRAP]
ore
# requirements addressed below, because the expectation in the STIG is that[WRAP]
 you
# will be using mod_ssl, not mod_nss.
NSSEngine on

NSSCipherSuite +rsa_3des_sha,+fips_3des_sha,+rsa_aes_128_sha,+rsa_aes_256_s[WRAP]
ha
NSSFIPS on

# SSLv2 is reputedly broken. Don't use it.
# Sharper: \implements{apachestig}{WG340} Use only TLSv1.
# Same: \implements{apachestig}{WG340 A22}
#
# And after all of that - "NSSFIPS on" above makes mod_nss ignore
# NSSProtocol directives and only use the right protocols anyway.
NSSProtocol TLSv1

# \implements{apachestig}{WG145 A22} We're going to use CRLs, not OCSP---fo[WRAP]
r now,
# at least.
NSSOCSP off

#   Use a default OCSP responder. If enabled this will be used regardless
#   of whether one is included in a client certificate. Note that the
#   server certificate is verified during startup.
#
#   NSSOCSPDefaultURL defines the service URL of the OCSP responder
#   NSSOCSPDefaultName is the nickname of the certificate to trust to
#       sign the OCSP responses.
#NSSOCSPDefaultResponder on
#NSSOCSPDefaultURL http://example.com/ocsp/status
#NSSOCSPDefaultName ocsp-nickname
```

### 12.2.3 common/nss-site-kerberos.conf

```
<Location />
  AuthType Kerberos
  KrbMethodNegotiate on
  KrbMethodK5Passwd off
  Krb5Keytab /etc/http.keytab
  # By not specifying KrbAuthRealms, we use the default realm in
  # /etc/krb5.conf. By not specifying KrbServiceName, we use the
  # default of HTTP (note! HTTP and http are different).
</Location>

# To make use of this, you need two more items of configuration anywhere yo[WRAP]
u’re
# going to require authentication, namely, an AuthName and a Require. These[WRAP]
 are
# usual Apache authentication fare, not Kerberos-specific, so see the Apach[WRAP]
e
# documentation.
#
# The Require is how you authorize people, and usually you would say, "Requ[WRAP]
ire
# group somethingorother." No groups are brought into existence by the abov[WRAP]
e
# configuration; you’ll have to make local groups containing usernames like
# user@REALM and use an AuthGroupFile directive, or use some other module t[WRAP]
o
# obtain groups from another server, e.g. via LDAP.
#
# You also need /etc/http.keytab to exist, with keys in it for
# HTTP/this_hosts_fqdn@REALM. If your Kerberos server is an Active Director[WRAP]
y
# host, you need to use ktpass.exe to make this keytab. This cannot be secu[WRAP]
rely
# automated.
```

## 12.3 audit/

For the policy that requires files in this section, see 11.11.2.

### 12.3.1 auditd.cron

```
#!/bin/sh

##########
# This script can be installed to get a daily log rotation
# based on a cron job.
##########

/sbin/service auditd rotate >/dev/null
EXITVALUE=$?
if [ $EXITVALUE != 0 ]; then
    /usr/bin/logger -t auditd "ALERT exited abnormally with [$EXITVALUE]"
fi
exit 0
```

### 12.3.2 i386-stig.rules

```
## This file contains the auditctl rules that are loaded
## whenever the audit daemon is started via the initscripts.
## The rules are simply the parameters that would be passed
## to auditctl.
##
## First rule - delete all
-D

## Increase the buffers to survive stress events.
## Make this bigger for busy systems
-b 8192

## Set failure mode to panic
-f 2

## NOTE:
## 1) if this is being used on a 32 bit machine, comment out the b64 lines
##    [they were deleted in this copy]
## 2) These rules assume that login under the root account is not allowed.
## 3) It is also assumed that 500 represents the first usable user account.
## 4) If these rules generate too much spurious data for your tastes, limit[WRAP]
 the
## the syscall file rules with a directory, like -F dir=/etc
## 5) You can search for the results on the key fields in the rules
##
##
## (GEN002880: CAT II) The IAO will ensure the auditing software can
## record the following for each audit event:
##- Date and time of the event
##- Userid that initiated the event
##- Type of event
##- Success or failure of the event
##- For I&A events, the origin of the request (e.g., terminal ID)
##- For events that introduce an object into a user[UNICODE \u2019 RIGHT SI[WRAP]
NGLE QUOTATION MARK]s address space, and
##  for object deletion events, the name of the object, and in MLS
##  systems, the object[UNICODE \u2019 RIGHT SINGLE QUOTATION MARK]s securi[WRAP]
ty level.
##
## Things that could affect time
# \implements{rhel5stig}{GEN002760-3,GEN002760-4,GEN002760-5,GEN002760-6}
-a always,exit -F arch=b32 -S adjtimex -S settimeofday -S stime -k time-cha[WRAP]
nge
-a always,exit -F arch=b32 -S clock_settime -k time-change
-w /etc/localtime -p wa -k time-change

# SRG v1r1 GEN002750, GEN002751, GEN002752, GEN002753, account modification[WRAP]
s,
# appear to be hardcoded into auditd.

## Things that affect identity
-w /etc/group -p wa -k identity
-w /etc/passwd -p wa -k identity
-w /etc/gshadow -p wa -k identity
-w /etc/shadow -p wa -k identity
-w /etc/security/opasswd -p wa -k identity

## Things that could affect system locale
-a exit,always -F arch=b32 -S sethostname -S setdomainname -k system-locale
-w /etc/issue -p wa -k system-locale
-w /etc/issue.net -p wa -k system-locale
-w /etc/hosts -p wa -k system-locale
-w /etc/sysconfig/network -p wa -k system-locale

## Things that could affect MAC policy
-w /etc/selinux/ -p wa -k MAC-policy


## (GEN002900: CAT III) The IAO will ensure audit files are retained at
## least one year; systems containing SAMI will be retained for five years.
##
## Site action - no action in config files

## (GEN002920: CAT III) The IAO will ensure audit files are backed up
## no less than weekly onto a different system than the system being
## audited or backup media.
##
## Can be done with cron script

## (GEN002700: CAT I) (Previously [UNICODE \u2013 EN DASH] G095) The SA wil[WRAP]
l ensure audit data
## files have permissions of 640, or more restrictive.
##
## Done automatically by auditd

## (GEN002720-GEN002840: CAT II) (Previously [UNICODE \u2013 EN DASH] G100-[WRAP]
G106) The SA will
## configure the auditing system to audit the following events for all
## users and root:
##
```

```
# SRG v1r1 GEN002800
## - Logon (unsuccessful and successful) and logout (successful)
##
## Handled by pam, sshd, login, and gdm
## Might also want to watch these files if needing extra information
# \implements{rhel5stig}{GEN002800}
-w /var/log/faillog -p wa -k logins
-w /var/log/lastlog -p wa -k logins


##- Process and session initiation (unsuccessful and successful)
##
## The session initiation is audited by pam without any rules needed.
## Might also want to watch this file if needing extra 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

##- Discretionary access control permission modification (unsuccessful
## and successful use of chown/chmod)
# \implements{rhel5stig}{GEN002820}
# "Any restrictions (such as with -F) beyond [architecture restrictions] ar[WRAP]
e
# not in strict compliance..." This sentence is written in some, but not al[WRAP]
l,
# of the audit requirements in the RHEL 5 STIG.
# \implements{rhel5stig}{GEN002820-2,GEN002820-3}
-a always,exit -F arch=b32 -S chmod -S fchmod -S fchmodat -k perm_mod
# \implements{rhel5stig}{GEN002820-4,GEN002820-5,GEN002820-6,GEN002820-7}
-a always,exit -F arch=b32 -S chown -S fchown -S fchownat -S lchown -k perm[WRAP]
_mod
# \implements{rhel5stig}{GEN002820-8,GEN002820-9,GEN002820-10,GEN002820-11,[WRAP]
GEN002820-12,GEN002820-13}
-a always,exit -F arch=b32 -S setxattr -S lsetxattr -S fsetxattr -S removex[WRAP]
attr -S lremovexattr -S fremovexattr -k perm_mod

# \implements{rhel5stig}{GEN002720-2,GEN002720-3,GEN002720-4,GEN002720-5}
##- Unauthorized access attempts to files (unsuccessful)
-a always,exit -F arch=b32 -S creat -S open -S openat -S truncate -S ftrunc[WRAP]
ate -F exit=-EACCES -F auid>=500 -F auid!=4294967295 -k access
-a always,exit -F arch=b32 -S creat -S open -S openat -S truncate -S ftrunc[WRAP]
ate -F exit=-EPERM -F auid>=500 -F auid!=4294967295 -k access

##- Use of privileged commands (unsuccessful and successful)
## use find /bin -type f -perm -04000 2>/dev/null and put all those files i[WRAP]
n a rule like this
-a always,exit -F path=/bin/ping -F perm=x -F auid>=500 -F auid!=4294967295[WRAP]
 -k privileged

##- Use of print command (unsuccessful and successful)

##- Export to media (successful)
## You have to mount media before using it. You must disable all automounti[WRAP]
ng
## so that its done manually in order to get the correct user requesting th[WRAP]
e
## export
```

```
-a always,exit -F arch=b32 -S mount -F auid>=500 -F auid!=4294967295 -k exp[WRAP]
ort

##- System startup and shutdown (unsuccessful and successful)

# SRG v1r1 GEN002740
##- Files and programs deleted by the user (successful and unsuccessful)
-a always,exit -F arch=b32 -S unlink -S unlinkat -S rename -S renameat -F a[WRAP]
uid>=500 -F auid!=4294967295 -k delete
# \implements{rhel5stig}{GEN002740-2}
-a always,exit -F arch=b32 -S rmdir -F auid>=500 -F auid!=4294967295 -k del[WRAP]
ete


# SRG v1r1 GEN002760
##- All system administration actions
##- All security personnel actions
##
## Look for pam_tty_audit and add it to your login entry point's pam config[WRAP]
s.
## If that is not found, use sudo which should be patched to record its
## commands to the audit system. Do not allow unrestricted root shells or
## sudo cannot record the action.
-w /etc/sudoers -p wa -k actions

# \implements{rhel5stig}{GEN002760-2}
-w /etc/audit.rules
-w /etc/audit/audit.rules

# \implements{rhel5stig}{GEN002760-7,GEN002760-8}
-a exit,always -F arch=b32 -S sethostname -S setdomainname

# \implements{rhel5stig}{GEN002760-9,GEN002760-10}
-a exit,always -F arch=b32 -S sched_setparam -S sched_setscheduler


## (GEN002860: CAT II) (Previously [UNICODE \u2013 EN DASH] G674) The SA an[WRAP]
d/or IAO will
##ensure old audit logs are closed and new audit logs are started daily.
##
## Site action. Can be assisted by a cron job

## Not specifically required by the STIG; but common sense items
## Optional - could indicate someone trying to do something bad or
## just debugging
#-a entry,always -F arch=b32 -S ptrace -k tracing

## Optional - could be an attempt to bypass audit or simply legacy program
#-a always,exit -F arch=b32 -S personality -k bypass

## Put your own watches after this point
# -w /your-file -p rwxa -k mykey

# SRG v1r1 GEN002825: dynamic kernel module loading and unloading
# \implements{rhel5stig}{GEN002825,GEN002825-2}
-a always,exit -F arch=b32 -S create_module -S init_module -S delete_module
# \implements{rhel5stig}{GEN002825-3}
```

```
-w /sbin/insmod -p x
# \implements{rhel5stig}{GEN002825-4}
-w /sbin/modprobe -p x
# \implements{rhel5stig}{GEN002825-5}
-w /sbin/rmmod -p x


## Make the configuration immutable - reboot is required to change audit ru[WRAP]
les
-e 2
```

### 12.3.3 x86_64-stig.rules

```
## This file contains the auditctl rules that are loaded
## whenever the audit daemon is started via the initscripts.
## The rules are simply the parameters that would be passed
## to auditctl.
##
## First rule - delete all
-D

## Increase the buffers to survive stress events.
## Make this bigger for busy systems
-b 32768

## Set failure mode to panic
-f 2

## NOTE:
## 1) if this is being used on a 32 bit machine, comment out the b64 lines
## 2) These rules assume that login under the root account is not allowed.
## 3) It is also assumed that 500 represents the first usable user account.
## 4) If these rules generate too much spurious data for your tastes, limit[WRAP]
 the
## the syscall file rules with a directory, like -F dir=/etc
## 5) You can search for the results on the key fields in the rules
##
##
## (GEN002880: CAT II) The IAO will ensure the auditing software can
## record the following for each audit event:
##- Date and time of the event
##- Userid that initiated the event
##- Type of event
##- Success or failure of the event
##- For I&A events, the origin of the request (e.g., terminal ID)
##- For events that introduce an object into a user[UNICODE \u2019 RIGHT SI[WRAP]
NGLE QUOTATION MARK]s address space, and
##  for object deletion events, the name of the object, and in MLS
##  systems, the object[UNICODE \u2019 RIGHT SINGLE QUOTATION MARK]s securi[WRAP]
ty level.
##
## Things that could affect time
# \implements{rhel5stig}{GEN002760-3,GEN002760-4,GEN002760-5,GEN002760-6}
-a always,exit -F arch=b32 -S adjtimex -S settimeofday -S stime -k time-cha[WRAP]
nge
# stime appears not to be a valid 64-bit syscall; removing so audit
# rules will load
-a always,exit -F arch=b64 -S adjtimex -S settimeofday -k time-change
-a always,exit -F arch=b32 -S clock_settime -k time-change
-a always,exit -F arch=b64 -S clock_settime -k time-change
-w /etc/localtime -p wa -k time-change

# SRG v1r1 GEN002750, GEN002751, GEN002752, GEN002753, account modification[WRAP]
s,
# appear to be hardcoded into auditd.

## Things that affect identity
-w /etc/group -p wa -k identity
-w /etc/passwd -p wa -k identity
-w /etc/gshadow -p wa -k identity
-w /etc/shadow -p wa -k identity
-w /etc/security/opasswd -p wa -k identity

## Things that could affect system locale
-a exit,always -F arch=b32 -S sethostname -S setdomainname -k system-locale
-a exit,always -F arch=b64 -S sethostname -S setdomainname -k system-locale
-w /etc/issue -p wa -k system-locale
-w /etc/issue.net -p wa -k system-locale
-w /etc/hosts -p wa -k system-locale
-w /etc/sysconfig/network -p wa -k system-locale

## Things that could affect MAC policy
-w /etc/selinux/ -p wa -k MAC-policy


## (GEN002900: CAT III) The IAO will ensure audit files are retained at
## least one year; systems containing SAMI will be retained for five years.
##
## Site action - no action in config files

## (GEN002920: CAT III) The IAO will ensure audit files are backed up
## no less than weekly onto a different system than the system being
## audited or backup media.
##
## Can be done with cron script

## (GEN002700: CAT I) (Previously [UNICODE \u2013 EN DASH] G095) The SA wil[WRAP]
l ensure audit data
## files have permissions of 640, or more restrictive.
##
## Done automatically by auditd

## (GEN002720-GEN002840: CAT II) (Previously [UNICODE \u2013 EN DASH] G100-[WRAP]
G106) The SA will
## configure the auditing system to audit the following events for all
## users and root:
##
# SRG v1r1 GEN002800
## - Logon (unsuccessful and successful) and logout (successful)
##
## Handled by pam, sshd, login, and gdm
## Might also want to watch these files if needing extra information
# \implements{rhel5stig}{GEN002800}
```

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


##- Process and session initiation (unsuccessful and successful)
##
## The session initiation is audited by pam without any rules needed.
## Might also want to watch this file if needing extra 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

##- Discretionary access control permission modification (unsuccessful
## and successful use of chown/chmod)
# \implements{rhel5stig}{GEN002820}
# "Any restrictions (such as with -F) beyond [architecture restrictions] ar[WRAP]
e
# not in strict compliance..." This sentence is written in some, but not al[WRAP]
l,
# of the audit requirements in the RHEL 5 STIG.
# \implements{rhel5stig}{GEN002820-2,GEN002820-3}
-a always,exit -F arch=b32 -S chmod -S fchmod -S fchmodat -k perm_mod
-a always,exit -F arch=b64 -S chmod -S fchmod -S fchmodat -k perm_mod
# \implements{rhel5stig}{GEN002820-4,GEN002820-5,GEN002820-6,GEN002820-7}
-a always,exit -F arch=b32 -S chown -S fchown -S fchownat -S lchown -k perm[WRAP]
_mod
-a always,exit -F arch=b64 -S chown -S fchown -S fchownat -S lchown -k perm[WRAP]
_mod
# \implements{rhel5stig}{GEN002820-8,GEN002820-9,GEN002820-10,GEN002820-11,[WRAP]
GEN002820-12,GEN002820-13}
-a always,exit -F arch=b32 -S setxattr -S lsetxattr -S fsetxattr -S removex[WRAP]
attr -S lremovexattr -S fremovexattr -k perm_mod
-a always,exit -F arch=b64 -S setxattr -S lsetxattr -S fsetxattr -S removex[WRAP]
attr -S lremovexattr -S fremovexattr -k perm_mod

# \implements{rhel5stig}{GEN002720-2,GEN002720-3,GEN002720-4,GEN002720-5}
##- Unauthorized access attempts to files (unsuccessful)
-a always,exit -F arch=b32 -S creat -S open -S openat -S truncate -S ftrunc[WRAP]
ate -F exit=-EACCES -F auid>=500 -F auid!=4294967295 -k access
-a always,exit -F arch=b32 -S creat -S open -S openat -S truncate -S ftrunc[WRAP]
ate -F exit=-EPERM -F auid>=500 -F auid!=4294967295 -k access
-a always,exit -F arch=b64 -S creat -S open -S openat -S truncate -S ftrunc[WRAP]
ate -F exit=-EACCES -F auid>=500 -F auid!=4294967295 -k access
-a always,exit -F arch=b64 -S creat -S open -S openat -S truncate -S ftrunc[WRAP]
ate -F exit=-EPERM -F auid>=500 -F auid!=4294967295 -k access

##- Use of privileged commands (unsuccessful and successful)
## use find /bin -type f -perm -04000 2>/dev/null and put all those files i[WRAP]
n a rule like this
-a always,exit -F path=/bin/ping -F perm=x -F auid>=500 -F auid!=4294967295[WRAP]
 -k privileged

##- Use of print command (unsuccessful and successful)

##- Export to media (successful)
## You have to mount media before using it. You must disable all automounti[WRAP]
ng
```

```
## so that its done manually in order to get the correct user requesting th[WRAP]
e
## export
-a always,exit -F arch=b32 -S mount -F auid>=500 -F auid!=4294967295 -k exp[WRAP]
ort
-a always,exit -F arch=b64 -S mount -F auid>=500 -F auid!=4294967295 -k exp[WRAP]
ort

##- System startup and shutdown (unsuccessful and successful)

# SRG v1r1 GEN002740
##- Files and programs deleted by the user (successful and unsuccessful)
-a always,exit -F arch=b32 -S unlink -S unlinkat -S rename -S renameat -F a[WRAP]
uid>=500 -F auid!=4294967295 -k delete
-a always,exit -F arch=b64 -S unlink -S unlinkat -S rename -S renameat -F a[WRAP]
uid>=500 -F auid!=4294967295 -k delete
# \implements{rhel5stig}{GEN002740-2}
-a always,exit -F arch=b32 -S rmdir -F auid>=500 -F auid!=4294967295 -k del[WRAP]
ete
-a always,exit -F arch=b64 -S rmdir -F auid>=500 -F auid!=4294967295 -k del[WRAP]
ete


# SRG v1r1 GEN002760
##- All system administration actions
##- All security personnel actions
##
## Look for pam_tty_audit and add it to your login entry point’s pam config[WRAP]
s.
## If that is not found, use sudo which should be patched to record its
## commands to the audit system. Do not allow unrestricted root shells or
## sudo cannot record the action.
-w /etc/sudoers -p wa -k actions

# \implements{rhel5stig}{GEN002760-2}
-w /etc/audit.rules
-w /etc/audit/audit.rules

# \implements{rhel5stig}{GEN002760-7,GEN002760-8}
-a exit,always -F arch=b32 -S sethostname -S setdomainname
-a exit,always -F arch=b64 -S sethostname -S setdomainname

# \implements{rhel5stig}{GEN002760-9,GEN002760-10}
-a exit,always -F arch=b32 -S sched_setparam -S sched_setscheduler
-a exit,always -F arch=b64 -S sched_setparam -S sched_setscheduler


## (GEN002860: CAT II) (Previously [UNICODE \u2013 EN DASH] G674) The SA an[WRAP]
d/or IAO will
##ensure old audit logs are closed and new audit logs are started daily.
##
## Site action. Can be assisted by a cron job

## Not specifically required by the STIG; but common sense items
## Optional - could indicate someone trying to do something bad or
## just debugging
#-a entry,always -F arch=b32 -S ptrace -k tracing
```

```
#-a entry,always -F arch=b64 -S ptrace -k tracing

## Optional - could be an attempt to bypass audit or simply legacy program
#-a always,exit -F arch=b32 -S personality -k bypass
#-a always,exit -F arch=b64 -S personality -k bypass

## Put your own watches after this point
# -w /your-file -p rwxa -k mykey

# SRG v1r1 GEN002825: dynamic kernel module loading and unloading
# \implements{rhel5stig}{GEN002825,GEN002825-2}
-a always,exit -F arch=b64 -S create_module -S init_module -S delete_module
-a always,exit -F arch=b32 -S create_module -S init_module -S delete_module
# \implements{rhel5stig}{GEN002825-3}
-w /sbin/insmod -p x
# \implements{rhel5stig}{GEN002825-4}
-w /sbin/modprobe -p x
# \implements{rhel5stig}{GEN002825-5}
-w /sbin/rmmod -p x


## Make the configuration immutable - reboot is required to change audit ru[WRAP]
les
-e 2
```

## 12.4 augeas/

For the policy that requires files in this section, see 11.12.3.

### 12.4.1 0.9.0/lenses/abrt.aug

```
(* ABRT 2 configuration is like an ini file with no sections *)
module Abrt =
    autoload xfm

    let comment = Inifile.comment "#" "#"
    let empty = Inifile.empty
    let eq = del /[ \t]*=/ " ="
    let entry = IniFile.entry IniFile.entry_re eq comment

    let lns = ( entry | empty ) *

    let xfm = transform lns (incl "/etc/abrt/*.conf" . incl "/etc/abrt/plug[WRAP]
ins/*.conf")
```

### 12.4.2 0.9.0/lenses/auditdconf.aug

```
module Auditdconf =
    autoload xfm

    let comment = Inifile.comment "#" "#"
    let empty = Inifile.empty
    let eq = del /[ \t]*=/ " ="
    let entry = IniFile.entry IniFile.entry_re eq comment

    let lns = ( entry | empty ) *

    let xfm = transform lns (incl "/etc/audit/auditd.conf")
```

### 12.4.3 0.9.0/lenses/automaster.aug

```
module Automaster =
    autoload xfm

    let eol = Util.eol
    let comment = Util.comment
    let empty = Util.empty

    let mount_point = store /\/[^# \t\n]+/
    let include = [ label "include" .
                    del /\+[ \t]*/ "+" .
                    store /[^# \t\n]+/ .
                    eol ]
    let options = [ label "options" . store /-[^ \t\n]+/ ]
    let map_param =
        let    name = [ label "name" . store /[^: \t\n]+/ ]
        in let type = [ label "type" . store /[a-z]+/ ]
        in let format = [ label "format" . store /[a-z]+/ ]
        in let options = [ label "options" . store /[^ \t\n]+/ ]
        in let prelude = ( type .
                         ( del "," "," . format ) ? .
                         del ":" ":" )
        in ( prelude ? .
             name .
             ( Util.del_ws_spc . options ) ? )
    let map_record = [ label "map" .
                       mount_point . Util.del_ws_spc .
                       map_param .
                       eol ]

    let lns = ( map_record |
                include |
                comment |
                empty ) *

    let relevant = (incl "/etc/auto.master") .
                   Util.stdexcl
    let xfm = transform lns relevant
```

### 12.4.4 0.9.0/lenses/gdm2conf.aug

```
(* it’s just an ini file. sections ("titles") are required *)
module Gdm2conf =
    autoload xfm

    let comment = IniFile.comment "#" "#"
    let sep = IniFile.sep "=" "="
    let entry = IniFile.indented_entry IniFile.entry_re sep comment
    let title = IniFile.indented_title IniFile.record_re
    let record = IniFile.record title entry

    let lns = IniFile.lns record comment

    let relevant = ( incl "/etc/gdm/custom.conf" ) .
                   ( incl "/etc/gdm/securitytokens.conf" )

    let xfm = transform lns relevant
```

### 12.4.5 0.9.0/lenses/gshadow.aug

```
(* based on the group module for Augeas by Free Ekanayaka <free@64studio.co[WRAP]
m>

 Reference: man 5 gshadow

*)

module Gshadow =

   autoload xfm

(************************************************************************
 *                          USEFUL PRIMITIVES
 *************************************************************************)
```

```
let eol        = Util.eol
let comment    = Util.comment
let empty      = Util.empty

let colon      = Sep.colon
let comma      = Sep.comma

let sto_to_spc = store Rx.space_in

let word    = Rx.word
let password = /[A-Za-z0-9_.!*\/$-]*/
let integer = Rx.integer

(************************************************************************
 *                              ENTRIES
 *************************************************************************)

let user      = [ label "user" . store word ]
let user_list = Build.opt_list user comma
let params    = [ label "password" . store password  . colon ]
                . [ label "admins" . user_list? . colon ]
                . [ label "members" . user_list? ]
let entry     = Build.key_value_line word colon params

(************************************************************************
 *                               LENS
 *************************************************************************)

let lns        = (comment|empty|entry) *

let filter
               = incl "/etc/gshadow"
               . Util.stdexcl

let xfm        = transform lns filter
```

### 12.4.6 0.9.0/lenses/hosts_access.aug

```
(*
Module: Hosts_Access
  Parses /etc/hosts.{allow,deny}

Author: Raphael Pinson <raphink@gmail.com>

About: Reference
  This lens tries to keep as close as possible to ‘man 5 hosts_access‘ and [WRAP]
‘man 5 hosts_options‘ where possible.

About: License
   This file is licenced under the LGPL v2+, like the rest of Augeas.

About: Lens Usage
   To be documented

About: Configuration files
   This lens applies to /etc/hosts.{allow,deny}. See <filter>.
*)
```

```
module Hosts_Access =

autoload xfm

(************************************************************************
 * Group:                 USEFUL PRIMITIVES
 *************************************************************************)

(* View: colon *)
let colon = del /[ \t]*(\\\\[ \t]*\n[ \t]+)?:[ \t]*(\\\\[ \t]*\n[ \t]+)?/ "[WRAP]
: "

(* Variable: comma_sep *)
let comma_sep = /([ \t]|(\\\\\n))*,([ \t]|(\\\\\n))*/

(* Variable: ws_sep *)
let ws_sep = / +/

(* View: list_sep *)
let list_sep = del ( comma_sep | ws_sep ) ", "

(* View: list_item *)
let list_item = store ( Rx.word - /EXCEPT/i )

(* View: client_host_item
   Allows @ for netgroups, supports [ipv6] syntax *)
let client_host_item =
  let client_hostname_rx = /[A-Za-z0-9_.@?*-][A-Za-z0-9_.?*-]*/ in
  let client_ipv6_rx = "[" . /[A-Za-z0-9:?*%]+/ . "]" in
    let client_host_rx = client_hostname_rx | client_ipv6_rx in
    let netmask = [ Util.del_str "/" . label "netmask" . store Rx.word ] in
      store ( client_host_rx - /EXCEPT/i ) . netmask?

(* View: client_file_item *)
let client_file_item =
  let client_file_rx = /\/[^ \t\n,:]+/ in
    store ( client_file_rx - /EXCEPT/i )

(* Variable: option_kw
   Since either an option or a shell command can be given, use an explicit [WRAP]
list
   of known options to avoid misinterpreting a command as an option *)
let option_kw = "severity"
              | "spawn"
              | "twist"
              | "keepalive"
              | "linger"
              | "rfc931"
              | "banners"
              | "nice"
              | "setenv"
              | "umask"
              | "user"
              | /allow/i
              | /deny/i
```

```
(* Variable: shell_command_rx *)
let shell_command_rx = /[^ \t\n:][^\n]*[^ \t\n]|[^ \t\n:\\\\]/
                         - ( option_kw . /.*/ )

(* View: sto_to_colon
   Allows escaped colon sequences *)
let sto_to_colon = store /[^ \t\n:=][^\n:]*((\\\\:|\\\\[ \t]*\n[ \t]+)[^\n:[WRAP]
]*)*[^ \\\t\n:]|[^ \t\n:\\\\]/

(* View: except
 * The except operator makes it possible to write very compact rules.
 *)
let except (lns:lens) = [ label "except" . Sep.space
                        . del /except/i "EXCEPT"
                        . Sep.space . lns ]

(************************************************************************
 * Group:                 ENTRY TYPES
 *************************************************************************)

(* View: daemon *)
let daemon =
  let host = [ label "host"
             . Util.del_str "@"
             . list_item ] in
   [ label "process"
   . list_item
   . host? ]

(* View: daemon_list
    A list of <daemon>s *)
let daemon_list = Build.opt_list daemon list_sep

(* View: client *)
let client =
  let user = [ label "user"
             . list_item
             . Util.del_str "@" ] in
    [ label "client"
    . user?
    . client_host_item ]

(* View: client_file *)
let client_file = [ label "file" . client_file_item ]

(* View: client_list
    A list of <client>s *)
let client_list = Build.opt_list ( client | client_file ) list_sep

(* View: option
   Optional extensions defined in hosts_options(5) *)
let option = [ key option_kw
             . ( del /([ \t]*=[ \t]*|[ \t]+)/ " " . sto_to_colon )? ]

(* View: shell_command *)
let shell_command = [ label "shell_command"
                    . store shell_command_rx ]
```

```
(* View: entry *)
let entry = [ seq "line"
            . daemon_list
            . (except daemon_list)?
            . colon
            . client_list
            . (except client_list)?
            . ( (colon . option)+ | (colon . shell_command)? )
            . Util.eol ]

(************************************************************************
 * Group:                 LENS AND FILTER
 *************************************************************************)

(* View: lns *)
let lns = (Util.empty | Util.comment | entry)*

(* View: filter *)
let filter = incl "/etc/hosts.allow"
           . incl "/etc/hosts.deny"

let xfm = transform lns filter
```

### 12.4.7 0.9.0/lenses/kdc.aug

```
module Kdc =

autoload xfm

let comment = Krb5.comment
let empty = Krb5.empty

let simple_section = Krb5.simple_section
let kdcdefaults =
  simple_section "kdcdefaults" /kdc_ports|kdc_tcp_ports/

let realm_re = Krb5.realm_re
let entry = Krb5.entry
let eq = Krb5.eq
(* the Krb5.eq_openbr didn't have a newline at the end *)
let eq_openbr = del /[ \t]*=[ \t\n]*\{([ \t]*\n)*/ " = {\n\n"
let closebr = Krb5.closebr
let indent = Krb5.indent
let eol = Krb5.eol
let record = Krb5.record
let realms_enctypes = [ indent . key "supported_enctypes" . eq .
        [ label "type" . store /[^ \t\n#]+/ . Util.del_ws_spc ] * .
        [ label "type" . store /[^ \t\n#]+/ . eol ] ]

let realms =
  let simple_option = /master_key_type|acl_file|dict_file|admin_keytab/ in
  let list_option = /supported_enctypes/ in
  let soption = entry simple_option eq comment in
  let realm = [ indent . label "realm" . store realm_re .
                  eq_openbr . eol . (soption|realms_enctypes)* . closebr . [WRAP]
eol ] in
```

```
    record "realms" (realm|comment)


let lns = (comment|empty)* .
  (kdcdefaults|realms)*

let xfm = transform lns (incl "/var/kerberos/krb5kdc/kdc.conf")
```

### 12.4.8 0.9.0/lenses/krb5.aug

```
module Krb5 =

autoload xfm

let comment = Inifile.comment "#" "#"
let empty = Inifile.empty
let eol = Inifile.eol
let dels = Util.del_str

let indent = del /[ \t]*/ ""
let eq = del /[ \t]*=[ \t]*/ " = "
let eq_openbr = del /[ \t]*=[ \t\n]*\{([ \t]*\n)*/ " = {"
let closebr = del /[ \t]*\}/ "}"

(* These two regexps for realms and apps are not entirely true
   - strictly speaking, there's no requirement that a realm is all upper ca[WRAP]
se
   and an application only uses lowercase. But it's what's used in practice[WRAP]
.

   Without that distinction we couldn't distinguish between applications
   and realms in the [appdefaults] section.
*)

let realm_re = /[A-Z][.a-zA-Z0-9-]*/
let app_re = /[a-z][a-zA-Z0-9_]*/
let name_re = /[.a-zA-Z0-9_-]+/

let value = store /[^;# \t\n{}]+/
let entry (kw:regexp) (sep:lens) (comment:lens)
    = [ indent . key kw . sep . value . (comment|eol) ] | comment

let simple_section (n:string) (k:regexp) =
  let title = Inifile.indented_title n in
  let entry = entry k eq comment in
    Inifile.record title entry

let record (t:string) (e:lens) =
  let title = Inifile.indented_title t in
    Inifile.record title e

let libdefaults =
  let option = entry (name_re - "v4_name_convert") eq comment in
  let subsec = [ indent . key /host|plain/ . eq_openbr .
                   (entry name_re eq comment)* . closebr . eol ] in
  let v4_name_convert = [ indent . key "v4_name_convert" . eq_openbr .
                          subsec* . closebr . eol ] in
  record "libdefaults" (option|v4_name_convert)

let login =
  let keys = /krb[45]_get_tickets|krb4_convert|krb_run_aklog/
    |/aklog_path|accept_passwd/ in
    simple_section "login" keys

let appdefaults =
  let option = entry (name_re - "realm" - "application") eq comment in
  let realm = [ indent . label "realm" . store realm_re .
                  eq_openbr . option* . closebr . eol ] in
  let app = [ indent . label "application" . store app_re .
                eq_openbr . (realm|option)* . closebr . eol] in
    record "appdefaults" (option|realm|app)

let realms =
  let simple_option = /kdc|admin_server|database_module|default_domain/
      |/v4_realm|auth_to_local(_names)?|master_kdc|kpasswd_server/
      |/admin_server/ in
  let subsec_option = /v4_instance_convert/ in
  let option = entry simple_option eq comment in
  let subsec = [ indent . key subsec_option . eq_openbr .
                   (entry name_re eq comment)* . closebr . eol ] in
(* ************************* Changes applied by AFSEO are below ***********[WRAP]
 *)
  let realm = [ indent . label "realm" . store realm_re .
(*                          vvvvv                                          [WRAP]
 *)
                  eq_openbr . eol . (option|subsec)* . closebr . eol ] in
(*                          ^^^^^                                          [WRAP]
 *)
(* ************************* Changes applied by AFSEO are above ***********[WRAP]
 *)
    record "realms" (realm|comment)

let domain_realm =
  simple_section "domain_realm" name_re

let logging =
  let keys = /kdc|admin_server|default/ in
  let xchg (m:regexp) (d:string) (l:string) =
    del m d . label l in
  let xchgs (m:string) (l:string) = xchg m m l in
  let dest =
    [ xchg /FILE[=:]/ "FILE=" "file" . value ]
    |[ xchgs "STDERR" "stderr" ]
    |[ xchgs "CONSOLE" "console" ]
    |[ xchgs "DEVICE=" "device" . value ]
    |[ xchgs "SYSLOG" "syslog" .
         ([ xchgs ":" "severity" . store /[A-Za-z0-9]+/ ].
          [ xchgs ":" "facility" . store /[A-Za-z0-9]+/ ]?)? ] in
  let entry = [ indent . key keys . eq . dest . (comment|eol) ] | comment i[WRAP]
n
    record "logging" entry

let capaths =
  let realm = [ indent . key realm_re .
```

```
                  eq_openbr .
                  (entry realm_re eq comment)* . closebr . eol ] in
    record "capaths" (realm|comment)

let dbdefaults =
  let keys = /database_module|ldap_kerberos_container_dn|ldap_kdc_dn/
    |/ldap_kadmind_dn|ldap_service_password_file|ldap_servers/
    |/ldap_conns_per_server/ in
    simple_section "dbdefaults" keys

let dbmodules =
  let keys = /db_library|ldap_kerberos_container_dn|ldap_kdc_dn/
    |/ldap_kadmind_dn|ldap_service_password_file|ldap_servers/
    |/ldap_conns_per_server/ in
    simple_section "dbmodules" keys

(* This section is not documented in the krb5.conf manpage,
   but the Fermi example uses it. *)
let instance_mapping =
  let value = dels "\"" . store /[^;# \t\n{}]*/ . dels "\"" in
  let map_node = label "mapping" . store /[a-zA-Z0-9\/*]+/ in
  let mapping = [ indent . map_node . eq .
                    [ label "value" . value ] . (comment|eol) ] in
  let instance = [ indent . key name_re .
                     eq_openbr . (mapping|comment)* . closebr . eol ] in
    record "instancemapping" instance

let kdc =
  simple_section "kdc" /profile/

let lns = (comment|empty)* .
  (libdefaults|login|appdefaults|realms|domain_realm
  |logging|capaths|dbdefaults|dbmodules|instance_mapping|kdc)*

let xfm = transform lns (incl "/etc/krb5.conf")
```

### 12.4.9 0.9.0/lenses/libreport_plugins.aug

```
module Libreport_plugins =

autoload xfm

let entry = Build.key_value_line /[A-Za-z]+/ Sep.equal (store /[^\n]*[^ \t\[WRAP]
n]+/)

let lns = ( Util.comment | Util.empty | entry ) *

let filter = (incl "/etc/libreport/plugins/*.conf") . Util.stdexcl
let xfm = transform lns filter
```

### 12.4.10 0.9.0/lenses/mimetypes.aug

```
module Mimetypes =
    autoload xfm

    (* RFC 2045, Page 11. Closing square bracket moved out of sequence to
       satisfy regex syntax. token_first excludes pound signs so as not to
       overlap with the syntax for comments. *)
    let token =
           let first = /[^]#()<>@,;:\\"\/[?= \t\n]/
        in let rest  =  /[^]()<>@,;:\\"\/[?= \t\n]*/
        in first . rest
    (* We can't use the mime type as a key, because it has a slash in it *)
    let mime_type = store (token . "/" . token)
    (* This will split up rules wrong if you use spaces within a rule, e.g.
    "ascii(34, 3)" or "string(34,'foo bar')". But all the rules I've ever s[WRAP]
een
    were just filename extensions, so this won't fail until people forget w[WRAP]
hat
    it is and have to dig to find it. *)
    let a_rule = [ Util.del_ws_spc . label "rule" . store /[^ \t\n]+/ ]
    let rules = [ label "rules" . mime_type . (a_rule *) . Util.eol ]
    let line = ( rules | Util.comment | Util.empty )
    let lns = ( line * )

    let xfm = transform lns (incl "/etc/mime.types")
```

## 12.4.11 0.9.0/lenses/pg_ident.aug

```
module Pg_Ident =
    autoload xfm
    let identifier = store /[a-z_][^ \t\n#]*/
    let record = [ seq "entries" .
                   [ label "map" . identifier ] .
                   Util.del_ws_spc .
                   [ label "os_user" . identifier ] .
                   Util.del_ws_spc .
                   [ label "db_user" . identifier ] .
                   Util.eol
                 ]
    let empty = Util.empty
    let comment = Util.comment
    let line = empty | comment | record
    let lns = line *
    let xfm = transform lns (incl "/var/lib/pgsql/data/pg_ident.conf")
```

## 12.4.12 0.9.0/lenses/postgresql.aug

```
module Postgresql =
    autoload xfm

    let comment = Inifile.comment "#" "#"
    let empty = Inifile.empty
    let eq = del /[ \t]*=/ " ="
    let entry = IniFile.entry IniFile.entry_re eq comment

    let lns = ( entry | empty ) *

    let xfm = transform lns (incl "/var/lib/pgsql/*/postgresql.conf")
```

### 12.4.13 0.9.0/lenses/someautomountmaps.aug

```
(* This lens does NOT parse all automount maps!
   It can deal with maps which are scripts (start with a hashbang), but not
   with multiple mounts nor with line continuations.
*)
module Someautomountmaps =
    autoload xfm

    let eol = Util.eol
    let script_content = [ label "script_content" . store /#!(.*[\n]*)*/ ]
    (* This is the same as Util.comment, except that it denies hashbangs.
       As a side effect it also denies comments that begin with a bang, lik[WRAP]
e
       "# !blabalabl". Sloppy, but it works here now, and that's the point [WRAP]
of
       this whole file. *)
    let indent = Util.indent
    let comment =
      [ indent . label "#comment" . del /#[ \t]*/ "# "
          . store /([^! \t\n].*[^ \t\n]|[^! \t\n])/ . eol ]
    (*                ^-- like so *)

    let automount_key = store /[^# \t\n]+/
    let options = [ label "options" .
                    ( del "-" "-" .
                      store /[^ \t\n]+/ .
                      Util.del_ws_spc ) ? ]
    let location = [ label "location" . store /[^ \t\n]+/ ]
    let entry = [ label "entry" .
                   automount_key . Util.del_ws_spc .
                   options .
                   location . eol ]

    let lns = script_content |
              ( comment | Util.empty | entry ) *

    let relevant = (incl "/etc/auto.*") .
                   (excl "/etc/auto.master") .
                   Util.stdexcl
    let xfm = transform lns relevant
```

### 12.4.14 0.9.0/lenses/sos.aug

```
module Sos =
 autoload xfm
 let lns = Puppet.lns
 let xfm = transform lns (incl "/etc/sos.conf")
```

### 12.4.15 0.9.0/lenses/ssh.aug

```
(*
Module: Ssh
  Parses ssh client configuration

Author: Jiri Suchomel <jsuchome@suse.cz>
About: Reference
    ssh_config man page

About: License
    This file is licensed under the GPL.

About: Lens Usage
  Sample usage of this lens in augtool

augtool> set /files/etc/ssh/ssh_config/Host example.com
augtool> set /files/etc/ssh/ssh_config/Host[.='example.com']/RemoteForward/[WRAP]
machine1:1234 machine2:5678
augtool> set /files/etc/ssh/ssh_config/Host[.='example.com']/Ciphers/1 aes1[WRAP]
28-ctr
augtool> set /files/etc/ssh/ssh_config/Host[.='example.com']/Ciphers/2 aes1[WRAP]
92-ctr

*)

module Ssh =
    autoload xfm

    let eol = del /[ \t]*\n/ "\n"
    let spc = Util.del_ws_spc

    let key_re = /[A-Za-z0-9]+/
               - /SendEnv|Host|ProxyCommand|RemoteForward|LocalForward|MACs[WRAP]
|Ciphers/

    let comment = Util.comment
    let empty = Util.empty
    let comma = Util.del_str ","
    let indent = Util.indent
    let value_to_eol = store /([^ \t\n].*[^ \t\n]|[^ \t\n])/
    let value_to_spc = store /[^ \t\n]+/
    let value_to_comma = store /[^, \t\n]+/

    let array_entry (k:string) =
        [ indent . key k . counter k . [ spc . seq k . value_to_spc]* . eol[WRAP]
 ]

    let commas_entry (k:string) =
[ key k . counter k . spc .
    [ seq k . value_to_comma] . ([ seq k . comma . value_to_comma])* . eol[WRAP]
 ]

    let send_env = array_entry "SendEnv"

    let proxy_command = [ indent . key "ProxyCommand" . spc . value_to_eol [WRAP]
. eol ]

    let fw_entry (k:string) = [ indent . key k . spc .
[ key /[^ \t\n\/]+/ . spc . value_to_eol . eol ]]

    let remote_fw = fw_entry "RemoteForward"
    let local_fw = fw_entry "LocalForward"
    let ciphers = commas_entry "Ciphers"
    let macs = commas_entry "MACs"

    let other_entry =
[ indent . key key_re . spc . value_to_spc . eol ]

    let entry = (comment | empty
| send_env
| proxy_command
| remote_fw
| local_fw
| macs
| ciphers
| other_entry)

    let host = [ key "Host" . spc . value_to_eol . eol . entry* ]

    let lns = entry* . host*

    let xfm = transform lns (incl "/etc/ssh/ssh_config" .
                             incl (Sys.getenv("HOME") . "/.ssh/config"))
```

### 12.4.16 0.9.0/lenses/subject_mapping.aug

```
(* Parse pam_pkcs11 subject_mapping file
   File is of the format:

   Certificate Distinguished Name, With Spaces and Commas, Bla Bla. -> user[WRAP]
name

   We're interested in preserving the one-to-one property, that is, that fo[WRAP]
r a
   given username there is only one certificate. Because of this, and becau[WRAP]
se
   the username is shorter and easier to type, we make the username the key
   instead of the certificate distinguished name.
*)

module Subject_mapping =
    autoload xfm
    (* can't have slashes in keys, that's another reason to make the userna[WRAP]
me
       the key *)
    let username = key /[^>\/ \t\n-]+/
    let arrow = del /[ \t]*->[ \t]*/ " -> "
    let certdn = store /[^ \t\n]+([ \t]+[^ \t\n]+)*/
    let line = [ certdn . arrow . username . Util.eol ]

    let lns = line *

    let relevant = (incl "/etc/pam_pkcs11/subject_mapping")
    let xfm = transform lns relevant
```

### 12.4.17 0.9.0/lenses/subversion.aug

```
(* it's just an ini file. sections ("titles") are required *)
```

```
module Subversion =
    autoload xfm

    let comment = IniFile.comment "#" "#"
    let sep = IniFile.sep "=" "="
    let entry = IniFile.indented_entry IniFile.entry_re sep comment
    let title = IniFile.indented_title IniFile.record_re
    let record = IniFile.record title entry

    let lns = IniFile.lns record comment

    let relevant = ( incl "/etc/subversion/servers" ) .
                   ( incl "/etc/subversion/config" )

    let xfm = transform lns relevant
```

### 12.4.18 0.9.0/lenses/tracini.aug

```
(* This began as a copy of <Puppet> *)

module Tracini =
  autoload xfm

(************************************************************************
 * INI File settings
 *
 * puppet.conf only supports "# as commentary and "=" as separator
 *************************************************************************)
let comment    = IniFile.comment "#" "#"
let sep        = IniFile.sep "=" "="


(************************************************************************
 *                        ENTRY
 * puppet.conf uses standard INI File entries
 *************************************************************************)
(* began with IniFile.entry_re *)
(* added star as a valid non-first char in entry keys *)
(* allowed single-character entry keys *)
let entry_re           = ( /[A-Za-z][A-Za-z0-9*\._-]*/ )
let entry   = IniFile.indented_entry entry_re sep comment


(************************************************************************
 *                        RECORD
 * puppet.conf uses standard INI File records
 *************************************************************************)
let title   = IniFile.indented_title IniFile.record_re
let record  = IniFile.record title entry


(************************************************************************
 *                        LENS & FILTER
 * puppet.conf uses standard INI File records
 *************************************************************************)
let lns     = IniFile.lns record comment
let filter = (incl "/var/www/tracs/*/conf/trac.ini")

let xfm = transform lns filter
```

### 12.4.19 0.9.0/lenses/up2date.aug

```
module Up2date =
    autoload xfm

    (* funky syntax: this matches one or more of a-z, A-Z, [ or ]. *)
    let akey = /[]a-zA-Z[]+/
    let avalue = /[^ \t\n]*([ \t]+[^ \t\n]+)*/
    let setting = Build.key_value_line akey (del "=" "=") (store avalue)
    let lns = ( Util.empty | Util.comment | setting ) *

    let xfm = transform lns (incl "/etc/sysconfig/rhn/up2date")
```

### 12.4.20 0.9.0/lenses/upstartinit.aug

```
(* Upstart init configuration files such as found in /etc/init *)

module Upstartinit =
    autoload xfm

    let eol = Util.eol
    let rest_of_line = /[^ \t\n]+([ \t]+[^ \t\n]+)*/
    let whole_line_maybe_indented = /[ \t]*[^ \t\n]+([ \t]+[^ \t\n]+)*/
    let no_params = [ key "task" . eol ]

    let param_is_rest_of_line (thekey:regexp) =
        Build.key_value_line thekey
                             Util.del_ws_spc
                             (store rest_of_line)

    let respawn = [ key "respawn" .
          (Util.del_ws_spc . store rest_of_line)? . eol ]


    let one_params = param_is_rest_of_line
          ( "start"
          | "stop"
          | "env"
          | "export"
          | "normal exit"
          | "instance"
          | "description"
          | "author"
          | "version"
          | "emits"
          | "console"
          | "umask"
          | "nice"
          | "oom"
          | "chroot"
          | "chdir"
          | "limit"
          | "unlimited"
          | "kill timeout"
          | "expect"
          | "usage"
          )

    (* exec and script are valid both at the top level and as a parameter o[WRAP]
f a
    lifecycle keyword *)
    let exec = param_is_rest_of_line "exec"

    let script_line = [ seq "line" .
                        store ( whole_line_maybe_indented - "end script" ) [WRAP]
.
                        eol ] |
                      [ seq "line" . eol]
    let end_script = del "end script\n" "end script\n"
    let script = [ key "script" . eol . script_line * . end_script ]

    let lifecycle = [ key /(pre|post)-(start|stop)/ .  Util.del_ws_spc . ( [WRAP]
exec | script ) ]

    let lns = ( Util.empty
              | Util.comment
              | script
              | exec
              | lifecycle
              | no_params
              | one_params
              | respawn
              ) *

    let relevant = (incl "/etc/init/*.conf") . Util.stdexcl
    let xfm = transform lns relevant
```

### 12.4.21 0.9.0/tests/test_abrt.aug

```
module Test_abrt =
    let lns = Abrt.lns
    test lns get "
# Configuration file for CCpp hook

# If you also want to dump file named \"core\"
# in crashed process' current dir, set to \"yes\"
MakeCompatCore = yes

# Do you want a copy of crashed binary be saved?
# (useful, for example, when _deleted binary_ segfaults)
SaveBinaryImage = no

# Used for debugging the hook
#VerboseLog = 2

# Specify where you want to store debuginfos (default: /var/cache/abrt-di)
#
#DebuginfoLocation = /var/cache/abrt-di
```

```
" = (
  {  }
  { "#comment" = "Configuration file for CCpp hook" }
  {  }
  { "#comment" = "If you also want to dump file named \"core\"" }
  { "#comment" = "in crashed process' current dir, set to \"yes\"" }
  { "MakeCompatCore" = "yes" }
  {  }
  { "#comment" = "Do you want a copy of crashed binary be saved?" }
  { "#comment" = "(useful, for example, when _deleted binary_ segfaults)" }
  { "SaveBinaryImage" = "no" }
  {  }
  { "#comment" = "Used for debugging the hook" }
  { "#comment" = "VerboseLog = 2" }
  {  }
  { "#comment" = "Specify where you want to store debuginfos (default: /var[WRAP]
/cache/abrt-di)" }
  { "#comment" }
  { "#comment" = "DebuginfoLocation = /var/cache/abrt-di" }
)
```

### 12.4.22 0.9.0/tests/test_auditdconf.aug

```
module Test_auditdconf =
    let empty = Auditdconf.empty
    let entry = Auditdconf.entry
    let lns = Auditdconf.lns

    test empty get "\n" = {}
    test entry get "\n" = *
    test lns get "#
# This file controls the configuration of the audit daemon
#

log_file = /var/log/audit/audit.log
log_format = RAW
log_group = root
priority_boost = 4
flush = INCREMENTAL
freq = 20
num_logs = 4
disp_qos = lossy
" = (
  { "#comment" }
  { "#comment" = "This file controls the configuration of the audit daemon"[WRAP]
 }
  { "#comment" }
  {  }
  { "log_file" = "/var/log/audit/audit.log" }
  { "log_format" = "RAW" }
  { "log_group" = "root" }
  { "priority_boost" = "4" }
  { "flush" = "INCREMENTAL" }
  { "freq" = "20" }
  { "num_logs" = "4" }
  { "disp_qos" = "lossy" }
)
```

### 12.4.23 0.9.0/tests/test_automaster.aug

```
module Test_automaster =
    let map_param = Automaster.map_param
    let map_record = Automaster.map_record
    let lns = Automaster.lns

    test map_param get "file:/bla/blu" =
        ( { "type" = "file" } { "name" = "/bla/blu" } )
    test map_param get "yp,hesiod:/bla/blu" =
        ( { "type" = "yp" }
          { "format" = "hesiod" }
          { "name" = "/bla/blu" } )
    test map_param get "bla" = { "name" = "bla" }
    test map_record get "/net /etc/auto.net\n" =
        { "map" = "/net"
            { "name" = "/etc/auto.net" } }

    test lns get "# c\n+auto.master\n/net /etc/auto.net\n\n" = (
        { "#comment" = "c" }
        { "include" = "auto.master" }
        { "map" = "/net"
            { "name" = "/etc/auto.net" }
        }
        {  } )

    test lns get "# c
+auto.master
# blank line


/net /etc/auto.net
/foo bla
" = (
  { "#comment" = "c" }
  { "include" = "auto.master" }
  { "#comment" = "blank line" }
  {  }
  {  }
  { "map" = "/net"
    { "name" = "/etc/auto.net" }
  }
  { "map" = "/foo"
    { "name" = "bla" }
  }
)

    test lns get "#
# Sample auto.master file
# This is an automounter map and it has the following format
# key [ -mount-options-separated-by-comma ] location
# For details of the format look at autofs(5).
#
/misc   /etc/auto.misc
#
```

```
# NOTE: mounts done from a hosts map will be mounted with the
#       \"nosuid\" and \"nodev\" options unless the \"suid\" and \"dev\"
#       options are explicitly given.
#
/net    -hosts
#
# Include central master map if it can be found using
# nsswitch sources.
#
# Note that if there are entries for /net or /misc (as
# above) in the included master map any keys that are the
# same will not be seen as the first read key seen takes
# precedence.
#
+auto.master
" = (
  {  }
  { "#comment" = "Sample auto.master file" }
  { "#comment" = "This is an automounter map and it has the following forma[WRAP]
t" }
  { "#comment" = "key [ -mount-options-separated-by-comma ] location" }
  { "#comment" = "For details of the format look at autofs(5)." }
  {  }
  { "map" = "/misc"
    { "name" = "/etc/auto.misc" }
  }
  {  }
  { "#comment" = "NOTE: mounts done from a hosts map will be mounted with t[WRAP]
he" }
  { "#comment" = "\"nosuid\" and \"nodev\" options unless the \"suid\" and [WRAP]
\"dev\"" }
  { "#comment" = "options are explicitly given." }
  {  }
  { "map" = "/net"
    { "name" = "-hosts" }
  }
  {  }
  { "#comment" = "Include central master map if it can be found using" }
  { "#comment" = "nsswitch sources." }
  {  }
  { "#comment" = "Note that if there are entries for /net or /misc (as" }
  { "#comment" = "above) in the included master map any keys that are the" [WRAP]
}
  { "#comment" = "same will not be seen as the first read key seen takes" }
  { "#comment" = "precedence." }
  {  }
  { "include" = "auto.master" }
)
```

### 12.4.24 0.9.0/tests/test_gshadow.aug

```
module Test_gshadow =
   let lns = Gshadow.lns
   let entry = Gshadow.entry
   test entry get "root:::\n" =
  { "root"
    { "password" = "" }
    { "admins" }
    { "members" }
  }

   test entry get "bin:::bin,daemon\n" =
  { "bin"
    { "password" = "" }
    { "admins" }
    { "members"
      { "user" = "bin" }
      { "user" = "daemon" }
    }
  }

   test entry get "dbus:!::\n" =
  { "dbus"
    { "password" = "!" }
    { "admins" }
    { "members" }
  }

   test entry get "ntp:!:foo,bar:baz,bletch\n" =
  { "ntp"
    { "password" = "!" }
    { "admins"
      { "user" = "foo" }
      { "user" = "bar" }
    }
    { "members"
      { "user" = "baz" }
      { "user" = "bletch" }
    }
  }

   test entry get "fooz:$5$GQPAI/174dH/Q$dQtmrhcGuolwm7DlKVFkeH.VCWbH1/XTYk[WRAP]
XU83WkIO9::\n" =
  { "fooz"
    { "password" = "$5$GQPAI/174dH/Q$dQtmrhcGuolwm7DlKVFkeH.VCWbH1/XTYkXU83[WRAP]
WkIO9" }
    { "admins" }
    { "members" }
  }




   test lns get
"root:::
bin:::bin,daemon
dbus:!::
ntp:!:foo,bar:baz,bletch
fooz:$5$GQPAI/174dH/Q$dQtmrhcGuolwm7DlKVFkeH.VCWbH1/XTYkXU83WkIO9::
" =
  { "root"
    { "password" = "" }
    { "admins" }
    { "members" }
  }
  { "bin"
    { "password" = "" }
    { "admins" }
    { "members"
      { "user" = "bin" }
      { "user" = "daemon" }
    }
  }
  { "dbus"
    { "password" = "!" }
    { "admins" }
    { "members" }
  }
  { "ntp"
    { "password" = "!" }
    { "admins"
      { "user" = "foo" }
      { "user" = "bar" }
    }
    { "members"
      { "user" = "baz" }
      { "user" = "bletch" }
    }
  }
  { "fooz"
    { "password" = "$5$GQPAI/174dH/Q$dQtmrhcGuolwm7DlKVFkeH.VCWbH1/XTYkXU83[WRAP]
WkIO9" }
    { "admins" }
    { "members" }
  }
```

### 12.4.25 0.9.0/tests/test_kdc.aug

```
module Test_kdc =
   let lns = Kdc.lns
   let realms_enctypes = Kdc.realms_enctypes
   test realms_enctypes get " supported_enctypes = aes256-cts:normal aes128[WRAP]
-cts:normal des3-hmac-sha1:normal arcfour-hmac:normal des-hmac-sha1:normal [WRAP]
des-cbc-md5:normal des-cbc-crc:normal
" =
  { "supported_enctypes"
    { "type" = "aes256-cts:normal" }
    { "type" = "aes128-cts:normal" }
    { "type" = "des3-hmac-sha1:normal" }
    { "type" = "arcfour-hmac:normal" }
    { "type" = "des-hmac-sha1:normal" }
    { "type" = "des-cbc-md5:normal" }
    { "type" = "des-cbc-crc:normal" }
  }


   test lns get "
[kdcdefaults]
 kdc_ports = 88
 kdc_tcp_ports = 88

[realms]
 EXAMPLE.COM = {
  #master_key_type = aes256-cts
  acl_file = /var/kerberos/krb5kdc/kadm5.acl
  dict_file = /usr/share/dict/words
  admin_keytab = /var/kerberos/krb5kdc/kadm5.keytab
  supported_enctypes = aes256-cts:normal aes128-cts:normal des3-hmac-sha1:n[WRAP]
ormal arcfour-hmac:normal des-hmac-sha1:normal des-cbc-md5:normal des-cbc-c[WRAP]
rc:normal
 }
" = (
  {  }
  { "kdcdefaults"
    { "kdc_ports" = "88" }
    { "kdc_tcp_ports" = "88" }
    {  }
  }
  { "realms"
    { "realm" = "EXAMPLE.COM"
      { "#comment" = "master_key_type = aes256-cts" }
      { "acl_file" = "/var/kerberos/krb5kdc/kadm5.acl" }
      { "dict_file" = "/usr/share/dict/words" }
      { "admin_keytab" = "/var/kerberos/krb5kdc/kadm5.keytab" }
      { "supported_enctypes"
        { "type" = "aes256-cts:normal" }
        { "type" = "aes128-cts:normal" }
        { "type" = "des3-hmac-sha1:normal" }
        { "type" = "arcfour-hmac:normal" }
        { "type" = "des-hmac-sha1:normal" }
        { "type" = "des-cbc-md5:normal" }
        { "type" = "des-cbc-crc:normal" }
      }
    }
  }
)

    test lns put "" after
        set "realms/realm[999]" "FOO.BAR.EXAMPLE.COM"
    = "[realms]
FOO.BAR.EXAMPLE.COM = {
}
"

    test lns put "[realms]
FOO.BAR.EXAMPLE.COM = {
}" after
        set "realms/realm[.='FOO.BAR.EXAMPLE.COM']/acl_file" "/var/kerberos[WRAP]
/krb5kdc/kadm5.acl"
    = "[realms]
FOO.BAR.EXAMPLE.COM = {
acl_file = /var/kerberos/krb5kdc/kadm5.acl
}
"
```

### 12.4.26 0.9.0/tests/test_libreport_plugins.aug

```
module Test_libreport_plugins =

    let lns = Libreport_plugins.lns
    let entry = Libreport_plugins.entry

    test entry get "Foo=bar\n" = ( { "Foo" = "bar" } )
    test lns get "
# String parameters:

Subject=bla
# EmailFrom=
" = (
  { }
  { "#comment" = "String parameters:" }
  { }
  { "Subject" = "bla" }
  { "#comment" = "EmailFrom=" }
)
```

### 12.4.27 0.9.0/tests/test_mimetypes.aug

```
module Test_mimetypes =
    let mime_type = Mimetypes.mime_type
    let rules = Mimetypes.rules
    let lns = Mimetypes.lns

    test [ mime_type ] get "text/plain" = { = "text/plain" }
    test [ mime_type ] get "application/beep+xml" = { = "application/beep+x[WRAP]
ml" }
    test [ mime_type ] get "application/vnd.fdf" = { = "application/vnd.fdf[WRAP]
" }
    (* who in their right mind made this mime type?! ... oh wait, they were[WRAP]
n’t,
       it’s microsoft *)
    test [ mime_type ] get
        "application/vnd.openxmlformats-officedocument.wordprocessingml.doc[WRAP]
ument" =
        { = "application/vnd.openxmlformats-officedocument.wordprocessingml[WRAP]
.document" }
    test rules get "text/plain txt\n" =
        { "rules" = "text/plain"
          { "rule" = "txt" } }
    test rules get "application/vnd.openxmlformats-officedocument.wordproce[WRAP]
ssingml.document docx\n" =
        { "rules" = "application/vnd.openxmlformats-officedocument.wordproc[WRAP]
essingml.document"
          { "rule" = "docx" } }
    test rules get "video/mpeg                      mpeg mpg mpe\n" =
        { "rules" = "video/mpeg"
          { "rule" = "mpeg" }
          { "rule" = "mpg" }
          { "rule" = "mpe" } }
    test lns get "
# This is a comment. I love comments.
# This file controls what Internet media types are sent to the client for
# given file extension(s).  Sending the correct media type to the client
# is important so they know how to handle the content of the file.
# Extra types can either be added here or by using an AddType directive
# in your config files. For more information about Internet media types,
# please read RFC 2045, 2046, 2047, 2048, and 2077.  The Internet media typ[WRAP]
e
# registry is at <http://www.iana.org/assignments/media-types/>.

# MIME type                     Extension
application/EDI-Consent
application/andrew-inset        ez
application/mac-binhex40        hqx
application/mac-compactpro      cpt
application/octet-stream        bin dms lha lzh exe class so dll img iso
application/ogg                 ogg

" = (
  {  }
  { "#comment" = "This is a comment. I love comments." }
  {  }
  { "#comment" = "This file controls what Internet media types are sent to [WRAP]
the client for" }
  { "#comment" = "given file extension(s).  Sending the correct media type [WRAP]
to the client" }
  { "#comment" = "is important so they know how to handle the content of th[WRAP]
e file." }
  { "#comment" = "Extra types can either be added here or by using an AddTy[WRAP]
pe directive" }
  { "#comment" = "in your config files. For more information about Internet[WRAP]
 media types," }
  { "#comment" = "please read RFC 2045, 2046, 2047, 2048, and 2077.  The In[WRAP]
ternet media type" }
  { "#comment" = "registry is at <http://www.iana.org/assignments/media-typ[WRAP]
es/>." }
  {  }
  { "#comment" = "MIME type                     Extension" }
  { "rules" = "application/EDI-Consent" }
  { "rules" = "application/andrew-inset"
    { "rule" = "ez" }
  }
  { "rules" = "application/mac-binhex40"
    { "rule" = "hqx" }
  }
  { "rules" = "application/mac-compactpro"
    { "rule" = "cpt" }
  }
  { "rules" = "application/octet-stream"
    { "rule" = "bin" }
    { "rule" = "dms" }
    { "rule" = "lha" }
    { "rule" = "lzh" }
    { "rule" = "exe" }
    { "rule" = "class" }
    { "rule" = "so" }
    { "rule" = "dll" }
    { "rule" = "img" }
    { "rule" = "iso" }
  }
  { "rules" = "application/ogg"
    { "rule" = "ogg" }
  }
  {  }
)

    test lns put "" after
             set "/rules[.=\"application/mac-binhex40\"]"
                 "application/mac-binhex40" ;
             set "/rules[.=\"application/mac-binhex40\"]/rule"
                 "hqx"
        = "application/mac-binhex40 hqx\n"
```

### 12.4.28 0.9.0/tests/test_pg_ident.aug

```
module Test_pg_ident =
    let empty = Pg_ident.empty
    let record = Pg_ident.record
    let lns = Pg_ident.lns

    test empty get "\n" = {}
    test record get "\n" = *
    test lns get "
# This is a comment
a b c
" = (
  { }
  { "#comment" = "This is a comment" }
  { "1"
    { "map" = "a" }
    { "os_user" = "b" }
    { "db_user" = "c" }
  }
)
```

### 12.4.29 0.9.0/tests/test_postgresql.aug

```
module Test_postgresql =
    let empty = Postgresql.empty
    let entry = Postgresql.entry
    let lns = Postgresql.lns

    test empty get "\n" = {}
    test entry get "\n" = *
    test lns get "
# This is a comment
setting = value
" = (
  { }
  { "#comment" = "This is a comment" }
  { "setting" = "value" }
)
```

```
    test lns get "
setting = value # same-line comment
" = (
  { }
  { "setting" = "value"
    { "#comment" = "same-line comment" }
  }
)

    (* i guess IniFile isn’t so smart as to remove and re-add quotes *)
    test lns get "
setting = \"value with spaces\"
" = (
  { }
  { "setting" = "\"value with spaces\"" }
)

    (* nor to ignore comment characters inside quotes *)
    test lns get "
setting = \"value with # bla\" # psyche out
" = (
  { }
  { "setting" = "\"value with"
    { "#comment" = "bla\" # psyche out" }
  }
)

    test lns get "

#--------------------------------------------------------------------------[WRAP]
----
# CLIENT CONNECTION DEFAULTS
#--------------------------------------------------------------------------[WRAP]
----

# These settings are initialized by initdb, but they can be changed.
lc_messages = ’en_US.UTF-8’                     # locale for system error m[WRAP]
essage
                                        # strings
lc_monetary = ’en_US.UTF-8’                     # locale for monetary forma[WRAP]
tting
lc_numeric = ’en_US.UTF-8’                      # locale for number formatt[WRAP]
ing
lc_time = ’en_US.UTF-8’                         # locale for time formattin[WRAP]
g

# default configuration for text search
default_text_search_config = ’pg_catalog.english’

# - Other Defaults -

#dynamic_library_path = ’$libdir’
#local_preload_libraries = ’’
" = (
  {  }
  {  }
  { "#comment" = "---------------------------------------------------------[WRAP]
---------------------" }
  { "#comment" = "CLIENT CONNECTION DEFAULTS" }
  { "#comment" = "----------------------------------------------------------[WRAP]
---------------------" }
  {  }
  { "#comment" = "These settings are initialized by initdb, but they can be[WRAP]
 changed." }
  { "lc_messages" = "'en_US.UTF-8'"
    { "#comment" = "locale for system error message" }
  }
  { "#comment" = "strings" }
  { "lc_monetary" = "'en_US.UTF-8'"
    { "#comment" = "locale for monetary formatting" }
  }
  { "lc_numeric" = "'en_US.UTF-8'"
    { "#comment" = "locale for number formatting" }
  }
  { "lc_time" = "'en_US.UTF-8'"
    { "#comment" = "locale for time formatting" }
  }
  {  }
  { "#comment" = "default configuration for text search" }
  { "default_text_search_config" = "'pg_catalog.english'" }
  {  }
  { "#comment" = "- Other Defaults -" }
  {  }
  { "#comment" = "dynamic_library_path = '$libdir'" }
  { "#comment" = "local_preload_libraries = ''" }
)
```

### 12.4.30 0.9.0/tests/test_someautomountmaps.aug

```
module Test_someautomountmaps =
    let script_content = Someautomountmaps.script_content
    let comment = Someautomountmaps.comment
    let automount_key = Someautomountmaps.automount_key
    let entry = Someautomountmaps.entry
    let lns = Someautomountmaps.lns

    test script_content get
        "#!/bin/bash\nfoo\n    bar\n\tbaz\nbletch\n#comment\n"
        = { "script_content" =
        "#!/bin/bash\nfoo\n    bar\n\tbaz\nbletch\n#comment\n" }
    test comment get "# bla\n" = { "#comment" = "bla" }
    test entry get "\n" = *
    test entry get "foo -fstype=nfs,ro filer:/vol/foo\n" =
        { "entry" = "foo"
            { "options" = "fstype=nfs,ro" }
            { "location" = "filer:/vol/foo" }
        }
    test entry get "foo filer:/vol/foo\n" =
        { "entry" = "foo"
            { "options" }
            { "location" = "filer:/vol/foo" }
        }
    test lns get "foo filer:/vol/foo\n" =
        { "entry" = "foo"
            { "options" }
            { "location" = "filer:/vol/foo" }
        }
    test lns get "\n" = { }
    test lns get "# first line comment but not a hashbang!
foo -fstype=nfs,ro filer:/vol/foo
bar filer2:/vol/bar
# another comment
baz asdfsf
" = (
  { "#comment" = "first line comment but not a hashbang!" }
  { "entry" = "foo"
    { "options" = "fstype=nfs,ro" }
    { "location" = "filer:/vol/foo" }
  }
  { "entry" = "bar"
    { "options" }
    { "location" = "filer2:/vol/bar" }
  }
  { "#comment" = "another comment" }
  { "entry" = "baz"
    { "options" }
    { "location" = "asdfsf" }
  }
)

    test lns put "foo filer:/vol/foo\n" after set "/entry[.='foo']/options"[WRAP]
 "proto=tcp" = "foo -proto=tcp filer:/vol/foo\n"
```

## 12.4.31 0.9.0/tests/test_ssh_config.aug

```
module Test_ssh_config =
    let host = Ssh_config.host
    let anything_but_host = Ssh_config.anything_but_host
    let toplevel_stanza = Ssh_config.toplevel_stanza
    let host_stanza = Ssh_config.host_stanza
    let lns = Ssh_config.lns

    test [host] get "Host *\n" =
        { "Host" = "*" }
    test [host] get "Host *.co.uk\n" =
        { "Host" = "*.co.uk" }
    test [host] get "Host 192.168.0.?\n" =
        { "Host" = "192.168.0.?" }
    test [host] get "host foo.example.com\n" =
        { "Host" = "foo.example.com" }
    test [host] get "   hOsT flarble\n" =
        { "Host" = "flarble" }


    test [anything_but_host] get "F 1\n" =
        { "F" = "1" }
    test [anything_but_host] get "BindAddress 127.0.0.1\n" =
        { "BindAddress" = "127.0.0.1" }
    test [anything_but_host] get "ForYou two words\n" =
        { "ForYou" = "two words" }
    test toplevel_stanza get "Line 1
                              User flarble
                              # A comment

                              Key Value\n" =
        { "toplevel"
            { "Line" = "1" }
            { "User" = "flarble" }
            { "#comment" = "A comment" }
            {  }
            { "Key" = "Value" }
        }

    test host_stanza get "Host mumble
                              User flarble
                              # A comment

                              Key Value\n" =
        { "Host" = "mumble"
            { "User" = "flarble" }
            { "#comment" = "A comment" }
            {  }
            { "Key" = "Value" }
        }

    (* keys can contain digits! *)
    test host_stanza get "Host *
                      User flarble
                      GSSAPIAuthentication yes
                      ForwardX11Trusted yes\n" =
        { "Host" = "*"
            { "User" = "flarble" }
            { "GSSAPIAuthentication" = "yes" }
            { "ForwardX11Trusted" = "yes" }
        }


    test lns get "
# $OpenBSD: ssh_config,v 1.25 2009/02/17 01:28:32 djm Exp $

# This is the ssh client system-wide configuration file.  See
# ssh_config(5) for more information.  This file provides defaults for
# users, and the values can be changed in per-user configuration files
# or on the command line.

# Configuration data is parsed as follows:
#  1. command line options
#  2. user-specific file
#  3. system-wide file
# Any configuration value is only changed the first time it is set.
# Thus, host-specific definitions should be at the beginning of the
# configuration file, and defaults at the end.

# Site-wide defaults for some commonly used options.  For a comprehensive
# list of available options, their meanings and defaults, please see the
# ssh_config(5) man page.

# Host *
#   ForwardAgent no
#   ForwardX11 no
#   RhostsRSAAuthentication no
#   RSAAuthentication yes
#   PasswordAuthentication yes
#   HostbasedAuthentication no
#   GSSAPIAuthentication no
#   GSSAPIDelegateCredentials no
#   GSSAPIKeyExchange no
#   GSSAPITrustDNS no
#   BatchMode no
#   CheckHostIP yes
#   AddressFamily any
#   ConnectTimeout 0
#   StrictHostKeyChecking ask
#   IdentityFile ~/.ssh/identity
#   IdentityFile ~/.ssh/id_rsa
#   IdentityFile ~/.ssh/id_dsa
#   Port 22
#   Protocol 2,1
#   Cipher 3des
#   Ciphers aes128-ctr,aes192-ctr,aes256-ctr,arcfour256,arcfour128,aes128-c[WRAP]
bc,3des-cbc
#   MACs hmac-md5,hmac-sha1,umac-64@openssh.com,hmac-ripemd160
#   EscapeChar ~
#   Tunnel no
#   TunnelDevice any:any
#   PermitLocalCommand no
#   VisualHostKey no
Host *
GSSAPIAuthentication yes
# If this option is set to yes then remote X11 clients will have full acces[WRAP]
s
# to the original X11 display. As virtually no X11 client supports the untr[WRAP]
usted
# mode correctly we set this to yes.
ForwardX11Trusted yes
# Send locale-related environment variables
SendEnv LANG LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONETARY LC_MESSAGE[WRAP]
S
SendEnv LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUREMENT
SendEnv LC_IDENTIFICATION LC_ALL LANGUAGE
SendEnv XMODIFIERS
" =

    { "toplevel"
        {  }
        { "#comment" = "$OpenBSD: ssh_config,v 1.25 2009/02/17 01:28:32 djm[WRAP]
 Exp $" }
        {  }
        { "#comment" = "This is the ssh client system-wide configuration fi[WRAP]
le.  See" }
        { "#comment" = "ssh_config(5) for more information.  This file prov[WRAP]
ides defaults for" }
```

```
        { "#comment" = "users, and the values can be changed in per-user co[WRAP]
nfiguration files" }
        { "#comment" = "or on the command line." }
        {  }
        { "#comment" = "Configuration data is parsed as follows:" }
        { "#comment" = "1. command line options" }
        { "#comment" = "2. user-specific file" }
        { "#comment" = "3. system-wide file" }
        { "#comment" = "Any configuration value is only changed the first t[WRAP]
ime it is set." }
        { "#comment" = "Thus, host-specific definitions should be at the be[WRAP]
ginning of the" }
        { "#comment" = "configuration file, and defaults at the end." }
        {  }
        { "#comment" = "Site-wide defaults for some commonly used options. [WRAP]
 For a comprehensive" }
        { "#comment" = "list of available options, their meanings and defau[WRAP]
lts, please see the" }
        { "#comment" = "ssh_config(5) man page." }
        {  }
        { "#comment" = "Host *" }
        { "#comment" = "ForwardAgent no" }
        { "#comment" = "ForwardX11 no" }
        { "#comment" = "RhostsRSAAuthentication no" }
        { "#comment" = "RSAAuthentication yes" }
        { "#comment" = "PasswordAuthentication yes" }
        { "#comment" = "HostbasedAuthentication no" }
        { "#comment" = "GSSAPIAuthentication no" }
        { "#comment" = "GSSAPIDelegateCredentials no" }
        { "#comment" = "GSSAPIKeyExchange no" }
        { "#comment" = "GSSAPITrustDNS no" }
        { "#comment" = "BatchMode no" }
        { "#comment" = "CheckHostIP yes" }
        { "#comment" = "AddressFamily any" }
        { "#comment" = "ConnectTimeout 0" }
        { "#comment" = "StrictHostKeyChecking ask" }
        { "#comment" = "IdentityFile ~/.ssh/identity" }
        { "#comment" = "IdentityFile ~/.ssh/id_rsa" }
        { "#comment" = "IdentityFile ~/.ssh/id_dsa" }
        { "#comment" = "Port 22" }
        { "#comment" = "Protocol 2,1" }
        { "#comment" = "Cipher 3des" }
        { "#comment" = "Ciphers aes128-ctr,aes192-ctr,aes256-ctr,arcfour256[WRAP]
,arcfour128,aes128-cbc,3des-cbc" }
        { "#comment" = "MACs hmac-md5,hmac-sha1,umac-64@openssh.com,hmac-ri[WRAP]
pemd160" }
        { "#comment" = "EscapeChar ~" }
        { "#comment" = "Tunnel no" }
        { "#comment" = "TunnelDevice any:any" }
        { "#comment" = "PermitLocalCommand no" }
        { "#comment" = "VisualHostKey no" }
    }
    { "Host" = "*"
        { "GSSAPIAuthentication" = "yes" }
        { "#comment" = "If this option is set to yes then remote X11 client[WRAP]
s will have full access" }
        { "#comment" = "to the original X11 display. As virtually no X11 cl[WRAP]
ient supports the untrusted" }
        { "#comment" = "mode correctly we set this to yes." }
        { "ForwardX11Trusted" = "yes" }
        { "#comment" = "Send locale-related environment variables" }
        { "SendEnv" = "LANG LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONET[WRAP]
ARY LC_MESSAGES" }
        { "SendEnv" = "LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUREM[WRAP]
ENT" }
        { "SendEnv" = "LC_IDENTIFICATION LC_ALL LANGUAGE" }
        { "SendEnv" = "XMODIFIERS" }
    }
```

### 12.4.32 0.9.0/tests/test_subject_mapping.aug

```
module Test_subject_mapping =
    let username = Subject_mapping.username
    let arrow = Subject_mapping.arrow
    let certdn = Subject_mapping.certdn
    let line = Subject_mapping.line

    test [ username ] get "foo" = { "foo" }
    test [ arrow ] get " -> " = {}
    test [ arrow ] get "\t->\t" = {}
    test [ arrow . username ] get "\t->\tfoo" = { "foo" }
    test [ certdn ] get "foo" = { = "foo" }
    test [ certdn ] get "foo bar" = { = "foo bar" }
    test line get "foo -> bar\n" = { "bar" = "foo" }
    test line get "Really Odd, Certificate Name. /#$%^&* -> un61\n" =
        { "un61" = "Really Odd, Certificate Name. /#$%^&*" }
```

### 12.4.33 0.9.0/tests/test_subversion.aug

```
module Test_subversion =
    let lns = Subversion.lns
    test lns get "
[global]
foo = bar
" = (
  { }
  { "global"
    { "foo" = "bar" }
  }
)
```

### 12.4.34 0.9.0/tests/test_tracini.aug

```
module Test_tracini =
    let lns = Tracini.lns
    test lns get "
# -*- coding: utf-8 -*-

[attachment]
max_size = 262144
render_unsafe_content = false
```

```
[browser]
hide_properties = svk:merge

[components]
tracgantt.* = enabled

[gantt-charts]
date_format = %Y/%m/%d
include_summary = true
show_opened = true
summary_length = 32
use_creation_date = true

[header_logo]
alt = Trac
height = 73
link = http://trac.edgewall.com/
src = common/trac_banner.png
width = 236

[intertrac]
z = zarquon
zarquon = zarquon
zarquon.title = Zarquon
zarquon.url = https://one.example.com/projects/zarquon
m = mahershalalhashbaz
mahershalalhashbaz = mahershalalhashbaz
mahershalalhashbaz.title = Mahershalalhashbaz trac
mahershalalhashbaz.url = https://two.example.com/projects/mahershalalhashba[WRAP]
z

[logging]
log_file = trac.log
log_level = DEBUG
log_type = none

[mimeviewer]
enscript_path = enscript
max_preview_size = 262144
php_path = php
tab_width = 8

[notification]
always_notify_owner = true
always_notify_reporter = true
smtp_always_cc =
smtp_defaultdomain = example.com
smtp_enabled = true
smtp_from = zarquon-trac@example.com
smtp_password =
smtp_port = 25
smtp_replyto = onewebmaster@example.com
smtp_server = localhost
smtp_user =

[project]
```

```
descr = Zarquon
footer = Visit the Trac open source project at<br /><a href=\"http://trac.e[WRAP]
dgewall.com/\">http://trac.edgewall.com/</a>
icon = common/trac.ico
name = Zarquon
url = https://one.example.com/projects/zarquon/

[ticket]
default_component = component1
default_milestone =
default_priority = major
default_type = defect
default_version =
restrict_owner = false

[ticket-custom]
dependencies = text
dependencies.label = Dependencies
dependencies.value =
due_assign = text
due_assign.label = Due to assign
due_assign.value = YYYY/MM/DD
due_close = text
due_close.label = Due to close
due_close.value = YYYY/MM/DD
include_gantt = checkbox
include_gantt.label = Include in GanttChart
include_gantt.value =

[ticket-workflow]
accept = new -> assigned
accept.operations = set_owner_to_self
accept.permissions = TICKET_MODIFY
leave = * -> *
leave.default = 1
leave.operations = leave_status
reassign = new,assigned,reopened -> new
reassign.operations = set_owner
reassign.permissions = TICKET_MODIFY
reopen = closed -> reopened
reopen.operations = del_resolution
reopen.permissions = TICKET_CREATE
resolve = new,assigned,reopened -> closed
resolve.operations = set_resolution
resolve.permissions = TICKET_MODIFY

[timeline]
changeset_show_files = 0
default_daysback = 30
ticket_show_details = false

[trac]
check_auth_ip = true
database = sqlite:db/trac.db
default_charset = iso-8859-15
default_handler = WikiModule
ignore_auth_case = false
```

```
mainnav = wiki,timeline,roadmap,browser,tickets,newticket,search
metanav = login,logout,settings,help,about
permission_store = DefaultPermissionStore
repository_dir = /var/www/svn/ftdb
templates_dir = /usr/share/trac/templates

[wiki]
ignore_missing_pages = false
" = (
  {  }
  { "#comment" = "-*- coding: utf-8 -*-" }
  {  }
  { "attachment"
    { "max_size" = "262144" }
    { "render_unsafe_content" = "false" }
    {  }
  }
  { "browser"
    { "hide_properties" = "svk:merge" }
    {  }
  }
  { "components"
    { "tracgantt.*" = "enabled" }
    {  }
  }
  { "gantt-charts"
    { "date_format" = "%Y/%m/%d" }
    { "include_summary" = "true" }
    { "show_opened" = "true" }
    { "summary_length" = "32" }
    { "use_creation_date" = "true" }
    {  }
  }
  { "header_logo"
    { "alt" = "Trac" }
    { "height" = "73" }
    { "link" = "http://trac.edgewall.com/" }
    { "src" = "common/trac_banner.png" }
    { "width" = "236" }
    {  }
  }
  { "intertrac"
    { "z" = "zarquon" }
    { "zarquon" = "zarquon" }
    { "zarquon.title" = "Zarquon" }
    { "zarquon.url" = "https://one.example.com/projects/zarquon" }
    { "m" = "mahershalalhashbaz" }
    { "mahershalalhashbaz" = "mahershalalhashbaz" }
    { "mahershalalhashbaz.title" = "Mahershalalhashbaz trac" }
    { "mahershalalhashbaz.url" = "https://two.example.com/projects/mahersha[WRAP]
lalhashbaz" }
    {  }
  }
  { "logging"
    { "log_file" = "trac.log" }
    { "log_level" = "DEBUG" }
    { "log_type" = "none" }
    {  }
  }
  { "mimeviewer"
    { "enscript_path" = "enscript" }
    { "max_preview_size" = "262144" }
    { "php_path" = "php" }
    { "tab_width" = "8" }
    {  }
  }
  { "notification"
    { "always_notify_owner" = "true" }
    { "always_notify_reporter" = "true" }
    { "smtp_always_cc" }
    { "smtp_defaultdomain" = "example.com" }
    { "smtp_enabled" = "true" }
    { "smtp_from" = "zarquon-trac@example.com" }
    { "smtp_password" }
    { "smtp_port" = "25" }
    { "smtp_replyto" = "onewebmaster@example.com" }
    { "smtp_server" = "localhost" }
    { "smtp_user" }
    {  }
  }
  { "project"
    { "descr" = "Zarquon" }
    { "footer" = "Visit the Trac open source project at<br /><a href=\"http[WRAP]
://trac.edgewall.com/\">http://trac.edgewall.com/</a>" }
    { "icon" = "common/trac.ico" }
    { "name" = "Zarquon" }
    { "url" = "https://one.example.com/projects/zarquon/" }
    {  }
  }
  { "ticket"
    { "default_component" = "component1" }
    { "default_milestone" }
    { "default_priority" = "major" }
    { "default_type" = "defect" }
    { "default_version" }
    { "restrict_owner" = "false" }
    {  }
  }
  { "ticket-custom"
    { "dependencies" = "text" }
    { "dependencies.label" = "Dependencies" }
    { "dependencies.value" }
    { "due_assign" = "text" }
    { "due_assign.label" = "Due to assign" }
    { "due_assign.value" = "YYYY/MM/DD" }
    { "due_close" = "text" }
    { "due_close.label" = "Due to close" }
    { "due_close.value" = "YYYY/MM/DD" }
    { "include_gantt" = "checkbox" }
    { "include_gantt.label" = "Include in GanttChart" }
    { "include_gantt.value" }
    {  }
  }
  { "ticket-workflow"
```

```
    { "accept" = "new -> assigned" }
    { "accept.operations" = "set_owner_to_self" }
    { "accept.permissions" = "TICKET_MODIFY" }
    { "leave" = "* -> *" }
    { "leave.default" = "1" }
    { "leave.operations" = "leave_status" }
    { "reassign" = "new,assigned,reopened -> new" }
    { "reassign.operations" = "set_owner" }
    { "reassign.permissions" = "TICKET_MODIFY" }
    { "reopen" = "closed -> reopened" }
    { "reopen.operations" = "del_resolution" }
    { "reopen.permissions" = "TICKET_CREATE" }
    { "resolve" = "new,assigned,reopened -> closed" }
    { "resolve.operations" = "set_resolution" }
    { "resolve.permissions" = "TICKET_MODIFY" }
    {  }
  }
  { "timeline"
    { "changeset_show_files" = "0" }
    { "default_daysback" = "30" }
    { "ticket_show_details" = "false" }
    {  }
  }
  { "trac"
    { "check_auth_ip" = "true" }
    { "database" = "sqlite:db/trac.db" }
    { "default_charset" = "iso-8859-15" }
    { "default_handler" = "WikiModule" }
    { "ignore_auth_case" = "false" }
    { "mainnav" = "wiki,timeline,roadmap,browser,tickets,newticket,search" [WRAP]
}
    { "metanav" = "login,logout,settings,help,about" }
    { "permission_store" = "DefaultPermissionStore" }
    { "repository_dir" = "/var/www/svn/ftdb" }
    { "templates_dir" = "/usr/share/trac/templates" }
    {  }
  }
  { "wiki"
    { "ignore_missing_pages" = "false" }
  }
)
```

### 12.4.35 0.9.0/tests/test_up2date.aug

```
module Test_up2date =
    let akey = Up2date.akey
    let avalue = Up2date.avalue
    let setting = Up2date.setting
    let lns = Up2date.lns

    test [key akey] get "hP[c]" = { "hP[c]" }

    test [store avalue] get "foo" = { = "foo" }
    test [store avalue] get "" = { = "" }

    test setting get
        "hP[c]=H py i ht:p ft, e.g. sqd.rt.c:3128\n" =
        { "hP[c]" = "H py i ht:p ft, e.g. sqd.rt.c:3128" }
    test setting get "foo=\n" = { "foo" = "" }

    test lns get
"# Automatically generated Red Hat Update Agent config file, do not edit.
# Format: 1.0
tmpDir[comment]=Use this Directory to place the temporary transport files
tmpDir=/tmp

disallowConfChanges[comment]=Config options that can not be overwritten by [WRAP]
a config update action
disallowConfChanges=noReboot;sslCACert;useNoSSLForPackages;noSSLServerURL;s[WRAP]
erverURL;disallowConfChanges;

skipNetwork[comment]=Skips network information in hardware profile sync dur[WRAP]
ing registration.
skipNetwork=0

networkRetries[comment]=Number of attempts to make at network connections b[WRAP]
efore giving up
networkRetries=1

hostedWhitelist[comment]=RHN Hosted URL's
hostedWhitelist=

enableProxy[comment]=Use a HTTP Proxy
enableProxy=0

writeChangesToLog[comment]=Log to /var/log/up2date which packages has been [WRAP]
added and removed
writeChangesToLog=0

serverURL[comment]=Remote server URL
serverURL=https://xmlrpc.rhn.redhat.com/XMLRPC

proxyPassword[comment]=The password to use for an authenticated proxy
proxyPassword=

networkSetup[comment]=None
networkSetup=1

proxyUser[comment]=The username for an authenticated proxy
proxyUser=

versionOverride[comment]=Override the automatically determined system versi[WRAP]
on
versionOverride=

sslCACert[comment]=The CA cert used to verify the ssl server
sslCACert=/usr/share/rhn/RHNS-CA-CERT

retrieveOnly[comment]=Retrieve packages only
retrieveOnly=0

debug[comment]=Whether or not debugging is enabled
debug=0
```

```
httpProxy[comment]=HTTP proxy in host:port format, e.g. squid.redhat.com:31[WRAP]
28
httpProxy=

systemIdPath[comment]=Location of system id
systemIdPath=/etc/sysconfig/rhn/systemid

enableProxyAuth[comment]=To use an authenticated proxy or not
enableProxyAuth=0

noReboot[comment]=Disable the reboot actions
noReboot=0
" = (
        { "#comment" = "Automatically generated Red Hat Update Agent config[WRAP]
 file, do not edit." }
        { "#comment" = "Format: 1.0" }
        { "tmpDir[comment]" = "Use this Directory to place the temporary tr[WRAP]
ansport files" }
        { "tmpDir" = "/tmp" }
        {  }
        { "disallowConfChanges[comment]" = "Config options that can not be [WRAP]
overwritten by a config update action" }
        { "disallowConfChanges" = "noReboot;sslCACert;useNoSSLForPackages;n[WRAP]
oSSLServerURL;serverURL;disallowConfChanges;" }
        {  }
        { "skipNetwork[comment]" = "Skips network information in hardware p[WRAP]
rofile sync during registration." }
        { "skipNetwork" = "0" }
        {  }
        { "networkRetries[comment]" = "Number of attempts to make at networ[WRAP]
k connections before giving up" }
        { "networkRetries" = "1" }
        {  }
        { "hostedWhitelist[comment]" = "RHN Hosted URL's" }
        { "hostedWhitelist" = "" }
        {  }
        { "enableProxy[comment]" = "Use a HTTP Proxy" }
        { "enableProxy" = "0" }
        {  }
        { "writeChangesToLog[comment]" = "Log to /var/log/up2date which pac[WRAP]
kages has been added and removed" }
        { "writeChangesToLog" = "0" }
        {  }
        { "serverURL[comment]" = "Remote server URL" }
        { "serverURL" = "https://xmlrpc.rhn.redhat.com/XMLRPC" }
        {  }
        { "proxyPassword[comment]" = "The password to use for an authentica[WRAP]
ted proxy" }
        { "proxyPassword" = "" }
        {  }
        { "networkSetup[comment]" = "None" }
        { "networkSetup" = "1" }
        {  }
        { "proxyUser[comment]" = "The username for an authenticated proxy" [WRAP]
}
        { "proxyUser" = "" }
        {  }
        { "versionOverride[comment]" = "Override the automatically determin[WRAP]
ed system version" }
        { "versionOverride" = "" }
        {  }
        { "sslCACert[comment]" = "The CA cert used to verify the ssl server[WRAP]
" }
        { "sslCACert" = "/usr/share/rhn/RHNS-CA-CERT" }
        {  }
        { "retrieveOnly[comment]" = "Retrieve packages only" }
        { "retrieveOnly" = "0" }
        {  }
        { "debug[comment]" = "Whether or not debugging is enabled" }
        { "debug" = "0" }
        {  }
        { "httpProxy[comment]" = "HTTP proxy in host:port format, e.g. squi[WRAP]
d.redhat.com:3128" }
        { "httpProxy" = "" }
        {  }
        { "systemIdPath[comment]" = "Location of system id" }
        { "systemIdPath" = "/etc/sysconfig/rhn/systemid" }
        {  }
        { "enableProxyAuth[comment]" = "To use an authenticated proxy or no[WRAP]
t" }
        { "enableProxyAuth" = "0" }
        {  }
        { "noReboot[comment]" = "Disable the reboot actions" }
        { "noReboot" = "0" }
    )
```

### 12.4.36 0.9.0/tests/test_upstartinit.aug

```
module Test_upstartinit =
    let lns = Upstartinit.lns
    let script_line = Upstartinit.script_line
    let script = Upstartinit.script
    let lifecycle = Upstartinit.lifecycle
    let respawn = Upstartinit.respawn

    test lns get "\n" = {}
    test lns get "# bla\n" = { "#comment" = "bla" }
    test script_line get "end script\n" = *
    test script_line get "foo\n" = { "1" = "foo" }
    test script get "script\nend script\n" =  { "script" }
    test script get "script\nfoo\nend script\n" =  { "script" { "1" = "foo"[WRAP]
 } }
    test script get "script\n\nend script\n" = { "script" { "1" } }
    test script get "script\n\tfoo\nend script\n" = { "script" { "1" = "\tf[WRAP]
oo" } }
    test lns get "script\nfoo\nbar\nend script\n" =
        { "script"
            { "1" = "foo" }
            { "2" = "bar" }
        }
    test lifecycle get "post-stop exec hi\n" =
        { "post-stop"
            { "exec" = "hi" }
        }
    test lns get "post-stop exec hi\n" =
        { "post-stop"
            { "exec" = "hi" }
        }
    test lns get "exec foo bar baz\n" = { "exec" = "foo bar baz" }

    test respawn get "respawn\n" = { "respawn" }
    test respawn get "respawn foo bar baz\n" = { "respawn" = "foo bar baz" [WRAP]
}

    test lns get "# tty - getty
#
# This service maintains a getty on the specified device.

stop on runlevel [S016]

respawn
instance $TTY
exec /sbin/mingetty $TTY
usage ’tty TTY=/dev/ttyX  - where X is console id’
" = (
  { "#comment" = "tty - getty" }
  {  }
  { "#comment" = "This service maintains a getty on the specified device." [WRAP]
}
  {  }
  { "stop" = "on runlevel [S016]" }
  {  }
  { "respawn" }
  { "instance" = "$TTY" }
  { "exec" = "/sbin/mingetty $TTY" }
  { "usage" = "’tty TTY=/dev/ttyX  - where X is console id’" }
)

(*
    test lns get "
# On machines where kexec isn’t going to be used, free the memory reserved [WRAP]
for it.

start on stopped rcS
task

script
if [ ! -x /sbin/kexec ] || ! chkconfig kdump 2>/dev/null ; then
echo -n \"0\" > /sys/kernel/kexec_crash_size 2>/dev/null
fi
exit 0
end script
" =
(
  {  }
  { "#comment" = "On machines where kexec isn’t going to be used, free the [WRAP]
memory reserved for it." }
  {  }
  { "start" = "on stopped rcS" }
  { "task" }
  {  }
  { "script"
    { "1" = "   if [ ! -x /sbin/kexec ] || ! chkconfig kdump 2>/dev/null ; [WRAP]
then" }
    { "2" = "           echo -n \"0\" > /sys/kernel/kexec_crash_size 2>/dev[WRAP]
/null" }
    { "3" = "   fi" }
    { "4" = "   exit 0" }
  }
)

*)
```

### 12.4.37 1.0.0/lenses/abrt.aug

```
(* abrt.conf is mostly like Puppet configuration, i.e., an ini file
   with # for comments; but it can have numeric keys *)
module Abrt =
 autoload xfm
 (* allow numeric keys; IniFile.entry_re does not have 0-9 in the first [] [WRAP]
*)
 let entry_re = /[A-Za-z0-9][A-Za-z0-9\._-]+/
 let entry = IniFile.indented_entry entry_re Puppet.sep Puppet.comment
 let record = IniFile.record Puppet.title entry
 let lns = IniFile.lns record Puppet.comment
 let xfm = transform lns (incl "/etc/abrt/abrt.conf")
```

### 12.4.38 1.0.0/lenses/automaster.aug

```
module Automaster =
    autoload xfm

    let eol = Util.eol
    let comment = Util.comment
    let empty = Util.empty

    let mount_point = store /\/[^# \t\n]+/
    let include = [ label "include" .
                    del /\+[ \t]*/ "+" .
                    store /[^# \t\n]+/ .
                    eol ]
    let options = [ label "options" . store /-[^ \t\n]+/ ]
    let map_param =
        let    name = [ label "name" . store /[^: \t\n]+/ ]
        in let type = [ label "type" . store /[a-z]+/ ]
        in let format = [ label "format" . store /[a-z]+/ ]
        in let options = [ label "options" . store /[^ \t\n]+/ ]
        in let prelude = ( type .
                           ( del "," "," . format ) ? .
                           del ":" ":" )
        in ( prelude ? .
             name .
             ( Util.del_ws_spc . options ) ? )
    let map_record = [ label "map" .
                       mount_point . Util.del_ws_spc .
                       map_param .
                       eol ]

    let lns = ( map_record |
                include |
                comment |
                empty ) *

    let relevant = (incl "/etc/auto.master") .
                   Util.stdexcl
    let xfm = transform lns relevant
```

### 12.4.39 1.0.0/lenses/automounter.aug

```
(*
Module: Automounter
  Parses automounter file based maps

Author: Dominic Cleal <dcleal@redhat.com>

About: Reference
  See autofs(5)

About: License
   This file is licenced under the LGPL v2+, like the rest of Augeas.

About: Lens Usage
   To be documented

About: Configuration files
   This lens applies to /etc/auto.*, auto_*, excluding known scripts.

About: Examples
   The <Test_Automounter> file contains various examples and tests.
*)

module Automounter =
autoload xfm

(************************************************************************
 * Group:                 USEFUL PRIMITIVES
 *************************************************************************)

(* View: eol *)
let eol = Util.eol

(* View: empty *)
let empty   = Util.empty

(* View: comment *)
let comment = Util.comment

(* View: path *)
let path = /[^-+#: \t\n][^#: \t\n]*/

(* View: hostname *)
let hostname = /[^-:#\(\), \n\t][^:#\(\), \n\t]*/
```

```
(* An option label can't contain comma, comment, equals, or space *)
let optlabel = /[^,#:\(\)= \n\t]+/
let spec    = /[^,#:\(\)= \n\t][^ \n\t]*/

(* View: weight *)
let weight = Rx.integer

(* View: map_name *)
let map_name = /[^: \t\n]+/

(* View: entry_multimount_sep
   Separator for multimount entries, permits line spanning with "\" *)
let entry_multimount_sep = del /[ \t]+(\\\\[ \t]*\n[ \t]+)?/ " "

(************************************************************************
 * Group:                 ENTRIES
 *************************************************************************)

(* View: entry_key
   Key for a map entry *)
let entry_mkey = store path

(* View: entry_path
   Path component of an entry location *)
let entry_path = [ label "path" . store path ]

(* View: entry_host
   Host component with optional weight of an entry location *)
let entry_host = [ label "host" . store hostname
                 . ( Util.del_str "(" . [ label "weight"
                     . store weight ] . Util.del_str ")" )? ]

(* View: comma_sep_list
   Parses options for filesystems *)
let comma_sep_list (l:string) =
  let value = [ label "value" . Util.del_str "=" . store Rx.neg1 ] in
    let lns = [ label l . store optlabel . value? ] in
       Build.opt_list lns Sep.comma

(* View: entry_options *)
let entry_options = Util.del_str "-" . comma_sep_list "opt" . Util.del_ws_t[WRAP]
ab

(* View: entry_location
   A single location with one or more hosts, and one path *)
let entry_location = ( entry_host . ( Sep.comma . entry_host )* )?
                     . Sep.colon . entry_path

(* View: entry_locations
   Multiple locations (each with one or more hosts), separated by spaces *)
let entry_locations = [ label "location" . counter "location"
                      . [ seq "location" . entry_location ]
                      . ( [ Util.del_ws_spc . seq "location" . entry_loca[WRAP]
tion ] )* ]

(* View: entry_multimount
   Parses one of many mountpoints given for a multimount line *)
let entry_multimount = entry_mkey . Util.del_ws_tab . entry_options? . entr[WRAP]
y_locations

(* View: entry_multimounts
   Parses multiple mountpoints given on an entry line *)
let entry_multimounts = [ label "mount" . counter "mount"
                          . [ seq "mount" . entry_multimount ]
                          . ( [ entry_multimount_sep . seq "mount" . entry_[WRAP]
multimount ] )* ]

(* View: entry
   A single map entry from start to finish, including multi-mounts *)
let entry = [ seq "entry" . entry_mkey . Util.del_ws_tab . entry_options?
              . ( entry_locations | entry_multimounts ) . Util.eol ]

(* View: include
   An include line starting with a "+" and a map name *)
let include = [ seq "entry" . store "+" . Util.del_opt_ws ""
                . [ label "map" . store map_name ] . Util.eol ]

(* View: lns *)
let lns = ( empty | comment | entry | include ) *

(* Variable: filter
   Exclude scripts/executable maps from here *)
let filter = incl "/etc/auto.*"
           . incl "/etc/auto_*"
           . excl "/etc/auto.master"
           . excl "/etc/auto_master"
           . excl "/etc/auto.net"
           . excl "/etc/auto.smb"
           . Util.stdexcl

let xfm = transform lns filter
```

### 12.4.40 1.0.0/lenses/gshadow.aug

```
(* based on the group module for Augeas by Free Ekanayaka <free@64studio.co[WRAP]
m>

 Reference: man 5 gshadow

*)

module Gshadow =

   autoload xfm

(************************************************************************
 *                           USEFUL PRIMITIVES
 *************************************************************************)

let eol        = Util.eol
let comment    = Util.comment
let empty      = Util.empty
let colon      = Sep.colon
let comma      = Sep.comma

let sto_to_spc = store Rx.space_in

let word    = Rx.word
let password = /[A-Za-z0-9_.!*\/$-]*/
let integer = Rx.integer

(************************************************************************
 *                               ENTRIES
 *************************************************************************)

let user      = [ label "user" . store word ]
let user_list = Build.opt_list user comma
let params    = [ label "password" . store password  . colon ]
                . [ label "admins" . user_list? . colon ]
                . [ label "members" . user_list? ]
let entry     = Build.key_value_line word colon params

(************************************************************************
 *                                LENS
 *************************************************************************)

let lns        = (comment|empty|entry) *

let filter
               = incl "/etc/gshadow"
               . Util.stdexcl

let xfm        = transform lns filter
```

### 12.4.41 1.0.0/lenses/kdc.aug

```
module Kdc =

autoload xfm

let comment = Krb5.comment
let empty = Krb5.empty

let simple_section = Krb5.simple_section
let kdcdefaults =
  simple_section "kdcdefaults" /kdc_ports|kdc_tcp_ports/

let realm_re = Krb5.realm_re
let entry = Krb5.entry
let eq = Krb5.eq
(* the Krb5.eq_openbr didn't have a newline at the end *)
let eq_openbr = del /[ \t]*=[ \t\n]*\{([ \t]*\n)*/ " = {\n\n"
let closebr = Krb5.closebr
let indent = Krb5.indent
let eol = Krb5.eol
let record = Krb5.record
let realms_enctypes = [ indent . key "supported_enctypes" . eq .
        [ label "type" . store /[^ \t\n#]+/ . Util.del_ws_spc ] * .
        [ label "type" . store /[^ \t\n#]+/ . eol ] ]
```

```
let realms =
  let simple_option = /master_key_type|acl_file|dict_file|admin_keytab/ in
  let list_option = /supported_enctypes/ in
  let soption = entry simple_option eq comment in
  let realm = [ indent . label "realm" . store realm_re .
                eq_openbr . eol . (soption|realms_enctypes)* . closebr . [WRAP]
eol ] in
    record "realms" (realm|comment)


let lns = (comment|empty)* .
  (kdcdefaults|realms)*

let xfm = transform lns (incl "/var/kerberos/krb5kdc/kdc.conf")
```

### 12.4.42 1.0.0/lenses/krb5.aug

```
module Krb5 =

autoload xfm

let comment = Inifile.comment "#" "#"
let empty = Inifile.empty
let eol = Inifile.eol
let dels = Util.del_str

let indent = del /[ \t]*/ ""
let eq = del /[ \t]*=[ \t]*/ " = "
let eq_openbr = del /[ \t]*=[ \t\n]*\{([ \t]*\n)*/ " = {"
let closebr = del /[ \t]*\}/ "}"

(* These two regexps for realms and apps are not entirely true
   - strictly speaking, there's no requirement that a realm is all upper ca[WRAP]
se
   and an application only uses lowercase. But it's what's used in practice[WRAP]
.

   Without that distinction we couldn't distinguish between applications
   and realms in the [appdefaults] section.
*)

let realm_re = /[A-Z][.a-zA-Z0-9-]*/
let app_re = /[a-z][a-zA-Z0-9_]*/
let name_re = /[.a-zA-Z0-9_-]+/

let value = store /[^;# \t\n{}]+/
let entry (kw:regexp) (sep:lens) (comment:lens)
    = [ indent . key kw . sep . value . (comment|eol) ] | comment

let simple_section (n:string) (k:regexp) =
  let title = Inifile.indented_title n in
  let entry = entry k eq comment in
    Inifile.record title entry

let record (t:string) (e:lens) =
  let title = Inifile.indented_title t in
```

```
    Inifile.record title e

let libdefaults =
  let option = entry (name_re - "v4_name_convert") eq comment in
  let subsec = [ indent . key /host|plain/ . eq_openbr .
                   (entry name_re eq comment)* . closebr . eol ] in
  let v4_name_convert = [ indent . key "v4_name_convert" . eq_openbr .
                          subsec* . closebr . eol ] in
  record "libdefaults" (option|v4_name_convert)

let login =
  let keys = /krb[45]_get_tickets|krb4_convert|krb_run_aklog/
    |/aklog_path|accept_passwd/ in
    simple_section "login" keys

let appdefaults =
  let option = entry (name_re - "realm" - "application") eq comment in
  let realm = [ indent . label "realm" . store realm_re .
                  eq_openbr . option* . closebr . eol ] in
  let app = [ indent . label "application" . store app_re .
                eq_openbr . (realm|option)* . closebr . eol] in
    record "appdefaults" (option|realm|app)

let realms =
  let simple_option = /kdc|admin_server|database_module|default_domain/
      |/v4_realm|auth_to_local(_names)?|master_kdc|kpasswd_server/
      |/admin_server/ in
  let subsec_option = /v4_instance_convert/ in
  let option = entry simple_option eq comment in
  let subsec = [ indent . key subsec_option . eq_openbr .
                   (entry name_re eq comment)* . closebr . eol ] in
(* ************************ Changes applied by AFSEO are below ***********[WRAP]
 *)
  let realm = [ indent . label "realm" . store realm_re .
(*                          vvvvv                                         [WRAP]
 *)
                  eq_openbr . eol . (option|subsec)* . closebr . eol ] in
(*                          ^^^^^                                         [WRAP]
 *)
(* ************************ Changes applied by AFSEO are above ***********[WRAP]
 *)
    record "realms" (realm|comment)

let domain_realm =
  simple_section "domain_realm" name_re

let logging =
  let keys = /kdc|admin_server|default/ in
  let xchg (m:regexp) (d:string) (l:string) =
    del m d . label l in
  let xchgs (m:string) (l:string) = xchg m m l in
  let dest =
    [ xchg /FILE[=:]/ "FILE=" "file" . value ]
    |[ xchgs "STDERR" "stderr" ]
    |[ xchgs "CONSOLE" "console" ]
    |[ xchgs "DEVICE=" "device" . value ]
    |[ xchgs "SYSLOG" "syslog" .
```

```
         ([ xchgs ":" "severity" . store /[A-Za-z0-9]+/ ].
          [ xchgs ":" "facility" . store /[A-Za-z0-9]+/ ]?)? ] in
  let entry = [ indent . key keys . eq . dest . (comment|eol) ] | comment i[WRAP]
n
    record "logging" entry

let capaths =
  let realm = [ indent . key realm_re .
                  eq_openbr .
                  (entry realm_re eq comment)* . closebr . eol ] in
    record "capaths" (realm|comment)

let dbdefaults =
  let keys = /database_module|ldap_kerberos_container_dn|ldap_kdc_dn/
    |/ldap_kadmind_dn|ldap_service_password_file|ldap_servers/
    |/ldap_conns_per_server/ in
    simple_section "dbdefaults" keys

let dbmodules =
  let keys = /db_library|ldap_kerberos_container_dn|ldap_kdc_dn/
    |/ldap_kadmind_dn|ldap_service_password_file|ldap_servers/
    |/ldap_conns_per_server/ in
    simple_section "dbmodules" keys

(* This section is not documented in the krb5.conf manpage,
   but the Fermi example uses it. *)
let instance_mapping =
  let value = dels "\"" . store /[^;# \t\n{}]*/ . dels "\"" in
  let map_node = label "mapping" . store /[a-zA-Z0-9\/*]+/ in
  let mapping = [ indent . map_node . eq .
                    [ label "value" . value ] . (comment|eol) ] in
  let instance = [ indent . key name_re .
                     eq_openbr . (mapping|comment)* . closebr . eol ] in
    record "instancemapping" instance

let kdc =
  simple_section "kdc" /profile/

let lns = (comment|empty)* .
  (libdefaults|login|appdefaults|realms|domain_realm
  |logging|capaths|dbdefaults|dbmodules|instance_mapping|kdc)*

let xfm = transform lns (incl "/etc/krb5.conf")
```

### 12.4.43 1.0.0/lenses/libreport_plugins.aug

```
module Libreport_plugins =

autoload xfm

let entry = Build.key_value_line /[A-Za-z]+/ Sep.equal (store /[^\n]*[^ \t\[WRAP]
n]+/)

let lns = ( Util.comment | Util.empty | entry ) *

let filter = (incl "/etc/libreport/plugins/*.conf") . Util.stdexcl
let xfm = transform lns filter
```

### 12.4.44 1.0.0/lenses/mac_ssh.aug

```
(* Tell Augeas to use the ssh lens on Macs, where SSH configuration is dire[WRAP]
ctly
   in /etc, not in /etc/ssh. *)
module Mac_ssh =
    let lns = Ssh.lns
    let xfm = transform lns (incl "/etc/ssh_config")
```

### 12.4.45 1.0.0/lenses/mac_sshd.aug

```
(* Tell Augeas to use the sshd lens on Macs, where SSH configuration is
   directly in /etc, not in /etc/ssh. *)
module Mac_sshd =
    let lns = Sshd.lns
    let xfm = transform lns (incl "/etc/sshd_config")
```

### 12.4.46 1.0.0/lenses/mimetypes.aug

```
module Mimetypes =
    autoload xfm

    (* RFC 2045, Page 11. Closing square bracket moved out of sequence to
       satisfy regex syntax. token_first excludes pound signs so as not to
       overlap with the syntax for comments. *)
    let token =
           let first = /[^]#()<>@,;:\\"\/[?= \t\n]/
        in let rest  =  /[^]()<>@,;:\\"\/[?= \t\n]*/
        in first . rest
    (* We can’t use the mime type as a key, because it has a slash in it *)
    let mime_type = store (token . "/" . token)
    (* This will split up rules wrong if you use spaces within a rule, e.g.
    "ascii(34, 3)" or "string(34,’foo bar’)". But all the rules I’ve ever s[WRAP]
een
    were just filename extensions, so this won’t fail until people forget w[WRAP]
hat
    it is and have to dig to find it. *)
    let a_rule = [ Util.del_ws_spc . label "rule" . store /[^ \t\n]+/ ]
    let rules = [ label "rules" . mime_type . (a_rule *) . Util.eol ]
    let line = ( rules | Util.comment | Util.empty )
    let lns = ( line * )

    let xfm = transform lns (incl "/etc/mime.types")
```

### 12.4.47 1.0.0/lenses/pg_ident.aug

```
module Pg_Ident =
    autoload xfm
    let identifier = store /[a-z_][^ \t\n#]*/
    let record = [ seq "entries" .
                   [ label "map" . identifier ] .
                   Util.del_ws_spc .
                   [ label "os_user" . identifier ] .
                   Util.del_ws_spc .
                   [ label "db_user" . identifier ] .
                   Util.eol
```

```
                ]
    let empty = Util.empty
    let comment = Util.comment
    let line = empty | comment | record
    let lns = line *
    let xfm = transform lns (incl "/var/lib/pgsql/data/pg_ident.conf")
```

### 12.4.48 1.0.0/lenses/postgresql.aug

```
module Postgresql =
    autoload xfm

    let comment = Inifile.comment "#" "#"
    let empty = Inifile.empty
    let eq = del /[ \t]*=/ " ="
    let entry = IniFile.entry IniFile.entry_re eq comment

    let lns = ( entry | empty ) *

    let xfm = transform lns (incl "/var/lib/pgsql/*/postgresql.conf")
```

### 12.4.49 1.0.0/lenses/sos.aug

```
module Sos =
 autoload xfm
 let lns = Puppet.lns
 let xfm = transform lns (incl "/etc/sos.conf")
```

### 12.4.50 1.0.0/lenses/subject_mapping.aug

```
(* Parse pam_pkcs11 subject_mapping file
   File is of the format:

   Certificate Distinguished Name, With Spaces and Commas, Bla Bla. -> user[WRAP]
name

   We're interested in preserving the one-to-one property, that is, that fo[WRAP]
r a
   given username there is only one certificate. Because of this, and becau[WRAP]
se
   the username is shorter and easier to type, we make the username the key
   instead of the certificate distinguished name.
*)

module Subject_mapping =
    autoload xfm
    (* can't have slashes in keys, that's another reason to make the userna[WRAP]
me
       the key *)
    let username = key /[^>\/ \t\n-]+/
    let arrow = del /[ \t]*->[ \t]*/ " -> "
    let certdn = store /[^ \t\n]+([ \t]+[^ \t\n]+)*/
    let line = [ certdn . arrow . username . Util.eol ]

    let lns = line *
    let relevant = (incl "/etc/pam_pkcs11/subject_mapping")
    let xfm = transform lns relevant
```

### 12.4.51 1.0.0/lenses/subversion.aug

```
(* it's just an ini file. sections ("titles") are required *)
module Subversion =
    autoload xfm

    let comment = IniFile.comment "#" "#"
    let sep = IniFile.sep "=" "="
    let entry = IniFile.indented_entry IniFile.entry_re sep comment
    let title = IniFile.indented_title IniFile.record_re
    let record = IniFile.record title entry

    let lns = IniFile.lns record comment

    let relevant = ( incl "/etc/subversion/servers" ) .
                   ( incl "/etc/subversion/config" )

    let xfm = transform lns relevant
```

### 12.4.52 1.0.0/lenses/tracini.aug

```
(* This began as a copy of <Puppet> *)

module Tracini =
  autoload xfm

(************************************************************************
 * INI File settings
 *
 * puppet.conf only supports "# as commentary and "=" as separator
 *************************************************************************)
let comment    = IniFile.comment "#" "#"
let sep        = IniFile.sep "=" "="


(************************************************************************
 *                        ENTRY
 * puppet.conf uses standard INI File entries
 *************************************************************************)
(* began with IniFile.entry_re *)
(* added star as a valid non-first char in entry keys *)
(* allowed single-character entry keys *)
let entry_re           = ( /[A-Za-z][A-Za-z0-9*\._-]*/ )
let entry   = IniFile.indented_entry entry_re sep comment


(************************************************************************
 *                        RECORD
 * puppet.conf uses standard INI File records
 *************************************************************************)
let title   = IniFile.indented_title IniFile.record_re
let record  = IniFile.record title entry
```

```
(************************************************************************
 *                        LENS & FILTER
 * puppet.conf uses standard INI File records
 ************************************************************************)
let lns     = IniFile.lns record comment

let filter = (incl "/var/www/tracs/*/conf/trac.ini")

let xfm = transform lns filter
```

### 12.4.53 1.0.0/lenses/up2date.aug

```
module Up2date =
    autoload xfm

    (* funky syntax: this matches one or more of a-z, A-Z, [ or ]. *)
    let akey = /[]a-zA-Z[]+/
    let avalue = /[^ \t\n]*([ \t]+[^ \t\n]+)*/
    let setting = Build.key_value_line akey (del "=" "=") (store avalue)
    let lns = ( Util.empty | Util.comment | setting ) *

    let xfm = transform lns (incl "/etc/sysconfig/rhn/up2date")
```

### 12.4.54 1.0.0/lenses/upstartinit.aug

```
(* Upstart init configuration files such as found in /etc/init *)

module Upstartinit =
    autoload xfm

    let eol = Util.eol
    let rest_of_line = /[^ \t\n]+([ \t]+[^ \t\n]+)*/
    let whole_line_maybe_indented = /[ \t]*[^ \t\n]+([ \t]+[^ \t\n]+)*/
    let no_params = [ key "task" . eol ]

    let param_is_rest_of_line (thekey:regexp) =
        Build.key_value_line thekey
                             Util.del_ws_spc
                             (store rest_of_line)

    let respawn = [ key "respawn" .
          (Util.del_ws_spc . store rest_of_line)? . eol ]


    let one_params = param_is_rest_of_line
          ( "start"
          | "stop"
          | "env"
          | "export"
          | "normal exit"
          | "instance"
          | "description"
          | "author"
          | "version"
          | "emits"
          | "console"
          | "umask"
          | "nice"
          | "oom"
          | "chroot"
          | "chdir"
          | "limit"
          | "unlimited"
          | "kill timeout"
          | "expect"
          | "usage"
          )

    (* exec and script are valid both at the top level and as a parameter o[WRAP]
f a
    lifecycle keyword *)
    let exec = param_is_rest_of_line "exec"

    let script_line = [ seq "line" .
                        store ( whole_line_maybe_indented - "end script" ) [WRAP]
.
                        eol ] |
                      [ seq "line" . eol]
    let end_script = del "end script\n" "end script\n"
    let script = [ key "script" . eol . script_line * . end_script ]

    let lifecycle = [ key /(pre|post)-(start|stop)/ .  Util.del_ws_spc . ( [WRAP]
exec | script ) ]

    let lns = ( Util.empty
              | Util.comment
              | script
              | exec
              | lifecycle
              | no_params
              | one_params
              | respawn
              ) *

    let relevant = (incl "/etc/init/*.conf") . Util.stdexcl
    let xfm = transform lns relevant
```

### 12.4.55 1.0.0/tests/test_abrt.aug

```
module Test_abrt =
    let lns = Abrt.lns
    test lns get "
[ Common ]
# With this option set to \"yes\",
# only crashes in signed packages will be analyzed.
# the list of public keys used to check the signature is
# in the file gpg_keys
#
OpenGPGCheck = yes

# Blacklisted packages
```

```
#
BlackList = nspluginwrapper, valgrind, strace, mono-core

# Process crashes in executables which do not belong to any package?
#
ProcessUnpackaged = no

# Blacklisted executable paths (shell patterns)
#
BlackListedPaths = /usr/share/doc/*, */example*, /usr/bin/nspluginviewer

# Which database plugin to use
#
Database = SQLite3

# Enable this if you want abrtd to auto-unpack crashdump tarballs which app[WRAP]
ear
# in this directory (for example, uploaded via ftp, scp etc).
# Note: you must ensure that whatever directory you specify here exists
# and is writable for abrtd. abrtd will not create it automatically.
#
#WatchCrashdumpArchiveDir = /var/spool/abrt-upload

# Max size for crash storage [MiB] or 0 for unlimited
#
MaxCrashReportsSize = 1000

# Vector of actions and reporters which are activated immediately
# after a crash occurs, comma separated.
#
#ActionsAndReporters = Mailx(\"[abrt] new crash was detected\")
#ActionsAndReporters = FileTransfer(\"store\")
ActionsAndReporters = SOSreport


# What actions or reporters to run on each crash type
#
[ AnalyzerActionsAndReporters ]
Kerneloops = RHTSupport, Logger
CCpp = RHTSupport, Logger
Python = RHTSupport, Logger
#CCpp:xorg-x11-apps = RunApp(\"date\", \"date.txt\")


# Which Action plugins to run repeatedly
#
[ Cron ]
#   h:m - at h:m
#   s - every s seconds

120 = KerneloopsScanner

#02:00 = FileTransfer
" = (
    {  }
    { " Common "
        { "#comment" = "With this option set to "yes"," }
        { "#comment" = "only crashes in signed packages will be analyzed." [WRAP]
}
        { "#comment" = "the list of public keys used to check the signature[WRAP]
 is" }
        { "#comment" = "in the file gpg_keys" }
        { "#comment" }
        { "OpenGPGCheck" = "yes" }
        {  }
        { "#comment" = "Blacklisted packages" }
        { "#comment" }
        { "BlackList" = "nspluginwrapper, valgrind, strace, mono-core" }
        {  }
        { "#comment" = "Process crashes in executables which do not belong [WRAP]
to any package?" }
        { "#comment" }
        { "ProcessUnpackaged" = "no" }
        {  }
        { "#comment" = "Blacklisted executable paths (shell patterns)" }
        { "#comment" }
        { "BlackListedPaths" = "/usr/share/doc/*, */example*, /usr/bin/nspl[WRAP]
uginviewer" }
        {  }
        { "#comment" = "Which database plugin to use" }
        { "#comment" }
        { "Database" = "SQLite3" }
        {  }
        { "#comment" = "Enable this if you want abrtd to auto-unpack crashd[WRAP]
ump tarballs which appear" }
        { "#comment" = "in this directory (for example, uploaded via ftp, s[WRAP]
cp etc)." }
        { "#comment" = "Note: you must ensure that whatever directory you s[WRAP]
pecify here exists" }
        { "#comment" = "and is writable for abrtd. abrtd will not create it[WRAP]
 automatically." }
        { "#comment" }
        { "#comment" = "WatchCrashdumpArchiveDir = /var/spool/abrt-upload" [WRAP]
}
        {  }
        { "#comment" = "Max size for crash storage [MiB] or 0 for unlimited[WRAP]
" }
        { "#comment" }
        { "MaxCrashReportsSize" = "1000" }
        {  }
        { "#comment" = "Vector of actions and reporters which are activated[WRAP]
 immediately" }
        { "#comment" = "after a crash occurs, comma separated." }
        { "#comment" }
        { "#comment" = "ActionsAndReporters = Mailx("[abrt] new crash was d[WRAP]
etected")" }
        { "#comment" = "ActionsAndReporters = FileTransfer("store")" }
        { "ActionsAndReporters" = "SOSreport" }
        {  }
        {  }
        { "#comment" = "What actions or reporters to run on each crash type[WRAP]
" }
        { "#comment" }
    }
    { " AnalyzerActionsAndReporters "
        { "Kerneloops" = "RHTSupport, Logger" }
        { "CCpp" = "RHTSupport, Logger" }
        { "Python" = "RHTSupport, Logger" }
        { "#comment" = "CCpp:xorg-x11-apps = RunApp("date", "date.txt")" }
        {  }
        {  }
        { "#comment" = "Which Action plugins to run repeatedly" }
        { "#comment" }
    }
    { " Cron "
        { "#comment" = "h:m - at h:m" }
        { "#comment" = "s - every s seconds" }
        {  }
        { "120" = "KerneloopsScanner" }
        {  }
        { "#comment" = "02:00 = FileTransfer" }
    }
)
```

### 12.4.56 1.0.0/tests/test_automaster.aug

```
module Test_automaster =
    let map_param = Automaster.map_param
    let map_record = Automaster.map_record
    let lns = Automaster.lns

    test map_param get "file:/bla/blu" =
        ( { "type" = "file" } { "name" = "/bla/blu" } )
    test map_param get "yp,hesiod:/bla/blu" =
        ( { "type" = "yp" }
          { "format" = "hesiod" }
          { "name" = "/bla/blu" } )
    test map_param get "bla" = { "name" = "bla" }
    test map_record get "/net /etc/auto.net\n" =
        { "map" = "/net"
            { "name" = "/etc/auto.net" } }

    test lns get "# c\n+auto.master\n/net /etc/auto.net\n\n" = (
        { "#comment" = "c" }
        { "include" = "auto.master" }
        { "map" = "/net"
            { "name" = "/etc/auto.net" }
        }
        {  } )

    test lns get "# c
+auto.master
# blank line


/net /etc/auto.net
/foo bla
" = (
  { "#comment" = "c" }
  { "include" = "auto.master" }
  { "#comment" = "blank line" }
  {  }
  {  }
  { "map" = "/net"
    { "name" = "/etc/auto.net" }
  }
  { "map" = "/foo"
    { "name" = "bla" }
  }
)

    test lns get "#
# Sample auto.master file
# This is an automounter map and it has the following format
# key [ -mount-options-separated-by-comma ] location
# For details of the format look at autofs(5).
#
/misc   /etc/auto.misc
#
# NOTE: mounts done from a hosts map will be mounted with the
#       \"nosuid\" and \"nodev\" options unless the \"suid\" and \"dev\"
#       options are explicitly given.
#
/net    -hosts
#
# Include central master map if it can be found using
# nsswitch sources.
#
# Note that if there are entries for /net or /misc (as
# above) in the included master map any keys that are the
# same will not be seen as the first read key seen takes
# precedence.
#
+auto.master
" = (
  {  }
  { "#comment" = "Sample auto.master file" }
  { "#comment" = "This is an automounter map and it has the following forma[WRAP]
t" }
  { "#comment" = "key [ -mount-options-separated-by-comma ] location" }
  { "#comment" = "For details of the format look at autofs(5)." }
  {  }
  { "map" = "/misc"
    { "name" = "/etc/auto.misc" }
  }
  {  }
  { "#comment" = "NOTE: mounts done from a hosts map will be mounted with t[WRAP]
he" }
  { "#comment" = "\"nosuid\" and \"nodev\" options unless the \"suid\" and [WRAP]
\"dev\"" }
  { "#comment" = "options are explicitly given." }
  {  }
  { "map" = "/net"
    { "name" = "-hosts" }
  }
  {  }
  { "#comment" = "Include central master map if it can be found using" }
  { "#comment" = "nsswitch sources." }
  {  }
  { "#comment" = "Note that if there are entries for /net or /misc (as" }
  { "#comment" = "above) in the included master map any keys that are the" [WRAP]
}
  { "#comment" = "same will not be seen as the first read key seen takes" }
  { "#comment" = "precedence." }
  {  }
  { "include" = "auto.master" }
)
```

### 12.4.57 1.0.0/tests/test_gshadow.aug

```
module Test_gshadow =
   let lns = Gshadow.lns
   let entry = Gshadow.entry
   test entry get "root:::\n" =
  { "root"
    { "password" = "" }
    { "admins" }
    { "members" }
  }

   test entry get "bin:::bin,daemon\n" =
  { "bin"
    { "password" = "" }
    { "admins" }
    { "members"
      { "user" = "bin" }
      { "user" = "daemon" }
    }
  }

   test entry get "dbus:!::\n" =
  { "dbus"
    { "password" = "!" }
    { "admins" }
    { "members" }
  }

   test entry get "ntp:!:foo,bar:baz,bletch\n" =
  { "ntp"
    { "password" = "!" }
    { "admins"
      { "user" = "foo" }
      { "user" = "bar" }
    }
    { "members"
      { "user" = "baz" }
      { "user" = "bletch" }
    }
  }

   test entry get "fooz:$5$GQPAI/174dH/Q$dQtmrhcGuolwm7DlKVFkeH.VCWbH1/XTYk[WRAP]
XU83WkIO9::\n" =
  { "fooz"
    { "password" = "$5$GQPAI/174dH/Q$dQtmrhcGuolwm7DlKVFkeH.VCWbH1/XTYkXU83[WRAP]
WkIO9" }
```

```
    { "admins" }
    { "members" }
  }




   test lns get
"root:::
bin:::bin,daemon
dbus:!::
ntp:!:foo,bar:baz,bletch
fooz:$5$GQPAI/174dH/Q$dQtmrhcGuolwm7DlKVFkeH.VCWbH1/XTYkXU83WkIO9::
" =
  { "root"
    { "password" = "" }
    { "admins" }
    { "members" }
  }
  { "bin"
    { "password" = "" }
    { "admins" }
    { "members"
      { "user" = "bin" }
      { "user" = "daemon" }
    }
  }
  { "dbus"
    { "password" = "!" }
    { "admins" }
    { "members" }
  }
  { "ntp"
    { "password" = "!" }
    { "admins"
      { "user" = "foo" }
      { "user" = "bar" }
    }
    { "members"
      { "user" = "baz" }
      { "user" = "bletch" }
    }
  }
  { "fooz"
    { "password" = "$5$GQPAI/174dH/Q$dQtmrhcGuolwm7DlKVFkeH.VCWbH1/XTYkXU83[WRAP]
WkIO9" }
    { "admins" }
    { "members" }
  }
```

### 12.4.58 1.0.0/tests/test_kdc.aug

```
module Test_kdc =
   let lns = Kdc.lns
   let realms_enctypes = Kdc.realms_enctypes
   test realms_enctypes get " supported_enctypes = aes256-cts:normal aes128[WRAP]
-cts:normal des3-hmac-sha1:normal arcfour-hmac:normal des-hmac-sha1:normal [WRAP]
des-cbc-md5:normal des-cbc-crc:normal
" =
  { "supported_enctypes"
    { "type" = "aes256-cts:normal" }
    { "type" = "aes128-cts:normal" }
    { "type" = "des3-hmac-sha1:normal" }
    { "type" = "arcfour-hmac:normal" }
    { "type" = "des-hmac-sha1:normal" }
    { "type" = "des-cbc-md5:normal" }
    { "type" = "des-cbc-crc:normal" }
  }


   test lns get "
[kdcdefaults]
 kdc_ports = 88
 kdc_tcp_ports = 88

[realms]
 EXAMPLE.COM = {
  #master_key_type = aes256-cts
  acl_file = /var/kerberos/krb5kdc/kadm5.acl
  dict_file = /usr/share/dict/words
  admin_keytab = /var/kerberos/krb5kdc/kadm5.keytab
  supported_enctypes = aes256-cts:normal aes128-cts:normal des3-hmac-sha1:n[WRAP]
ormal arcfour-hmac:normal des-hmac-sha1:normal des-cbc-md5:normal des-cbc-c[WRAP]
rc:normal
 }
" = (
  {  }
  { "kdcdefaults"
    { "kdc_ports" = "88" }
    { "kdc_tcp_ports" = "88" }
    {  }
  }
  { "realms"
    { "realm" = "EXAMPLE.COM"
      { "#comment" = "master_key_type = aes256-cts" }
      { "acl_file" = "/var/kerberos/krb5kdc/kadm5.acl" }
      { "dict_file" = "/usr/share/dict/words" }
      { "admin_keytab" = "/var/kerberos/krb5kdc/kadm5.keytab" }
      { "supported_enctypes"
        { "type" = "aes256-cts:normal" }
        { "type" = "aes128-cts:normal" }
        { "type" = "des3-hmac-sha1:normal" }
        { "type" = "arcfour-hmac:normal" }
        { "type" = "des-hmac-sha1:normal" }
        { "type" = "des-cbc-md5:normal" }
        { "type" = "des-cbc-crc:normal" }
      }
    }
  }
)

    test lns put "" after
```

```
        set "realms/realm[999]" "FOO.BAR.EXAMPLE.COM"
    = "[realms]
FOO.BAR.EXAMPLE.COM = {
}
"

    test lns put "[realms]
FOO.BAR.EXAMPLE.COM = {
}" after
        set "realms/realm[.='FOO.BAR.EXAMPLE.COM']/acl_file" "/var/kerberos[WRAP]
/krb5kdc/kadm5.acl"
    = "[realms]
FOO.BAR.EXAMPLE.COM = {
acl_file = /var/kerberos/krb5kdc/kadm5.acl
}
"
```

### 12.4.59 1.0.0/tests/test_libreport_plugins.aug

```
module Test_libreport_plugins =

    let lns = Libreport_plugins.lns
    let entry = Libreport_plugins.entry

    test entry get "Foo=bar\n" = ( { "Foo" = "bar" } )
    test lns get "
# String parameters:

Subject=bla
# EmailFrom=
" = (
  { }
  { "#comment" = "String parameters:" }
  { }
  { "Subject" = "bla" }
  { "#comment" = "EmailFrom=" }
)
```

### 12.4.60 1.0.0/tests/test_mimetypes.aug

```
module Test_mimetypes =
    let mime_type = Mimetypes.mime_type
    let rules = Mimetypes.rules
    let lns = Mimetypes.lns

    test [ mime_type ] get "text/plain" = { = "text/plain" }
    test [ mime_type ] get "application/beep+xml" = { = "application/beep+x[WRAP]
ml" }
    test [ mime_type ] get "application/vnd.fdf" = { = "application/vnd.fdf[WRAP]
" }
    (* who in their right mind made this mime type?! ... oh wait, they were[WRAP]
n't,
       it's microsoft *)
    test [ mime_type ] get
        "application/vnd.openxmlformats-officedocument.wordprocessingml.doc[WRAP]
```

```
ument" =
        { = "application/vnd.openxmlformats-officedocument.wordprocessingml[WRAP]
.document" }
    test rules get "text/plain txt\n" =
        { "rules" = "text/plain"
          { "rule" = "txt" } }
    test rules get "application/vnd.openxmlformats-officedocument.wordproce[WRAP]
ssingml.document docx\n" =
        { "rules" = "application/vnd.openxmlformats-officedocument.wordproc[WRAP]
essingml.document"
          { "rule" = "docx" } }
    test rules get "video/mpeg                      mpeg mpg mpe\n" =
        { "rules" = "video/mpeg"
          { "rule" = "mpeg" }
          { "rule" = "mpg" }
          { "rule" = "mpe" } }
    test lns get "
# This is a comment. I love comments.

# This file controls what Internet media types are sent to the client for
# given file extension(s).  Sending the correct media type to the client
# is important so they know how to handle the content of the file.
# Extra types can either be added here or by using an AddType directive
# in your config files. For more information about Internet media types,
# please read RFC 2045, 2046, 2047, 2048, and 2077.  The Internet media typ[WRAP]
e
# registry is at <http://www.iana.org/assignments/media-types/>.

# MIME type                     Extension
application/EDI-Consent
application/andrew-inset        ez
application/mac-binhex40        hqx
application/mac-compactpro      cpt
application/octet-stream        bin dms lha lzh exe class so dll img iso
application/ogg                 ogg

" = (
  {  }
  { "#comment" = "This is a comment. I love comments." }
  {  }
  { "#comment" = "This file controls what Internet media types are sent to [WRAP]
the client for" }
  { "#comment" = "given file extension(s).  Sending the correct media type [WRAP]
to the client" }
  { "#comment" = "is important so they know how to handle the content of th[WRAP]
e file." }
  { "#comment" = "Extra types can either be added here or by using an AddTy[WRAP]
pe directive" }
  { "#comment" = "in your config files. For more information about Internet[WRAP]
 media types," }
  { "#comment" = "please read RFC 2045, 2046, 2047, 2048, and 2077.  The In[WRAP]
ternet media type" }
  { "#comment" = "registry is at <http://www.iana.org/assignments/media-typ[WRAP]
es/>." }
  {  }
  { "#comment" = "MIME type                     Extension" }
  { "rules" = "application/EDI-Consent" }
  { "rules" = "application/andrew-inset"
    { "rule" = "ez" }
  }
  { "rules" = "application/mac-binhex40"
    { "rule" = "hqx" }
  }
  { "rules" = "application/mac-compactpro"
    { "rule" = "cpt" }
  }
  { "rules" = "application/octet-stream"
    { "rule" = "bin" }
    { "rule" = "dms" }
    { "rule" = "lha" }
    { "rule" = "lzh" }
    { "rule" = "exe" }
    { "rule" = "class" }
    { "rule" = "so" }
    { "rule" = "dll" }
    { "rule" = "img" }
    { "rule" = "iso" }
  }
  { "rules" = "application/ogg"
    { "rule" = "ogg" }
  }
  {  }
)

    test lns put "" after
             set "/rules[.=\"application/mac-binhex40\"]"
                 "application/mac-binhex40" ;
             set "/rules[.=\"application/mac-binhex40\"]/rule"
                 "hqx"
        = "application/mac-binhex40 hqx\n"
```

## 12.4.61 1.0.0/tests/test_pg_ident.aug

```
module Test_pg_ident =
    let empty = Pg_ident.empty
    let record = Pg_ident.record
    let lns = Pg_ident.lns

    test empty get "\n" = {}
    test record get "\n" = *
    test lns get "
# This is a comment
a b c
" = (
  { }
  { "#comment" = "This is a comment" }
  { "1"
    { "map" = "a" }
    { "os_user" = "b" }
    { "db_user" = "c" }
  }
)
```

### 12.4.62 1.0.0/tests/test_postgresql.aug

```
module Test_postgresql =
    let empty = Postgresql.empty
    let entry = Postgresql.entry
    let lns = Postgresql.lns

    test empty get "\n" = {}
    test entry get "\n" = *
    test lns get "
# This is a comment
setting = value
" = (
  { }
  { "#comment" = "This is a comment" }
  { "setting" = "value" }
)

    test lns get "
setting = value # same-line comment
" = (
  { }
  { "setting" = "value"
    { "#comment" = "same-line comment" }
  }
)

    (* i guess IniFile isn’t so smart as to remove and re-add quotes *)
    test lns get "
setting = \"value with spaces\"
" = (
  { }
  { "setting" = "\"value with spaces\"" }
)

    (* nor to ignore comment characters inside quotes *)
    test lns get "
setting = \"value with # bla\" # psyche out
" = (
  { }
  { "setting" = "\"value with"
    { "#comment" = "bla\" # psyche out" }
  }
)

    test lns get "

#--------------------------------------------------------------------------[WRAP]
----
# CLIENT CONNECTION DEFAULTS
#--------------------------------------------------------------------------[WRAP]
----

# These settings are initialized by initdb, but they can be changed.
lc_messages = ’en_US.UTF-8’                     # locale for system error m[WRAP]
essage
                                        # strings
lc_monetary = ’en_US.UTF-8’                     # locale for monetary forma[WRAP]
```

```
tting
lc_numeric = ’en_US.UTF-8’                      # locale for number formatt[WRAP]
ing
lc_time = ’en_US.UTF-8’                         # locale for time formattin[WRAP]
g

# default configuration for text search
default_text_search_config = ’pg_catalog.english’

# - Other Defaults -

#dynamic_library_path = ’$libdir’
#local_preload_libraries = ’’
" = (
  {  }
  {  }
  { "#comment" = "---------------------------------------------------------[WRAP]
---------------------" }
  { "#comment" = "CLIENT CONNECTION DEFAULTS" }
  { "#comment" = "---------------------------------------------------------[WRAP]
---------------------" }
  {  }
  { "#comment" = "These settings are initialized by initdb, but they can be[WRAP]
 changed." }
  { "lc_messages" = "’en_US.UTF-8’"
    { "#comment" = "locale for system error message" }
  }
  { "#comment" = "strings" }
  { "lc_monetary" = "’en_US.UTF-8’"
    { "#comment" = "locale for monetary formatting" }
  }
  { "lc_numeric" = "’en_US.UTF-8’"
    { "#comment" = "locale for number formatting" }
  }
  { "lc_time" = "’en_US.UTF-8’"
    { "#comment" = "locale for time formatting" }
  }
  {  }
  { "#comment" = "default configuration for text search" }
  { "default_text_search_config" = "’pg_catalog.english’" }
  {  }
  { "#comment" = "- Other Defaults -" }
  {  }
  { "#comment" = "dynamic_library_path = ’$libdir’" }
  { "#comment" = "local_preload_libraries = ’’" }
)
```

### 12.4.63 1.0.0/tests/test_ssh_config.aug

```
module Test_ssh_config =
    let host = Ssh_config.host
    let anything_but_host = Ssh_config.anything_but_host
    let toplevel_stanza = Ssh_config.toplevel_stanza
    let host_stanza = Ssh_config.host_stanza
    let lns = Ssh_config.lns
```

```
test [host] get "Host *\n" =
    { "Host" = "*" }
test [host] get "Host *.co.uk\n" =
    { "Host" = "*.co.uk" }
test [host] get "Host 192.168.0.?\n" =
    { "Host" = "192.168.0.?" }
test [host] get "host foo.example.com\n" =
    { "Host" = "foo.example.com" }
test [host] get "   hOsT flarble\n" =
    { "Host" = "flarble" }


test [anything_but_host] get "F 1\n" =
    { "F" = "1" }
test [anything_but_host] get "BindAddress 127.0.0.1\n" =
    { "BindAddress" = "127.0.0.1" }
test [anything_but_host] get "ForYou two words\n" =
    { "ForYou" = "two words" }


test toplevel_stanza get "Line 1
                          User flarble
                          # A comment

                          Key Value\n" =
    { "toplevel"
        { "Line" = "1" }
        { "User" = "flarble" }
        { "#comment" = "A comment" }
        {  }
        { "Key" = "Value" }
    }

test host_stanza get "Host mumble
                          User flarble
                          # A comment

                          Key Value\n" =
    { "Host" = "mumble"
        { "User" = "flarble" }
        { "#comment" = "A comment" }
        {  }
        { "Key" = "Value" }
    }

(* keys can contain digits! *)
test host_stanza get "Host *
                  User flarble
                  GSSAPIAuthentication yes
                  ForwardX11Trusted yes\n" =
    { "Host" = "*"
        { "User" = "flarble" }
        { "GSSAPIAuthentication" = "yes" }
        { "ForwardX11Trusted" = "yes" }
    }
    test lns get "
# $OpenBSD: ssh_config,v 1.25 2009/02/17 01:28:32 djm Exp $

# This is the ssh client system-wide configuration file.  See
# ssh_config(5) for more information.  This file provides defaults for
# users, and the values can be changed in per-user configuration files
# or on the command line.

# Configuration data is parsed as follows:
#  1. command line options
#  2. user-specific file
#  3. system-wide file
# Any configuration value is only changed the first time it is set.
# Thus, host-specific definitions should be at the beginning of the
# configuration file, and defaults at the end.

# Site-wide defaults for some commonly used options.  For a comprehensive
# list of available options, their meanings and defaults, please see the
# ssh_config(5) man page.

# Host *
#   ForwardAgent no
#   ForwardX11 no
#   RhostsRSAAuthentication no
#   RSAAuthentication yes
#   PasswordAuthentication yes
#   HostbasedAuthentication no
#   GSSAPIAuthentication no
#   GSSAPIDelegateCredentials no
#   GSSAPIKeyExchange no
#   GSSAPITrustDNS no
#   BatchMode no
#   CheckHostIP yes
#   AddressFamily any
#   ConnectTimeout 0
#   StrictHostKeyChecking ask
#   IdentityFile ~/.ssh/identity
#   IdentityFile ~/.ssh/id_rsa
#   IdentityFile ~/.ssh/id_dsa
#   Port 22
#   Protocol 2,1
#   Cipher 3des
#   Ciphers aes128-ctr,aes192-ctr,aes256-ctr,arcfour256,arcfour128,aes128-c[WRAP]
bc,3des-cbc
#   MACs hmac-md5,hmac-sha1,umac-64@openssh.com,hmac-ripemd160
#   EscapeChar ~
#   Tunnel no
#   TunnelDevice any:any
#   PermitLocalCommand no
#   VisualHostKey no
Host *
GSSAPIAuthentication yes
# If this option is set to yes then remote X11 clients will have full acces[WRAP]
s
# to the original X11 display. As virtually no X11 client supports the untr[WRAP]
usted
# mode correctly we set this to yes.
```

```
ForwardX11Trusted yes
# Send locale-related environment variables
SendEnv LANG LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONETARY LC_MESSAGE[WRAP]
S
SendEnv LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUREMENT
SendEnv LC_IDENTIFICATION LC_ALL LANGUAGE
SendEnv XMODIFIERS
" =

    { "toplevel"
        {  }
        { "#comment" = "$OpenBSD: ssh_config,v 1.25 2009/02/17 01:28:32 djm[WRAP]
 Exp $" }
        {  }
        { "#comment" = "This is the ssh client system-wide configuration fi[WRAP]
le.  See" }
        { "#comment" = "ssh_config(5) for more information.  This file prov[WRAP]
ides defaults for" }
        { "#comment" = "users, and the values can be changed in per-user co[WRAP]
nfiguration files" }
        { "#comment" = "or on the command line." }
        {  }
        { "#comment" = "Configuration data is parsed as follows:" }
        { "#comment" = "1. command line options" }
        { "#comment" = "2. user-specific file" }
        { "#comment" = "3. system-wide file" }
        { "#comment" = "Any configuration value is only changed the first t[WRAP]
ime it is set." }
        { "#comment" = "Thus, host-specific definitions should be at the be[WRAP]
ginning of the" }
        { "#comment" = "configuration file, and defaults at the end." }
        {  }
        { "#comment" = "Site-wide defaults for some commonly used options. [WRAP]
 For a comprehensive" }
        { "#comment" = "list of available options, their meanings and defau[WRAP]
lts, please see the" }
        { "#comment" = "ssh_config(5) man page." }
        {  }
        { "#comment" = "Host *" }
        { "#comment" = "ForwardAgent no" }
        { "#comment" = "ForwardX11 no" }
        { "#comment" = "RhostsRSAAuthentication no" }
        { "#comment" = "RSAAuthentication yes" }
        { "#comment" = "PasswordAuthentication yes" }
        { "#comment" = "HostbasedAuthentication no" }
        { "#comment" = "GSSAPIAuthentication no" }
        { "#comment" = "GSSAPIDelegateCredentials no" }
        { "#comment" = "GSSAPIKeyExchange no" }
        { "#comment" = "GSSAPITrustDNS no" }
        { "#comment" = "BatchMode no" }
        { "#comment" = "CheckHostIP yes" }
        { "#comment" = "AddressFamily any" }
        { "#comment" = "ConnectTimeout 0" }
        { "#comment" = "StrictHostKeyChecking ask" }
        { "#comment" = "IdentityFile ~/.ssh/identity" }
        { "#comment" = "IdentityFile ~/.ssh/id_rsa" }
        { "#comment" = "IdentityFile ~/.ssh/id_dsa" }
        { "#comment" = "Port 22" }
        { "#comment" = "Protocol 2,1" }
        { "#comment" = "Cipher 3des" }
        { "#comment" = "Ciphers aes128-ctr,aes192-ctr,aes256-ctr,arcfour256[WRAP]
,arcfour128,aes128-cbc,3des-cbc" }
        { "#comment" = "MACs hmac-md5,hmac-sha1,umac-64@openssh.com,hmac-ri[WRAP]
pemd160" }
        { "#comment" = "EscapeChar ~" }
        { "#comment" = "Tunnel no" }
        { "#comment" = "TunnelDevice any:any" }
        { "#comment" = "PermitLocalCommand no" }
        { "#comment" = "VisualHostKey no" }
    }
    { "Host" = "*"
        { "GSSAPIAuthentication" = "yes" }
        { "#comment" = "If this option is set to yes then remote X11 client[WRAP]
s will have full access" }
        { "#comment" = "to the original X11 display. As virtually no X11 cl[WRAP]
ient supports the untrusted" }
        { "#comment" = "mode correctly we set this to yes." }
        { "ForwardX11Trusted" = "yes" }
        { "#comment" = "Send locale-related environment variables" }
        { "SendEnv" = "LANG LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONET[WRAP]
ARY LC_MESSAGES" }
        { "SendEnv" = "LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUREM[WRAP]
ENT" }
        { "SendEnv" = "LC_IDENTIFICATION LC_ALL LANGUAGE" }
        { "SendEnv" = "XMODIFIERS" }
    }
```

### 12.4.64 1.0.0/tests/test_subject_mapping.aug

```
module Test_subject_mapping =
    let username = Subject_mapping.username
    let arrow = Subject_mapping.arrow
    let certdn = Subject_mapping.certdn
    let line = Subject_mapping.line

    test [ username ] get "foo" = { "foo" }
    test [ arrow ] get " -> " = {}
    test [ arrow ] get "\t->\t" = {}
    test [ arrow . username ] get "\t->\tfoo" = { "foo" }
    test [ certdn ] get "foo" = { = "foo" }
    test [ certdn ] get "foo bar" = { = "foo bar" }
    test line get "foo -> bar\n" = { "bar" = "foo" }
    test line get "Really Odd, Certificate Name. /#$%^&* -> un61\n" =
        { "un61" = "Really Odd, Certificate Name. /#$%^&*" }
```

### 12.4.65 1.0.0/tests/test_subversion.aug

```
module Test_subversion =
    let lns = Subversion.lns
    test lns get "
[global]
foo = bar
```

```
" = (
  { }
  { "global"
    { "foo" = "bar" }
  }
)
```

### 12.4.66 1.0.0/tests/test_tracini.aug

```
module Test_tracini =
    let lns = Tracini.lns
    test lns get "
# -*- coding: utf-8 -*-

[attachment]
max_size = 262144
render_unsafe_content = false

[browser]
hide_properties = svk:merge

[components]
tracgantt.* = enabled

[gantt-charts]
date_format = %Y/%m/%d
include_summary = true
show_opened = true
summary_length = 32
use_creation_date = true

[header_logo]
alt = Trac
height = 73
link = http://trac.edgewall.com/
src = common/trac_banner.png
width = 236

[intertrac]
z = zarquon
zarquon = zarquon
zarquon.title = Zarquon
zarquon.url = https://one.example.com/projects/zarquon
m = mahershalalhashbaz
mahershalalhashbaz = mahershalalhashbaz
mahershalalhashbaz.title = Mahershalalhashbaz trac
mahershalalhashbaz.url = https://two.example.com/projects/mahershalalhashba[WRAP]
z

[logging]
log_file = trac.log
log_level = DEBUG
log_type = none

[mimeviewer]
enscript_path = enscript
```

```
max_preview_size = 262144
php_path = php
tab_width = 8

[notification]
always_notify_owner = true
always_notify_reporter = true
smtp_always_cc =
smtp_defaultdomain = example.com
smtp_enabled = true
smtp_from = zarquon-trac@example.com
smtp_password =
smtp_port = 25
smtp_replyto = onewebmaster@example.com
smtp_server = localhost
smtp_user =

[project]
descr = Zarquon
footer = Visit the Trac open source project at<br /><a href=\"http://trac.e[WRAP]
dgewall.com/\">http://trac.edgewall.com/</a>
icon = common/trac.ico
name = Zarquon
url = https://one.example.com/projects/zarquon/

[ticket]
default_component = component1
default_milestone =
default_priority = major
default_type = defect
default_version =
restrict_owner = false

[ticket-custom]
dependencies = text
dependencies.label = Dependencies
dependencies.value =
due_assign = text
due_assign.label = Due to assign
due_assign.value = YYYY/MM/DD
due_close = text
due_close.label = Due to close
due_close.value = YYYY/MM/DD
include_gantt = checkbox
include_gantt.label = Include in GanttChart
include_gantt.value =

[ticket-workflow]
accept = new -> assigned
accept.operations = set_owner_to_self
accept.permissions = TICKET_MODIFY
leave = * -> *
leave.default = 1
leave.operations = leave_status
reassign = new,assigned,reopened -> new
reassign.operations = set_owner
reassign.permissions = TICKET_MODIFY
```

```
reopen = closed -> reopened
reopen.operations = del_resolution
reopen.permissions = TICKET_CREATE
resolve = new,assigned,reopened -> closed
resolve.operations = set_resolution
resolve.permissions = TICKET_MODIFY

[timeline]
changeset_show_files = 0
default_daysback = 30
ticket_show_details = false

[trac]
check_auth_ip = true
database = sqlite:db/trac.db
default_charset = iso-8859-15
default_handler = WikiModule
ignore_auth_case = false
mainnav = wiki,timeline,roadmap,browser,tickets,newticket,search
metanav = login,logout,settings,help,about
permission_store = DefaultPermissionStore
repository_dir = /var/www/svn/ftdb
templates_dir = /usr/share/trac/templates

[wiki]
ignore_missing_pages = false
" = (
  {  }
  { "#comment" = "-*- coding: utf-8 -*-" }
  {  }
  { "attachment"
    { "max_size" = "262144" }
    { "render_unsafe_content" = "false" }
    {  }
  }
  { "browser"
    { "hide_properties" = "svk:merge" }
    {  }
  }
  { "components"
    { "tracgantt.*" = "enabled" }
    {  }
  }
  { "gantt-charts"
    { "date_format" = "%Y/%m/%d" }
    { "include_summary" = "true" }
    { "show_opened" = "true" }
    { "summary_length" = "32" }
    { "use_creation_date" = "true" }
    {  }
  }
  { "header_logo"
    { "alt" = "Trac" }
    { "height" = "73" }
    { "link" = "http://trac.edgewall.com/" }
    { "src" = "common/trac_banner.png" }
    { "width" = "236" }
    {  }
  }
  { "intertrac"
    { "z" = "zarquon" }
    { "zarquon" = "zarquon" }
    { "zarquon.title" = "Zarquon" }
    { "zarquon.url" = "https://one.example.com/projects/zarquon" }
    { "m" = "mahershalalhashbaz" }
    { "mahershalalhashbaz" = "mahershalalhashbaz" }
    { "mahershalalhashbaz.title" = "Mahershalalhashbaz trac" }
    { "mahershalalhashbaz.url" = "https://two.example.com/projects/mahersha[WRAP]
lalhashbaz" }
    {  }
  }
  { "logging"
    { "log_file" = "trac.log" }
    { "log_level" = "DEBUG" }
    { "log_type" = "none" }
    {  }
  }
  { "mimeviewer"
    { "enscript_path" = "enscript" }
    { "max_preview_size" = "262144" }
    { "php_path" = "php" }
    { "tab_width" = "8" }
    {  }
  }
  { "notification"
    { "always_notify_owner" = "true" }
    { "always_notify_reporter" = "true" }
    { "smtp_always_cc" }
    { "smtp_defaultdomain" = "example.com" }
    { "smtp_enabled" = "true" }
    { "smtp_from" = "zarquon-trac@example.com" }
    { "smtp_password" }
    { "smtp_port" = "25" }
    { "smtp_replyto" = "onewebmaster@example.com" }
    { "smtp_server" = "localhost" }
    { "smtp_user" }
    {  }
  }
  { "project"
    { "descr" = "Zarquon" }
    { "footer" = "Visit the Trac open source project at<br /><a href=\"http[WRAP]
://trac.edgewall.com/\">http://trac.edgewall.com/</a>" }
    { "icon" = "common/trac.ico" }
    { "name" = "Zarquon" }
    { "url" = "https://one.example.com/projects/zarquon/" }
    {  }
  }
  { "ticket"
    { "default_component" = "component1" }
    { "default_milestone" }
    { "default_priority" = "major" }
    { "default_type" = "defect" }
    { "default_version" }
    { "restrict_owner" = "false" }
    {  }
  }
  { "ticket-custom"
    { "dependencies" = "text" }
    { "dependencies.label" = "Dependencies" }
    { "dependencies.value" }
    { "due_assign" = "text" }
    { "due_assign.label" = "Due to assign" }
    { "due_assign.value" = "YYYY/MM/DD" }
    { "due_close" = "text" }
    { "due_close.label" = "Due to close" }
    { "due_close.value" = "YYYY/MM/DD" }
    { "include_gantt" = "checkbox" }
    { "include_gantt.label" = "Include in GanttChart" }
    { "include_gantt.value" }
    {  }
  }
  { "ticket-workflow"
    { "accept" = "new -> assigned" }
    { "accept.operations" = "set_owner_to_self" }
    { "accept.permissions" = "TICKET_MODIFY" }
    { "leave" = "* -> *" }
    { "leave.default" = "1" }
    { "leave.operations" = "leave_status" }
    { "reassign" = "new,assigned,reopened -> new" }
    { "reassign.operations" = "set_owner" }
    { "reassign.permissions" = "TICKET_MODIFY" }
    { "reopen" = "closed -> reopened" }
    { "reopen.operations" = "del_resolution" }
    { "reopen.permissions" = "TICKET_CREATE" }
    { "resolve" = "new,assigned,reopened -> closed" }
    { "resolve.operations" = "set_resolution" }
    { "resolve.permissions" = "TICKET_MODIFY" }
    {  }
  }
  { "timeline"
    { "changeset_show_files" = "0" }
    { "default_daysback" = "30" }
    { "ticket_show_details" = "false" }
    {  }
  }
  { "trac"
    { "check_auth_ip" = "true" }
    { "database" = "sqlite:db/trac.db" }
    { "default_charset" = "iso-8859-15" }
    { "default_handler" = "WikiModule" }
    { "ignore_auth_case" = "false" }
    { "mainnav" = "wiki,timeline,roadmap,browser,tickets,newticket,search" [WRAP]
}
    { "metanav" = "login,logout,settings,help,about" }
    { "permission_store" = "DefaultPermissionStore" }
    { "repository_dir" = "/var/www/svn/ftdb" }
    { "templates_dir" = "/usr/share/trac/templates" }
    {  }
  }
  { "wiki"
    { "ignore_missing_pages" = "false" }
  }
)
```

## 12.4.67 1.0.0/tests/test_up2date.aug

```
module Test_up2date =
    let akey = Up2date.akey
    let avalue = Up2date.avalue
    let setting = Up2date.setting
    let lns = Up2date.lns

    test [key akey] get "hP[c]" = { "hP[c]" }

    test [store avalue] get "foo" = { = "foo" }
    test [store avalue] get "" = { = "" }

    test setting get
        "hP[c]=H py i ht:p ft, e.g. sqd.rt.c:3128\n" =
        { "hP[c]" = "H py i ht:p ft, e.g. sqd.rt.c:3128" }
    test setting get "foo=\n" = { "foo" = "" }

    test lns get
"# Automatically generated Red Hat Update Agent config file, do not edit.
# Format: 1.0
tmpDir[comment]=Use this Directory to place the temporary transport files
tmpDir=/tmp

disallowConfChanges[comment]=Config options that can not be overwritten by [WRAP]
a config update action
disallowConfChanges=noReboot;sslCACert;useNoSSLForPackages;noSSLServerURL;s[WRAP]
erverURL;disallowConfChanges;

skipNetwork[comment]=Skips network information in hardware profile sync dur[WRAP]
ing registration.
skipNetwork=0

networkRetries[comment]=Number of attempts to make at network connections b[WRAP]
efore giving up
networkRetries=1

hostedWhitelist[comment]=RHN Hosted URL's
hostedWhitelist=

enableProxy[comment]=Use a HTTP Proxy
enableProxy=0

writeChangesToLog[comment]=Log to /var/log/up2date which packages has been [WRAP]
added and removed
writeChangesToLog=0

serverURL[comment]=Remote server URL
serverURL=https://xmlrpc.rhn.redhat.com/XMLRPC

proxyPassword[comment]=The password to use for an authenticated proxy
proxyPassword=

networkSetup[comment]=None
```

```
networkSetup=1

proxyUser[comment]=The username for an authenticated proxy
proxyUser=

versionOverride[comment]=Override the automatically determined system versi[WRAP]
on
versionOverride=

sslCACert[comment]=The CA cert used to verify the ssl server
sslCACert=/usr/share/rhn/RHNS-CA-CERT

retrieveOnly[comment]=Retrieve packages only
retrieveOnly=0

debug[comment]=Whether or not debugging is enabled
debug=0

httpProxy[comment]=HTTP proxy in host:port format, e.g. squid.redhat.com:31[WRAP]
28
httpProxy=

systemIdPath[comment]=Location of system id
systemIdPath=/etc/sysconfig/rhn/systemid

enableProxyAuth[comment]=To use an authenticated proxy or not
enableProxyAuth=0

noReboot[comment]=Disable the reboot actions
noReboot=0
" = (
        { "#comment" = "Automatically generated Red Hat Update Agent config[WRAP]
 file, do not edit." }
        { "#comment" = "Format: 1.0" }
        { "tmpDir[comment]" = "Use this Directory to place the temporary tr[WRAP]
ansport files" }
        { "tmpDir" = "/tmp" }
        {  }
        { "disallowConfChanges[comment]" = "Config options that can not be [WRAP]
overwritten by a config update action" }
        { "disallowConfChanges" = "noReboot;sslCACert;useNoSSLForPackages;n[WRAP]
oSSLServerURL;serverURL;disallowConfChanges;" }
        {  }
        { "skipNetwork[comment]" = "Skips network information in hardware p[WRAP]
rofile sync during registration." }
        { "skipNetwork" = "0" }
        {  }
        { "networkRetries[comment]" = "Number of attempts to make at networ[WRAP]
k connections before giving up" }
        { "networkRetries" = "1" }
        {  }
        { "hostedWhitelist[comment]" = "RHN Hosted URL's" }
        { "hostedWhitelist" = "" }
        {  }
        { "enableProxy[comment]" = "Use a HTTP Proxy" }
        { "enableProxy" = "0" }
        {  }
        { "writeChangesToLog[comment]" = "Log to /var/log/up2date which pac[WRAP]
kages has been added and removed" }
        { "writeChangesToLog" = "0" }
        {  }
        { "serverURL[comment]" = "Remote server URL" }
        { "serverURL" = "https://xmlrpc.rhn.redhat.com/XMLRPC" }
        {  }
        { "proxyPassword[comment]" = "The password to use for an authentica[WRAP]
ted proxy" }
        { "proxyPassword" = "" }
        {  }
        { "networkSetup[comment]" = "None" }
        { "networkSetup" = "1" }
        {  }
        { "proxyUser[comment]" = "The username for an authenticated proxy" [WRAP]
}
        { "proxyUser" = "" }
        {  }
        { "versionOverride[comment]" = "Override the automatically determin[WRAP]
ed system version" }
        { "versionOverride" = "" }
        {  }
        { "sslCACert[comment]" = "The CA cert used to verify the ssl server[WRAP]
" }
        { "sslCACert" = "/usr/share/rhn/RHNS-CA-CERT" }
        {  }
        { "retrieveOnly[comment]" = "Retrieve packages only" }
        { "retrieveOnly" = "0" }
        {  }
        { "debug[comment]" = "Whether or not debugging is enabled" }
        { "debug" = "0" }
        {  }
        { "httpProxy[comment]" = "HTTP proxy in host:port format, e.g. squi[WRAP]
d.redhat.com:3128" }
        { "httpProxy" = "" }
        {  }
        { "systemIdPath[comment]" = "Location of system id" }
        { "systemIdPath" = "/etc/sysconfig/rhn/systemid" }
        {  }
        { "enableProxyAuth[comment]" = "To use an authenticated proxy or no[WRAP]
t" }
        { "enableProxyAuth" = "0" }
        {  }
        { "noReboot[comment]" = "Disable the reboot actions" }
        { "noReboot" = "0" }
    )
```

### 12.4.68 1.0.0/tests/test_upstartinit.aug

```
module Test_upstartinit =
    let lns = Upstartinit.lns
    let script_line = Upstartinit.script_line
    let script = Upstartinit.script
    let lifecycle = Upstartinit.lifecycle
    let respawn = Upstartinit.respawn
    test lns get "\n" = {}
    test lns get "# bla\n" = { "#comment" = "bla" }
    test script_line get "end script\n" = *
    test script_line get "foo\n" = { "1" = "foo" }
    test script get "script\nend script\n" =  { "script" }
    test script get "script\nfoo\nend script\n" =  { "script" { "1" = "foo"[WRAP]
 } }
    test script get "script\n\nend script\n" = { "script" { "1" } }
    test script get "script\n\tfoo\nend script\n" = { "script" { "1" = "\tf[WRAP]
oo" } }
    test lns get "script\nfoo\nbar\nend script\n" =
        { "script"
            { "1" = "foo" }
            { "2" = "bar" }
        }
    test lifecycle get "post-stop exec hi\n" =
        { "post-stop"
            { "exec" = "hi" }
        }
    test lns get "post-stop exec hi\n" =
        { "post-stop"
            { "exec" = "hi" }
        }
    test lns get "exec foo bar baz\n" = { "exec" = "foo bar baz" }

    test respawn get "respawn\n" = { "respawn" }
    test respawn get "respawn foo bar baz\n" = { "respawn" = "foo bar baz" [WRAP]
}

    test lns get "# tty - getty
#
# This service maintains a getty on the specified device.

stop on runlevel [S016]

respawn
instance $TTY
exec /sbin/mingetty $TTY
usage 'tty TTY=/dev/ttyX  - where X is console id'
" = (
  { "#comment" = "tty - getty" }
  {  }
  { "#comment" = "This service maintains a getty on the specified device." [WRAP]
}
  {  }
  { "stop" = "on runlevel [S016]" }
  {  }
  { "respawn" }
  { "instance" = "$TTY" }
  { "exec" = "/sbin/mingetty $TTY" }
  { "usage" = "'tty TTY=/dev/ttyX  - where X is console id'" }
)

(*
    test lns get "
# On machines where kexec isn't going to be used, free the memory reserved [WRAP]
for it.
start on stopped rcS
task

script
if [ ! -x /sbin/kexec ] || ! chkconfig kdump 2>/dev/null ; then
echo -n \"0\" > /sys/kernel/kexec_crash_size 2>/dev/null
fi
exit 0
end script
" =
(
  {  }
  { "#comment" = "On machines where kexec isn't going to be used, free the [WRAP]
memory reserved for it." }
  {  }
  { "start" = "on stopped rcS" }
  { "task" }
  {  }
  { "script"
    { "1" = "   if [ ! -x /sbin/kexec ] || ! chkconfig kdump 2>/dev/null ; [WRAP]
then" }
    { "2" = "           echo -n \"0\" > /sys/kernel/kexec_crash_size 2>/dev[WRAP]
/null" }
    { "3" = "   fi" }
    { "4" = "   exit 0" }
  }
)

*)
```

### 12.4.69 1.2.0/lenses/abrt.aug

```
(* abrt.conf is mostly like Puppet configuration, i.e., an ini file
   with # for comments; but it can have numeric keys *)
module Abrt =
 autoload xfm
 (* allow numeric keys; IniFile.entry_re does not have 0-9 in the first [] [WRAP]
*)
 let entry_re = /[A-Za-z0-9][A-Za-z0-9\._-]+/
 let entry = IniFile.indented_entry entry_re Puppet.sep Puppet.comment
 let record = IniFile.record Puppet.title entry
 let lns = IniFile.lns record Puppet.comment
 let xfm = transform lns (incl "/etc/abrt/abrt.conf")
```

### 12.4.70 1.2.0/lenses/automaster.aug

```
module Automaster =
    autoload xfm

    let eol = Util.eol
    let comment = Util.comment
    let empty = Util.empty

    let mount_point = store /\/[^# \t\n]+/
    let include = [ label "include" .
```

```
                    del /\+[ \t]*/ "+" .
                    store /[^# \t\n]+/ .
                    eol ]
    let options = [ label "options" . store /-[^ \t\n]+/ ]
    let map_param =
        let    name = [ label "name" . store /[^: \t\n]+/ ]
        in let type = [ label "type" . store /[a-z]+/ ]
        in let format = [ label "format" . store /[a-z]+/ ]
        in let options = [ label "options" . store /[^ \t\n]+/ ]
        in let prelude = ( type .
                           ( del "," "," . format ) ? .
                           del ":" ":" )
        in ( prelude ? .
             name .
             ( Util.del_ws_spc . options ) ? )
    let map_record = [ label "map" .
                       mount_point . Util.del_ws_spc .
                       map_param .
                       eol ]

    let lns = ( map_record |
                include |
                comment |
                empty ) *

    let relevant = (incl "/etc/auto.master") .
                   Util.stdexcl
    let xfm = transform lns relevant
```

### 12.4.71 1.2.0/lenses/automounter.aug

```
(*
Module: Automounter
  Parses automounter file based maps

Author: Dominic Cleal <dcleal@redhat.com>

About: Reference
  See autofs(5)

About: License
   This file is licenced under the LGPL v2+, like the rest of Augeas.

About: Lens Usage
   To be documented

About: Configuration files
   This lens applies to /etc/auto.*, auto_*, excluding known scripts.

About: Examples
   The <Test_Automounter> file contains various examples and tests.
*)

module Automounter =
autoload xfm
```

```
(************************************************************************
 * Group:                 USEFUL PRIMITIVES
 *************************************************************************)

(* View: eol *)
let eol = Util.eol

(* View: empty *)
let empty   = Util.empty

(* View: comment *)
let comment = Util.comment

(* View: path *)
let path = /[^-+#: \t\n][^#: \t\n]*/

(* View: hostname *)
let hostname = /[^-:#\(\), \n\t][^:#\(\), \n\t]*/

(* An option label can’t contain comma, comment, equals, or space *)
let optlabel = /[^,#:\(\)= \n\t]+/
let spec    = /[^,#:\(\)= \n\t][^ \n\t]*/

(* View: weight *)
let weight = Rx.integer

(* View: map_name *)
let map_name = /[^: \t\n]+/

(* View: entry_multimount_sep
   Separator for multimount entries, permits line spanning with "\" *)
let entry_multimount_sep = del /[ \t]+(\\\\[ \t]*\n[ \t]+)?/ " "

(************************************************************************
 * Group:                 ENTRIES
 *************************************************************************)

(* View: entry_key
   Key for a map entry *)
let entry_mkey = store path

(* View: entry_path
   Path component of an entry location *)
let entry_path = [ label "path" . store path ]

(* View: entry_host
   Host component with optional weight of an entry location *)
let entry_host = [ label "host" . store hostname
                   . ( Util.del_str "(" . [ label "weight"
                       . store weight ] . Util.del_str ")" )? ]

(* View: comma_sep_list
   Parses options for filesystems *)
let comma_sep_list (l:string) =
  let value = [ label "value" . Util.del_str "=" . store Rx.neg1 ] in
    let lns = [ label l . store optlabel . value? ] in
       Build.opt_list lns Sep.comma
```

```
(* View: entry_options *)
let entry_options = Util.del_str "-" . comma_sep_list "opt" . Util.del_ws_t[WRAP]
ab

(* View: entry_location
   A single location with one or more hosts, and one path *)
let entry_location = ( entry_host . ( Sep.comma . entry_host )* )?
                       . Sep.colon . entry_path

(* View: entry_locations
   Multiple locations (each with one or more hosts), separated by spaces *)
let entry_locations = [ label "location" . counter "location"
                        . [ seq "location" . entry_location ]
                        . ( [ Util.del_ws_spc . seq "location" . entry_loca[WRAP]
tion ] )* ]

(* View: entry_multimount
   Parses one of many mountpoints given for a multimount line *)
let entry_multimount = entry_mkey . Util.del_ws_tab . entry_options? . entr[WRAP]
y_locations

(* View: entry_multimounts
   Parses multiple mountpoints given on an entry line *)
let entry_multimounts = [ label "mount" . counter "mount"
                          . [ seq "mount" . entry_multimount ]
                          . ( [ entry_multimount_sep . seq "mount" . entry_[WRAP]
multimount ] )* ]

(* View: entry
   A single map entry from start to finish, including multi-mounts *)
let entry = [ seq "entry" . entry_mkey . Util.del_ws_tab . entry_options?
              . ( entry_locations | entry_multimounts ) . Util.eol ]

(* View: include
   An include line starting with a "+" and a map name *)
let include = [ seq "entry" . store "+" . Util.del_opt_ws ""
                . [ label "map" . store map_name ] . Util.eol ]

(* View: lns *)
let lns = ( empty | comment | entry | include ) *

(* Variable: filter
   Exclude scripts/executable maps from here *)
let filter = incl "/etc/auto.*"
           . incl "/etc/auto_*"
           . excl "/etc/auto.master"
           . excl "/etc/auto_master"
           . excl "/etc/auto.net"
           . excl "/etc/auto.smb"
           . Util.stdexcl

let xfm = transform lns filter
```

### 12.4.72 1.2.0/lenses/gshadow.aug

```
(* based on the group module for Augeas by Free Ekanayaka <free@64studio.co[WRAP]
m>

 Reference: man 5 gshadow

*)

module Gshadow =

   autoload xfm

(************************************************************************
 *                           USEFUL PRIMITIVES
 *************************************************************************)

let eol        = Util.eol
let comment    = Util.comment
let empty      = Util.empty

let colon      = Sep.colon
let comma      = Sep.comma

let sto_to_spc = store Rx.space_in

let word    = Rx.word
let password = /[A-Za-z0-9_.!*\/$-]*/
let integer = Rx.integer

(************************************************************************
 *                               ENTRIES
 *************************************************************************)

let user      = [ label "user" . store word ]
let user_list = Build.opt_list user comma
let params    = [ label "password" . store password  . colon ]
                . [ label "admins" . user_list? . colon ]
                . [ label "members" . user_list? ]
let entry     = Build.key_value_line word colon params

(************************************************************************
 *                                LENS
 *************************************************************************)

let lns        = (comment|empty|entry) *

let filter
               = incl "/etc/gshadow"
               . Util.stdexcl

let xfm        = transform lns filter
```

### 12.4.73 1.2.0/lenses/kdc.aug

```
module Kdc =

autoload xfm

let comment = Krb5.comment
let empty = Krb5.empty

let simple_section = Krb5.simple_section
let kdcdefaults =
  simple_section "kdcdefaults" /kdc_ports|kdc_tcp_ports/

let realm_re = Krb5.realm_re
let entry = Krb5.entry
let eq = Krb5.eq
(* the Krb5.eq_openbr didn't have a newline at the end *)
let eq_openbr = del /[ \t]*=[ \t\n]*\{([ \t]*\n)*/ " = {\n\n"
let closebr = Krb5.closebr
let indent = Krb5.indent
let eol = Krb5.eol
let record = Krb5.record
let realms_enctypes = [ indent . key "supported_enctypes" . eq .
        [ label "type" . store /[^ \t\n#]+/ . Util.del_ws_spc ] * .
        [ label "type" . store /[^ \t\n#]+/ . eol ] ]

let realms =
  let simple_option = /master_key_type|acl_file|dict_file|admin_keytab/ in
  let list_option = /supported_enctypes/ in
  let soption = entry simple_option eq comment in
  let realm = [ indent . label "realm" . store realm_re .
                  eq_openbr . eol . (soption|realms_enctypes)* . closebr . [WRAP]
eol ] in
    record "realms" (realm|comment)


let lns = (comment|empty)* .
  (kdcdefaults|realms)*

let xfm = transform lns (incl "/var/kerberos/krb5kdc/kdc.conf")
```

### 12.4.74 1.2.0/lenses/krb5.aug

```
module Krb5 =

autoload xfm

let comment = Inifile.comment "#" "#"
let empty = Inifile.empty
let eol = Inifile.eol
let dels = Util.del_str

let indent = del /[ \t]*/ ""
let eq = del /[ \t]*=[ \t]*/ " = "
let eq_openbr = del /[ \t]*=[ \t\n]*\{([ \t]*\n)*/ " = {"
let closebr = del /[ \t]*\}/ "}"

(* These two regexps for realms and apps are not entirely true
   - strictly speaking, there's no requirement that a realm is all upper ca[WRAP]
se
   and an application only uses lowercase. But it's what's used in practice[WRAP]
.

   Without that distinction we couldn't distinguish between applications
   and realms in the [appdefaults] section.
*)

let realm_re = /[A-Z][.a-zA-Z0-9-]*/
let app_re = /[a-z][a-zA-Z0-9_]*/
let name_re = /[.a-zA-Z0-9_-]+/

let value = store /[^;# \t\n{}]+/
let entry (kw:regexp) (sep:lens) (comment:lens)
    = [ indent . key kw . sep . value . (comment|eol) ] | comment

let simple_section (n:string) (k:regexp) =
  let title = Inifile.indented_title n in
  let entry = entry k eq comment in
    Inifile.record title entry

let record (t:string) (e:lens) =
  let title = Inifile.indented_title t in
    Inifile.record title e

let libdefaults =
  let option = entry (name_re - "v4_name_convert") eq comment in
  let subsec = [ indent . key /host|plain/ . eq_openbr .
                   (entry name_re eq comment)* . closebr . eol ] in
  let v4_name_convert = [ indent . key "v4_name_convert" . eq_openbr .
                          subsec* . closebr . eol ] in
  record "libdefaults" (option|v4_name_convert)

let login =
  let keys = /krb[45]_get_tickets|krb4_convert|krb_run_aklog/
    |/aklog_path|accept_passwd/ in
    simple_section "login" keys

let appdefaults =
  let option = entry (name_re - "realm" - "application") eq comment in
  let realm = [ indent . label "realm" . store realm_re .
                  eq_openbr . option* . closebr . eol ] in
  let app = [ indent . label "application" . store app_re .
                eq_openbr . (realm|option)* . closebr . eol] in
    record "appdefaults" (option|realm|app)

let realms =
  let simple_option = /kdc|admin_server|database_module|default_domain/
      |/v4_realm|auth_to_local(_names)?|master_kdc|kpasswd_server/
      |/admin_server/ in
  let subsec_option = /v4_instance_convert/ in
  let option = entry simple_option eq comment in
  let subsec = [ indent . key subsec_option . eq_openbr .
                   (entry name_re eq comment)* . closebr . eol ] in
(* ************************ Changes applied by AFSEO are below ***********[WRAP]
 *)
  let realm = [ indent . label "realm" . store realm_re .
(*                          vvvvv                                         [WRAP]
 *)
                  eq_openbr . eol . (option|subsec)* . closebr . eol ] in
(*                          ^^^^^                                         [WRAP]
 *)
```

```
(* ************************* Changes applied by AFSEO are above ***********[WRAP]
 *)
    record "realms" (realm|comment)

let domain_realm =
  simple_section "domain_realm" name_re

let logging =
  let keys = /kdc|admin_server|default/ in
  let xchg (m:regexp) (d:string) (l:string) =
    del m d . label l in
  let xchgs (m:string) (l:string) = xchg m m l in
  let dest =
    [ xchg /FILE[=:]/ "FILE=" "file" . value ]
    |[ xchgs "STDERR" "stderr" ]
    |[ xchgs "CONSOLE" "console" ]
    |[ xchgs "DEVICE=" "device" . value ]
    |[ xchgs "SYSLOG" "syslog" .
         ([ xchgs ":" "severity" . store /[A-Za-z0-9]+/ ].
          [ xchgs ":" "facility" . store /[A-Za-z0-9]+/ ]?)? ] in
  let entry = [ indent . key keys . eq . dest . (comment|eol) ] | comment i[WRAP]
n
    record "logging" entry

let capaths =
  let realm = [ indent . key realm_re .
                  eq_openbr .
                  (entry realm_re eq comment)* . closebr . eol ] in
    record "capaths" (realm|comment)

let dbdefaults =
  let keys = /database_module|ldap_kerberos_container_dn|ldap_kdc_dn/
    |/ldap_kadmind_dn|ldap_service_password_file|ldap_servers/
    |/ldap_conns_per_server/ in
    simple_section "dbdefaults" keys

let dbmodules =
  let keys = /db_library|ldap_kerberos_container_dn|ldap_kdc_dn/
    |/ldap_kadmind_dn|ldap_service_password_file|ldap_servers/
    |/ldap_conns_per_server/ in
    simple_section "dbmodules" keys

(* This section is not documented in the krb5.conf manpage,
   but the Fermi example uses it. *)
let instance_mapping =
  let value = dels "\"" . store /[^;# \t\n{}]*/ . dels "\"" in
  let map_node = label "mapping" . store /[a-zA-Z0-9\/*]+/ in
  let mapping = [ indent . map_node . eq .
                    [ label "value" . value ] . (comment|eol) ] in
  let instance = [ indent . key name_re .
                     eq_openbr . (mapping|comment)* . closebr . eol ] in
    record "instancemapping" instance

let kdc =
  simple_section "kdc" /profile/

let lns = (comment|empty)* .
```

```
  (libdefaults|login|appdefaults|realms|domain_realm
  |logging|capaths|dbdefaults|dbmodules|instance_mapping|kdc)*

let xfm = transform lns (incl "/etc/krb5.conf")
```

### 12.4.75 1.2.0/lenses/libreport_plugins.aug

```
module Libreport_plugins =

autoload xfm

let entry = Build.key_value_line /[A-Za-z]+/ Sep.equal (store /[^\n]*[^ \t\[WRAP]
n]+/)

let lns = ( Util.comment | Util.empty | entry ) *

let filter = (incl "/etc/libreport/plugins/*.conf") . Util.stdexcl
let xfm = transform lns filter
```

### 12.4.76 1.2.0/lenses/mac_ssh.aug

```
(* Tell Augeas to use the ssh lens on Macs, where SSH configuration is dire[WRAP]
ctly
   in /etc, not in /etc/ssh. *)
module Mac_ssh =
    let lns = Ssh.lns
    let xfm = transform lns (incl "/etc/ssh_config")
```

### 12.4.77 1.2.0/lenses/mac_sshd.aug

```
(* Tell Augeas to use the sshd lens on Macs, where SSH configuration is
   directly in /etc, not in /etc/ssh. *)
module Mac_sshd =
    let lns = Sshd.lns
    let xfm = transform lns (incl "/etc/sshd_config")
```

### 12.4.78 1.2.0/lenses/mimetypes.aug

```
module Mimetypes =
    autoload xfm

    (* RFC 2045, Page 11. Closing square bracket moved out of sequence to
       satisfy regex syntax. token_first excludes pound signs so as not to
       overlap with the syntax for comments. *)
    let token =
           let first = /[^]#()<>@,;:\\"\/[?= \t\n]/
        in let rest  =  /[^]()<>@,;:\\"\/[?= \t\n]*/
        in first . rest
    (* We can't use the mime type as a key, because it has a slash in it *)
    let mime_type = store (token . "/" . token)
    (* This will split up rules wrong if you use spaces within a rule, e.g.
    "ascii(34, 3)" or "string(34,'foo bar')". But all the rules I've ever s[WRAP]
een
    were just filename extensions, so this won't fail until people forget w[WRAP]
hat
```

```
    it is and have to dig to find it. *)
    let a_rule = [ Util.del_ws_spc . label "rule" . store /[^ \t\n]+/ ]
    let rules = [ label "rules" . mime_type . (a_rule *) . Util.eol ]
    let line = ( rules | Util.comment | Util.empty )
    let lns = ( line * )

    let xfm = transform lns (incl "/etc/mime.types")
```

### 12.4.79 1.2.0/lenses/pg_ident.aug

```
module Pg_Ident =
    autoload xfm
    let identifier = store /[a-z_][^ \t\n#]*/
    let record = [ seq "entries" .
                   [ label "map" . identifier ] .
                   Util.del_ws_spc .
                   [ label "os_user" . identifier ] .
                   Util.del_ws_spc .
                   [ label "db_user" . identifier ] .
                   Util.eol
                 ]
    let empty = Util.empty
    let comment = Util.comment
    let line = empty | comment | record
    let lns = line *
    let xfm = transform lns (incl "/var/lib/pgsql/data/pg_ident.conf")
```

### 12.4.80 1.2.0/lenses/postgresql.aug

```
module Postgresql =
    autoload xfm

    let comment = Inifile.comment "#" "#"
    let empty = Inifile.empty
    let eq = del /[ \t]*=/ " ="
    let entry = IniFile.entry IniFile.entry_re eq comment

    let lns = ( entry | empty ) *

    let xfm = transform lns (incl "/var/lib/pgsql/*/postgresql.conf")
```

### 12.4.81 1.2.0/lenses/sos.aug

```
module Sos =
 autoload xfm
 let lns = Puppet.lns
 let xfm = transform lns (incl "/etc/sos.conf")
```

### 12.4.82 1.2.0/lenses/subject_mapping.aug

```
(* Parse pam_pkcs11 subject_mapping file
   File is of the format:

   Certificate Distinguished Name, With Spaces and Commas, Bla Bla. -> user[WRAP]
name
```

```
   We're interested in preserving the one-to-one property, that is, that fo[WRAP]
r a
   given username there is only one certificate. Because of this, and becau[WRAP]
se
   the username is shorter and easier to type, we make the username the key
   instead of the certificate distinguished name.
*)

module Subject_mapping =
    autoload xfm
    (* can't have slashes in keys, that's another reason to make the userna[WRAP]
me
       the key *)
    let username = key /[^>\/ \t\n-]+/
    let arrow = del /[ \t]*->[ \t]*/ " -> "
    let certdn = store /[^ \t\n]+([ \t]+[^ \t\n]+)*/
    let line = [ certdn . arrow . username . Util.eol ]

    let lns = line *

    let relevant = (incl "/etc/pam_pkcs11/subject_mapping")
    let xfm = transform lns relevant
```

### 12.4.83 1.2.0/lenses/subversion.aug

```
(* it's just an ini file. sections ("titles") are required *)
module Subversion =
    autoload xfm

    let comment = IniFile.comment "#" "#"
    let sep = IniFile.sep "=" "="
    let entry = IniFile.indented_entry IniFile.entry_re sep comment
    let title = IniFile.indented_title IniFile.record_re
    let record = IniFile.record title entry

    let lns = IniFile.lns record comment

    let relevant = ( incl "/etc/subversion/servers" ) .
                   ( incl "/etc/subversion/config" )

    let xfm = transform lns relevant
```

### 12.4.84 1.2.0/lenses/tracini.aug

```
(* This began as a copy of <Puppet> *)

module Tracini =
  autoload xfm

(************************************************************************
 * INI File settings
 *
 * puppet.conf only supports "# as commentary and "=" as separator
 *************************************************************************)
let comment    = IniFile.comment "#" "#"
let sep        = IniFile.sep "=" "="


(************************************************************************
 *                        ENTRY
 * puppet.conf uses standard INI File entries
 *************************************************************************)
(* began with IniFile.entry_re *)
(* added star as a valid non-first char in entry keys *)
(* allowed single-character entry keys *)
let entry_re           = ( /[A-Za-z][A-Za-z0-9*\._-]*/ )
let entry   = IniFile.indented_entry entry_re sep comment


(************************************************************************
 *                        RECORD
 * puppet.conf uses standard INI File records
 *************************************************************************)
let title   = IniFile.indented_title IniFile.record_re
let record  = IniFile.record title entry


(************************************************************************
 *                        LENS & FILTER
 * puppet.conf uses standard INI File records
 *************************************************************************)
let lns     = IniFile.lns record comment

let filter = (incl "/var/www/tracs/*/conf/trac.ini")

let xfm = transform lns filter
```

### 12.4.85 1.2.0/lenses/up2date.aug

```
module Up2date =
    autoload xfm

    (* funky syntax: this matches one or more of a-z, A-Z, [ or ]. *)
    let akey = /[]a-zA-Z[]+/
    let avalue = /[^ \t\n]*([ \t]+[^ \t\n]+)*/
    let setting = Build.key_value_line akey (del "=" "=") (store avalue)
    let lns = ( Util.empty | Util.comment | setting ) *

    let xfm = transform lns (incl "/etc/sysconfig/rhn/up2date")
```

### 12.4.86 1.2.0/lenses/upstartinit.aug

```
(* Upstart init configuration files such as found in /etc/init *)

module Upstartinit =
    autoload xfm

    let eol = Util.eol
    let rest_of_line = /[^ \t\n]+([ \t]+[^ \t\n]+)*/
    let whole_line_maybe_indented = /[ \t]*[^ \t\n]+([ \t]+[^ \t\n]+)*/
    let no_params = [ key "task" . eol ]
    let param_is_rest_of_line (thekey:regexp) =
        Build.key_value_line thekey
                             Util.del_ws_spc
                             (store rest_of_line)

    let respawn = [ key "respawn" .
          (Util.del_ws_spc . store rest_of_line)? . eol ]


    let one_params = param_is_rest_of_line
          ( "start"
          | "stop"
          | "env"
          | "export"
          | "normal exit"
          | "instance"
          | "description"
          | "author"
          | "version"
          | "emits"
          | "console"
          | "umask"
          | "nice"
          | "oom"
          | "chroot"
          | "chdir"
          | "limit"
          | "unlimited"
          | "kill timeout"
          | "expect"
          | "usage"
          )

    (* exec and script are valid both at the top level and as a parameter o[WRAP]
f a
    lifecycle keyword *)
    let exec = param_is_rest_of_line "exec"

    let script_line = [ seq "line" .
                        store ( whole_line_maybe_indented - "end script" ) [WRAP]
.
                        eol ] |
                      [ seq "line" . eol]
    let end_script = del "end script\n" "end script\n"
    let script = [ key "script" . eol . script_line * . end_script ]

    let lifecycle = [ key /(pre|post)-(start|stop)/ .  Util.del_ws_spc . ( [WRAP]
exec | script ) ]

    let lns = ( Util.empty
              | Util.comment
              | script
              | exec
              | lifecycle
              | no_params
              | one_params
              | respawn
              ) *

    let relevant = (incl "/etc/init/*.conf") . Util.stdexcl
    let xfm = transform lns relevant
```

### 12.4.87 1.2.0/tests/test_abrt.aug

```
module Test_abrt =
    let lns = Abrt.lns
    test lns get "
[ Common ]
# With this option set to \"yes\",
# only crashes in signed packages will be analyzed.
# the list of public keys used to check the signature is
# in the file gpg_keys
#
OpenGPGCheck = yes

# Blacklisted packages
#
BlackList = nspluginwrapper, valgrind, strace, mono-core

# Process crashes in executables which do not belong to any package?
#
ProcessUnpackaged = no

# Blacklisted executable paths (shell patterns)
#
BlackListedPaths = /usr/share/doc/*, */example*, /usr/bin/nspluginviewer

# Which database plugin to use
#
Database = SQLite3

# Enable this if you want abrtd to auto-unpack crashdump tarballs which app[WRAP]
ear
# in this directory (for example, uploaded via ftp, scp etc).
# Note: you must ensure that whatever directory you specify here exists
# and is writable for abrtd. abrtd will not create it automatically.
#
#WatchCrashdumpArchiveDir = /var/spool/abrt-upload

# Max size for crash storage [MiB] or 0 for unlimited
#
MaxCrashReportsSize = 1000

# Vector of actions and reporters which are activated immediately
# after a crash occurs, comma separated.
#
#ActionsAndReporters = Mailx(\"[abrt] new crash was detected\")
#ActionsAndReporters = FileTransfer(\"store\")
ActionsAndReporters = SOSreport


# What actions or reporters to run on each crash type
```

```
#
[ AnalyzerActionsAndReporters ]
Kerneloops = RHTSupport, Logger
CCpp = RHTSupport, Logger
Python = RHTSupport, Logger
#CCpp:xorg-x11-apps = RunApp(\"date\", \"date.txt\")


# Which Action plugins to run repeatedly
#
[ Cron ]
#   h:m - at h:m
#   s - every s seconds

120 = KerneloopsScanner

#02:00 = FileTransfer
" = (
    {  }
    { " Common "
        { "#comment" = "With this option set to "yes"," }
        { "#comment" = "only crashes in signed packages will be analyzed." [WRAP]
}
        { "#comment" = "the list of public keys used to check the signature[WRAP]
 is" }
        { "#comment" = "in the file gpg_keys" }
        { "#comment" }
        { "OpenGPGCheck" = "yes" }
        {  }
        { "#comment" = "Blacklisted packages" }
        { "#comment" }
        { "BlackList" = "nspluginwrapper, valgrind, strace, mono-core" }
        {  }
        { "#comment" = "Process crashes in executables which do not belong [WRAP]
to any package?" }
        { "#comment" }
        { "ProcessUnpackaged" = "no" }
        {  }
        { "#comment" = "Blacklisted executable paths (shell patterns)" }
        { "#comment" }
        { "BlackListedPaths" = "/usr/share/doc/*, */example*, /usr/bin/nspl[WRAP]
uginviewer" }
        {  }
        { "#comment" = "Which database plugin to use" }
        { "#comment" }
        { "Database" = "SQLite3" }
        {  }
        { "#comment" = "Enable this if you want abrtd to auto-unpack crashd[WRAP]
ump tarballs which appear" }
        { "#comment" = "in this directory (for example, uploaded via ftp, s[WRAP]
cp etc)." }
        { "#comment" = "Note: you must ensure that whatever directory you s[WRAP]
pecify here exists" }
        { "#comment" = "and is writable for abrtd. abrtd will not create it[WRAP]
 automatically." }
        { "#comment" }
        { "#comment" = "WatchCrashdumpArchiveDir = /var/spool/abrt-upload" [WRAP]
}
        {  }
        { "#comment" = "Max size for crash storage [MiB] or 0 for unlimited[WRAP]
" }
        { "#comment" }
        { "MaxCrashReportsSize" = "1000" }
        {  }
        { "#comment" = "Vector of actions and reporters which are activated[WRAP]
 immediately" }
        { "#comment" = "after a crash occurs, comma separated." }
        { "#comment" }
        { "#comment" = "ActionsAndReporters = Mailx("[abrt] new crash was d[WRAP]
etected")" }
        { "#comment" = "ActionsAndReporters = FileTransfer("store")" }
        { "ActionsAndReporters" = "SOSreport" }
        {  }
        {  }
        { "#comment" = "What actions or reporters to run on each crash type[WRAP]
" }
        { "#comment" }
    }
    { " AnalyzerActionsAndReporters "
        { "Kerneloops" = "RHTSupport, Logger" }
        { "CCpp" = "RHTSupport, Logger" }
        { "Python" = "RHTSupport, Logger" }
        { "#comment" = "CCpp:xorg-x11-apps = RunApp("date", "date.txt")" }
        {  }
        {  }
        { "#comment" = "Which Action plugins to run repeatedly" }
        { "#comment" }
    }
    { " Cron "
        { "#comment" = "h:m - at h:m" }
        { "#comment" = "s - every s seconds" }
        {  }
        { "120" = "KerneloopsScanner" }
        {  }
        { "#comment" = "02:00 = FileTransfer" }
    }
)
```

### 12.4.88 1.2.0/tests/test_automaster.aug

```
module Test_automaster =
    let map_param = Automaster.map_param
    let map_record = Automaster.map_record
    let lns = Automaster.lns

    test map_param get "file:/bla/blu" =
        ( { "type" = "file" } { "name" = "/bla/blu" } )
    test map_param get "yp,hesiod:/bla/blu" =
        ( { "type" = "yp" }
          { "format" = "hesiod" }
          { "name" = "/bla/blu" } )
    test map_param get "bla" = { "name" = "bla" }
    test map_record get "/net /etc/auto.net\n" =
        { "map" = "/net"
```

```
            { "name" = "/etc/auto.net" } }

    test lns get "# c\n+auto.master\n/net /etc/auto.net\n\n" = (
        { "#comment" = "c" }
        { "include" = "auto.master" }
        { "map" = "/net"
            { "name" = "/etc/auto.net" }
        }
        {  } )

    test lns get "# c
+auto.master
# blank line


/net /etc/auto.net
/foo bla
" = (
  { "#comment" = "c" }
  { "include" = "auto.master" }
  { "#comment" = "blank line" }
  {  }
  {  }
  { "map" = "/net"
    { "name" = "/etc/auto.net" }
  }
  { "map" = "/foo"
    { "name" = "bla" }
  }
)

    test lns get "#
# Sample auto.master file
# This is an automounter map and it has the following format
# key [ -mount-options-separated-by-comma ] location
# For details of the format look at autofs(5).
#
/misc   /etc/auto.misc
#
# NOTE: mounts done from a hosts map will be mounted with the
#       \"nosuid\" and \"nodev\" options unless the \"suid\" and \"dev\"
#       options are explicitly given.
#
/net    -hosts
#
# Include central master map if it can be found using
# nsswitch sources.
#
# Note that if there are entries for /net or /misc (as
# above) in the included master map any keys that are the
# same will not be seen as the first read key seen takes
# precedence.
#
+auto.master
" = (
  {  }
  { "#comment" = "Sample auto.master file" }
  { "#comment" = "This is an automounter map and it has the following forma[WRAP]
t" }
  { "#comment" = "key [ -mount-options-separated-by-comma ] location" }
  { "#comment" = "For details of the format look at autofs(5)." }
  {  }
  { "map" = "/misc"
    { "name" = "/etc/auto.misc" }
  }
  {  }
  { "#comment" = "NOTE: mounts done from a hosts map will be mounted with t[WRAP]
he" }
  { "#comment" = "\"nosuid\" and \"nodev\" options unless the \"suid\" and [WRAP]
\"dev\"" }
  { "#comment" = "options are explicitly given." }
  {  }
  { "map" = "/net"
    { "name" = "-hosts" }
  }
  {  }
  { "#comment" = "Include central master map if it can be found using" }
  { "#comment" = "nsswitch sources." }
  {  }
  { "#comment" = "Note that if there are entries for /net or /misc (as" }
  { "#comment" = "above) in the included master map any keys that are the" [WRAP]
}
  { "#comment" = "same will not be seen as the first read key seen takes" }
  { "#comment" = "precedence." }
  {  }
  { "include" = "auto.master" }
)
```

### 12.4.89 1.2.0/tests/test_gshadow.aug

```
module Test_gshadow =
   let lns = Gshadow.lns
   let entry = Gshadow.entry
   test entry get "root:::\n" =
  { "root"
    { "password" = "" }
    { "admins" }
    { "members" }
  }

   test entry get "bin:::bin,daemon\n" =
  { "bin"
    { "password" = "" }
    { "admins" }
    { "members"
      { "user" = "bin" }
      { "user" = "daemon" }
    }
  }

   test entry get "dbus:!::\n" =
  { "dbus"
    { "password" = "!" }
    { "admins" }
    { "members" }
  }

   test entry get "ntp:!:foo,bar:baz,bletch\n" =
  { "ntp"
    { "password" = "!" }
    { "admins"
      { "user" = "foo" }
      { "user" = "bar" }
    }
    { "members"
      { "user" = "baz" }
      { "user" = "bletch" }
    }
  }

   test entry get "fooz:$5$GQPAI/174dH/Q$dQtmrhcGuolwm7DlKVFkeH.VCWbH1/XTYk[WRAP]
XU83WkIO9::\n" =
  { "fooz"
    { "password" = "$5$GQPAI/174dH/Q$dQtmrhcGuolwm7DlKVFkeH.VCWbH1/XTYkXU83[WRAP]
WkIO9" }
    { "admins" }
    { "members" }
  }




   test lns get
"root:::
bin:::bin,daemon
dbus:!::
ntp:!:foo,bar:baz,bletch
fooz:$5$GQPAI/174dH/Q$dQtmrhcGuolwm7DlKVFkeH.VCWbH1/XTYkXU83WkIO9::
" =
  { "root"
    { "password" = "" }
    { "admins" }
    { "members" }
  }
  { "bin"
    { "password" = "" }
    { "admins" }
    { "members"
      { "user" = "bin" }
      { "user" = "daemon" }
    }
  }
  { "dbus"
    { "password" = "!" }
    { "admins" }
    { "members" }
  }
  { "ntp"
    { "password" = "!" }
    { "admins"
```

```
      { "user" = "foo" }
      { "user" = "bar" }
    }
    { "members"
      { "user" = "baz" }
      { "user" = "bletch" }
    }
  }
  { "fooz"
    { "password" = "$5$GQPAI/174dH/Q$dQtmrhcGuolwm7DlKVFkeH.VCWbH1/XTYkXU83[WRAP]
WkIO9" }
    { "admins" }
    { "members" }
  }
```

### 12.4.90 1.2.0/tests/test_kdc.aug

```
module Test_kdc =
   let lns = Kdc.lns
   let realms_enctypes = Kdc.realms_enctypes
   test realms_enctypes get " supported_enctypes = aes256-cts:normal aes128[WRAP]
-cts:normal des3-hmac-sha1:normal arcfour-hmac:normal des-hmac-sha1:normal [WRAP]
des-cbc-md5:normal des-cbc-crc:normal
" =
  { "supported_enctypes"
    { "type" = "aes256-cts:normal" }
    { "type" = "aes128-cts:normal" }
    { "type" = "des3-hmac-sha1:normal" }
    { "type" = "arcfour-hmac:normal" }
    { "type" = "des-hmac-sha1:normal" }
    { "type" = "des-cbc-md5:normal" }
    { "type" = "des-cbc-crc:normal" }
  }


   test lns get "
[kdcdefaults]
 kdc_ports = 88
 kdc_tcp_ports = 88

[realms]
 EXAMPLE.COM = {
  #master_key_type = aes256-cts
  acl_file = /var/kerberos/krb5kdc/kadm5.acl
  dict_file = /usr/share/dict/words
  admin_keytab = /var/kerberos/krb5kdc/kadm5.keytab
  supported_enctypes = aes256-cts:normal aes128-cts:normal des3-hmac-sha1:n[WRAP]
ormal arcfour-hmac:normal des-hmac-sha1:normal des-cbc-md5:normal des-cbc-c[WRAP]
rc:normal
 }
" = (
  {  }
  { "kdcdefaults"
    { "kdc_ports" = "88" }
    { "kdc_tcp_ports" = "88" }
    {  }
  }
  { "realms"
    { "realm" = "EXAMPLE.COM"
      { "#comment" = "master_key_type = aes256-cts" }
      { "acl_file" = "/var/kerberos/krb5kdc/kadm5.acl" }
      { "dict_file" = "/usr/share/dict/words" }
      { "admin_keytab" = "/var/kerberos/krb5kdc/kadm5.keytab" }
      { "supported_enctypes"
        { "type" = "aes256-cts:normal" }
        { "type" = "aes128-cts:normal" }
        { "type" = "des3-hmac-sha1:normal" }
        { "type" = "arcfour-hmac:normal" }
        { "type" = "des-hmac-sha1:normal" }
        { "type" = "des-cbc-md5:normal" }
        { "type" = "des-cbc-crc:normal" }
      }
    }
  }
)

    test lns put "" after
        set "realms/realm[999]" "FOO.BAR.EXAMPLE.COM"
    = "[realms]
FOO.BAR.EXAMPLE.COM = {
}
"

    test lns put "[realms]
FOO.BAR.EXAMPLE.COM = {
}" after
        set "realms/realm[.='FOO.BAR.EXAMPLE.COM']/acl_file" "/var/kerberos[WRAP]
/krb5kdc/kadm5.acl"
    = "[realms]
FOO.BAR.EXAMPLE.COM = {
acl_file = /var/kerberos/krb5kdc/kadm5.acl
}
"
```

### 12.4.91 1.2.0/tests/test_libreport_plugins.aug

```
module Test_libreport_plugins =

    let lns = Libreport_plugins.lns
    let entry = Libreport_plugins.entry

    test entry get "Foo=bar\n" = ( { "Foo" = "bar" } )
    test lns get "
# String parameters:

Subject=bla
# EmailFrom=
" = (
  { }
  { "#comment" = "String parameters:" }
  { }
  { "Subject" = "bla" }
  { "#comment" = "EmailFrom=" }
)
```

### 12.4.92 1.2.0/tests/test_mimetypes.aug

```
module Test_mimetypes =
    let mime_type = Mimetypes.mime_type
    let rules = Mimetypes.rules
    let lns = Mimetypes.lns

    test [ mime_type ] get "text/plain" = { = "text/plain" }
    test [ mime_type ] get "application/beep+xml" = { = "application/beep+x[WRAP]
ml" }
    test [ mime_type ] get "application/vnd.fdf" = { = "application/vnd.fdf[WRAP]
" }
    (* who in their right mind made this mime type?! ... oh wait, they were[WRAP]
n't,
       it's microsoft *)
    test [ mime_type ] get
        "application/vnd.openxmlformats-officedocument.wordprocessingml.doc[WRAP]
ument" =
        { = "application/vnd.openxmlformats-officedocument.wordprocessingml[WRAP]
.document" }
    test rules get "text/plain txt\n" =
        { "rules" = "text/plain"
          { "rule" = "txt" } }
    test rules get "application/vnd.openxmlformats-officedocument.wordproce[WRAP]
ssingml.document docx\n" =
        { "rules" = "application/vnd.openxmlformats-officedocument.wordproc[WRAP]
essingml.document"
          { "rule" = "docx" } }
    test rules get "video/mpeg                    mpeg mpg mpe\n" =
        { "rules" = "video/mpeg"
          { "rule" = "mpeg" }
          { "rule" = "mpg" }
          { "rule" = "mpe" } }
    test lns get "
# This is a comment. I love comments.

# This file controls what Internet media types are sent to the client for
# given file extension(s).  Sending the correct media type to the client
# is important so they know how to handle the content of the file.
# Extra types can either be added here or by using an AddType directive
# in your config files. For more information about Internet media types,
# please read RFC 2045, 2046, 2047, 2048, and 2077.  The Internet media typ[WRAP]
e
# registry is at <http://www.iana.org/assignments/media-types/>.

# MIME type                     Extension
application/EDI-Consent
application/andrew-inset        ez
application/mac-binhex40        hqx
application/mac-compactpro      cpt
application/octet-stream        bin dms lha lzh exe class so dll img iso
application/ogg                 ogg

" = (
  {  }
  { "#comment" = "This is a comment. I love comments." }
  {  }
  { "#comment" = "This file controls what Internet media types are sent to [WRAP]
the client for" }
  { "#comment" = "given file extension(s).  Sending the correct media type [WRAP]
to the client" }
  { "#comment" = "is important so they know how to handle the content of th[WRAP]
e file." }
  { "#comment" = "Extra types can either be added here or by using an AddTy[WRAP]
pe directive" }
  { "#comment" = "in your config files. For more information about Internet[WRAP]
 media types," }
  { "#comment" = "please read RFC 2045, 2046, 2047, 2048, and 2077.  The In[WRAP]
ternet media type" }
  { "#comment" = "registry is at <http://www.iana.org/assignments/media-typ[WRAP]
es/>." }
  {  }
  { "#comment" = "MIME type                    Extension" }
  { "rules" = "application/EDI-Consent" }
  { "rules" = "application/andrew-inset"
    { "rule" = "ez" }
  }
  { "rules" = "application/mac-binhex40"
    { "rule" = "hqx" }
  }
  { "rules" = "application/mac-compactpro"
    { "rule" = "cpt" }
  }
  { "rules" = "application/octet-stream"
    { "rule" = "bin" }
    { "rule" = "dms" }
    { "rule" = "lha" }
    { "rule" = "lzh" }
    { "rule" = "exe" }
    { "rule" = "class" }
    { "rule" = "so" }
    { "rule" = "dll" }
    { "rule" = "img" }
    { "rule" = "iso" }
  }
  { "rules" = "application/ogg"
    { "rule" = "ogg" }
  }
  {  }
)

    test lns put "" after
             set "/rules[.=\"application/mac-binhex40\"]"
                 "application/mac-binhex40" ;
             set "/rules[.=\"application/mac-binhex40\"]/rule"
                 "hqx"
        = "application/mac-binhex40 hqx\n"
```

### 12.4.93 1.2.0/tests/test_pg_ident.aug

```
module Test_pg_ident =
    let empty = Pg_ident.empty
    let record = Pg_ident.record
    let lns = Pg_ident.lns

    test empty get "\n" = {}
    test record get "\n" = *
    test lns get "
# This is a comment
a b c
" = (
  { }
  { "#comment" = "This is a comment" }
  { "1"
    { "map" = "a" }
    { "os_user" = "b" }
    { "db_user" = "c" }
  }
)
```

### 12.4.94 1.2.0/tests/test_postgresql.aug

```
module Test_postgresql =
    let empty = Postgresql.empty
    let entry = Postgresql.entry
    let lns = Postgresql.lns

    test empty get "\n" = {}
    test entry get "\n" = *
    test lns get "
# This is a comment
setting = value
" = (
  { }
  { "#comment" = "This is a comment" }
  { "setting" = "value" }
)

    test lns get "
setting = value # same-line comment
" = (
  { }
  { "setting" = "value"
    { "#comment" = "same-line comment" }
  }
)

    (* i guess IniFile isn’t so smart as to remove and re-add quotes *)
    test lns get "
setting = \"value with spaces\"
" = (
  { }
  { "setting" = "\"value with spaces\"" }
)

    (* nor to ignore comment characters inside quotes *)
    test lns get "
```

```
setting = \"value with # bla\" # psyche out
" = (
  { }
  { "setting" = "\"value with"
    { "#comment" = "bla\" # psyche out" }
  }
)

    test lns get "

#--------------------------------------------------------------------------[WRAP]
----
# CLIENT CONNECTION DEFAULTS
#--------------------------------------------------------------------------[WRAP]
----

# These settings are initialized by initdb, but they can be changed.
lc_messages = ’en_US.UTF-8’                     # locale for system error m[WRAP]
essage
                                        # strings
lc_monetary = ’en_US.UTF-8’                     # locale for monetary forma[WRAP]
tting
lc_numeric = ’en_US.UTF-8’                      # locale for number formatt[WRAP]
ing
lc_time = ’en_US.UTF-8’                         # locale for time formattin[WRAP]
g

# default configuration for text search
default_text_search_config = ’pg_catalog.english’

# - Other Defaults -

#dynamic_library_path = ’$libdir’
#local_preload_libraries = ’’
" = (
  {  }
  {  }
  { "#comment" = "---------------------------------------------------------[WRAP]
---------------------" }
  { "#comment" = "CLIENT CONNECTION DEFAULTS" }
  { "#comment" = "---------------------------------------------------------[WRAP]
---------------------" }
  {  }
  { "#comment" = "These settings are initialized by initdb, but they can be[WRAP]
 changed." }
  { "lc_messages" = "’en_US.UTF-8’"
    { "#comment" = "locale for system error message" }
  }
  { "#comment" = "strings" }
  { "lc_monetary" = "’en_US.UTF-8’"
    { "#comment" = "locale for monetary formatting" }
  }
  { "lc_numeric" = "’en_US.UTF-8’"
    { "#comment" = "locale for number formatting" }
  }
  { "lc_time" = "’en_US.UTF-8’"
    { "#comment" = "locale for time formatting" }
  }
  {  }
  { "#comment" = "default configuration for text search" }
  { "default_text_search_config" = "'pg_catalog.english'" }
  {  }
  { "#comment" = "- Other Defaults -" }
  {  }
  { "#comment" = "dynamic_library_path = '$libdir'" }
  { "#comment" = "local_preload_libraries = ''" }
)
```

### 12.4.95 1.2.0/tests/test_ssh_config.aug

```
module Test_ssh_config =
    let host = Ssh_config.host
    let anything_but_host = Ssh_config.anything_but_host
    let toplevel_stanza = Ssh_config.toplevel_stanza
    let host_stanza = Ssh_config.host_stanza
    let lns = Ssh_config.lns

    test [host] get "Host *\n" =
        { "Host" = "*" }
    test [host] get "Host *.co.uk\n" =
        { "Host" = "*.co.uk" }
    test [host] get "Host 192.168.0.?\n" =
        { "Host" = "192.168.0.?" }
    test [host] get "host foo.example.com\n" =
        { "Host" = "foo.example.com" }
    test [host] get "   hOsT flarble\n" =
        { "Host" = "flarble" }


    test [anything_but_host] get "F 1\n" =
        { "F" = "1" }
    test [anything_but_host] get "BindAddress 127.0.0.1\n" =
        { "BindAddress" = "127.0.0.1" }
    test [anything_but_host] get "ForYou two words\n" =
        { "ForYou" = "two words" }


    test toplevel_stanza get "Line 1
                              User flarble
                              # A comment

                              Key Value\n" =
        { "toplevel"
            { "Line" = "1" }
            { "User" = "flarble" }
            { "#comment" = "A comment" }
            {  }
            { "Key" = "Value" }
        }

    test host_stanza get "Host mumble
                              User flarble
                              # A comment
                              Key Value\n" =
        { "Host" = "mumble"
            { "User" = "flarble" }
            { "#comment" = "A comment" }
            {  }
            { "Key" = "Value" }
        }

    (* keys can contain digits! *)
    test host_stanza get "Host *
                      User flarble
                      GSSAPIAuthentication yes
                      ForwardX11Trusted yes\n" =
        { "Host" = "*"
            { "User" = "flarble" }
            { "GSSAPIAuthentication" = "yes" }
            { "ForwardX11Trusted" = "yes" }
        }


    test lns get "
# $OpenBSD: ssh_config,v 1.25 2009/02/17 01:28:32 djm Exp $

# This is the ssh client system-wide configuration file.  See
# ssh_config(5) for more information.  This file provides defaults for
# users, and the values can be changed in per-user configuration files
# or on the command line.

# Configuration data is parsed as follows:
#  1. command line options
#  2. user-specific file
#  3. system-wide file
# Any configuration value is only changed the first time it is set.
# Thus, host-specific definitions should be at the beginning of the
# configuration file, and defaults at the end.

# Site-wide defaults for some commonly used options.  For a comprehensive
# list of available options, their meanings and defaults, please see the
# ssh_config(5) man page.

# Host *
#   ForwardAgent no
#   ForwardX11 no
#   RhostsRSAAuthentication no
#   RSAAuthentication yes
#   PasswordAuthentication yes
#   HostbasedAuthentication no
#   GSSAPIAuthentication no
#   GSSAPIDelegateCredentials no
#   GSSAPIKeyExchange no
#   GSSAPITrustDNS no
#   BatchMode no
#   CheckHostIP yes
#   AddressFamily any
#   ConnectTimeout 0
#   StrictHostKeyChecking ask
#   IdentityFile ~/.ssh/identity
#   IdentityFile ~/.ssh/id_rsa
#   IdentityFile ~/.ssh/id_dsa
#   Port 22
#   Protocol 2,1
#   Cipher 3des
#   Ciphers aes128-ctr,aes192-ctr,aes256-ctr,arcfour256,arcfour128,aes128-c[WRAP]
bc,3des-cbc
#   MACs hmac-md5,hmac-sha1,umac-64@openssh.com,hmac-ripemd160
#   EscapeChar ~
#   Tunnel no
#   TunnelDevice any:any
#   PermitLocalCommand no
#   VisualHostKey no
Host *
GSSAPIAuthentication yes
# If this option is set to yes then remote X11 clients will have full acces[WRAP]
s
# to the original X11 display. As virtually no X11 client supports the untr[WRAP]
usted
# mode correctly we set this to yes.
ForwardX11Trusted yes
# Send locale-related environment variables
SendEnv LANG LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONETARY LC_MESSAGE[WRAP]
S
SendEnv LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUREMENT
SendEnv LC_IDENTIFICATION LC_ALL LANGUAGE
SendEnv XMODIFIERS
" =

    { "toplevel"
        {  }
        { "#comment" = "$OpenBSD: ssh_config,v 1.25 2009/02/17 01:28:32 djm[WRAP]
 Exp $" }
        {  }
        { "#comment" = "This is the ssh client system-wide configuration fi[WRAP]
le.  See" }
        { "#comment" = "ssh_config(5) for more information.  This file prov[WRAP]
ides defaults for" }
        { "#comment" = "users, and the values can be changed in per-user co[WRAP]
nfiguration files" }
        { "#comment" = "or on the command line." }
        {  }
        { "#comment" = "Configuration data is parsed as follows:" }
        { "#comment" = "1. command line options" }
        { "#comment" = "2. user-specific file" }
        { "#comment" = "3. system-wide file" }
        { "#comment" = "Any configuration value is only changed the first t[WRAP]
ime it is set." }
        { "#comment" = "Thus, host-specific definitions should be at the be[WRAP]
ginning of the" }
        { "#comment" = "configuration file, and defaults at the end." }
        {  }
        { "#comment" = "Site-wide defaults for some commonly used options. [WRAP]
 For a comprehensive" }
        { "#comment" = "list of available options, their meanings and defau[WRAP]
lts, please see the" }
```

```
        { "#comment" = "ssh_config(5) man page." }
        {  }
        { "#comment" = "Host *" }
        { "#comment" = "ForwardAgent no" }
        { "#comment" = "ForwardX11 no" }
        { "#comment" = "RhostsRSAAuthentication no" }
        { "#comment" = "RSAAuthentication yes" }
        { "#comment" = "PasswordAuthentication yes" }
        { "#comment" = "HostbasedAuthentication no" }
        { "#comment" = "GSSAPIAuthentication no" }
        { "#comment" = "GSSAPIDelegateCredentials no" }
        { "#comment" = "GSSAPIKeyExchange no" }
        { "#comment" = "GSSAPITrustDNS no" }
        { "#comment" = "BatchMode no" }
        { "#comment" = "CheckHostIP yes" }
        { "#comment" = "AddressFamily any" }
        { "#comment" = "ConnectTimeout 0" }
        { "#comment" = "StrictHostKeyChecking ask" }
        { "#comment" = "IdentityFile ~/.ssh/identity" }
        { "#comment" = "IdentityFile ~/.ssh/id_rsa" }
        { "#comment" = "IdentityFile ~/.ssh/id_dsa" }
        { "#comment" = "Port 22" }
        { "#comment" = "Protocol 2,1" }
        { "#comment" = "Cipher 3des" }
        { "#comment" = "Ciphers aes128-ctr,aes192-ctr,aes256-ctr,arcfour256[WRAP]
,arcfour128,aes128-cbc,3des-cbc" }
        { "#comment" = "MACs hmac-md5,hmac-sha1,umac-64@openssh.com,hmac-ri[WRAP]
pemd160" }
        { "#comment" = "EscapeChar ~" }
        { "#comment" = "Tunnel no" }
        { "#comment" = "TunnelDevice any:any" }
        { "#comment" = "PermitLocalCommand no" }
        { "#comment" = "VisualHostKey no" }
    }
    { "Host" = "*"
        { "GSSAPIAuthentication" = "yes" }
        { "#comment" = "If this option is set to yes then remote X11 client[WRAP]
s will have full access" }
        { "#comment" = "to the original X11 display. As virtually no X11 cl[WRAP]
ient supports the untrusted" }
        { "#comment" = "mode correctly we set this to yes." }
        { "ForwardX11Trusted" = "yes" }
        { "#comment" = "Send locale-related environment variables" }
        { "SendEnv" = "LANG LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONET[WRAP]
ARY LC_MESSAGES" }
        { "SendEnv" = "LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUREM[WRAP]
ENT" }
        { "SendEnv" = "LC_IDENTIFICATION LC_ALL LANGUAGE" }
        { "SendEnv" = "XMODIFIERS" }
    }
```

### 12.4.96 1.2.0/tests/test_subject_mapping.aug

```
module Test_subject_mapping =
    let username = Subject_mapping.username
    let arrow = Subject_mapping.arrow
    let certdn = Subject_mapping.certdn
    let line = Subject_mapping.line

    test [ username ] get "foo" = { "foo" }
    test [ arrow ] get " -> " = {}
    test [ arrow ] get "\t->\t" = {}
    test [ arrow . username ] get "\t->\tfoo" = { "foo" }
    test [ certdn ] get "foo" = { = "foo" }
    test [ certdn ] get "foo bar" = { = "foo bar" }
    test line get "foo -> bar\n" = { "bar" = "foo" }
    test line get "Really Odd, Certificate Name. /#$%^&* -> un61\n" =
        { "un61" = "Really Odd, Certificate Name. /#$%^&*" }
```

### 12.4.97 1.2.0/tests/test_subversion.aug

```
module Test_subversion =
    let lns = Subversion.lns
    test lns get "
[global]
foo = bar
" = (
  { }
  { "global"
    { "foo" = "bar" }
  }
)
```

### 12.4.98 1.2.0/tests/test_tracini.aug

```
module Test_tracini =
    let lns = Tracini.lns
    test lns get "
# -*- coding: utf-8 -*-

[attachment]
max_size = 262144
render_unsafe_content = false

[browser]
hide_properties = svk:merge

[components]
tracgantt.* = enabled

[gantt-charts]
date_format = %Y/%m/%d
include_summary = true
show_opened = true
summary_length = 32
use_creation_date = true

[header_logo]
alt = Trac
height = 73
link = http://trac.edgewall.com/
```

```
src = common/trac_banner.png
width = 236

[intertrac]
z = zarquon
zarquon = zarquon
zarquon.title = Zarquon
zarquon.url = https://one.example.com/projects/zarquon
m = mahershalalhashbaz
mahershalalhashbaz = mahershalalhashbaz
mahershalalhashbaz.title = Mahershalalhashbaz trac
mahershalalhashbaz.url = https://two.example.com/projects/mahershalalhashba[WRAP]
z

[logging]
log_file = trac.log
log_level = DEBUG
log_type = none

[mimeviewer]
enscript_path = enscript
max_preview_size = 262144
php_path = php
tab_width = 8

[notification]
always_notify_owner = true
always_notify_reporter = true
smtp_always_cc =
smtp_defaultdomain = example.com
smtp_enabled = true
smtp_from = zarquon-trac@example.com
smtp_password =
smtp_port = 25
smtp_replyto = onewebmaster@example.com
smtp_server = localhost
smtp_user =

[project]
descr = Zarquon
footer = Visit the Trac open source project at<br /><a href=\"http://trac.e[WRAP]
dgewall.com/\">http://trac.edgewall.com/</a>
icon = common/trac.ico
name = Zarquon
url = https://one.example.com/projects/zarquon/

[ticket]
default_component = component1
default_milestone =
default_priority = major
default_type = defect
default_version =
restrict_owner = false

[ticket-custom]
dependencies = text
dependencies.label = Dependencies
dependencies.value =
due_assign = text
due_assign.label = Due to assign
due_assign.value = YYYY/MM/DD
due_close = text
due_close.label = Due to close
due_close.value = YYYY/MM/DD
include_gantt = checkbox
include_gantt.label = Include in GanttChart
include_gantt.value =

[ticket-workflow]
accept = new -> assigned
accept.operations = set_owner_to_self
accept.permissions = TICKET_MODIFY
leave = * -> *
leave.default = 1
leave.operations = leave_status
reassign = new,assigned,reopened -> new
reassign.operations = set_owner
reassign.permissions = TICKET_MODIFY
reopen = closed -> reopened
reopen.operations = del_resolution
reopen.permissions = TICKET_CREATE
resolve = new,assigned,reopened -> closed
resolve.operations = set_resolution
resolve.permissions = TICKET_MODIFY

[timeline]
changeset_show_files = 0
default_daysback = 30
ticket_show_details = false

[trac]
check_auth_ip = true
database = sqlite:db/trac.db
default_charset = iso-8859-15
default_handler = WikiModule
ignore_auth_case = false
mainnav = wiki,timeline,roadmap,browser,tickets,newticket,search
metanav = login,logout,settings,help,about
permission_store = DefaultPermissionStore
repository_dir = /var/www/svn/ftdb
templates_dir = /usr/share/trac/templates

[wiki]
ignore_missing_pages = false
" = (
  {  }
  { "#comment" = "-*- coding: utf-8 -*-" }
  {  }
  { "attachment"
    { "max_size" = "262144" }
    { "render_unsafe_content" = "false" }
    {  }
  }
  { "browser"
```

```
    { "hide_properties" = "svk:merge" }
    {  }
  }
  { "components"
    { "tracgantt.*" = "enabled" }
    {  }
  }
  { "gantt-charts"
    { "date_format" = "%Y/%m/%d" }
    { "include_summary" = "true" }
    { "show_opened" = "true" }
    { "summary_length" = "32" }
    { "use_creation_date" = "true" }
    {  }
  }
  { "header_logo"
    { "alt" = "Trac" }
    { "height" = "73" }
    { "link" = "http://trac.edgewall.com/" }
    { "src" = "common/trac_banner.png" }
    { "width" = "236" }
    {  }
  }
  { "intertrac"
    { "z" = "zarquon" }
    { "zarquon" = "zarquon" }
    { "zarquon.title" = "Zarquon" }
    { "zarquon.url" = "https://one.example.com/projects/zarquon" }
    { "m" = "mahershalalhashbaz" }
    { "mahershalalhashbaz" = "mahershalalhashbaz" }
    { "mahershalalhashbaz.title" = "Mahershalalhashbaz trac" }
    { "mahershalalhashbaz.url" = "https://two.example.com/projects/mahersha[WRAP]
lalhashbaz" }
    {  }
  }
  { "logging"
    { "log_file" = "trac.log" }
    { "log_level" = "DEBUG" }
    { "log_type" = "none" }
    {  }
  }
  { "mimeviewer"
    { "enscript_path" = "enscript" }
    { "max_preview_size" = "262144" }
    { "php_path" = "php" }
    { "tab_width" = "8" }
    {  }
  }
  { "notification"
    { "always_notify_owner" = "true" }
    { "always_notify_reporter" = "true" }
    { "smtp_always_cc" }
    { "smtp_defaultdomain" = "example.com" }
    { "smtp_enabled" = "true" }
    { "smtp_from" = "zarquon-trac@example.com" }
    { "smtp_password" }
    { "smtp_port" = "25" }
    { "smtp_replyto" = "onewebmaster@example.com" }
    { "smtp_server" = "localhost" }
    { "smtp_user" }
    {  }
  }
  { "project"
    { "descr" = "Zarquon" }
    { "footer" = "Visit the Trac open source project at<br /><a href=\"http[WRAP]
://trac.edgewall.com/\">http://trac.edgewall.com/</a>" }
    { "icon" = "common/trac.ico" }
    { "name" = "Zarquon" }
    { "url" = "https://one.example.com/projects/zarquon/" }
    {  }
  }
  { "ticket"
    { "default_component" = "component1" }
    { "default_milestone" }
    { "default_priority" = "major" }
    { "default_type" = "defect" }
    { "default_version" }
    { "restrict_owner" = "false" }
    {  }
  }
  { "ticket-custom"
    { "dependencies" = "text" }
    { "dependencies.label" = "Dependencies" }
    { "dependencies.value" }
    { "due_assign" = "text" }
    { "due_assign.label" = "Due to assign" }
    { "due_assign.value" = "YYYY/MM/DD" }
    { "due_close" = "text" }
    { "due_close.label" = "Due to close" }
    { "due_close.value" = "YYYY/MM/DD" }
    { "include_gantt" = "checkbox" }
    { "include_gantt.label" = "Include in GanttChart" }
    { "include_gantt.value" }
    {  }
  }
  { "ticket-workflow"
    { "accept" = "new -> assigned" }
    { "accept.operations" = "set_owner_to_self" }
    { "accept.permissions" = "TICKET_MODIFY" }
    { "leave" = "* -> *" }
    { "leave.default" = "1" }
    { "leave.operations" = "leave_status" }
    { "reassign" = "new,assigned,reopened -> new" }
    { "reassign.operations" = "set_owner" }
    { "reassign.permissions" = "TICKET_MODIFY" }
    { "reopen" = "closed -> reopened" }
    { "reopen.operations" = "del_resolution" }
    { "reopen.permissions" = "TICKET_CREATE" }
    { "resolve" = "new,assigned,reopened -> closed" }
    { "resolve.operations" = "set_resolution" }
    { "resolve.permissions" = "TICKET_MODIFY" }
    {  }
  }
  { "timeline"
```

```
    { "changeset_show_files" = "0" }
    { "default_daysback" = "30" }
    { "ticket_show_details" = "false" }
    {  }
  }
  { "trac"
    { "check_auth_ip" = "true" }
    { "database" = "sqlite:db/trac.db" }
    { "default_charset" = "iso-8859-15" }
    { "default_handler" = "WikiModule" }
    { "ignore_auth_case" = "false" }
    { "mainnav" = "wiki,timeline,roadmap,browser,tickets,newticket,search" [WRAP]
}
    { "metanav" = "login,logout,settings,help,about" }
    { "permission_store" = "DefaultPermissionStore" }
    { "repository_dir" = "/var/www/svn/ftdb" }
    { "templates_dir" = "/usr/share/trac/templates" }
    {  }
  }
  { "wiki"
    { "ignore_missing_pages" = "false" }
  }
)
```

### 12.4.99 1.2.0/tests/test_up2date.aug

```
module Test_up2date =
    let akey = Up2date.akey
    let avalue = Up2date.avalue
    let setting = Up2date.setting
    let lns = Up2date.lns

    test [key akey] get "hP[c]" = { "hP[c]" }

    test [store avalue] get "foo" = { = "foo" }
    test [store avalue] get "" = { = "" }

    test setting get
        "hP[c]=H py i ht:p ft, e.g. sqd.rt.c:3128\n" =
        { "hP[c]" = "H py i ht:p ft, e.g. sqd.rt.c:3128" }
    test setting get "foo=\n" = { "foo" = "" }

    test lns get
"# Automatically generated Red Hat Update Agent config file, do not edit.
# Format: 1.0
tmpDir[comment]=Use this Directory to place the temporary transport files
tmpDir=/tmp

disallowConfChanges[comment]=Config options that can not be overwritten by [WRAP]
a config update action
disallowConfChanges=noReboot;sslCACert;useNoSSLForPackages;noSSLServerURL;s[WRAP]
erverURL;disallowConfChanges;

skipNetwork[comment]=Skips network information in hardware profile sync dur[WRAP]
ing registration.
skipNetwork=0
```

```
networkRetries[comment]=Number of attempts to make at network connections b[WRAP]
efore giving up
networkRetries=1

hostedWhitelist[comment]=RHN Hosted URL's
hostedWhitelist=

enableProxy[comment]=Use a HTTP Proxy
enableProxy=0

writeChangesToLog[comment]=Log to /var/log/up2date which packages has been [WRAP]
added and removed
writeChangesToLog=0

serverURL[comment]=Remote server URL
serverURL=https://xmlrpc.rhn.redhat.com/XMLRPC

proxyPassword[comment]=The password to use for an authenticated proxy
proxyPassword=

networkSetup[comment]=None
networkSetup=1

proxyUser[comment]=The username for an authenticated proxy
proxyUser=

versionOverride[comment]=Override the automatically determined system versi[WRAP]
on
versionOverride=

sslCACert[comment]=The CA cert used to verify the ssl server
sslCACert=/usr/share/rhn/RHNS-CA-CERT

retrieveOnly[comment]=Retrieve packages only
retrieveOnly=0

debug[comment]=Whether or not debugging is enabled
debug=0

httpProxy[comment]=HTTP proxy in host:port format, e.g. squid.redhat.com:31[WRAP]
28
httpProxy=

systemIdPath[comment]=Location of system id
systemIdPath=/etc/sysconfig/rhn/systemid

enableProxyAuth[comment]=To use an authenticated proxy or not
enableProxyAuth=0

noReboot[comment]=Disable the reboot actions
noReboot=0
" = (
        { "#comment" = "Automatically generated Red Hat Update Agent config[WRAP]
 file, do not edit." }
        { "#comment" = "Format: 1.0" }
        { "tmpDir[comment]" = "Use this Directory to place the temporary tr[WRAP]
ansport files" }
```

```
        { "tmpDir" = "/tmp" }
        {  }
        { "disallowConfChanges[comment]" = "Config options that can not be [WRAP]
overwritten by a config update action" }
        { "disallowConfChanges" = "noReboot;sslCACert;useNoSSLForPackages;n[WRAP]
oSSLServerURL;serverURL;disallowConfChanges;" }
        {  }
        { "skipNetwork[comment]" = "Skips network information in hardware p[WRAP]
rofile sync during registration." }
        { "skipNetwork" = "0" }
        {  }
        { "networkRetries[comment]" = "Number of attempts to make at networ[WRAP]
k connections before giving up" }
        { "networkRetries" = "1" }
        {  }
        { "hostedWhitelist[comment]" = "RHN Hosted URL’s" }
        { "hostedWhitelist" = "" }
        {  }
        { "enableProxy[comment]" = "Use a HTTP Proxy" }
        { "enableProxy" = "0" }
        {  }
        { "writeChangesToLog[comment]" = "Log to /var/log/up2date which pac[WRAP]
kages has been added and removed" }
        { "writeChangesToLog" = "0" }
        {  }
        { "serverURL[comment]" = "Remote server URL" }
        { "serverURL" = "https://xmlrpc.rhn.redhat.com/XMLRPC" }
        {  }
        { "proxyPassword[comment]" = "The password to use for an authentica[WRAP]
ted proxy" }
        { "proxyPassword" = "" }
        {  }
        { "networkSetup[comment]" = "None" }
        { "networkSetup" = "1" }
        {  }
        { "proxyUser[comment]" = "The username for an authenticated proxy" [WRAP]
}
        { "proxyUser" = "" }
        {  }
        { "versionOverride[comment]" = "Override the automatically determin[WRAP]
ed system version" }
        { "versionOverride" = "" }
        {  }
        { "sslCACert[comment]" = "The CA cert used to verify the ssl server[WRAP]
" }
        { "sslCACert" = "/usr/share/rhn/RHNS-CA-CERT" }
        {  }
        { "retrieveOnly[comment]" = "Retrieve packages only" }
        { "retrieveOnly" = "0" }
        {  }
        { "debug[comment]" = "Whether or not debugging is enabled" }
        { "debug" = "0" }
        {  }
        { "httpProxy[comment]" = "HTTP proxy in host:port format, e.g. squi[WRAP]
d.redhat.com:3128" }
        { "httpProxy" = "" }
        {  }
        { "systemIdPath[comment]" = "Location of system id" }
        { "systemIdPath" = "/etc/sysconfig/rhn/systemid" }
        {  }
        { "enableProxyAuth[comment]" = "To use an authenticated proxy or no[WRAP]
t" }
        { "enableProxyAuth" = "0" }
        {  }
        { "noReboot[comment]" = "Disable the reboot actions" }
        { "noReboot" = "0" }
    )
```

### 12.4.100 1.2.0/tests/test_upstartinit.aug

```
module Test_upstartinit =
    let lns = Upstartinit.lns
    let script_line = Upstartinit.script_line
    let script = Upstartinit.script
    let lifecycle = Upstartinit.lifecycle
    let respawn = Upstartinit.respawn

    test lns get "\n" = {}
    test lns get "# bla\n" = { "#comment" = "bla" }
    test script_line get "end script\n" = *
    test script_line get "foo\n" = { "1" = "foo" }
    test script get "script\nend script\n" =  { "script" }
    test script get "script\nfoo\nend script\n" =  { "script" { "1" = "foo"[WRAP]
 } }
    test script get "script\n\nend script\n" = { "script" { "1" } }
    test script get "script\n\tfoo\nend script\n" = { "script" { "1" = "\tf[WRAP]
oo" } }
    test lns get "script\nfoo\nbar\nend script\n" =
        { "script"
            { "1" = "foo" }
            { "2" = "bar" }
        }
    test lifecycle get "post-stop exec hi\n" =
        { "post-stop"
            { "exec" = "hi" }
        }
    test lns get "post-stop exec hi\n" =
        { "post-stop"
            { "exec" = "hi" }
        }
    test lns get "exec foo bar baz\n" = { "exec" = "foo bar baz" }

    test respawn get "respawn\n" = { "respawn" }
    test respawn get "respawn foo bar baz\n" = { "respawn" = "foo bar baz" [WRAP]
}

    test lns get "# tty - getty
#
# This service maintains a getty on the specified device.

stop on runlevel [S016]

respawn
```

```
instance $TTY
exec /sbin/mingetty $TTY
usage 'tty TTY=/dev/ttyX  - where X is console id'
" = (
  { "#comment" = "tty - getty" }
  {  }
  { "#comment" = "This service maintains a getty on the specified device." [WRAP]
}
  {  }
  { "stop" = "on runlevel [S016]" }
  {  }
  { "respawn" }
  { "instance" = "$TTY" }
  { "exec" = "/sbin/mingetty $TTY" }
  { "usage" = "'tty TTY=/dev/ttyX  - where X is console id'" }
)

(*
    test lns get "
# On machines where kexec isn't going to be used, free the memory reserved [WRAP]
for it.

start on stopped rcS
task

script
if [ ! -x /sbin/kexec ] || ! chkconfig kdump 2>/dev/null ; then
echo -n \"0\" > /sys/kernel/kexec_crash_size 2>/dev/null
fi
exit 0
end script
" =
(
  {  }
  { "#comment" = "On machines where kexec isn't going to be used, free the [WRAP]
memory reserved for it." }
  {  }
  { "start" = "on stopped rcS" }
  { "task" }
  {  }
  { "script"
    { "1" = "   if [ ! -x /sbin/kexec ] || ! chkconfig kdump 2>/dev/null ; [WRAP]
then" }
    { "2" = "           echo -n \"0\" > /sys/kernel/kexec_crash_size 2>/dev[WRAP]
/null" }
    { "3" = "   fi" }
    { "4" = "   exit 0" }
  }
)

*)
```

## 12.5 dod_login_warnings/

For the policy that requires files in this section, see 11.29.1.

### 12.5.1 80col

```
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::[WRAP]
::::
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 defence, personnel misconduct (PM), la[WRAP]
w
  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 sub[WRAP]
ject
  to routine monitoring, interception, and search, and may be disclosed or [WRAP]
used
  for any USG-authorized purpose.
- This IS includes security measures (e.g., authentication and access contr[WRAP]
ols)
  to protect USG interests--not for your personal benefit or privacy.
- Notwithstanding the above, using this IS does not constitute consent to P[WRAP]
M,
  LE or CI investigative searching or monitoring of the content of privileg[WRAP]
ed
  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 Us[WRAP]
er
  Agreement for details.
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::[WRAP]
::::
```

### 12.5.2 paragraphs

```
You are accessing a U.S. Government (USG) information system (IS) that is p[WRAP]
rovided for USG-authorized use only. By using this IS (which includes any d[WRAP]
evice attached to this IS), you consent to the following conditions:

- The USG routinely intercepts and monitors communications on this IS for p[WRAP]
urposes including, but not limited to, penetration testing, COMSEC monitori[WRAP]
ng, network operations and defence, personnel misconduct (PM), law enforcem[WRAP]
ent (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 sub[WRAP]
ject to routine monitoring, interception, and search, and may be disclosed [WRAP]
or used for any USG-authorized purpose.
```

```
- This IS includes security measures (e.g., authentication and access contr[WRAP]
ols) to protect USG interests--not for your personal benefit or privacy.

- Notwithstanding the above, using this IS does not constitute consent to P[WRAP]
M, LE or CI investigative searching or monitoring of the content of privile[WRAP]
ged communications, or work product, related to personal representation or [WRAP]
services by attorneys, psychotherapists, or clergy, and their assistants.  [WRAP]
Such communications and work product are private and confidential. See User[WRAP]
 Agreement for details.
```

### 12.5.3 paragraphs.rtf

```
{\rtf1\ansi\ansicpg1252\cocoartf1265\cocoasubrtf210
{\fonttbl\f0\fswiss\fcharset0 Helvetica;}
{\colortbl;\red255\green255\blue255;}
\margl1440\margr1440\vieww10800\viewh8400\viewkind0
\pard\tx720\tx1440\tx2160\tx2880\tx3600\tx4320\tx5040\tx5760\tx6480\tx7200\[WRAP]
tx7920\tx8640\pardirnatural

\f0\fs24 \cf0 You are accessing a U.S. Government (USG) information system [WRAP]
(IS) that is provided for USG-authorized use only. By using this IS (which [WRAP]
includes any device attached to this IS), you consent to the following cond[WRAP]
itions:\
\
- The USG routinely intercepts and monitors communications on this IS for p[WRAP]
urposes including, but not limited to, penetration testing, COMSEC monitori[WRAP]
ng, network operations and defence, personnel misconduct (PM), law enforcem[WRAP]
ent (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 sub[WRAP]
ject to routine monitoring, interception, and search, and may be disclosed [WRAP]
or used for any USG-authorized purpose.\
\
- This IS includes security measures (e.g., authentication and access contr[WRAP]
ols) to protect USG interests--not for your personal benefit or privacy.\
\
- Notwithstanding the above, using this IS does not constitute consent to P[WRAP]
M, LE or CI investigative searching or monitoring of the content of privile[WRAP]
ged communications, or work product, related to personal representation or [WRAP]
services by attorneys, psychotherapists, or clergy, and their assistants.  [WRAP]
Such communications and work product are private and confidential. See User[WRAP]
 Agreement for details.\
}
```

## 12.6 gdm/

For the policy that requires files in this section, see 11.35.1.

### 12.6.1 logo/afseo/logo-48x48.png

The file `gdm/logo/afseo/logo-48x48.png` appears not to be human-readable. It is not included here.

### 12.6.2 logo/afseo/logo-scalable.png

The file `gdm/logo/afseo/logo-scalable.png` appears not to be human-readable. It is not included here.

## 12.7 gluster/

For the policy that requires files in this section, see 11.36.3.

### 12.7.1 Makefile

```
TEs = $(wildcard *.te)
PPs = $(addsuffix .pp,$(basename $(TEs)))

all: $(PPs)

# Puppet files end with .pp, and so do SELinux policy packages. The
# unified-policy-document has some magic in its Makefile that finds all *.p[WRAP]
p
# files, and we don’t want it to try to treat these as Puppet files, so ins[WRAP]
ide
# the policy we call them *.selinux.pp.

clean:
rm -f *.selinux.pp *.mod

%.pp: %.mod
semodule_package -m $< -o $@
mv $@ $(addsuffix .selinux.pp,$(basename $@))

%.mod: %.te
checkmodule -M -m $< -o $@
```

### 12.7.2 gluster_automount.selinux.pp

The file gluster/gluster_automount.selinux.pp appears not to be human-readable. It is not included here.

### 12.7.3 gluster_automount.te

```
module gluster_automount 1.0.0;

require {
    type mount_t;
    type automount_t;
    class fifo_file { open };
}

allow mount_t automount_t:fifo_file open;
```

## 12.8 hpc_cluster/

For the policy that requires files in this section, see 11.42.4.

### 12.8.1 gather.cron

```
#!/bin/sh

# gather all non-system users and write in /srv/passwd/passwd
getent passwd | (IFS='
'; while read line; do
    uid=$(echo "$line" | cut -d: -f3)
    if [ $uid -gt 1000 -a $uid -ne 65534 ]; then
        echo $line;
    fi; done) > /srv/passwd/passwd.new
mv /srv/passwd/passwd.new /srv/passwd/passwd

# same with groups
getent group | (IFS='
'; while read line; do
    gid=$(echo "$line" | cut -d: -f3)
    if [ $gid -gt 1000 -a $gid -ne 65534 ]; then
        echo $line
    fi; done) > /srv/passwd/group.new
mv /srv/passwd/group.new /srv/passwd/group
```

### 12.8.2 integrate.cron

```
#!/bin/sh

set -e

# gather all system users and write in new passwd file
getent passwd | (IFS='
'; while read line; do
    uid=$(echo "$line" | cut -d: -f3)
    if [ $uid -le 1000 -o $uid -eq 65534 ]; then
        echo $line;
    fi; done) > /etc/passwd.new

# now grab the non-system users
cat /net/passwd/passwd >> /etc/passwd.new

mv -f /etc/passwd.new /etc/passwd

# same with system groups
getent group | (IFS='
'; while read line; do
    gid=$(echo "$line" | cut -d: -f3)
    if [ $gid -le 1000 -o $gid -eq 65534 ]; then
        echo $line
    fi; done) > /etc/group.new

# non-system groups
cat /net/passwd/group >> /etc/group.new
```

```
mv -f /etc/group.new /etc/group
```

## 12.9 log/

For the policy that requires files in this section, see 11.55.1.

### 12.9.1 backup/to_net_admin.sh

```
#!/bin/sh

DESTDIR=/net/admin/BACKUPS/‘hostname -s‘/LOGS

# $TMPDIR must have enough space to hold all the repositories roughly twice[WRAP]
,
# and be writable by whoever is running this script.
TMPDIR=/tmp
NAME=system_logs-‘date +%Y-%m-%d--%H-%M-%S‘




set -e

TMP=‘mktemp -dt $ME.XXXXXXXXXXX‘
# Exclude lastlog: it is very large, though sparse, so it takes a long time[WRAP]
 to
# tar. Its data is in other log files as well, so we’re not losing any data[WRAP]
.
# files
tar -c -z --one-file-system -C /var --exclude log/lastlog -f $TMP/$NAME.tar[WRAP]
.gz log
mv $TMP/$NAME.tar.gz $DESTDIR
rmdir $TMP

/usr/sbin/logrotate -f /etc/logrotate.conf
```

### 12.9.2 rsyslog/Makefile

```
TEs = $(wildcard *.te)
PPs = $(addsuffix .pp,$(basename $(TEs)))

all: $(PPs)

# Puppet files end with .pp, and so do SELinux policy packages. The
# unified-policy-document has some magic in its Makefile that finds all *.p[WRAP]
p
# files, and we don’t want it to try to treat these as Puppet files, so ins[WRAP]
ide
# the policy we call them *.selinux.pp.

clean:
rm -f *.selinux.pp *.mod

%.pp: %.mod
semodule_package -m $< -o $@
mv $@ $(addsuffix .selinux.pp,$(basename $@))

%.mod: %.te
```

```
checkmodule -M -m $< -o $@
```

### 12.9.3 rsyslog/rsyslog_client.selinux.pp

The file `log/rsyslog/rsyslog_client.selinux.pp` appears not to be human-readable. It is not included here.

### 12.9.4 rsyslog/rsyslog_client.te

```
module rsyslog_client 1.0.13;

require {
type syslogd_t;
type port_t;
type var_spool_t;
        type random_device_t;
class capability ipc_lock;
class tcp_socket name_connect;
class dir search;
        class chr_file read;
}

# Allow syslogd to connect via TCP to the loghost.
allow syslogd_t port_t:tcp_socket name_connect;
allow syslogd_t self:capability ipc_lock;

# Let rsyslogd find /var/spool/rsyslog in /var/spool; the default context o[WRAP]
f
# /var/spool/rsyslog is var_log_t, so everything that needs to be done insi[WRAP]
de
# it is already allowed by the default policy.
allow syslogd_t var_spool_t:dir search;

allow syslogd_t random_device_t:chr_file read;
```

### 12.9.5 rsyslog/rsyslog_loghost.selinux.pp

The file `log/rsyslog/rsyslog_loghost.selinux.pp` appears not to be human-readable. It is not included here.

### 12.9.6 rsyslog/rsyslog_loghost.te

```
module rsyslog_loghost 1.0.1;

require {
        type syslogd_t;
        type port_t;
        type random_device_t;
        class capability ipc_lock;
        class tcp_socket name_bind;
        class chr_file read;
}
```

```
allow syslogd_t port_t:tcp_socket name_bind;
allow syslogd_t self:capability ipc_lock;
allow syslogd_t random_device_t:chr_file read;
```

## 12.10 nvidia/

For the policy that requires files in this section, see 11.71.

### 12.10.1 01-nvidia.conf

```
Section "Device"
Identifier "nvidia Device 0"
Driver "nvidia"
EndSection
Section "Screen"
Identifier "nvidia Screen 0"
Device "nvidia Device 0"
EndSection
Section "ServerLayout"
Identifier "nvidia Layout"
Screen "nvidia Screen 0"
EndSection
```

## 12.11 pki/

For the policy that requires files in this section, see 11.75.4.

### 12.11.1 all-ca-certs/ADO-CA014.crt

```
-----BEGIN CERTIFICATE-----
MIIGAzCCBOugAwIBAgIUXiEkzh4axp4wSaxxnWz5yCkDmeowDQYJKoZIhvcNAQEF
BQAwVzELMAkGA1UEBhMCQVUxDDAKBgNVBAoTA0dPVjEMMAoGA1UECxMDRG9EMQww
CgYDVQQLEwNQS0kxDDAKBgNVBAsTA0NBczEQMA4GA1UEAxMHQURPQ0EwMzAeFw0x
MzA1MjIwMDAyMThaFw0xNjA1MjIwMDAyMThaMFgxCzAJBgNVBAYTAkFVMQwwCgYD
VQQKEwNHT1YxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMQwwCgYDVQQLEwND
QXMxETAPBgNVBAMTCEFET0NBMDE0MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA1F0M9m5RpA6rMO9ooANnABmiADYsCslWSmWD0e4CIhf/S3ko/8Di99pu
pKPh6lb/b3wARyjRcD1yUpuz/UM2BUqeMufbXokKdMDdvKC7AY5cXd3VNnGXGeyP
webBU3b4GYotAuMIGKkQk/s2QNEXlkIt//yZlsABibuaGaqU9L/LGfDZMHuy2CA8
a33ax3Whv+FvmlhqT2Y3VYhjkxJhoOY85V/qvoPLC2aAtCZbRff6hgKsl3PL78Pd
yMhwodSLGkK/3mdEAYYXu3kktzVgSMBX63K1hXTih24X6Ev0TOCxhB8w/HZLyDuQ
GVQOO+xtjY+R43VizTEISL4H1VAlAQIDAQABo4ICxDCCAsAwDwYDVR0TAQH/BAUw
AwEB/zCB/AYIKwYBBQUHAQEEge8wgewwJgYIKwYBBQUHMAGGGmh0dHA6Ly9vY3Nw
LmRlZmVuY2UuZ292LmF1MD4GCCsGAQUFBzAChjJodHRwOi8vd3d3LmRlZmVuY2Uu
Z292LmF1L3BraS9jZXJ0aWZpY2F0ZXMvQURPQ0EwMzCBgQYIKwYBBQUHMAKGdWxk
YXA6Ly9kaXIuZGVmZW5jZS5nb3YuYXUvY249QURPQ0EwMyxvdT1DQXMsb3U9UEtJ
LG91PURvRCxvPUdPVixjPUFVP2NBQ2VydGlmaWNhdGU7YmluYXJ5LGNyb3NzQ2Vy
dGlmaWNhdGVQYWlyO2JpbmFyeTCBqAYDVR0gBIGgMIGdMDgGCSokAYJOAQEBAzAr
MCkGCCsGAQUFBwIBFh1odHRwOi8vd3d3LmRlZmVuY2UuZ292LmF1L3BraTALBgkq
JAGCTgECAQEwCwYJKiQBgk4BAgECMAsGCSokAYJOAQIBAzALBgkqJAGCTgECAQQw
CwYJKiQBgk4BAgIBMAsGCSokAYJOAQICAjALBgkqJAGCTgECAgMwBgYEVR0gADAO
BgNVHQ8BAf8EBAMCAcYwHwYDVR0jBBgwFoAUPhPsALSF7LPqMm4x3UOduHSXxuYw
gbIGA1UdHwSBqjCBpzAzoDGgL4YtaHR0cDovL3d3dy5kZWZlbmNlLmdvdi5hdS9w
a2kvY3JsL0FET0NBMDMuY3JsMHCgbqBshmpsZGFwOi8vZGlyLmRlZmVuY2UuZ292
LmF1L2NuJTNkQURPQ0EwMyxvdSUzZENBcyxvdSUzZFBLSSxvdSUzZERvRCxvJTNk
R09WLGMlM2RBVT9jZXJ0aWZpY2F0ZVJldm9jYXRpb25MaXN0MB0GA1UdDgQWBBSS
xCDIev26lftTSv2SKUMjj99GwTANBgkqhkiG9w0BAQUFAAOCAQEAE0UlBXhqmtnm
s85mYUeUtxtTd9rJWqdFogWVjPeNWwRO++r3oG7Cyi1nE3BEcKrAEGdqHkj3gvA7
sqOP7k1+atlHJ7g6vGddaHz2tHVjyswttozxsiFGqQHEHE24R+7rZpevYNFObvFC
1gA9XWwtWzRY73LTU94A03wAKD2BgaqiS+qGh9Ms3unWhg0eNnG0Jpav/WFTcU3V
3HCAlGqzaqZADxs7xKDzkBLtzLFsFHYZJdEWJaGuxjOjZHMGJAostG0NSUzpMtle
DQKZSroeUdI8bRFXC+f95tmPkQA8RMC5KJRmg3ppvvBSAad4pIp3vRM2FRDcyaoH
b+QyIIwLIw==
-----END CERTIFICATE-----
```

### 12.11.2 all-ca-certs/ADO-CA016.crt

```
-----BEGIN CERTIFICATE-----
MIIGAzCCBOugAwIBAgIUZchd8io+kjQ0brohyLcYdwYwqtEwDQYJKoZIhvcNAQEF
BQAwVzELMAkGA1UEBhMCQVUxDDAKBgNVBAoTA0dPVjEMMAoGA1UECxMDRG9EMQww
CgYDVQQLEwNQS0kxDDAKBgNVBAsTA0NBczEQMA4GA1UEAxMHQURPQ0EwMzAeFw0x
MzEyMTcyMzQ4MDlaFw0xNjEyMTcyMzQ4MDlaMFgxCzAJBgNVBAYTAkFVMQwwCgYD
VQQKEwNHT1YxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMQwwCgYDVQQLEwND
QXMxETAPBgNVBAMTCEFET0NBMDE2MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAvJTO/bOM0TAS6hXOv6Fs+tBeZNatTA8z9zWCQHhyUie7xD4cJCM69X38
1u4Flt+eTX1FAwFxIi3boqHWRnrt8cxNrHICDs8kmNKqxaRsWrvd5py07Y60lhZ1
QOXGx9PWyXo346CuE8uKuowMf7bUVo+8drI6iGmOzh4rlFaitmKk8J/cAs+qMsct
ZqT4pl0a72AFIdTcjlDvdZj0iGmjFWm7kx1HO8sIWGLvTHnMbttUC2uKkCvKaQ8C
BplSzkd2PQJfB9bu+eBqeCA5fBOi29NWpNcAbEvzHLWT+hyie/Hmlnghm4UBGJXu
```

```
DBsqx5CJ9qWELYC95EsffCF94l/nfwIDAQABo4ICxDCCAsAwDwYDVR0TAQH/BAUw
AwEB/zCB/AYIKwYBBQUHAQEEge8wgewwJgYIKwYBBQUHMAGGGmh0dHA6Ly9vY3Nw
LmRlZmVuY2UuZ292LmF1MD4GCCsGAQUFBzAChjJodHRwOi8vd3d3LmRlZmVuY2Uu
Z292LmF1L3BraS9jZXJ0aWZpY2F0ZXMvQURPQ0EwMzCBgQYIKwYBBQUHMAKGdWxk
YXA6Ly9kaXIuZGVmZW5jZS5nb3YuYXUvY249QURPQ0EwMyxvdT1DQXMsb3U9UEtJ
LG91PURvRCxvPUdPVixjPUFVP2NBQ2VydGlmaWNhdGU7YmluYXJ5LGNyb3NzQ2Vy
dGlmaWNhdGVQYWlyO2JpbmFyeTCBqAYDVR0gBIGgMIGdMDgGCSokAYJOAQEBAzAr
MCkGCCsGAQUFBwIBFh1odHRwOi8vd3d3LmRlZmVuY2UuZ292LmF1L3BraTALBgkq
JAGCTgECAQEwCwYJKiQBgk4BAgECMAsGCSokAYJOAQIBAzALBgkqJAGCTgECAQQw
CwYJKiQBgk4BAgIBMAsGCSokAYJOAQICAjALBgkqJAGCTgECAgMwBgYEVR0gADAO
BgNVHQ8BAf8EBAMCAcYwHwYDVR0jBBgwFoAUPhPsALSF7LPqMm4x3UOduHSXxuYw
gbIGA1UdHwSBqjCBpzAzoDGgL4YtaHR0cDovL3d3dy5kZWZlbmNlLmdvdi5hdS9w
a2kvY3JsL0FET0NBMDMuY3JsMHCgbqBshmpsZGFwOi8vZGlyLmRlZmVuY2UuZ292
LmF1L2NuJTNkQURPQ0EwMyxvdSUzZENBcyxvdSUzZFBLSSxvdSUzZERvRCxvJTNk
R09WLGMlM2RBVT9jZXJ0aWZpY2F0ZVJldm9jYXRpb25MaXN0MB0GA1UdDgQWBBTe
gSo+nKNoUct/y6/PhEbZAVeJCzANBgkqhkiG9w0BAQUFAAOCAQEAHFqc9wmTxib8
U1SS/G13eBksCNV4LPFUQxc7Uu9Wq7vaHl40yA6qqqm9WdcrRbxa1uUkIiMnZ9iK
O6HG/srAzdYYrezjgNkEdgx97H+IjR/iuT7n2jNuxIDS4zkPOhpgvU+A/7PN3Xnq
9bxK7XBUsRiTKoBvJNEDVQA2vhkOxkWBaxMeOaNBOM282QAYI94BQc0TzithPfcx
MmKRdeJzHHd63Xt7YDmJMnZgNa5vWMCF/s8zMzIzAuGeZDfVhuI3XfhBCyQKL+w9
gGIInItQ4gLhMj2PQTr8CyMGvbSIRYw7Bxz/NUcrXW5Ll324X20IkEAyjG6BMWza
hxNBCn8Atg==
-----END CERTIFICATE-----
```

### 12.11.3 all-ca-certs/Bridge-DoDCCEBIRCA1-ADOCA03.crt

```
-----BEGIN CERTIFICATE-----
MIIGATCCBOmgAwIBAgICASwwDQYJKoZIhvcNAQEFBQAwdDELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxLzAtBgNVBAMTJlVTIERvRCBDQ0VCIEludGVyb3BlcmFiaWxpdHkgUm9v
dCBDQSAxMB4XDTE0MDQxNjEyNTE0MVoXDTE3MDQxNjEyNTE0MVowVzELMAkGA1UE
BhMCQVUxDDAKBgNVBAoTA0dPVjEMMAoGA1UECxMDRG9EMQwwCgYDVQQLEwNQS0kx
DDAKBgNVBAsTA0NBczEQMA4GA1UEAxMHQURPQ0EwMzCCASIwDQYJKoZIhvcNAQEB
BQADggEPADCCAQoCggEBALfbxHp2mM5WWW8Jofrl/tQHC04mP8D3BVg3ypMdO/58
PA55pwJgOAnpXfJHF4D3PaiNSsVitL1TAvTjBVNUVzP4NKLjPzZM0SlEArv+Av+f
W8VIaFUJB6TM05yy+0ViCN3j+VLyZbAIYLJD391ltTyXkSLUptV8GaK++8kF/5U2
u4RKe1od4n7PeMKu0lbNi0s1OG+f4vVS4x6YbSe7tREdOHSvcceJ+1W+FXXrQL8C
glFaRxo0jjcra85auddX1OdoQ0HwfVzPknE5u0otsG07kKW18v1KgQc49pIRnzm8
wrieXyYHhkr9TpCSMskuCCL3SWLoHc2c0wPKbvFwdWkCAwEAAaOCArgwggK0MB8G
A1UdIwQYMBaAFJcYX6t7sKzkg7lQY9Vhi0yAAU0hMB0GA1UdDgQWBBQ+E+wAtIXs
s+oybjHdQ524dJfG5jAOBgNVHQ8BAf8EBAMCAQYwMAYDVR0gBCkwJzALBglghkgB
ZQIBCwUwCwYJYIZIAWUCAQsRMAsGCWCGSAFlAgELEzBRBgNVHSEESjBIMBYGCWCG
SAFlAgELBQYJKiQBgk4BAgIBMBYGCWCGSAFlAgELEQYJKiQBgk4BAgICMBYGCWCG
SAFlAgELEwYJKiQBgk4BAgECMBIGA1UdEwEB/wQIMAYBAf8CAQIwcQYDVR0eAQH/
BGcwZaBjMC2kKzApMQswCQYDVQQGEwJBVTEMMAoGA1UEChMDR09WMQwwCgYDVQQL
EwNEb0QwCIIGZ292LmF1MAiBBmdvdi5hdTAJgQcuZ292LmF1MAiGBmdvdi5hdTAJ
hgcuZ292LmF1MBIGA1UdJAEB/wQIMAaAAQCBAQAwggECBgNVHR8EgfowgfcwQqBA
oD6GPGh0dHA6Ly9jcmwuZGlzYS5taWwvY3JsL1VTRE9EQ0NFQklOVEVST1BFUkFC
SUxJVFlST09UQ0ExLmNybDCBsKCBraCBqoaBp2xkYXA6Ly9jcmwuZ2RzLmRpc2Eu
bWlsL2NuJTNkVVMlMjBEb0QlMjBDQ0VCJTIwSW50ZXJvcGVyYWJpbGl0eSUyMFJv
b3QlMjBDQSUyMDElMmNvdSUzZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBH
b3Zlcm5tZW50JTJjYyUzZFVTP2NlcnRpZmljYXRlUmV2b2NhdGlvbkxpc3Q7Ymlu
YXJ5MDAGCCsGAQUFBwEBBCQwIjAgBggrBgEFBQcwAYYUaHR0cDovL29jc3AuZGlz
YS5taWwwCgYDVR02BAMCAQAwDQYJKoZIhvcNAQEFBQADggEBAAXf7MApz4fNfMS/
eI9YsosZOLKSXm1LDPfNDs4D0RoJhZnhWuoArFzAffJn9Lyl9Qgoi1rSCscQMN9w
rGQMmn8nJuNhkk5h65wTMldZ0lGc3MOK1IH6XC2HvytF3moXx6GPWpt13f+e8Mnb
rBX0Xq/yt4aT93THoIpM4zRD6HvFtaeTanzVX2ZAETIBtVTEJNm2MUYWgEq3rXDS
```

```
Ecp0+7ghb8aZ6tqo9kckhGiJCM2RvlZPZZjHbHCb72dEqmsQJsgYyjwyvs0QhB9S
VJkCJPoKBNXWFLEbYmQBZi3UbjobgNPBps/tUbQHjDhnV/IwgvnI3Zs3leH+VCGF
P7/TS74=
-----END CERTIFICATE-----
```

## 12.11.4 all-ca-certs/DoD-CCEB-Interop-Root-CA1.crt

```
-----BEGIN CERTIFICATE-----
MIIEDTCCAvWgAwIBAgIBATANBgkqhkiG9w0BAQUFADB0MQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEvMC0GA1UEAxMmVVMgRG9EIENDRUIgSW50ZXJvcGVyYWJpbGl0eSBSb290
IENBIDEwHhcNMTAxMTI5MTc0NzIzWhcNMzAxMTI0MTc0NzIzWjB0MQswCQYDVQQG
EwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAK
BgNVBAsTA1BLSTEvMC0GA1UEAxMmVVMgRG9EIENDRUIgSW50ZXJvcGVyYWJpbGl0
eSBSb290IENBIDEwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC66dnT
Jigsr98CMQ2SgN2g2fzN8yWbVm6Ir0om0BtSQxIKqYioy3+EHiPT7iKvYQVjfxA2
iacLScFjdVSU50SymDuEWhqkCBMb1JC445mdZBcIs7nfy2LQxEH2VxZjnKceNVzw
Py/zqYbfzGoT4Z7XyA1x2wAZEeZSsCfebyal0a75eeyLi5uvQyifUx2ocAXkl0K+
kPy8Tvfp/Y6vAfIFCz5BCht0WwMn+yFY3+DTvpheR+NEG2KWRB2a9UQHbnrpG0rb
wQ/Q0IV6ojjp4vomq0h5BYFm2NYErxNKze70hnDnn5TZh7SJuciLUBNkm/YxpJ0M
81bYsybITBECbwi7AgMBAAGjgakwgaYwDwYDVR0TAQH/BAUwAwEB/zA0BgNVHQ8B
Af8EBAMCAYYwHQYDVR0OBBYEFJcYX6t7sKzkg7lQY9Vhi0yAAU0hMGQGCCsGAQUF
BwELBFgwVjBUBggrBgEFBQcwBYZIaHR0cDovL2NybC5nZHMuZGlzYS5taWwvaXNz
dWVkYnkvVVNET0RDQ0VCSU5URVJPUEVSQUJJTElUWVJPT1RDQTFfSUIucDdjMA0G
CSqGSIb3DQEBBQUAA4IBAQBVpBkKFFuwAxYZot8ob0kGkyYS2EL7Yz8piAwdS/Bc
2AQJvkmxU7gj50A8M09qw2NRTUS0x/uoySyjLq+iT9pSAQxACHs0oliQQ0q9kTxT
CjXamk0zkmlP45GC8m0UXv0EMVmy62cB0ieg0fXbPrm54rMIuF9w1qf0dVSvcZlX
wzFyLYETJzSyoH6i0p9+bN7edf1XhjG6CLAAVNrxk0grWxRQTNBfNbkTxB11kNkQ
nG5eIXdWpC396R88/QMEuKengZjM8R7rXD82pGk3B5pM2ihRYhRWq2xrNp+GF40y
DPIDkUlPwsjm1xQ9J61BDM+NS0rCWKvQu3lVWaMnxfGv
-----END CERTIFICATE-----
```

## 12.11.5 all-ca-certs/DoD-Class3-Root.crt

```
-----BEGIN CERTIFICATE-----
MIICZzCCAdCgAwIBAgIBBDANBgkqhkiG9w0BAQUFADBhMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEcMBoGA1UEAxMTRG9EIENMQVNTIDMgUm9vdCBDQTAeFw0wMDA1MTkxMzEz
MDBaFw0yMDA1MTQxMzEzMDBaMGExCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMu
IEdvdmVybm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRwwGgYDVQQD
ExNEb0QgQ0xBU1MgMyBSb290IENBMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQC1MP5kvurMbe2BLPd/6Rm6DmlqK0GpqcuVWB/x5pppU+CIP5HFUbljl6jmIYwT
XjY8qFf6+HAsTGrLvzCnTBbkMlz4ErBR+BZXjS+0TfouqJToKmHUVw1Hzm4sL36Y
Z8wACKu2lhY1woWR5VugCsdmUmLzYXWVF668KlYppeArUwIDAQABoy8wLTAdBgNV
HQ4EFgQUbJyl8FyPbUGNxBc7kFfCD6PNbf4wDAYDVR0TBAUwAwEB/zANBgkqhkiG
9w0BAQUFAAOBgQCvcUT5lyPMaGmMQwdBuoggsyIAQciYoFUczT9usZNcrfoYmrsc
c2/9JEKPh59Rz76Gn+nXikhPCNlplKw/5g8tlw8ok3ZPYt//oM1h+KaGDDE0INx/
L6j70b6V7jhZAmLB3mwVT+DfnbvkeXMk/WNklfdKqJkfSGWVx3u/eDLneg==
-----END CERTIFICATE-----
```

## 12.11.6 all-ca-certs/DoD-Interop-Root-CA1.crt

```
-----BEGIN CERTIFICATE-----
MIIEqTCCA5GgAwIBAgIBBjANBgkqhkiG9w0BAQUFADBsMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEnMCUGA1UEAxMeRG9EIEludGVyb3BlcmFiaWxpdHkgUm9vdCBDQSAxMB4X
```

```
DTA3MDYyMDE0NDkxMVoXDTI3MDYxNTE0NDkxMVowbDELMAkGA1UEBhMCVVMxGDAW
BgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQLEwNQ
S0kxJzAlBgNVBAMTHkRvRCBJbnRlcm9wZXJhYmlsaXR5IFJvb3QgQ0EgMTCCASIw
DQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAJx9svbNWhpbtzD1Mi8LnprDHKhC
MmtsWVdwQjxKqamx3ULgWb54lsD/e7kjv15JMvEbaXPrxnJH138pv23Wk+e0YeSk
fBzwVs3Qs5kkxMF91e7Sg41oaIyOmOVdKTjIprPn7SErWLqDW7z0A5A/YO4rL5bU
LjePqyy0G6KADTWJR2NERsSqRUN7wTzSIU0KAnYTbXtXGb+nOHmbX/pv3/phjAfZ
TfIm7/KQNR/X/XRykCk28W50xNz3tRvXhXghryDJeqV9DnThjyRsQ9MJ9IOlORIu
T9SU+rNfB9sXgyeeB6bKB96vShPuNy43rJHoLWyCUr0CCK4fCbwKI6BZ/N0CAwEA
AaOCAVQwggFQMB0GA1UdDgQWBBR2hh7f7QDJfhQxfFuUgiFJV75wBzALBgNVHQ8E
BAMCAYYwDwYDVR0TAQH/BAUwAwEB/zCCAQ8GCCsGAQUFBwELBIIBATCB/jBWBggr
BgEFBQcwBYZKaHR0cDovL2NybC5nZHMuZGlzYS5taWwvZ2V0SXNzdWVkQnk/RG9E
JTIwSW50ZXJvcGVyYWJpbGl0eSUyMFJvb3QlMjBDQSUyMDEwgaMGCCsGAQUFBzAF
hoGWbGRhcDovL2NybC5nZHMuZGlzYS5taWwvY24lM2REb0QlMjBJbnRlcm9wZXJh
YmlsaXR5JTIwUm9vdCUyMENBJTIwMSUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNv
JTNkVS5TLiUyMEdvdmVybm1lbnQlMmNjJTNkVVM/Y3Jvc3NDZXJ0aWZpY2F0ZVBh
aXI7YmluYXJ5MA0GCSqGSIb3DQEBBQUAA4IBAQAMvkJLzrSRTFEn1P/yger36tI/
QFjkIOKKNMS5ZBUBCxlPnNfFOR1V6cks2usdcgXf2fN1FDnzLNcNpZ/9lw2oRJ7/
Hvmloxm6oKzHeUfGexla/KHT02qr4aDEzC10bxN/q3WiQS8L7abONpkpJK1xJAkf
IOYGqwgisEeiJqjSXBuqF/SLqzzzUoYqwogQnFFCFMnyyFyeSaAaSDtsSu2mJBYO
cGfDHk57VEWkNXu+fo0lWyunECw7HpFv+D2YkulL5zGMJOLmMqeMDldVRZDv8bRd
BEuS6coMK0ffefUzsAmGe/vxfBozWguJ1RZ2hqhmNQXmENg0tttzJZe7cUqs
-----END CERTIFICATE-----
```

### 12.11.7 all-ca-certs/DoD-Root2-CA21.crt

```
-----BEGIN CERTIFICATE-----
MIIFiTCCBHGgAwIBAgIBTDANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0wOTAxMjYxNjM1MDNaFw0x
NTAxMjUxNjM1MDNaMFcxCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRIwEAYDVQQDEwlET0Qg
Q0EtMjEwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDdlElknbLr9TVZ
5hTjI5zGC1inX0nBxgikNyl7IxR5CP4aLtpxFGKAL2NSlnuEl/bASHmxo0kIh9Ov
t49pTRAi4v5wXTyTCpxYXm8qXYH+HWI5LruZDgNan8bldy2IDWDMtIp3TF+b5qU/
pq8E6cxSnqyAZIOlaRXzVE30qAI6c5wWxEKFK0E3CUDEWCNPp0snxwdD5TgsDH/Y
A5WCCX+2mWhWhogD4dJUKnUXS2XK8xJFy5YQ7BPMG76bBFT7PFGbNH53jn35MbOO
n3zoHjfLUk6IPecJvVgjAJbyvKcDtDXmDHZvaCMicq2Lt/f/Ju0tHrVZQA2o/a0n
H1Hkue1BAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0jBBgwFoAU
SXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFAmZE+Kj1ed02PY/tdz71LUW
7UzTMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYDVR0gBIGX
MIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsKMAsGCWCG
SAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFlAwIBAwYw
DAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0wDAYKYIZI
AWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2EubWlsL2dl
dGNybD9Eb0QlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB7jA/Bggr
BgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9Eb0QlMjBS
b290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCB
iAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNkRG9EJTIw
Um9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNkVS5TLiUy
MEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkwDQYJKoZI
hvcNAQEFBQADggEBACXuf0uCpdBRmSoj3P0tJyXAaXlIADIm0u5sHBy78MAM09gs
dFilVQlDolr5J/7YWujgqKS9vQWlC5UmHA4IiA7k+R97fphBDD0gjkTC8azehAGG
7DXs/4G7YH2Ot1byTJACH9OIPOkhbowrvG8bQBlisuMUcL/RgEukcT8U7uD06R71
BYESPdT8AIOyH8IFLGMgCcJHnVsek3emIwsWY3Ba5M3eJSbcrVcIMSNmm5+cCRpU
/IlYa4P632JwHHr5MjX7w+jPBmrS2Tm6PY+uYHsqZgA5xVCpXkNNobwKsiT7EjZX
zfjKO19+y8URKtUEBftfWOdUB2epSQeOSlYTZks=
-----END CERTIFICATE-----
```

### 12.11.8 all-ca-certs/DoD-Root2-CA22.crt

```
-----BEGIN CERTIFICATE-----
MIIFiTCCBHGgAwIBAgIBSDANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0wOTAxMjYyMDE4NTlaFw0x
NTAxMjUyMDE4NTlaMFcxCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRIwEAYDVQQDEwlET0Qg
Q0EtMjIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCb/OGrH/FwNEUF
Xwn8HNfVJpPSkGmzHs7YElNwlEIM/KUuzn++aISDhCyPHeLfp9sF1SPzoYd41Cq+
MXVIwvcwa0sVJTyYC8cQLVXPKHazu0MgcqLDAWES3uquvdLklg567ZRhJPutmdri
ZhXN1bt374FPYS3PqatVGOhav4mNKc4gW0ATMVaSYEEGywqhM/5uS49bHV4pl+OB
9L3pBD3RMsagbcCThwEXQYcBwiMtsf6waQfIwp8TyoRt0f1yv76avWpgc1aIOsat
G8QXvQ0b41Jj/K/B+8wvbjXS3TrYENHEKLe2bP+T4PZy8CkTZws4PBkojWwZk0k9
Wz2XhNcdAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0jBBgwFoAU
SXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFCgwH1FRjtXdraHLIMJYFUYw
pkRPMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYDVR0gBIGX
MIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsKMAsGCWCG
SAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFlAwIBAwYw
DAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0wDAYKYIZI
AWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2EubWlsL2dl
dGNybD9Eb0QlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB7jA/Bggr
BgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9Eb0QlMjBS
b290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCB
iAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNkRG9EJTIw
Um9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNkVS5TLiUy
MEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkwDQYJKoZI
hvcNAQEFBQADggEBAKfeVjVjzvm0/tj/uSwN7p62qFbVQf0mfmf8spCNq9k45ndV
zTeoXrnXvGMkh5HOu5e9mOjlOFfO+w4zbbSUme+5QdilGBYB7v/mvOz4BtHUwWoA
9u24b97jC5hUG4ABnc2hR88OM88oibJJ+nuG/J7iyZaeOLEfJLPMFAWyYzhRazlo
Sb+ZgnNZE+HdRtIq87pkCVGflrq6ZrO44ZwT9IbkQQsoet2V2nU3sK/4Z77xrDxH
7GLw0zYJc0UX+L4qFpu8fodFHMPZyetLJ81GrVe2vsA1qBL6EUjbxNrx6ur0D0D8
bteeV3V3vKwMl+xSDr6nmLV4fnzWxZ89fCOn/yU=
-----END CERTIFICATE-----
```

### 12.11.9 all-ca-certs/DoD-Root2-CA23.crt

```
-----BEGIN CERTIFICATE-----
MIIFiTCCBHGgAwIBAgIBSzANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0wOTAxMjYxNjM4NDVaFw0x
NTAxMjUxNjM4NDVaMFcxCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRIwEAYDVQQDEwlET0Qg
Q0EtMjMwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDWp4YjHG0C2Jia
JH+l/1ujmJrrtdR/Hat6SUrtYZ/5yBZhuI/x/mlxLsVOYqUolgv6O1VmxkcB2Pcj
dzprs9+wNjLzXhRZ0eYf09wb0S8QJsWmFcGa9Bh7MYuXZOswxbACaTvaX4ex74r4
jv5fhur+hFquf6EXJrQCkVObfahVQk3+T+yOzZL14/OONJRSoMsUV3dloBX8SNEK
BpKJyu3rsnHHtyjgIJf9B1P7Ov88mrkcXKVPPllZo4tw151q8L371dL8n72Pp8jM
xKGlgSrKLpKQUMSIQ/OqlO5U7aayiFntw5EQlG0PZDTE2g7Nc1FgDYfGmRlLUZSt
ZQLvDY3FAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0jBBgwFoAU
SXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFKpB9xKjHIMNK9eKPD3F/GxS
T81YMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYDVR0gBIGX
MIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsKMAsGCWCG
SAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFlAwIBAwYw
```

```
DAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAwOwDAYKYIZI
AWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2EubWlsL2dl
dGNybD9EbOQlMjBSb29OJTIwQOElMjAyMIH+BggrBgEFBQcBAQSB8TCB7jA/Bggr
BgEFBQcwAoYzaHROcDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9EbOQlMjBS
b29OJTIwQOElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCB
iAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNkRG9EJTIw
Um9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REbOQlMmNvJTNkVS5TLiUy
MEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkwDQYJKoZI
hvcNAQEFBQADggEBAFBB2iPTjh4CXUh+1DFeZoCj8cv1sEq+6g9sYaRyRjlImVxD
6JNZOd+1GAFcMktQnD/UykP9YDJLlr2YXwxwndDcMy4+Te2VUq7iOJ5jf81sFHgA
dn9qcGye5KtYQgweLAdT3smkL42Ox71s3rOKgdtI48PirZRL38p5kzhpOKh8Nsxz
t9tPGRtHg+mLmjyqWw+H6x35qQPNpH5vpKOLGkp6rpbXsCZkmsl+8BcXuiRvjaeV
As79cvCZtR/OggZj9lDUc/rIez4kApCKTR+mQxVVWRUIeg7PhljqgRAvks65VL7Y
lBPxzmqBR7rAToQy1HEeheokiRWXbapNrysMjnk=
-----END CERTIFICATE-----
```

## 12.11.10 all-ca-certs/DoD-Root2-CA24.crt

```
-----BEGIN CERTIFICATE-----
MIIFiTCCBHGgAwIBAgIBRzANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEbOQxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQOEgMjAeFw0wOTAxMjYyMDIzMTFaFw0x
NTAxMjUyMDIzMTFaMFcxCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRIwEAYDVQQDEwlETOQg
QOEtMjQwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDCIK5JJXz7fDvS
Jt6L4UiWGj9ou3JeYNk27nSEPRY8/AfZ1w/lMLjtTBn4nBUKNWel+thm0yJR1G7B
5GBYAvH3e4dn6UENdAddCFcfWz1iqwQzNQxGOPqcuvo6v/lBwWfXsnpQ62e+5TYa
81E+fPz8//n/7dhKoG82PN8n7PL6FmFz7hxVVJdEbfbmVAdFS0ZrA+fMy0Yrch8T
JLVNv6bkZtX7Os0aMe9lLJIyyTM1bIxBBEHvNoO97zdN0YCd8tHizjlqPfpcScY0
a17h3eo9LmWpCTG68hJK2LbEMu4nBMpUso+TGLsmmQnsPHegCLjvlNGoxdraHBeA
dxWlBq4BAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0jBBgwFoAU
SXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFBZQF3XOO4qutQhFpKVw4PY3
tr5PMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYDVR0gBIGX
MIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsKMAsGCWCG
SAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFlAwIBAwYw
DAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAwOwDAYKYIZI
AWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2EubWlsL2dl
dGNybD9EbOQlMjBSb29OJTIwQOElMjAyMIH+BggrBgEFBQcBAQSB8TCB7jA/Bggr
BgEFBQcwAoYzaHROcDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9EbOQlMjBS
b29OJTIwQOElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCB
iAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNkRG9EJTIw
Um9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REbOQlMmNvJTNkVS5TLiUy
MEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkwDQYJKoZI
hvcNAQEFBQADggEBALHKt07LwFOLLAkRmxMxFfY9uS9iRnYqrEtV6wzXzihrC5Wr
CjgWy9euzIexVbomJZpVqTPZ44nqjlMHASDk4Ww8edZdWwHgajrMgVPVxhVOieTD
FqQFQoxn48Z890aeFD3MvGviZEtzYGuMX7ybYioVSD0MU56AOejEqhpwEmLGwu1q
eUMvpJpjGktkN8JRb8o6lh4/S3kgL4RfdDMU5c7v11UusJEe5KGXuzrb2VqhAHIZ
wuHypW/cdXVZQ/LW8MqZdLRtRSSxn4CQPNdvWKE1y8NIUz+jNl407SiuOE2Gfssx
tbJtjV4qqP+Sw2T3FJNId9ynV4C7+GR/1WyaJqY=
-----END CERTIFICATE-----
```

## 12.11.11 all-ca-certs/DoD-Root2-CA25.crt

```
-----BEGIN CERTIFICATE-----
MIIFiTCCBHGgAwIBAgIBTjANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEbOQxDDAKBgNVBAsT
```

```
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0xMDAxMTQxNzMzMTJaFw0x
NjAxMTQxNzMzMTJaMFcxCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRIwEAYDVQQDEwlET0Qg
Q0EtMjUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDh4mC3zlG3Zo5I
YPsLqLpaUdTYKmX0Opbto1iZoomoVYaFYa0I/7MSFnXSOPoc7pqgYlqR4czhyQ01
fhdffxSsAXXkX0pfLSLLSdSCsRkXrm0yOClylwubbhXQIwh7LUtEu6EVyZZuptkU
AoicXt/5gjEURqiLAT7krq76U1A3VLpkU2ihoo98gJf5O/KP5fL/RviK7FglgHdG
YGG6bmA+H3o8pNcXDlefoy63QIqAtuPX189tARPygJNH87lpmwtWffeLQKhwwk6N
BBl+izlUIw+7ivB8d9XphFbMBbdDcv7vYkIHhUhPROmC9BCWtLEjxegfe+qbkwv/
y+6EzDIHAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0jBBgwFoAU
SXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFC4LZfnWZd5LoyV1pKEuhSFA
c7kKMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYDVR0gBIGX
MIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsKMAsGCWCG
SAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFlAwIBAwYw
DAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0wDAYKYIZI
AWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2EubWlsL2dl
dGNybD9Eb0QlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB7jA/Bggr
BgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9EbOQlMjBS
b290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCB
iAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNkRG9EJTIw
Um9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNkVS5TLiUy
MEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkwDQYJKoZI
hvcNAQEFBQADggEBAJYLyBa7kmLoEo1gJeYSrHexk5alqlj4H4Az1cx+LSyxhE4r
1kUPDTi1OqaInVmu8M6lesX47p1546dzXmy7uKtkSLw7uloaXVVTmPRoVI41uCMH
tqR8dcUUoyKenxG2FjCRLNieoAKsouHHg0Hhwc1ihFg3kQNcOFgwHBFh0gFJhGrg
cQR0u5RwevnwzzsW6Xm1C6IFwnID5d9g0mRyswMGQBLR0wujC55CbbDrlUeaNkaC
JGVT1bwWCF8g7ldcAiTZx9QWvEuIGDrMDCojcX0IwX/2svETp+2CTuwL4ROuwjWB
QNUOntd5GN0+Zw9DsHbSqM56bXf6J8lYrbFp2hc=
-----END CERTIFICATE-----
```

## 12.11.12 all-ca-certs/DoD-Root2-CA26.crt

```
-----BEGIN CERTIFICATE-----
MIIFiTCCBHGgAwIBAgIBUDANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0xMDAxMTQxNzM4MDVaFw0x
NjAxMTQxNzM4MDVaMFcxCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRIwEAYDVQQDEwlET0Qg
Q0EtMjYwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDC5HG6/0QfpRHl
jlgaNX4EQnz7VOHOXiWj8APAq2wrPgCLH8qNRhMRF0OV6ZDm6Z3X09KN5pdWvFxo
rv8f6UwuRkEGto0NMexzQSIHd+5Evjtgs0KUZEfvJF/FurbcQzEEz8HaXyO9cJVc
P6ZYK14YrNGQ09atVhBbJ0DrkMJMfKsXZsIpliN1fwwLA0fnC/ko8pXTqW+dKE9i
6mnOjAZIf8ocKUQ1czZK6J571DfPmpM8U1TmHJO173lpdEQIak3vEtgvY6+Zy0U7
igl0FC/N+14mYGhhIIJXcRRvJTw9rw/aN5pt/KZFjL612+KUC9BHwrZUozKaafoi
N9TaOziZAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0jBBgwFoAU
SXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFGpfufR6NizidfC7ZDLB8bRM
pSz9MAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYDVR0gBIGX
MIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsKMAsGCWCG
SAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFlAwIBAwYw
DAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0wDAYKYIZI
AWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2EubWlsL2dl
dGNybD9EbOQlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB7jA/Bggr
BgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9EbOQlMjBS
b290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCB
iAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNkRG9EJTIw
Um9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNkVS5TLiUy
MEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkwDQYJKoZI
```

```
hvcNAQEFBQADggEBAHo+bKwGz/Juy/3tsGjSwpb04zw3EC1mlacdVmkdSiYppS9V
5j/TsDJFjRSh23WkbZj8bvXKftjKzlKkhQGYnRkiYFrKwi71IhMLGK1rxhzy2aaS
tPuQBxivQpsrUCrFLQPoBiyf9nkeiU0tOXYgX8iYqN4OYQosvgoEXjZ1z21rBe0q
XqMMcpDMmM4s+amXG8X838AspZA5rKCvY9xjhqrMHT/n22LaEgtjPENJ+AU5VS3G
gJZRAWRRXMsmeuq2qCmA4nfC6IwWcoV9b440pV9QvcNOjfV6fcjWYa7c+kgSVBId
SF6W8OX7qKF1YUxWgi2I1xi5CVW/sX5ZlMIsYJM=
-----END CERTIFICATE-----
```

### 12.11.13 all-ca-certs/DoD-Root2-CA27.crt

```
-----BEGIN CERTIFICATE-----
MIIFTDCCBDSgAwIBAgICAbIwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTEwOTA4MTU1MDI1WhcN
MTcwOTA4MTU1MDI1WjBXMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTESMBAGA1UEAxMJRE9E
IENBLTI3MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAloQI/Xq6tpSD
0J07GQvPBN+IKp64GljrhyIqYzp/OcNra+e8GqgRAvVhzQGkmHVzxheMiTxCx+KO
yYmxqP+fngq7aN663rYRAZRDdJy9z+G+4M4cbWp8fH9i6F7aNuqUxQaLRojiwMIk
CQVQf/PZ5RIFXtLXzjXCe0c1GBVXzcWc9+kxeiqMOfE1ji6hUJFAN6KOks6MVrf8
7C92PILewMi7R9Z+96koXCkelgCtJ4Z0hLQEuqdVFmkOk9S8jGJNT1aCtse0eC99
2dND7XMm8VPu/7PVsORutr4tG2gNd1iEV0LPQHCMvMrAOXVO9xmIfGvsKcZkHehT
UL2UquSD3QIDAQABo4ICHDCCAhgwDgYDVR0PAQH/BAQDAgGGMB8GA1UdIwQYMBaA
FEl0uwxeunr+AlTve6DGlcYJgHCWMB0GA1UdDgQWBBQbBARARV59K14LzJllTfOk
3pB3FzASBgNVHRMBAf8ECDAGAQH/AgEAMAwGA1UdJAQFMAOAAQAwZgYDVR0gBF8w
XTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETALBglghkgB
ZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUDAgEDGzA3
BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9ET0RST09U
Q0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5odHRwOi8v
Y3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAGCCsGAQUF
BzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNsZGFwOi8v
Y3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIlMmNvdSUz
ZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJjYyUzZFVT
P2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeeTANBgkqhkiG9w0BAQUFAAOCAQEA
QMAMS5+NI6Yx0TSunpFcX4TdnASVYc2AFB9u3oiXfOmvpfX3cCfhEESTCjGkCaaf
cEsikE6/Fv9iT9f0gjVCFwfutabLa4S3Gm4XHEUXTNHNz+XcDNfF9sa6LpyzO29c
Fl5DbCVKnPWw7/mqE866pj2yUNx0LLgUFXm95h2RHWgaPhW1B3dR0V8DrAGmOmbo
CtevM6EzlQht/IiSkvVy+i6PkzGkvykjoyFTuj6wdSDkFUA6WMjIHt3LdRC1QQHm
Flpo9zc51hG2MCX7tH7IEsbYn2Op2W2G0jZHoXMpIv4C9GMrKSCXrrU0vFjmYLYR
14Km5+I3fjt7Bngmb2xFcw==
-----END CERTIFICATE-----
```

### 12.11.14 all-ca-certs/DoD-Root2-CA28.crt

```
-----BEGIN CERTIFICATE-----
MIIFTDCCBDSgAwIBAgICAbMwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTEwOTA4MTU1NzAxWhcN
MTcwOTA4MTU1NzAxWjBXMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTESMBAGA1UEAxMJRE9E
IENBLTI4MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAq0tVmtGDxkU9
0AMvq/GYqzTDYcvkz/8zTQwW3ox7pCPGBMjT19KYLsJ8PXnf7PNFFE/wqRUz4dGm
PBaEmTwulHhndzESz6bBeBnTxMz8tvkcVU8flKfPshsnWW+n53gZyT54TVJI0iEY
5x0iaLv+eD21Ci+w7HqC6dhl/fDbPaTzXjj9Tes3+gIALFT/ebXLnjDu10E88T0+
9hIaNQRTTTQXcf9kuTgU1ndHVy23rM/hN1Ak7tHKP0TX6frS4EM1aY3mUJXf8Uhv
JFysVUfS43WzNVEKIrbyg4+icb5Aubr0S3pzevnzQK1f25gd5hN/39okUWvY0IvS
```

```
n4VhofbSPwIDAQABo4ICHDCCAhgwDgYDVR0PAQH/BAQDAgGGMB8GA1UdIwQYMBaA
FEl0uwxeunr+AlTve6DGlcYJgHCWMB0GA1UdDgQWBBQmtK6qLY7pjYpvtrVbnep0
rrGcaTASBgNVHRMBAf8ECDAGAQH/AgEAMAwGA1UdJAQFMAOAAQAwZgYDVR0gBF8w
XTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETALBglghkgB
ZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUDAgEDGzA3
BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9ET0RST09U
Q0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5odHRwOi8v
Y3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAGCCsGAQUF
BzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNsZGFwOi8v
Y3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIlMmNvdSUz
ZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJjYyUzZFVT
P2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUFAAOCAQEA
aLPRSRGrW929tB3qWt6nuM+H2qiQxSNJ0EknveoAK/x+HYvnm05esuPPg+GIg7FL
J2FE1aRb/wfFyELUdjxw4DghDI0Ty+8X0yPSH91NhDaMpodZBPvZLh5pnlZ0SLTG
UrEKfS4QXGWC/AyGYXCpxTpGYF9tvoGIZt+zxI6Zm7D8Pd7B2owbRCDUo3rAABBH
cIlEFWyLV+p+tY940BRzzD+VkexbrVNwTHn1gSGY0X6vOLE4h85w3iMjECX4GorR
RmqlTsZ39egCg+vcPzJZUiZsAGlkJZAVCZbj3mSxfKCcwNP+6+mMe6WMlLEBBGcz
BHlJWGqmQelYJ1aVmAP37g==
-----END CERTIFICATE-----
```

### 12.11.15 all-ca-certs/DoD-Root2-CA29.crt

```
-----BEGIN CERTIFICATE-----
MIIFTDCCBDSgAwIBAgICAbQwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTEwOTA4MTU1ODI2WhcN
MTcwOTA4MTU1ODI2WjBXMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTESMBAGA1UEAxMJRE9E
IENBLTI5MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAu0oQdhk0QB5y
K2GQMojwO3UcPj1qdzqXeQvs6FEPfmoAQ5jE5qHgRVstA/pmaKWj0OCSgT30d20D
29sl5nZglD/2X99aGEHi8zdI8Zf+Kq1E5/wx4xb4vJp1pf0vVpqCSTrNTU9wzT9/
ABEWgwquV31pBIOg83fKBcH/+4XfmDj0+4ATPTh3b84MmxhTvNj0JP88upuwK8fi
kNH9A/M478xSw37jemyhSBFo7gA4Tco4fDA9h43uICQGBF9cINEFxoC/CUnga5rE
Cy2rYXZeHHDZgsiLDjODGXAuKvF+6RlK+wgek83zV3e4F+VznzWjNF5ViHUnYNAV
pgXDFvZLpQIDAQABo4ICHDCCAhgwDgYDVR0PAQH/BAQDAgGGMB8GA1UdIwQYMBaA
FEl0uwxeunr+AlTve6DGlcYJgHCWMB0GA1UdDgQWBBSbxZQ/72FhV6jh/lmu4mkY
2xhX3jASBgNVHRMBAf8ECDAGAQH/AgEAMAwGA1UdJAQFMAOAAQAwZgYDVR0gBF8w
XTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETALBglghkgB
ZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUDAgEDGzA3
BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9ET0RST09U
Q0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5odHRwOi8v
Y3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAGCCsGAQUF
BzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNsZGFwOi8v
Y3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIlMmNvdSUz
ZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJjYyUzZFVT
P2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUFAAOCAQEA
LnpHz7S3uaNWil4y66XY1lTD4zpvwsWIFK24CPUFqE1XOLxF5IYPgxx7SIY+ggbz
iNtUKMk0hemfA7ytTA+eBlX23xMC3WWtCTgB6GIwq5neg1Fn7WirugBkPffaWT4N
E1/e3Cl738wjW5wCTWqZsDlxg6QrcSzTxoThFQhjF9gFTVQ+Ty3nLZojNghSaVtv
DnHvqJAvMT5zrvR60pnfJIQH83Fvx/g6elyaXBFa98g/Om9DtYg3ekGqGt8YFPCv
kP6iRWGy6ok3NKsNPOjA0AiJg1WlRwtLTDf1Kamgbx51qUa1cvxsniz6P96OlU09
u4gfB0hC0RC+AouNMDAIfA==
-----END CERTIFICATE-----
```

## 12.11.16 all-ca-certs/DoD-Root2-CA30.crt

```
-----BEGIN CERTIFICATE-----
MIIFTDCCBDSgAwIBAgICAbUwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTEwOTA4MTU1OTI0WhcN
MTcwOTA4MTU1OTI0WjBXMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTESMBAGA1UEAxMJRE9E
IENBLTMwMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzTyTZyYPHuaB
Xu6fzvlQHt1iohWEJeV3VsJTx6DgyUJTKuZOZ1I+cF3GaLgVZcjddtCy1ZrJqizj
xAkiBPd9iaSI2cKD7Fl7SRDvmo3Ihvlz3fIOYHqc2Y9Pd4N4DEtMLd7tn7GvHEMy
rLDQODpUniYPFEuNwW71JpUkN4ft7eDD1e/A8A119W+avv1kPCoirzgSK3MtDQl+
Eer8azJzTVzEWRfaxFmBBgS2CwLQZ7OWnHkTQxUkXsSV/VDRXgieH7ShlpI5K2is
vYw+hokuPrbrReC8HJsrC3jvbfEaYN3mR/h19PLKRKj7gFngUWOFC7b7Fizj8/9v
92q+m801gQIDAQABo4ICHDCCAhgwDgYDVR0PAQH/BAQDAgGGMB8GA1UdIwQYMBaA
FEl0uwxeunr+AlTve6DGlcYJgHCWMB0GA1UdDgQWBBQITtWkPCoEm5MbtwQIjnS5
BnwNozASBgNVHRMBAf8ECDAGAQH/AgEAMAwGA1UdJAQFMAOAAQAwZgYDVR0gBF8w
XTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETALBglghkgB
ZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUDAgEDGzA3
BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9ET0RST09U
Q0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5odHRwOi8v
Y3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAGCCsGAQUF
BzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNsZGFwOi8v
Y3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIlMmNvdSUz
ZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJjYyUzZFVT
P2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUFAAOCAQEA
mffXrLElu68fAzW/Vnv1oWCm2pTuj93MMtE1DZ/lXqZ0En8BKlozIcDXBsq/3Rtm
VE8CVfym32gX0r/0XWuO+chz21tUOt294WnZ+pHbKloPx46INQgjq2Rn298fa/yO
X3Kfl4GHgeWlIX3YT/4xm6F5pCZUQfBFkK9fQsEelof5z8ekGkRTkREOOIBktNkT
1iOOiMepsSAkVnwH+8R79PmcerUORLcyVzpNg5HEdRiUls9f9m82K65zGfjg/GnO
hn//QiE++TjDXnqZKN6YLLCciBCyNB6qCArLTgHFZOtNpafzCD0LenU6lkr3/c8c
r3JMcULZ/iO5WrStVwX9JA==
-----END CERTIFICATE-----
```

## 12.11.17 all-ca-certs/DoD-Root2-CA31.crt

```
-----BEGIN CERTIFICATE-----
MIIFTDCCBDSgAwIBAgICA50wDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTMwMTE2MTQ0OTMwWhcN
MTkwMTE2MTQ0OTMwWjBXMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTESMBAGA1UEAxMJRE9E
IENBLTMxMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxicQL5CWONnf
5l8/uon7ZoLrtqXt8FaQFkDnbKKweWZZ15hiMdEzIlPjHlykVmamTVb7w+JCEqv5
wEpLQO+RE4Y5MFHWbo4ntOGJKQHuWEZzBHFEXGlDPjLmZN+za5kscKLQPk3YWBJt
RfA9k1S+3+L7zxH//IoBN++nLrpADGo+HOQKMoBpvSI57Et2ybFakzwhhDjdcxOC
+VOMgQqpslNO2QuOwOiXuz1fE4y1uTvs9rudjiD2a7ydFDLcfrniY7BqwYC5FvyR
76yyCZ9SR1gTXmJ+mhKGW8UgH+GOZgB2U+znIokhTF+56b6gUpMOpsjezLeCrSJt
i9AwUzZVVwIDAQABo4ICHDCCAhgwHQYDVR0OBBYEFETjRqNB7mCxXqeTJfSgU+63
Sb67MB8GA1UdIwQYMBaAFEl0uwxeunr+AlTve6DGlcYJgHCWMBIGA1UdEwEB/wQI
MAYBAf8CAQAwDAYDVR0kBAUwA4ABADAOBgNVHQ8BAf8EBAMCAYYwZgYDVR0gBF8w
XTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETALBglghkgB
ZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUDAgEDGzA3
BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9ET0RST09U
Q0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5odHRwOi8v
Y3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAGCCsGAQUF
BzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNsZGFwOi8v
```

```
Y3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIlMmNvdSUz
ZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJjYyUzZFVT
P2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUFAAOCAQEA
R1FS3PSgc5pC5wvsI5GNJXW0RII0qvlGdVHD9g745+MvtCDD76FlNOCdh8HmLmLw
J+jrxc81ldJAgIuSCbamG9USZDHbtdQO3wqKtlb1vHaSkxl8v2V9coHYZHs5NIp2
WMwdQ/cHzxyDA30+0BfbdK1pCRF87djWAo1mPatryjPbx3pmxd6nJ0gPZhLuaCTA
75HqBhkqUFgT4CL8DrEk++u0QgIPd4gVi+by9VO3f0BVmxPWtnDKc3DjUyXBKB57
xCxJbpDbqstbAxvCh4f1q75RcXNtJmZ7mx0X4O3jwN4dJ7HtDTRGPt0uXvSCcNrR
kxt53dZK5875P3MfzormFg==
-----END CERTIFICATE-----
```

## 12.11.18 all-ca-certs/DoD-Root2-CA32.crt

```
-----BEGIN CERTIFICATE-----
MIIFTDCCBDSgAwIBAgICA6EwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTMwMjA0MjA0NDA1WhcN
MTkwMjA0MjA0NDA1WjBXMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTESMBAGA1UEAxMJRE9E
IENBLTMyMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAs+KVHZM2LSWl
Dv146e/qk9E6ydhXvRnf0cei0ejZ/dKOFajdvT5k9Lb+nAPfS7Blt6sEGDIZbBMB
UtHmtchBEre+O8tNQBCIyp62/TV3bSb2ZK0RhwypJXpYn7C9mPaTXxvv77KXrfgV
59zmoGp1DVHfVR1oQVJJLsecaFdWR4/e9lIugW9WvAaJEpSfI70/gceGAnUwXjOh
3OETu/15VgE8Shn0LOuQZGTX6AovUYbVCJuE+/npi0LKZdKQBxyCl4xEI1cGLHVp
KHCy7T5M1eOWdxX9upXPW5ZpAnfWgNmPhynj5wV2r8qNEmA0cseznThuTJYynpA1
rXWL0WJACQIDAQABo4ICHDCCAhgwHQYDVR0OBBYEFC/Kk1MDrG919Xb6vv6O6hCL
t+eQMB8GA1UdIwQYMBaAFEl0uwxeunr+AlTve6DGlcYJgHCWMBIGA1UdEwEB/wQI
MAYBAf8CAQAwDAYDVR0kBAUwA4ABADAOBgNVHQ8BAf8EBAMCAYYwZgYDVR0gBF8w
XTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETALBglghkgB
ZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUDAgEDGzA3
BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9ET0RST09U
Q0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5odHRwOi8v
Y3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAGCCsGAQUF
BzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNsZGFwOi8v
Y3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIlMmNvdSUz
ZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJjYyUzZFVT
P2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUFAAOCAQEA
MI3VVmO9mQaLTbbSDgO5xoTSm3dBGojS/8Pa4uZnYb3Zeu04OV6rC1g0+droYnmv
OXLzSqfjTjkQzenSCOrUnpqnNTWTkwJZ4kwAHPP8ayFTSoxh52HL0EYL0T+cafXv
UIrwQLMrVloda2JZBbOPJxgFCkNbAu/dUl5bwKkcVuOVbJdPAYNWcl3XfVHjWlQu
uJj9ck4lj4sW0bDhM+OSfBBVMyRmrw8zBlNIA4eftGR0tdI9InK3OY43ERM5357n
OAwLilkRMmX/9rlGvT82nqeUAFfwwBnhLNxM9y9MkB1D764I43OeOr+Z7CK5B1iu
2TVSS1G7gTaPn24hCqaOhw==
-----END CERTIFICATE-----
```

## 12.11.19 all-ca-certs/DoD-Root2-Root.crt

```
-----BEGIN CERTIFICATE-----
MIIDcDCCAligAwIBAgIBBTANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0wNDEyMTMxNTAwMTBaFw0y
OTEyMDUxNTAwMTBaMFsxCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRYwFAYDVQQDEw1EbOQg
Um9vdCBDQSAyMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAwCzB9oO7
rP8/PNZxvrh0IgfscEEV/KtA4weqwcPYn/7aTDq/P8jYKHtLNgHArEUlw9IOCo+F
GGQQPRoTcCpvjtfcjZOzQQ84Ic2tq8I9KgXTVxE3Dc2MUfmT48xGSSGOFLTNyxQ+
OM1yMe6rEvJl6jQuVl3/7mN1y226kTT8nvP0LRy+UMRC31mI/2qz+qhsPctWcXEF
lrufgOWARVlnQbDrw61gpIB1BhecDvRD4JkOG/t/9bPMsoGCsf0ywbi+QaRktWA6
WlEwjM7eQSwZR1xJEGS5dKmHQa99brrBuKG/ZTE6BGf5tbuOkooAY7ix5ow4X4P/
UNU7ol1rshDMYwIDAQABoz8wPTAdBgNVHQ4EFgQUSXS7DF66ev4CVO97oMaVxgmA
cJYwCwYDVR0PBAQDAgGGMA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcNAQEFBQAD
ggEBAJiRjT+JyLv1wGlzKTs1rLqzCHY9cAmS6YREIQF9FHYb7lFsHY0VNy17MWn0
mkS4r0bMNPojywMnGdKDIXUr5+AbmSbchECV6KjSzPZYXGbvP0qXEIIdugqi3VsG
K52nZE7rLgE1pLQ/E61V5NVzqGmbEfGY8jEeb0DU+HifjpGgb3AEkGaqBivO4XqS
tX3h4NGW56E6LcyxnR8FRO2HmdNNGnA5wQQM5X7Z8a/XIA7xInolpHOZzD+kByeW
qKKV7YK5Ft0eC4fCwfKI9WLfaN/HvGlR7bFc3FRUKQ8JOZqsA8HbDE2ubwp6Fknx
v5HSOJTT9pUst2zJQraNypCNhdk=
-----END CERTIFICATE-----
```

### 12.11.20 all-ca-certs/DoD-email-Root2-CA21.crt

```
-----BEGIN CERTIFICATE-----
MIIFjzCCBHegAwIBAgIBSjANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0wOTAxMjYxNjQxMTNaFw0x
NTAxMjUxNjQxMTNaMF0xCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRgwFgYDVQQDEw9ET0Qg
RU1BSUwgQ0EtMjEwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCa7Qjc
I6BER5v1w57JO9huz+v5xoNiegyD+Y84foLAjZzRQizLiA5iUSKpgBdYQXoMRps+
JahqKKm7Ev38hSvvs1sxT8oVxdO3mEVmBXL2CZpy6Sb/vZAmNQolvbusv9DWOId5
YSx7OQ7TKvUSP0DkmHNowkmsj9SMChevPkpEqT85DWm7Fg2Gjg7pvlN2eYMfXW6K
53HWRcGkzzJySODnEPmxC7XzdPBkGhNAlNITbbIJIVfh3akHV6a9wKSEV765HVFJ
H3xbxubSI/02VVeIyHlF22PPS2o7Mey1PV1nvLJXpS3V7fxM2DuH0UdzGHRvcFNC
hm4vwHgWwbm2MoclAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0j
BBgwFoAUSXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFFFnhBz/q6exnR2E
ZISZXdAL171bMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYD
VR0gBIGXMIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsK
MAsGCWCGSAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFl
AwIBAwYwDAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0w
DAYKYIZIAWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2Eu
bWlsL2dldGNybD9Eb0QlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB
7jA/BggrBgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9E
b0QlMjBSb290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNh
Lm1pbDCBiAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNk
RG9EJTIwUm9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNk
VS5TLiUyMEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkw
DQYJKoZIhvcNAQEFBQADggEBAHLEcRdOUOHK7vqTxTQS+kQV2uBjSdayG16jnh8h
b3iiDwrC2tS6ZscibJNNCUspZxSNmQ4FBvet3EfcDpkICi7yCZfo9SGFzvINNP+a
Q1+TdkqjDNgqIYsNYE52Hq0K7xO7NC4MFVY7tjF7Np85iIvLSLPZBE+fEVjl2a2Z
wBIoI5hw+p1IA2u8oNhOPbaRqaKIaIbCsUgTUtjAgJD4bOghISfjej7RspxknhiC
aDBXhAexdVqZOJIpa/0bMQa3l/rl6zqCZNVOebd2B7c0bqZJykLGIjuDsKQ42zSm
sJUkH8vxH7bA/3um3A/4/SW2sjLWdpkkS3fq/S3EYbmx/y4=
-----END CERTIFICATE-----
```

### 12.11.21 all-ca-certs/DoD-email-Root2-CA22.crt

```
-----BEGIN CERTIFICATE-----
MIIFjzCCBHegAwIBAgIBRjANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0wOTAxMjYyMDI1MDdaFw0x
NTAxMjUyMDI1MDdaMF0xCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRgwFgYDVQQDEw9ET0Qg
```

```
RU1BSUwgQ0EtMjIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCmAAYl
I4D59WDEBgYoBUcuG2cfvrRF5Rw0xvTFutJMaJ1TJNeXv9joB894zproZMUQedNv
xx0zm4jDAPHwKfhT/eClMShoyS6MOAeSRbQ/CALL9+4BgS1fSxWx3YDyucD/qe2g
9Sebeex7JSslESmr/V8RPGKTl0J5SMCdBtG3IyWZV94GVcoeh5MU9xJDMdEmDm3S
RUw44tKa5xKvyUxd48h/H8fKCTnxCU/GoudhgXmZC9KMC2V6uTwYFc4Quy/AZBoy
CNGwoBKMEMuzbKRwQKy9VgtUpdTxRjPc7PZRUq8nJy6dVaQd911a+GRoQYlYvS93
nSjeDXhfiGf8HLyxAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0j
BBgwFoAUSXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFIUvDQqvNQhQCY2b
HHCsqP6Jd5RaMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYD
VR0gBIGXMIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsK
MAsGCWCGSAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFl
AwIBAwYwDAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0w
DAYKYIZIAWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2Eu
bWlsL2dldGNybD9Eb0QlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB
7jA/BggrBgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9E
b0QlMjBSb290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNh
Lm1pbDCBiAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNk
RG9EJTIwUm9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNk
VS5TLiUyMEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkw
DQYJKoZIhvcNAQEFBQADggEBADhU9UuJMePz9RcXpXSfyz+JU+9B6ldRxeTizrr6
QB+YTG5Day3PwBlu9BdH3ZaQRqZzL+3xJ3iHT7ftATRueWi/hclcZWy5e5gqip5d
YUAmvOSHNZ8D6s7JeQwGfmjenVXD0QoIf9jm5zqDVpfj4cOybztEdrhzbOrwxyBM
jzFVgIZdHuY5RJmONKFp+W1fcg4FR2maCOxl2SmAn+CvfgEDuAvpE/dYIdYw/qDu
cnuBeYENlWCEPcpItgx7iXNfmF17Hg9pqgQmfGqRcP3zYthQT3l2umlW+r5uu4xX
b7HH/i7fhWXCshcUGRwWE/1+HW+yJ9YTHAxZkHC9VryuAoY=
-----END CERTIFICATE-----
```

### 12.11.22 all-ca-certs/DoD-email-Root2-CA23.crt

```
-----BEGIN CERTIFICATE-----
MIIFjzCCBHegAwIBAgIBSTANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0wOTAxMjYxNjQzMjVaFw0x
NTAxMjUxNjQzMjVaMF0xCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRgwFgYDVQQDEw9ET0Qg
RU1BSUwgQ0EtMjMwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC99ClN
V5NEAweNBh+u+jw0VjRA57EYf2wlbheFBQUj6fbFUCVPgeQjMEJxaer3uJ73b6ze
Xar81uCNvGvufZmVIjuzWMaxUhyyqL8xQCIG/o0o0qlQVWoeB3D4pkjJbf2u7L6A
bD3PkNQHok6RFAO/V1kS9XTeQ5ZaWrnPuUfof9COsPjY6Us0XsxLF44C8BK/8gRs
HRO/qxzeDQnsy5tW7dmQ55alfyZlYcHEm2gkpc3SeSNvwzzBhR5I+T5QcWKgQbpy
RKVD46Vybs3Oq9rLhNIavx9uchE/LZkfbbD7BTDO5uwjKmVHH3icDZ9MJHVsdtLV
0xdEFrKKKEjXuQ2vAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0j
BBgwFoAUSXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFFuWLTGl4faaalxe
gVE2YR6WJBnRMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYD
VR0gBIGXMIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsK
MAsGCWCGSAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFl
AwIBAwYwDAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0w
DAYKYIZIAWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2Eu
bWlsL2dldGNybD9Eb0QlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB
7jA/BggrBgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9E
b0QlMjBSb290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNh
Lm1pbDCBiAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNk
RG9EJTIwUm9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNk
VS5TLiUyMEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkw
DQYJKoZIhvcNAQEFBQADggEBAD729uQP1FfNvu3pxEVS2kVbGKY2OTAKtn7r9on8
p2Iusz+DfeESGobTY5T0dPTOcZfq+8RqOl3imaesw+I2+Oh49NjEUO6KgVX6ioNl
DMXyTczpH497Rt0DCpzq4qxjdwfLMlTbCFWyWAB9XKa5FjxfZW0vBc5aP5rbScuS
```

```
o6HZb1HU9cAIwaM5W9BBY4HElGVkYylMXfBfcYdqnZaS5ceC/S101wJsyuLboLPb
cdUOhj4+F4m9bkIXG7T2OfkaveYuLJsONzsQX0T+e7WUWNZxJeU50SO6NADBQ224
9A6Xq7Iw9oinjo6KEAOEdNyuTfnlmXQaqaIKbQsHFZTFP2M=
-----END CERTIFICATE-----
```

### 12.11.23 all-ca-certs/DoD-email-Root2-CA24.crt

```
-----BEGIN CERTIFICATE-----
MIIFjzCCBHegAwIBAgIBRTANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0wOTAxMjYyMDI2MTVaFw0x
NTAxMjUyMDI2MTVaMF0xCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRgwFgYDVQQDEw9ET0Qg
RU1BSUwgQ0EtMjQwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQClLlh6
od7mmlv2AvHV1Nw1I5p7bihkdBpwPJYdzMKfdAQ8DDmSIQgNEk6g1zeo0snGJ50o
+lXXshcEGc4yvPB5nvVoqy7MzzcEsvgKZZpJIBQlwbwSaqBCbRsItIehQiKrE5na
AgE5H14IV2tg3hN+aGp+QfWJgDh6/Zey0uKWSzaAYrbsJbvQD6ejzVGo99J5VZA0
JqPkXM27aCZ0CTeh5q/N5D6ZR/9/wke8ZYS6MimjDvDColt66rJKfQvGw26svRB/
T6l2Oj0CASwqMLT3yKDSmDp8CNBaiQ+1ioL6DTAeftbRx7ZDJ7EoqQzjswd432Jk
mkWMTs2vDq6cWDbfAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0j
BBgwFoAUSXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFFSqcyrHs3fqzSJA
eUh7EfunmSKCMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYD
VR0gBIGXMIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsK
MAsGCWCGSAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFl
AwIBAwYwDAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0w
DAYKYIZIAWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2Eu
bWlsL2dldGNybD9EbOQlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB
7jA/BggrBgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9E
b0QlMjBSb290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNh
Lm1pbDCBiAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNk
RG9EJTIwUm9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNk
VS5TLiUyMEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkw
DQYJKoZIhvcNAQEFBQADggEBAHIW3DlzY02T6Tccz7LtnNhN9wwySomes8q68wSs
cWYxpiq9un1U2C8JY0qICOhsE6HsXntWFzAtyNLt141HRGnPEW/L2OdSdbVRyKod
afAZHzDwB8c2vc4M3jt2/QrOy7YTutaFi/FcEpHKr+h/EqisLYvWdlCU7Db6ow/f
xjLqx3NG/IQami/E6CccSMJGNvYX7O1nMg+4ouC30l6QBhOUIWFDbH3zO2tl7ePb
qP/Fm7KS5+tf7u+/8zmMs/UX0obVw2xKOmw/nq/oWx02W6YmFUYLRmvH1ICq564c
uCtO+iFyn1+fga+07lvJlymJfOnceOJO4HSf0oZ4ZqLHmKg=
-----END CERTIFICATE-----
```

### 12.11.24 all-ca-certs/DoD-email-Root2-CA25.crt

```
-----BEGIN CERTIFICATE-----
MIIFjzCCBHegAwIBAgIBTzANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0xMDAxMTQxNzM2MzJaFw0x
NjAxMTQxNzM2MzJaMF0xCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRgwFgYDVQQDEw9ET0Qg
RU1BSUwgQ0EtMjUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCiR3Pf
QKAVpqftoOnIf/vSjk8J5cWNXGQaQ37hc7mvEEV9My3qYCZzYiGLL66cF8zV5jih
Le4Cs/C53qaLiNG0Yz3IiIgcf15C6x2T86t46ZQdAz1NPhkXfO8JIoL+w+Sfns3Z
vYKEOQxSt327QX/1jaQq9tBcYjHI4+q3t3jWm05iXrUS28p0XbhqEUNJFYO5aWOP
TWLC8gR3WQSrBc6sFF6ZfR9Oi9TJope6ztCc4502/oyB/Gg5TZ4j4oOz06vg2d+Z
ZArINPKs4vVQnl5t0Q9FslLrTpJvH2nIoTcYbWHIhqrPxLfMN0n6fBtmcoFKRoxB
SqsgCnb3zhLU0AZVAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0j
BBgwFoAUSXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFCbb67FFLtgSkE31
```

```
EkH1w/AezODOMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYD
VR0gBIGXMIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsK
MAsGCWCGSAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFl
AwIBAwYwDAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0w
DAYKYIZIAWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2Eu
bWlsL2dldGNybD9EbOQlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB
7jA/BggrBgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9E
bOQlMjBSb290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNh
Lm1pbDCBiAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNk
RG9EJTIwUm9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REbOQlMmNvJTNk
VS5TLiUyMEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkw
DQYJKoZIhvcNAQEFBQADggEBAGmQDW4i6bsFtubPOCSokDN5muLZYBEi1ewoR5Ag
N+KEhLJo9n3+i0sRuR2/28zc6XJ+RsmMSKk2hmCQmdr4WNty0KmObmDIvGDrBNAO
+HGF51vpwfvskpWqA2n9yDQFZdUCT0+ZgxrIRlw7/vhx2Hw7PVzRzYJMQ431Gqao
L0sCdNco1pFG0E1jja30lIiYIyOLtu2OQE6G9Nnp0TZK1FPAS5bwsbhuQJxqMnxl
bbZg7YFKUFdTY2bod8d53HcjCz1jSm276E9DJM9tmFwR6C+IpTrlTsTY0P6cmOQy
rY4nFFWr2si3dkL7WRiSuAormmbMMPvEY2omt7eRHRiiPpw=
-----END CERTIFICATE-----
```

### 12.11.25 all-ca-certs/DoD-email-Root2-CA26.crt

```
-----BEGIN CERTIFICATE-----
MIIFjzCCBHegAwIBAgIBUTANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0xMDAxMTQxNzM5MjdaFw0x
NjAxMTQxNzM5MjdaMF0xCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRgwFgYDVQQDEw9ET0Qg
RU1BSUwgQ0EtMjYwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCsXxBb
PSoH/e7PTsawTj1aoABgUHnCAkVmTzg0gVwcvydgannMppcL1onm2NuVfFc6B+5G
5WQqExePHTD8Lfo2fzdhJOUUov0iVxxhrk0uA1BmVUbdaif4qXmrXCrlJV1cG/tx
D7W4FY9flHsDz+6rkggK5L2joV2D1z3Hn9REEDqiX1/khpRvA6A184PY4bgZn3q6
dc8ABdDbI6RqJddpcEXGXXiLB19FrJ3Wo0tdGM+PTAoRodkR2/mcpdWPnOoPR70l
gpT5YJJKFPi6m6ls38oVEaGLOb76GU28uxRv3WB9spyQB3yAR7mFjLg+o3W5rl53
kXPBdYlVuk2G5K27AgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0j
BBgwFoAUSXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFDLfyG3z/z4p/ekM
lylQ8KIQLG4vMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYD
VR0gBIGXMIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsK
MAsGCWCGSAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFl
AwIBAwYwDAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0w
DAYKYIZIAWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2Eu
bWlsL2dldGNybD9EbOQlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB
7jA/BggrBgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9E
bOQlMjBSb290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNh
Lm1pbDCBiAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNk
RG9EJTIwUm9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REbOQlMmNvJTNk
VS5TLiUyMEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkw
DQYJKoZIhvcNAQEFBQADggEBAGrAUCmE+NxnE7GW7rpc1OWS+c1kZTOFKAugGqtT
HYxAF5G6Ztpra7ysjmEBw2c1EVlShXBdoYbnesEcw9hey3e7zFzcGt0EX/qI7bNu
tbREyzo1naBOHMBFtfbUzQZ5Oho57CUmcZzZuG+TbNY7NDtnmapfpbhtTMcJ6snA
dJnZWYspiZArgZXZh/1V+Fh1UqZ/ImhthdZ9rooNLzS/1yhsxlutvP8bOsZkhaSc
fYSVn6gDZeR/TcwMdXpKBURYgIs5NE8zPytE8dZ07+98mtjcCxg98uWdUDsjKeX0
z2DqYE8cYMEaspxaAgSwfMHJFWrbKq8LCLs+cXqmPOUdRCI=
-----END CERTIFICATE-----
```

### 12.11.26 all-ca-certs/DoD-email-Root2-CA27.crt

```
-----BEGIN CERTIFICATE-----
MIIFUjCCBDqgAwIBAgICAbYwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTEwOTA4MTYwMDE4WhcN
MTcwOTA4MTYwMDE4WjBdMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTEYMBYGA1UEAxMPRE9E
IEVNQUlMIENBLTI3MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA053C
B7D1fszurrirqjPqp5JuE1ZAaOUfxiG8wIGXYSxw0VoVF/6co+9IaYm3W1L5rOA6
nKZDViAogmzN9Zb+gJ3ZjfHR7oGavOzwhTUWQbkmiQwmekBu0AmAUcAC2O6Eb8ws
giKqNYVepF6FBNEJmaS4fVKxIXpN2CGnvERPyhWijDEuidY5L0BWN3jrLl0u0RhH
Fu2soITUC4KYvQMYcLAZXYxr3jUkYlrI+w+6euzIQElyVp4aTVTATuUQNE9h0dLt
Td/RWbDrAkIvDBtSDBWg8u66Nlf7zKwR8ZotTIspGPHwcJJo1kEmzFt8dXbYBWBS
Own8rcBAHKRG1jAtewIDAQABo4ICHDCCAhgwDgYDVR0PAQH/BAQDAgGGMB8GA1Ud
IwQYMBaAFEl0uwxeunr+AlTve6DGlcYJgHCWMB0GA1UdDgQWBBS/yO1EDrsz5sfK
QSylMbnJYGGJLjASBgNVHRMBAf8ECDAGAQH/AgEAMAwGA1UdJAQFMAOAAQAwZgYD
VR0gBF8wXTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETAL
BglghkgBZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUD
AgEDGzA3BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9E
T0RST09UQ0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5o
dHRwOi8vY3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAG
CCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNs
ZGFwOi8vY3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIl
MmNvdSUzZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJj
YyUzZFVTP2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUF
AAOCAQEAEX3yTl2o1kYa59nUZrFRxoHg5n89ca5Gp3ALg7S9wEAzUJuHQ5SHBWlO
vmdt3SrsCjEEv3iVb9ix7EMnCs8AgAEWPH2XN4WYW6aAcwyzd/7JcDNSi3p1t7ku
/rwtJUaW+kVteCjN25uZTAeeLGINitt/eFUFRxIb25kCN/lnHwQx7yiBd3ZaLpSL
dXg9icx40EsFmKLAcBaHcP+LfAnS4S0y7QPtYSuN2s7N0jzj5o/2ceO6L1yEgm6I
pl06q8Ft/mf56avlOETvQxvlKrEw+/T7b32kIABUYCI+XTNku5TqWnaVn8iPBms6
YOjCRiZTq7rmKMv5W469Q63xx2gyvg==
-----END CERTIFICATE-----
```

### 12.11.27 all-ca-certs/DoD-email-Root2-CA28.crt

```
-----BEGIN CERTIFICATE-----
MIIFUjCCBDqgAwIBAgICAbcwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTEwOTA4MTYwMTE5WhcN
MTcwOTA4MTYwMTE5WjBdMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTEYMBYGA1UEAxMPRE9E
IEVNQUlMIENBLTI4MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAoL6S
zgqY0wjhxffuXYJK28/KZvS9cG6TG8qbOQFQGMDTVnrLWGfdBaTUKzFBYnoK/cL7
HoUsivnrqbOfs8papHhWVRdBl4n1zccwxt7IpRbq2OCGKBEMeA2dN+4Y+RYtX66E
bjSLukY79D16oz/jrpuph3Z9w7fgsi2COkF/uWnhUCKxv0xRakr5Aw4UtzKpX40b
71FXvIcpW2UmP/nzoZI4qNkxxxgRt+uKGNOQpc0JsrUs7wlpnsil12IiD9qF4Bqj
NLQYjKl1ScbHboSNqaSQX61brhOXXalfYA/cxJGSNgN7/WlZydb659zg/lo9XD/0
PwAbWF/TCUfvUHLIMQIDAQABo4ICHDCCAhgwDgYDVR0PAQH/BAQDAgGGMB8GA1Ud
IwQYMBaAFEl0uwxeunr+AlTve6DGlcYJgHCWMB0GA1UdDgQWBBRZiDBI5m3+YSem
xNWFjVtznu/BzTASBgNVHRMBAf8ECDAGAQH/AgEAMAwGA1UdJAQFMAOAAQAwZgYD
VR0gBF8wXTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETAL
BglghkgBZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUD
AgEDGzA3BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9E
T0RST09UQ0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5o
dHRwOi8vY3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAG
CCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNs
```

```
ZGFwOi8vY3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIl
MmNvdSUzZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJj
YyUzZFVTP2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUF
AAOCAQEAHWveFxw66X0qZH10AyB2ZE2foB7ZW0VKdzHMZSta6bZfXu42iBAU27d8
AEyxbTkJGXiMMml3qmSefHHXSbEsoN8nMVIqmYL01lNGSszA876YH2ATi+KKB2R+
hUyxCbHpWIrNmX4SwpNL1/WkFD7EewgwQ8gmfhf2U0m/au62A5LDAATJSQeJ8EGt
19/M1/MmhGJQshQ2ygsG0imA+Y0rpUSG4oEs7SADS0SvD5hBVMXAGIchy9WDTGaR
exYTV5GXdJK9AZUoe07i2tZWIDbSy0Z9dMqK4/nWwEInSQP0PwUPqtilvzMuFg+u
HbH/yUvYcWuTxaH/ajtVXhk3XlsjyQ==
-----END CERTIFICATE-----
```

### 12.11.28 all-ca-certs/DoD-email-Root2-CA29.crt

```
-----BEGIN CERTIFICATE-----
MIIFUjCCBDqgAwIBAgICAbgwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTEwOTA4MTYwMjE0WhcN
MTcwOTA4MTYwMjE0WjBdMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTEYMBYGA1UEAxMPRE9E
IEVNQUlMIENBLTI5MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAkmIv
QcIgABYGVWSfvaeIFW6Cm0jBhXe9AqsM2fErYIEBuj51cI4Spqc4hJCz6UCAEtxq
ylHNrS2GEMxEvA7FWDgZshQyJUFUWFxDDshscw/DDBgYFgSaUj2BonHOPDIAn3FV
uvjONnceIbcolOc9Pqb2wHoxYJEol3ciUPLGk26yG8VBxvmhN/sQv9pWpvtSTV+/
78SWdyjlMv/o4RjMQ1IYrI13mnJM6JODXrCi7+Td0ufmp6ZSreGYCJZKQ8xzPUui
jYnv3IJMuEqAJGUrHpGC9QT2ch9XGEAX8DlRto/ziTtn91hOSrza+Q7BwAy98whx
+IMPyS6AlfSFDs6uqQIDAQABo4ICHDCCAhgwDgYDVR0PAQH/BAQDAgGGMB8GA1Ud
IwQYMBaAFEl0uwxeunr+AlTve6DGlcYJgHCWMB0GA1UdDgQWBBS4Q4NkIXrucIHe
pd4MYCiHeK5eeDASBgNVHRMBAf8ECDAGAQH/AgEAMAwGA1UdJAQFMAOAAQAwZgYD
VR0gBF8wXTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETAL
BglghkgBZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUD
AgEDGzA3BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9E
T0RST09UQ0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5o
dHRwOi8vY3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAG
CCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNs
ZGFwOi8vY3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIl
MmNvdSUzZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJj
YyUzZFVTP2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUF
AAOCAQEALGsseTXb8B4ch3ur4ehpajeL23pPVWBplS9TncbKQ7bUN5HWA11+WrG4
HfeegdOuUFQwpG9LLrsUGxeqXBDTlHoxOZakVHn16VYuVcMbFuqqAsjPUfcygSLG
NDqpzZqqSJPH6fseMn5xxHbwRVQSHVXqvVwyhzquk5pumSJfqFE17rJTYF/2T0W4
FoQdZVXNFcoQAR+p0pynV5Gj1+ewhj0t9Ik62Ml3cFDGb0/y65j4EKo92shcKa3O
uHNJTKGSu+btzbqCGmMhGWX0Bhm/g6pz5dMbsZj/Rd/7Scxz60LnB5YAMel/2SQI
58pEekgGw0LYP/l5h6U3khaphCCSYw==
-----END CERTIFICATE-----
```

### 12.11.29 all-ca-certs/DoD-email-Root2-CA30.crt

```
-----BEGIN CERTIFICATE-----
MIIFUjCCBDqgAwIBAgICAbkwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTEwOTA4MTYwMzA4WhcN
MTcwOTA4MTYwMzA4WjBdMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTEYMBYGA1UEAxMPRE9E
IEVNQUlMIENBLTMwMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA5iki
1BQm0ZgaUl7FhINzfsFgs7PQlL79HJRVv/aELJvJwHRz78zCmfKZyW3KFNN0/74Q
8vctv8u7BqPumFBBZQHhVyy2y+TKHKx+UjQ0sY4HJj4yNa+jYQrF5Qi2EnmMVMF6
```

```
6fFQH12DOmcwsynbHTpMOSFQ2BgsjQZ17mNyeGitYpx1pJQGOzJrEq8GBym+E6DA
p/AlT7f+H7dX4BgSjSFqFblaVPt3ZdhMP/W6PMA34QZ+wr6eI4woOZrXxmc413PJ
vQcdhW/VlQqa3No6TijwpesJ3+XbC81Hr4rNu2+UQONZnFCfyQ6pcQK53OlpgDqJ
OOUFIhgFhLUS8DzAgQIDAQABo4ICHDCCAhgwDgYDVR0PAQH/BAQDAgGGMB8GA1Ud
IwQYMBaAFElOuwxeunr+AlTve6DGlcYJgHCWMB0GA1UdDgQWBBQ1YWYoCbxWJVuL
zL+BXmEsMDnTITASBgNVHRMBAf8ECDAGAQH/AgEAMAwGA1UdJAQFMAOAAQAwZgYD
VR0gBF8wXTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETAL
BglghkgBZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUD
AgEDGzA3BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9E
T0RST09UQ0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5o
dHRwOi8vY3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAG
CCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNs
ZGFwOi8vY3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIl
MmNvdSUzZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJj
YyUzZFVTP2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUF
AAOCAQEACohWHKVXJlpiy3XQ3YbFUuIv87wRZD+MLz4R/JhgQPKADSiCmmj+4EhL
J9M6CnuV9gMMgRSRQjpgbOIrUy3s3xGu9VQX8AH5lwenm6sL26yXiQnG7/kHNBYA
qH4RU558L6E4opl5OTRBbn24WDBWiJ7kqmRF2aBEYjq35THTkYDxGxCyZ3DVW6tZ
tFpIFkLEAkzabGjKUB0xvjeZx89TzEIpVsOdF8oD5xBa8Tk8HMz7G5cKJvMx3+Cr
XCSdnt44fQJRZ0b5k3CF7QpVwvTBaFqfCMkde5t23FTvOYwY5QxE7vcGsh/1y+YO
vdSh/9T5kQciUnm3wP3ssviF9ET7XA==
-----END CERTIFICATE-----
```

## 12.11.30 all-ca-certs/DoD-email-Root2-CA31.crt

```
-----BEGIN CERTIFICATE-----
MIIFUjCCBDqgAwIBAgICA58wDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTMwMTE2MTQ1MjQzWhcN
MTkwMTE2MTQ1MjQzWjBdMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTEYMBYGA1UEAxMPRE9E
IEVNQUlMIENBLTMxMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA6K4C
LEBMOlLoi3OStHfnOEvA8KpKGFzH9zXDSvDwlnell74n78REIYDqFjS3MNFEOH8q
zgTGkWWpblB8yE7+vcC1SxbkOFIV27O391M98rEH25FmXcG38ndmxFGaY5QRSwId
DUt8swBHB3kY+nizkx/Udm2ZBMUeNkb8BjQL42hvHnyfLM9huEv/tN8Gn6BflF7r
Nf8JXTVAB/Kd7ZYJ2Xbq/m4x/sv0ResweEhobKEpPoZ9k0FK6ucMTOWRUCqlQ2a8
IsD8Gyzk8y9iHgTUIb+sHyZ3NdAdvOK7RsLy6+QUrviza7P6cTiwcSnt0Ysb1wIb
3srsfu6h3Eil8T6UqQIDAQABo4ICHDCCAhgwHQYDVR0OBBYEFIbxW2hv3TDzlIJo
1Ez3RB24ymiBMB8GA1UdIwQYMBaAFElOuwxeunr+AlTve6DGlcYJgHCWMBIGA1Ud
EwEB/wQIMAYBAf8CAQAwDAYDVR0kBAUwA4ABADAOBgNVHQ8BAf8EBAMCAYYwZgYD
VR0gBF8wXTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETAL
BglghkgBZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUD
AgEDGzA3BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9E
T0RST09UQ0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5o
dHRwOi8vY3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAG
CCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNs
ZGFwOi8vY3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIl
MmNvdSUzZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJj
YyUzZFVTP2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUF
AAOCAQEAWTKtqsP435xknHEJNMG9vGMAHi3b7anICOO5GOSvyq4Uwd27+XODg1eO
lMmgqgMHzmecteUXWT8ouBc22rqNw5YRAWpQ1gbaaKRK0guFfM2I3/9ed+b1pEiR
OE6iZ2r4aO+qF0Xv2JYK3c/wPoe2v4g/01S+PhLOofkLbzLRVL+EWzWg2wdktavp
eR7i8qp0cueREvfHu27u5XSQECSLt+fNnIWQR+Tib38gvSy7g5YjTahM2H4uXhUp
uCV9VzULLRVUjKnc4OU3nahPIJWDK8USNj2oc+FOiEmlubv6CUooWjO55JJ5W3v4
pU/zyTTNmYywumB+n4Q+5jz6flrr5g==
-----END CERTIFICATE-----
```

### 12.11.31 all-ca-certs/DoD-email-Root2-CA32.crt

```
-----BEGIN CERTIFICATE-----
MIIFUjCCBDqgAwIBAgICA6IwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTMwMjA0MjA0ODEyWhcN
MTkwMjA0MjA0ODEyWjBdMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTEYMBYGA1UEAxMPRE9E
IEVNQUlMIENBLTMyMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAo/qq
hsqKGhsDTnFtQbbZZZpu/zYqPwLTfJVliFqk969jt1LHGvu7lXMHQmGLSqZ76VYH
NhuqNwIgHKTO+7bQaav8OEzI20ZW96JefucxtO7B/81kv3mCQSt30vh9q0yP98Ye
PPiOLz0Ug9qSmAnYOMZaWTaLh6KJ3b5KXsvNtkd+QaYJVGxBlnRbBsPUwS5GfV42
342iRnGsSrrEsffJFwov3aPshCHPqAXqueMub59+fbsdFnVPkh0D5hE4mDZ6odQA
PK0QWK8VxzZL4zubTbW0kL6tq9PAhLP83BWICYwRUFAv5HDstwquSlPiNsQFboB1
Eo03RvJLDDgcSR+sgwIDAQABo4ICHDCCAhgwHQYDVR0OBBYEFAqwqjhWR3sWfb6r
k5a8VN2F++0sMB8GA1UdIwQYMBaAFEl0uwxeunr+AlTve6DGlcYJgHCWMBIGA1Ud
EwEB/wQIMAYBAf8CAQAwDAYDVR0kBAUwA4ABADAOBgNVHQ8BAf8EBAMCAYYwZgYD
VR0gBF8wXTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETAL
BglghkgBZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUD
AgEDGzA3BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9E
T0RST09UQ0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5o
dHRwOi8vY3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAG
CCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNs
ZGFwOi8vY3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIl
MmNvdSUzZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJj
YyUzZFVTP2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUF
AAOCAQEAD72PR/+5yb1D5c6+tfM5y0UWWaPftlIkPAlVS9m/lXq9dtngMIfNSqmj
LZ7ZKATGlq4BFIDQJVbxWANV79KoIlKrge8A/q/HSdKMIC6kcYH3Jss0pW3VQXd7
LTO7m7N8nD89/8LuefKJChCMkHRdNGdwvgL+gEYZB859L5aoxBPQ758psTSpuYyl
iTSzjD5H+GaMkdHuq8HqcYXJX7Cp7tsA1DAqQs5XYxAiMKichkESXb5QfBP66yhz
X3IziV9/DWikPf0WJugKk/57H4aBgCe+Z3GGG33Hb7epcQHGY7NzfQFrMyLteYmK
DuZyAnM3P8sxge2k+wtqO1KEukz3jg==
-----END CERTIFICATE-----
```

### 12.11.32 all-ca-certs/ECA-IdenTrust3.crt

```
-----BEGIN CERTIFICATE-----
MIIFczCCBFugAwIBAgIBDTANBgkqhkiG9w0BAQUFADBNMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNFQ0ExFjAUBgNVBAMT
DUVDQSBSb290IENBIDIwHhcNMTEwMzMwMTMzOTIzWhcNMTcwMzI4MTMzOTIzWjBz
MQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQL
EwNFQ0ExIjAgBgNVBAsTGUNlcnRpZmljYXRpb24gQXV0aG9yaXRpZXMxGDAWBgNV
BAMTD0lkZW5UcnVzdCBFQ0EgMzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoC
ggEBANk/CDdX01aUBBefatM5zr+RBDmTO23vnZQbOOCDV/qX+k3l1urBEqs17YUO
ws7uxCA4+ubPsmb4RRa9V/8uCvAweAT9ppw5l2Kisg+IW6Qj1FCdPfHzv/Kem+DO
QD2AQWdGATfGAeZom02XuMs1dvzTNed11kaPtrL1OibhJ3H32vYkKs/I9rxcgwua
ot2kB9IZlvjr+J15yqbvALR24nwFpTkpvXPfCY+3wwJ3oY3pWuUuHgoKpCwFANrD
IV4JH1ZJjysf+9SJG/QgIfef3+uoDaQNa5KB0t+BSzWa+qul1Lx4HfuoirFgjPDE
J3REQSk05x/HvfkmdWK79qsB0tMCAwEAAaOCAjYwggIyMB0GA1UdDgQWBBT+Y47I
a3q7bhLQ4VLiJKEMLgcvwTAfBgNVHSMEGDAWgBTt5IfQJ8RQ5oQ698z36zpJ/FJO
ITAOBgNVHQ8BAf8EBAMCAYYwEgYDVR0TAQH/BAgwBgEB/wIBADAzBgNVHSAELDAq
MAwGCmCGSAFlAwIBDAEwDAYKYIZIAWUDAgEMAjAMBgpghkgBZQMCAQwDMIHABgNV
HR8EgbgwgbUwLKAqoCiGJmh0dHA6Ly9jcmwuZGlzYS5taWwvY3JsL0VDQVJPT1RD
QTIuY3JsMIGEoIGBoH+GfWxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNkRUNB
JTIwUm9vdCUyMENBJTIwMiUyY291JTNkRUNBJTJjbyUzZFUuUy4lMjBHb3Zlcm5t
ZW50JTJjYyUzZFVTP2NlcnRpZmljYXRlUmV2b2NhdGlvbkxpc3Q7YmluYXJ5MIHT
BggrBgEFBQcBAQSBxjCBwzA6BggrBgEFBQcwAoYuaHR0cDovL2NybC5kaXNhLm1p
```

```
bC9pc3N1ZWR0by9FQ0FST09UQ0EyX01ULnA3YzCBhAYIKwYBBQUHMAKGeGxkYXA6
Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNkRUNBJTIwUm9vdCUyMENBJTIwMiUyY291
JTNkRUNBJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJjYyUzZFVTP2Nyb3NzQ2Vy
dGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUFAAOCAQEAsLDHwjGsJNIq
Z+eYNsEC7r+XRSST8IUnyZpNMArIWsTBxGzR5Vl3TLhQsmnvzUBVUFmyTiHQuGRI
EA0UCqVDMabQf06JyD/Uq1lvH/SrNaEgXhtVopz1TsleBR9k1e7c75l/BMsX6yag
P16T0h7tEZ2mlLLpO0+C59aPhREc6uGaSzf8hBByP5l4+y1BT0HnX5bgLKybzUc7
zkWpf65SCsjhgAZNg07sLQaTa9r7ZNn+2oCoJug+pdaBcz+NI4YnIadEm+bjDpYZ
gDEkuS8crPO/imsQezF/MFa9cYLsGx9ldQ1layTsxrX2rcTIZCLbiYjaoeagIHoW
RmforANPiQ==
-----END CERTIFICATE-----
```

## 12.11.33 all-ca-certs/ECA-IdenTrust4.crt

```
-----BEGIN CERTIFICATE-----
MIIFczCCBFugAwIBAgIBFTANBgkqhkiG9w0BAQUFADBNMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNFQ0ExFjAUBgNVBAMT
DUVDQSBSb290IENBIDIwHhcNMTQwMTE2MTQzNTMyWhcNMjAwMTE2MTQzNTMyWjBz
MQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQL
EwNFQ0ExIjAgBgNVBAsTGUNlcnRpZmljYXRpb24gQXV0aG9yaXRpZXMxGDAWBgNV
BAMTD0lkZW5UcnVzdCBFQ0EgNDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoC
ggEBALHnQ6QFAEyZmXYyHAp1pSaRGro2Jn5aeUhmk/z58RLw6ULITryK62OZCnkE
hhqzpDjspgUCLNZWCiQIgWU6153wEOle+mj6DwnTi8nO3MplwVmCY/68qdaJP5Gj
BzjDGlSHBEnMD9ikoydOde92Sdzc4GMAFsJvKO/xZDwDHOJP4qc2ryb2qHlFxkMO
LwkWMRzmS2aGLZ+GYOty4yo6gOK99fYWQ81N1fe4AbGkwPGt3ot3EqxDo09VCXoB
2dRdrvx60WDEPqN2yDXffqYWzm/QgrFbRzoW3LoOWrYVPBA44M+CmvtFsPCHPedN
MxWAGaSttBSHz1QJXmEY5fXSU4UCAwEAAaOCAjYwggIyMB0GA1UdDgQWBBQCpTH+
NMQorYysM2CdbQgWdluTKTAfBgNVHSMEGDAWgBTt5IfQJ8RQ5oQ698z36zpJ/FJO
ITASBgNVHRMBAf8ECDAGAQH/AgEAMA4GA1UdDwEB/wQEAwIBhjAzBgNVHSAELDAq
MAwGCmCGSAFlAwIBDAEwDAYKYIZIAWUDAgEMAjAMBgpghkgBZQMCAQwDMIHABgNV
HR8EgbgwgbUwLKAqoCiGJmh0dHA6Ly9jcmwuZGlzYS5taWwvY3JsL0VDQVJPT1RD
QTIuY3JsMIGEoIGBoH+GfWxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNkRUNB
JTIwUk9PVCUyMENBJTIwMiUyY291JTNkRUNBJTJjbyUzZFUuUy4lMjBHb3Zlcm5t
ZW50JTJjYyUzZFVTP2NlcnRpZmljYXRlUmV2b2NhdGlvbkxpc3Q7YmluYXJ5MIHT
BggrBgEFBQcBAQSBxjCBwzA6BggrBgEFBQcwAoYuaHR0cDovL2NybC5kaXNhLm1p
bC9pc3N1ZWR0by9FQ0FST09UQ0EyX01ULnA3YzCBhAYIKwYBBQUHMAKGeGxkYXA6
Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNkRUNBJTIwUm9vdCUyMENBJTIwMiUyY291
JTNkRUNBJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJjYyUzZFVTP2Nyb3NzQ2Vy
dGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUFAAOCAQEAGtC8m53wUMrE
fr3UeIHvSODXpy9xlGohvWlFWIIm03BpRVZ/FeTvhXMlLhylfLK1w03XF2xlzJiM
OVYUrAsHaHsR9t+x/3sdbKt8V6Fy76KN57ePls4CeCp5eYN+C9GAbb1ytwdwBll5
5XeFJYJF8+cRKeJt9eXhCTPyzC9JbzEh5HeYx5KF/ay3L+sSlV6Y45cqo9W23xQ/
xeDdW89r5znydDPRjI7dg+WTGkBMibeC2F/FO0xXD8D0jNncEzquSJdYxsy2JaK7
9p/OwmgXSuopbl9dqhV23rOKyYG9dLhaUXvLTvtM2aeHP9WdlMTMqVPIG2wdz4JT
ENxM3bOcWQ==
-----END CERTIFICATE-----
```

## 12.11.34 all-ca-certs/ECA-ORC-HW4.crt

```
-----BEGIN CERTIFICATE-----
MIIFcDCCBFigAwIBAgIBDjANBgkqhkiG9w0BAQUFADBNMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNFQ0ExFjAUBgNVBAMT
DUVDQSBSb290IENBIDIwHhcNMTEwNjAxMTM0MTMwWhcNMTcwNTMwMTM0MTMwWjBw
MQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQL
EwNFQ0ExIjAgBgNVBAsTGUNlcnRpZmljYXRpb24gQXV0aG9yaXRpZXMxFTATBgNV
BAMTDE9SQyBFQ0EgSFcgNDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
```

```
A0B79n1VvgYvD3h8KYKQ5zYEkaDlC/ZTI0aho2J8c+d7bH9glo0X+tQ1pJmqX8Nz
lNXhh0O00iNlcNFMvX0OujNtlEKln0HTSajCGK1it2Xg51UVstE1tC2b6FpvRVZ4
R78m+W2H0Y+YRoAdxssgXWrH/VtxeMSnwETzin5ajFeeJVl/dEGW/QU63jykjHBt
vek6YhN3VRLmw+JGhDsp0NUn95Xry1+O0dr+Qu5TL4qNtCg20aeDvUEKWoFpTdiF
c/VJ979Km7SI6cfv+FDg4T9YLZtuXnReub5V0Z+EwXLHHFt2ykY3zqTphCaJICq0
rTBhZrF1FEiye3tPj1Qev1UCAwEAAaOCAjYwggIyMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDgYDVR0PAQH/BAQDAgGGMB0GA1UdDgQWBBRbVMHfW3fdLfTnYG0RuJrIXAzn
6TAfBgNVHSMEGDAWgBTt5IfQJ8RQ5oQ698z36zpJ/FJOITAzBgNVHSAELDAqMAwG
CmCGSAFlAwIBDAEwDAYKYIZIAWUDAgEMAjAMBgpghkgBZQMCAQwDMIHABgNVHR8E
gbgwgbUwLKAqoCiGJmh0dHA6Ly9jcmwuZGlzYS5taWwvY3JsL0VDQVJPT1RDQTIu
Y3JsMIGEoIGBoH+GfWxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNkRUNBJTIw
Um9vdCUyMENBJTIwMiUyY291JTNkRUNBJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50
JTJjYyUzZFVTP2NlcnRpZmljYXRlUmV2b2NhdGlvbkxpc3Q7YmluYXJ5MIHTBggr
BgEFBQcBAQSBxjCBwzA6BggrBgEFBQcwAoYuaHR0cDovL2NybC5kaXNhLm1pbC9p
c3N1ZWR0by9FQ0FST09UQ0EyX0lULnA3YzCBhAYIKwYBBQUHMAKGeGxkYXA6Ly9j
cmwuZ2RzLmRpc2EubWlsL2NuJTNkRUNBJTIwUm9vdCUyMENBJTIwMiUyY291JTNk
RUNBJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJjYyUzZFVTP2Nyb3NzQ2VydGlm
aWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUFAAOCAQEACASvDJGPQAneNIKJ
Q+AHY3K/FCEqGN9w1dJlYJAHsWbsPP5ns3A2YO0JRPOvWOMtYX6tpB7/bBGpWodF
Eg//jd+DhjgNzONb2XYCCFInCywjSbD5W8crAxJ999FXlWRRRose0XMEwUqb5Toj
whh9dEOWK+lviMM6yNU7gxsQTgDqpP7jFCTIq+7lsmE05QyGZkf7pZ8spL6rhkNA
fxFRg80XEHoxLmxAU8/53vCiDyCsCwkPezdJkAiYpZY5pgkrz3vkGMwYr8tYsCew
UCd4pMIfXkR8Qewo3Ir6WEwBMfQG6BJ7Lx46Kk4NFetd82lBwGk2jFOCf2xHSlgF
99ZeIw==
-----END CERTIFICATE-----
```

### 12.11.35 all-ca-certs/ECA-ORC-SW4.crt

```
-----BEGIN CERTIFICATE-----
MIIFcDCCBFigAwIBAgIBDzANBgkqhkiG9w0BAQUFADBNMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNFQ0ExFjAUBgNVBAMT
DUVDQSBSb290IENBIDIwHhcNMTEwNjAxMTM0MzMzWhcNMTcwNTMwMTM0MzMzWjBw
MQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQL
EwNFQ0ExIjAgBgNVBAsTGUNlcnRpZmljYXRpb24gQXV0aG9yaXRpZXMxFTATBgNV
BAMTDE9SQyBFQ0EgU1cgNDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
AMyn/MU0Al+NhqANfvTXFGkrpCNFruuG1HZT8IgTWlNIBHUEg1+xg0e3b5uLRIfh
LBnrVfD2EyoIwE/LTkkml56sTPTGkNuSoPPY0OoGbTavB1xEWo5ZCfw5/cAskqik
AXplKR4XWPsoUIpCUie0AjIn9z5MfJkkkPQ2zhJSuZCYGyberSQSXTqVPswcs9O/
kteQH7k9rKEAlRYRer+JQSEsMGy1NoPU0Y6V4gyy/eLVfTVqYH0bLA3a/+QqV8a4
ZCkUBaLRBpsLiEx9SMzbXtsZBLT+/VVXXXMGlGUQTMfTMmBBANdZDL5Xu9Fstq/Z
srehbC81MkaFvYJYdqHsWuUCAwEAAaOCAjYwggIyMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDgYDVR0PAQH/BAQDAgGGMB0GA1UdDgQWBBRCnPS6b23pMNc61kb0sjTundJP
JjAfBgNVHSMEGDAWgBTt5IfQJ8RQ5oQ698z36zpJ/FJOITAzBgNVHSAELDAqMAwG
CmCGSAFlAwIBDAEwDAYKYIZIAWUDAgEMAjAMBgpghkgBZQMCAQwDMIHABgNVHR8E
gbgwgbUwLKAqoCiGJmh0dHA6Ly9jcmwuZGlzYS5taWwvY3JsL0VDQVJPT1RDQTIu
Y3JsMIGEoIGBoH+GfWxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNkRUNBJTIw
Um9vdCUyMENBJTIwMiUyY291JTNkRUNBJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50
JTJjYyUzZFVTP2NlcnRpZmljYXRlUmV2b2NhdGlvbkxpc3Q7YmluYXJ5MIHTBggr
BgEFBQcBAQSBxjCBwzA6BggrBgEFBQcwAoYuaHR0cDovL2NybC5kaXNhLm1pbC9p
c3N1ZWR0by9FQ0FST09UQ0EyX0lULnA3YzCBhAYIKwYBBQUHMAKGeGxkYXA6Ly9j
cmwuZ2RzLmRpc2EubWlsL2NuJTNkRUNBJTIwUm9vdCUyMENBJTIwMiUyY291JTNk
RUNBJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJjYyUzZFVTP2Nyb3NzQ2VydGlm
aWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUFAAOCAQEAg/gzrTwgRkl1cHJJ
4eOWCJO2xBOr3GjqJzL0Vr/NolqD5KgaK1WiGTbokBfjhz5axN06a0eoJE4UzBEP
Pc5BrlAEu3n48ZuxmEv6zUvhcuHr73rUAtnEyLzyOIhxHvW4GdmqbdaciaZ/R5uc
rg3w3xkltB+dxuNmU44+jk25WESLbYyrwsdl3pQyX3F1JUBwcFXQX6wQE9jpLw7C
m1PPv5e6yScpKRU+2EkQRiekemSlwFV70djYzjbUTwxJh5dnG4q8SM0wxGTamQfy
```

```
U5ZTW4qw0KMdBi8rsYm2m0Wlzlops4iAj+NKtKuqNzJtmt4PvqVvW9nyVxseycb6
TbYIIA==
-----END CERTIFICATE-----
```

### 12.11.36 all-ca-certs/ECA-Root.crt

```
-----BEGIN CERTIFICATE-----
MIICmDCCAgGgAwIBAgIBDjANBgkqhkiG9w0BAQUFADBLMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNFQ0ExFDASBgNVBAMT
C0VDQSBSb290IENBMB4XDTA0MDYxNDEwMjAwOVoXDTQwMDYxNDEwMjAwOVowSzEL
MAkGA1UEBhMCVVMxGDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMD
RUNBMRQwEgYDVQQDEwtFQ0EgUm9vdCBDQTCBnzANBgkqhkiG9w0BAQEFAAOBjQAw
gYkCgYEArkr2eXIS6oAKIpDkOlcQZdMGdncoygCEIU+ktqY3of5SVVXU7/it7kJ1
EUzR4ii2vthQtbww9aAnpQxcEmXZk8eEyiGEPy+cCQMllBY+efOtKgjbQNDZ3lB9
19qzUJwBl2BMxslU1XsJQw9SK10lPbQm4asa8E8e5zTUknZBWnECAwEAAaOBizCB
iDAfBgNVHSMEGDAWgBT2uAQnDlYW2blj2f2hVGVBoAhILzAdBgNVHQ4EFgQU9rgE
Jw5WFtm5Y9n9oVRlQaAISC8wDgYDVR0PAQH/BAQDAgGGMA8GA1UdEwEB/wQFMAMB
Af8wJQYDVR0gBB4wHDAMBgpghkgBZQMCAQwBMAwGCmCGSAFlAwIBDAIwDQYJKoZI
hvcNAQEFBQADgYEAHh0EQY2cZ209aBb5q0wW1ER0dc40GzsLyqjHfaQ4TEaMmUwL
AJRta/c4KVWLiwbODsvgJk+CaWmSL03gRW/ciVb/qDV7qh9Pyd1cOlanZTAnPog2
i82yL3i2fK9DCC84uoxEQbgqK2jx9bIjFTwlAqITk9fGAm5mdT84IEwq1Gw=
-----END CERTIFICATE-----
```

### 12.11.37 all-ca-certs/ECA-Root2.crt

```
-----BEGIN CERTIFICATE-----
MIIEOjCCAyKgAwIBAgIBBTANBgkqhkiG9w0BAQUFADBNMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNFQ0ExFjAUBgNVBAMT
DUVDQSBSb290IENBIDIwHhcNMDgwNDA0MTQyNDQ5WhcNMjgwMzMwMTQyNDQ5WjBN
MQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQL
EwNFQ0ExFjAUBgNVBAMTDUVDQSBSb290IENBIDIwggEiMA0GCSqGSIb3DQEBAQUA
A4IBDwAwggEKAoIBAQCzkNge75rtbkexRmFGjaNWCJxKMsdF8tKkpGa+fj2XgO/p
12vp4MA7jeL4EvoTNaTNWoaLsNAvNIz3Be2Hb6cwF9AWxlhGvZX5VhqI1E4eflgf
G6JzFZ1eiwYkE3Y8n2BirbC19OAkwVP32Rw3HeY88PrT8fmhmPE3hdjPAmDwoDiU
S1EoCcIX/ECkYXkOtIwbIIJi9eSa1imKNvofgcChETFL/lb92L9RkydZ2vWDPkSk
39UgZ+ufAKf73jG2YzEcPfccbAmNf/S6nKh8HwHGgG0BAdg3qP4tfSvCy9bpnLqU
jXYW3syBdQR7QXcVZRXJrAC2v+sc3LmNZC3/NdvFAgMBAAGjggEjMIIBHzAdBgNV
HQ4EFgQU7eSH0CfEU0aE0vfM9+s6SfxSTiEwDgYDVR0PAQH/BAQDAgGGMA8GA1Ud
EwEB/wQFMAMBAf8wgdwGCCsGAQUFBwELBIHPMIHMMEMGCCsGAQUFBzAFhjdodHRw
Oi8vY3JsLmdkcy5kaXNhLm1pbC9nZXRJc3N1ZWRCeT9FQ0ElMjBSb290JTIwQ0El
MjAyMIGEBggrBgEFBQcwBYZ4bGRhcDovL2NybC5nZHMuZGlzYS5taWwvY24lM2RF
Q0ElMjBSb290JTIwQ0ElMjAyJTJjb3UlM2RFQ0ElMmNvJTNkVS5TLiUyMEdvdmVy
bm1lbnQlMmNjJTNkVVM/Y3Jvc3NDZXJ0aWZpY2F0ZVBhaXI7YmluYXJ5MA0GCSqG
SIb3DQEBBQUAA4IBAQBKzBvnhYiTI9m5vs+68TWUwUs8E3Bdgy5OfsFWDy3Bfh8N
6d3+Apl8YOOGqrmGI8MXfscRD+PQQlajHPNwj7GIkbW8DoklZsg/XzYqDj62Z0CM
PUDfp2LgOBj6JG0tI9PAuHRelURUeiGXD1rNw/yVPSIiFd0YYF2FN6lN94H8vUqm
WIxPv8dlof/i0zRIGhcBMvKE894FsmRJmDfNIfWogVtkxdp8WLBhYR+twK3tNLaU
q8ec7GWXKYWUlDPENSSLPc38Nyq7+qS2G5DylVcXSODvqZ2tAOKxZNlneRGDuZro
CzI2ziBn43ptS+peU+UvMEHd+OVUnanXacZ12UVH
-----END CERTIFICATE-----
```

### 12.11.38 all-ca-certs/ECA-Verisign-G3.crt

```
-----BEGIN CERTIFICATE-----
MIIFxTCCBK2gAwIBAgIBEDANBgkqhkiG9w0BAQUFADBNMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNFQ0ExFjAUBgNVBAMT
```

```
DUVDQSBSb290IENBIDIwHhcNMTEwNzA2MTQwNTM5WhcNMTcwNzA0MTQwNTM5WjCB
mTELMAkGA1UEBhMCVVMxGDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UE
CxMDRUNBMSIwIAYDVQQLExlDZXJ0aWZpY2F0aW9uIEF1dGhvcml0aWVzMT4wPAYD
VQQDEzVWZXJpU2lnbiBDbGllbnQgRXh0ZXJuYWwgQ2VydGlmaWNhdGlvbiBBdXRo
b3JpdHkgLSBHMzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANRyuJwg
XDpXzi7VcxXeaUF5O5ALhmySkeK+fQ3nr7DXYphmssB6VA3XARUzymUUlbV9nrlO
4dChWYPibWlshcTDDuNnNyvxuO6eC+K3Mvx54YUjOPDYqcIXmOESAP5fM7KOh+OP
T+BHNBrkOO+WlE2DFcfOBCfBIKrIhTNgNEq76kiu7uPHvbSTpt8t/a328n5EKICz
hYgA98766RE6gPmNMLd+AobcWTqCwJvjQcA+HzoVjuvAD5gWOAfKURxMZQ2MPe9d
pH+gdJNF7At2qpkZiUDAhosK+PKiMAeF4bJFW5zp1fS84Nbr9SbfbqBaT1ShtAt4
IQN3Qt4XPalq/jMCAwEAAaOCAmEwggJdMBIGA1UdEwEB/wQIMAYBAf8CAQAwDgYD
VR0PAQH/BAQDAgGGMCkGA1UdEQQiMCCkHjAcMRowGAYDVQQDExFWZXJpU2lnbk1Q
S0ktMi02OTAdBgNVHQ4EFgQUsx1ZPOnebXvHtuZh8DB6Mw9QZuQwHwYDVR0jBBgw
FoAU7eSHOCfEUOaEOvfM9+s6SfxSTiEwMwYDVR0gBCwwKjAMBgpghkgBZQMCAQwB
MAwGCmCGSAFlAwIBDAIwDAYKYIZIAWUDAgEMAzCBwAYDVR0fBIG4MIG1MCygKqAo
hiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9FQ0FST09UQ0EyLmNybDCBhKCBgaB/
hn1sZGFwOi8vY3JsLmdkcy5kaXNhLm1pbC9jbiUzZEVDQSUyMFJvb3QlMjBDQSUy
MDIlMmNvdSUzZEVDQSUyY28lM2RVLlMuJTIwR292ZXJubWVudCUyY2MlM2RVUz9j
ZXJ0aWZpY2F0ZVJldm9jYXRpb25MaXN0O2JpbmFyeTCB0wYIKwYBBQUHAQEEgcYw
gcMwOgYIKwYBBQUHMAKGLmh0dHA6Ly9jcmwuZGlzYS5taWwvaXNzdWVkdG8vRUNB
Uk9PVENBMl9JVC5wN2MwgYQGCCsGAQUFBzAChnhsZGFwOi8vY3JsLmdkcy5kaXNh
Lm1pbC9jbiUzZEVDQSUyMFJvb3QlMjBDQSUyMDIlMmNvdSUzZEVDQSUyY28lM2RV
LlMuJTIwR292ZXJubWVudCUyY2MlM2RVUz9jcm9zc0NlcnRpZmljYXRlUGFpcjti
aW5hcnkwDQYJKoZIhvcNAQEFBQADggEBAHXwkkVTaa4/bkOyBGXf3d68nGbg+OKN
6vFIGmXgp2WAybRuYgws0Xh8O+tHlMik8ve08uxsna8l6WDleDyQbS+TJXVeyVFK
VfGAaPpl+ed5VcVdL/StIyLL1x4a4w/qCNJkSlUf9Nkn5mr6Yd4OQNeqe4LUrebs
L1441z8jClB7Rf+GTZAyoWoC72+4XuaDXY+uNnol5/Zr6dlxpegLpp2ADsLWukY1
UVwwiYDRZDjclMSy+hzG/sneei/CEkTOkeMNs/KwxuaCv+9MZ9+3432kOXE/O5cw
cqankd+BYyZU/BuT4GGU3jHNlK0LkxKBA+fItE9zM966q1AM4j9K7cU=
-----END CERTIFICATE-----
```

### 12.11.39 all-ca-certs/reflow-cert.py

```
import re
import unittest
from cStringIO import StringIO
import getopt
import sys

def reflow_cert(iterable, width):
    in_cert = False
    for line in iterable:
        if re.match('^-+BEGIN [A-Z ]+-+$', line):
            in_cert = True
            cert_lines = []
            yield line
        elif re.match('^-+END [A-Z ]+-+$', line):
            in_cert = False
            # regurgitate reflowed cert
            cert = ''.join(cert_lines)
            flowed_cert_lines = []
            while cert != '':
                flowed_cert_lines.append(cert[0:width])
                cert = cert[width:]
            for x in flowed_cert_lines:
                yield x + '\n'
            # now put out the END line
            yield line
        elif in_cert:
            cert_lines.append(line.strip())
        else:
            yield line


class TestReflowCert(unittest.TestCase):
    def testLongLine(self):
        """Certs with long lines, when reflowed, have shorter lines."""
        cert = """\
---BEGIN CERTIFICATE---
weofijwf90239fhj20vmqf84fums9p8vhsmvp9mhap98w4ctapwmt8cjamwpt\
48hmp349tc8ha3mp4t9c8hamtp948chamw9pt4cahw8mt4p98chm34p98cham\
p948chma3p498chma34p9c8hm3ap9f8ch4m3p98
---END CERTIFICATE---
"""
        self.assertEqual(’’.join(reflow_cert(StringIO(cert), 32)), """\
---BEGIN CERTIFICATE---
weofijwf90239fhj20vmqf84fums9p8v
hsmvp9mhap98w4ctapwmt8cjamwpt48h
mp349tc8ha3mp4t9c8hamtp948chamw9
pt4cahw8mt4p98chm34p98champ948ch
ma3p498chma34p9c8hm3ap9f8ch4m3p9
8
---END CERTIFICATE---
""")

    def testPreamble(self):
        """Reflowing a cert leaves non-certificate parts alone."""
        cert = """\
A big long description, longer than 32 characters.
---BEGIN CERTIFICATE---
weofijwf90239fhj20vmqf84fums9p8vhsmvp9mhap98w4ctapwmt8cjamwpt\
48hmp349tc8ha3mp4t9c8hamtp948chamw9pt4cahw8mt4p98chm34p98cham\
p948chma3p498chma34p9c8hm3ap9f8ch4m3p98
---END CERTIFICATE---
A big long postamble, longer than 32 characters.
"""
        self.assertEqual(’’.join(reflow_cert(StringIO(cert), 32)), """\
A big long description, longer than 32 characters.
---BEGIN CERTIFICATE---
weofijwf90239fhj20vmqf84fums9p8v
hsmvp9mhap98w4ctapwmt8cjamwpt48h
mp349tc8ha3mp4t9c8hamtp948chamw9
pt4cahw8mt4p98chm34p98champ948ch
ma3p498chma34p9c8hm3ap9f8ch4m3p9
8
---END CERTIFICATE---
A big long postamble, longer than 32 characters.
""")

    def testCertChain(self):
        """Reflowing works for files containing multiple certs."""
        cert = """\
A big long description, longer than 32 characters.
---BEGIN CERTIFICATE---
```

```
weofijwf90239fhj20vmqf84fums9p8vhsmvp9mhap98w4ctapwmt8cjamwpt\
48hmp349tc8ha3mp4t9c8hamtp948chamw9pt4cahw8mt4p98chm34p98cham\
p948chma3p498chma34p9c8hm3ap9f8ch4m3p98
---END CERTIFICATE---
A big long postamble, longer than 32 characters.
Some other stuff.
---BEGIN CERTIFICATE---
WEOFIJWF90239FHJ20VMQF84FUMS9P8VHSMVP9MHAP98W4CTAPWMT8CJAMWPT\
48HMP349TC8HA3MP4T9C8HAMTP948CHAMW9PT4CAHW8MT4P98CHM34P98CHAM\
P948CHMA3P498CHMA34P9C8HM3AP9F8CH4M3P98
---END CERTIFICATE---
"""
        self.assertEqual(’’.join(reflow_cert(StringIO(cert), 32)), """\
A big long description, longer than 32 characters.
---BEGIN CERTIFICATE---
weofijwf90239fhj20vmqf84fums9p8v
hsmvp9mhap98w4ctapwmt8cjamwpt48h
mp349tc8ha3mp4t9c8hamtp948chamw9
pt4cahw8mt4p98chm34p98champ948ch
ma3p498chma34p9c8hm3ap9f8ch4m3p9
8
---END CERTIFICATE---
A big long postamble, longer than 32 characters.
Some other stuff.
---BEGIN CERTIFICATE---
WEOFIJWF90239FHJ20VMQF84FUMS9P8V
HSMVP9MHAP98W4CTAPWMT8CJAMWPT48H
MP349TC8HA3MP4T9C8HAMTP948CHAMW9
PT4CAHW8MT4P98CHM34P98CHAMP948CH
MA3P498CHMA34P9C8HM3AP9F8CH4M3P9
8
---END CERTIFICATE---
""")

def usage(progname):
    print >> sys.stderr, """\
Usage: %s [--help] [--test] [-w n] certificate.pem

--help: Show this message.
--test: Run unit tests instead of reflowing certificates.
 -w n : Reflow certificate to a line width of n characters.

Input file must be a PEM-encoded object. Lines which are part of the
object are reflowed; other lines (e.g. descriptions) are not.

Output is stdout.
""" % progname

if __name__ == ’__main__’:
    try:
        ovs, remaining = getopt.getopt(sys.argv[1:], ’w:’,[’test’, ’help’])
    except getopt.GetoptError, e:
        print >> sys.stderr, e
        usage(sys.argv[0])
        sys.exit(1)
    testInstead = False
    width = 64
    for o, v in ovs:
        if o == '--test':
            testInstead = True
        elif o == '-w':
            width = int(v)
        elif o == '--help':
            usage(sys.argv[0])
            sys.exit(1)
    if testInstead:
        # Forget about the args we've already parsed; they won't be
        # useful to unittest. Whatever args unittest could have used, we
        # would not have recognized above,
        sys.argv = sys.argv[0:1]
        unittest.main()
    if len(remaining) != 1:
        # no files to process!
        print >> sys.stderr, "no filename given"
        usage(sys.argv[0])
        sys.exit(1)
    else:
        for line in reflow_cert(file(remaining[0]), width):
            sys.stdout.write(line)
```

### 12.11.40 get_crl/refresh_crls.py

```
#!/usr/bin/python
# CMITS - Configuration Management for Information Technology Systems
# Based on <https://github.com/afseo/cmits>.
# Copyright 2015 Jared Jennings <mailto:jjennings@fastmail.fm>.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#    http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
# Read some CA certificates. Fetch their corresponding CRLs, using HTTP
# or LDAP.
#
# Works with Python 2.4 or later.

import os
from sys import stderr
import ldap
import sys
import base64
from subprocess import Popen, PIPE
from time import strptime, time, gmtime
from datetime import datetime, timedelta
import glob
from urllib import quote, unquote
from urllib2 import urlopen, HTTPError

class OpenSSLExitedWithError(Exception): pass
class UnexpectedOpenSSLResponse(Exception): pass

class CACertParseError(Exception): pass

class CRLFetchError(Exception): pass
class NoCRLDistributionPoints(CRLFetchError): pass
class DontKnowHowToFetch(CRLFetchError): pass
class AllFetchAttemptsFailed(CRLFetchError): pass
class UnexpectedMimeType(CRLFetchError): pass
class NoSuchObjectOnServer(CRLFetchError): pass
class UnexpectedLDAPResponse(CRLFetchError): pass
class ServerDown(CRLFetchError): pass
class CACertExpired(CRLFetchError): pass

# Installation and upgrade are easier if we just depend on the OpenSSL
# binary instead of requiring non-stock Python libraries such as pyasn1
def openssl(*args):
    cmdline = ('openssl',) + args
    p = Popen(cmdline, stdin=None, stdout=PIPE, stderr=PIPE)
    output, errors = p.communicate()
    # apparently, crl -noout makes openssl return with exitcode 1
    # apparently, that's fixed in RHEL6
    if p.returncode != 0:
        raise OpenSSLExitedWithError('command: %r' % (cmdline,),
                'output: %r' % output, 'errors: %r' % errors,
                'exit code: %d' % p.returncode)
    for line in output.strip().split('\n'):
        yield line

openssl_crl = openssl

class CACert(object):
    def __init__(self, filename):
        self.filename = filename
        self.dn = None
        self.cn = None
        for thisline in openssl('x509', '-subject', '-noout', '-in', filena[WRAP]
me):
            # [1:]: the dn starts with '/'. get rid of the empty first
            # element after splitting
            try:
                self.dn = thisline[len('subject='):].strip().split('/')[1:]
            except Exception, e:
                raise CACertParseError(e)
        try:
            self.cn = [x[len('cn='):] for x in self.dn if
                    x.lower().startswith('cn=')][0]
        except IndexError:
            raise CACertParseError('missing Common Name')
        notbefore = None
        notafter = None
        for thisline in openssl('x509', '-dates', '-noout', '-in', filename[WRAP]
):
            if thisline.startswith('notBefore'):
                nbs = thisline[len('notBefore='):].strip()
                notbefore = strptime(nbs, "%b %d %H:%M:%S %Y %Z")
            elif thisline.startswith('notAfter'):
                nas = thisline[len('notAfter='):].strip()
                notafter  = strptime(nas, "%b %d %H:%M:%S %Y %Z")
        if notbefore is not None and notafter is not None:
            self.validityPeriod = (notbefore, notafter)

    def __repr__(self):
        return '<CACert with subject %s>' % ('/' + '/'.join(self.dn))

    def __str__(self):
        return '/' + '/'.join(self.dn)

    def getSources(self):
        # I thought I could use the X509v3 extension
        # cRLDistributionPoints to find out where to get CRLs for a
        # given CA, making the CACert object the authority on where to
        # get CRLs. But it appears that, say, for CA-22, that extension
        # indicates where to get the CRL which may say, "The CA-22
        # certificate is revoked" -- not where to get the CRL which may
        # say, "These certificates, signed by CA-22, are revoked."
        #
        # So, nothing in the CA certificate will help us find the
        # corresponding CRL, and we have to know a place where we can
        # get it. We don't know that for all certificates, so we'll have
        # to make sure a CA cert is familiar before claiming to know CRL
        # sources corresponding to it. Let's find out what we've got.
        # - maybe the DN starts from the most general. but we need it to
        # be the other way around.
        dnPieces = list(self.dn)
        if not dnPieces[0].lower().startswith('cn'):
            dnPieces.reverse()
        if [x.lower() for x in dnPieces[-3:]] in \
                [['ou=dod', 'o=u.s. government', 'c=us'],
                 ['ou=eca', 'o=u.s. government', 'c=us']]:
            escaped_cn = quote(self.cn)
            http = ( 'http://crl.gds.disa.mil/getcrl?' + escaped_cn,
                     'http://crl.disa.mil/getcrl?' + escaped_cn )
            dn = ', '.join(dnPieces)
            escaped_dn = quote(dn)
            return http
        else:
            # we know no sources for the CRL.
            return []

    def isValid(self):
        if self.validityPeriod is None:
            return True
        else:
            now = gmtime()
            validityStarts, validityEnds = self.validityPeriod
            return ((now > validityStarts) and (now < validityEnds))

class CRL(object):
    mime_type = 'application/pkix-crl'
    def __init__(self, cacert, dir, getLDAPConnection):
        self.cacert = cacert
        stem = '.'.join(os.path.basename(cacert.filename).split('.')[:-1])
        self.filename = os.path.join(dir, stem + '.crl')
        self.getLDAPConnection = getLDAPConnection

    def isExpired(self):
        if not os.path.exists(self.filename):
            return True
        g = openssl_crl('crl', '-in', self.filename, '-noout',
                '-nextupdate')
        firstLine = g.next()
        # parse output
        try:
            expireDateString = firstLine.split('=')[1].strip()
            expireTuple = strptime(expireDateString, '%b %d %H:%M:%S %Y %Z'[WRAP]
)
            # The added (0,) is the number of microseconds.
            expireDatetime = datetime(*(expireTuple[0:6]+(0,)))
            tomorrow = datetime.utcnow() + timedelta(1)
            return tomorrow > expireDatetime
        except:
            raise UnexpectedOpenSSLResponse(firstLine)

    def fetchIfNecessary(self):
        if self.isExpired():
            print >> stderr, "Fetching CRL for: %s" % self.cacert
            a = time()
            newCRLData = pemEncode(self.fetch())
            # write out file, then atomically move into place
            newName = self.filename + '.new'
            newFile = file(newName, 'w')
            newFile.write(newCRLData)
            newFile.close()
            os.rename(newName, self.filename)
            b = time()
            elapsed = int(b-a)
            print >> stderr, "Fetch complete after %d seconds." % elapsed

    def fetch_ldap(self, url):
        # we expect url to be something like 'ldap://server/dn?bla;bla'.
        # we want 'ldap://server', 'dn' and 'bla;bla'.
        # the split by slashes would look like ['ldap:', '', 'server',
        # 'dn?bla;bla'].
        serverURL = '/'.join(url.split('/')[:3])
        dn, attribute = '/'.join(url.split('/')[3:]).split('?')
        # The URL has all funny characters escaped. We need to pass
        # those as-is
        dn = unquote(dn)

        l = self.getLDAPConnection(serverURL)
        try:
            result = l.search_s(dn, ldap.SCOPE_SUBTREE,
                    attrlist=[attribute])
        except ldap.NO_SUCH_OBJECT:
            raise NoSuchObjectOnServer(dn, serverURL)
        except ldap.SERVER_DOWN:
            raise ServerDown(serverURL)
        # the CRL is inside some data structures inside result. if the serv[WRAP]
er
        # returns something empty or unexpected this will raise an exceptio[WRAP]
n.
        try:
            crl = result[0][1][attribute][0]
            return crl
        except:
            raise UnexpectedLDAPResponse(url, result)

    def fetch_http(self, url):
        try:
            u = urlopen(url)
        except HTTPError, e:
            raise CRLFetchError(e)
        t = u.info().type
        if t != self.mime_type:
            raise UnexpectedMimeType(url, t)
        return u.read()

    def fetch(self):
        if not self.cacert.isValid():
            raise CACertExpired(self.cacert)
        urls = list(self.cacert.getSources())
        if len(urls) == 0:
            raise NoCRLDistributionPoints(self.cacert)
        # 'http' comes before 'ldap' alphabetically. take advantage
        urls.sort()
        succeededYet = False
        crl = None
        for url in urls:
            if not succeededYet:
                scheme, dontcare = url.split(':',1)
                try:
                    fetcher = getattr(self, 'fetch_' + scheme)
                except AttributeError:
                    raise DontKnowHowToFetch(url)
                try:
                    print "    using %r" % url
                    crl = fetcher(url)
                    succeededYet = True
                except CRLFetchError, e:
                    print "    exception %s: %s" %\
                            (e.__class__.__name__,
                             str(e))

        if succeededYet:
            return crl
        else:
            raise AllFetchAttemptsFailed(self)

    def __repr__(self):
        return "<CRL for %r>" % self.cacert

    # i don't know why i bothered
    def __str__(self):
        return "CRL for %s" % self.cacert


def pemEncode(binary, objectType = "X509 CRL"):
    """PEM-encode some binary data. binary is the data; objectType is what [WRAP]
sort
    of thing it is. For example, a CRL's objectType is "X509 CRL". The
    objectType goes into the -----BEGIN something----- and -----END
    something----- lines at the beginning and end of the PEM file. Some oth[WRAP]
er
    possible values for objectType are "CERTIFICATE", "CERTIFICATE REQUEST"[WRAP]
,
    "RSA PRIVATE KEY", "DSA PRIVATE KEY"."""

    intro = "-----BEGIN %s-----\n" % objectType
    outro = "-----END %s-----\n" % objectType

    content = base64.encodestring(binary)
    return intro + content + outro


class LDAPConnectionPool(object):
    def __init__(self):
        self.pool = {}

    def __call__(self, url):
        if self.pool.has_key(url):
            return self.pool[url]
        else:
            t1 = time()
            l = ldap.initialize(url)
            l.protocol_version = ldap.VERSION3
            # no DN, no password: anonymous
            l.simple_bind_s()
            t2 = time()
            if (t2 - t1) > 10:
                print >> stderr, "Connect to %s took %d seconds" % \
                        (url, t2-t1)
            self.pool[url] = l
            return l

    def close(self):
        for k,v in self.pool.items():
            try:
                v.unbind_s()
            except Exception, e:
                print >> stderr, "While unbinding %s: %r" % (k,e)

def usage():
    prog = sys.argv[0]
    print >> sys.stderr, """\
usage: %(prog)s /dir/with/CA/certs /dir/for/CRLs

Check Certificate Revocation Lists (CRLs) in /dir/for/CRLs, which relate
to the Certification Authorities (CAs) whose CA certs are in
/dir/with/CA/certs. If any are expired, fetch new ones.
```

```
CA certs are expected to be files in PEM format whose names end with
'.crt'.

""" % locals()

if __name__ == '__main__':
    if len(sys.argv) != 3:
        usage()
        sys.exit(1)
    caCertDir, destination = sys.argv[1:]
    if not os.path.isdir(caCertDir):
        print >> sys.stderr, \
                "Given CA certificate dir %s is not a directory" % caCertDi[WRAP]
r
        sys.exit(2)
    if not os.path.isdir(destination):
        print >> sys.stderr, \
                "CRL destination dir %s is not a directory" % destination
        sys.exit(3)
    pool = LDAPConnectionPool()
    for f in glob.glob(os.path.join(caCertDir, '*.crt')):
        if 'Makefile' not in f:
            c = CACert(f)
            r = CRL(c, destination, pool)
            try:
                r.fetchIfNecessary()
            except KeyboardInterrupt:
                raise
            except CRLFetchError, e:
                print "Fetch failed: %s %s" %\
                        (e.__class__.__name__,
                         str(e))
```

## 12.11.41 get_crl/refresh_crls_nss.py

```
#!/usr/bin/python
# CMITS - Configuration Management for Information Technology Systems
# Based on <https://github.com/afseo/cmits>.
# Copyright 2015 Jared Jennings <mailto:jjennings@fastmail.fm>.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#    http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
# Read some CA certificates. Fetch their corresponding CRLs, using HTTP
# or LDAP.
#
# Works with Python 2.4 or later.
```

```
import os
import re
import sys
import logging
import getopt
from subprocess import Popen, PIPE
from time import time
from datetime import datetime, timedelta
from urllib import quote, unquote
from urllib2 import urlopen, HTTPError
from tempfile import NamedTemporaryFile

class NSSUtilExitedWithError(Exception): pass
class UnexpectedNSSUtilResponse(Exception): pass

class CACertParseError(Exception): pass

class CRLFetchError(Exception): pass
class NoCRLDistributionPoints(CRLFetchError): pass
class DontKnowHowToFetch(CRLFetchError): pass
class AllFetchAttemptsFailed(CRLFetchError): pass
class UnexpectedMimeType(CRLFetchError): pass
class NoSuchObjectOnServer(CRLFetchError): pass
class UnexpectedLDAPResponse(CRLFetchError): pass
class ServerDown(CRLFetchError): pass
class CACertExpired(CRLFetchError): pass

# We use the NSS utilities because PyNSS doesn’t appear to do all this
# stuff.

# This is the format in which certutil outputs validity dates. They
# appear to be in UTC.
NSS_TIME_FORMAT = ’%a %b %d %H:%M:%S %Y’


class NSSDB(object):
    def __init__(self, dbdir, pwfile, sqlite):
        self.dbdir = dbdir
        self.pwfile = pwfile
        self.sqlite = sqlite

    def _get_cmdline(self, command, *args):
        if self.sqlite:
            dbspec = ’sql:’ + self.dbdir
        else:
            dbspec = self.dbdir
        cmdline = (command, ’-f’, self.pwfile, ’-d’, dbspec) + args
        return cmdline

    def _util(self, command, *args):
        p = Popen(self._get_cmdline(command, *args),
                  stdin=None, stdout=PIPE, stderr=PIPE)
        output, errors = p.communicate()
        if p.returncode != 0:
            raise NSSUtilExitedWithError(’command: %r’ % (cmdline,),
                    ’output: %r’ % output, ’errors: %r’ % errors,
                    ’exit code: %d’ % p.returncode)
        for line in output.strip().split('\n'):
            yield line

    def _util_head(self, n, command, *args):
        p = Popen(self._get_cmdline(command, *args),
                  stdin=None, stdout=PIPE, stderr=PIPE)
        # assume stderr will not fill up
        try:
            for lineno in range(n):
                yield p.stdout.next().strip()
        except StopIteration:
            pass
        p.terminate()


    def _certList(self):
        lines = self._util('certutil', '-L')
        # skip header
        for i in range(3):
            lines.next()
        for line in lines:
            words = line.split()
            trustargs = words[-1]
            # certutil does not preserve leading or trailing spaces in
            # cert nicknames when listing them - so we can't, either.
    # there may be trailing spaces after the trustarg; strip them
            nickname = line.strip()[:-len(trustargs)].strip()
            yield (nickname, tuple(trustargs.split(',')))

    def _crlList(self):
        lines = self._util_head(14, 'crlutil', '-L')
        # skip header
        for i in range(4):
            lines.next()
        for line in lines:
            words = line.split()
            crltype = words[-1]
            # certutil does not preserve leading or trailing spaces in
            # cert nicknames when listing them - so we can't, either
            nickname = line[:-len(crltype)].strip()
            yield (nickname, crltype)

    # in haskell: nicknames = map fst. oh haskell i miss you now
    def certNicknames(self):
        for nickname, trustargs in self._certList():
            yield nickname

    def caCertNicknames(self):
        for nickname, trustargs in self._certList():
            for use in trustargs:
                # C: trusted to issue client certs; T: trusted to issue ser[WRAP]
ver
                # certs; c: valid CA; run certutil -H to find out more
                if 'u' not in use:
                    # it's not a user cert; we need a CRL for it
                    yield nickname
                    break
    def crlNicknames(self):
        for nickname, crltype in self._crlList():
            yield nickname


def _splitOnUnquotedCommasGenerator(s):
    # There may be commas in names. NSS utils deal with this by
    # double-quoting the names. So if we split on commas and have a
    # value with an odd number of double quotes in it, it isn't a whole
    # value. Accumulate more.
    value = None
    for x in s.split(','):
        if value is None:
            value = x
            if len(re.findall('"', value)) % 2 == 0:
                yield value
                value = None
        else:
            if len(re.findall('"', value)) % 2 == 0:
                yield value
                value = x
            else:
                value = value + ',' + x
    if value is not None:
        yield value

def splitOnUnquotedCommas(s):
    return list(_splitOnUnquotedCommasGenerator(s))


class CACert(object):

    def __init__(self, nssdb, nickname):
        self.db = nssdb
        self.nickname = nickname
        self.dn = self._getDn()
        self.cn = [x for x in self.dn if
                x.lower().startswith('cn=')][0][len('cn='):]

    def __repr__(self):
        return self.nickname

    def __str__(self):
        return ','.join(self.dn)

    def _getDn(self):
        lines = self.db._util('certutil', '-L', '-n', self.nickname)
        continuing = False
        column = 0
        value = ""
        s = 'Subject: '
        for line in lines:
            if continuing:
                value += line[column:]
                if line.endswith('"'):
                    break
            else:
                try:
                    column = line.index(s) + 4
                    value = line[line.index(s) + len(s):]
                    if line.endswith('"'):
                        break
                    else:
                        continuing = True
                except ValueError:
                    # substring not found
                    pass
        unquoted = value[1:-1]
        # there may be url encoding in there; we are not presently
        # dealing with it.
        return tuple(splitOnUnquotedCommas(unquoted))

    # returns a pair of UTC datetimes.
    def _getValidity(self):
        lines = self.db._util('certutil', '-L', '-n', self.nickname)
        expects = ['Validity:', 'Not Before: ', 'Not After : ']
        thisone = 0
        notbefore = None
        notafter = None
        for line in lines:
            s = line.strip()
            if s.startswith(expects[thisone]):
                value = s[len(expects[thisone]):]
                if thisone == 1:
                    notbefore = datetime.strptime(value,
                            NSS_TIME_FORMAT)
                elif thisone == 2:
                    notafter = datetime.strptime(value,
                            NSS_TIME_FORMAT)
                    break
                thisone += 1
        return (notbefore, notafter)

    def isValid(self):
        notbefore, notafter = self._getValidity()
        now = datetime.utcnow()
        return ((now >= notbefore) and (now <= notafter))

    def getCRLSources(self):
        # I thought I could use the X509v3 extension
        # cRLDistributionPoints to find out where to get CRLs for a
        # given CA, making the CACert object the authority on where to
        # get CRLs. But it appears that, say, for CA-22, that extension
        # indicates where to get the CRL which may say, "The CA-22
        # certificate is revoked" -- not where to get the CRL which may
        # say, "These certificates, signed by CA-22, are revoked."
        #
        # So, nothing in the CA certificate will help us find the
        # corresponding CRL, and we have to know a place where we can
        # get it. We don't know that for all certificates, so we'll have
        # to make sure a CA cert is familiar before claiming to know CRL
        # sources corresponding to it. Let's find out what we've got.
        # - maybe the DN starts from the most general. but we need it to
        # be the other way around.
        dnPieces = list(self.dn)
        if not dnPieces[0].lower().startswith('cn'):
            dnPieces.reverse()
        if [x.lower() for x in dnPieces[-3:]] in \
                [['ou=dod', 'o=u.s. government', 'c=us'],
                 ['ou=eca', 'o=u.s. government', 'c=us']]:
            escaped_cn = quote(self.cn)
            http = ( 'http://crl.gds.disa.mil/getcrl?' + escaped_cn,
                     'http://crl.disa.mil/getcrl?' + escaped_cn )
            dn = ', '.join(dnPieces)
            escaped_dn = quote(dn)
            return http
        elif [x.lower() for x in dnPieces[-3:]] in \
                [['ou=dod', 'o=gov', 'c=au']]:
            escaped_cn = quote(self.cn)
            http = ( 'http://www.defence.gov.au/pki/crl/%s.crl' % escaped_c[WRAP]
n, )
            return http
        else:
            # we know no sources for the CRL.
            return []


class CRL(object):
    mime_type = 'application/pkix-crl'

    def __init__(self, db, cacert):
        self.db = db
        self.cacert = cacert
        self.log = logging.getLogger(repr(self))

    def _getValidity(self):
        lines = self.db._util_head(10, 'crlutil', '-L', '-n', self.cacert.n[WRAP]
ickname)
        expects = ['This Update: ', 'Next Update: ']
        thisupdate = None
        nextupdate = None
        for line in lines:
            s = line.strip()
            if s.startswith('This Update: '):
                thisupdate = datetime.strptime(s[len('This Update: '):],
                        NSS_TIME_FORMAT)
            if s.startswith('Next Update: '):
                nextupdate = datetime.strptime(s[len('Next Update: '):],
                        NSS_TIME_FORMAT)
        return (thisupdate, nextupdate)

    def isExpired(self):
        if self.cacert.nickname not in self.db.crlNicknames():
            return True
        lastupdate, nextupdate = self._getValidity()
        tomorrow = datetime.utcnow() + timedelta(1)
        return tomorrow > nextupdate

    def fetchIfNecessary(self):
        if self.isExpired():
            self.log.info('fetching')
            a = time()
            newCRLData = self.fetch()
            # write out file, then atomically move into place
            newFile = NamedTemporaryFile()
            newName = newFile.name
            newFile.write(newCRLData)
            newFile.flush()
            newFile.seek(0)
            # list: we have to use up the output to make the generator's
            # code happen
            list(self.db._util('crlutil', '-I', '-i', newFile.name))
            newFile.close()
            b = time()
            elapsed = int(b-a)
            self.log.info('complete after %d seconds', elapsed)

    def fetch_http(self, url):
        try:
            u = urlopen(url)
        except HTTPError, e:
            raise CRLFetchError(e)
        t = u.info().type
        if t != self.mime_type:
            raise UnexpectedMimeType(url, t)
        return u.read()

    def fetch(self):
        if not self.cacert.isValid():
            raise CACertExpired(self.cacert)
        urls = list(self.cacert.getCRLSources())
        if len(urls) == 0:
            raise NoCRLDistributionPoints(self.cacert)
        # 'http' comes before 'ldap' alphabetically. take advantage
        urls.sort()
        succeededYet = False
        crl = None
        for url in urls:
            if not succeededYet:
                scheme, dontcare = url.split(':',1)
                try:
                    fetcher = getattr(self, 'fetch_' + scheme)
                except AttributeError:
                    raise DontKnowHowToFetch(url)
                try:
                    self.log.info('using %r', url)
                    crl = fetcher(url)
                    succeededYet = True
                except CRLFetchError, e:
                    self.log.exception('while fetching,')
        if succeededYet:
            return crl
        else:
            raise AllFetchAttemptsFailed(self)

    def __repr__(self):
        return "CRL for %r" % self.cacert
    # i don’t know why i bothered
    def __str__(self):
        return "CRL for %s" % self.cacert

def usage():
    prog = sys.argv[0]
    print >> sys.stderr, """\
usage: %(prog)s [-v] [-B] /nss/database/directory /path/to/passwordfile

Check Certificate Revocation Lists (CRLs) in the given NSS database,
which relate to the Certification Authorities (CAs) whose CA certs are
in the database. If any are expired or missing, fetch new ones. The
password file contains any passwords necessary to open the database, in
the form module:password. Modules of interest (don’t type the quotes)
are "internal", "NSS Certificate DB", and "NSS FIPS 140-2 Certificate
DB".

If -v is given, non-error fetching activity is shown.

The new format of NSS database (cert9.db, key4.db, SQLite) is used by
default. If -B is given, the old format (cert8.db, key3.db, Berkeley
DB) is used.

""" % locals()

if __name__ == ’__main__’:
    ovpairs, rest = getopt.getopt(sys.argv[1:], ’vB’)
    loglevel = logging.WARNING
    sqlite = True
    for o, v in ovpairs:
        if o == ’-v’:
            loglevel = logging.DEBUG
        if o == ’-B’:
            sqlite = False
    if len(rest) != 2:
        usage()
        sys.exit(1)
    dbdir, pwfile = rest
    logging.basicConfig(stream=sys.stderr, level=loglevel,
            format=’%(name)s: %(message)s’)
    toplog = logging.getLogger(’main’)
    db = NSSDB(dbdir, pwfile, sqlite)
    caCerts = [CACert(db, nick) for nick in db.caCertNicknames()]
    crls = [CRL(db, ca) for ca in caCerts]
    for crl in crls:
        try:
            crl.fetchIfNecessary()
        except KeyboardInterrupt:
            toplog.error("KeyboardInterrupt: quitting.")
            sys.exit(2)
        except CACertExpired, e:
            toplog.error(’CA cert %s has expired’, str(crl.cacert))
        except CRLFetchError, e:
            toplog.exception(’Fetch failed’)
        except Exception, e:
            # "Unexpected error."
            e.args = ('While fetching', crl,) + e.args
            raise
```

### 12.11.42 get_crl/test_refresh_crls.py

```
# CMITS - Configuration Management for Information Technology Systems
# Based on <https://github.com/afseo/cmits>.
# Copyright 2015 Jared Jennings <mailto:jjennings@fastmail.fm>.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#    http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
from refresh_crls import CACert, CRL, openssl
from refresh_crls import CACertExpired
import unittest
from tempfile import mkdtemp
from shutil import rmtree
import os
import time

class HasDir(unittest.TestCase):
    def setUp(self):
        self.dir = mkdtemp(prefix='fetchcrltest')
        self.oldcwd = os.getcwd()
        os.chdir(self.dir)
    def tearDown(self):
        os.chdir(self.oldcwd)
        #rmtree(self.dir)
        pass

class CACertBase(HasDir):
    def makeCert(self, dnElements=('C=US', 'O=Test', 'OU=Test',
            'CN=Flarble'), additionalConfig='',
            additionalSwitches=''):

        cnf = file('cnf', 'w')
        print >> cnf, """
[ req ]
default_bits = 2048
default_keyfile  = privkey.pem
distinguished_name = req_distinguished_name
x509_extensions = v3_ca # The extentions to add to the self signed cert
input_password = secret
output_password = secret
days=-1
prompt = no
%s
[ req_distinguished_name ]
%s
```

```
[ v3_ca ]
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid:always,issuer:always
basicConstraints = CA:true
""" % (additionalConfig, '\n'.join(dnElements))
        cnf.close()
        cert = '\n'.join(openssl('req', '-new', '-x509', '-config', 'cnf',
            *additionalSwitches.split()))
        cert_filename = 'cert'
        cfile = file(cert_filename, 'w')
        print >> cfile, cert
        cfile.close()
        time.sleep(1) # make sure we're after the not-valid-before time
        return cert_filename

class TestCACert(CACertBase):
    def testCN(self):
        c = CACert(self.makeCert())
        self.assertEqual(c.cn, 'Flarble')

    def testValid(self):
        c = CACert(self.makeCert())
        self.assert_(c.isValid())

    def testInvalid(self):
        c = CACert(self.makeCert(additionalSwitches='-days -1'))
        self.assert_(not c.isValid())
        crl = CRL(c, self.dir, None)
        self.assertRaises(CACertExpired, crl.fetch)

    def testDoDCRLSources(self):
        # see req(1), 'DISTINGUISHED NAME ... FORMAT' section, about the
        # 1.OU, 2.OU
        c = CACert(self.makeCert(['C=US', 'O=U.S. Government',
            '1.OU=DoD', '2.OU=PKI', 'CN=Unit Test CA']))
        self.assert_(len(c.getSources()) > 0)

if __name__ == '__main__':
    unittest.main()
```

## 12.11.43 get_crl/test_refresh_crls_nss.py

```
# CMITS - Configuration Management for Information Technology Systems
# Based on <https://github.com/afseo/cmits>.
# Copyright 2015 Jared Jennings <mailto:jjennings@fastmail.fm>.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#    http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
```

```
#
# The tests in this module deal with CA certificates and CRLs. Rather
# than create a CA and issue a CRL at test time, I've used a DoD CA
# certificate and a real CRL. This saves work in the short term, but
# it means that in a couple of years, the tests will start failing
# even though the code has not changed.

import unittest
import tempfile
import shutil
import os
import base64
import datetime
from subprocess import Popen, PIPE
from refresh_crls_nss import NSSDB, CACert, splitOnUnquotedCommas, CRL

class TestSplitOnUnquotedCommas(unittest.TestCase):
    def testSplitNoQuotes(self):
        self.assertEqual(splitOnUnquotedCommas('a,b,c,d,e'),
                ['a', 'b', 'c', 'd', 'e'])

    def testSplitWithQuotes(self):
        self.assertEqual(splitOnUnquotedCommas('a,"b,c",d,e'),
                ['a', '"b,c"', 'd', 'e'])

    def testSplitDNWithQuotes(self):
        self.assertEqual(splitOnUnquotedCommas(
            'CN="Bletch, Quux, Zart",OU="Foo, Bar, Baz",' \
            'O="Goo, Bar, Baz",L=fi,ST=gb,C=us'),
            ['CN="Bletch, Quux, Zart"', 'OU="Foo, Bar, Baz"',
                'O="Goo, Bar, Baz"', 'L=fi', 'ST=gb', 'C=us'])

class CommonDataForTest(object):
    certs = {
            'DoD-Root2-CA32': """\
-----BEGIN CERTIFICATE-----
MIIFTDCCBDSgAwIBAgICA6EwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTMwMjA0MjA0NDA1WhcN
MTkwMjA0MjA0NDA1WjBXMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTESMBAGA1UEAxMJRE9E
IENBLTMyMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAs+KVHZM2LSWl
Dv146e/qk9E6ydhXvRnf0cei0ejZ/dK0FajdvT5k9Lb+nAPfS7Blt6sEGDIZbBMB
UtHmtchBEre+O8tNQBCIyp62/TV3bSb2ZK0RhwypJXpYn7C9mPaTXxvv77KXrfgV
59zmoGp1DVHfVR1oQVJJLsecaFdWR4/e9lIugW9WvAaJEpSfI70/gceGAnUwXjOh
3OETu/15VgE8Shn0LOuQZGTX6AovUYbVCJuE+/npi0LKZdKQBxyCl4xEI1cGLHVp
KHCy7T5M1eOWdxX9upXPW5ZpAnfWgNmPhynj5wV2r8qNEmA0cseznThuTJYynpA1
rXWL0WJACQIDAQABo4ICHDCCAhgwHQYDVR0OBBYEFC/Kk1MDrG919Xb6vv6O6hCL
t+eQMB8GA1UdIwQYMBaAFEl0uwxeunr+AlTve6DGlcYJgHCWMBIGA1UdEwEB/wQI
MAYBAf8CAQAwDAYDVR0kBAUwA4ABADAOBgNVHQ8BAf8EBAMCAYYwZgYDVR0gBF8w
XTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETALBglghkgB
ZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUDAgEDGzA3
BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9ET0RST09U
Q0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5odHRwOi8v
Y3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAGCCsGAQUF
BzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNsZGFwOi8v
Y3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIlMmNvdSUz
```

```
ZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJjYyUzZFVT
P2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUFAAOCAQEA
MI3VVmO9mQaLTbbSDgO5xoTSm3dBGojS/8Pa4uZnYb3Zeu04OV6rC1g0+droYnmv
OXLzSqfjTjkQzenSCOrUnpqnNTWTkwJZ4kwAHPP8ayFTSoxh52HL0EYL0T+cafXv
UIrwQLMrVloda2JZBbOPJxgFCkNbAu/dUl5bwKkcVuOVbJdPAYNWcl3XfVHjWlQu
uJj9ck4lj4sW0bDhM+0SfBBVMyRmrw8zBlNIA4eftGR0tdI9InK30Y43ERM5357n
OAwLilkRMmX/9rlGvT82nqeUAFfwwBnhLNxM9y9MkB1D764I43OeOr+Z7CK5B1iu
2TVSS1G7gTaPn24hCqaOhw==
-----END CERTIFICATE-----
""",
            'commasInName': """\
-----BEGIN CERTIFICATE-----
MIIDuzCCAqOgAwIBAgIJALN9MAh64NFXMA0GCSqGSIb3DQEBBQUAMHQxCzAJBgNV
BAYTAnVzMQswCQYDVQQIDAJnYjELMAkGA1UEBwwCZmkxFjAUBgNVBAoMDUdvbywg
QmFyLCBCYXoxFjAUBgNVBAsMDUZvbywgQmFyLCBCYXoxGzAZBgNVBAMMEkJsZXRj
aCwgUXV1eCwgWmFydDAeFw0xMTA5MTMxNDI4MjFaFw0xMjA5MTIxNDI4MjFaMHQx
CzAJBgNVBAYTAnVzMQswCQYDVQQIDAJnYjELMAkGA1UEBwwCZmkxFjAUBgNVBAoM
DUdvbywgQmFyLCBCYXoxFjAUBgNVBAsMDUZvbywgQmFyLCBCYXoxGzAZBgNVBAMM
EkJsZXRjaCwgUXV1eCwgWmFydDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoC
ggEBANHNk3810A+83PPpli+Qeml4I4S9A6LTN9WjJjbQUEQmFBEqNaCh9DTJ+JeT
WPijvEqyBRPNxX8u//EyGSxZJGjAqwXK2pXhchUj7PnfwGTOZbIQRQRrqGCkL7r1
Y4ofz8TjPW5FI2wnRb0R54U7RMeDGOLOPSYKosZKqVeZ5ZYJ+gbfHqqBOolcZQZS
ijZrajGCeB+zvwwias7R6/91YZ7lbcQxxKcnidaSlXeR+UvC3nGgEJIpFQ/ODPvo
J8DW+JTaXAsHJB7LU3yJWssj94o9NZJbT1pF1ZF1AKdWWPA+rAUqLNChDsrQbLAb
O6OES02u6ZdAwUfdg4oLydiyiQ0CAwEAAaNQME4wHQYDVR0OBBYEFBRZmPynr7eR
1tVpQ9XYYhqgzNhuMB8GA1UdIwQYMBaAFBRZmPynr7eR1tVpQ9XYYhqgzNhuMAwG
A1UdEwQFMAMBAf8wDQYJKoZIhvcNAQEFBQADggEBADSildV6BTLUAaJ7c/TCGtQ/
Q+EWnW9EpK/AKekd7Lex8YNcfkJHDtkRwmuCE9xHtPosBlRgN7w3FGhCAilZ4A+h
aaiVSdTh18lVN474t8c1PxTRJBz1aNRdG5UMpznjwhsTCIKQXfs6qr761DU1SE7a
hJqaVh0quiNcYYeXrN8SAQefdFQCKwhbkeH4UocCqOpDcsDeSDXQw05IiAtEHRKg
VQS6c2wa4yjQAypQQeTl/ceXqx3zyh67wojWnzJ0MosPuc+kFKqRa/+pZrRfr2Z+
vQ/h/2mAJMBalFiIOOFU+egI2HeMGXF6zcSYIy09bQ1X88OiJ/PYGrG6ZGUlZQA=
-----END CERTIFICATE-----
""",
        'commasAndSpaces': """\
-----BEGIN CERTIFICATE-----
MIIDwzCCAqugAwIBAgIJAL0lAvUxQACRMA0GCSqGSIb3DQEBBQUAMHgxCzAJBgNV
BAYTAnVzMQswCQYDVQQIDAJnYjELMAkGA1UEBwwCZmkxGjAYBgNVBAoMEUdvbywg
QmFyLCAgICAgQmF6MRYwFAYDVQQLDA1Gb28sIEJhciwgQmF6MRswGQYDVQQDDBJC
bGV0Y2gsIFF1dXgsIFphcnQwHhcNMTEwOTEzMTUwMzUzWhcNMTIwOTEyMTUwMzUz
WjB4MQswCQYDVQQGEwJ1czELMAkGA1UECAwCZ2IxCzAJBgNVBAcMAmZpMRowGAYD
VQQKDBFHb28sIEJhciwgICAgIEJhejEWMBQGA1UECwwNRm9vLCBCYXIsIEJhejEb
MBkGA1UEAwwSQmxldGNoLCBRdXV4LCBaYXJ0MIIBIjANBgkqhkiG9w0BAQEFAAOC
AQ8AMIIBCgKCAQEAvu+nbGuAubKXN8Ivg6t+OKEOKOzz4XOIxYNuWuFXdqUM5VJz
+yD7EgHbq0rulY2jjaGkPil24W1fiy5tBbcRFEvhZYek9SqgNOMU6twhKSsUhhuC
k5y07A1BYBxsZh+JbZ1WQnKjIjPewOKueOjAOvOZYyyZxdijMAfKb9CVqxIx0iiF
rKGe3LptQYpzIXjJGuHtZNz/hVY/RajHKoYmH6E9qDemjoVoEmfDY664Q2uS8jGD
2U+SExvQEFWLit0YMbYJ+2syxc4W7OQPr8746Khw+eCvuM/6kPHZmkrVHgLP1+j1
Iwdh0h0DBcZ+zWuN+B4kRvH6UVtRtxeW3/dnrwIDAQABo1AwTjAdBgNVHQ4EFgQU
nLnn2aMlIzvMocHBLAZoBKGM/gowHwYDVR0jBBgwFoAUnLnn2aMlIzvMocHBLAZo
BKGM/gowDAYDVR0TBAUwAwEB/zANBgkqhkiG9w0BAQUFAAOCAQEAPYTqTq9zQJbg
3sczgc35aqOO9neVorFlA/d7oDqRgR439PNbrBjM0uaE49+Lhp+1x3f8KIVmc5fW
cgK7UQS1cvIn4JY3Q+6uv92FntHWLc5WRsC3YmtOZNYZtXG5ouMJyPIIQ7y6VoIS
DuzxnNYL7OHFSGZXTAomwOoJiNK13imBI6Bgb6GBPbxo7x9N21b2/gqKugA2ReYc
edEIJ6A+kSylAVgn0LtOWmUopSfhZaFx0tx8mpyQmE6G3tDGHIXe6LSlUzqEwWXe
C8G+2BnGb2Q45yamokhLEOK7U8jfjDZ7RyO6K9l6GcuaFUDFRstb1/znXMn9ONaZ
LBon4kaikA==
-----END CERTIFICATE-----
"""
    }

    # this one is long. go ahead and scroll down. or, in vim, use the
    # } command. or in emacs, M-}. crlutil expects a CRL in DER format
    # so we'll have to b64decode this.
    ca32CRL = """\
-----BEGIN X509 CRL-----
MIIcITCCGwkCAQEwDQYJKoZIhvcNAQEFBQAwVzELMAkGA1UEBhMCVVMxGDAWBgNV
BAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQLEwNQS0kx
EjAQBgNVBAMTCURPRCBDQS0zMhcNMTMwOTIzMDgwMDAwWhcNMTMwOTMwMTcwMDAw
WjCCGkowEwICC/gXDTEzMDkxMDEzNDM0MlowEwICHfAXDTEzMDkxNzEzMTcxNFow
EwICHegXDTEzMDkxNzEzMzAzNFowEwICEekXDTEzMDkxMDE0MzgzMFowEwICI+MX
DTEzMDkxNjE4NTg0MVowEwICBewXDTEzMDkxNjEyNDcwOVowEwICEegXDTEzMDkw
OTIzMDY1MFowEwICBeoXDTEzMDkxMDExNTE0MFowEwICBegXDTEzMDkwOTIwMDE0
NFowEwICEeIXDTEzMDkxNzE1MjkzOVowEwICJtsXDTEzMDkxNjEyMjAyNlowEwIC
FOAXDTEzMDkxNzAxMjc0OFowEwICC98XDTEzMDkxMDEzNTUyMlowEwICFNkXDTEz
MDkxNzAxMjc1MFowEwICDtcXDTEzMDkxMDEyNTYyM1owEwICI9AXDTEzMDkxNjEx
NTIzOVowEwICEcwXDTEzMDkxNzE1Mjk0MVowEwICFMsXDTEzMDkxMDAzMjY0MFow
EwICHcYXDTEzMDkxNzEzMTcxNlowEwICFMcXDTEzMDkxMDAzMjY0MlowEwICBccX
DTEzMDkwOTIwMDE0NVowEwICJrsXDTEzMDkxNzE2MTcxNVowEwICC7kXDTEzMDkx
MDEzNTUyNFowEwICDrUXDTEzMDkxMDEyNTYyNVowEwICCLYXDTEzMDkwOTE4Mjcy
OFowEwICJqsXDTEzMDkxNjE0Mzg0MVowEwICBbMXDTEzMDkxNjEyNDcxMVowEwIC
FK4XDTEzMDkxNTIzMTQ1MFowEwICEawXDTEzMDkwOTIwMzgyM1owEwICJqUXDTEz
MDkxMDE0NDI1MFowEwICBa8XDTEzMDkxMDExNTE0MlowEwICFKgXDTEzMDkxNTIz
MTQ1MVowEwICCKAXDTEzMDkwOTE4MjczMFowEwICApwXDTEzMDkwOTIwMDgyM1ow
EwICBZsXDTEzMDkwOTE4NTYyNVowEwICC5cXDTEzMDkwOTE5MDcyNlowEwICApkX
DTEzMDkwOTIwMDgyNVowEwICJo0XDTEzMDkxNzE2MTcxN1owEwICJooXDTEzMDkx
NjE0Mzg0MlowEwICEZAXDTEzMDkwOTIwMzgyNVowEwICApEXDTEzMDkxMDEyNTU0
MFowEwICJoQXDTEzMDkxMDE0NDI0OFowEwICCIsXDTEzMDkxMDEwMTc0MFowEwIC
F4IXDTEzMDkxMDA3Mjg0MFowEwICAoYXDTEzMDkxMDEyNTU0MlowEwICBYUXDTEz
MDkwOTE4NTYyN1owEwICF38XDTEzMDkxMDA3Mjg0MlowEwICEXgXDTEzMDkxMDE0
MzcxMFowEwICIG8XDTEzMDkxNzEyNTc0NlowEwICAncXDTEzMDkwOTE3MTUxNVow
EwICC3MXDTEzMDkxMDEzNTc0OFowEwICAnUXDTEzMDkwOTE3MTUxN1owEwICC3AX
DTEzMDkwOTE5MDcyOFowEwICEW0XDTEzMDkxMDE0MzcxMlowEwICI2IXDTEzMDkx
MDE0MDkzOFowEwICI18XDTEzMDkxNjEzMTA0MlowEwICIFUXDTEzMDkxNjE3NTgw
N1owEwICC1gXDTEzMDkxMDEzNTc0OVowEwICDk0XDTEzMDkxNzE1MDYzNlowEwIC
IEMXDTEzMDkxNzEyNTc0OFowEwICAkMXDTEzMDkxMDExMzcyNVowEwICHTgXDTEz
MDkxMDE1MDQ0NFowEwICAj0XDTEzMDkxMDExMzcyN1owEwICIzIXDTEzMDkxNjEx
MzYwM1owEwICHTIXDTEzMDkxNjEyNTAxOFowEwICIDEXDTEzMDkxNjE3NTgwOVow
EwICCDUXDTEzMDkxMDEwMTc0MlowEwICGi8XDTEzMDkxMDEyMDExNVowEwICBTUX
DTEzMDkwOTE4NTkxMlowEwICGi0XDTEzMDkxNzExMTk1MFowEwICHSwXDTEzMDkx
NjEyNTkzMVowEwICIyYXDTEzMDkxNzE1MzIwMVowEwICBS0XDTEzMDkxMDEzMDg1
MlowEwICDicXDTEzMDkxNzE1MDYzOFowEwICIyAXDTEzMDkxNjEzMTA0NFowEwIC
GiAXDTEzMDkxMDEyMDExM1owEwICGh8XDTEzMDkxNzExMTk1MlowEwICESEXDTEz
MDkwOTIwMzAyMVowEwICDiEXDTEzMDkxMDEzNTgyOVowEwICBSAXDTEzMDkwOTE4
NTkxNFowEwICHRgXDTEzMDkxMDE1MDQ0NVowEwICIxYXDTEzMDkxMDE0NDYyOFow
EwICGhQXDTEzMDkxMDEyMDU0M1owEwICIw8XDTEzMDkxNjExMzYwNlowEwICGgwX
DTEzMDkxNzE0NTA1MlowEwICIwgXDTEzMDkxNzE1MzIwM1owEwICEQwXDTEzMDkx
NzE1MDgzMVowEwICEQkXDTEzMDkwOTIwMzAyM1owEwICAgsXDTEzMDkwOTExMDk0
M1owEwICFAQXDTEzMDkwOTIyMTA0NVowEwICAgkXDTEzMDkwOTExMDk0MFowEwIC
BQcXDTEzMDkxMDEzMDg1NFowEwICCAYXDTEzMDkxNjE4MDUzOVowEwICHP8XDTEz
MDkxNjEyNTkzM1owEwICE/4XDTEzMDkwOTIyMTA0N1owEwICHPkXDTEzMDkxNjEy
NTIxOVowEwICBQAXDTEzMDkxNjEyMjM0OVowEwICDfoXDTEzMDkxMDEzNTgzMVow
EwICEPUXDTEzMDkxNzE1MDgzMlowEwICGfEXDTEzMDkxNjE1MjE0OFowEwICGe8X
DTEzMDkxNzE0NTA1NFowEwICB/IXDTEzMDkxMDEyNTc1NVowEwICAfIXDTEzMDkx
```

```
MDExNDg0NlowEwICB/AXDTEzMDkwOTE4MTQzMVowEwICBO8XDTEzMDkwOTE4NTQw
M1owEwICAe4XDTEzMDkxMDExNDg0OFowEwICBOOXDTEzMDkxNjEyMjM1MFowEwIC
B+wXDTEzMDkxNjE4MDU0MVowEwICDegXDTEzMDkxMDEyMzU0OVowEwICAeUXDTEz
MDkxMDA5MzkxMlowEwICAeIXDTEzMDkxMDA5MzkxNFowEwICEN0XDTEzMDkwOTIy
NDM1MlowEwICCtsXDTEzMDkwOTE5MDQwOFowEwICDdIXDTEzMDkxMDEyMzU1Mlow
EwICGcgXDTEzMDkxNjE1MjE1MFowEwICIsUXDTEzMDkxMDE0MDIwM1owEwICBM4X
DTEzMDkwOTE4NTQwNVowEwICB8cXDTEzMDkwOTE4MTQyOFowEwICIr4XDTEzMDkx
MDE0NDYzMFowEwICB7QXDTEzMDkxMDEyNTc1N1owEwICFqwXDTEzMDkxNzA5MzYy
N1owEwICDa4XDTEzMDkxNjE2MTYyN1owEwICFqkXDTEzMDkxNzA5MzYyOVowEwIC
DZ8XDTEzMDkxMDE0MTQyOVowEwICDZYXDTEzMDkxMDEzNTQxOFowEwICH4oXDTEz
MDkxNjEyMDkzN1owEwICJYgXDTEzMDkxMDE0NDQxOVowEwICDY8XDTEzMDkxMDEy
MDg0MVowEwICCo8XDTEzMDkwOTE5MDQxMFowEwICBI8XDTEzMDkwOTE3MjE0Nlow
EwICBI0XDTEzMDkwOTE5NTA1NFowEwICDYoXDTEzMDkxNjE2MTYyOVowEwICJYIX
DTEzMDkxMDE0MzUxNlowEwICE4cXDTEzMDkxMDExNDQyMFowEwICIoEXDTEzMDkx
MDE0MDIwNVowEwICE4AXDTEzMDkxMDExNDQyMVowEwICEH4XDTEzMDkwOTIwMjUw
NVowEwICAX8XDTEzMDkxMDA4NDU1MFowEwICH3MXDTEzMDkxNjEyMDkzOVowEwIC
AXwXDTEzMDkxNzAyNDA1NVowEwICB3gXDTEzMDkxMDEyNTgyM1owEwICAXkXDTEz
MDkxMDA4NDU1MlowEwICDXQXDTEzMDkwOTIzNTcxMlowEwICAXcXDTEzMDkxNzAy
NDA1N1owEwICBHYXDTEzMDkwOTE3MjE0OFowEwICImsXDTEzMDkxNjE4NTEzNlow
EwICAXYXDTEzMDkxMDA5MTI0N1owEwICH2sXDTEzMDkxNzE1NDE1OVowEwICAXUX
DTEzMDkxMDA5MTI0OVowEwICDXEXDTEzMDkxMDEyMDg0NFowEwICDXAXDTEzMDkx
NjE0MTczOFowEwICEGwXDTEzMDkwOTIwMjUwNlowEwICJWQXDTEzMDkxMDE0MzYx
MFowEwICDWoXDTEzMDkxMDEzNTQyMFowEwICAWwXDTEzMDkxNzEyMzAxM1owEwIC
AWkXDTEzMDkxNzEyMzAxNVowEwICDWQXDTEzMDkxMDE0MTQzMVowEwICJVcXDTEz
MDkxMDE0NDQyMVowEwICBGEXDTEzMDkwOTE5NTA1NlowEwICAV8XDTEzMDkxMDA3
MzU1OVowEwICJVMXDTEzMDkxMDE0MjgwNlowEwICE1UXDTEzMDkwOTIyMjU0OFow
EwICIlAXDTEzMDkxNjE4NTEzOFowEwICAVgXDTEzMDkxMDA3MzYwMVowEwICAVMX
DTEzMDkxMDAzMDQ0MFowEwICIkcXDTEzMDkxMDEzNTcxM1owEwICIkUXDTEzMDkx
NjEzNTQ0M1owEwICAU8XDTEzMDkxMDAzMDQ0MlowEwICCkoXDTEzMDkxMDEzNDUz
MlowEwICDUkXDTEzMDkxNjE0MTc0MFowEwICH0IXDTEzMDkxNzE1NDIwMVowEwIC
GToXDTEzMDkxMDEzNTUzMlowEwICCj4XDTEzMDkxMDE0MDAzMFowEwICEDsXDTEz
MDkxNjExNTIyNVowEwICEzkXDTEzMDkwOTIyMjU1MFowEwICHzUXDTEzMDkxNzEy
NTgxMVowEwICGTUXDTEzMDkxMDE0Mjg1NVowEwICFjAXDTEzMDkxMDAzMzI1MVow
EwICBzMXDTEzMDkxMDEyNTgyNVowEwICGSwXDTEzMDkxMDEzNTUzNFowEwICFiwX
DTEzMDkxMDAzMzI1M1owEwICDSwXDTEzMDkwOTIzNTcxNFowEwICGSYXDTEzMDkx
MDE0Mjg1N1owEwICJSAXDTEzMDkxMDE0MjgwOFowEwICECUXDTEzMDkxNjExNTIy
NlowEwICBCAXDTEzMDkxNjE3MTQyN1owEwICHxYXDTEzMDkxNjEyMTgzOFowEwIC
HxQXDTEzMDkxNzEyNTgxM1owEwICChoXDTEzMDkxMDEzNDUzNFowEwICExMXDTEz
MDkxNzE0MDY1OFowEwICBBYXDTEzMDkxMDE0NDQwM1owEwICBxAXDTEzMDkwOTE5
MTYyNFowEwICKAMXDTEzMDkxNjIwMDAwMlowEwICKAIXDTEzMDkxNDAxMzYyNFow
EwICKAEXDTEzMDkxNDAxMzYyNFowEwICKAAXDTEzMDkxNDAxMzYyNFowEwICJ/8X
DTEzMDkxNDAxMzYyNFowEwICJ/4XDTEzMDkxNDAxMzYyNFowEwICEwQXDTEzMDkw
OTIzMjcxMlowEwICJ/0XDTEzMDkxNDAxMzYyNFowEwICJ/wXDTEzMDkxNDAxMzYy
NFowEwICIf0XDTEzMDkxNjEzNTQ0NVowEwICEwEXDTEzMDkxNzE0MDcwMFowEwIC
BAAXDTEzMDkxMDE0NDQwNVowEwICA/sXDTEzMDkxNjE3MTQyOVowEwICCfkXDTEz
MDkxMDE0MDAzMlowEwICEvYXDTEzMDkwOTIzMjcxNFowEwICHucXDTEzMDkxNjEy
MTg0MFowEwICD+UXDTEzMDkxMDAwMDIwN1owEwICA+gXDTEzMDkwOTE5MDIzOFow
EwICEuEXDTEzMDkxNjE4NTk0OVowEwICCeMXDTEzMDkwOTE5MzcwM1owEwICCeIX
DTEzMDkxNjE3MDQ0N1owEwICJNkXDTEzMDkxNjE0NTEwOFowEwICD94XDTEzMDkx
MDE0NDYyMFowEwICBuAXDTEzMDkwOTE5MTYyNlowEwICA98XDTEzMDkxMDEyMzkw
OVowEwICBtoXDTEzMDkxMDEzMDc0MFowEwICG9EXDTEzMDkxMDE0MDYwN1owEwIC
EtEXDTEzMDkxNjE4NTk1MVowEwICCdMXDTEzMDkwOTE4NTEzOVowEwICDNEXDTEz
MDkwOTIwMzIzNFowEwICJ78XDTEzMDkxNjEzNDIyN1owEwICD8UXDTEzMDkxMDE0
NDYyMlowEwICEsIXDTEzMDkwOTIyNTQ1MlowEwICA8YXDTEzMDkwOTE5MDI0MFow
EwICCcMXDTEzMDkxNzEzMDc1NFowEwICA8QXDTEzMDkxMDEyMzkxMVowEwICJLUX
DTEzMDkxNjE0NTExMFowEwICBroXDTEzMDkxMDEzMjk1N1owEwICCbkXDTEzMDkx
MDE0NTY1NFowEwICCbgXDTEzMDkxNjE3MDQ0OVowEwICG7AXDTEzMDkxMDE0MDYw
```

```
OVowEwICD7IXDTEzMDkxMDAwMDIwOFowEwICALQXDTEzMDkxMDAzMDMwN1owEwIC
DLAXDTEzMDkxNjE1NTg0NVowEwICALMXDTEzMDkxMDAzMDMwOVowEwICEq0XDTEz
MDkwOTIyNTQ1NFowEwICBq8XDTEzMDkxMDEzMDc0MlowEwICEqoXDTEzMDkxNjIw
MjgyM1owEwICJ6EXDTEzMDkxNjEzNDIyOVowEwICCaYXDTEzMDkwOTE4NTE0MFow
EwICD54XDTEzMDkxMDEzNDEwNFowEwICCZwXDTEzMDkxNzEzMDc1NlowEwICCZsX
DTEzMDkwOTE5MzcwNVowEwICEpYXDTEzMDkxNjIwMjgyNVowEwICGJQXDTEzMDkx
NzEyNTEwNlowEwICBpgXDTEzMDkxMDEzMjk1OVowEwICDJYXDTEzMDkwOTIwMzIz
NVowEwICGJEXDTEzMDkxNzEyNTEwOFowEwICCZQXDTEzMDkxMDE0NTY1NlowEwIC
AJYXDTEzMDkxMDA2MjkwOVowEwICDI8XDTEzMDkxNjE1NTg0NlowEwICAJIXDTEz
MDkxMDA2MjkxMVowEwICEowXDTEzMDkxNzE4NTcyMlowEwICA44XDTEzMDkxNjE2
NTQxNVowEwICD4kXDTEzMDkxMDEzNDEwNlowEwICJ4EXDTEzMDkxNjE1MzgzN1ow
EwICEoEXDTEzMDkxNzE4NTcyM1owEwICG3sXDTEzMDkxMDE0MzUyM1owEwICA3wX
DTEzMDkxNjE2NTQxNlowEwICCXkXDTEzMDkwOTE4NDcwNVowEwICD28XDTEzMDkx
MDExNDMyN1owEwICIWkXDTEzMDkxMDE0MDcwOFowEwICA3IXDTEzMDkwOTE4NDAw
N1owEwICFWoXDTEzMDkxNzAwNTM1MlowEwICBmsXDTEzMDkwOTIwMTczNVowEwIC
BmoXDTEzMDkxNjE0MTAzNFowEwICGGQXDTEzMDkxMDA5NTA0NFowEwICA2oXDTEz
MDkwOTE4NDAwOVowEwICBmkXDTEzMDkxMDEzMDIyMVowEwICFWQXDTEzMDkxNzAw
NTM1NVowEwICGGEXDTEzMDkxMDA5NTA0MVowEwICD2AXDTEzMDkxMDExNDMyOVow
EwICJ1cXDTEzMDkxNjE1MzgzOVowEgIBYhcNMTMwOTE2MDE1MTIwWjASAgFfFw0x
MzA5MTYwMTUxMjJaMBMCAglcFw0xMzA5MDkxOTE5MDBaMBMCAhtUFw0xMzA5MTcx
MjM3NDRaMBMCAh5TFw0xMzA5MTAxMzU4NTVaMBMCAgxVFw0xMzA5MTYxODAwNTda
MBMCAhtQFw0xMzA5MTAxNDM1MjVaMBMCAgZMFw0xMzA5MTAxMzAyMjNaMBMCAhJH
Fw0xMzA5MDkyMDU0MzBaMBMCAiFCFw0xMzA5MTAxNDA3MTBaMBMCAgNKFw0xMzA5
MDkyMTU4MzhaMBMCAhJAFw0xMzA5MDkyMDU0MzFaMBMCAiE6Fw0xMzA5MTAxNDMw
NDJaMBMCAhU+Fw0xMzA5MTAwMTQ1MzdaMBMCAgxAFw0xMzA5MTcxMzQ4MzRaMBMC
AhU7Fw0xMzA5MTAwMTQ1MzlaMBMCAgk+Fw0xMzA5MDkxODQ3MDdaMBMCAgNAFw0x
MzA5MDkyMTU4NDBaMBMCAgY+Fw0xMzA5MTYxNDEyMzVaMBMCAhI5Fw0xMzA5MTYx
OTI1MzFaMBMCAhsyFw0xMzA5MTcxMjQyNTdaMBMCAgk1Fw0xMzA5MDkxOTE5MDJa
MBMCAgY0Fw0xMzA5MDkyMDE3MzZaMBMCAgMzFw0xMzA5MTYxNzI3NDZaMBICATQX
DTEzMDkwOTAwMzQ0N1owEwICHioXDTEzMDkxNzEzMzAzMlowEgIBMxcNMTMwOTA5
MDAzNDQ5WjATAgIhJhcNMTMwOTEwMTQzMDQ0WjATAgIkJBcNMTMwOTE2MTg1ODM5
WjATAgIDLBcNMTMwOTE2MTcyNzQ4WjATAgISJhcNMTMwOTE2MTkyNTMzWjASAgEq
Fw0xMzA5MDMxODExMTBaMBICASkXDTEzMDkwMzE4MTExM1owEgIBKBcNMTMwODMw
MTkzNDEwWjASAgElFw0xMzA4MzAxNTUxMjZaMBMCAicXFw0xMzA5MTYxMjIwMjRa
MBMCAh4aFw0xMzA5MTAxMzU4NTdaMBICASQXDTEzMDgzMDE1NTEyOVowEgIBIRcN
MTMwODMwMTUxOTA4WjASAgEgFw0xMzA4MzAxNTE5MTFaMBMCAgwaFw0xMzA5MTAx
MzQzNDBaMBMCAhIPFw0xMzA5MTAxNDM4MjlaMBMCAgwRFw0xMzA5MTcxMzQ4MzZa
MBMCAiQIFw0xMzA5MTYxMTUyMzdaMBMCAhIHFw0xMzA5MDkyMzA2NDhaMBMCAgwB
Fw0xMzA5MTYxODAwNTlaoDAwLjAfBgNVHSMEGDAWgBQvypNTA6xvdfV2+r7+juoQ
i7fnkDALBgNVHRQEBAICAiowDQYJKoZIhvcNAQEFBQADggEBAJi3Ze+5x2FBHmgK
wjJiMMpiwkr2UW67/bx9RpraYG9EV3JWVVbJECFegUAXYkiv1YP4WHIukX1efEIO
4ju+o3t3UGBIOpU/J5rbg4i5aUsnYBKRGZiRDxiSIlqrWlHfvF5pyGNPhLC+bzi8
iujjkHj5LunyH4OHFFPbM88Q3PP7sPEBO/w26LqqWKBo/bqqKIRlaXgc4U4CaPKK
QONEyC21ixAF9Vqdod4HmAxdRRfZ3OWChTRGP7hyVGt2z1AEK4glSkyJrMDa0iYJ
ALrF3/75ZRdJo2vMZbxDdxRfYSZWeZLZV0oj4stdgjCy2xYjT/xBgHO5a9CMAimZ
OYmeJRw=
-----END X509 CRL-----
"""
```

```
class DBSetupBerkeleyDB(CommonDataForTest):
    db_prefix = ’’
    sqlite = False
    def setUp(self):
        self.dir = tempfile.mkdtemp()
        db_spec = self.db_prefix + self.dir
        self.pwfile = os.path.join(self.dir, ’pwfile’)
        with file(self.pwfile, ’w’) as f:
            print >> f, ’internal:ridiculous password’
            print >> f, ’NSS Certificate DB:ridiculous password’
            print >> f, ’NSS FIPS 140-2 Certificate DB:ridiculous password’
        dashn = Popen((’certutil’, ’-N’, ’-d’, db_spec, ’-f’, self.pwfile),
                      stdin=PIPE, stdout=PIPE, stderr=PIPE)
        out, err = dashn.communicate()
        if dashn.returncode != 0:
            raise Exception(’Test NSS database creation failed’,
                    dashn.returncode, out, err)
        for nick, cert in self.certs.items():
            dasha = Popen((’certutil’, ’-A’, ’-d’, db_spec, ’-f’,
                self.pwfile, ’-n’, nick, ’-t’, ’CT,C,C’),
                stdin=PIPE, stdout=PIPE, stderr=PIPE)
            out, err = dasha.communicate(cert)
            if dasha.returncode != 0:
                raise Exception(’Test NSS certificate add failed’,
                        nick, dasha.returncode, out, err)
        self.db = NSSDB(self.dir, self.pwfile, self.sqlite)

    def tearDown(self):
        shutil.rmtree(self.dir)

class DBSetupSqliteDB(DBSetupBerkeleyDB):
    db_prefix = ’sql:’
    sqlite = True

class CACertTests(object):
    def testListCerts(self):
        self.assertEqual(set(self.db.certNicknames()),
                set(self.certs.keys()))

    def testFetchAbsentCRL(self):
        ca32 = CACert(self.db, ’DoD-Root2-CA32’)
        crl32 = CRL(self.db, ca32)
        self.assertEqual(tuple(self.db.crlNicknames()), ())
        crl32.fetchIfNecessary()
        self.assertEqual(tuple(self.db.crlNicknames()),
                (’DoD-Root2-CA32’,))

    def testCACertDN(self):
        ca32 = CACert(self.db, ’DoD-Root2-CA32’)
        self.assertEqual(ca32.dn, (’CN=DOD CA-32’, ’OU=PKI’, ’OU=DoD’,
            ’O=U.S. Government’, ’C=US’))

    def testCACertCN(self):
        ca32 = CACert(self.db, ’DoD-Root2-CA32’)
        self.assertEqual(ca32.cn, ’DOD CA-32’)

    def testCommasDN(self):
        commas = CACert(self.db, ’commasInName’)
        self.assertEqual(commas.dn, (’CN="Bletch, Quux, Zart"’,
            ’OU="Foo, Bar, Baz"’, ’O="Goo, Bar, Baz"’, ’L=fi’, ’ST=gb’,
            ’C=us’))

    def testCommasAndSpacesDN(self):
        comspace = CACert(self.db, ’commasAndSpaces’)
        self.assertEqual(comspace.dn, (’CN="Bletch, Quux, Zart"’,
            'OU="Foo, Bar, Baz"', 'O="Goo, Bar,     Baz"', 'L=fi',
            'ST=gb', 'C=us'))

class TestCACertBerkeley(CACertTests, DBSetupBerkeleyDB, unittest.TestCase)[WRAP]
:
    pass
class TestCACertSqlite(CACertTests, DBSetupSqliteDB, unittest.TestCase):
    pass

class WithCRLSetup(object):
    def setUp(self):
        super(WithCRLSetup, self).setUp()
        db_spec = self.db_prefix + self.dir
        for crl in (self.ca32CRL,):
            just_base64 = '\n'.join(crl.split('\n')[1:-2])
            f = file(os.path.join(self.dir, 'crlin'), 'w')
            f.write(base64.b64decode(just_base64))
            f.close()
            dasha = Popen(('crlutil', '-I', '-d', db_spec, '-f',
                self.pwfile, '-a', '-i', os.path.join(self.dir,
                    'crlin')),
                stdin=PIPE, stdout=PIPE, stderr=PIPE)
            out, err = dasha.communicate()
            if dasha.returncode != 0:
                raise Exception('Test NSS CRL add failed',
                        dasha.returncode, out, err)

class WithCRLTests(object):
    def testListCRLs(self):
        self.assertEqual(tuple(self.db.crlNicknames()),
                ('DoD-Root2-CA32',))

    def testCRLDates(self):
        ca = CACert(self.db, 'DoD-Root2-CA32')
        crl = CRL(self.db, ca)
        self.assertEqual(crl._getValidity(),
                (datetime.datetime(2013, 9, 23, 8, 0),
                    datetime.datetime(2013, 9, 30, 17, 0)))

class TestWithCRLBerkeley(WithCRLTests, WithCRLSetup, DBSetupBerkeleyDB,
                          unittest.TestCase):
    pass
class TestWithCRLSqlite(WithCRLTests, WithCRLSetup, DBSetupSqliteDB,
                          unittest.TestCase):
    pass

if __name__ == '__main__':
    unittest.main()
```

### 12.11.44 pam_pkcs11.conf

```
#
# Configuration file for pam_pkcs11 module
#
# Version 0.4
# Original author: Juan Antonio Martinez <jonsito@teleline.es>
# Modified: Jared Jennings <jared.jennings.ctr@us.af.mil>
#
# This file is automatically put in place by puppet.
#

pam_pkcs11 {
  # No empty passwords.
  nullok = false;

  # Enable debugging support. Very useful, but *!WARNING!* this results in [WRAP]
PINs
  # being visible, in the clear, on the screen.
  debug = false;

  # If the smart card is inserted, only use it
  card_only = true;

  # Do not prompt the user for the passwords but take them from the
  # PAM_ items instead.
  use_first_pass = false;

  # Do not prompt the user for the passwords unless PAM_(OLD)AUTHTOK
  # is unset.
  try_first_pass = false;

  # Like try_first_pass, but fail if the new PAM_AUTHTOK has not been
  # previously set (intended for stacking password modules only).
  use_authtok = false;

  # Filename of the PKCS #11 module. The default value is "default"
  use_pkcs11_module = coolkey;

  screen_savers = gnome-screensaver,xscreensaver,kscreensaver

  pkcs11_module coolkey {
    module = libcoolkeypk11.so;
    description = "Cool Key"
    # Slot-number to use. One for the first, two for the second and so
    # on. The default value is zero which means to use the first slot
    # with an available token.
    slot_num = 0;

    # Path to the directory where the CA certificates are stored. The
    # directory must contain an openssl hash-link to each certificate.
    # The default value is /etc/pam_pkcs11/cacerts.
    ca_dir = /etc/pam_pkcs11/cacerts;
    nss_dir = /etc/pki/pam_pkcs11;

    # Path to the directory where the CRLs are stored. The directory
    # must contain an openssl hash-link to each CRL. The default value
    # is /etc/pam_pkcs11/crls.
    crl_dir = /etc/pam_pkcs11/crls;

    # sets the certificate verification policy.
    # "none"        performs no verification
    # "ca"          does ca check
    # "crl_online"  downloads the crl form the location given by the
    #               crl distribution point extension of the certificate
    # "crl_offline" uses the locally stored crls
    # "crl_auto"    is a combination of online and offline; it first
    #               tries to download the crl from a possibly given crl
    #               distribution point and if this fails, uses the local
    #               crls
    # "ocsp_on"     turn on ocsp.
    # "signature"   does also a signature check to ensure that private
    #               and public key matches
    # you can use a combination of ca,crl, and signature flags, or just
    # use "none".
    cert_policy=ca, signature;
  }

  pkcs11_module opensc {
    module = opensc-pkcs11.so;
    description = "OpenSC PKCS#11 module";
    # Slot-number to use. One for the first, two for the second and so
    # on. The default value is zero which means to use the first slot
    # with an available token.
    slot_num = 0;

    # Path to the directory where the CA certificates are stored. The
    # directory must contain an openssl hash-link to each certificate.
    # The default value is /etc/pam_pkcs11/cacerts.
    ca_dir = /etc/pam_pkcs11/cacerts;

    # Path to the directory where the CRLs are stored. The directory
    # must contain an openssl hash-link to each CRL. The default value
    # is /etc/pam_pkcs11/crls.
    crl_dir = /etc/pam_pkcs11/crls;

    # Sets the Certificate Policy, (see above)
    cert_policy=ca, signature;
  }

  # Default pkcs11 module
  pkcs11_module default {
    module = /usr/$LIB/pam_pkcs11/pkcs11_module.so;
    description = "Default pkcs#11 module";
    slot_num = 0;
    ca_dir = /etc/pam_pkcs11/cacerts;
    crl_dir = /etc/pam_pkcs11/crls;
    cert_policy=ca, signature;
  }

  # Which mappers ( Cert to login ) to use?
  # you can use several mappers:
  #
  # subject - Cert Subject to login file based mapper
  # pwent   - CN to getpwent() login or gecos fields mapper
  # ldap    - LDAP mapper
  # opensc  - Search certificate in ${HOME}/.eid/authorized_certificates
  # openssh - Search certificate public key in ${HOME}/.ssh/authorized_keys
  # mail    - Compare email fields from certificate
  # ms      - Use Microsoft Universal Principal Name extension
```

```
  # krb     - Compare againts Kerberos Principal Name
  # cn      - Compare Common Name (CN)
  # uid     - Compare Unique Identifier
  # digest  - Certificate digest to login (mapfile based) mapper
  # generic - User defined certificate contents mapped
  # null    - blind access/deny mapper
  #
  # You can select a comma-separated mapper list.
  # If used null mapper should be the last in the list :-)
  # Also you should select at least one mapper, otherwise
  # certificate will not match :-)
  use_mappers = subject, null;

  # When no absolute path or module info is provided, use this
  # value as module search path
  # TODO:
  # This is not still functional: use absolute pathnames or LD_LIBRARY_PATH[WRAP]

  mapper_search_path = /usr/$LIB/pam_pkcs11;

  #
  # Generic certificate contents mapper
  mapper generic {
        debug = true;
        module = /usr/$LIB/pam_pkcs11/generic_mapper.so;
        # ignore letter case on match/compare
        ignorecase = false;
        # Use one of "cn" , "subject" , "kpn" , "email" , "upn" or "uid"
        cert_item  = cn;
        # Define mapfile if needed, else select "none"
        mapfile = file:///etc/pam_pkcs11/generic_mapping
        # Decide if use getpwent() to map login
        use_getpwent = false;
  }

  # Certificate Subject to login based mapper
  # provided file stores one or more "Subject -> login" lines
  mapper subject {
debug = false;
# module = /usr/$LIB/pam_pkcs11/subject_mapper.so;
module = internal;
ignorecase = false;
mapfile = file:///etc/pam_pkcs11/subject_mapping;
  }

  # Search public keys from $HOME/.ssh/authorized_keys to match users
  mapper openssh {
debug = false;
module = /usr/$LIB/pam_pkcs11/openssh_mapper.so;
  }

  # Search certificates from $HOME/.eid/authorized_certificates to match us[WRAP]
ers
  mapper opensc {
debug = false;
module = /usr/$LIB/pam_pkcs11/opensc_mapper.so;
  }
  # Certificate Common Name ( CN ) to getpwent() mapper
  mapper pwent {
debug = false;
ignorecase = false;
module = internal;
# module = /usr/$LIB/pam_pkcs11/pwent_mapper.so;
  }

  # Null ( no map ) mapper. when user as finder matchs to NULL or "nobody"
  mapper null {
debug = false;
# module = /usr/$LIB/pam_pkcs11/null_mapper.so;
module = internal ;
# select behavior: always match, or always fail
default_match = false;
# on match, select returned user
        default_user = nobody ;
  }

  # Directory ( ldap style ) mapper
  mapper ldap {
debug = false;
module = /usr/$LIB/pam_pkcs11/ldap_mapper.so;
# where base directory resides
basedir = /etc/pam_pkcs11/mapdir;
# hostname of ldap server
        ldaphost = "localhost";
# Port on ldap server to connect
        ldapport = 389;
        # Scope of search: 0 = x, 1 = y, 2 = z
        scope = 2;
# DN to bind with. Must have read-access for user entries under "base"
        binddn = "cn=pam,o=example,c=com";
# Password for above DN
        passwd = "test";
# Searchbase for user entries
        base = "ou=People,o=example,c=com";
# Attribute of user entry which contains the certificate
        attribute = "userCertificate";
# Searchfilter for user entry. Must only let pass user entry for the login[WRAP]
 user.
        filter = "(&(objectClass=posixAccount)(uid=%s))"
  }

  # Assume common name (CN) to be the login
  mapper cn {
debug = false;
module = internal;
# module = /usr/$LIB/pam_pkcs11/cn_mapper.so;
ignorecase = true;
mapfile = file:///etc/pam_pkcs11/cn_map;
  }

  # mail -  Compare email field from certificate
  mapper mail {
debug = false;
module = internal;
# module = /usr/$LIB/pam_pkcs11/mail_mapper.so;
# Declare mapfile or
# leave empty "" or "none" to use no map
mapfile = file:///etc/pam_pkcs11/mail_mapping;
# Some certs store email in uppercase. take care on this
ignorecase = true;
# Also check that host matches mx domain
# when using mapfile this feature is ignored
ignoredomain = false;
  }

  # ms - Use Microsoft Universal Principal Name extension
  # UPN is in format login@ADS_Domain. No map is needed, just
  # check domain name.
  mapper ms {
debug = false;
module = internal;
# module = /usr/$LIB/pam_pkcs11/ms_mapper.so;
ignorecase = false;
ignoredomain = false;
domain = "domain.com";
  }

  # krb  - Compare againts Kerberos Principal Name
  mapper krb {
debug = false;
module = internal;
# module = /usr/$LIB/pam_pkcs11/krb_mapper.so;
ignorecase = false;
mapfile = "none";
  }

  # uid  - Maps Subject Unique Identifier field (if exist) to login
  mapper uid {
debug = false;
module = internal;
# module = /usr/$LIB/pam_pkcs11/uid_mapper.so;
ignorecase = false;
mapfile = "none";
  }

  # digest - elaborate certificate digest and map it into a file
  mapper digest {
debug = false;
module = internal;
# module = /usr/$LIB/pam_pkcs11/digest_mapper.so;
# algorithm used to evaluate certificate digest
      # Select one of:
# "null","md2","md4","md5","sha","sha1","dss","dss1","ripemd160"
algorithm = "sha1";
mapfile = file:///etc/pam_pkcs11/digest_mapping;
# mapfile = "none";
  }

}
```

## 12.11.45 pkinit/DoD-Root2-CA21.crt

```
-----BEGIN CERTIFICATE-----
MIIFiTCCBHGgAwIBAgIBTDANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0wOTAxMjYxNjM1MDNaFw0x
NTAxMjUxNjM1MDNaMFcxCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRIwEAYDVQQDEwlET0Qg
Q0EtMjEwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDdlElknbLr9TVZ
5hTjI5zGC1inX0nBxgikNyl7IxR5CP4aLtpxFGKAL2NSlnuEl/bASHmxo0kIh9Ov
t49pTRAi4v5wXTyTCpxYXm8qXYH+HWI5LruZDgNan8bldy2IDWDMtIp3TF+b5qU/
pq8E6cxSnqyAZIOlaRXzVE30qAI6c5wWxEKFK0E3CUDEWCNPp0snxwdD5TgsDH/Y
A5WCCX+2mWhWhogD4dJUKnUXS2XK8xJFy5YQ7BPMG76bBFT7PFGbNH53jn35Mb0O
n3zoHjfLUk6IPecJvVgjAJbyvKcDtDXmDHZvaCMicq2Lt/f/Ju0tHrVZQA2o/a0n
H1Hkue1BAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0jBBgwFoAU
SXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFAmZE+Kj1ed02PY/tdz71LUW
7UzTMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYDVR0gBIGX
MIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsKMAsGCWCG
SAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFlAwIBAwYw
DAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0wDAYKYIZI
AWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2EubWlsL2dl
dGNybD9Eb0QlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB7jA/Bggr
BgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9Eb0QlMjBS
b290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCB
iAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNkRG9EJTIw
Um9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNkVS5TLiUy
MEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkwDQYJKoZI
hvcNAQEFBQADggEBACXuf0uCpdBRmSoj3POtJyXAaXlIADIm0u5sHBy78MAM09gs
dFilVQlDolr5J/7YWujgqKS9vQWlC5UmHA4IiA7k+R97fphBDDOgjkTC8azehAGG
7DXs/4G7YH2Ot1byTJACH9OIPOkhbowrvG8bQBlisuMUcL/RgEukcT8U7uD06R71
BYESPdT8AIOyH8IFLGMgCcJHnVsek3emIwsWY3Ba5M3eJSbcrVcIMSNmm5+cCRpU
/IlYa4P632JwHHr5MjX7w+jPBmrS2Tm6PY+uYHsqZgA5xVCpXkNNobwKsiT7EjZX
zfjKO19+y8URKtUEBftfWOdUB2epSQeOSlYTZks=
-----END CERTIFICATE-----
```

## 12.11.46 pkinit/DoD-Root2-CA22.crt

```
-----BEGIN CERTIFICATE-----
MIIFiTCCBHGgAwIBAgIBSDANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0wOTAxMjYyMDE4NTlaFw0x
NTAxMjUyMDE4NTlaMFcxCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRIwEAYDVQQDEwlET0Qg
Q0EtMjIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCb/OGrH/FwNEUF
Xwn8HNfVJpPSkGmzHs7YElNwlEIM/KUuzn++aISDhCyPHeLfp9sF1SPzoYd41Cq+
MXVIwvcwa0sVJTyYC8cQLVXPKHazu0MgcqLDAWES3uquvdLklg567ZRhJPutmdri
ZhXN1bt374FPYS3PqatVGOhav4mNKc4gW0ATMVaSYEEGywqhM/5uS49bHV4pl+OB
9L3pBD3RMsagbcCThwEXQYcBwiMtsf6waQfIwp8TyoRt0f1yv76avWpgc1aIOsat
G8QXvQ0b41Jj/K/B+8wvbjXS3TrYENHEKLe2bP+T4PZy8CkTZws4PBkojWwZk0k9
Wz2XhNcdAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0jBBgwFoAU
SXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFCgwH1FRjtXdraHLIMJYFUYw
pkRPMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYDVR0gBIGX
MIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsKMAsGCWCG
SAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFlAwIBAwYw
DAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0wDAYKYIZI
AWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2EubWlsL2dl
dGNybD9Eb0QlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB7jA/Bggr
```

```
BgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9EbOQlMjBS
b290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCB
iAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNkRG9EJTIw
Um9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNkVS5TLiUy
MEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkwDQYJKoZI
hvcNAQEFBQADggEBAKfeVjVjzvm0/tj/uSwN7p62qFbVQf0mfmf8spCNq9k45ndV
zTeoXrnXvGMkh5HOu5e9m0jlOFfO+w4zbbSUme+5QdilGBYB7v/mvOz4BtHUwWoA
9u24b97jC5hUG4ABnc2hR88OM88oibJJ+nuG/J7iyZaeOLEfJLPMFAWyYzhRazlo
Sb+ZgnNZE+HdRtIq87pkCVGflrq6ZrO44ZwT9IbkQQsoet2V2nU3sK/4Z77xrDxH
7GLw0zYJc0UX+L4qFpu8fodFHMPZyetLJ81GrVe2vsA1qBL6EUjbxNrx6ur0D0D8
bteeV3V3vKwMl+xSDr6nmLV4fnzWxZ89fCOn/yU=
-----END CERTIFICATE-----
```

## 12.11.47 pkinit/DoD-Root2-CA23.crt

```
-----BEGIN CERTIFICATE-----
MIIFiTCCBHGgAwIBAgIBSzANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0wOTAxMjYxNjM4NDVaFw0x
NTAxMjUxNjM4NDVaMFcxCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRIwEAYDVQQDEwlET0Qg
Q0EtMjMwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDWp4YjHG0C2Jia
JH+l/1ujmJrrtdR/Hat6SUrtYZ/5yBZhuI/x/mlxLsVOYqUolgv6O1VmxkcB2Pcj
dzprs9+wNjLzXhRZ0eYf09wb0S8QJsWmFcGa9Bh7MYuXZOswxbACaTvaX4ex74r4
jv5fhur+hFquf6EXJrQCkVObfahVQk3+T+yOzZL14/OONJRSoMsUV3dloBX8SNEK
BpKJyu3rsnHHtyjgIJf9B1P7Ov88mrkcXKVPPllZo4tw151q8L371dL8n72Pp8jM
xKGlgSrKLpKQUMSIQ/Oql05U7aayiFntw5EQlG0PZDTE2g7Nc1FgDYfGmRlLUZSt
ZQLvDY3FAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0jBBgwFoAU
SXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFKpB9xKjHIMNK9eKPD3F/GxS
T81YMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYDVR0gBIGX
MIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsKMAsGCWCG
SAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFlAwIBAwYw
DAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAwOwDAYKYIZI
AWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2EubWlsL2dl
dGNybD9EbOQlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB7jA/Bggr
BgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9EbOQlMjBS
b290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCB
iAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNkRG9EJTIw
Um9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNkVS5TLiUy
MEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkwDQYJKoZI
hvcNAQEFBQADggEBAFBB2iPTjh4CXUh+1DFeZoCj8cv1sEq+6g9sYaRyRjlImVxD
6JNZOd+1GAFcMktQnD/UykP9YDJLlr2YXwxwndDcMy4+Te2VUq7iOJ5jf81sFHgA
dn9qcGye5KtYQgweLAdT3smkL42Ox71s3rOKgdtI48PirZRL38p5kzhpOKh8Nsxz
t9tPGRtHg+mLmjyqWw+H6x35qQPNpH5vpKOLGkp6rpbXsCZkmsl+8BcXuiRvjaeV
As79cvCZtR/0ggZj9lDUc/rIez4kApCKTR+mQxVVWRUIeg7PhljqgRAvks65VL7Y
lBPxzmqBR7rAToQy1HEeheokiRWXbapNrysMjnk=
-----END CERTIFICATE-----
```

## 12.11.48 pkinit/DoD-Root2-CA24.crt

```
-----BEGIN CERTIFICATE-----
MIIFiTCCBHGgAwIBAgIBRzANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0wOTAxMjYyMDIzMTFaFw0x
NTAxMjUyMDIzMTFaMFcxCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRIwEAYDVQQDEwlET0Qg
```

```
Q0EtMjQwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDCIK5JJXz7fDvS
Jt6L4UiWGj9ou3JeYNk27nSEPRY8/AfZ1w/lMLjtTBn4nBUKNWel+thm0yJR1G7B
5GBYAvH3e4dn6UENdAddCFcfWz1iqwQzNQxGOPqcuvo6v/lBwWfXsnpQ62e+5TYa
81E+fPz8//n/7dhKoG82PN8n7PL6FmFz7hxVVJdEbfbmVAdFS0ZrA+fMy0Yrch8T
JLVNv6bkZtX7Os0aMe9lLJIyyTM1bIxBBEHvNoO97zdN0YCd8tHizjlqPfpcScY0
a17h3eo9LmWpCTG68hJK2LbEMu4nBMpUso+TGLsmmQnsPHegCLjvlNGoxdraHBeA
dxWlBq4BAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0jBBgwFoAU
SXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFBZQF3XOO4qutQhFpKVw4PY3
tr5PMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYDVR0gBIGX
MIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsKMAsGCWCG
SAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFlAwIBAwYw
DAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0wDAYKYIZI
AWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2EubWlsL2dl
dGNybD9Eb0QlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB7jA/Bggr
BgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9Eb0QlMjBS
b290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCB
iAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNkRG9EJTIw
Um9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNkVS5TLiUy
MEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkwDQYJKoZI
hvcNAQEFBQADggEBALHKt07LwFOLLAkRmxMxFfY9uS9iRnYqrEtV6wzXzihrC5Wr
CjgWy9euzIexVbomJZpVqTPZ44nqjlMHASDk4Ww8edZdWwHgajrMgVPVxhVOieTD
FqQFQoxn48Z890aeFD3MvGviZEtzYGuMX7ybYioVSD0MU56AOejEqhpwEmLGwu1q
eUMvpJpjGktkN8JRb8o6lh4/S3kgL4RfdDMU5c7v11UusJEe5KGXuzrb2VqhAHIZ
wuHypW/cdXVZQ/LW8MqZdLRtRSSxn4CQPNdvWKE1y8NIUz+jNl407SiuOE2Gfssx
tbJtjV4qqP+Sw2T3FJNId9ynV4C7+GR/1WyaJqY=
-----END CERTIFICATE-----
```

### 12.11.49 pkinit/DoD-Root2-CA25.crt

```
-----BEGIN CERTIFICATE-----
MIIFiTCCBHGgAwIBAgIBTjANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0xMDAxMTQxNzMzMTJaFw0x
NjAxMTQxNzMzMTJaMFcxCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRIwEAYDVQQDEwlETOQg
Q0EtMjUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDh4mC3zlG3Zo5I
YPsLqLpaUdTYKmX00pbto1iZoomoVYaFYa0I/7MSFnXSOPoc7pqgYlqR4czhyQ01
fhdffxSsAXXkX0pfLSLLSdSCsRkXrm0yOClylwubbhXQIwh7LUtEu6EVyZZuptkU
AoicXt/5gjEURqiLAT7krq76U1A3VLpkU2ihoo98gJf5O/KP5fL/RviK7FglgHdG
YGG6bmA+H3o8pNcXDlefoy63QIqAtuPX189tARPygJNH87lpmwtWffeLQKhwwk6N
BBl+izlUIw+7ivB8d9XphFbMBbdDcv7vYkIHhUhPR0mC9BCWtLEjxegfe+qbkwv/
y+6EzDIHAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0jBBgwFoAU
SXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFC4LZfnWZd5LoyV1pKEuhSFA
c7kKMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYDVR0gBIGX
MIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsKMAsGCWCG
SAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFlAwIBAwYw
DAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0wDAYKYIZI
AWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2EubWlsL2dl
dGNybD9Eb0QlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB7jA/Bggr
BgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9Eb0QlMjBS
b290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCB
iAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNkRG9EJTIw
Um9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNkVS5TLiUy
MEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkwDQYJKoZI
hvcNAQEFBQADggEBAJYLyBa7kmLoEo1gJeYSrHexk5alqlj4H4Az1cx+LSyxhE4r
1kUPDTi1OqaInVmu8M6lesX47p1546dzXmy7uKtkSLw7uloaXVVTmPRoVI41uCMH
tqR8dcUUoyKenxG2FjCRLNieoAKsouHHg0Hhwc1ihFg3kQNcOFgwHBFh0gFJhGrg
```

```
cQR0u5RwevnwzzsW6Xm1C6IFwnID5d9g0mRyswMGQBLR0wujC55CbbDrlUeaNkaC
JGVT1bwWCF8g7ldcAiTZx9QWvEuIGDrMDCojcX0IwX/2svETp+2CTuwL4R0uwjWB
QNU0ntd5GN0+Zw9DsHbSqM56bXf6J8lYrbFp2hc=
-----END CERTIFICATE-----
```

### 12.11.50 pkinit/DoD-Root2-CA26.crt

```
-----BEGIN CERTIFICATE-----
MIIFiTCCBHGgAwIBAgIBUDANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0xMDAxMTQxNzM4MDVaFw0x
NjAxMTQxNzM4MDVaMFcxCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRIwEAYDVQQDEwlET0Qg
Q0EtMjYwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDC5HG6/0QfpRHl
jlgaNX4EQnz7V0H0XiWj8APAq2wrPgCLH8qNRhMRF00V6ZDm6Z3X09KN5pdWvFxo
rv8f6UwuRkEGto0NMexzQSIHd+5Evjtgs0KUZEfvJF/FurbcQzEEz8HaXy09cJVc
P6ZYK14YrNGQ09atVhBbJ0DrkMJMfKsXZsIpliN1fwwLA0fnC/ko8pXTqW+dKE9i
6mn0jAZIf8ocKUQ1czZK6J571DfPmpM8U1TmHJ0173lpdEQIak3vEtgvY6+Zy0U7
igl0FC/N+14mYGhhIIJXcRRvJTw9rw/aN5pt/KZFjL612+KUC9BHwrZUozKaafoi
N9Ta0ziZAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0jBBgwFoAU
SXS7DF66ev4CV097oMaVxgmAcJYwHQYDVR0OBBYEFGpfufR6NizidfC7ZDLB8bRM
pSz9MAwGA1UdJAQFMA0AAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYDVR0gBIGX
MIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsKMAsGCWCG
SAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFlAwIBAwYw
DAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0wDAYKYIZI
AWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRw0i8vY3JsLmRpc2EubWlsL2dl
dGNybD9Eb0QlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB7jA/Bggr
BgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9Eb0QlMjBS
b290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRw0i8vb2NzcC5kaXNhLm1pbDCB
iAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNkRG9EJTIw
Um9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNkVS5TLiUy
MEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkwDQYJKoZI
hvcNAQEFBQADggEBAHo+bKwGz/Juy/3tsGjSwpb04zw3EC1mlacdVmkdSiYppS9V
5j/TsDJFjRSh23WkbZj8bvXKftjKzlKkhQGYnRkiYFrKwi71IhMLGK1rxhzy2aaS
tPuQBxivQpsrUCrFLQPoBiyf9nkeiU0t0XYgX8iYqN40YQosvgoEXjZ1z21rBe0q
XqMMcpDMmM4s+amXG8X838AspZA5rKCvY9xjhqrMHT/n22LaEgtjPENJ+AU5VS3G
gJZRAWRRXMsmeuq2qCmA4nfC6IwWcoV9b440pV9QvcN0jfV6fcjWYa7c+kgSVBId
SF6W80X7qKF1YUxWgi2I1xi5CVW/sX5ZlMIsYJM=
-----END CERTIFICATE-----
```

### 12.11.51 pkinit/DoD-Root2-CA27.crt

```
-----BEGIN CERTIFICATE-----
MIIFTDCCBDSgAwIBAgICAbIwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTEwOTA4MTU1MDI1WhcN
MTcwOTA4MTU1MDI1WjBXMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTESMBAGA1UEAxMJRE9E
IENBLTI3MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAloQI/Xq6tpSD
0J07GQvPBN+IKp64GljrhyIqYzp/0cNra+e8GqgRAvVhzQGkmHVzxheMiTxCx+K0
yYmxqP+fngq7aN663rYRAZRDdJy9z+G+4M4cbWp8fH9i6F7aNuqUxQaLRojiwMIk
CQVQf/PZ5RIFXtLXzjXCe0c1GBVXzcWc9+kxeiqM0fE1ji6hUJFAN6K0ks6MVrf8
7C92PILewMi7R9Z+96koXCkelgCtJ4Z0hLQEuqdVFmk0k9S8jGJNT1aCtse0eC99
2dND7XMm8VPu/7PVs0Rutr4tG2gNd1iEV0LPQHCMvMrA0XV09xmIfGvsKcZkHehT
UL2UquSD3QIDAQABo4ICHDCCAhgwDgYDVR0PAQH/BAQDAgGGMB8GA1UdIwQYMBaA
FEl0uwxeunr+AlTve6DGlcYJgHCWMB0GA1UdDgQWBBQbBARARV59K14LzJllTf0k
```

```
3pB3FzASBgNVHRMBAf8ECDAGAQH/AgEAMAwGA1UdJAQFMAOAAQAwZgYDVR0gBF8w
XTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETALBglghkgB
ZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUDAgEDGzA3
BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9ET0RST09U
Q0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5odHRwOi8v
Y3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAGCCsGAQUF
BzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNsZGFwOi8v
Y3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIlMmNvdSUz
ZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJjYyUzZFVT
P2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUFAAOCAQEA
QMAMS5+NI6Yx0TSunpFcX4TdnASVYc2AFB9u3oiXf0mvpfX3cCfhEESTCjGkCaaf
cEsikE6/Fv9iT9f0gjVCFwfutabLa4S3Gm4XHEUXTNHNz+XcDNfF9sa6LpyzO29c
Fl5DbCVKnPWw7/mqE866pj2yUNx0LLgUFXm95h2RHWgaPhW1B3dR0V8DrAGm0mbo
CtevM6EzlQht/IiSkvVy+i6PkzGkvykjoyFTuj6wdSDkFUA6WMjIHt3LdRC1QQHm
Flpo9zc51hG2MCX7tH7IEsbYn2Op2W2G0jZHoXMpIv4C9GMrKSCXrrU0vFjmYLYR
14Km5+I3fjt7Bngmb2xFcw==
-----END CERTIFICATE-----
```

## 12.11.52 pkinit/DoD-Root2-CA28.crt

```
-----BEGIN CERTIFICATE-----
MIIFTDCCBDSgAwIBAgICAbMwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTEwOTA4MTU1NzAxWhcN
MTcwOTA4MTU1NzAxWjBXMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTESMBAGA1UEAxMJRE9E
IENBLTI4MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAq0tVmtGDxkU9
0AMvq/GYqzTDYcvkz/8zTQwW3ox7pCPGBMjT19KYLsJ8PXnf7PNFFE/wqRUz4dGm
PBaEmTwulHhndzESz6bBeBnTxMz8tvkcVU8flKfPshsnWW+n53gZyT54TVJI0iEY
5x0iaLv+eD21Ci+w7HqC6dhl/fDbPaTzXjj9Tes3+gIALFT/ebXLnjDu10E88T0+
9hIaNQRTTTQXcf9kuTgU1ndHVy23rM/hN1Ak7tHKP0TX6frS4EM1aY3mUJXf8Uhv
JFysVUfS43WzNVEKIrbyg4+icb5Aubr0S3pzevnzQK1f25gd5hN/39okUWvY0IvS
n4VhofbSPwIDAQABo4ICHDCCAhgwDgYDVR0PAQH/BAQDAgGGMB8GA1UdIwQYMBaA
FEl0uwxeunr+AlTve6DGlcYJgHCWMB0GA1UdDgQWBBQmtK6qLY7pjYpvtrVbnep0
rrGcaTASBgNVHRMBAf8ECDAGAQH/AgEAMAwGA1UdJAQFMAOAAQAwZgYDVR0gBF8w
XTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETALBglghkgB
ZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUDAgEDGzA3
BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9ET0RST09U
Q0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5odHRwOi8v
Y3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAGCCsGAQUF
BzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNsZGFwOi8v
Y3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIlMmNvdSUz
ZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJjYyUzZFVT
P2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUFAAOCAQEA
aLPRSRGrW929tB3qWt6nuM+H2qiQxSNJ0EknveoAK/x+HYvnm05esuPPg+GIg7FL
J2FE1aRb/wfFyELUdjxw4DghDI0Ty+8X0yPSH91NhDaMpodZBPvZLh5pnlZ0SLTG
UrEKfS4QXGWC/AyGYXCpxTpGYF9tvoGIZt+zxI6Zm7D8Pd7B2owbRCDUo3rAABBH
cIlEFWyLV+p+tY940BRzzD+VkexbrVNwTHn1gSGY0X6v0LE4h85w3iMjECX4GorR
RmqlTsZ39egCg+vcPzJZUiZsAGlkJZAVCZbj3mSxfKCcwNP+6+mMe6WMlLEBBGcz
BHlJWGqmQelYJ1aVmAP37g==
-----END CERTIFICATE-----
```

## 12.11.53 pkinit/DoD-Root2-CA29.crt

```
-----BEGIN CERTIFICATE-----
MIIFTDCCBDSgAwIBAgICAbQwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
```

```
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTEwOTA4MTU1ODI2WhcN
MTcwOTA4MTU1ODI2WjBXMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTESMBAGA1UEAxMJRE9E
IENBLTI5MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAu0oQdhk0QB5y
K2GQMojwO3UcPj1qdzqXeQvs6FEPfmoAQ5jE5qHgRVstA/pmaKWj0OCSgT30d20D
29sl5nZglD/2X99aGEHi8zdI8Zf+Kq1E5/wx4xb4vJp1pf0vVpqCSTrNTU9wzT9/
ABEWgwquV31pBIOg83fKBcH/+4XfmDj0+4ATPTh3b84MmxhTvNj0JP88upuwK8fi
kNH9A/M478xSw37jemyhSBFo7gA4Tco4fDA9h43uICQGBF9cINEFxoC/CUnga5rE
Cy2rYXZeHHDZgsiLDjODGXAuKvF+6RlK+wgek83zV3e4F+VznzWjNF5ViHUnYNAV
pgXDFvZLpQIDAQABo4ICHDCCAhgwDgYDVR0PAQH/BAQDAgGGMB8GA1UdIwQYMBaA
FEl0uwxeunr+AlTve6DGlcYJgHCWMB0GA1UdDgQWBBSbxZQ/72FhV6jh/lmu4mkY
2xhX3jASBgNVHRMBAf8ECDAGAQH/AgEAMAwGA1UdJAQFMAOAAQAwZgYDVR0gBF8w
XTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETALBglghkgB
ZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUDAgEDGzA3
BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9ET0RST09U
Q0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5odHRwOi8v
Y3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAGCCsGAQUF
BzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNsZGFwOi8v
Y3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIlMmNvdSUz
ZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJjYyUzZFVT
P2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUFAAOCAQEA
LnpHz7S3uaNWil4y66XY1lTD4zpvwsWIFK24CPUFqE1XOLxF5IYPgxx7SIY+ggbz
iNtUKMk0hemfA7ytTA+eBlX23xMC3WWtCTgB6GIwq5neg1Fn7WirugBkPffaWT4N
E1/e3Cl738wjW5wCTWqZsDlxg6QrcSzTxoThFQhjF9gFTVQ+Ty3nLZojNghSaVtv
DnHvqJAvMT5zrvR60pnfJIQH83Fvx/g6elyaXBFa98g/Om9DtYg3ekGqGt8YFPCv
kP6iRWGy6ok3NKsNPOjA0AiJg1WlRwtLTDf1Kamgbx51qUa1cvxsniz6P96OlU09
u4gfB0hC0RC+AouNMDAIfA==
-----END CERTIFICATE-----
```

### 12.11.54 pkinit/DoD-Root2-CA30.crt

```
-----BEGIN CERTIFICATE-----
MIIFTDCCBDSgAwIBAgICAbUwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTEwOTA4MTU1OTI0WhcN
MTcwOTA4MTU1OTI0WjBXMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTESMBAGA1UEAxMJRE9E
IENBLTMwMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzTyTZyYPHuaB
Xu6fzvlQHt1iohWEJeV3VsJTx6DgyUJTKuZOZ1I+cF3GaLgVZcjddtCy1ZrJqizj
xAkiBPd9iaSI2cKD7Fl7SRDvmo3Ihvlz3fIOYHqc2Y9Pd4N4DEtMLd7tn7GvHEMy
rLDQODpUniYPFEuNwW71JpUkN4ft7eDD1e/A8A119W+avv1kPCoirzgSK3MtDQl+
Eer8azJzTVzEWRfaxFmBBgS2CwLQZ7OWnHkTQxUkXsSV/VDRXgieH7ShlpI5K2is
vYw+hokuPrbrReC8HJsrC3jvbfEaYN3mR/h19PLKRKj7gFngUWOFC7b7Fizj8/9v
92q+m801gQIDAQABo4ICHDCCAhgwDgYDVR0PAQH/BAQDAgGGMB8GA1UdIwQYMBaA
FEl0uwxeunr+AlTve6DGlcYJgHCWMB0GA1UdDgQWBBQITtWkPCoEm5MbtwQIjnS5
BnwNozASBgNVHRMBAf8ECDAGAQH/AgEAMAwGA1UdJAQFMAOAAQAwZgYDVR0gBF8w
XTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETALBglghkgB
ZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUDAgEDGzA3
BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9ET0RST09U
Q0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5odHRwOi8v
Y3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAGCCsGAQUF
BzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNsZGFwOi8v
Y3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIlMmNvdSUz
ZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJjYyUzZFVT
P2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUFAAOCAQEA
mffXrLElu68fAzW/Vnv1oWCm2pTuj93MMtE1DZ/lXqZ0En8BKlozIcDXBsq/3Rtm
```

```
VE8CVfym32gX0r/0XWuO+chz21tUOt294WnZ+pHbKloPx46INQgjq2Rn298fa/yO
X3Kfl4GHgeWlIX3YT/4xm6F5pCZUQfBFkK9fQsEelof5z8ekGkRTkREOOIBktNkT
1iOOiMepsSAkVnwH+8R79PmcerUORLcyVzpNg5HEdRiUls9f9m82K65zGfjg/GnO
hn//QiE++TjDXnqZKN6YLLCciBCyNB6qCArLTgHFZOtNpafzCD0LenU6lkr3/c8c
r3JMcULZ/iO5WrStVwX9JA==
-----END CERTIFICATE-----
```

## 12.11.55 pkinit/DoD-Root2-CA31.crt

```
-----BEGIN CERTIFICATE-----
MIIFTDCCBDSgAwIBAgICA50wDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTMwMTE2MTQ0OTMwWhcN
MTkwMTE2MTQ0OTMwWjBXMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTESMBAGA1UEAxMJRE9E
IENBLTMxMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxicQL5CWONnf
5l8/uon7ZoLrtqXt8FaQFkDnbKKweWZZ15hiMdEzIlPjHlykVmamTVb7w+JCEqv5
wEpLQO+RE4Y5MFHWbo4nt0GJKQHuWEZzBHFEXGlDPjLmZN+za5kscKLQPk3YWBJt
RfA9k1S+3+L7zxH//IoBN++nLrpADGo+HOQKMoBpvSI57Et2ybFakzwhhDjdcxOC
+V0MgQqpslN02QuOwOiXuz1fE4y1uTvs9rudjiD2a7ydFDLcfrniY7BqwYC5FvyR
76yyCZ9SR1gTXmJ+mhKGW8UgH+GOZgB2U+znIokhTF+56b6gUpMOpsjezLeCrSJt
i9AwUzZVVwIDAQABo4ICHDCCAhgwHQYDVR0OBBYEFETjRqNB7mCxXqeTJfSgU+63
Sb67MB8GA1UdIwQYMBaAFElOuwxeunr+AlTve6DGlcYJgHCWMBIGA1UdEwEB/wQI
MAYBAf8CAQAwDAYDVR0kBAUwA4ABADAOBgNVHQ8BAf8EBAMCAYYwZgYDVR0gBF8w
XTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETALBglghkgB
ZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUDAgEDGzA3
BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9ET0RST09U
Q0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5odHRwOi8v
Y3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAGCCsGAQUF
BzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNsZGFwOi8v
Y3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIlMmNvdSUz
ZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJjYyUzZFVT
P2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUFAAOCAQEA
R1FS3PSgc5pC5wvsI5GNJXW0RII0qvlGdVHD9g745+MvtCDD76FlNOCdh8HmLmLw
J+jrxc81ldJAgIuSCbamG9USZDHbtdQO3wqKtlb1vHaSkxl8v2V9coHYZHs5NIp2
WMwdQ/cHzxyDA3O+OBfbdK1pCRF87djWAo1mPatryjPbx3pmxd6nJ0gPZhLuaCTA
75HqBhkqUFgT4CL8DrEk++uOQgIPd4gVi+by9VO3fOBVmxPWtnDKc3DjUyXBKB57
xCxJbpDbqstbAxvCh4f1q75RcXNtJmZ7mx0X4O3jwN4dJ7HtDTRGPt0uXvSCcNrR
kxt53dZK5875P3MfzormFg==
-----END CERTIFICATE-----
```

## 12.11.56 pkinit/DoD-Root2-CA32.crt

```
-----BEGIN CERTIFICATE-----
MIIFTDCCBDSgAwIBAgICA6EwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTMwMjA0MjA0NDA1WhcN
MTkwMjA0MjA0NDA1WjBXMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTESMBAGA1UEAxMJRE9E
IENBLTMyMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAs+KVHZM2LSWl
Dv146e/qk9E6ydhXvRnf0cei0ejZ/dKOFajdvT5k9Lb+nAPfS7Blt6sEGDIZbBMB
UtHmtchBEre+O8tNQBCIyp62/TV3bSb2ZK0RhwypJXpYn7C9mPaTXxvv77KXrfgV
59zmoGp1DVHfVR1oQVJJLsecaFdWR4/e9lIugW9WvAaJEpSfI70/gceGAnUwXj0h
3OETu/15VgE8Shn0LOuQZGTX6AovUYbVCJuE+/npi0LKZdKQBxyCl4xEI1cGLHVp
KHCy7T5M1eOWdxX9upXPW5ZpAnfWgNmPhynj5wV2r8qNEmA0cseznThuTJYynpA1
rXWLOWJACQIDAQABo4ICHDCCAhgwHQYDVR0OBBYEFC/Kk1MDrG919Xb6vv6O6hCL
```

```
t+eQMB8GA1UdIwQYMBaAFEl0uwxeunr+AlTve6DGlcYJgHCWMBIGA1UdEwEB/wQI
MAYBAf8CAQAwDAYDVR0kBAUwA4ABADAOBgNVHQ8BAf8EBAMCAYYwZgYDVR0gBF8w
XTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETALBglghkgB
ZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUDAgEDGzA3
BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9ET0RST09U
Q0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5odHRwOi8v
Y3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAGCCsGAQUF
BzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNsZGFwOi8v
Y3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIlMmNvdSUz
ZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJjYyUzZFVT
P2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUFAAOCAQEA
MI3VVmO9mQaLTbbSDgO5xoTSm3dBGojS/8Pa4uZnYb3Zeu04OV6rC1g0+droYnmv
OXLzSqfjTjkQzenSCOrUnpqnNTWTkwJZ4kwAHPP8ayFTSoxh52HL0EYL0T+cafXv
UIrwQLMrVloda2JZBbOPJxgFCkNbAu/dUl5bwKkcVuOVbJdPAYNWcl3XfVHjWlQu
uJj9ck4lj4sW0bDhM+OSfBBVMyRmrw8zBlNIA4eftGR0tdI9InK30Y43ERM5357n
0AwLilkRMmX/9rlGvT82nqeUAFfwwBnhLNxM9y9MkB1D764I43OeOr+Z7CK5B1iu
2TVSS1G7gTaPn24hCqaOhw==
-----END CERTIFICATE-----
```

## 12.11.57 pkinit/DoD-email-Root2-CA21.crt

```
-----BEGIN CERTIFICATE-----
MIIFjzCCBHegAwIBAgIBSjANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0wOTAxMjYxNjQxMTNaFw0x
NTAxMjUxNjQxMTNaMF0xCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRgwFgYDVQQDEw9ET0Qg
RU1BSUwgQ0EtMjEwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCa7Qjc
I6BER5v1w57JO9huz+v5xoNiegyD+Y84foLAjZzRQizLiA5iUSKpgBdYQXoMRps+
JahqKKm7Ev38hSvvs1sxT8oVxdO3mEVmBXL2CZpy6Sb/vZAmNQolvbusv9DWOId5
YSx7OQ7TKvUSP0DkmHNowkmsj9SMChevPkpEqT85DWm7Fg2Gjg7pvlN2eYMfXW6K
53HWRcGkzzJySODnEPmxC7XzdPBkGhNAlNITbbIJIVfh3akHV6a9wKSEV765HVFJ
H3xbxubSI/02VVeIyHlF22PPS2o7Mey1PV1nvLJXpS3V7fxM2DuH0UdzGHRvcFNC
hm4vwHgWwbm2MoclAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0j
BBgwFoAUSXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFFFnhBz/q6exnR2E
ZISZXdAL171bMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYD
VR0gBIGXMIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsK
MAsGCWCGSAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFl
AwIBAwYwDAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0w
DAYKYIZIAWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2Eu
bWlsL2dldGNybD9Eb0QlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB
7jA/BggrBgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9E
b0QlMjBSb290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNh
Lm1pbDCBiAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNk
RG9EJTIwUm9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNk
VS5TLiUyMEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkw
DQYJKoZIhvcNAQEFBQADggEBAHLEcRdOU0HK7vqTxTQS+kQV2uBjSdayG16jnh8h
b3iiDwrC2tS6ZscibJNNCUspZxSNmQ4FBvet3EfcDpkICi7yCZfo9SGFzvINNP+a
Q1+TdkqjDNgqIYsNYE52Hq0K7xO7NC4MFVY7tjF7Np85iIvLSLPZBE+fEVjl2a2Z
wBIoI5hw+p1IA2u8oNhOPbaRqaKIaIbCsUgTUtjAgJD4bOghISfjej7RspxknhiC
aDBXhAexdVqZOJIpa/0bMQa3l/rl6zqCZNVOebd2B7c0bqZJykLGIjuDsKQ42zSm
sJUkH8vxH7bA/3um3A/4/SW2sjLWdpkkS3fq/S3EYbmx/y4=
-----END CERTIFICATE-----
```

### 12.11.58 pkinit/DoD-email-Root2-CA22.crt

```
-----BEGIN CERTIFICATE-----
MIIFjzCCBHegAwIBAgIBRjANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0wOTAxMjYyMDI1MDdaFw0x
NTAxMjUyMDI1MDdaMF0xCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRgwFgYDVQQDEw9ET0Qg
RU1BSUwgQ0EtMjIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCmAAYl
I4D59WDEBgYoBUcuG2cfvrRF5Rw0xvTFutJMaJ1TJNeXv9joB894zproZMUQedNv
xx0zm4jDAPHwKfhT/eClMShoyS6MOAeSRbQ/CALL9+4BgS1fSxWx3YDyucD/qe2g
9Sebeex7JSslESmr/V8RPGKTl0J5SMCdBtG3IyWZV94GVcoeh5MU9xJDMdEmDm3S
RUw44tKa5xKvyUxd48h/H8fKCTnxCU/GoudhgXmZC9KMC2V6uTwYFc4Quy/AZBoy
CNGwoBKMEMuzbKRwQKy9VgtUpdTxRjPc7PZRUq8nJy6dVaQd911a+GRoQYlYvS93
nSjeDXhfiGf8HLyxAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0j
BBgwFoAUSXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFIUvDQqvNQhQCY2b
HHCsqP6Jd5RaMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYD
VR0gBIGXMIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsK
MAsGCWCGSAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFl
AwIBAwYwDAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0w
DAYKYIZIAWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2Eu
bWlsL2dldGNybD9Eb0QlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB
7jA/BggrBgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9E
b0QlMjBSb290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNh
Lm1pbDCBiAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNk
RG9EJTIwUm9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNk
VS5TLiUyMEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkw
DQYJKoZIhvcNAQEFBQADggEBADhU9UuJMePz9RcXpXSfyz+JU+9B6ldRxeTizrr6
QB+YTG5Day3PwBlu9BdH3ZaQRqZzL+3xJ3iHT7ftATRueWi/hclcZWy5e5gqip5d
YUAmvOSHNZ8D6s7JeQwGfmjenVXD0QoIf9jm5zqDVpfj4c0ybztEdrhzb0rwxyBM
jzFVgIZdHuY5RJmONKFp+W1fcg4FR2maCOxl2SmAn+CvfgEDuAvpE/dYIdYw/qDu
cnuBeYENlWCEPcpItgx7iXNfmF17Hg9pqgQmfGqRcP3zYthQT3l2umlW+r5uu4xX
b7HH/i7fhWXCshcUGRwWE/1+HW+yJ9YTHAxZkHC9VryuAoY=
-----END CERTIFICATE-----
```

### 12.11.59 pkinit/DoD-email-Root2-CA23.crt

```
-----BEGIN CERTIFICATE-----
MIIFjzCCBHegAwIBAgIBSTANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0wOTAxMjYxNjQzMjVaFw0x
NTAxMjUxNjQzMjVaMF0xCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRgwFgYDVQQDEw9ET0Qg
RU1BSUwgQ0EtMjMwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC99ClN
V5NEAweNBh+u+jw0VjRA57EYf2wlbheFBQUj6fbFUCVPgeQjMEJxaer3uJ73b6ze
Xar81uCNvGvufZmVIjuzWMaxUhyyqL8xQCIG/o0o0qlQVWoeB3D4pkjJbf2u7L6A
bD3PkNQHok6RFAO/V1kS9XTeQ5ZaWrnPuUfof9COsPjY6Us0XsxLF44C8BK/8gRs
HRO/qxzeDQnsy5tW7dmQ55alfyZlYcHEm2gkpc3SeSNvwzzBhR5I+T5QcWKgQbpy
RKVD46Vybs3Oq9rLhNIavx9uchE/LZkfbbD7BTDO5uwjKmVHH3icDZ9MJHVsdtLV
OxdEFrKKKEjXuQ2vAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0j
BBgwFoAUSXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFFuWLTGl4faaalxe
gVE2YR6WJBnRMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYD
VR0gBIGXMIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsK
MAsGCWCGSAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFl
AwIBAwYwDAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0w
DAYKYIZIAWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2Eu
bWlsL2dldGNybD9Eb0QlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB
```

```
7jA/BggrBgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9E
b0QlMjBSb290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNh
Lm1pbDCBiAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNk
RG9EJTIwUm9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNk
VS5TLiUyMEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkw
DQYJKoZIhvcNAQEFBQADggEBAD729uQP1FfNvu3pxEVS2kVbGKY2OTAKtn7r9on8
p2Iusz+DfeESGobTY5T0dPTOcZfq+8RqOl3imaesw+I2+Oh49NjEUO6KgVX6ioNl
DMXyTczpH497Rt0DCpzq4qxjdwfLMlTbCFWyWAB9XKa5FjxfZW0vBc5aP5rbScuS
o6HZb1HU9cAIwaM5W9BBY4HElGVkYylMXfBfcYdqnZaS5ceC/S101wJsyuLboLPb
cdUOhj4+F4m9bkIXG7T2OfkaveYuLJsONzsQX0T+e7WUWNZxJeU50SO6NADBQ224
9A6Xq7Iw9oinjo6KEAOEdNyuTfnlmXQaqaIKbQsHFZTFP2M=
-----END CERTIFICATE-----
```

### 12.11.60 pkinit/DoD-email-Root2-CA24.crt

```
-----BEGIN CERTIFICATE-----
MIIFjzCCBHegAwIBAgIBRTANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0wOTAxMjYyMDI2MTVaFw0x
NTAxMjUyMDI2MTVaMF0xCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRgwFgYDVQQDEw9ET0Qg
RU1BSUwgQ0EtMjQwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQClLlh6
od7mmlv2AvHV1Nw1I5p7bihkdBpwPJYdzMKfdAQ8DDmSIQgNEk6g1zeo0snGJ50o
+lXXshcEGc4yvPB5nvVoqy7MzzcEsvgKZZpJIBQlwbwSaqBCbRsItIehQiKrE5na
AgE5H14IV2tg3hN+aGp+QfWJgDh6/Zey0uKWSzaAYrbsJbvQD6ejzVGo99J5VZA0
JqPkXM27aCZ0CTeh5q/N5D6ZR/9/wke8ZYS6MimjDvDColt66rJKfQvGw26svRB/
T6l2OjOCASwqMLT3yKDSmDp8CNBaiQ+1ioL6DTAeftbRx7ZDJ7EoqQzjswd432Jk
mkWMTs2vDq6cWDbfAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0j
BBgwFoAUSXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFFSqcyrHs3fqzSJA
eUh7EfunmSKCMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYD
VR0gBIGXMIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsK
MAsGCWCGSAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFl
AwIBAwYwDAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0w
DAYKYIZIAWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2Eu
bWlsL2dldGNybD9Eb0QlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB
7jA/BggrBgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9E
b0QlMjBSb290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNh
Lm1pbDCBiAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNk
RG9EJTIwUm9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNk
VS5TLiUyMEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkw
DQYJKoZIhvcNAQEFBQADggEBAHIW3DlzY02T6Tccz7LtnNhN9wwySomes8q68wSs
cWYxpiq9un1U2C8JY0qICOhsE6HsXntWFzAtyNLt141HRGnPEW/L2OdSdbVRyKod
afAZHzDwB8c2vc4M3jt2/QrOy7YTutaFi/FcEpHKr+h/EqisLYvWdlCU7Db6ow/f
xjLqx3NG/IQami/E6CccSMJGNvYX7O1nMg+4ouC30l6QBhOUIWFDbH3zO2tl7ePb
qP/Fm7KS5+tf7u+/8zmMs/UX0obVw2xKOmw/nq/oWx02W6YmFUYLRmvH1ICq564c
uCtO+iFyn1+fga+07lvJlymJfOnceOJO4HSf0oZ4ZqLHmKg=
-----END CERTIFICATE-----
```

### 12.11.61 pkinit/DoD-email-Root2-CA25.crt

```
-----BEGIN CERTIFICATE-----
MIIFjzCCBHegAwIBAgIBTzANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0xMDAxMTQxNzM2MzJaFw0x
NjAxMTQxNzM2MzJaMF0xCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRgwFgYDVQQDEw9ET0Qg
```

```
RU1BSUwgQ0EtMjUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCiR3Pf
QKAVpqftoOnIf/vSjk8J5cWNXGQaQ37hc7mvEEV9My3qYCZzYiGLL66cF8zV5jih
Le4Cs/C53qaLiNG0Yz3IiIgcf15C6x2T86t46ZQdAz1NPhkXf08JIoL+w+Sfns3Z
vYKEOQxSt327QX/1jaQq9tBcYjHI4+q3t3jWm05iXrUS28p0XbhqEUNJFY05aWOP
TWLC8gR3WQSrBc6sFF6ZfR9Oi9TJope6ztCc4502/oyB/Gg5TZ4j4oOz06vg2d+Z
ZArINPKs4vVQnl5tOQ9FslLrTpJvH2nIoTcYbWHIhqrPxLfMN0n6fBtmcoFKRoxB
SqsgCnb3zhLUOAZVAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0j
BBgwFoAUSXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFCbb67FFLtgSkE31
EkH1w/AezODOMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYD
VR0gBIGXMIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsK
MAsGCWCGSAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFl
AwIBAwYwDAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0w
DAYKYIZIAWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2Eu
bWlsL2dldGNybD9EbOQlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB
7jA/BggrBgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9E
b0QlMjBSb290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNh
Lm1pbDCBiAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNk
RG9EJTIwUm9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNk
VS5TLiUyMEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkw
DQYJKoZIhvcNAQEFBQADggEBAGmQDW4i6bsFtubPOCSokDN5muLZYBEi1ewoR5Ag
N+KEhLJo9n3+i0sRuR2/28zc6XJ+RsmMSKk2hmCQmdr4WNty0KmObmDIvGDrBNAO
+HGF51vpwfvskpWqA2n9yDQFZdUCTO+ZgxrIRlw7/vhx2Hw7PVzRzYJMQ431Gqao
L0sCdNco1pFG0E1jja30lIiYIy0Ltu2OQE6G9Nnp0TZK1FPAS5bwsbhuQJxqMnxl
bbZg7YFKUFdTY2bod8d53HcjCz1jSm276E9DJM9tmFwR6C+IpTrlTsTY0P6cmOQy
rY4nFFWr2si3dkL7WRiSuAormmbMMPvEY2omt7eRHRiiPpw=
-----END CERTIFICATE-----
```

## 12.11.62 pkinit/DoD-email-Root2-CA26.crt

```
-----BEGIN CERTIFICATE-----
MIIFjzCCBHegAwIBAgIBUTANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0xMDAxMTQxNzM5MjdaFw0x
NjAxMTQxNzM5MjdaMF0xCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRgwFgYDVQQDEw9ET0Qg
RU1BSUwgQ0EtMjYwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCsXxBb
PSoH/e7PTsawTj1aoABgUHnCAkVmTzg0gVwcvydgannMppcL1onm2NuVfFc6B+5G
5WQqExePHTD8Lfo2fzdhJOUUov0iVxxhrk0uA1BmVUbdaif4qXmrXCrlJV1cG/tx
D7W4FY9flHsDz+6rkggK5L2joV2D1z3Hn9REEDqiX1/khpRvA6A184PY4bgZn3q6
dc8ABdDbI6RqJddpcEXGXXiLB19FrJ3Wo0tdGM+PTAoRodkR2/mcpdWPnOoPR70l
gpT5YJJKFPi6m6ls38oVEaGLOb76GU28uxRv3WB9spyQB3yAR7mFjLg+o3W5rl53
kXPBdYlVuk2G5K27AgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0j
BBgwFoAUSXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFDLfyG3z/z4p/ekM
lylQ8KIQLG4vMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYD
VR0gBIGXMIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsK
MAsGCWCGSAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFl
AwIBAwYwDAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0w
DAYKYIZIAWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2Eu
bWlsL2dldGNybD9Eb0QlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB
7jA/BggrBgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9E
b0QlMjBSb290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNh
Lm1pbDCBiAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNk
RG9EJTIwUm9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNk
VS5TLiUyMEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkw
DQYJKoZIhvcNAQEFBQADggEBAGrAUCmE+NxnE7GW7rpc1OWS+c1kZTOFKAugGqtT
HYxAF5G6Ztpra7ysjmEBw2c1EVlShXBdoYbnesEcw9hey3e7zFzcGt0EX/qI7bNu
tbREyzo1naBOHMBFtfbUzQZ50ho57CUmcZzZuG+TbNY7NDtnmapfpbhtTMcJ6snA
```

```
dJnZWYspiZArgZXZh/1V+Fh1UqZ/ImhthdZ9rooNLzS/1yhsxlutvP8bOsZkhaSc
fYSVn6gDZeR/TcwMdXpKBURYgIs5NE8zPytE8dZ07+98mtjcCxg98uWdUDsjKeX0
z2DqYE8cYMEaspxaAgSwfMHJFWrbKq8LCLs+cXqmPOUdRCI=
-----END CERTIFICATE-----
```

## 12.11.63 pkinit/DoD-email-Root2-CA27.crt

```
-----BEGIN CERTIFICATE-----
MIIFUjCCBDqgAwIBAgICAbYwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTEwOTA4MTYwMDE4WhcN
MTcwOTA4MTYwMDE4WjBdMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTEYMBYGA1UEAxMPRE9E
IEVNQUlMIENBLTI3MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA053C
B7D1fszurrirqjPqp5JuE1ZAaOUfxiG8wIGXYSxw0VoVF/6co+9IaYm3W1L5rOA6
nKZDViAogmzN9Zb+gJ3ZjfHR7oGavOzwhTUWQbkmiQwmekBu0AmAUcAC2O6Eb8ws
giKqNYVepF6FBNEJmaS4fVKxIXpN2CGnvERPyhWijDEuidY5L0BWN3jrLl0u0RhH
Fu2soITUC4KYvQMYcLAZXYxr3jUkYlrI+w+6euzIQElyVp4aTVTATuUQNE9h0dLt
Td/RWbDrAkIvDBtSDBWg8u66Nlf7zKwR8ZotTIspGPHwcJJo1kEmzFt8dXbYBWBS
Own8rcBAHKRG1jAtewIDAQABo4ICHDCCAhgwDgYDVR0PAQH/BAQDAgGGMB8GA1Ud
IwQYMBaAFEl0uwxeunr+AlTve6DGlcYJgHCWMB0GA1UdDgQWBBS/yO1EDrsz5sfK
QSylMbnJYGGJLjASBgNVHRMBAf8ECDAGAQH/AgEAMAwGA1UdJAQFMAOAAQAwZgYD
VR0gBF8wXTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETAL
BglghkgBZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUD
AgEDGzA3BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9E
T0RST09UQ0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5o
dHRwOi8vY3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAG
CCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNs
ZGFwOi8vY3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIl
MmNvdSUzZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJj
YyUzZFVTP2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUF
AAOCAQEAEX3yTl2o1kYa59nUZrFRxoHg5n89ca5Gp3ALg7S9wEAzUJuHQ5SHBWlO
vmdt3SrsCjEEv3iVb9ix7EMnCs8AgAEWPH2XN4WYW6aAcwyzd/7JcDNSi3p1t7ku
/rwtJUaW+kVteCjN25uZTAeeLGINitt/eFUFRxIb25kCN/lnHwQx7yiBd3ZaLpSL
dXg9icx40EsFmKLAcBaHcP+LfAnS4S0y7QPtYSuN2s7N0jzj5o/2ceO6L1yEgm6I
pl06q8Ft/mf56avlOETvQxvlKrEw+/T7b32kIABUYCI+XTNku5TqWnaVn8iPBms6
YOjCRiZTq7rmKMv5W469Q63xx2gyvg==
-----END CERTIFICATE-----
```

## 12.11.64 pkinit/DoD-email-Root2-CA28.crt

```
-----BEGIN CERTIFICATE-----
MIIFUjCCBDqgAwIBAgICAbcwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTEwOTA4MTYwMTE5WhcN
MTcwOTA4MTYwMTE5WjBdMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTEYMBYGA1UEAxMPRE9E
IEVNQUlMIENBLTI4MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAoL6S
zgqY0wjhxffuXYJK28/KZvS9cG6TG8qbOQFQGMDTVnrLWGfdBaTUKzFBYnoK/cL7
HoUsivnrqbOfs8papHhWVRdBl4n1zccwxt7IpRbq2OCGKBEMeA2dN+4Y+RYtX66E
bjSLukY79D16oz/jrpuph3Z9w7fgsi2COkF/uWnhUCKxv0xRakr5Aw4UtzKpX40b
71FXvIcpW2UmP/nzoZI4qNkxxxgRt+uKGNOQpc0JsrUs7wlpnsil12IiD9qF4Bqj
NLQYjKl1ScbHboSNqaSQX61brhOXXalfYA/cxJGSNgN7/WlZydb659zg/lo9XD/0
PwAbWF/TCUfvUHLIMQIDAQABo4ICHDCCAhgwDgYDVR0PAQH/BAQDAgGGMB8GA1Ud
IwQYMBaAFEl0uwxeunr+AlTve6DGlcYJgHCWMB0GA1UdDgQWBBRZiDBI5m3+YSem
xNWFjVtznu/BzTASBgNVHRMBAf8ECDAGAQH/AgEAMAwGA1UdJAQFMAOAAQAwZgYD
```

```
VR0gBF8wXTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETAL
BglghkgBZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUD
AgEDGzA3BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9E
T0RST09UQ0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5o
dHRwOi8vY3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAG
CCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNs
ZGFwOi8vY3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIl
MmNvdSUzZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJj
YyUzZFVTP2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUF
AAOCAQEAHWveFxw66X0qZH10AyB2ZE2foB7ZW0VKdzHMZSta6bZfXu42iBAU27d8
AEyxbTkJGXiMMml3qmSefHHXSbEsoN8nMVIqmYL01lNGSszA876YH2ATi+KKB2R+
hUyxCbHpWIrNmX4SwpNL1/WkFD7EewgwQ8gmfhf2U0m/au62A5LDAATJSQeJ8EGt
19/M1/MmhGJQshQ2ygsG0imA+Y0rpUSG4oEs7SADS0SvD5hBVMXAGIchy9WDTGaR
exYTV5GXdJK9AZUoe07i2tZWIDbSy0Z9dMqK4/nWwEInSQP0PwUPqtilvzMuFg+u
HbH/yUvYcWuTxaH/ajtVXhk3XlsjyQ==
-----END CERTIFICATE-----
```

## 12.11.65 pkinit/DoD-email-Root2-CA29.crt

```
-----BEGIN CERTIFICATE-----
MIIFUjCCBDqgAwIBAgICAbgwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTEwOTA4MTYwMjE0WhcN
MTcwOTA4MTYwMjE0WjBdMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTEYMBYGA1UEAxMPRE9E
IEVNQUlMIENBLTI5MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAkmIv
QcIgABYGVWSfvaeIFW6Cm0jBhXe9AqsM2fErYIEBuj51cI4Spqc4hJCz6UCAEtxq
ylHNrS2GEMxEvA7FWDgZshQyJUFUWFxDDshscw/DDBgYFgSaUj2BonHOPDIAn3FV
uvjONnceIbcolOc9Pqb2wHoxYJEol3ciUPLGk26yG8VBxvmhN/sQv9pWpvtSTV+/
78SWdyjlMv/o4RjMQ1IYrI13mnJM6J0DXrCi7+Td0ufmp6ZSreGYCJZKQ8xzPUui
jYnv3IJMuEqAJGUrHpGC9QT2ch9XGEAX8DlRto/ziTtn91hOSrza+Q7BwAy98whx
+IMPyS6AlfSFDs6uqQIDAQABo4ICHDCCAhgwDgYDVR0PAQH/BAQDAgGGMB8GA1Ud
IwQYMBaAFEl0uwxeunr+AlTve6DGlcYJgHCWMB0GA1UdDgQWBBS4Q4NkIXrucIHe
pd4MYCiHeK5eeDASBgNVHRMBAf8ECDAGAQH/AgEAMAwGA1UdJAQFMAOAAQAwZgYD
VR0gBF8wXTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETAL
BglghkgBZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUD
AgEDGzA3BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9E
T0RST09UQ0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5o
dHRwOi8vY3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAG
CCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNs
ZGFwOi8vY3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIl
MmNvdSUzZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJj
YyUzZFVTP2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUF
AAOCAQEALGsseTXb8B4ch3ur4ehpajeL23pPVWBplS9TncbKQ7bUN5HWA11+WrG4
HfeegdOuUFQwpG9LLrsUGxeqXBDTlHoxOZakVHn16VYuVcMbFuqqAsjPUfcygSLG
NDqpzZqqSJPH6fseMn5xxHbwRVQSHVXqvVwyhzquk5pumSJfqFE17rJTYF/2T0W4
FoQdZVXNFcoQAR+p0pynV5Gj1+ewhj0t9Ik62Ml3cFDGbO/y65j4EKo92shcKa3O
uHNJTKGSu+btzbqCGmMhGWX0Bhm/g6pz5dMbsZj/Rd/7Scxz6OLnB5YAMel/2SQI
58pEekgGw0LYP/l5h6U3khaphCCSYw==
-----END CERTIFICATE-----
```

## 12.11.66 pkinit/DoD-email-Root2-CA30.crt

```
-----BEGIN CERTIFICATE-----
MIIFUjCCBDqgAwIBAgICAbkwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
```

```
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTEwOTA4MTYwMzA4WhcN
MTcwOTA4MTYwMzA4WjBdMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTEYMBYGA1UEAxMPRE9E
IEVNQUlMIENBLTMwMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA5iki
1BQm0ZgaUl7FhINzfsFgs7PQlL79HJRVv/aELJvJwHRz78zCmfKZyW3KFNN0/74Q
8vctv8u7BqPumFBBZQHhVyy2y+TKHKx+UjQ0sY4HJj4yNa+jYQrF5Qi2EnmMVMF6
6fFQH12D0mcwsynbHTpMOSFQ2BgsjQZ17mNyeGitYpx1pJQG0zJrEq8GBym+E6DA
p/AlT7f+H7dX4BgSjSFqFblaVPt3ZdhMP/W6PMA34QZ+wr6eI4wo0ZrXxmc413PJ
vQcdhW/VlQqa3No6TijwpesJ3+XbC81Hr4rNu2+UQ0NZnFCfyQ6pcQK53OlpgDqJ
O0UFIhgFhLUS8DzAgQIDAQABo4ICHDCCAhgwDgYDVR0PAQH/BAQDAgGGMB8GA1Ud
IwQYMBaAFEl0uwxeunr+AlTve6DGlcYJgHCWMB0GA1UdDgQWBBQ1YWYoCbxWJVuL
zL+BXmEsMDnTITASBgNVHRMBAf8ECDAGAQH/AgEAMAwGA1UdJAQFMAOAAQAwZgYD
VR0gBF8wXTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETAL
BglghkgBZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUD
AgEDGzA3BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9E
T0RST09UQ0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5o
dHRwOi8vY3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAG
CCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNs
ZGFwOi8vY3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIl
MmNvdSUzZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJj
YyUzZFVTP2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUF
AAOCAQEACohWHKVXJlpiy3XQ3YbFUuIv87wRZD+MLz4R/JhgQPKADSiCmmj+4EhL
J9M6CnuV9gMMgRSRQjpgbOIrUy3s3xGu9VQX8AH5lwenm6sL26yXiQnG7/kHNBYA
qH4RU558L6E4opl5OTRBbn24WDBWiJ7kqmRF2aBEYjq35THTkYDxGxCyZ3DVW6tZ
tFpIFkLEAkzabGjKUB0xvjeZx89TzEIpVsOdF8oD5xBa8Tk8HMz7G5cKJvMx3+Cr
XCSdnt44fQJRZ0b5k3CF7QpVwvTBaFqfCMkde5t23FTvOYwY5QxE7vcGsh/1y+YO
vdSh/9T5kQciUnm3wP3ssviF9ET7XA==
-----END CERTIFICATE-----
```

## 12.11.67 pkinit/DoD-email-Root2-CA31.crt

```
-----BEGIN CERTIFICATE-----
MIIFUjCCBDqgAwIBAgICA58wDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTMwMTE2MTQ1MjQzWhcN
MTkwMTE2MTQ1MjQzWjBdMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTEYMBYGA1UEAxMPRE9E
IEVNQUlMIENBLTMxMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA6K4C
LEBMOlLoi3OStHfnOEvA8KpKGFzH9zXDSvDwlnell74n78REIYDqFjS3MNFEOH8q
zgTGkWWpblB8yE7+vcC1SxbkOFIV27O391M98rEH25FmXcG38ndmxFGaY5QRSwId
DUt8swBHB3kY+nizkx/Udm2ZBMUeNkb8BjQL42hvHnyfLM9huEv/tN8Gn6BflF7r
Nf8JXTVAB/Kd7ZYJ2Xbq/m4x/sv0ResweEhobKEpPoZ9k0FK6ucMTOWRUCqlQ2a8
IsD8Gyzk8y9iHgTUIb+sHyZ3NdAdvOK7RsLy6+QUrviza7P6cTiwcSnt0Ysb1wIb
3srsfu6h3Eil8T6UqQIDAQABo4ICHDCCAhgwHQYDVR0OBBYEFIbxW2hv3TDzlIJo
1Ez3RB24ymiBMB8GA1UdIwQYMBaAFEl0uwxeunr+AlTve6DGlcYJgHCWMBIGA1Ud
EwEB/wQIMAYBAf8CAQAwDAYDVR0kBAUwA4ABADAOBgNVHQ8BAf8EBAMCAYYwZgYD
VR0gBF8wXTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETAL
BglghkgBZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUD
AgEDGzA3BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9E
T0RST09UQ0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5o
dHRwOi8vY3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAG
CCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNs
ZGFwOi8vY3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIl
MmNvdSUzZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJj
YyUzZFVTP2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUF
AAOCAQEAWTKtqsP435xknHEJNMG9vGMAHi3b7anICOO5GOSvyq4Uwd27+XODg1eO
lMmgqgMHzmecteUXWT8ouBc22rqNw5YRAWpQ1gbaaKRK0guFfM2I3/9ed+b1pEiR
```

```
0E6iZ2r4aO+qF0Xv2JYK3c/wPoe2v4g/01S+PhLOofkLbzLRVL+EWzWg2wdktavp
eR7i8qp0cueREvfHu27u5XSQECSLt+fNnIWQR+Tib38gvSy7g5YjTahM2H4uXhUp
uCV9VzULLRVUjKnc4OU3nahPIJWDK8USNj2oc+FOiEmlubv6CUooWjO55JJ5W3v4
pU/zyTTNmYywumB+n4Q+5jz6flrr5g==
-----END CERTIFICATE-----
```

## 12.11.68 pkinit/DoD-email-Root2-CA32.crt

```
-----BEGIN CERTIFICATE-----
MIIFUjCCBDqgAwIBAgICA6IwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTMwMjA0MjA0ODEyWhcN
MTkwMjA0MjA0ODEyWjBdMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTEYMBYGA1UEAxMPRE9E
IEVNQUlMIENBLTMyMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAo/qq
hsqKGhsDTnFtQbbZZZpu/zYqPwLTfJVliFqk969jt1LHGvu7lXMHQmGLSqZ76VYH
NhuqNwIgHKTO+7bQaav8OEzI2OZW96JefucxtO7B/81kv3mCQSt30vh9q0yP98Ye
PPiOLz0Ug9qSmAnYOMZaWTaLh6KJ3b5KXsvNtkd+QaYJVGxBlnRbBsPUwS5GfV42
342iRnGsSrrEsffJFwov3aPshCHPqAXqueMub59+fbsdFnVPkh0D5hE4mDZ6odQA
PK0QWK8VxzZL4zubTbW0kL6tq9PAhLP83BWICYwRUFAv5HDstwquSlPiNsQFboB1
Eo03RvJLDDgcSR+sgwIDAQABo4ICHDCCAhgwHQYDVR0OBBYEFAqwqjhWR3sWfb6r
k5a8VN2F++0sMB8GA1UdIwQYMBaAFEl0uwxeunr+AlTve6DGlcYJgHCWMBIGA1Ud
EwEB/wQIMAYBAf8CAQAwDAYDVR0kBAUwA4ABADAOBgNVHQ8BAf8EBAMCAYYwZgYD
VR0gBF8wXTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETAL
BglghkgBZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUD
AgEDGzA3BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9E
T0RST09UQ0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5o
dHRwOi8vY3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAG
CCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNs
ZGFwOi8vY3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIl
MmNvdSUzZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJj
YyUzZFVTP2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUF
AAOCAQEAD72PR/+5yb1D5c6+tfM5y0UWWaPftlIkPAlVS9m/lXq9dtngMIfNSqmj
LZ7ZKATGlq4BFIDQJVbxWANV79KoIlKrge8A/q/HSdKMIC6kcYH3JssOpW3VQXd7
LTO7m7N8nD89/8LuefKJChCMkHRdNGdwvgL+gEYZB859L5aoxBPQ758psTSpuYyl
iTSzjD5H+GaMkdHuq8HqcYXJX7Cp7tsA1DAqQs5XYxAiMKichkESXb5QfBP66yhz
X3IziV9/DWikPf0WJugKk/57H4aBgCe+Z3GGG33Hb7epcQHGY7NzfQFrMyLteYmK
DuZyAnM3P8sxge2k+wtqO1KEukz3jg==
-----END CERTIFICATE-----
```

## 12.11.69 pkinit/root/DoD-Root2-Root.crt

```
-----BEGIN CERTIFICATE-----
MIIDcDCCAligAwIBAgIBBTANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0wNDEyMTMxNTAwMTBaFw0y
OTEyMDUxNTAwMTBaMFsxCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRYwFAYDVQQDEw1EbOQg
Um9vdCBDQSAyMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAwCzB9o07
rP8/PNZxvrh0IgfscEEV/KtA4weqwcPYn/7aTDq/P8jYKHtLNgHArEUlw9IOCo+F
GGQQPRoTcCpvjtfcjZOzQQ84Ic2tq8I9KgXTVxE3Dc2MUfmT48xGSSGOFLTNyxQ+
OM1yMe6rEvJl6jQuVl3/7mN1y226kTT8nvP0LRy+UMRC31mI/2qz+qhsPctWcXEF
lrufgOWARVlnQbDrw61gpIB1BhecDvRD4JkOG/t/9bPMsoGCsf0ywbi+QaRktWA6
WlEwjM7eQSwZR1xJEGS5dKmHQa99brrBuKG/ZTE6BGf5tbuOkooAY7ix5ow4X4P/
UNU7ol1rshDMYwIDAQABoz8wPTAdBgNVHQ4EFgQUSXS7DF66ev4CVO97oMaVxgmA
cJYwCwYDVR0PBAQDAgGGMA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcNAQEFBQAD
```

```
ggEBAJiRjT+JyLv1wGlzKTs1rLqzCHY9cAmS6YREIQF9FHYb7lFsHY0VNy17MWn0
mkS4r0bMNPojywMnGdKDIXUr5+AbmSbchECV6KjSzPZYXGbvP0qXEIIdugqi3VsG
K52nZE7rLgE1pLQ/E61V5NVzqGmbEfGY8jEeb0DU+HifjpGgb3AEkGaqBiv04XqS
tX3h4NGW56E6LcyxnR8FRO2HmdNNGnA5wQQM5X7Z8a/XIA7xInolpHOZzD+kByeW
qKKV7YK5Ft0eC4fCwfKI9WLfaN/HvGlR7bFc3FRUKQ8JOZqsA8HbDE2ubwp6Fknx
v5HSOJTT9pUst2zJQraNypCNhdk=
-----END CERTIFICATE-----
```

## 12.11.70 tls/DoD-Class3-Root.crt

```
-----BEGIN CERTIFICATE-----
MIICZzCCAdCgAwIBAgIBBDANBgkqhkiG9w0BAQUFADBhMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEcMBoGA1UEAxMTRG9EIENMQVNTIDMgUm9vdCBDQTAeFw0wMDA1MTkxMzEz
MDBaFw0yMDA1MTQxMzEzMDBaMGExCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMu
IEdvdmVybm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRwwGgYDVQQD
ExNEb0QgQ0xBU1MgMyBSb290IENBMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQC1MP5kvurMbe2BLPd/6Rm6DmlqKOGpqcuVWB/x5pppU+CIP5HFUbljl6jmIYwT
XjY8qFf6+HAsTGrLvzCnTBbkMlz4ErBR+BZXjS+0TfouqJToKmHUVw1Hzm4sL36Y
Z8wACKu2lhY1woWR5VugCsdmUmLzYXWVF668KlYppeArUwIDAQABoy8wLTAdBgNV
HQ4EFgQUbJyl8FyPbUGNxBc7kFfCD6PNbf4wDAYDVR0TBAUwAwEB/zANBgkqhkiG
9w0BAQUFAAOBgQCvcUT5lyPMaGmMQwdBuoggsyIAQciYoFUczT9usZNcrfoYmrsc
c2/9JEKPh59Rz76Gn+nXikhPCNlplKw/5g8tlw8ok3ZPYt//oM1h+KaGDDE0INx/
L6j7Ob6V7jhZAmLB3mwVT+DfnbvkeXMk/WNklfdKqJkfSGWVx3u/eDLneg==
-----END CERTIFICATE-----
```

## 12.11.71 tls/DoD-Root2-CA21.crt

```
-----BEGIN CERTIFICATE-----
MIIFiTCCBHGgAwIBAgIBTDANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0wOTAxMjYxNjM1MDNaFw0x
NTAxMjUxNjM1MDNaMFcxCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRIwEAYDVQQDEwlET0Qg
Q0EtMjEwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDdlElknbLr9TVZ
5hTjI5zGC1inX0nBxgikNyl7IxR5CP4aLtpxFGKAL2NSlnuEl/bASHmxo0kIh9Ov
t49pTRAi4v5wXTyTCpxYXm8qXYH+HWI5LruZDgNan8bldy2IDWDMtIp3TF+b5qU/
pq8E6cxSnqyAZIOlaRXzVE30qAI6c5wWxEKFK0E3CUDEWCNPp0snxwdD5TgsDH/Y
A5WCCX+2mWhWhogD4dJUKnUXS2XK8xJFy5YQ7BPMG76bBFT7PFGbNH53jn35Mb0O
n3zoHjfLUk6IPecJvVgjAJbyvKcDtDXmDHZvaCMicq2Lt/f/Ju0tHrVZQA2o/a0n
H1Hkue1BAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0jBBgwFoAU
SXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFAmZE+Kj1ed02PY/tdz71LUW
7UzTMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYDVR0gBIGX
MIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsKMAsGCWCG
SAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFlAwIBAwYw
DAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0wDAYKYIZI
AWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2EubWlsL2dl
dGNybD9Eb0QlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB7jA/Bggr
BgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9Eb0QlMjBS
b290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCB
iAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNkRG9EJTIw
Um9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNkVS5TLiUy
MEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkwDQYJKoZI
hvcNAQEFBQADggEBACXuf0uCpdBRmSoj3P0tJyXAaXlIADIm0u5sHBy78MAM09gs
dFilVQlDolr5J/7YWujgqKS9vQWlC5UmHA4IiA7k+R97fphBDD0gjkTC8azehAGG
7DXs/4G7YH2Ot1byTJACH9OIP0khbowrvG8bQBlisuMUcL/RgEukcT8U7uD06R71
```

```
BYESPdT8AIOyH8IFLGMgCcJHnVsek3emIwsWY3Ba5M3eJSbcrVcIMSNmm5+cCRpU
/IlYa4P632JwHHr5MjX7w+jPBmrS2Tm6PY+uYHsqZgA5xVCpXkNNobwKsiT7EjZX
zfjKO19+y8URKtUEBftfWOdUB2epSQeOSlYTZks=
-----END CERTIFICATE-----
```

### 12.11.72 tls/DoD-Root2-CA22.crt

```
-----BEGIN CERTIFICATE-----
MIIFiTCCBHGgAwIBAgIBSDANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0wOTAxMjYyMDE4NTlaFw0x
NTAxMjUyMDE4NTlaMFcxCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRIwEAYDVQQDEwlET0Qg
Q0EtMjIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCb/OGrH/FwNEUF
Xwn8HNfVJpPSkGmzHs7YElNwlEIM/KUuzn++aISDhCyPHeLfp9sF1SPzoYd41Cq+
MXVIwvcwa0sVJTyYC8cQLVXPKHazu0MgcqLDAWES3uquvdLklg567ZRhJPutmdri
ZhXN1bt374FPYS3PqatVGOhav4mNKc4gW0ATMVaSYEEGywqhM/5uS49bHV4pl+OB
9L3pBD3RMsagbcCThwEXQYcBwiMtsf6waQfIwp8TyoRt0f1yv76avWpgc1aIOsat
G8QXvQ0b41Jj/K/B+8wvbjXS3TrYENHEKLe2bP+T4PZy8CkTZws4PBkojWwZk0k9
Wz2XhNcdAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0jBBgwFoAU
SXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFCgwH1FRjtXdraHLIMJYFUYw
pkRPMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYDVR0gBIGX
MIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsKMAsGCWCG
SAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFlAwIBAwYw
DAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0wDAYKYIZI
AWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2EubWlsL2dl
dGNybD9Eb0QlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB7jA/Bggr
BgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9Eb0QlMjBS
b290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCB
iAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNkRG9EJTIw
Um9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNkVS5TLiUy
MEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkwDQYJKoZI
hvcNAQEFBQADggEBAKfeVjVjzvm0/tj/uSwN7p62qFbVQf0mfmf8spCNq9k45ndV
zTeoXrnXvGMkh5HOu5e9mOjlOFfO+w4zbbSUme+5QdilGBYB7v/mvOz4BtHUwWoA
9u24b97jC5hUG4ABnc2hR88OM88oibJJ+nuG/J7iyZaeOLEfJLPMFAWyYzhRazlo
Sb+ZgnNZE+HdRtIq87pkCVGflrq6ZrO44ZwT9IbkQQsoet2V2nU3sK/4Z77xrDxH
7GLw0zYJc0UX+L4qFpu8fodFHMPZyetLJ81GrVe2vsA1qBL6EUjbxNrx6ur0D0D8
bteeV3V3vKwMl+xSDr6nmLV4fnzWxZ89fCOn/yU=
-----END CERTIFICATE-----
```

### 12.11.73 tls/DoD-Root2-CA23.crt

```
-----BEGIN CERTIFICATE-----
MIIFiTCCBHGgAwIBAgIBSzANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0wOTAxMjYxNjM4NDVaFw0x
NTAxMjUxNjM4NDVaMFcxCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRIwEAYDVQQDEwlET0Qg
Q0EtMjMwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDWp4YjHG0C2Jia
JH+l/1ujmJrrtdR/Hat6SUrtYZ/5yBZhuI/x/mlxLsVOYqUolgv6O1VmxkcB2Pcj
dzprs9+wNjLzXhRZ0eYf09wb0S8QJsWmFcGa9Bh7MYuXZOswxbACaTvaX4ex74r4
jv5fhur+hFquf6EXJrQCkVObfahVQk3+T+yOzZL14/0ONJRSoMsUV3dloBX8SNEK
BpKJyu3rsnHHtyjgIJf9B1P7Ov88mrkcXKVPPllZo4tw151q8L371dL8n72Pp8jM
xKGlgSrKLpKQUMSIQ/0qlO5U7aayiFntw5EQlG0PZDTE2g7Nc1FgDYfGmRlLUZSt
ZQLvDY3FAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0jBBgwFoAU
SXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFKpB9xKjHIMNK9eKPD3F/GxS
```

```
T81YMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYDVR0gBIGX
MIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsKMAsGCWCG
SAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFlAwIBAwYw
DAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0wDAYKYIZI
AWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2EubWlsL2dl
dGNybD9Eb0QlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB7jA/Bggr
BgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9EbOQlMjBS
b290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCB
iAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNkRG9EJTIw
Um9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNkVS5TLiUy
MEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkwDQYJKoZI
hvcNAQEFBQADggEBAFBB2iPTjh4CXUh+1DFeZoCj8cv1sEq+6g9sYaRyRjlImVxD
6JNZOd+1GAFcMktQnD/UykP9YDJLlr2YXwxwndDcMy4+Te2VUq7iOJ5jf81sFHgA
dn9qcGye5KtYQgweLAdT3smkL42Ox71s3rOKgdtI48PirZRL38p5kzhpOKh8Nsxz
t9tPGRtHg+mLmjyqWw+H6x35qQPNpH5vpKOLGkp6rpbXsCZkmsl+8BcXuiRvjaeV
As79cvCZtR/0ggZj9lDUc/rIez4kApCKTR+mQxVVWRUIeg7PhljqgRAvks65VL7Y
lBPxzmqBR7rAToQy1HEeheokiRWXbapNrysMjnk=
-----END CERTIFICATE-----
```

## 12.11.74 tls/DoD-Root2-CA24.crt

```
-----BEGIN CERTIFICATE-----
MIIFiTCCBHGgAwIBAgIBRzANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0wOTAxMjYyMDIzMTFaFw0x
NTAxMjUyMDIzMTFaMFcxCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRIwEAYDVQQDEwlET0Qg
Q0EtMjQwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDCIK5JJXz7fDvS
Jt6L4UiWGj9ou3JeYNk27nSEPRY8/AfZ1w/lMLjtTBn4nBUKNWel+thm0yJR1G7B
5GBYAvH3e4dn6UENdAddCFcfWz1iqwQzNQxGOPqcuvo6v/lBwWfXsnpQ62e+5TYa
81E+fPz8//n/7dhKoG82PN8n7PL6FmFz7hxVVJdEbfbmVAdFS0ZrA+fMy0Yrch8T
JLVNv6bkZtX7Os0aMe9lLJIyyTM1bIxBBEHvNoO97zdN0YCd8tHizjlqPfpcScY0
a17h3eo9LmWpCTG68hJK2LbEMu4nBMpUso+TGLsmmQnsPHegCLjvlNGoxdraHBeA
dxWlBq4BAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0jBBgwFoAU
SXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFBZQF3XOO4qutQhFpKVw4PY3
tr5PMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYDVR0gBIGX
MIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsKMAsGCWCG
SAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFlAwIBAwYw
DAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0wDAYKYIZI
AWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2EubWlsL2dl
dGNybD9EbOQlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB7jA/Bggr
BgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9Eb0QlMjBS
b290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCB
iAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNkRG9EJTIw
Um9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNkVS5TLiUy
MEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkwDQYJKoZI
hvcNAQEFBQADggEBALHKt07LwFOLLAkRmxMxFfY9uS9iRnYqrEtV6wzXzihrC5Wr
CjgWy9euzIexVbomJZpVqTPZ44nqjlMHASDk4Ww8edZdWwHgajrMgVPVxhVOieTD
FqQFQoxn48Z890aeFD3MvGviZEtzYGuMX7ybYioVSD0MU56AOejEqhpwEmLGwu1q
eUMvpJpjGktkN8JRb8o6lh4/S3kgL4RfdDMU5c7v11UusJEe5KGXuzrb2VqhAHIZ
wuHypW/cdXVZQ/LW8MqZdLRtRSSxn4CQPNdvWKE1y8NIUz+jNl407SiuOE2Gfssx
tbJtjV4qqP+Sw2T3FJNId9ynV4C7+GR/1WyaJqY=
-----END CERTIFICATE-----
```

### 12.11.75 tls/DoD-Root2-CA25.crt

```
-----BEGIN CERTIFICATE-----
MIIFiTCCBHGgAwIBAgIBTjANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0xMDAxMTQxNzMzMTJaFw0x
NjAxMTQxNzMzMTJaMFcxCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRIwEAYDVQQDEwlET0Qg
Q0EtMjUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDh4mC3zlG3Zo5I
YPsLqLpaUdTYKmX00pbto1iZoomoVYaFYa0I/7MSFnXSOPoc7pqgYlqR4czhyQ01
fhdffxSsAXXkX0pfLSLLSdSCsRkXrm0yOClylwubbhXQIwh7LUtEu6EVyZZuptkU
AoicXt/5gjEURqiLAT7krq76U1A3VLpkU2ihoo98gJf5O/KP5fL/RviK7FglgHdG
YGG6bmA+H3o8pNcXDlefoy63QIqAtuPX189tARPygJNH87lpmwtWffeLQKhwwk6N
BBl+izlUIw+7ivB8d9XphFbMBbdDcv7vYkIHhUhPROmC9BCWtLEjxegfe+qbkwv/
y+6EzDIHAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0jBBgwFoAU
SXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFC4LZfnWZd5LoyV1pKEuhSFA
c7kKMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYDVR0gBIGX
MIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsKMAsGCWCG
SAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFlAwIBAwYw
DAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0wDAYKYIZI
AWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2EubWlsL2dl
dGNybD9Eb0QlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB7jA/Bggr
BgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9Eb0QlMjBS
b290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCB
iAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNkRG9EJTIw
Um9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNkVS5TLiUy
MEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkwDQYJKoZI
hvcNAQEFBQADggEBAJYLyBa7kmLoEo1gJeYSrHexk5alqlj4H4Az1cx+LSyxhE4r
1kUPDTi1OqaInVmu8M6lesX47p1546dzXmy7uKtkSLw7uloaXVVTmPRoVI41uCMH
tqR8dcUUoyKenxG2FjCRLNieoAKsouHHg0Hhwc1ihFg3kQNcOFgwHBFh0gFJhGrg
cQR0u5RwevnwzzsW6Xm1C6IFwnID5d9g0mRyswMGQBLR0wujC55CbbDrlUeaNkaC
JGVT1bwWCF8g7ldcAiTZx9QWvEuIGDrMDCojcX0IwX/2svETp+2CTuwL4ROuwjWB
QNUOntd5GN0+Zw9DsHbSqM56bXf6J8lYrbFp2hc=
-----END CERTIFICATE-----
```

### 12.11.76 tls/DoD-Root2-CA26.crt

```
-----BEGIN CERTIFICATE-----
MIIFiTCCBHGgAwIBAgIBUDANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0xMDAxMTQxNzM4MDVaFw0x
NjAxMTQxNzM4MDVaMFcxCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRIwEAYDVQQDEwlET0Qg
Q0EtMjYwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDC5HG6/0QfpRHl
jlgaNX4EQnz7VOHOXiWj8APAq2wrPgCLH8qNRhMRF0OV6ZDm6Z3X09KN5pdWvFxo
rv8f6UwuRkEGtoONMexzQSIHd+5Evjtgs0KUZEfvJF/FurbcQzEEz8HaXyO9cJVc
P6ZYK14YrNGQ09atVhBbJ0DrkMJMfKsXZsIpliN1fwwLAOfnC/ko8pXTqW+dKE9i
6mn0jAZIf8ocKUQ1czZK6J571DfPmpM8U1TmHJO173lpdEQIak3vEtgvY6+Zy0U7
igl0FC/N+14mYGhhIIJXcRRvJTw9rw/aN5pt/KZFjL612+KUC9BHwrZUozKaafoi
N9TaOziZAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0jBBgwFoAU
SXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFGpfufR6NizidfC7ZDLB8bRM
pSz9MAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYDVR0gBIGX
MIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsKMAsGCWCG
SAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFlAwIBAwYw
DAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0wDAYKYIZI
AWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2EubWlsL2dl
dGNybD9Eb0QlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB7jA/Bggr
```

```
BgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9Eb0QlMjBS
b290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCB
iAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNkRG9EJTIw
Um9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNkVS5TLiUy
MEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkwDQYJKoZI
hvcNAQEFBQADggEBAHo+bKwGz/Juy/3tsGjSwpb04zw3EC1mlacdVmkdSiYppS9V
5j/TsDJFjRSh23WkbZj8bvXKftjKzlKkhQGYnRkiYFrKwi71IhMLGK1rxhzy2aaS
tPuQBxivQpsrUCrFLQPoBiyf9nkeiU0tOXYgX8iYqN40YQosvgoEXjZ1z21rBe0q
XqMMcpDMmM4s+amXG8X838AspZA5rKCvY9xjhqrMHT/n22LaEgtjPENJ+AU5VS3G
gJZRAWRRXMsmeuq2qCmA4nfC6IwWcoV9b440pV9QvcN0jfV6fcjWYa7c+kgSVBId
SF6W8OX7qKF1YUxWgi2I1xi5CVW/sX5ZlMIsYJM=
-----END CERTIFICATE-----
```

## 12.11.77 tls/DoD-Root2-CA27.crt

```
-----BEGIN CERTIFICATE-----
MIIFTDCCBDSgAwIBAgICAbIwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTEwOTA4MTU1MDI1WhcN
MTcwOTA4MTU1MDI1WjBXMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTESMBAGA1UEAxMJRE9E
IENBLTI3MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAloQI/Xq6tpSD
0J07GQvPBN+IKp64GljrhyIqYzp/0cNra+e8GqgRAvVhzQGkmHVzxheMiTxCx+K0
yYmxqP+fngq7aN663rYRAZRDdJy9z+G+4M4cbWp8fH9i6F7aNuqUxQaLRojiwMIk
CQVQf/PZ5RIFXtLXzjXCe0c1GBVXzcWc9+kxeiqM0fE1ji6hUJFAN6K0ks6MVrf8
7C92PILewMi7R9Z+96koXCkelgCtJ4Z0hLQEuqdVFmk0k9S8jGJNT1aCtse0eC99
2dND7XMm8VPu/7PVsORutr4tG2gNd1iEV0LPQHCMvMrAOXVO9xmIfGvsKcZkHehT
UL2UquSD3QIDAQABo4ICHDCCAhgwDgYDVR0PAQH/BAQDAgGGMB8GA1UdIwQYMBaA
FEl0uwxeunr+AlTve6DGlcYJgHCWMB0GA1UdDgQWBBQbBARARV59K14LzJllTf0k
3pB3FzASBgNVHRMBAf8ECDAGAQH/AgEAMAwGA1UdJAQFMAOAAQAwZgYDVR0gBF8w
XTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETALBglghkgB
ZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUDAgEDGzA3
BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9ET0RST09U
Q0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5odHRwOi8v
Y3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAGCCsGAQUF
BzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNsZGFwOi8v
Y3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIlMmNvdSUz
ZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJjYyUzZFVT
P2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUFAAOCAQEA
QMAMS5+NI6Yx0TSunpFcX4TdnASVYc2AFB9u3oiXf0mvpfX3cCfhEESTCjGkCaaf
cEsikE6/Fv9iT9f0gjVCFwfutabLa4S3Gm4XHEUXTNHNz+XcDNfF9sa6LpyzO29c
Fl5DbCVKnPWw7/mqE866pj2yUNx0LLgUFXm95h2RHWgaPhW1B3dR0V8DrAGm0mbo
CtevM6EzlQht/IiSkvVy+i6PkzGkvykjoyFTuj6wdSDkFUA6WMjIHt3LdRC1QQHm
Flpo9zc51hG2MCX7tH7IEsbYn2Op2W2G0jZHoXMpIv4C9GMrKSCXrrU0vFjmYLYR
14Km5+I3fjt7Bngmb2xFcw==
-----END CERTIFICATE-----
```

## 12.11.78 tls/DoD-Root2-CA28.crt

```
-----BEGIN CERTIFICATE-----
MIIFTDCCBDSgAwIBAgICAbMwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTEwOTA4MTU1NzAxWhcN
MTcwOTA4MTU1NzAxWjBXMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTESMBAGA1UEAxMJRE9E
IENBLTI4MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAq0tVmtGDxkU9
```

```
0AMvq/GYqzTDYcvkz/8zTQwW3ox7pCPGBMjT19KYLsJ8PXnf7PNFFE/wqRUz4dGm
PBaEmTwulHhndzESz6bBeBnTxMz8tvkcVU8flKfPshsnWW+n53gZyT54TVJI0iEY
5x0iaLv+eD21Ci+w7HqC6dhl/fDbPaTzXjj9Tes3+gIALFT/ebXLnjDu10E88T0+
9hIaNQRTTTQXcf9kuTgU1ndHVy23rM/hN1Ak7tHKP0TX6frS4EM1aY3mUJXf8Uhv
JFysVUfS43WzNVEKIrbyg4+icb5Aubr0S3pzevnzQK1f25gd5hN/39okUWvY0IvS
n4VhofbSPwIDAQABo4ICHDCCAhgwDgYDVR0PAQH/BAQDAgGGMB8GA1UdIwQYMBaA
FEl0uwxeunr+AlTve6DGlcYJgHCWMB0GA1UdDgQWBBQmtK6qLY7pjYpvtrVbnep0
rrGcaTASBgNVHRMBAf8ECDAGAQH/AgEAMAwGA1UdJAQFMAOAAQAwZgYDVR0gBF8w
XTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETALBglghkgB
ZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUDAgEDGzA3
BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9ET0RST09U
Q0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5odHRwOi8v
Y3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAGCCsGAQUF
BzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNsZGFwOi8v
Y3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIlMmNvdSUz
ZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJjYyUzZFVT
P2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUFAAOCAQEA
aLPRSRGrW929tB3qWt6nuM+H2qiQxSNJ0EknveoAK/x+HYvnm05esuPPg+GIg7FL
J2FE1aRb/wfFyELUdjxw4DghDI0Ty+8X0yPSH91NhDaMpodZBPvZLh5pnlZ0SLTG
UrEKfS4QXGWC/AyGYXCpxTpGYF9tvoGIZt+zxI6Zm7D8Pd7B2owbRCDUo3rAABBH
cIlEFWyLV+p+tY940BRzzD+VkexbrVNwTHn1gSGY0X6v0LE4h85w3iMjECX4GorR
RmqlTsZ39egCg+vcPzJZUiZsAGlkJZAVCZbj3mSxfKCcwNP+6+mMe6WMlLEBBGcz
BHlJWGqmQelYJ1aVmAP37g==
-----END CERTIFICATE-----
```

## 12.11.79 tls/DoD-Root2-CA29.crt

```
-----BEGIN CERTIFICATE-----
MIIFTDCCBDSgAwIBAgICAbQwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTEwOTA4MTU1ODI2WhcN
MTcwOTA4MTU1ODI2WjBXMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTESMBAGA1UEAxMJRE9E
IENBLTI5MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAu0oQdhk0QB5y
K2GQMojw03UcPj1qdzqXeQvs6FEPfmoAQ5jE5qHgRVstA/pmaKWj0OCSgT30d20D
29sl5nZglD/2X99aGEHi8zdI8Zf+Kq1E5/wx4xb4vJp1pf0vVpqCSTrNTU9wzT9/
ABEWgwquV31pBIOg83fKBcH/+4XfmDj0+4ATPTh3b84MmxhTvNj0JP88upuwK8fi
kNH9A/M478xSw37jemyhSBFo7gA4Tco4fDA9h43uICQGBF9cINEFxoC/CUnga5rE
Cy2rYXZeHHDZgsiLDjODGXAuKvF+6RlK+wgek83zV3e4F+VznzWjNF5ViHUnYNAV
pgXDFvZLpQIDAQABo4ICHDCCAhgwDgYDVR0PAQH/BAQDAgGGMB8GA1UdIwQYMBaA
FEl0uwxeunr+AlTve6DGlcYJgHCWMB0GA1UdDgQWBBSbxZQ/72FhV6jh/lmu4mkY
2xhX3jASBgNVHRMBAf8ECDAGAQH/AgEAMAwGA1UdJAQFMAOAAQAwZgYDVR0gBF8w
XTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETALBglghkgB
ZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUDAgEDGzA3
BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9ET0RST09U
Q0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5odHRwOi8v
Y3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAGCCsGAQUF
BzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNsZGFwOi8v
Y3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIlMmNvdSUz
ZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJjYyUzZFVT
P2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUFAAOCAQEA
LnpHz7S3uaNWil4y66XY1lTD4zpvwsWIFK24CPUFqE1XOLxF5IYPgxx7SIY+ggbz
iNtUKMk0hemfA7ytTA+eBlX23xMC3WWtCTgB6GIwq5neg1Fn7WirugBkPffaWT4N
E1/e3Cl738wjW5wCTWqZsDlxg6QrcSzTxoThFQhjF9gFTVQ+Ty3nLZojNghSaVtv
DnHvqJAvMT5zrvR60pnfJIQH83Fvx/g6elyaXBFa98g/Om9DtYg3ekGqGt8YFPCv
kP6iRWGy6ok3NKsNPOjA0AiJg1WlRwtLTDf1Kamgbx51qUa1cvxsniz6P96OlU09
u4gfB0hC0RC+AouNMDAIfA==
-----END CERTIFICATE-----
```

### 12.11.80 tls/DoD-Root2-CA30.crt

```
-----BEGIN CERTIFICATE-----
MIIFTDCCBDSgAwIBAgICAbUwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTEwOTA4MTU1OTI0WhcN
MTcwOTA4MTU1OTI0WjBXMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTESMBAGA1UEAxMJRE9E
IENBLTMwMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzTyTZyYPHuaB
Xu6fzvlQHt1iohWEJeV3VsJTx6DgyUJTKuZOZ1I+cF3GaLgVZcjddtCy1ZrJqizj
xAkiBPd9iaSI2cKD7Fl7SRDvmo3Ihvlz3fIOYHqc2Y9Pd4N4DEtMLd7tn7GvHEMy
rLDQODpUniYPFEuNwW71JpUkN4ft7eDD1e/A8A119W+avv1kPCoirzgSK3MtDQl+
Eer8azJzTVzEWRfaxFmBBgS2CwLQZ7OWnHkTQxUkXsSV/VDRXgieH7ShlpI5K2is
vYw+hokuPrbrReC8HJsrC3jvbfEaYN3mR/h19PLKRKj7gFngUWOFC7b7Fizj8/9v
92q+m801gQIDAQABo4ICHDCCAhgwDgYDVR0PAQH/BAQDAgGGMB8GA1UdIwQYMBaA
FEl0uwxeunr+AlTve6DGlcYJgHCWMB0GA1UdDgQWBBQITtWkPCoEm5MbtwQIjnS5
BnwNozASBgNVHRMBAf8ECDAGAQH/AgEAMAwGA1UdJAQFMAOAAQAwZgYDVR0gBF8w
XTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETALBglghkgB
ZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUDAgEDGzA3
BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9ET0RST09U
Q0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5odHRwOi8v
Y3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAGCCsGAQUF
BzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNsZGFwOi8v
Y3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIlMmNvdSUz
ZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJjYyUzZFVT
P2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUFAAOCAQEA
mffXrLElu68fAzW/Vnv1oWCm2pTuj93MMtE1DZ/lXqZ0En8BKlozIcDXBsq/3Rtm
VE8CVfym32gX0r/0XWuO+chz21tUOt294WnZ+pHbKloPx46INQgjq2Rn298fa/yO
X3Kfl4GHgeWlIX3YT/4xm6F5pCZUQfBFkK9fQsEelof5z8ekGkRTkREOOIBktNkT
1iOOiMepsSAkVnwH+8R79PmcerUORLcyVzpNg5HEdRiUls9f9m82K65zGfjg/GnO
hn//QiE++TjDXnqZKN6YLLCciBCyNB6qCArLTgHFZOtNpafzCD0LenU6lkr3/c8c
r3JMcULZ/iO5WrStVwX9JA==
-----END CERTIFICATE-----
```

### 12.11.81 tls/DoD-Root2-CA31.crt

```
-----BEGIN CERTIFICATE-----
MIIFTDCCBDSgAwIBAgICA50wDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTMwMTE2MTQ0OTMwWhcN
MTkwMTE2MTQ0OTMwWjBXMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTESMBAGA1UEAxMJRE9E
IENBLTMxMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxicQL5CWONnf
5l8/uon7ZoLrtqXt8FaQFkDnbKKweWZZ15hiMdEzIlPjHlykVmamTVb7w+JCEqv5
wEpLQO+RE4Y5MFHWbo4nt0GJKQHuWEZzBHFEXGlDPjLmZN+za5kscKLQPk3YWBJt
RfA9k1S+3+L7zxH//IoBN++nLrpADGo+HOQKMoBpvSI57Et2ybFakzwhhDjdcxOC
+VOMgQqpslNO2QuOwOiXuz1fE4y1uTvs9rudjiD2a7ydFDLcfrniY7BqwYC5FvyR
76yyCZ9SR1gTXmJ+mhKGW8UgH+GOZgB2U+znIokhTF+56b6gUpMOpsjezLeCrSJt
i9AwUzZVVwIDAQABo4ICHDCCAhgwHQYDVR0OBBYEFETjRqNB7mCxXqeTJfSgU+63
Sb67MB8GA1UdIwQYMBaAFEl0uwxeunr+AlTve6DGlcYJgHCWMBIGA1UdEwEB/wQI
MAYBAf8CAQAwDAYDVR0kBAUwA4ABADAOBgNVHQ8BAf8EBAMCAYYwZgYDVR0gBF8w
XTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETALBglghkgB
ZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUDAgEDGzA3
BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9ET0RST09U
```

```
Q0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5odHRwOi8v
Y3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAGCCsGAQUF
BzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNsZGFwOi8v
Y3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIlMmNvdSUz
ZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJjYyUzZFVT
P2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUFAAOCAQEA
R1FS3PSgc5pC5wvsI5GNJXW0RII0qvlGdVHD9g745+MvtCDD76FlN0Cdh8HmLmLw
J+jrxc81ldJAgIuSCbamG9USZDHbtdQ03wqKtlb1vHaSkxl8v2V9coHYZHs5NIp2
WMwdQ/cHzxyDA30+0BfbdK1pCRF87djWAo1mPatryjPbx3pmxd6nJ0gPZhLuaCTA
75HqBhkqUFgT4CL8DrEk++u0QgIPd4gVi+by9V03f0BVmxPWtnDKc3DjUyXBKB57
xCxJbpDbqstbAxvCh4f1q75RcXNtJmZ7mx0X403jwN4dJ7HtDTRGPt0uXvSCcNrR
kxt53dZK5875P3MfzormFg==
-----END CERTIFICATE-----
```

## 12.11.82 tls/DoD-Root2-CA32.crt

```
-----BEGIN CERTIFICATE-----
MIIFTDCCBDSgAwIBAgICA6EwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTMwMjA0MjA0NDA1WhcN
MTkwMjA0MjA0NDA1WjBXMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTESMBAGA1UEAxMJRE9E
IENBLTMyMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAs+KVHZM2LSWl
Dv146e/qk9E6ydhXvRnf0cei0ejZ/dK0FajdvT5k9Lb+nAPfS7Blt6sEGDIZbBMB
UtHmtchBEre+08tNQBCIyp62/TV3bSb2ZK0RhwypJXpYn7C9mPaTXxvv77KXrfgV
59zmoGp1DVHfVR1oQVJJLsecaFdWR4/e9lIugW9WvAaJEpSfI70/gceGAnUwXj0h
30ETu/15VgE8Shn0L0uQZGTX6AovUYbVCJuE+/npi0LKZdKQBxyCl4xEI1cGLHVp
KHCy7T5M1e0WdxX9upXPW5ZpAnfWgNmPhynj5wV2r8qNEmA0cseznThuTJYynpA1
rXWL0WJACQIDAQABo4ICHDCCAhgwHQYDVR0OBBYEFC/Kk1MDrG919Xb6vv606hCL
t+eQMB8GA1UdIwQYMBaAFEl0uwxeunr+AlTve6DGlcYJgHCWMBIGA1UdEwEB/wQI
MAYBAf8CAQAwDAYDVR0kBAUwA4ABADAOBgNVHQ8BAf8EBAMCAYYwZgYDVR0gBF8w
XTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETALBglghkgB
ZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUDAgEDGzA3
BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9ET0RST09U
Q0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5odHRwOi8v
Y3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAGCCsGAQUF
BzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNsZGFwOi8v
Y3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIlMmNvdSUz
ZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJjYyUzZFVT
P2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUFAAOCAQEA
MI3VVm09mQaLTbbSDg05xoTSm3dBGojS/8Pa4uZnYb3Zeu040V6rC1g0+droYnmv
0XLzSqfjTjkQzenSC0rUnpqnNTWTkwJZ4kwAHPP8ayFTSoxh52HL0EYL0T+cafXv
UIrwQLMrVloda2JZBb0PJxgFCkNbAu/dUl5bwKkcVu0VbJdPAYNWcl3XfVHjWlQu
uJj9ck4lj4sW0bDhM+0SfBBVMyRmrw8zBlNIA4eftGR0tdI9InK30Y43ERM5357n
0AwLilkRMmX/9rlGvT82nqeUAFfwwBnhLNxM9y9MkB1D764I430e0r+Z7CK5B1iu
2TVSS1G7gTaPn24hCqa0hw==
-----END CERTIFICATE-----
```

## 12.11.83 tls/DoD-Root2-Root.crt

```
-----BEGIN CERTIFICATE-----
MIIDcDCCAligAwIBAgIBBTANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0wNDEyMTMxNTAwMTBaFw0y
OTEyMDUxNTAwMTBaMFsxCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRYwFAYDVQQDEw1Eb0Qg
```

```
Um9vdCBDQSAyMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAwCzB9oO7
rP8/PNZxvrh0IgfscEEV/KtA4weqwcPYn/7aTDq/P8jYKHtLNgHArEUlw9IOCo+F
GGQQPRoTcCpvjtfcjZOzQQ84Ic2tq8I9KgXTVxE3Dc2MUfmT48xGSSGOFLTNyxQ+
OM1yMe6rEvJl6jQuVl3/7mN1y226kTT8nvP0LRy+UMRC31mI/2qz+qhsPctWcXEF
lrufgOWARVlnQbDrw61gpIB1BhecDvRD4JkOG/t/9bPMsoGCsf0ywbi+QaRktWA6
WlEwjM7eQSwZR1xJEGS5dKmHQa99brrBuKG/ZTE6BGf5tbuOkooAY7ix5ow4X4P/
UNU7ol1rshDMYwIDAQABoz8wPTAdBgNVHQ4EFgQUSXS7DF66ev4CVO97oMaVxgmA
cJYwCwYDVR0PBAQDAgGGMA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcNAQEFBQAD
ggEBAJiRjT+JyLv1wGlzKTs1rLqzCHY9cAmS6YREIQF9FHYb7lFsHY0VNy17MWn0
mkS4r0bMNPojywMnGdKDIXUr5+AbmSbchECV6KjSzPZYXGbvP0qXEIIdugqi3VsG
K52nZE7rLgE1pLQ/E61V5NVzqGmbEfGY8jEeb0DU+HifjpGgb3AEkGaqBivO4XqS
tX3h4NGW56E6LcyxnR8FRO2HmdNNGnA5wQQM5X7Z8a/XIA7xInolpHOZzD+kByeW
qKKV7YK5Ft0eC4fCwfKI9WLfaN/HvGlR7bFc3FRUKQ8JOZqsA8HbDE2ubwp6Fknx
v5HSOJTT9pUst2zJQraNypCNhdk=
-----END CERTIFICATE-----
```

## 12.11.84 tls/DoD-email-Root2-CA21.crt

```
-----BEGIN CERTIFICATE-----
MIIFjzCCBHegAwIBAgIBSjANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0wOTAxMjYxNjQxMTNaFw0x
NTAxMjUxNjQxMTNaMF0xCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRgwFgYDVQQDEw9ET0Qg
RU1BSUwgQ0EtMjEwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCa7Qjc
I6BER5v1w57JO9huz+v5xoNiegyD+Y84foLAjZzRQizLiA5iUSKpgBdYQXoMRps+
JahqKKm7Ev38hSvvs1sxT8oVxdO3mEVmBXL2CZpy6Sb/vZAmNQolvbusv9DWOId5
YSx70Q7TKvUSP0DkmHNowkmsj9SMChevPkpEqT85DWm7Fg2Gjg7pvlN2eYMfXW6K
53HWRcGkzzJySODnEPmxC7XzdPBkGhNAlNITbbIJIVfh3akHV6a9wKSEV765HVFJ
H3xbxubSI/O2VVeIyHlF22PPS2o7Mey1PV1nvLJXpS3V7fxM2DuH0UdzGHRvcFNC
hm4vwHgWwbm2MoclAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0j
BBgwFoAUSXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFFFnhBz/q6exnR2E
ZISZXdAL171bMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYD
VR0gBIGXMIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsK
MAsGCWCGSAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFl
AwIBAwYwDAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0w
DAYKYIZIAWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2Eu
bWlsL2dldGNybD9Eb0QlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB
7jA/BggrBgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9E
b0QlMjBSb290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNh
Lm1pbDCBiAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNk
RG9EJTIwUm9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNk
VS5TLiUyMEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkw
DQYJKoZIhvcNAQEFBQADggEBAHLEcRd0U0HK7vqTxTQS+kQV2uBjSdayG16jnh8h
b3iiDwrC2tS6ZscibJNNCUspZxSNmQ4FBvet3EfcDpkICi7yCZfo9SGFzvINNP+a
Q1+TdkqjDNgqIYsNYE52Hq0K7xO7NC4MFVY7tjF7Np85iIvLSLPZBE+fEVjl2a2Z
wBIoI5hw+p1IA2u8oNhOPbaRqaKIaIbCsUgTUtjAgJD4b0ghISfjej7RspxknhiC
aDBXhAexdVqZOJIpa/0bMQa3l/rl6zqCZNV0ebd2B7c0bqZJykLGIjuDsKQ42zSm
sJUkH8vxH7bA/3um3A/4/SW2sjLWdpkkS3fq/S3EYbmx/y4=
-----END CERTIFICATE-----
```

## 12.11.85 tls/DoD-email-Root2-CA22.crt

```
-----BEGIN CERTIFICATE-----
MIIFjzCCBHegAwIBAgIBRjANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
```

```
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0wOTAxMjYyMDI1MDdaFw0x
NTAxMjUyMDI1MDdaMF0xCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRgwFgYDVQQDEw9ETOQg
RU1BSUwgQ0EtMjIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCmAAYl
I4D59WDEBgYoBUcuG2cfvrRF5Rw0xvTFutJMaJ1TJNeXv9joB894zproZMUQedNv
xx0zm4jDAPHwKfhT/eClMShoyS6MOAeSRbQ/CALL9+4BgS1fSxWx3YDyucD/qe2g
9Sebeex7JSslESmr/V8RPGKTl0J5SMCdBtG3IyWZV94GVcoeh5MU9xJDMdEmDm3S
RUw44tKa5xKvyUxd48h/H8fKCTnxCU/GoudhgXmZC9KMC2V6uTwYFc4Quy/AZBoy
CNGwoBKMEMuzbKRwQKy9VgtUpdTxRjPc7PZRUq8nJy6dVaQd911a+GRoQYlYvS93
nSjeDXhfiGf8HLyxAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0j
BBgwFoAUSXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFIUvDQqvNQhQCY2b
HHCsqP6Jd5RaMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYD
VR0gBIGXMIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsK
MAsGCWCGSAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFl
AwIBAwYwDAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0w
DAYKYIZIAWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2Eu
bWlsL2dldGNybD9EbOQlMjBSb29OJTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB
7jA/BggrBgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9E
bOQlMjBSb29OJTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNh
Lm1pbDCBiAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNk
RG9EJTIwUm9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNk
VS5TLiUyMEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkw
DQYJKoZIhvcNAQEFBQADggEBADhU9UuJMePz9RcXpXSfyz+JU+9B6ldRxeTizrr6
QB+YTG5Day3PwBlu9BdH3ZaQRqZzL+3xJ3iHT7ftATRueWi/hclcZWy5e5gqip5d
YUAmvOSHNZ8D6s7JeQwGfmjenVXD0QoIf9jm5zqDVpfj4cOybztEdrhzbOrwxyBM
jzFVgIZdHuY5RJmONKFp+W1fcg4FR2maCOxl2SmAn+CvfgEDuAvpE/dYIdYw/qDu
cnuBeYENlWCEPcpItgx7iXNfmF17Hg9pqgQmfGqRcP3zYthQT3l2umlW+r5uu4xX
b7HH/i7fhWXCshcUGRwWE/1+HW+yJ9YTHAxZkHC9VryuAoY=
-----END CERTIFICATE-----
```

## 12.11.86 tls/DoD-email-Root2-CA23.crt

```
-----BEGIN CERTIFICATE-----
MIIFjzCCBHegAwIBAgIBSTANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0wOTAxMjYxNjQzMjVaFw0x
NTAxMjUxNjQzMjVaMF0xCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRgwFgYDVQQDEw9ETOQg
RU1BSUwgQ0EtMjMwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC99ClN
V5NEAweNBh+u+jw0VjRA57EYf2wlbheFBQUj6fbFUCVPgeQjMEJxaer3uJ73b6ze
Xar81uCNvGvufZmVIjuzWMaxUhyyqL8xQCIG/oOoOqlQVWoeB3D4pkjJbf2u7L6A
bD3PkNQHok6RFAO/V1kS9XTeQ5ZaWrnPuUfof9COsPjY6UsOXsxLF44C8BK/8gRs
HRO/qxzeDQnsy5tW7dmQ55alfyZlYcHEm2gkpc3SeSNvwzzBhR5I+T5QcWKgQbpy
RKVD46Vybs3Oq9rLhNIavx9uchE/LZkfbbD7BTDO5uwjKmVHH3icDZ9MJHVsdtLV
OxdEFrKKKEjXuQ2vAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0j
BBgwFoAUSXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFFuWLTGl4faaalxe
gVE2YR6WJBnRMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYD
VR0gBIGXMIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsK
MAsGCWCGSAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFl
AwIBAwYwDAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0w
DAYKYIZIAWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2Eu
bWlsL2dldGNybD9EbOQlMjBSb29OJTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB
7jA/BggrBgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9E
bOQlMjBSb29OJTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNh
Lm1pbDCBiAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNk
RG9EJTIwUm9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNk
VS5TLiUyMEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkw
```

```
DQYJKoZIhvcNAQEFBQADggEBAD729uQP1FfNvu3pxEVS2kVbGKY2OTAKtn7r9on8
p2Iusz+DfeESGobTY5T0dPTOcZfq+8RqOl3imaesw+I2+Oh49NjEUO6KgVX6ioNl
DMXyTczpH497Rt0DCpzq4qxjdwfLMlTbCFWyWAB9XKa5FjxfZW0vBc5aP5rbScuS
o6HZb1HU9cAIwaM5W9BBY4HElGVkYylMXfBfcYdqnZaS5ceC/S101wJsyuLboLPb
cdUOhj4+F4m9bkIXG7T2OfkaveYuLJsONzsQX0T+e7WUWNZxJeU50SO6NADBQ224
9A6Xq7Iw9oinjo6KEAOEdNyuTfnlmXQaqaIKbQsHFZTFP2M=
-----END CERTIFICATE-----
```

## 12.11.87 tls/DoD-email-Root2-CA24.crt

```
-----BEGIN CERTIFICATE-----
MIIFjzCCBHegAwIBAgIBRTANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0wOTAxMjYyMDI2MTVaFw0x
NTAxMjUyMDI2MTVaMF0xCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRgwFgYDVQQDEw9ET0Qg
RU1BSUwgQ0EtMjQwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQClLlh6
od7mmlv2AvHV1Nw1I5p7bihkdBpwPJYdzMKfdAQ8DDmSIQgNEk6g1zeo0snGJ50o
+lXXshcEGc4yvPB5nvVoqy7MzzcEsvgKZZpJIBQlwbwSaqBCbRsItIehQiKrE5na
AgE5H14IV2tg3hN+aGp+QfWJgDh6/Zey0uKWSzaAYrbsJbvQD6ejzVGo99J5VZA0
JqPkXM27aCZ0CTeh5q/N5D6ZR/9/wke8ZYS6MimjDvDColt66rJKfQvGw26svRB/
T6l2Oj0CASwqMLT3yKDSmDp8CNBaiQ+1ioL6DTAeftbRx7ZDJ7EoqQzjswd432Jk
mkWMTs2vDq6cWDbfAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0j
BBgwFoAUSXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFFSqcyrHs3fqzSJA
eUh7EfunmSKCMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYD
VR0gBIGXMIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsK
MAsGCWCGSAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFl
AwIBAwYwDAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0w
DAYKYIZIAWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2Eu
bWlsL2dldGNybD9Eb0QlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB
7jA/BggrBgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9E
b0QlMjBSb290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNh
Lm1pbDCBiAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNk
RG9EJTIwUm9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNk
VS5TLiUyMEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkw
DQYJKoZIhvcNAQEFBQADggEBAHIW3DlzY02T6Tccz7LtnNhN9wwySomes8q68wSs
cWYxpiq9un1U2C8JY0qICOhsE6HsXntWFzAtyNLt141HRGnPEW/L2OdSdbVRyKod
afAZHzDwB8c2vc4M3jt2/QrOy7YTutaFi/FcEpHKr+h/EqisLYvWdlCU7Db6ow/f
xjLqx3NG/IQami/E6CccSMJGNvYX7O1nMg+4ouC30l6QBhOUIWFDbH3zO2tl7ePb
qP/Fm7KS5+tf7u+/8zmMs/UX0obVw2xKOmw/nq/oWx02W6YmFUYLRmvH1ICq564c
uCtO+iFyn1+fga+07lvJlymJf0nceOJO4HSf0oZ4ZqLHmKg=
-----END CERTIFICATE-----
```

## 12.11.88 tls/DoD-email-Root2-CA25.crt

```
-----BEGIN CERTIFICATE-----
MIIFjzCCBHegAwIBAgIBTzANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0xMDAxMTQxNzM2MzJaFw0x
NjAxMTQxNzM2MzJaMF0xCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRgwFgYDVQQDEw9ET0Qg
RU1BSUwgQ0EtMjUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCiR3Pf
QKAVpqftoOnIf/vSjk8J5cWNXGQaQ37hc7mvEEV9My3qYCZzYiGLL66cF8zV5jih
Le4Cs/C53qaLiNG0Yz3IiIgcf15C6x2T86t46ZQdAz1NPhkXfO8JIoL+w+Sfns3Z
vYKEOQxSt327QX/1jaQq9tBcYjHI4+q3t3jWm05iXrUS28p0XbhqEUNJFYO5aWOP
TWLC8gR3WQSrBc6sFF6ZfR9Oi9TJope6ztCc4502/oyB/Gg5TZ4j4oOz06vg2d+Z
```

```
ZArINPKs4vVQnl5tOQ9FslLrTpJvH2nIoTcYbWHIhqrPxLfMNOn6fBtmcoFKRoxB
SqsgCnb3zhLUOAZVAgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0j
BBgwFoAUSXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFCbb67FFLtgSkE31
EkH1w/AezODOMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYD
VR0gBIGXMIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsK
MAsGCWCGSAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFl
AwIBAwYwDAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0w
DAYKYIZIAWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2Eu
bWlsL2dldGNybD9Eb0QlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB
7jA/BggrBgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9E
b0QlMjBSb290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNh
Lm1pbDCBiAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNk
RG9EJTIwUm9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNk
VS5TLiUyMEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkw
DQYJKoZIhvcNAQEFBQADggEBAGmQDW4i6bsFtubPOCSokDN5muLZYBEi1ewoR5Ag
N+KEhLJo9n3+i0sRuR2/28zc6XJ+RsmMSKk2hmCQmdr4WNty0KmObmDIvGDrBNAO
+HGF51vpwfvskpWqA2n9yDQFZdUCT0+ZgxrIRlw7/vhx2Hw7PVzRzYJMQ431Gqao
L0sCdNco1pFG0E1jja30lIiYIyOLtu2OQE6G9Nnp0TZK1FPAS5bwsbhuQJxqMnxl
bbZg7YFKUFdTY2bod8d53HcjCz1jSm276E9DJM9tmFwR6C+IpTrlTsTY0P6cmOQy
rY4nFFWr2si3dkL7WRiSuAormmbMMPvEY2omt7eRHRiiPpw=
-----END CERTIFICATE-----
```

## 12.11.89 tls/DoD-email-Root2-CA26.crt

```
-----BEGIN CERTIFICATE-----
MIIFjzCCBHegAwIBAgIBUTANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsT
A1BLSTEWMBQGA1UEAxMNRG9EIFJvb3QgQ0EgMjAeFw0xMDAxMTQxNzM5MjdaFw0x
NjAxMTQxNzM5MjdaMF0xCzAJBgNVBAYTAlVTMRgwFgYDVQQKEw9VLlMuIEdvdmVy
bm1lbnQxDDAKBgNVBAsTA0RvRDEMMAoGA1UECxMDUEtJMRgwFgYDVQQDEw9ET0Qg
RU1BSUwgQ0EtMjYwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCsXxBb
PSoH/e7PTsawTj1aoABgUHnCAkVmTzg0gVwcvydgannMppcL1onm2NuVfFc6B+5G
5WQqExePHTD8Lfo2fzdhJOUUov0iVxxhrk0uA1BmVUbdaif4qXmrXCrlJV1cG/tx
D7W4FY9flHsDz+6rkggK5L2joV2D1z3Hn9REEDqiX1/khpRvA6A184PY4bgZn3q6
dc8ABdDbI6RqJddpcEXGXXiLB19FrJ3Wo0tdGM+PTAoRodkR2/mcpdWPnOoPR70l
gpT5YJJKFPi6m6ls38oVEaGLOb76GU28uxRv3WB9spyQB3yAR7mFjLg+o3W5rl53
kXPBdYlVuk2G5K27AgMBAAGjggJaMIICVjAOBgNVHQ8BAf8EBAMCAYYwHwYDVR0j
BBgwFoAUSXS7DF66ev4CVO97oMaVxgmAcJYwHQYDVR0OBBYEFDLfyG3z/z4p/ekM
lylQ8KIQLG4vMAwGA1UdJAQFMAOAAQAwEgYDVR0TAQH/BAgwBgEB/wIBADCBnwYD
VR0gBIGXMIGUMAsGCWCGSAFlAgELBTALBglghkgBZQIBCwkwCwYJYIZIAWUCAQsK
MAsGCWCGSAFlAgELEjALBglghkgBZQIBCxMwCwYJYIZIAWUCAQsUMAwGCmCGSAFl
AwIBAwYwDAYKYIZIAWUDAgEDBzAMBgpghkgBZQMCAQMIMAwGCmCGSAFlAwIBAw0w
DAYKYIZIAWUDAgEDETA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3JsLmRpc2Eu
bWlsL2dldGNybD9Eb0QlMjBSb290JTIwQ0ElMjAyMIH+BggrBgEFBQcBAQSB8TCB
7jA/BggrBgEFBQcwAoYzaHR0cDovL2NybC5kaXNhLm1pbC9nZXRJc3N1ZWRUbz9E
b0QlMjBSb290JTIwQ0ElMjAyMCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNh
Lm1pbDCBiAYIKwYBBQUHMAKGfGxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNk
RG9EJTIwUm9vdCUyMENBJTIwMiUyY291JTNkUEtJJTJjb3UlM2REb0QlMmNvJTNk
VS5TLiUyMEdvdmVybm1lbnQlMmNjJTNkVVM/Y0FDZXJ0aWZpY2F0ZTtiaW5hcnkw
DQYJKoZIhvcNAQEFBQADggEBAGrAUCmE+NxnE7GW7rpc1OWS+c1kZTOFKAugGqtT
HYxAF5G6Ztpra7ysjmEBw2c1EVlShXBdoYbnesEcw9hey3e7zFzcGt0EX/qI7bNu
tbREyzo1naBOHMBFtfbUzQZ50ho57CUmcZzZuG+TbNY7NDtnmapfpbhtTMcJ6snA
dJnZWYspiZArgZXZh/1V+Fh1UqZ/ImhthdZ9rooNLzS/1yhsxlutvP8bOsZkhaSc
fYSVn6gDZeR/TcwMdXpKBURYgIs5NE8zPytE8dZ07+98mtjcCxg98uWdUDsjKeX0
z2DqYE8cYMEaspxaAgSwfMHJFWrbKq8LCLs+cXqmPOUdRCI=
-----END CERTIFICATE-----
```

### 12.11.90 tls/DoD-email-Root2-CA27.crt

```
-----BEGIN CERTIFICATE-----
MIIFUjCCBDqgAwIBAgICAbYwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTEwOTA4MTYwMDE4WhcN
MTcwOTA4MTYwMDE4WjBdMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTEYMBYGA1UEAxMPRE9E
IEVNQUlMIENBLTI3MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA053C
B7D1fszurrirqjPqp5JuE1ZAaOUfxiG8wIGXYSxw0VoVF/6co+9IaYm3W1L5rOA6
nKZDViAogmzN9Zb+gJ3ZjfHR7oGavOzwhTUWQbkmiQwmekBu0AmAUcAC2O6Eb8ws
giKqNYVepF6FBNEJmaS4fVKxIXpN2CGnvERPyhWijDEuidY5L0BWN3jrLl0u0RhH
Fu2soITUC4KYvQMYcLAZXYxr3jUkYlrI+w+6euzIQElyVp4aTVTATuUQNE9h0dLt
Td/RWbDrAkIvDBtSDBWg8u66Nlf7zKwR8ZotTIspGPHwcJJo1kEmzFt8dXbYBWBS
Own8rcBAHKRG1jAtewIDAQABo4ICHDCCAhgwDgYDVR0PAQH/BAQDAgGGMB8GA1Ud
IwQYMBaAFEl0uwxeunr+AlTve6DGlcYJgHCWMB0GA1UdDgQWBBS/yO1EDrsz5sfK
QSylMbnJYGGJLjASBgNVHRMBAf8ECDAGAQH/AgEAMAwGA1UdJAQFMAOAAQAwZgYD
VR0gBF8wXTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETAL
BglghkgBZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUD
AgEDGzA3BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9E
T0RST09UQ0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5o
dHRwOi8vY3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAG
CCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNs
ZGFwOi8vY3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIl
MmNvdSUzZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJj
YyUzZFVTP2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUF
AAOCAQEAEX3yTl2o1kYa59nUZrFRxoHg5n89ca5Gp3ALg7S9wEAzUJuHQ5SHBWl0
vmdt3SrsCjEEv3iVb9ix7EMnCs8AgAEWPH2XN4WYW6aAcwyzd/7JcDNSi3p1t7ku
/rwtJUaW+kVteCjN25uZTAeeLGINitt/eFUFRxIb25kCN/lnHwQx7yiBd3ZaLpSL
dXg9icx40EsFmKLAcBaHcP+LfAnS4S0y7QPtYSuN2s7N0jzj5o/2ceO6L1yEgm6I
pl06q8Ft/mf56avlOETvQxvlKrEw+/T7b32kIABUYCI+XTNku5TqWnaVn8iPBms6
YOjCRiZTq7rmKMv5W469Q63xx2gyvg==
-----END CERTIFICATE-----
```

### 12.11.91 tls/DoD-email-Root2-CA28.crt

```
-----BEGIN CERTIFICATE-----
MIIFUjCCBDqgAwIBAgICAbcwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTEwOTA4MTYwMTE5WhcN
MTcwOTA4MTYwMTE5WjBdMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTEYMBYGA1UEAxMPRE9E
IEVNQUlMIENBLTI4MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAoL6S
zgqY0wjhxffuXYJK28/KZvS9cG6TG8qbOQFQGMDTVnrLWGfdBaTUKzFBYnoK/cL7
HoUsivnrqbOfs8papHhWVRdBl4n1zccwxt7IpRbq2OCGKBEMeA2dN+4Y+RYtX66E
bjSLukY79D16oz/jrpuph3Z9w7fgsi2COkF/uWnhUCKxv0xRakr5Aw4UtzKpX40b
71FXvIcpW2UmP/nzoZI4qNkxxxgRt+uKGNOQpc0JsrUs7wlpnsil12IiD9qF4Bqj
NLQYjKl1ScbHboSNqaSQX61brhOXXalfYA/cxJGSNgN7/WlZydb659zg/lo9XD/0
PwAbWF/TCUfvUHLIMQIDAQABo4ICHDCCAhgwDgYDVR0PAQH/BAQDAgGGMB8GA1Ud
IwQYMBaAFEl0uwxeunr+AlTve6DGlcYJgHCWMB0GA1UdDgQWBBRZiDBI5m3+YSem
xNWFjVtznu/BzTASBgNVHRMBAf8ECDAGAQH/AgEAMAwGA1UdJAQFMAOAAQAwZgYD
VR0gBF8wXTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETAL
BglghkgBZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUD
AgEDGzA3BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9E
T0RST09UQ0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5o
dHRwOi8vY3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAG
CCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNs
```

```
ZGFwOi8vY3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIl
MmNvdSUzZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJj
YyUzZFVTP2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUF
AAOCAQEAHWveFxw66X0qZH10AyB2ZE2foB7ZW0VKdzHMZSta6bZfXu42iBAU27d8
AEyxbTkJGXiMMml3qmSefHHXSbEsoN8nMVIqmYL01lNGSszA876YH2ATi+KKB2R+
hUyxCbHpWIrNmX4SwpNL1/WkFD7EewgwQ8gmfhf2U0m/au62A5LDAATJSQeJ8EGt
19/M1/MmhGJQshQ2ygsGOimA+YOrpUSG4oEs7SADS0SvD5hBVMXAGIchy9WDTGaR
exYTV5GXdJK9AZUoe07i2tZWIDbSy0Z9dMqK4/nWwEInSQPOPwUPqtilvzMuFg+u
HbH/yUvYcWuTxaH/ajtVXhk3XlsjyQ==
-----END CERTIFICATE-----
```

## 12.11.92 tls/DoD-email-Root2-CA29.crt

```
-----BEGIN CERTIFICATE-----
MIIFUjCCBDqgAwIBAgICAbgwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTEwOTA4MTYwMjE0WhcN
MTcwOTA4MTYwMjE0WjBdMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTEYMBYGA1UEAxMPRE9E
IEVNQUlMIENBLTI5MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAkmIv
QcIgABYGVWSfvaeIFW6Cm0jBhXe9AqsM2fErYIEBuj51cI4Spqc4hJCz6UCAEtxq
ylHNrS2GEMxEvA7FWDgZshQyJUFUWFxDDshscw/DDBgYFgSaUj2BonHOPDIAn3FV
uvjONnceIbcolOc9Pqb2wHoxYJEol3ciUPLGk26yG8VBxvmhN/sQv9pWpvtSTV+/
78SWdyjlMv/o4RjMQ1IYrI13mnJM6JODXrCi7+Td0ufmp6ZSreGYCJZKQ8xzPUui
jYnv3IJMuEqAJGUrHpGC9QT2ch9XGEAX8DlRto/ziTtn91hOSrza+Q7BwAy98whx
+IMPyS6AlfSFDs6uqQIDAQABo4ICHDCCAhgwDgYDVR0PAQH/BAQDAgGGMB8GA1Ud
IwQYMBaAFEl0uwxeunr+AlTve6DGlcYJgHCWMB0GA1UdDgQWBBS4Q4NkIXrucIHe
pd4MYCiHeK5eeDASBgNVHRMBAf8ECDAGAQH/AgEAMAwGA1UdJAQFMAOAAQAwZgYD
VR0gBF8wXTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETAL
BglghkgBZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUD
AgEDGzA3BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9E
T0RST09UQ0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5o
dHRwOi8vY3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAG
CCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNs
ZGFwOi8vY3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIl
MmNvdSUzZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJj
YyUzZFVTP2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUF
AAOCAQEALGsseTXb8B4ch3ur4ehpajeL23pPVWBplS9TncbKQ7bUN5HWA11+WrG4
HfeegdOuUFQwpG9LLrsUGxeqXBDTlHoxOZakVHn16VYuVcMbFuqqAsjPUfcygSLG
NDqpzZqqSJPH6fseMn5xxHbwRVQSHVXqvVwyhzquk5pumSJfqFE17rJTYF/2T0W4
FoQdZVXNFcoQAR+p0pynV5Gj1+ewhj0t9Ik62Ml3cFDGbO/y65j4EKo92shcKa3O
uHNJTKGSu+btzbqCGmMhGWX0Bhm/g6pz5dMbsZj/Rd/7Scxz6OLnB5YAMel/2SQI
58pEekgGw0LYP/l5h6U3khaphCCSYw==
-----END CERTIFICATE-----
```

## 12.11.93 tls/DoD-email-Root2-CA30.crt

```
-----BEGIN CERTIFICATE-----
MIIFUjCCBDqgAwIBAgICAbkwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTEwOTA4MTYwMzA4WhcN
MTcwOTA4MTYwMzA4WjBdMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTEYMBYGA1UEAxMPRE9E
IEVNQUlMIENBLTMwMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA5iki
1BQm0ZgaUl7FhINzfsFgs7PQlL79HJRVv/aELJvJwHRz78zCmfKZyW3KFNN0/74Q
8vctv8u7BqPumFBBZQHhVyy2y+TKHKx+UjQ0sY4HJj4yNa+jYQrF5Qi2EnmMVMF6
```

```
6fFQH12DOmcwsynbHTpMOSFQ2BgsjQZ17mNyeGitYpx1pJQG0zJrEq8GBym+E6DA
p/AlT7f+H7dX4BgSjSFqFblaVPt3ZdhMP/W6PMA34QZ+wr6eI4wo0ZrXxmc413PJ
vQcdhW/VlQqa3No6TijwpesJ3+XbC81Hr4rNu2+UQONZnFCfyQ6pcQK53OlpgDqJ
OOUFIhgFhLUS8DzAgQIDAQABo4ICHDCCAhgwDgYDVR0PAQH/BAQDAgGGMB8GA1Ud
IwQYMBaAFEl0uwxeunr+AlTve6DGlcYJgHCWMB0GA1UdDgQWBBQ1YWYoCbxWJVuL
zL+BXmEsMDnTITASBgNVHRMBAf8ECDAGAQH/AgEAMAwGA1UdJAQFMAOAAQAwZgYD
VR0gBF8wXTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETAL
BglghkgBZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUD
AgEDGzA3BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9E
T0RST09UQ0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5o
dHRwOi8vY3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAG
CCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNs
ZGFwOi8vY3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIl
MmNvdSUzZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJj
YyUzZFVTP2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUF
AAOCAQEACohWHKVXJlpiy3XQ3YbFUuIv87wRZD+MLz4R/JhgQPKADSiCmmj+4EhL
J9M6CnuV9gMMgRSRQjpgbOIrUy3s3xGu9VQX8AH5lwenm6sL26yXiQnG7/kHNBYA
qH4RU558L6E4opl5OTRBbn24WDBWiJ7kqmRF2aBEYjq35THTkYDxGxCyZ3DVW6tZ
tFpIFkLEAkzabGjKUB0xvjeZx89TzEIpVsOdF8oD5xBa8Tk8HMz7G5cKJvMx3+Cr
XCSdnt44fQJRZ0b5k3CF7QpVwvTBaFqfCMkde5t23FTvOYwY5QxE7vcGsh/1y+YO
vdSh/9T5kQciUnm3wP3ssviF9ET7XA==
-----END CERTIFICATE-----
```

## 12.11.94 tls/DoD-email-Root2-CA31.crt

```
-----BEGIN CERTIFICATE-----
MIIFUjCCBDqgAwIBAgICA58wDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTMwMTE2MTQ1MjQzWhcN
MTkwMTE2MTQ1MjQzWjBdMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTEYMBYGA1UEAxMPRE9E
IEVNQUlMIENBLTMxMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA6K4C
LEBMOlLoi3OStHfnOEvA8KpKGFzH9zXDSvDwlnell74n78REIYDqFjS3MNFEOH8q
zgTGkWWpblB8yE7+vcC1SxbkOFIV27O391M98rEH25FmXcG38ndmxFGaY5QRSwId
DUt8swBHB3kY+nizkx/Udm2ZBMUeNkb8BjQL42hvHnyfLM9huEv/tN8Gn6BflF7r
Nf8JXTVAB/Kd7ZYJ2Xbq/m4x/sv0ResweEhobKEpPoZ9k0FK6ucMTOWRUCqlQ2a8
IsD8Gyzk8y9iHgTUIb+sHyZ3NdAdvOK7RsLy6+QUrviza7P6cTiwcSnt0Ysb1wIb
3srsfu6h3Eil8T6UqQIDAQABo4ICHDCCAhgwHQYDVR0OBBYEFIbxW2hv3TDzlIJo
1Ez3RB24ymiBMB8GA1UdIwQYMBaAFEl0uwxeunr+AlTve6DGlcYJgHCWMBIGA1Ud
EwEB/wQIMAYBAf8CAQAwDAYDVR0kBAUwA4ABADAOBgNVHQ8BAf8EBAMCAYYwZgYD
VR0gBF8wXTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETAL
BglghkgBZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUD
AgEDGzA3BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9E
T0RST09UQ0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5o
dHRwOi8vY3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAG
CCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNs
ZGFwOi8vY3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIl
MmNvdSUzZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJj
YyUzZFVTP2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUF
AAOCAQEAWTKtqsP435xknHEJNMG9vGMAHi3b7anICOO5GOSvyq4Uwd27+XODg1eO
lMmgqgMHzmecteUXWT8ouBc22rqNw5YRAWpQ1gbaaKRK0guFfM2I3/9ed+b1pEiR
0E6iZ2r4aO+qF0Xv2JYK3c/wPoe2v4g/01S+PhLOofkLbzLRVL+EWzWg2wdktavp
eR7i8qp0cueREvfHu27u5XSQECSLt+fNnIWQR+Tib38gvSy7g5YjTahM2H4uXhUp
uCV9VzULLRVUjKnc4OU3nahPIJWDK8USNj2oc+FOiEmlubv6CUooWjO55JJ5W3v4
pU/zyTTNmYywumB+n4Q+5jz6flrr5g==
-----END CERTIFICATE-----
```

### 12.11.95 tls/DoD-email-Root2-CA32.crt

```
-----BEGIN CERTIFICATE-----
MIIFUjCCBDqgAwIBAgICA6IwDQYJKoZIhvcNAQEFBQAwWzELMAkGA1UEBhMCVVMx
GDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMDRG9EMQwwCgYDVQQL
EwNQS0kxFjAUBgNVBAMTDURvRCBSb290IENBIDIwHhcNMTMwMjA0MjA0ODEyWhcN
MTkwMjA0MjA0ODEyWjBdMQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zl
cm5tZW50MQwwCgYDVQQLEwNEb0QxDDAKBgNVBAsTA1BLSTEYMBYGA1UEAxMPRE9E
IEVNQUlMIENBLTMyMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAo/qq
hsqKGhsDTnFtQbbZZZpu/zYqPwLTfJVliFqk969jt1LHGvu7lXMHQmGLSqZ76VYH
NhuqNwIgHKTO+7bQaav8OEzI2OZW96JefucxtO7B/81kv3mCQSt30vh9q0yP98Ye
PPiOLz0Ug9qSmAnYOMZaWTaLh6KJ3b5KXsvNtkd+QaYJVGxBlnRbBsPUwS5GfV42
342iRnGsSrrEsffJFwov3aPshCHPqAXqueMub59+fbsdFnVPkh0D5hE4mDZ6odQA
PKOQWK8VxzZL4zubTbW0kL6tq9PAhLP83BWICYwRUFAv5HDstwquSlPiNsQFboB1
Eo03RvJLDDgcSR+sgwIDAQABo4ICHDCCAhgwHQYDVR0OBBYEFAqwqjhWR3sWfb6r
k5a8VN2F++0sMB8GA1UdIwQYMBaAFEl0uwxeunr+AlTve6DGlcYJgHCWMBIGA1Ud
EwEB/wQIMAYBAf8CAQAwDAYDVR0kBAUwA4ABADAOBgNVHQ8BAf8EBAMCAYYwZgYD
VR0gBF8wXTALBglghkgBZQIBCwUwCwYJYIZIAWUCAQsJMAsGCWCGSAFlAgELETAL
BglghkgBZQIBCxIwCwYJYIZIAWUCAQsTMAwGCmCGSAFlAwIBAxowDAYKYIZIAWUD
AgEDGzA3BgNVHR8EMDAuMCygKqAohiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9E
T0RST09UQ0EyLmNybDCCAQEGCCsGAQUFBwEBBIH0MIHxMDoGCCsGAQUFBzAChi5o
dHRwOi8vY3JsLmRpc2EubWlsL2lzc3VlZHRvL0RPRFJPT1RDQTJfSVQucDdjMCAG
CCsGAQUFBzABhhRodHRwOi8vb2NzcC5kaXNhLm1pbDCBkAYIKwYBBQUHMAKGgYNs
ZGFwOi8vY3JsLmdkcy5kaXNhLm1pbC9jbiUzZERvRCUyMFJvb3QlMjBDQSUyMDIl
MmNvdSUzZFBLSSUyY291JTNkRG9EJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJj
YyUzZFVTP2Nyb3NzQ2VydGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUF
AAOCAQEAD72PR/+5yb1D5c6+tfM5y0UWWaPftlIkPAlVS9m/lXq9dtngMIfNSqmj
LZ7ZKATGlq4BFIDQJVbxWANV79KoIlKrge8A/q/HSdKMIC6kcYH3Jss0pW3VQXd7
LTO7m7N8nD89/8LuefKJChCMkHRdNGdwvgL+gEYZB859L5aoxBPQ758psTSpuYyl
iTSzjD5H+GaMkdHuq8HqcYXJX7Cp7tsA1DAqQs5XYxAiMKichkESXb5QfBP66yhz
X3IziV9/DWikPf0WJugKk/57H4aBgCe+Z3GGG33Hb7epcQHGY7NzfQFrMyLteYmK
DuZyAnM3P8sxge2k+wtqO1KEukz3jg==
-----END CERTIFICATE-----
```

### 12.11.96 tls/ECA-IdenTrust3.crt

```
-----BEGIN CERTIFICATE-----
MIIFczCCBFugAwIBAgIBDTANBgkqhkiG9w0BAQUFADBNMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNFQ0ExFjAUBgNVBAMT
DUVDQSBSb290IENBIDIwHhcNMTEwMzMwMTMzOTIzWhcNMTcwMzI4MTMzOTIzWjBz
MQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQL
EwNFQ0ExIjAgBgNVBAsTGUNlcnRpZmljYXRpb24gQXV0aG9yaXRpZXMxGDAWBgNV
BAMTD0lkZW5UcnVzdCBFQ0EgMzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoC
ggEBANk/CDdX01aUBBefatM5zr+RBDmTO23vnZQbOOCDV/qX+k3l1urBEqs17YUO
ws7uxCA4+ubPsmb4RRa9V/8uCvAweAT9ppw5l2Kisg+IW6Qj1FCdPfHzv/Kem+DO
QD2AQWdGATfGAeZom02XuMs1dvzTNed11kaPtrL1Oibhj3H32vYkKs/I9rxcgwua
ot2kB9IZlvjr+J15yqbvALR24nwFpTkpvXPfCY+3wwJ3oY3pWuUuHgoKpCwFANrD
IV4JH1ZJjysf+9SJG/QgIfef3+uoDaQNa5KB0t+BSzWa+qul1Lx4HfuoirFgjPDE
J3REQSk05x/HvfkmdWK79qsB0tMCAwEAAaOCAjYwggIyMB0GA1UdDgQWBBT+Y47I
a3q7bhLQ4VLiJKEMLgcvwTAfBgNVHSMEGDAWgBTt5IfQJ8RQ5oQ698z36zpJ/FJO
ITAOBgNVHQ8BAf8EBAMCAYYwEgYDVR0TAQH/BAgwBgEB/wIBADAzBgNVHSAELDAq
MAwGCmCGSAFlAwIBDAEwDAYKYIZIAWUDAgEMAjAMBgpghkgBZQMCAQwDMIHABgNV
HR8EgbgwgbUwLKAqoCiGJmh0dHA6Ly9jcmwuZGlzYS5taWwvY3JsL0VDQVJPT1RD
QTIuY3JsMIGEoIGBoH+GfWxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNkRUNB
JTIwUm9vdCUyMENBJTIwMiUyY291JTNkRUNBJTJjbyUzZFUuUy4lMjBHb3Zlcm5t
ZW50JTJjYyUzZFVTP2NlcnRpZmljYXRlUmV2b2NhdGlvbkxpc3Q7YmluYXJ5MIHT
BggrBgEFBQcBAQSBxjCBwzA6BggrBgEFBQcwAoYuaHR0cDovL2NybC5kaXNhLm1p
```

```
bC9pc3N1ZWR0by9FQ0FST09UQ0EyX0lULnA3YzCBhAYIKwYBBQUHMAKGeGxkYXA6
Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNkRUNBJTIwUm9vdCUyMENBJTIwMiUyY291
JTNkRUNBJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJjYyUzZFVTP2Nyb3NzQ2Vy
dGlmaWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUFAAOCAQEAsLDHwjGsJNIq
Z+eYNsEC7r+XRSST8IUnyZpNMArIWsTBxGzR5Vl3TLhQsmnvzUBVUFmyTiHQuGRI
EA0UCqVDMabQf06JyD/Uq1lvH/SrNaEgXhtVopz1TsleBR9k1e7c75l/BMsX6yag
P16T0h7tEZ2mlLLpOO+C59aPhREc6uGaSzf8hBByP5l4+y1BT0HnX5bgLKybzUc7
zkWpf65SCsjhgAZNgO7sLQaTa9r7ZNn+2oCoJug+pdaBcz+NI4YnIadEm+bjDpYZ
gDEkuS8crPO/imsQezF/MFa9cYLsGx9ldQ1layTsxrX2rcTIZCLbiYjaoeagIHoW
RmforANPiQ==
-----END CERTIFICATE-----
```

### 12.11.97 tls/ECA-ORC-HW4.crt

```
-----BEGIN CERTIFICATE-----
MIIFcDCCBFigAwIBAgIBDjANBgkqhkiG9w0BAQUFADBNMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNFQ0ExFjAUBgNVBAMT
DUVDQSBSb290IENBIDIwHhcNMTEwNjAxMTM0MTMwWhcNMTcwNTMwMTM0MTMwWjBw
MQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQL
EwNFQ0ExIjAgBgNVBAsTGUNlcnRpZmljYXRpb24gQXV0aG9yaXRpZXMxFTATBgNV
BAMTDE9SQyBFQ0EgSFcgNDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
AOB79n1VvgYvD3h8KYKQ5zYEkaDlC/ZTI0aho2J8c+d7bH9glo0X+tQ1pJmqX8Nz
lNXhhOOO0iNlcNFMvXOOujNtlEKlnOHTSajCGK1it2Xg51UVstE1tC2b6FpvRVZ4
R78m+W2HOY+YRoAdxssgXWrH/VtxeMSnwETzin5ajFeeJVl/dEGW/QU63jykjHBt
vek6YhN3VRLmw+JGhDsp0NUn95Xry1+O0dr+Qu5TL4qNtCg20aeDvUEKWoFpTdiF
c/VJ979Km7SI6cfv+FDg4T9YLZtuXnReub5VOZ+EwXLHHFt2ykY3zqTphCaJICqO
rTBhZrF1FEiye3tPj1Qev1UCAwEAAaOCAjYwggIyMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDgYDVR0PAQH/BAQDAgGGMB0GA1UdDgQWBBRbVMHfW3fdLfTnYG0RuJrIXAzn
6TAfBgNVHSMEGDAWgBTt5IfQJ8RQ5oQ698z36zpJ/FJOITAzBgNVHSAELDAqMAwG
CmCGSAFlAwIBDAEwDAYKYIZIAWUDAgEMAjAMBgpghkgBZQMCAQwDMIHABgNVHR8E
gbgwgbUwLKAqoCiGJmh0dHA6Ly9jcmwuZGlzYS5taWwvY3JsL0VDQVJPT1RDQTIu
Y3JsMIGEoIGBoH+GfWxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNkRUNBJTIw
Um9vdCUyMENBJTIwMiUyY291JTNkRUNBJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50
JTJjYyUzZFVTP2NlcnRpZmljYXRlUmV2b2NhdGlvbkxpc3Q7YmluYXJ5MIHTBggr
BgEFBQcBAQSBxjCBwzA6BggrBgEFBQcwAoYuaHR0cDovL2NybC5kaXNhLm1pbC9p
c3N1ZWR0by9FQ0FST09UQ0EyX0lULnA3YzCBhAYIKwYBBQUHMAKGeGxkYXA6Ly9j
cmwuZ2RzLmRpc2EubWlsL2NuJTNkRUNBJTIwUm9vdCUyMENBJTIwMiUyY291JTNk
RUNBJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJjYyUzZFVTP2Nyb3NzQ2VydGlm
aWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUFAAOCAQEACASvDJGPQAneNIKJ
Q+AHY3K/FCEqGN9w1dJlYJAHsWbsPP5ns3A2YO0JRP0vWOMtYX6tpB7/bBGpWodF
Eg//jd+DhjgNzONb2XYCCFInCywjSbD5W8crAxJ999FXlWRRRose0XMEwUqb5Toj
whh9dEOWK+lviMM6yNU7gxsQTgDqpP7jFCTIq+7lsmE05QyGZkf7pZ8spL6rhkNA
fxFRg80XEHoxLmxAU8/53vCiDyCsCwkPezdJkAiYpZY5pgkrz3vkGMwYr8tYsCew
UCd4pMIfXkR8Qewo3Ir6WEwBMfQG6BJ7Lx46Kk4NFetd82lBwGk2jFOCf2xHSlgF
99ZeIw==
-----END CERTIFICATE-----
```

### 12.11.98 tls/ECA-ORC-SW4.crt

```
-----BEGIN CERTIFICATE-----
MIIFcDCCBFigAwIBAgIBDzANBgkqhkiG9w0BAQUFADBNMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNFQ0ExFjAUBgNVBAMT
DUVDQSBSb290IENBIDIwHhcNMTEwNjAxMTM0MzMzWhcNMTcwNTMwMTM0MzMzWjBw
MQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQL
EwNFQ0ExIjAgBgNVBAsTGUNlcnRpZmljYXRpb24gQXV0aG9yaXRpZXMxFTATBgNV
BAMTDE9SQyBFQ0EgU1cgNDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
```

```
AMyn/MU0Al+NhqANfvTXFGkrpCNFruuG1HZT8IgTWlNIBHUEg1+xg0e3b5uLRIfh
LBnrVfD2EyoIwE/LTkkml56sTPTGkNuSoPPY0OoGbTavB1xEWo5ZCfw5/cAskqik
AXplKR4XWPsoUIpCUie0AjIn9z5MfJkkkPQ2zhJSuZCYGyberSQSXTqVPswcs9O/
kteQH7k9rKEAlRYRer+JQSEsMGy1NoPU0Y6V4gyy/eLVfTVqYH0bLA3a/+QqV8a4
ZCkUBaLRBpsLiEx9SMzbXtsZBLT+/VVXXXMGlGUQTMfTMmBBANdZDL5Xu9Fstq/Z
srehbC81MkaFvYJYdqHsWuUCAwEAAaOCAjYwggIyMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDgYDVR0PAQH/BAQDAgGGMB0GA1UdDgQWBBRCnPS6b23pMNc61kb0sjTundJP
JjAfBgNVHSMEGDAWgBTt5IfQJ8RQ5oQ698z36zpJ/FJOITAzBgNVHSAELDAqMAwG
CmCGSAFlAwIBDAEwDAYKYIZIAWUDAgEMAjAMBgpghkgBZQMCAQwDMIHABgNVHR8E
gbgwgbUwLKAqoCiGJmh0dHA6Ly9jcmwuZGlzYS5taWwvY3JsL0VDQVJPT1RDQTIu
Y3JsMIGEoIGBoH+GfWxkYXA6Ly9jcmwuZ2RzLmRpc2EubWlsL2NuJTNkRUNBJTIw
Um9vdCUyMENBJTIwMiUyY291JTNkRUNBJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50
JTJjYyUzZFVTP2NlcnRpZmljYXRlUmV2b2NhdGlvbkxpc3Q7YmluYXJ5MIHTBggr
BgEFBQcBAQSBxjCBwzA6BggrBgEFBQcwAoYuaHR0cDovL2NybC5kaXNhLm1pbC9p
c3N1ZWR0by9FQ0FST09UQ0EyX0lULnA3YzCBhAYIKwYBBQUHMAKGeGxkYXA6Ly9j
cmwuZ2RzLmRpc2EubWlsL2NuJTNkRUNBJTIwUm9vdCUyMENBJTIwMiUyY291JTNk
RUNBJTJjbyUzZFUuUy4lMjBHb3Zlcm5tZW50JTJjYyUzZFVTP2Nyb3NzQ2VydGlm
aWNhdGVQYWlyO2JpbmFyeTANBgkqhkiG9w0BAQUFAAOCAQEAg/gzrTwgRkl1cHJJ
4eOWCJO2xBOr3GjqJzL0Vr/NolqD5KgaK1WiGTbokBfjhz5axN06aOeoJE4UzBEP
Pc5BrlAEu3n48ZuxmEv6zUvhcuHr73rUAtnEyLzyOIhxHvW4GdmqbdaciaZ/R5uc
rg3w3xkltB+dxuNmU44+jk25WESLbYyrwsdl3pQyX3F1JUBwcFXQX6wQE9jpLw7C
m1PPv5e6yScpKRU+2EkQRiekemSlwFV7OdjYzjbUTwxJh5dnG4q8SM0wxGTamQfy
U5ZTW4qw0KMdBi8rsYm2m0Wlzlops4iAj+NKtKuqNzJtmt4PvqVvW9nyVxseycb6
TbYIIA==
-----END CERTIFICATE-----
```

## 12.11.99 tls/ECA-Root.crt

```
-----BEGIN CERTIFICATE-----
MIICmDCCAgGgAwIBAgIBDjANBgkqhkiG9w0BAQUFADBLMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNFQ0ExFDASBgNVBAMT
C0VDQSBSb290IENBMB4XDTA0MDYxNDEwMjAwOVoXDTQwMDYxNDEwMjAwOVowSzEL
MAkGA1UEBhMCVVMxGDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UECxMD
RUNBMRQwEgYDVQQDEwtFQ0EgUm9vdCBDQTCBnzANBgkqhkiG9w0BAQEFAAOBjQAw
gYkCgYEArkr2eXIS6oAKIpDkOlcQZdMGdncoygCEIU+ktqY3of5SVVXU7/it7kJ1
EUzR4ii2vthQtbww9aAnpQxcEmXZk8eEyiGEPy+cCQMllBY+efOtKgjbQNDZ3lB9
19qzUJwBl2BMxslU1XsJQw9SK10lPbQm4asa8E8e5zTUknZBWnECAwEAAaOBizCB
iDAfBgNVHSMEGDAWgBT2uAQnDlYW2blj2f2hVGVBoAhILzAdBgNVHQ4EFgQU9rgE
Jw5WFtm5Y9n9oVRlQaAISC8wDgYDVR0PAQH/BAQDAgGGMA8GA1UdEwEB/wQFMAMB
Af8wJQYDVR0gBB4wHDAMBgpghkgBZQMCAQwBMAwGCmCGSAFlAwIBDAIwDQYJKoZI
hvcNAQEFBQADgYEAHh0EQY2cZ209aBb5q0wW1ER0dc4OGzsLyqjHfaQ4TEaMmUwL
AJRta/c4KVWLiwbODsvgJk+CaWmSL03gRW/ciVb/qDV7qh9Pyd1cOlanZTAnPog2
i82yL3i2fK9DCC84uoxEQbgqK2jx9bIjFTwlAqITk9fGAm5mdT84IEwq1Gw=
-----END CERTIFICATE-----
```

## 12.11.100 tls/ECA-Root2.crt

```
-----BEGIN CERTIFICATE-----
MIIEOjCCAyKgAwIBAgIBBTANBgkqhkiG9w0BAQUFADBNMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNFQ0ExFjAUBgNVBAMT
DUVDQSBSb290IENBIDIwHhcNMDgwNDA0MTQyNDQ5WhcNMjgwMzMwMTQyNDQ5WjBN
MQswCQYDVQQGEwJVUzEYMBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQL
EwNFQ0ExFjAUBgNVBAMTDUVDQSBSb290IENBIDIwggEiMA0GCSqGSIb3DQEBAQUA
A4IBDwAwggEKAoIBAQCzkNge75rtbkexRmFGjaNWCJxKMsdF8tKkpGa+fj2XgO/p
12vp4MA7jeL4EvoTNaTNWoaLsNAvNIz3Be2Hb6cwF9AWxlhGvZX5VhqI1E4eflgf
G6JzFZ1eiwYkE3Y8n2BirbC19OAkwVP32Rw3HeY88PrT8fmhmPE3hdjPAmDwoDiU
```

```
S1EoCcIX/ECkYXkOtIwbIIJi9eSa1imKNvofgcChETFL/lb92L9RkydZ2vWDPkSk
39UgZ+ufAKf73jG2YzEcPfccbAmNf/S6nKh8HwHGgG0BAdg3qP4tfSvCy9bpnLqU
jXYW3syBdQR7QXcVZRXJrAC2v+sc3LmNZC3/NdvFAgMBAAGjggEjMIIBHzAdBgNV
HQ4EFgQU7eSHOCfEUOaEOvfM9+s6SfxSTiEwDgYDVR0PAQH/BAQDAgGGMA8GA1Ud
EwEB/wQFMAMBAf8wgdwGCCsGAQUFBwELBIHPMIHMMEMGCCsGAQUFBzAFhjdodHRw
Oi8vY3JsLmdkcy5kaXNhLm1pbC9nZXRJc3N1ZWRCeT9FQ0ElMjBSb290JTIwQ0El
MjAyMIGEBggrBgEFBQcwBYZ4bGRhcDovL2NybC5nZHMuZGlzYS5taWwvY24lM2RF
Q0ElMjBSb290JTIwQ0ElMjAyJTJjb3UlM2RFQ0ElMmNvJTNkVS5TLiUyMEdvdmVy
bm1lbnQlMmNjJTNkVVM/Y3Jvc3NDZXJ0aWZpY2F0ZVBhaXI7YmluYXJ5MA0GCSqG
SIb3DQEBBQUAA4IBAQBKzBvnhYiTI9m5vs+68TWUwUs8E3Bdgy5OfsFWDy3Bfh8N
6d3+Apl8YOOGqrmGI8MXfscRD+PQQlajHPNwj7GIkbW8DoklZsg/XzYqDj62ZOCM
PUDfp2LgOBj6JG0tI9PAuHRelURUeiGXD1rNw/yVPSIiFd0YYF2FN6lN94H8vUqm
WIxPv8dlof/i0zRIGhcBMvKE894FsmRJmDfNIfWogVtkxdp8WLBhYR+twK3tNLaU
q8ec7GWXKYWUlDPENSSLPc38Nyq7+qS2G5DylVcXSODvqZ2tAOKxZNlneRGDuZro
CzI2ziBn43ptS+peU+UvMEHd+OVUnanXacZ12UVH
-----END CERTIFICATE-----
```

## 12.11.101 tls/ECA-Verisign-G3.crt

```
-----BEGIN CERTIFICATE-----
MIIFxTCCBK2gAwIBAgIBEDANBgkqhkiG9w0BAQUFADBNMQswCQYDVQQGEwJVUzEY
MBYGA1UEChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNFQ0ExFjAUBgNVBAMT
DUVDQSBSb290IENBIDIwHhcNMTEwNzA2MTQwNTM5WhcNMTcwNzA0MTQwNTM5WjCB
mTELMAkGA1UEBhMCVVMxGDAWBgNVBAoTD1UuUy4gR292ZXJubWVudDEMMAoGA1UE
CxMDRUNBMSIwIAYDVQQLExlDZXJ0aWZpY2F0aW9uIEF1dGhvcml0aWVzMT4wPAYD
VQQDEzVWZXJpU2lnbiBDbGllbnQgRXh0ZXJuYWwgQ2VydGlmaWNhdGlvbiBBdXRo
b3JpdHkgLSBHMzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANRyuJwg
XDpXzi7VcxXeaUF5O5ALhmySkeK+fQ3nr7DXYphmssB6VA3XARUzymUUlbV9nrlO
4dChWYPibWlshcTDDuNnNyvxuO6eC+K3Mvx54YUjOPDYqcIXmOESAP5fM7K0h+OP
T+BHNBrk0O+WlE2DFcfOBCfBIKrIhTNgNEq76kiu7uPHvbSTpt8t/a328n5EKICz
hYgA98766RE6gPmNMLd+AobcWTqCwJvjQcA+HzoVjuvAD5gW0AfKURxMZQ2MPe9d
pH+gdJNF7At2qpkZiUDAhosK+PKiMAeF4bJFW5zp1fS84Nbr9SbfbqBaT1ShtAt4
IQN3Qt4XPalq/jMCAwEAAaOCAmEwggJdMBIGA1UdEwEB/wQIMAYBAf8CAQAwDgYD
VR0PAQH/BAQDAgGGMCkGA1UdEQQiMCCkHjAcMRowGAYDVQQDExFWZXJpU2lnbk1Q
S0ktMi02OTAdBgNVHQ4EFgQUsx1ZPOnebXvHtuZh8DB6Mw9QZuQwHwYDVR0jBBgw
FoAU7eSHOCfEUOaEOvfM9+s6SfxSTiEwMwYDVR0gBCwwKjAMBgpghkgBZQMCAQwB
MAwGCmCGSAFlAwIBDAIwDAYKYIZIAWUDAgEMAzCBwAYDVR0fBIG4MIG1MCygKqAo
hiZodHRwOi8vY3JsLmRpc2EubWlsL2NybC9FQ0FST09UQ0EyLmNybDCBhKCBgaB/
hn1sZGFwOi8vY3JsLmdkcy5kaXNhLm1pbC9jbiUzZEVDQSUyMFJvb3QlMjBDQSUy
MDIlMmNvdSUzZEVDQSUyY28lM2RVLlMuJTIwR292ZXJubWVudCUyY2MlM2RVUz9j
ZXJ0aWZpY2F0ZVJldm9jYXRpb25MaXN0O2JpbmFyeTCB0wYIKwYBBQUHAQEEgcYw
gcMwOgYIKwYBBQUHMAKGLmh0dHA6Ly9jcmwuZGlzYS5taWwvaXNzdWVkdG8vRUNB
Uk9PVENBMl9JVC5wN2MwgYQGCCsGAQUFBzAChnhsZGFwOi8vY3JsLmdkcy5kaXNh
Lm1pbC9jbiUzZEVDQSUyMFJvb3QlMjBDQSUyMDIlMmNvdSUzZEVDQSUyY28lM2RV
LlMuJTIwR292ZXJubWVudCUyY2MlM2RVUz9jcm9zc0NlcnRpZmljYXRlUGFpcjti
aW5hcnkwDQYJKoZIhvcNAQEFBQADggEBAHXwkkVTaa4/bk0yBGXf3d68nGbg+0KN
6vFIGmXgp2WAybRuYgws0Xh8O+tHlMik8ve08uxsna8l6WDleDyQbS+TJXVeyVFK
VfGAaPpl+ed5VcVdL/StIyLL1x4a4w/qCNJkSlUf9Nkn5mr6Yd4OQNeqe4LUrebs
L1441z8jClB7Rf+GTZAyoWoC72+4XuaDXY+uNnol5/Zr6dlxpegLpp2ADsLWukY1
UVwwiYDRZDjclMSy+hzG/sneei/CEkTOkeMNs/KwxuaCv+9MZ9+3432kOXE/O5cw
cqankd+BYyZU/BuT4GGU3jHNlK0LkxKBA+fItE9zM966q1AM4j9K7cU=
-----END CERTIFICATE-----
```

## 12.12 postgresql/

For the policy that requires files in this section, see 11.77.3.

### 12.12.1 privs-report.sh

```
#!/bin/bash

# e.g. _times 30 echo hello
_times () { local n=$1 i; shift; for (( i=0; $i < $n; i++ )); do "$@"; done[WRAP]
; }

# e.g. hdecoration 70 1
_habove () {
    local width="$1" level="$2"
    case $level in
        1) echo; _times $width echo -n ’*’; echo;;
        2)       _times $width echo -n ’.’; echo;;
    esac
}

# future expansion
_hbelow () {
    local width="$1" level="$2"
    case $level in
        1) _times $width echo -n ’*’; echo;;
        2) _times $width echo -n ’.’; echo;;
    esac
}

# e.g. header 1 This is some text
header () {
    local width=70 level="$1"
    shift
    local message="$*"
    echo
    _habove $width $level
    # if the message is narrower than width, center it
    if [ ${#message} -lt $width ]; then
        _times $(( ( $width - ${#message} ) / 2 )) echo -n " "
    fi
    echo "$message"
    _hbelow $width $level
    echo
}


# parameters: database, piece of sql
AS_PG_IN () {
    db="$1"; shift; runuser postgres -c "psql -d ’$db’ <<<’$*’"
}
# same, but no text alignment, column headers, or row count
RAW_AS_PG_IN () {
    db="$1"; shift; runuser postgres -c "psql -Atq -d ’$db’ <<<’$*’"
}
```

```
# no parameters; lists connectable databases
databases () {
    # template0 does not allow connections; do not list it
    RAW_AS_PG_IN postgres 'select datname from pg_database where datallowco[WRAP]
nn'
}

header 1 "Roles:"
AS_PG_IN postgres '\du'
header 1 "Databases and database-level privileges:"
# do not show encodings, which \l does
AS_PG_IN postgres 'select datname, datacl from pg_database'
header 1 "Privileges inside each database:"
for db in $(databases); do
    header 2 "$db"
    AS_PG_IN "$db" '\dp'
done
```

## 12.13 puppet/

For the policy that requires files in this section, see 11.80.2.

### 12.13.1 Makefile

```
TEs = $(wildcard *.te)
PPs = $(addsuffix .pp,$(basename $(TEs)))

all: $(PPs)

# Puppet files end with .pp, and so do SELinux policy packages. The
# unified-policy-document has some magic in its Makefile that finds all *.p[WRAP]
p
# files, and we don’t want it to try to treat these as Puppet files, so ins[WRAP]
ide
# the policy we call them *.selinux.pp.

clean:
rm -f *.selinux.pp *.mod

%.pp: %.mod
semodule_package -m $< -o $@
mv $@ $(addsuffix .selinux.pp,$(basename $@))

%.mod: %.te
checkmodule -M -m $< -o $@
```

### 12.13.2 expect_host

```
#!/bin/bash

DOMAIN=eglin.hpc.mil
EXPECTING_DATABASE=/var/spool/sign_expected/db

usage () {
    cat >&2 <<EOF
Usage: expect_host hostname hostname2...
       unexpect_host hostname hostname2...

Expects the given hosts to submit Puppet CSRs; makes ready to turn those in[WRAP]
to
certificates when they appear. Or, removes the expectation that those hosts
will submit Puppet CSRs.

Any unqualified hostnames will have the domain $DOMAIN added to them.

EOF
}

if [ $# = 0 ]; then
    usage
    exit 1
fi
```

```
for hostname; do
    if [[ $hostname != *.* ]]; then
        hostname="$hostname.$DOMAIN"
    fi
    if [ $(basename $0) = unexpect_host ]; then
sqlite3 $EXPECTING_DATABASE "
DELETE FROM expecting_hosts
  WHERE hostname = '$hostname';"
    else
if [ $(sqlite3 $EXPECTING_DATABASE "
  SELECT COUNT(*) FROM expecting_hosts
  WHERE hostname = '$hostname';") -gt 0 ]; then
    echo "$hostname is already expected; updating expectation time" >&2
    sqlite3 $EXPECTING_DATABASE "
  UPDATE expecting_hosts SET entered = '$(date +%s)'
  WHERE hostname = '$hostname';"
else
    sqlite3 $EXPECTING_DATABASE "
  INSERT
  INTO expecting_hosts (entered, hostname)
  VALUES ('$(date +%s)', '$hostname');"
fi
    fi
done
```

### 12.13.3 puppetmaster.selinux.pp

The file puppet/puppetmaster.selinux.pp appears not to be human-readable. It is not included here.

### 12.13.4 puppetmaster.te

```
module puppetmaster 1.0.6;

require {
type httpd_t;
        type puppetmaster_t;
        type passwd_exec_t;
        type sysfs_t;
type puppet_var_lib_t;

type pcscd_t;
type rhnsd_t;
type hald_t;
type puppet_t;
type insmod_t;
type postgresql_t;
type system_dbusd_t;
type cupsd_t;
        type ntpd_t;

        class file { getattr execute append relabelfrom relabelto create wr[WRAP]
ite unlink setattr rename };
class dir { write read create add_name search remove_name getattr rmdir };
}
```

```
allow puppetmaster_t passwd_exec_t:file { getattr execute };
allow puppetmaster_t sysfs_t:dir search;

# allow Puppet master to write report files (overly broad:
# puppet_var_lib_t covers much more than report files)
allow httpd_t puppet_var_lib_t:dir { write read create add_name remove_name[WRAP]
 rmdir };
allow httpd_t puppet_var_lib_t:file { relabelfrom relabelto create write ap[WRAP]
pend unlink setattr rename };

# Puppet master tries to get info about other processes from httpd_t;
# it may be attempting to enforce policy or something. This spams the
# log. Avoid spam:
dontaudit httpd_t cupsd_t:dir getattr;
dontaudit httpd_t hald_t:dir getattr;
dontaudit httpd_t insmod_t:dir getattr;
dontaudit httpd_t pcscd_t:dir getattr;
dontaudit httpd_t postgresql_t:dir getattr;
dontaudit httpd_t puppet_t:dir getattr;
dontaudit httpd_t rhnsd_t:dir getattr;
dontaudit httpd_t system_dbusd_t:dir getattr;
dontaudit httpd_t ntpd_t:dir getattr;
```

### 12.13.5 sign_expected

```
#!/bin/bash

DOMAIN=eglin.hpc.mil

# The interval used here must make sense to date(1).
INTERVAL="48 hours"
EXPECTING_DATABASE=/var/spool/sign_expected/db
CHECK_EVERY_SECONDS=60

usage () {
cat <<EOF >&2

Usage: $0

Sign Puppet certificates for hosts named in the SQLite 3 database
$EXPECTING_DATABASE, when they submit certificate signing requests.

To enter hosts in the expecting database, use the expect_host script.

If a host listed in the database does not submit a CSR within $INTERVAL, it
expires out of the database.

EOF
}

sql () {
sqlite3 -noheader $EXPECTING_DATABASE "$@"
}

sql "CREATE TABLE IF NOT EXISTS
        expecting_hosts
        (entered integer, hostname text);"
```

```
d=$(mktemp -d)

check () {
puppet cert list --all > $d/all
}

exists () {
cat $d/all | grep "^  \"$1\"" >&/dev/null
}

signed () {
cat $d/all | grep "^+ \"$1\"" >&/dev/null
}

sign () {
puppet cert sign $1
}
remove () {
sql "DELETE FROM expecting_hosts
WHERE hostname = ’$1’;"
}

decanonicalize () {
echo "${1%.$DOMAIN}"
}
decanonicalize_many () {
for h; do
echo $(decanonicalize $h)
done
}

log () {
echo "$(date +%Y-%m-%dT%H:%M:%S): $@"
}

expire_hosts () {
expire_if_entered_before=$(date -d "now - $INTERVAL" +%s)
expire_hosts=$(sql "SELECT hostname
                FROM expecting_hosts
WHERE entered < $expire_if_entered_before;")
for xh in $expire_hosts; do
log "$(decanonicalize $xh) expired; removing"
done
sql "DELETE FROM expecting_hosts
WHERE entered < $expire_if_entered_before;"
}

sign_hosts () {
for sh in $(sql "SELECT hostname
FROM expecting_hosts;"); do
if signed $sh; then
log "$(decanonicalize $sh) already signed; removing"
remove $sh
else
if exists $sh; then
log "$(decanonicalize $sh) being signed"
if sign $sh; then
log "$(decanonicalize $sh) signed; removing"
remove $sh
fi
fi
fi
done
}

if [ $# -gt 0 ]; then
usage
exit 1
fi

while true; do
check
expire_hosts
nexpected=$(sql "SELECT COUNT(*) FROM expecting_hosts")
if [ "$nexpected" -gt 0 ]; then
log "expecting these hosts: $(echo $(decanonicalize_many \
$(sql "SELECT hostname FROM expecting_hosts ORDER BY hostname")))"
else
log "not expecting any hosts"
fi
sign_hosts
sleep $CHECK_EVERY_SECONDS
done
```

## 12.14 root/

For the policy that requires files in this section, see 11.83.1.

### 12.14.1 bashrc.default

```
# .bashrc

############################################################
## This file is automatically overwritten by the policy. ##
############################################################

# User specific aliases and functions

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

# Source global definitions
if [ -f /etc/bashrc ]; then
. /etc/bashrc
fi

# \implements{unixstig}{GEN000940,GEN000945,GEN00950}%
# Make sure that the PATH and LD_LIBRARY_PATH are "the vendor default and
# contain only absolute paths," and that LD_PRELOAD is empty---\emph{after}[WRAP]
 all
# other settings have been established.
export PATH=/bin:/sbin:/usr/bin:/usr/sbin
export LD_LIBRARY_PATH=
export LD_PRELOAD=

# \implementsunixstig{GEN000960} Make sure there are no world writable
# directories in the PATH.
OIFS="$IFS"
IFS=:
insecure_path=0
for d in $PATH; do
    if [[ $(stat -c %a $d) = *[2367] ]]; then
        echo "DIRECTORY $d ON PATH IS WORLD WRITABLE!!" >&2
        insecure_path=1
    fi
done
IFS="$OIFS"

# If there are world-writable entries on the path, get rid of the whole pat[WRAP]
h.
# The (now-root) user can sort it out.
if [ "$insecure_path" = 1 ]; then
    export PATH=
    echo "PATH VARIABLE HAS BEEN EMPTIED" >&2
fi

trap '' SIGINT
echo
echo "Who are you and what are you doing?"
echo "Press Ctrl-D on an empty line when finished explaining."
sed 's/[[:cntrl:]]/(CONTROL CHAR)/g' | \
    logger -t "ROOT LOGIN, user said"
echo "What you typed has been logged. Continuing."
trap - SIGINT
```

### 12.14.2 bashrc.no_questions

```
# .bashrc

##########################################################
## This file is automatically overwritten by the policy. ##
##########################################################

# User specific aliases and functions

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

# Source global definitions
if [ -f /etc/bashrc ]; then
. /etc/bashrc
fi

# \implements{unixstig}{GEN000940,GEN000945,GEN00950}%
# Make sure that the PATH and LD_LIBRARY_PATH are "the vendor default and
# contain only absolute paths," and that LD_PRELOAD is empty---\emph{after}[WRAP]
 all
# other settings have been established.
export PATH=/bin:/sbin:/usr/bin:/usr/sbin
export LD_LIBRARY_PATH=
export LD_PRELOAD=

# \implementsunixstig{GEN000960} Make sure there are no world writable
# directories in the PATH.
OIFS="$IFS"
IFS=:
insecure_path=0
for d in $PATH; do
    if [[ $(stat -c %a $d) = *[2367] ]]; then
        echo "DIRECTORY $d ON PATH IS WORLD WRITABLE!!" >&2
        insecure_path=1
    fi
done
IFS="$OIFS"

# If there are world-writable entries on the path, get rid of the whole pat[WRAP]
h.
# The (now-root) user can sort it out.
if [ "$insecure_path" = 1 ]; then
    export PATH=
    echo "PATH VARIABLE HAS BEEN EMPTIED" >&2
fi
```

### 12.14.3 login/securetty

```
console
vc/1
vc/2
vc/3
vc/4
vc/5
vc/6
vc/7
vc/8
vc/9
vc/10
vc/11
tty1
tty2
tty3
tty4
tty5
tty6
tty7
tty8
tty9
tty10
tty11
```

## 12.15 rpm/

For the policy that requires files in this section, see 11.84.5.

### 12.15.1 rpmV.cron

```
#!/bin/sh

# These functions reject lines of output from rpm -Va, for various
# reasons, in order to bring the unexpected changes to the forefront.

# We’ve changed a bunch of config files using this very policy.
reject_config_files () {
    grep -v ’^[^[:space:]]\+  c ’
}

# A couple of them are deleted by this policy.
reject_missing_config_file_namely () {
    grep -v ’^missing   c ’"$1"
}

# Some symlinks are changed.
reject_changed_symlink_namely () {
    grep -v ’^\.\.\.\.L\.\.\.\.    ’"$1"
}

reject_missing_file_namely () {
    grep -v ’missing     ’"$1"
}

reject_changed () {
    grep -v ’[.S]\.[.5]\.\.\.\.T\.    ’"$1"
}

reject_changed_files_under () {
    reject_changed "$1/.*"
}

# We’ve deleted some kernel modules (see the ’network’ Puppet module).
reject_deleted_kernel_modules_in () {
    grep -v ’^missing[[:space:]]\+/lib/modules/.*/’"$1"
}


# We’ve changed the mode, owner or group of some configuration
# files. If such are changed outside the purview of this policy, that
# may be a significant event; but it will be caught by AIDE and
# auditable.
reject_changes_solely_in_mode_owner_or_group () {
    # this will also reject lines which start with ’........’, but
    # there aren’t any of those: if there were no changes, rpm -Va
    # would not print a line
    grep -v ’^\.\(M\|\.\)\.\.\.\(U\|\.\)\(G\|\.\)\.\.’
}

# The NVIDIA driver changes some OpenGL files.
```

```
reject_nvidia_changes () {
    reject_changed_symlink_namely /usr/lib64/xorg/modules/extensions/libglx[WRAP]
.so | \
    reject_changed_symlink_namely /usr/lib64/libGL.so.1
}

# We remove the PackageKit update icon, because updating packages
# isn't done by users around here.
reject_package_updater_removal () {
    reject_missing_file_namely /etc/xdg/autostart/gpk-update-icon.desktop
}

# It seems Centrify overwrites its own configuration files during
# operation.
reject_centrify_changes () {
    reject_changed_files_under /etc/centrifydc             | \
    reject_changed             /etc/init.d/centrifydc      | \
    reject_changed             /etc/logrotate.d/centrifydc
}

# It seems McAfee CMA overwrites its own configuration files during
# operation.
reject_mcafee_changes () {
    reject_changed_files_under /etc/cma\\.d       | \
    reject_changed_files_under /opt/McAfee     | \
    reject_missing_file_namely /opt/McAfee/cma/scratch/Server\\.xml | \
    reject_missing_file_namely /opt/McAfee/cma/srpubkey\\.bin
}

# I don't know why this is gone, but if we ever want no KACE agent on
# a system, we can re-kickstart it or something.
reject_kace_changes () {
    reject_missing_file_namely /opt/dell/kace/bin/RemoveKbox50
}

# Not sure what does this change:
# --- sshd 2012-12-13 07:50:45.000000000 -0600
# +++ sshd.changed 2013-08-01 12:57:38.098355483 -0500
# @@ -130,7 +130,6 @@
#   [ -f /etc/ssh/sshd_config ] || exit 6
#   # Create keys if necessary
#   if [ "x${AUTOCREATE_SERVER_KEYS}" != xNO ]; then
# -do_rsa1_keygen
#   do_rsa_keygen
#   do_dsa_keygen
#   fi
reject_sshd_init_script_change () {
    reject_changed /etc/rc.d/init.d/sshd
}


reject_expected_changes () {
    # the cat is so that every reject_* command will always end with a
    # | \
    reject_config_files                               | \
    reject_deleted_kernel_modules_in firewire         | \
    reject_deleted_kernel_modules_in dccp             | \
    reject_deleted_kernel_modules_in rds              | \
    reject_deleted_kernel_modules_in sctp             | \
    reject_deleted_kernel_modules_in bluetooth        | \
    reject_changes_solely_in_mode_owner_or_group      | \
    reject_missing_config_file_namely /etc/cron.deny  | \
    reject_missing_config_file_namely /etc/at.deny    | \
    reject_package_updater_removal                    | \
    reject_nvidia_changes                             | \
    reject_centrify_changes                           | \
    reject_mcafee_changes                             | \
    reject_kace_changes                               | \
    reject_changed /etc/init/control-alt-delete.conf  | \
    reject_sshd_init_script_change                    | \
    cat
}

rpm -Va | reject_expected_changes
```

## 12.16 sbu/

For the policy that requires files in this section, see 11.88.4.

### 12.16.1 selinux/Makefile

```
TEs = $(wildcard *.te)
PPs = $(addsuffix .pp,$(basename $(TEs)))

all: $(PPs)

# Puppet files end with .pp, and so do SELinux policy packages. The
# unified-policy-document has some magic in its Makefile that finds all *.p[WRAP]
p
# files, and we don't want it to try to treat these as Puppet files, so ins[WRAP]
ide
# the policy we call them *.selinux.pp.

clean:
rm -f *.selinux.pp *.mod

%.pp: %.mod
semodule_package -m $< -o $@
mv $@ $(addsuffix .selinux.pp,$(basename $@))

%.mod: %.te
checkmodule -M -m $< -o $@
```

### 12.16.2 selinux/sbu_apps.selinux.pp

The file `sbu/selinux/sbu_apps.selinux.pp` appears not to be human-readable. It is not included here.

### 12.16.3 selinux/sbu_apps.te

```
module sbu_apps 1.0.0;

require {
        type httpd_sys_script_t;
        type devlog_t;
        type syslogd_t;
        class sock_file write;
        class unix_dgram_socket sendto;
}

# Allow scripts that httpd runs to log errors.
allow httpd_sys_script_t devlog_t:sock_file write;
allow httpd_sys_script_t syslogd_t:unix_dgram_socket sendto;
```

### 12.16.4 trac/classbar.html

```
<html xmlns="http://www.w3.org/1999/xhtml"
```

```
      xmlns:py="http://genshi.edgewall.org/"
      py:strip="">

  <!--! Add security label style sheet -->
  <head py:match="head" py:attrs="select('@*')">
    ${select('*|comment()|text()')}
    <link rel="stylesheet" type="text/css"
          href="/styles/classbar.css" />
  </head>

  <body py:match="body" py:attrs="select('@*')">
    <!--! Add security label header -->
    <div id="siteheader">
        <div class="unclassified classbar">
            <span class="classtext">UNCLASSIFIED//FOUO</span>
        </div>
    </div>
    ${select('*|text()')}
  </body>
</html>
```

### 12.16.5 trac/site.html

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:xi="http://www.w3.org/2001/XInclude"
      xmlns:py="http://genshi.edgewall.org/"
      py:strip="">

  <xi:include href="classbar.html"><xi:fallback /></xi:include>

</html>
```

### 12.16.6 trac/trac.wsgi

```
#!/usr/bin/python
import trac.web.main
application = trac.web.main.dispatch_request
```

## 12.17 searde_svn/

For the policy that requires files in this section, see 11.91.4.

### 12.17.1 selinux/Makefile

```
TEs = $(wildcard *.te)
PPs = $(addsuffix .pp,$(basename $(TEs)))

all: $(PPs)

# Puppet files end with .pp, and so do SELinux policy packages. The
# unified-policy-document has some magic in its Makefile that finds all *.p[WRAP]
p
# files, and we don't want it to try to treat these as Puppet files, so ins[WRAP]
ide
# the policy we call them *.selinux.pp.

clean:
rm -f *.selinux.pp *.mod

%.pp: %.mod
semodule_package -m $< -o $@
mv $@ $(addsuffix .selinux.pp,$(basename $@))

%.mod: %.te
checkmodule -M -m $< -o $@
```

### 12.17.2 selinux/sbu_apps.selinux.pp

The file searde_svn/selinux/sbu_apps.selinux.pp appears not to be human-readable. It is not included here.

### 12.17.3 selinux/sbu_apps.te

```
module sbu_apps 1.0.0;

require {
        type httpd_sys_script_t;
        type devlog_t;
        type syslogd_t;
        class sock_file write;
        class unix_dgram_socket sendto;
}

# Allow scripts that httpd runs to log errors.
allow httpd_sys_script_t devlog_t:sock_file write;
allow httpd_sys_script_t syslogd_t:unix_dgram_socket sendto;
```

### 12.17.4 trac/classbar.html

```
<html xmlns="http://www.w3.org/1999/xhtml"
```

```
      xmlns:py="http://genshi.edgewall.org/"
      py:strip="">

  <!--! Add security label style sheet -->
  <head py:match="head" py:attrs="select('@*')">
    ${select('*|comment()|text()')}
    <link rel="stylesheet" type="text/css"
          href="/styles/classbar.css" />
  </head>

  <body py:match="body" py:attrs="select('@*')">
    <!--! Add security label header -->
    <div id="siteheader">
        <div class="unclassified classbar">
            <span class="classtext">UNCLASSIFIED//FOUO</span>
        </div>
    </div>
    ${select('*|text()')}
  </body>
</html>
```

### 12.17.5 trac/site.html

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:xi="http://www.w3.org/2001/XInclude"
      xmlns:py="http://genshi.edgewall.org/"
      py:strip="">

  <xi:include href="classbar.html"><xi:fallback /></xi:include>

</html>
```

### 12.17.6 trac/trac.wsgi

```
#!/usr/bin/python
import trac.web.main
application = trac.web.main.dispatch_request
```

## 12.18 shell/

For the policy that requires files in this section, see 11.93.

### 12.18.1 valid-shells

```
#!/bin/sh
IFS="
"
for line in $(cat /etc/passwd); do
    user=$(echo "$line" | cut -d: -f1)
    shell=$(echo "$line" | cut -d: -f7)
    if ! grep "^$shell\$" /etc/shells >&/dev/null; then
        echo "User $user has invalid shell \"$shell\"; \
changing to /sbin/nologin"
        chsh -s /sbin/nologin "$user"
    fi
done
```

## 12.19 stig_misc/

For the policy that requires files in this section, see 11.100.14.

### 12.19.1 device_files/device-files.cron

```
#!/bin/sh
# Find extraneous device files on local filesystems.
# /dev is its own filesystem, so any device file on a local disk-based
# filesystem is extraneous.
for fstype in ext4 ext3 ext2 xfs; do
    # mount says things like this:
    # "/dev/vda2 on / type ext3 (rw)"
    # we want the /
    for fs in $(mount -t $fstype | cut -d' ' -f3); do
        # -xdev: do not cross into another mount.
        find $fs -xdev -type b -o -type c -printf \
            'EXTRANEOUS DEVICE FILE: %f\n'
    done
done
```

### 12.19.2 login_history/gdm-post-login.sh

```
#!/bin/sh
# Fulfill AFMAN 33-223, section 5.5.2, and UNIX SRG rules GEN000452 and
# GEN000454.
text="`/usr/sbin/loginhistory $LOGNAME`"
[[ "$text" =~ \! ]] && sw=--error || sw=--info
zenity $sw --text="$text"
```

## 12.20 usb/

For the policy that requires files in this section, see 11.111.1.

### 12.20.1 mass_storage/admin-udisks.pkla

```
[require admin authentication for disk actions]
Identity=*
Action=org.freedesktop.udisks.*
ResultAny=auth_admin
ResultActive=auth_admin
ResultInactive=auth_admin
```

# Chapter 13

# Attendant templates

Here follow the template files used by the policy.

Wherever you see `[WRAP]` at the end of a line, that line was wrapped in order to fit on the page; if you find yourself in the unfortunate position of typing that line into a computer, do not type [WRAP] and do not start a new line. Lines not ending with `[WRAP]` end with a newline in the original text of the file.

Wherever you see something like `[UNICODE \u5678 MAYBE SOME WORDS]`, the original text of the file contained a Unicode character which could not be reproduced exactly in this document. If the Unicode character database includes a description of the character, it is included; if not, only the character's identity is included.

## 13.1 contingency_backup/

For the policy that requires files in this section, see 11.21.4.

### 13.1.1 cron.erb

```
#!/bin/bash
#
# Automatically back up the policy onto optical media, so that everything
# necessary to implement this policy will be ready to hand in case of any
# contingency.
#
# Do the backup every month. Be willing to try several times. Any qualified
# host can do a backup, and if one goes down, another should in fact do it.[WRAP]
 If
# one host successfully completes a backup, all hosts should stop trying un[WRAP]
til
# next month. (The multiple tries are the reason why this script is run dai[WRAP]
ly,
# even though the backup is a monthly product.)
#
# We use stamp files on an NFS-mounted filesystem to broadcast the fact of [WRAP]
a
# successful backup. If the host doing the backup automounts, the check for[WRAP]
 the
# stamp file could cause the filesystem to be mounted, and if the host is n[WRAP]
ot
# properly on the network, that could hang. But this is a cron job; it has [WRAP]
all
# day.
STAMP_DIR=<%= stamp_directory %>

#
# These days we're making a DVD, and piping the iso straight to the drive
# rather than making it ahead. So we should only need 5GB. This figure is i[WRAP]
n
# KiB:
SPACE_NEEDED=5000000

set -e

# If there has already been a successful backup this month, go no further. [WRAP]
The
# existence of this month's stamp file will let us know a successful backup
# has happened already.

stamp_file=$(/bin/date +'%Y-%m-backed-up')
if [ -f "$STAMP_DIR/$stamp_file" ]; then
    exit 0
fi

# Try backups only when the day number is in the twenties. Exit otherwise.

if [[ $(date +%d) != 2? ]]; then
    exit 0
fi
```

```
# Keep a copy of the policy - including the backup scripts - checked out in
# root's home. Routinely destroy local modifications to this working copy i[WRAP]
n
# order to make sure that what we have is exactly what is in the repository[WRAP]
.
# The variable wc should not have any spaces in it: if it did, one inadequa[WRAP]
tely
# quoted name in any level of scripts or utilities under this one could cau[WRAP]
se
# the whole backup operation to fail.
#
# Make sure there's enough room too.

ao=critical-backup--AUTOMATICALLY-OVERWRITTEN
wc_has_enough=0
for wc in /tmp/$ao /var/tmp/$ao; do
    rm -rf $wc
    mkdir -p $wc
    # Filesystem 1K-blocks Used Available Use% Mounted-on
    # But if the device (Filesystem) is long, the line will be split, so co[WRAP]
unt
    # from the right instead.
    free=$(df -k $wc | tail -n 1 | awk '{print $(NF-2)}')
    if [ $free -ge $SPACE_NEEDED ]; then
        wc_has_enough=1
        break
    fi
done
if [ $wc_has_enough = 0 ]; then
    echo "Could not find a temp dir with enough space! Aborting." >&2
    exit 42
fi


# HOME=/root: if root has cached authentication credentials, use them to ta[WRAP]
lk
# to the Subversion server.

/usr/bin/env HOME=/root \
    /usr/bin/svn --non-interactive --username $(hostname -s) \
    co -q <%= contingency_backup_url -%> \
    "$wc"

chmod -R go-rwx "$wc"
chown -R nobody "$wc"


# Get the reStructuredText utilities onto the path: they are needed to buil[WRAP]
d
# the SBU manual. HOME=/root: as above.

cd $wc

<%
if @add_to_path
    to_add_to_path = if @add_to_path.is_a?(Array); @add_to_path; else [@add[WRAP]
_to_path]; end
    path_addition = '\:' + to_add_to_path.join('\:')
else
    path_addition = ''
end
if @add_to_pythonpath
    to_add_to_pythonpath = if @add_to_pythonpath.is_a?(Array); @add_to_pyth[WRAP]
onpath; else [@add_to_pythonpath]; end
    pythonpath = to_add_to_pythonpath.join('\:')
else
    pythonpath = ''
end
%>
# run the documentation builds as nobody to lower our security profile
runuser -s /bin/bash nobody -c "/usr/bin/env \
     PATH=/bin\:/sbin\:/usr/bin\:/usr/sbin<%= path_addition -%> \
     PYTHONPATH=<%= pythonpath -%> \
     make"

# burn has to run as root, for access to the optical disc writer device
/usr/bin/env HOME=/root \
     PATH=/bin\:/sbin\:/usr/bin\:/usr/sbin \
     make burn


touch $stamp_dir/$stamp_file
```

## 13.2 dod_login_warnings/

For the policy that requires files in this section, see 11.29.1.

### 13.2.1 paragraphs

```
You are accessing a U.S. Government (USG) information system (IS) that is p[WRAP]
rovided for USG-authorized use only. By using this IS (which includes any d[WRAP]
evice attached to this IS), you consent to the following conditions:

- The USG routinely intercepts and monitors communications on this IS for p[WRAP]
urposes including, but not limited to, penetration testing, COMSEC monitori[WRAP]
ng, network operations and defence, personnel misconduct (PM), law enforcem[WRAP]
ent (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 sub[WRAP]
ject to routine monitoring, interception, and search, and may be disclosed [WRAP]
or used for any USG-authorized purpose.

- This IS includes security measures (e.g., authentication and access contr[WRAP]
ols) to protect USG interests--not for your personal benefit or privacy.

- Notwithstanding the above, using this IS does not constitute consent to P[WRAP]
M, LE or CI investigative searching or monitoring of the content of privile[WRAP]
ged communications, or work product, related to personal representation or [WRAP]
services by attorneys, psychotherapists, or clergy, and their assistants.  [WRAP]
Such communications and work product are private and confidential. See User[WRAP]
 Agreement for details.
```

## 13.3 filers/

For the policy that requires files in this section, see 11.31.1.

### 13.3.1 users_to_filer.cron

```
#!/bin/bash

# Try not to run this at the same time as other hosts.
sleep $(( $RANDOM % 60 ))

FILER_ETC=<%=etc_dir%>

# To avoid race conditions if multiple hosts try to run this at the
# same time:
SUFFIX=$(hostname -s).$$.$(date +%s)

set -e



# gather all system users from filer and write in new passwd file
cat $FILER_ETC/passwd | (IFS=’
’; while read line; do
    uid=$(echo "$line" | cut -d: -f3)
    if [ $uid -le 1000 -o $uid -eq 65533 \
         -o $uid -eq 65534 -o $uid -eq 65535 ]; then
        echo $line;
    fi; done) > $FILER_ETC/passwd.new.$SUFFIX

# now gather all non-system users from my passwd database, and add
# them to the system users
getent passwd | (IFS=’
’; while read line; do
    uid=$(echo "$line" | cut -d: -f3)
    if [ $uid -gt 1000 -a $uid -ne 65533 \
         -a $uid -ne 65534 -a $uid -ne 65535 ]; then
        echo $line;
    fi; done) >> $FILER_ETC/passwd.new.$SUFFIX


# same with system groups
cat $FILER_ETC/group | (IFS=’
’; while read line; do
    gid=$(echo "$line" | cut -d: -f3)
    if [ $gid -le 1000 -o $gid -eq 65533 \
        -o $gid -eq 65534 -o $gid -eq 65535 ]; then
        echo $line
    fi; done) > $FILER_ETC/group.new.$SUFFIX


# same with non-system groups
getent group | (IFS=’
’; while read line; do
    gid=$(echo "$line" | cut -d: -f3)
    if [ $gid -gt 1000 -a $gid -ne 65533 \
        -a $gid -ne 65534 -a $gid -ne 65535 ]; then
        echo $line
    fi; done) >> $FILER_ETC/group.new.$SUFFIX

maybe_backup_then_replace () {
    local new="$1"
    local orig="$2"
    if [ "$(cat $orig | sha256sum)" != "$(cat $new | sha256sum)" ]; then
cp $orig $orig.backup.$(date +'%Y_%m_%d_%H_%M_%S').$SUFFIX
mv $new $orig
    else
        rm $new
    fi
}
maybe_backup_then_replace $FILER_ETC/passwd.new.$SUFFIX $FILER_ETC/passwd
maybe_backup_then_replace $FILER_ETC/group.new.$SUFFIX  $FILER_ETC/group
```

## 13.4 ip6tables/

For the policy that requires files in this section, see 11.46.1.

### 13.4.1 katello-1.3-server

```
<% # variables needed:
   #     site: a CIDR block expressing the LAN this host will be on.
-%>
<%=scope.function_template "ip6tables/pieces/preamble"-%>
<%=scope.function_template "ip6tables/pieces/connected"-%>
<%=scope.function_template "ip6tables/pieces/loopback"-%>
<%=scope.function_template "ip6tables/pieces/dhcp-client"-%>
<%=scope.function_template "ip6tables/pieces/input-icmp"-%>
<%=scope.function_template "ip6tables/pieces/output-icmp"-%>
<%=scope.function_template "ip6tables/pieces/dns"-%>
<%=scope.function_template "ip6tables/pieces/puppet-client"-%>
<%=scope.function_template "searde/ip6tables/pieces/satellite-client"-%>
<%=scope.function_template "ip6tables/pieces/ssh-server"-%>
<%=scope.function_template "ip6tables/pieces/ssh-client"-%>

<%=scope.function_template "ip6tables/pieces/katello-qpid"-%>

<%=scope.function_template "ip6tables/pieces/source-routed"-%>
<%=scope.function_template "ip6tables/pieces/mdns"-%>

<%=scope.function_template "ip6tables/pieces/fallthrough"-%>

COMMIT
```

### 13.4.2 pieces/connected

```
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
```

### 13.4.3 pieces/dhcp-client

```
# Allow DHCP requests to go out, responses in.
# We have no way of knowing, from the client, what exact DHCP server will
# respond.
# RFC 3315 covers DHCPv6.
-A OUTPUT -s fe80::/16 -d ff02::1:2 -p udp -m udp --sport dhcpv6-client --d[WRAP]
port dhcpv6-server -j ACCEPT
-A INPUT -s <%=site-%> -d fe80::/16 -p udp -m udp --sport dhcpv6-server --d[WRAP]
port dhcpv6-client -j ACCEPT
```

### 13.4.4 pieces/dns

```
# DNS client
-A INPUT -p udp -m udp --sport domain -j ACCEPT
-A OUTPUT -p udp -m udp --dport domain -j ACCEPT
```

### 13.4.5 pieces/fallthrough

```
-A INPUT -j LOG --log-prefix "INPUT fallthrough: "
-A OUTPUT -j LOG --log-prefix "OUTPUT fallthrough: "
```

### 13.4.6 pieces/http-https-client

```
# There is a place for limiting outgoing web page requests.
# As of right now that place is not at the client.
-A OUTPUT -p tcp -m tcp --dport 80 -j ACCEPT
-A OUTPUT -p tcp -m tcp --dport 443 -j ACCEPT
```

### 13.4.7 pieces/input-icmp

```
# ICMPv6 is a different animal from ICMPv4. Many more of its message types [WRAP]
are
# necessary and useful. It doesn’t have a timestamp request message type li[WRAP]
ke
# ICMPv4 (the STIG requires ICMP timestamps to be blocked; see the iptables
# module).
#
# Allow loopback ICMP.
-A INPUT -p icmpv6 -m ipv6header --soft ! --header frag -s ::1 -j ACCEPT
# Allow ICMP within the enclave.
-A INPUT -p icmpv6 -m ipv6header --soft ! --header frag -s <%=site-%> -j AC[WRAP]
CEPT
# Allow link-local ICMP. This encompasses router advertisements, multicast
# listener reports, neighbor discovery, etc.
-A INPUT -p icmpv6 -m ipv6header --soft ! --header frag -s fe80::/10 -j ACC[WRAP]
EPT
```

### 13.4.8 pieces/katello-qpid

```
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 443 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 5671 -j ACCEPT
```

### 13.4.9 pieces/krb5-client

```
-A OUTPUT -p tcp -m tcp --dport 88 -j ACCEPT
```

### 13.4.10 pieces/loopback

```
-A INPUT -s ::1 -d ::1 -j ACCEPT
-A OUTPUT -s ::1 -d ::1 -j ACCEPT
```

### 13.4.11 pieces/mdns

```
# GEN007850 says not to send dynamic DNS updates "unless needed." mDNS is n[WRAP]
ot
# strictly the same, but its purpose is also to "transmit unencrypted
# information about a system including its name and address." As we don’t
# presently need mDNS, we can just turn it off without questioning the sani[WRAP]
ty
# of such a dictum.
-A OUTPUT -d ff02::fb -p udp -m udp --dport 5353 -j DROP
```

### 13.4.12 pieces/ntp-client

```
-A OUTPUT -p udp --dport 123 -j ACCEPT
```

### 13.4.13 pieces/ntp-server

```
-A INPUT -s <%=site-%> -p udp --dport 123 -j ACCEPT
```

### 13.4.14 pieces/output-icmp

```
# Allow loopback ICMP.
-A OUTPUT -p icmpv6 -m ipv6header --soft ! --header frag -d ::1 -j ACCEPT
# Allow ICMP within the enclave.
-A OUTPUT -p icmpv6 -m ipv6header --soft ! --header frag -d <%=site-%> -j A[WRAP]
CCEPT
# Allow link-local ICMP. This encompasses router advertisements, multicast
# listener reports, neighbor discovery, etc.
-A OUTPUT -p icmpv6 -m ipv6header --soft ! --header frag -d fe80::/10 -j AC[WRAP]
CEPT
# Interface-local multicast ICMP.
-A OUTPUT -p icmpv6 -m ipv6header --soft ! --header frag -d ff01::/8 -j ACC[WRAP]
EPT
# Link-local multicast ICMP.
-A OUTPUT -p icmpv6 -m ipv6header --soft ! --header frag -d ff02::/8 -j ACC[WRAP]
EPT
```

### 13.4.15 pieces/output-smtp

```
-A OUTPUT -p tcp -m tcp --dport 25 -j ACCEPT
```

### 13.4.16 pieces/preamble

```
*filter
# UNIX SRG GEN008540: drop by default.
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
```

### 13.4.17 pieces/puppet-client

```
# Puppet client. We should nail down better exactly which host it’s talking[WRAP]
 to
# - but that would require each firewall rule to be a resource, an approach[WRAP]
 we
# rejected. (Then the Puppet host could export a resource allowing clients [WRAP]
to
# connect to it.)
-A OUTPUT -d <%=site-%> -p tcp --dport 8140 -j ACCEPT
```

### 13.4.18 pieces/puppet-master

```
# Puppet master. This rule assumes that Puppet masters always listen to the
# whole enclave.
-A INPUT -s <%=site-%> -p tcp --dport 8140 -m state --state NEW -j ACCEPT
```

### 13.4.19 pieces/rsyslog-client

```
# Rsyslog client. We connect using SSL to the loghost and forward log messa[WRAP]
ges.
# (Use this piece only on hosts which include log::rsyslog::client, which p[WRAP]
uts
# an entry for loghost in /etc/hosts.)
-A OUTPUT -p tcp -m tcp -d loghost --dport 10514 -j ACCEPT
```

### 13.4.20 pieces/source-routed

```
# GEN003605, GEN003606: drop all source routed packets; input, output and
# forwarding.
#
# In IPv6 source routing is accomplished with a routing header of type 0,
# commonly known as RH0. See http://lwn.net/Articles/232781/.

-A INPUT -m rt --rt-type 0 -j DROP
-A OUTPUT -m rt --rt-type 0 -j DROP
-A FORWARD -m rt --rt-type 0 -j DROP

# According to http://www.sixxs.net/faq/connectivity/?faq=filters, "RH0
# processing is disabled per default since Linux 2.6.20.9," but only in an
# INPUT sense: RH0 headers could still be forwarded, and the above rules wi[WRAP]
ll
# stop that from happening.
```

### 13.4.21 pieces/ssh-client

```
-A OUTPUT -p tcp -m tcp --dport ssh -j ACCEPT
```

### 13.4.22 pieces/ssh-server

```
# Serve ssh
-A INPUT -s <%=site-%> -p tcp -m tcp --dport ssh -j ACCEPT
-A OUTPUT -p tcp -m tcp --sport ssh -j ACCEPT
```

### 13.4.23 puppetmaster

```
<% # variables needed:
   #     site: a CIDR block expressing the LAN this host will be on.
-%>
<%=scope.function_template "ip6tables/pieces/preamble"-%>
<%=scope.function_template "ip6tables/pieces/connected"-%>
<%=scope.function_template "ip6tables/pieces/loopback"-%>
<%=scope.function_template "ip6tables/pieces/dhcp-client"-%>
<%=scope.function_template "ip6tables/pieces/input-icmp"-%>
<%=scope.function_template "ip6tables/pieces/output-icmp"-%>
<%=scope.function_template "ip6tables/pieces/dns"-%>
<%=scope.function_template "ip6tables/pieces/puppet-client"-%>
<%=scope.function_template "searde/ip6tables/pieces/satellite-client"-%>
<%=scope.function_template "searde/ip6tables/pieces/kace-client"-%>
<%=scope.function_template "ip6tables/pieces/ssh-server"-%>
<%=scope.function_template "ip6tables/pieces/ssh-client"-%>
<%=scope.function_template "ip6tables/pieces/ntp-client"-%>
<%=scope.function_template "ip6tables/pieces/ntp-server"-%>

<%=scope.function_template "ip6tables/pieces/puppet-master"-%>

<%=scope.function_template "ip6tables/pieces/source-routed"-%>
<%=scope.function_template "ip6tables/pieces/mdns"-%>

<%=scope.function_template "ip6tables/pieces/fallthrough"-%>

COMMIT
```

### 13.4.24 workstation

```
<% # variables needed:
   #     site: a CIDR block expressing the LAN this host will be on.
-%>
<%=scope.function_template "ip6tables/pieces/preamble"-%>
<%=scope.function_template "ip6tables/pieces/connected"-%>
<%=scope.function_template "ip6tables/pieces/loopback"-%>
<%=scope.function_template "ip6tables/pieces/dns"-%>
<%=scope.function_template "ip6tables/pieces/puppet-client"-%>
<%=scope.function_template "ip6tables/pieces/dhcp-client"-%>
<%=scope.function_template "searde/ip6tables/pieces/satellite-client"-%>
<%=scope.function_template "searde/ip6tables/pieces/mcafee-hbss-client"-%>
<%=scope.function_template "searde/ip6tables/pieces/searde-ad-ldap-client"-[WRAP]
%>
<%=scope.function_template "searde/ip6tables/pieces/kace-client"-%>
<%=scope.function_template "ip6tables/pieces/ssh-server"-%>
<%=scope.function_template "ip6tables/pieces/ssh-client"-%>
<%=scope.function_template "ip6tables/pieces/krb5-client"-%>
<%=scope.function_template "ip6tables/pieces/http-https-client"-%>
<%=scope.function_template "ip6tables/pieces/output-smtp"-%>
<%=scope.function_template "ip6tables/pieces/input-icmp"-%>
<%=scope.function_template "ip6tables/pieces/output-icmp"-%>
```

```
<%=scope.function_template "ip6tables/pieces/source-routed"-%>
<%=scope.function_template "ip6tables/pieces/mdns"-%>

<%=scope.function_template "ip6tables/pieces/fallthrough"-%>

COMMIT
```

## 13.5 iptables/

For the policy that requires files in this section, see 11.47.

### 13.5.1 admin-workstation

```
<%=scope.function_template "iptables/pieces/preamble"-%>
<%=scope.function_template "iptables/pieces/connected"-%>
<%=scope.function_template "iptables/pieces/loopback"-%>
<%=scope.function_template "iptables/pieces/dns"-%>
<%=scope.function_template "iptables/pieces/nfs-client"-%>
<%=scope.function_template "searde/iptables/pieces/nfs-client"-%>
<%=scope.function_template "iptables/pieces/site-highports"-%>
<%=scope.function_template "iptables/pieces/dhcp-client"-%>
<%=scope.function_template "iptables/pieces/ddns-client"-%>
<%=scope.function_template "searde/iptables/pieces/puppet-client"-%>
<%=scope.function_template "iptables/pieces/ssh-server"-%>
<%=scope.function_template "iptables/pieces/ssh-client"-%>
<%=scope.function_template "iptables/pieces/http-client"-%>
<%=scope.function_template "iptables/pieces/https-client"-%>
<%=scope.function_template "iptables/pieces/centrify-client"-%>
<%=scope.function_template "searde/iptables/pieces/mcafee-hbss-client"-%>
<%=scope.function_template "searde/iptables/pieces/kace-client"-%>
<%=scope.function_template "searde/iptables/pieces/https-sites"-%>
<%=scope.function_template "iptables/pieces/imaps-client"-%>
<%=scope.function_template "iptables/pieces/imap-client"-%>
<%=scope.function_template "iptables/pieces/xmpp-client"-%>
<%=scope.function_template "iptables/pieces/ntp-client"-%>
<%=scope.function_template "searde/iptables/pieces/satellite-client"-%>
<%=scope.function_template "searde/iptables/pieces/proxy-client"-%>
<%=scope.function_template "searde/iptables/pieces/license-server-client"-%[WRAP]
>
<%=scope.function_template "searde/iptables/pieces/jetdirect-client"-%>
<%=scope.function_template "iptables/pieces/input-icmp"-%>
<%=scope.function_template "iptables/pieces/output-icmp"-%>
<%=scope.function_template "iptables/pieces/output-smtp"-%>
<%=scope.function_template "iptables/pieces/source-routed"-%>
<%=scope.function_template "iptables/pieces/input-junk"-%>
<%=scope.function_template "iptables/pieces/mdns"-%>
<%=scope.function_template "iptables/pieces/fallthrough"-%>

COMMIT
```

### 13.5.2 audithost

```
<%=scope.function_template "iptables/pieces/preamble"-%>
<%=scope.function_template "iptables/pieces/loopback"-%>
<%=scope.function_template "iptables/pieces/connected"-%>
<%=scope.function_template "iptables/pieces/dns"-%>
<%=scope.function_template "iptables/pieces/puppet-client"-%>
<%=scope.function_template "eue/iptables/pieces/eglin-ntp"-%>
<%=scope.function_template "eue/iptables/pieces/eglin-afseo-filers"-%>
<%=scope.function_template "iptables/pieces/ssh-server"-%>
<%=scope.function_template "iptables/pieces/dhcp-client"-%>
<%=scope.function_template "iptables/pieces/input-icmp"-%>
<%=scope.function_template "iptables/pieces/output-icmp"-%>
```

```
<%=scope.function_template "iptables/pieces/source-routed"-%>
<%=scope.function_template "iptables/pieces/input-junk"-%>
<%=scope.function_template "iptables/pieces/ddns"-%>

<%=scope.function_template "iptables/pieces/audit-server"-%>
<%=scope.function_template "iptables/pieces/kerberos-client"-%>

<%=scope.function_template "iptables/pieces/fallthrough"-%>
COMMIT
```

### 13.5.3 katello-1.3-server

```
<% # variables needed:
   #     site: a CIDR block expressing the LAN this host will be on.
-%>
<%=scope.function_template "iptables/pieces/preamble"-%>
<%=scope.function_template "iptables/pieces/loopback"-%>
<%=scope.function_template "iptables/pieces/connected"-%>
<%=scope.function_template "iptables/pieces/dns"-%>
<%=scope.function_template "searde/iptables/pieces/puppet-client"-%>
<%=scope.function_template "iptables/pieces/ssh-server"-%>
<%=scope.function_template "iptables/pieces/ssh-client"-%>
<%=scope.function_template "iptables/pieces/katello-qpid"-%>
<%=scope.function_template "iptables/pieces/centrify-client"-%>
<%=scope.function_template "iptables/pieces/nfs-client"-%>
<%=scope.function_template "iptables/pieces/dhcp-client"-%>
<%=scope.function_template "iptables/pieces/ddns-client"-%>
<%=scope.function_template "iptables/pieces/ntp-client"-%>
<%=scope.function_template "iptables/pieces/satellite-client"-%>
<%=scope.function_template "iptables/pieces/input-icmp"-%>
<%=scope.function_template "iptables/pieces/output-icmp"-%>
<%=scope.function_template "iptables/pieces/source-routed"-%>
<%=scope.function_template "iptables/pieces/input-junk"-%>
<%=scope.function_template "iptables/pieces/mdns"-%>
<%=scope.function_template "iptables/pieces/fallthrough"-%>

COMMIT
```

### 13.5.4 loghost

```
<% # variables needed:
   #     site: a CIDR block expressing the LAN this host will be on.
-%>
<%=scope.function_template "iptables/pieces/preamble"-%>
<%=scope.function_template "iptables/pieces/loopback"-%>
<%=scope.function_template "iptables/pieces/connected"-%>
<%=scope.function_template "iptables/pieces/dns"-%>
<%=scope.function_template "iptables/pieces/puppet-client"-%>
# Talk to local web servers and proxies
-A OUTPUT -p tcp -m tcp -d <%=site-%> --dport 443 -j ACCEPT
-A OUTPUT -p tcp -m tcp -d <%=site-%> --dport 8080 -j ACCEPT
<%=scope.function_template "eue/iptables/pieces/eglin-local-https"-%>
<%=scope.function_template "eue/iptables/pieces/eglin-proxy"-%>
<%=scope.function_template "eue/iptables/pieces/eglin-ntp"-%>
<%=scope.function_template "eue/iptables/pieces/eglin-afseo-filers"-%>
<%=scope.function_template "iptables/pieces/ssh-server"-%>
```

```
# rsyslog
-A INPUT -p tcp -m tcp --dport 10514 -j ACCEPT
-A OUTPUT -p tcp -m tcp --sport 10514 -j ACCEPT

<%=scope.function_template "iptables/pieces/dhcp-client"-%>
<%=scope.function_template "iptables/pieces/input-icmp"-%>
<%=scope.function_template "iptables/pieces/output-icmp"-%>
<%=scope.function_template "iptables/pieces/source-routed"-%>
<%=scope.function_template "iptables/pieces/input-junk"-%>
<%=scope.function_template "iptables/pieces/ddns"-%>

<%=scope.function_template "iptables/pieces/fallthrough"-%>

COMMIT
```

### 13.5.5 pieces/audit-server

```
<% site_subnets.each do |subnet| %>
-A INPUT -s <%=subnet-%> -p tcp -m tcp --sport 48 --dport 48 -j ACCEPT
<% end %>
```

### 13.5.6 pieces/centrify-client

```
-A OUTPUT -m tcp -p tcp --dport 445 -j ACCEPT
# Not sure why conntrack wasn't working for this one.
-A INPUT -m tcp -p tcp --sport 3268 -j ACCEPT
-A OUTPUT -m tcp -p tcp --dport 3268 -j ACCEPT
-A OUTPUT -m tcp -p tcp --dport 389 -j ACCEPT
-A OUTPUT -m udp -p udp --dport 389 -j ACCEPT
<%= scope.function_template "iptables/pieces/kerberos-client"-%>
```

### 13.5.7 pieces/connected

```
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
```

### 13.5.8 pieces/ddns-client

```
-A OUTPUT -m tcp -p tcp --dport 53 -j ACCEPT
```

### 13.5.9 pieces/dhcp-client

```
# Allow DHCP requests to go out, responses in
-A INPUT -p udp -m udp --dport 68 -j ACCEPT
-A OUTPUT -p udp -m udp --dport 67 -j ACCEPT

# Drop responses being sent and requests being received: we are not a DHCP
# server
-A INPUT -p udp -m udp --dport 67 -j DROP
-A OUTPUT -p udp -m udp --dport 68 -j DROP
```

### 13.5.10 pieces/dns

```
# DNS client
-A INPUT -p udp -m udp --sport domain -j ACCEPT
-A OUTPUT -p udp -m udp --dport domain -j ACCEPT
```

### 13.5.11 pieces/dns-server

```
# DNS server
-A INPUT -p udp -m udp --dport domain -j ACCEPT
-A OUTPUT -p udp -m udp --sport domain -j ACCEPT
```

### 13.5.12 pieces/fallthrough

```
-A INPUT -j LOG --log-prefix "INPUT fallthrough: "
-A OUTPUT -j LOG --log-prefix "OUTPUT fallthrough: "
```

### 13.5.13 pieces/http-client

```
-A OUTPUT -m tcp -p tcp --dport 80 -j ACCEPT
```

### 13.5.14 pieces/https-client

```
-A OUTPUT -m tcp -p tcp --dport 443 -j ACCEPT
```

### 13.5.15 pieces/imap-client

```
-A OUTPUT -m tcp -p tcp --dport 143 -j ACCEPT
```

### 13.5.16 pieces/imaps-client

```
-A OUTPUT -m tcp -p tcp --dport 993 -j ACCEPT
```

### 13.5.17 pieces/input-icmp

```
# UNIX SRG GEN003602, GEN003604: reject ICMP timestamp requests. Since we'r[WRAP]
e
# dropping packets by default, what we do here is accept a bunch of ICMP
# messages that aren't timestamp requests.
#
# http://www.ciscopress.com/articles/article.asp?p=174313&seqNum=4 provides
# useful industry guidance for ICMP security.
#
# "! -f": Reject ICMP fragments. Legitimate ICMP messages are so short that
# they would never be fragmented.
#
# Enable pinging.
-A INPUT -p icmp -m icmp ! -f --icmp-type ping -s 127.0.0.1 -j ACCEPT
-A INPUT -p icmp -m icmp ! -f --icmp-type pong -s 127.0.0.1 -j ACCEPT
<% site_subnets.each do |subnet| %>
```

```
-A INPUT -p icmp -m icmp ! -f --icmp-type ping -s <%=subnet-%> -j ACCEPT
-A INPUT -p icmp -m icmp ! -f --icmp-type pong -s <%=subnet-%> -j ACCEPT
<% end %>
# This type has many codes. Code 4 (fragmentation needed but do-not-fragmen[WRAP]
t
# flag set) needed for path MTU discovery. "Interesting implications in IPs[WRAP]
ec."
-A INPUT -p icmp -m icmp ! -f --icmp-type destination-unreachable -s 127.0.[WRAP]
0.1 -j ACCEPT
<% site_subnets.each do |subnet| %>
-A INPUT -p icmp -m icmp ! -f --icmp-type destination-unreachable -s <%=sub[WRAP]
net-%> -j ACCEPT
<% end %>
# Enable traceroute.
-A INPUT -p icmp -m icmp ! -f --icmp-type time-exceeded -j ACCEPT
```

### 13.5.18 pieces/input-junk

```
# reject junk
-A INPUT -j JUNK
# packets we never care about
# Apple Remote Desktop
-A JUNK -p udp -m udp --dport 3283 -j DROP
# Building 350 windows hosts
-A JUNK -d <%=broadcast-%> -p udp -m udp --sport 137 --dport 137 -j DROP
-A JUNK -d 255.255.255.255 -p udp -m udp --sport 1036 -j DROP
-A JUNK -d <%=broadcast-%> -p udp -m udp --dport 1947 -j DROP
-A JUNK -d <%=broadcast-%> -p udp -m udp --dport 8083 -j DROP
# Broadcast NTP
-A JUNK -d 255.255.255.255 -p udp -m udp --dport 123 -j DROP
-A JUNK -d <%=broadcast-%> -p udp -m udp --dport 123 -j DROP
-A JUNK -p udp -m udp --sport 1038 -j DROP
-A JUNK -p udp -m udp --dport 8421 -j DROP
# Windows chatter
-A JUNK -p udp -m udp --dport 137 -j DROP
-A JUNK -p udp -m udp --dport 138 -j DROP
-A JUNK -p udp -m udp --dport 139 -j DROP
-A JUNK -p udp -m udp --dport 631 -j DROP
-A JUNK -p udp -m udp --dport 177 -j DROP
# Mac chatter
-A JUNK -p udp -m udp -d 255.255.255.255 --dport 111 -j DROP
# Multicast
-A JUNK -s <%=gateway-%> -d 224.0.0.1 -j DROP
# Multicast DNS (Avahi, Zeroconf)
-A JUNK -d 224.0.0.251 -p udp -m udp --dport 5353 -j DROP
-A JUNK -d <%=broadcast-%> -p udp -m udp --dport 5353 -j DROP
# Broadcast highports
-A JUNK -d 255.255.255.255 -p udp -m udp --dport 1024:65535 -j DROP
# Likely wake-on-lan packets. If we're awake enough to receive packets and
# filter them, we don't need to hear about them. They'll get as far as the
# Ethernet adapter anyway, whether we drop them here or not.
-A JUNK -d 255.255.255.255 -p udp -m udp --dport 9 -j DROP
-A JUNK -j RETURN
```

### 13.5.19 pieces/katello-qpid

```
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 443 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 5671 -j ACCEPT
```

### 13.5.20 pieces/kerberos-client

```
-A OUTPUT -m tcp -p tcp --dport 464 -j ACCEPT
-A OUTPUT -m tcp -p tcp --dport 88 -j ACCEPT
-A OUTPUT -m udp -p udp --dport 88 -j ACCEPT
```

### 13.5.21 pieces/local-http-client

```
<% site_subnets.each do |subnet| %>
-A OUTPUT -m tcp -p tcp -d <%=subnet-%> --dport 80 -j ACCEPT
<% end %>
```

### 13.5.22 pieces/local-https-client

```
<% site_subnets.each do |subnet| %>
-A OUTPUT -m tcp -p tcp -d <%=subnet-%> --dport 443 -j ACCEPT
<% end %>
```

### 13.5.23 pieces/loopback

```
-A INPUT -s 127.0.0.1 -d 127.0.0.1 -j ACCEPT
-A OUTPUT -s 127.0.0.1 -d 127.0.0.1 -j ACCEPT
```

### 13.5.24 pieces/mdns

```
# GEN007850: don't send dynamic DNS updates "unless needed." mDNS is not
# strictly the same, but its purpose is also to "transmit unencrypted
# information about a system including its name and address." As we don't
# presently need mDNS, we can just turn it off without questioning the sani[WRAP]
ty
# of such a dictum.
-A OUTPUT -d 224.0.0.251 -p udp -m udp --dport 5353 -j DROP
# Internet Group Management Protocol (IGMP, multicast)
-A OUTPUT -d 224.0.0.22 -j DROP
```

### 13.5.25 pieces/nat-preamble

```
*nat
:PREROUTING ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
```

### 13.5.26 pieces/nfs-client

```
-A OUTPUT -m udp -p udp --dport 2049 -j ACCEPT
-A OUTPUT -m tcp -p tcp --dport 2049 -j ACCEPT
-A OUTPUT -m tcp -p tcp --dport 635 -j ACCEPT
-A OUTPUT -m tcp -p tcp --dport 637 -j ACCEPT
-A OUTPUT -m udp -p udp --dport 111 -j ACCEPT
-A OUTPUT -m tcp -p tcp --dport 111 -j ACCEPT
-A INPUT -m udp -p udp --dport 111 -j ACCEPT
-A INPUT -m tcp -p tcp --dport 111 -j ACCEPT
```

### 13.5.27 pieces/nfs-server

```
-A INPUT -m tcp -p tcp --dport 2049 -j ACCEPT
-A INPUT -m udp -p udp --dport 111 -j ACCEPT
-A INPUT -m tcp -p tcp --dport 111 -j ACCEPT
```

### 13.5.28 pieces/ntp-client

```
-A OUTPUT -m udp -p udp --dport 123 -j ACCEPT
```

### 13.5.29 pieces/ntp-server

```
-A INPUT -m udp -p udp --dport 123
```

### 13.5.30 pieces/output-icmp

```
# UNIX SRG GEN003602: reject ICMP timestamp requests. Since we’re dropping
# packets by default, what we do here is accept a bunch of ICMP messages th[WRAP]
at
# aren’t timestamp requests.
#
# http://www.ciscopress.com/articles/article.asp?p=174313&seqNum=4 provides
# useful industry guidance for ICMP security.
#
# "! -f": Reject ICMP fragments. Legitimate ICMP messages are so short that
# they would never be fragmented.
#
# Enable pinging.
-A OUTPUT -p icmp -m icmp ! -f --icmp-type ping -d 127.0.0.1 -j ACCEPT
-A OUTPUT -p icmp -m icmp ! -f --icmp-type pong -d 127.0.0.1 -j ACCEPT
<% site_subnets.each do |subnet| %>
-A OUTPUT -p icmp -m icmp ! -f --icmp-type ping -d <%=subnet-%> -j ACCEPT
-A OUTPUT -p icmp -m icmp ! -f --icmp-type pong -d <%=subnet-%> -j ACCEPT
<% end %>
# This type has many codes. Code 4 (fragmentation needed but do-not-fragmen[WRAP]
t
# flag set) needed for path MTU discovery. "Interesting implications in IPs[WRAP]
ec."
-A OUTPUT -p icmp -m icmp ! -f --icmp-type destination-unreachable -d 127.0[WRAP]
.0.1 -j ACCEPT
<% site_subnets.each do |subnet| %>
-A OUTPUT -p icmp -m icmp ! -f --icmp-type destination-unreachable -d <%=su[WRAP]
bnet-%> -j ACCEPT
<% end %>
```

```
# Enable traceroute.
-A OUTPUT -p icmp -m icmp ! -f --icmp-type time-exceeded -j ACCEPT
```

### 13.5.31 pieces/output-smtp

```
-A OUTPUT -p tcp -m tcp --dport 25 -j ACCEPT
```

### 13.5.32 pieces/preamble

```
*filter
# UNIX SRG GEN008540: drop by default.
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
:JUNK - [0:0]
:SVN - [0:0]
```

### 13.5.33 pieces/puppet-client

```
# Puppet client. We should tighten this up, but it’s hard to know at this e[WRAP]
arly
# stage of Puppet deployment and use what our server will be.
-A OUTPUT -p tcp --dport 8140 -j ACCEPT
```

### 13.5.34 pieces/puppet-master

```
# Puppet master.
<% site_subnets.each do |subnet| %>
-A INPUT -s <%=subnet-%> -p tcp --dport 8140 -m state --state NEW -j ACCEPT
<% end %>
```

### 13.5.35 pieces/rhn-satellite-5.4-server

```
# Serve unencrypted HTTP to kickstarting systems
<% site_subnets.each do |subnet| %>
-A INPUT -s <%=subnet-%> -p tcp -m tcp --dport http -j ACCEPT
<% end %>
# Serve package updates and other RHN-based host management traffic over
# HTTPS
<% site_subnets.each do |subnet| %>
-A INPUT -s <%=subnet-%> -p tcp -m tcp --dport https -j ACCEPT
<% end %>
```

### 13.5.36 pieces/rsyslog-client

```
# Rsyslog client. We connect using SSL to the loghost and forward log messa[WRAP]
ges.
# (Use this piece only on hosts which include log::rsyslog::client.)
-A OUTPUT -p tcp -m tcp -d loghost --dport 10514 -j ACCEPT
```

### 13.5.37 pieces/satellite-client

```
-A OUTPUT -m tcp -p tcp --dport 443 -j ACCEPT
```

### 13.5.38 pieces/sbu-password-client

```
# We check out some things onto SBU test servers from the real SBU server, [WRAP]
port
# 4443. (We can’t use certificates because these are test servers; they don[WRAP]
’t
# have certificates yet.) Allow this.
-A OUTPUT -m tcp -p tcp --dport 4443 -j ACCEPT
```

### 13.5.39 pieces/site-highports

```
# Talk to hosts in my site on TCP and UDP high ports
<% @site_subnets.each do |subnet| %>
-A INPUT -s <%=subnet-%> -p tcp -m tcp --dport 1024:65535 -j ACCEPT
-A INPUT -s <%=subnet-%> -p udp -m udp --dport 1024:65535 -j ACCEPT
-A OUTPUT -d <%=subnet-%> -p tcp -m tcp --dport 1024:65535 -j ACCEPT
-A OUTPUT -d <%=subnet-%> -p udp -m udp --dport 1024:65535 -j ACCEPT
<% end %>
```

### 13.5.40 pieces/source-routed

```
# Removed: GEN003600, GEN003605, GEN003606: drop all source routed
# packets; input, output and forwarding. See previous versions of this
# file in Subversion.
```

### 13.5.41 pieces/ssh-client

```
-A OUTPUT -m tcp -p tcp --dport 22 -j ACCEPT
```

### 13.5.42 pieces/ssh-server

```
# Serve ssh
<% @site_subnets.each do |subnet| %>
-A INPUT -s <%=subnet-%> -p tcp -m tcp --dport ssh -j ACCEPT
<% end %>
-A OUTPUT -p tcp -m tcp --sport ssh -j ACCEPT
```

### 13.5.43 pieces/xmpp-client

```
-A OUTPUT -m tcp -p tcp --dport 5222 -j ACCEPT
```

### 13.5.44 puppetmaster

```
<% # variables needed:
   #     site: a CIDR block expressing the LAN this host will be on.
-%>
<%=scope.function_template "iptables/pieces/preamble"-%>
```

```
<%=scope.function_template "iptables/pieces/loopback"-%>
<%=scope.function_template "iptables/pieces/connected"-%>
<%=scope.function_template "iptables/pieces/dns"-%>
<%=scope.function_template "searde/iptables/pieces/puppet-client"-%>
<%=scope.function_template "iptables/pieces/ssh-server"-%>
<%=scope.function_template "iptables/pieces/ssh-client"-%>
<%=scope.function_template "iptables/pieces/puppet-master"-%>
<%=scope.function_template "iptables/pieces/centrify-client"-%>
<%=scope.function_template "iptables/pieces/nfs-client"-%>
<%=scope.function_template "iptables/pieces/dhcp-client"-%>
<%=scope.function_template "iptables/pieces/ddns-client"-%>
<%=scope.function_template "iptables/pieces/ntp-client"-%>
<%=scope.function_template "iptables/pieces/ntp-server"-%>
<%=scope.function_template "iptables/pieces/satellite-client"-%>
<%=scope.function_template "searde/iptables/pieces/kace-client"-%>
<%=scope.function_template "searde/iptables/pieces/smtp-client"-%>
<%=scope.function_template "iptables/pieces/input-icmp"-%>
<%=scope.function_template "iptables/pieces/output-icmp"-%>
<%=scope.function_template "iptables/pieces/source-routed"-%>
<%=scope.function_template "iptables/pieces/input-junk"-%>
<%=scope.function_template "iptables/pieces/mdns"-%>
<%=scope.function_template "iptables/pieces/fallthrough"-%>

COMMIT
```

### 13.5.45 rhn-satellite-5.4-server

```
<% # variables needed:
   #     site: a CIDR block expressing the LAN this host will be on.
-%>
<%=scope.function_template "iptables/pieces/preamble"-%>
<%=scope.function_template "iptables/pieces/loopback"-%>
<%=scope.function_template "iptables/pieces/connected"-%>
<%=scope.function_template "iptables/pieces/dns"-%>
<%=scope.function_template "searde/iptables/pieces/puppet-client"-%>
<%=scope.function_template "iptables/pieces/ssh-server"-%>
<%=scope.function_template "iptables/pieces/ssh-client"-%>
<%=scope.function_template "iptables/pieces/centrify-client"-%>
<%=scope.function_template "iptables/pieces/nfs-client"-%>
<%=scope.function_template "iptables/pieces/dhcp-client"-%>
<%=scope.function_template "iptables/pieces/ddns-client"-%>
<%=scope.function_template "iptables/pieces/ntp-client"-%>
<%=scope.function_template "iptables/pieces/rhn-satellite-5.4-server"-%>
# get updates from Red Hat via HTTPS
<%=scope.function_template "iptables/pieces/https-client"-%>
<%=scope.function_template "iptables/pieces/input-icmp"-%>
<%=scope.function_template "iptables/pieces/output-icmp"-%>
<%=scope.function_template "iptables/pieces/source-routed"-%>
<%=scope.function_template "iptables/pieces/input-junk"-%>
<%=scope.function_template "iptables/pieces/mdns"-%>
<%=scope.function_template "iptables/pieces/fallthrough"-%>

COMMIT
```

### 13.5.46 standalone

```
<%=scope.function_template(["iptables/pieces/preamble"])-%>
<%=scope.function_template(["iptables/pieces/loopback"])-%>
<%=scope.function_template(["iptables/pieces/connected"])-%>

<%=scope.function_template(["iptables/pieces/fallthrough"])-%>

COMMIT
```

### 13.5.47 vagrant

```
<%=scope.function_template(["iptables/pieces/preamble"])-%>
<%=scope.function_template(["iptables/pieces/loopback"])-%>
<%=scope.function_template(["iptables/pieces/connected"])-%>
<%=scope.function_template(["iptables/pieces/ssh-server"])-%>
<%=scope.function_template(["iptables/pieces/dns"])-%>
<%=scope.function_template(["iptables/pieces/http-client"])-%>
<%=scope.function_template(["iptables/pieces/https-client"])-%>
<%=scope.function_template(["iptables/pieces/site-highports"])-%>
<%=scope.function_template(["iptables/pieces/fallthrough"])-%>

COMMIT
```

### 13.5.48 workstation

```
<%=scope.function_template "iptables/pieces/preamble"-%>
<%=scope.function_template "iptables/pieces/connected"-%>
<%=scope.function_template "iptables/pieces/loopback"-%>
<%=scope.function_template "iptables/pieces/dns"-%>
<%=scope.function_template "iptables/pieces/nfs-client"-%>
<%=scope.function_template "searde/iptables/pieces/nfs-client"-%>
<%=scope.function_template "iptables/pieces/site-highports"-%>
<%=scope.function_template "iptables/pieces/dhcp-client"-%>
<%=scope.function_template "iptables/pieces/ddns-client"-%>
<%=scope.function_template "searde/iptables/pieces/puppet-client"-%>
<%=scope.function_template "iptables/pieces/ssh-server"-%>
<%=scope.function_template "iptables/pieces/ssh-client"-%>
<%=scope.function_template "iptables/pieces/http-client"-%>
<%=scope.function_template "iptables/pieces/https-client"-%>
<%=scope.function_template "iptables/pieces/centrify-client"-%>
<%=scope.function_template "searde/iptables/pieces/mcafee-hbss-client"-%>
<%=scope.function_template "searde/iptables/pieces/kace-client"-%>
<%=scope.function_template "searde/iptables/pieces/https-sites"-%>
<%=scope.function_template "searde/iptables/pieces/taz-client"-%>
<%=scope.function_template "searde/iptables/pieces/ocsp-http-client"-%>
<%=scope.function_template "iptables/pieces/imap-client"-%>
<%=scope.function_template "iptables/pieces/imaps-client"-%>
<%=scope.function_template "iptables/pieces/xmpp-client"-%>
<%=scope.function_template "iptables/pieces/ntp-client"-%>
<%=scope.function_template "searde/iptables/pieces/satellite-client"-%>
<%=scope.function_template "searde/iptables/pieces/proxy-client"-%>
<%=scope.function_template "searde/iptables/pieces/license-server-client"-%[WRAP]
>
<%=scope.function_template "searde/iptables/pieces/jetdirect-client"-%>
<%=scope.function_template "iptables/pieces/input-icmp"-%>
<%=scope.function_template "iptables/pieces/output-icmp"-%>
```

```
<%=scope.function_template "iptables/pieces/output-smtp"-%>
<%=scope.function_template "iptables/pieces/source-routed"-%>
<%=scope.function_template "iptables/pieces/input-junk"-%>
<%=scope.function_template "iptables/pieces/mdns"-%>
<%=scope.function_template "iptables/pieces/fallthrough"-%>

COMMIT
```

## 13.6 log/

For the policy that requires files in this section, see 11.55.1.

### 13.6.1 rsyslog/00common-global.conf

```
$ModLoad imuxsock.so # provides support for local system logging (e.g. via [WRAP]
logger command)
$ModLoad imklog.so # provides kernel logging support (previously done by rk[WRAP]
logd)
#$ModLoad immark.so # provides --MARK-- message capability
```

### 13.6.2 rsyslog/10gnutls-global.conf

```
$DefaultNetstreamDriver gtls
$DefaultNetstreamDriverCAFile   /etc/pki/rsyslog/ca.crt
$DefaultNetstreamDriverCertFile /etc/pki/rsyslog/<%= scope.lookupvar('::hos[WRAP]
tname') -%>.crt
$DefaultNetstreamDriverKeyFile  /etc/pki/rsyslog/private/<%= scope.lookupva[WRAP]
r('::hostname') -%>.key
```

### 13.6.3 rsyslog/50local.conf

```
# Use default timestamp format
$ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat

# File syncing capability is disabled by default. This feature is usually n[WRAP]
ot required,
# not useful and an extreme performance hit
#$ActionFileEnableSync on


#### RULES ####

# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                                 /dev/console

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none                /var/log/messages

# The authpriv file has restricted access.
authpriv.*                                              /var/log/secure

# Log all the mail messages in one place.
mail.*                                                  -/var/log/maillog


# Log cron stuff
cron.*                                                  /var/log/cron

# Everybody gets emergency messages
*.emerg                                                 *
```

```
# Save news errors of level crit and higher in a special file.
uucp,news.crit                                          /var/log/spooler

# Save boot messages also to boot.log
local7.*                                                /var/log/boot.log
```

### 13.6.4 rsyslog/client-only/80send-to-loghost.conf

```
$ActionSendStreamDriverAuthMode x509/name
$ActionSendStreamDriverPermittedPeer <%= loghost %>
$ActionSendStreamDriverMode 1

# ### begin forwarding rule ###
# The statement between the begin ... end define a SINGLE forwarding
# rule. They belong together, do NOT split them. If you create multiple
# forwarding rules, duplicate the whole block!
# Remote Logging (we use TCP for reliable delivery)
#
# An on-disk queue is created for this action. If the remote host is
# down, messages are spooled to disk and sent when it is up again.
$WorkDirectory /var/spool/rsyslog # where to place spool files
$ActionQueueFileName fwdRule1 # unique name prefix for spool files
$ActionQueueMaxDiskSpace 1g   # 1gb space limit (use as much as possible)
$ActionQueueSaveOnShutdown on # save messages to disk on shutdown
$ActionQueueType LinkedList   # run asynchronously
$ActionResumeRetryCount -1    # infinite retries if host is down
# remote host is: name/ip:port, e.g. 192.168.0.1:514, port optional
*.* @@<%= loghost -%>:10514
# ### end of the forwarding rule ###
```

### 13.6.5 rsyslog/loghost-only/20loghost.conf

```
# Provides TCP syslog reception
$ModLoad imtcp.so
$InputTCPServerStreamDriverMode 1
$InputTCPServerStreamDriverAuthMode x509/name
$InputTCPServerStreamDriverPermittedPeer *.<%=domain %>
$InputTCPServerRun 10514
```

## 13.7 nvidia/

For the policy that requires files in this section, see 11.71.

### 13.7.1 nvidia-rebuild.sh.erb

```
#!/bin/bash
#
# chkconfig: - 65 25
# description: nvidia-rebuild rebuilds the nVidia drivers when necessary. O[WRAP]
n
#              hosts with no nVidia card, it safely does nothing.
#
# If $INSTALLER_DIR is on an NFS mount, NFS mounts must happen before this
# script.

. /etc/rc.d/init.d/functions

INSTALLER_DIR=<%= installer_dir %>

# This setting may be superseded if the host contains a legacy chipset; see
# below
driver_installer=latest-‘uname -m‘

reconnoiter () {
    eval $(facter -p \
        has_nvidia_graphics_card \
        has_nvidia_legacy_304_graphics_card \
        has_nvidia_legacy_17314_graphics_card \
        using_nouveau_driver \
        nvidia_ko_exists \
        nvidia_libGL_installed \
        nvidia_glx_extension_installed \
        kernelrelease \
| sed ’s/’\’’/’\’’\’\’’’\’’/g; s/ => \(.*\)/=’\’’\1’\’’/’ )
# replace          ’ w. ’ \  ’   ’   ; replace => stuff with =’stuff’
#
# This serves to (2) enclose every fact value in single quotes; and
# (1) escape single quotes found in fact values. Single quotes in bash
# are escaping-free: a backslash inside a single-quoted string means
# just a backslash. So to have a single-quoted string with a single
# quote inside it, you must end the single quoted string, put no
# space, put a backslash-escaped single quote outside any quoting, put
# no space, and start another single-quoted string. So for example if
# we have the string a’b and we want to put it in single quotes, we
# say ’a’\’’b’. The reason to be so careful with single quotes is to
# avoid shell command injection.
}

start () {
    echo -n "NVIDIA proprietary driver: "
    reconnoiter
    if [ "$has_nvidia_graphics_card" = "true" ]; then
        if [ "$using_nouveau_driver" = "true" ]; then
            echo -n "Nouveau driver precludes"
            failure "NVIDIA proprietary driver"
        else
            install=no
            # reasons to reinstall are in ascending order of how fundamenta[WRAP]
l
            # they are; message is overwritten by more important reasons
            if [ "$nvidia_glx_extension_installed" != "true" ]; then
                install=yes
                message="GLX extension looks wrong"
            fi
            if [ "$nvidia_libGL_installed" != "true" ]; then
                install=yes
                message="NVIDIA proprietary libGL not installed"
            fi
            if [ "$nvidia_ko_exists" != "true" ]; then
                install=yes
                message="nvidia.ko not found for kernel $kernel_release"
            fi
            if [ "$has_nvidia_legacy_304_graphics_card" = "true" ]; then
                driver_installer=legacy-304-‘uname -m‘
            fi
            if [ "$has_nvidia_legacy_17314_graphics_card" = "true" ]; then
                driver_installer=legacy-17314-‘uname -m‘
            fi

            if [ "$install" = "yes" ]; then
                # this function does its own success/failure calls
                reinstall_driver "$message"
            else
                echo -n "looks good"
                success "NVIDIA proprietary driver"
            fi
        fi
    else
        echo -n "No card, or no facts known"
        # It’s not an intrinsic failure to not have an NVIDIA card installe[WRAP]
d
        success "NVIDIA proprietary driver"
    fi
}

reinstall_driver () {
    message="$1"
    qualified_driver_installer="$INSTALLER_DIR/$driver_installer"
    if [ -f "$qualified_driver_installer" ]; then
        echo -n "needs reinstall: $message"
        cat <<EOF


**************************************************************************

$0: Reinstalling nVidia driver. This will take ~15 min.

**************************************************************************
EOF
        if sh "$qualified_driver_installer" -Ns; then
            echo "$0: Driver installer done. nvidia-xconfig, perhaps?" >&2
            echo -n "NVIDIA proprietary driver: installed"
            success "NVIDIA proprietary driver"
        else
            echo -n "NVIDIA proprietary driver: install failed"
            failure "NVIDIA proprietary driver"
        fi
    else
        echo "$0: Installer \"$qualified_driver_installer\" not found." >&2
        failure "NVIDIA proprietary driver"
        return 1
    fi
}

# See how we were called.

case "$1" in
  start)
start
RETVAL=$?
        ;;
  stop)
RETVAL=0
        ;;
  status)
RETVAL=0
;;
  restart|reload)
start
RETVAL=$?
;;
  condrestart)
RETVAL=0
;;
  *)
        echo $"Usage: $0 {start|stop|restart|condrestart|status}"
        exit 1
esac

echo
exit $RETVAL
```

## 13.8 rpm/

For the policy that requires files in this section, see 11.84.5.

### 13.8.1 rpm-signatures.cron.erb

```
#!/bin/sh
# Warn about any unsigned packages installed on the system. These are
# discernible because their %{sigpgp:pgpsig} is (none). gpg-pubkey packages[WRAP]
,
# being themselves public keys, are normally not signed; this is no cause f[WRAP]
or
# concern, so filter them out. If there are no unsigned packages, there is [WRAP]
no
# output.
rpm -qa --qf "Unsigned package found installed: \
%{name} %{version}-%{release}.%{arch}, \
signature %{sigpgp:pgpsig}, buildhost %{buildhost}\\n" | \
    grep '(none)' | \
<% @known_unsigned_packages.each do |pkg| %>    grep -v '<%=pkg-%>' | \
<% end -%>
    grep -v ': gpg-pubkey '
```

# 13.9 sbu/

For the policy that requires files in this section, see 11.88.4.

## 13.9.1 sbu.conf

```
##
## SSL Virtual Host Context
##

<VirtualHost _default_:443>

DocumentRoot "/var/www/html"
ServerName <%=@fqdn%>
ServerAdmin <%=@server_admin_email%>

Include common/nss-site-common.conf
Include common/nss-site-cac.conf

#   The nickname of the RSA server certificate you are going to use.
NSSNickname <%=@cert_nickname-%>
<% if @mode != ’production’ %>
NSSEnforceValidCerts off
<% end %>

#   The NSS security database directory that holds the certs and keys
NSSCertificateDatabase /etc/pki/mod_nss



#########################
# Authentication defaults
#########################
<Location />
<IfModule mod_auth_pgsql.c>
Auth_PG_database auth
Auth_PG_user sbu_mod_auth_pgsql
Auth_PG_pwd_table cert_users
Auth_PG_uid_field user_name
Auth_PG_pwd_field user_passwd
Auth_PG_grp_table cert_groups
Auth_PG_grp_user_field  user_name
Auth_PG_grp_group_field group_name
Auth_PG_hash_type MD5

# No real passwords are stored in the database: the views
# provide ’password’ as the password, as required by
# FakeBasicAuth
Auth_PG_encrypted off
#Auth_PG_log_table log
#Auth_PG_log_uname_field uname
#Auth_PG_log_date_field date
#Auth_PG_log_uri_field uri
#Auth_PG_log_pwd_field password

Auth_PG_Authoritative on
```

```
</IfModule>

AuthType Basic
        # Anyone who sees a username/password prompt has already been rejec[WRAP]
ted.
        # Try to funnel them to the fine 401 page that's been written.
        AuthName ">>> ACCESS DENIED; click cancel for help <<<"
</Location>

<Directory /var/www/html>
Require valid-user
        # Do not show auto-indexes where index.html does not exist.
        Options -Indexes
</Directory>

<Location "/favicon.png">
    Satisfy Any
</Location>
<Location "/favicon.ico">
    Satisfy Any
</Location>
<Location "/robots.txt">
    Satisfy Any
</Location>

# Some people may have ancient bookmarks for the signup page.
Redirect permanent /cert/WelcomePage/welcome.htm https://<%=web_fqdn-%>/


##############################
# When authentication fails...
##############################
ErrorDocument 401 /pages/401.html
# Let unauthenticated users actually get that file
<Location /pages/401.html>
Satisfy Any
</Location>

<Files ~ "\.(cgi|shtml|phtml|php3?)$">
    NSSOptions +StdEnvVars
</Files>

ScriptAlias /cgi-bin/ /var/www/cgi-bin/
<Directory "/var/www/cgi-bin">
    SetEnv PYTHON_EGG_CACHE "/tmp"
    NSSOptions +StdEnvVars +FakeBasicAuth
</Directory>

CustomLog logs/ssl_request_log \
          "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"

CustomLog logs/ssl_activity_log \
  "%{%s}t:%>s:%u"

#########
#########
#########                        web applications
```

```
#########
#########

Alias /authapp/ /var/www/sbu-apps/authapp/public/
Alias /request/ /var/www/sbu-apps/authapp/public/go.py/request/
Alias /upload/ /var/www/sbu-apps/upload/public/
Alias /authapp-static/ /var/www/sbu-apps/authapp/static/
Alias /upload-static/ /var/www/sbu-apps/upload/static/

<Directory /var/www/sbu-apps/*/public>
NSSOptions +StdEnvVars

# stock mod_python uses python 2.3, which we can’t anymore..
# SetHandler mod_python
# PythonHandler quixote.server.mod_python_handler
# PythonOption quixote-publisher-factory go.create_publisher
# PythonDebug On
# PythonPath "sys.path + [’/var/www/apps’]"
# PythonEnablePdb on
Options +ExecCGI
AddHandler cgi-script .py
        SetEnv PYTHON_EGG_CACHE "/tmp"
Order allow,deny
Allow from all
</Directory>
<Directory /var/www/sbu-apps/*/static>
SetHandler None
Order allow,deny
Allow from all
</Directory>

<Location /authapp>
Require valid-user
SetEnv PYTHONPATH "/var/www/sbu-apps/authapp"
Order allow,deny
Allow from all
</Location>
<Location /authapp/go.py/agree>
# This message will only be shown if a username and password box is
# shown; and that will only happen if the user’s certificate DN is not
# found in the cert_users_needing_to_agree table. This in turn is
# either because the user has agreed to the present AUP (no further
# need to agree at this time), or because the user is disabled,
# expired, or otherwise unable to log in for a non-AUP-related problem.
#
# Unfortunately, that whole message may not be shown by the browser in
# the username and password dialog box. So we settle for something more
# terse.
AuthName "AUP agreement page access denied. Talk to <%=@server_admin_email[WRAP]
-%>."
<IfModule mod_auth_pgsql.c>
Auth_PG_database auth
Auth_PG_user sbu_mod_auth_pgsql
#                 vvvvvvvvvvvvvvvvvvvvvvvvvvv
Auth_PG_pwd_table cert_users_needing_to_agree
#                 ^^^^^^^^^^^^^^^^^^^^^^^^^^^
Auth_PG_uid_field user_name
```

```
Auth_PG_pwd_field user_passwd
Auth_PG_grp_table cert_groups
Auth_PG_grp_user_field  user_name
Auth_PG_grp_group_field group_name
Auth_PG_hash_type MD5

# No real passwords are stored in the database: the views
# provide 'password' as the password, as required by
# FakeBasicAuth
Auth_PG_encrypted off
#Auth_PG_log_table log
#Auth_PG_log_uname_field uname
#Auth_PG_log_date_field date
#Auth_PG_log_uri_field uri
#Auth_PG_log_pwd_field password

Auth_PG_Authoritative on
</IfModule>

        Require valid-user
</Location>


<Location /request>
    SetEnv PYTHONPATH "/var/www/sbu-apps/authapp"
    # Let anyone in: to connect they must have provided a certificate; if t[WRAP]
hey
    # are using /request, we are not yet familiar with that certificate.
    Satisfy any
    Order allow,deny
    Allow from all
</Location>
<Location /authapp-static>
Order allow,deny
Allow from all
</Location>

<Directory /var/www/sbu-apps/authapp/public>
        Require valid-user
        SetEnv PYTHONPATH "/var/www/sbu-apps/authapp"
Order allow,deny
Allow from all
</Directory>

<Directory /var/www/sbu-apps/upload/public>
Require valid-user
        SetEnv PYTHONPATH "/var/www/sbu-apps/upload"
Order allow,deny
Allow from all
</Directory>



#######
#######
#######
#######     Miscellaneous permissions
```

```
#######
#######
#######


# Disallow access to .svn dirs in the main website.
<DirectoryMatch "^/var/www/html.*\.svn">
Order deny,allow
Deny from all
</DirectoryMatch>





#######
#######
#######
#######                  SBU per-directory permissions
#######
#######
#######

<Directory /var/www/html>
<IfModule mod_auth_pgsql.c>
Auth_PG_database auth
Auth_PG_user sbu_mod_auth_pgsql
Auth_PG_pwd_table cert_users
Auth_PG_uid_field user_name
Auth_PG_pwd_field user_passwd
Auth_PG_grp_table cert_groups
Auth_PG_grp_user_field  user_name
Auth_PG_grp_group_field group_name
Auth_PG_hash_type MD5

#Auth_PG_log_table log
#Auth_PG_log_uname_field uname
#Auth_PG_log_date_field date
#Auth_PG_log_uri_field uri
#Auth_PG_log_pwd_field password

# mod_auth_pgsql must be consulted first (after SSL
# verification, anyway); it falls through to other modules by
# being non-authoritative
Auth_PG_Authoritative on
</IfModule>
</Directory>

#######
#######
#######
#######     Trac
#######
#######
#######

######################
```

```
# Subversion via https
######################

# There wasn’t a <Python> :(
<Perl>
#!/usr/bin/perl

#####
##### Create a <Location> directive for each Subversion repository
##### named, for example, foo, that limits access to the svn-foo
##### group
#####

# Directory where svn repositories are, in the filesystem
my $svn_dir       = "/var/www/svn";
# Location under which they will appear, at the end of the URL
my $svn_location  = "/svn";

opendir(SVN_ROOT, $svn_dir) or die "Couldn’t open Subversion root directory[WRAP]
 ($svn_dir)";

while (my $name = readdir(SVN_ROOT)) {
# entirely alphanumeric? (i.e. not . or ..)
if($name =~ /^[[:alnum:]_]+$/) {
# Create a <Location> directive
$Location{"$svn_location/$name"} = {
# This is what goes in the <Location> directive
AuthType => "Basic",
Require => "group svn-readonly-$name svn-$name",
# http://svnbook.red-bean.com/en/1.0/ch06s04.html#svn-ch-6-sect-4.4.1
LimitExcept => {
    "GET PROPFIND OPTIONS REPORT" => {
        Require  => "group svn-$name",
    },
},
DAV      => "svn",
SVNPath  => "$svn_dir/$name",
                        # apply XSLT style that adds classification bar
                        SVNIndexXSLT => "/styles/svnindex.xsl",
                        # allow Web Folder writes to be commits
                        SVNAutoversioning => "On"
};
}
}

closedir(SVN_ROOT);
__END__
</Perl>

<Location /svn>
Options -Indexes
        # Let users do other HTTP verbs in this location, contravening the
        # global default in ../conf/httpd.conf
        <LimitExcept GET POST OPTIONS>
            Allow from all
        </LimitExcept>
</Location>
```

```
###############
# Trac
###############

# static things like pictures and CSS
Alias /trac/ /var/www/trac-shared/htdocs/common/
<Directory /var/www/trac-shared/>
# the trac htdocs are not a secret.
Satisfy any
Options -Indexes +MultiViews
AllowOverride None
Order allow,deny
Allow from all
</Directory>
<Directory /var/www/wsgi-bin>
Order allow,deny
Allow from all
</Directory>



<Perl>
#!/usr/bin/perl

## Create a <Location> directive for each Trac site
## named, for example, foo, that limits access to the trac-foo
## group

# Heavily adapted from
# http://projects.edgewall.com/trac/wiki/TracMultipleProjects?version=69
# Directory where trac configurations are, in the filesystem
my $trac_dir       = "/var/www/tracs";
# Location under which the projects will appear, at the end of the URL
my $trac_location  = "/projects";

opendir(TRAC_ROOT, $trac_dir) or die "Couldn't open Trac root directory ($t[WRAP]
rac_dir)";

while (my $name = readdir(TRAC_ROOT)) {
# entirely alphanumeric? (i.e. not . or ..)
if($name =~ /^[[:alnum:]_]+$/) {
push @WSGIScriptAlias,
["$trac_location/$name",
"/var/www/wsgi-bin/trac.wsgi"];
# Create a <Location> directive
$Location{"$trac_location/$name"} = {
# This is what goes in the <Location> directive
AuthType => "Basic",
# require group svn-$name. same as the svn repos
Require  => "group trac-$name",
SetEnv => ["trac.env_path", "/var/www/tracs/$name"],
# http://code.google.com/p/modwsgi/wiki/IntegrationWithTrac
# look in page for this string: 'the case of hosting
# multiple sites'
WSGIApplicationGroup => "%{GLOBAL}"
};
}
}
closedir(TRAC_ROOT);
__END__
</Perl>



#######
#######
#######
#######     Static pages
#######
#######
#######





<Directory /var/www/html/pages>
AuthType Basic
Require valid-user
</Directory>

<Directory /var/www/html/styles>
Satisfy any
</Directory>
<Directory /var/www/html/images>
Satisfy any
</Directory>
<Directory /var/www/html/scripts>
Satisfy any
</Directory>

<Directory /var/www/html/Data>
AuthType Basic
Require group admins
        # Show auto-indexes
        Options +Indexes
        # We don’t want uploaders hijacking a dir by uploading index.html.
        # But there doesn’t seem to be a way to have no DirectoryIndex at a[WRAP]
ll.
        # So we’ll just set it to something obscure.
DirectoryIndex c0c751fb-200b-4b74-bbc1-b64431ca256741c68bf1-bbd7-4536-84b0[WRAP]
-0f96246db932b6a3c593-2d8f-43c8-a9e6-fa85680512a828a0ecb0-4202-4201-813d-3d[WRAP]
8540d469e6
        HeaderName /pages/files_header.html
        # Make no files special in Data
        # especially, execute nothing!
Options -ExecCGI
# do NOT execute Incoming PHP pages
        <IfModule mod_php4.c>
            php_flag engine off
        </IfModule>
</Directory>


Include conf.d/Data.perms

</VirtualHost>
```

## 13.10 searde_svn/

For the policy that requires files in this section, see 11.91.4.

```
##
## SSL Virtual Host Context
##

<VirtualHost _default_:443>

DocumentRoot "/var/www/html"
ServerName <%=@fqdn%>
ServerAdmin <%=@server_admin_email%>

Include common/nss-site-common.conf
Include common/nss-site-cac.conf

#   The nickname of the RSA server certificate you are going to use.
NSSNickname <%=@cert_nickname-%>
<% if @mode != 'production' %>
NSSEnforceValidCerts off
<% end %>

#   The NSS security database directory that holds the certs and keys
NSSCertificateDatabase /etc/pki/mod_nss



#########################
# Authentication defaults
#########################
<Location />
<IfModule mod_auth_pgsql.c>
Auth_PG_database auth
Auth_PG_user sbu_mod_auth_pgsql
Auth_PG_pwd_table cert_users
Auth_PG_uid_field user_name
Auth_PG_pwd_field user_passwd
Auth_PG_grp_table cert_groups
Auth_PG_grp_user_field  user_name
Auth_PG_grp_group_field group_name
Auth_PG_hash_type MD5

# No real passwords are stored in the database: the views
# provide 'password' as the password, as required by
# FakeBasicAuth
Auth_PG_encrypted off
#Auth_PG_log_table log
#Auth_PG_log_uname_field uname
#Auth_PG_log_date_field date
#Auth_PG_log_uri_field uri
#Auth_PG_log_pwd_field password

Auth_PG_Authoritative on
</IfModule>

AuthType Basic
        # Anyone who sees a username/password prompt has already been rejec[WRAP]
```

```
ted.
        # Try to funnel them to the fine 401 page that's been written.
        AuthName ">>> ACCESS DENIED; click cancel for help <<<"
</Location>

<Directory /var/www/html>
Require valid-user
        # Do not show auto-indexes where index.html does not exist.
        Options -Indexes
</Directory>

<Location "/favicon.png">
    Satisfy Any
</Location>
<Location "/favicon.ico">
    Satisfy Any
</Location>
<Location "/robots.txt">
    Satisfy Any
</Location>

# Some people may have ancient bookmarks for the signup page.
Redirect permanent /cert/WelcomePage/welcome.htm https://<%=web_fqdn-%>/


##############################
# When authentication fails...
##############################
ErrorDocument 401 /pages/401.html
# Let unauthenticated users actually get that file
<Location /pages/401.html>
Satisfy Any
</Location>

<Files ~ "\.(cgi|shtml|phtml|php3?)$">
    NSSOptions +StdEnvVars
</Files>

ScriptAlias /cgi-bin/ /var/www/cgi-bin/
<Directory "/var/www/cgi-bin">
    SetEnv PYTHON_EGG_CACHE "/tmp"
    NSSOptions +StdEnvVars +FakeBasicAuth
</Directory>

CustomLog logs/ssl_request_log \
          "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"

CustomLog logs/ssl_activity_log \
  "%{%s}t:%>s:%u"

#########
#########
#########                         web applications
#########
#########

Alias /authapp/ /var/www/sbu-apps/authapp/public/
Alias /request/ /var/www/sbu-apps/authapp/public/go.py/request/
Alias /upload/ /var/www/sbu-apps/upload/public/
Alias /authapp-static/ /var/www/sbu-apps/authapp/static/
Alias /upload-static/ /var/www/sbu-apps/upload/static/

<Directory /var/www/sbu-apps/*/public>
NSSOptions +StdEnvVars

# stock mod_python uses python 2.3, which we can’t anymore..
# SetHandler mod_python
# PythonHandler quixote.server.mod_python_handler
# PythonOption quixote-publisher-factory go.create_publisher
# PythonDebug On
# PythonPath "sys.path + [’/var/www/apps’]"
# PythonEnablePdb on
Options +ExecCGI
AddHandler cgi-script .py
        SetEnv PYTHON_EGG_CACHE "/tmp"
Order allow,deny
Allow from all
</Directory>
<Directory /var/www/sbu-apps/*/static>
SetHandler None
Order allow,deny
Allow from all
</Directory>

<Location /authapp>
Require valid-user
SetEnv PYTHONPATH "/var/www/sbu-apps/authapp"
Order allow,deny
Allow from all
</Location>
<Location /authapp/go.py/agree>
# This message will only be shown if a username and password box is
# shown; and that will only happen if the user’s certificate DN is not
# found in the cert_users_needing_to_agree table. This in turn is
# either because the user has agreed to the present AUP (no further
# need to agree at this time), or because the user is disabled,
# expired, or otherwise unable to log in for a non-AUP-related problem.
#
# Unfortunately, that whole message may not be shown by the browser in
# the username and password dialog box. So we settle for something more
# terse.
AuthName "AUP agreement page access denied. Talk to <%=@server_admin_email[WRAP]
-%>."
<IfModule mod_auth_pgsql.c>
Auth_PG_database auth
Auth_PG_user sbu_mod_auth_pgsql
#                 vvvvvvvvvvvvvvvvvvvvvvvvvvv
Auth_PG_pwd_table cert_users_needing_to_agree
#                 ^^^^^^^^^^^^^^^^^^^^^^^^^^^
Auth_PG_uid_field user_name
Auth_PG_pwd_field user_passwd
Auth_PG_grp_table cert_groups
Auth_PG_grp_user_field  user_name
Auth_PG_grp_group_field group_name
```

```
Auth_PG_hash_type MD5

# No real passwords are stored in the database: the views
# provide 'password' as the password, as required by
# FakeBasicAuth
Auth_PG_encrypted off
#Auth_PG_log_table log
#Auth_PG_log_uname_field uname
#Auth_PG_log_date_field date
#Auth_PG_log_uri_field uri
#Auth_PG_log_pwd_field password

Auth_PG_Authoritative on
</IfModule>

        Require valid-user
</Location>


<Location /request>
    SetEnv PYTHONPATH "/var/www/sbu-apps/authapp"
    # Let anyone in: to connect they must have provided a certificate; if t[WRAP]
hey
    # are using /request, we are not yet familiar with that certificate.
    Satisfy any
    Order allow,deny
    Allow from all
</Location>
<Location /authapp-static>
Order allow,deny
Allow from all
</Location>

<Directory /var/www/sbu-apps/authapp/public>
        Require valid-user
        SetEnv PYTHONPATH "/var/www/sbu-apps/authapp"
Order allow,deny
Allow from all
</Directory>

<Directory /var/www/sbu-apps/upload/public>
Require valid-user
        SetEnv PYTHONPATH "/var/www/sbu-apps/upload"
Order allow,deny
Allow from all
</Directory>



#######
#######
#######
#######     Miscellaneous permissions
#######
#######
#######
```

```
# Disallow access to .svn dirs in the main website.
<DirectoryMatch "^/var/www/html.*\.svn">
Order deny,allow
Deny from all
</DirectoryMatch>





#######
#######
#######
#######                  SBU per-directory permissions
#######
#######
#######

<Directory /var/www/html>
<IfModule mod_auth_pgsql.c>
Auth_PG_database auth
Auth_PG_user sbu_mod_auth_pgsql
Auth_PG_pwd_table cert_users
Auth_PG_uid_field user_name
Auth_PG_pwd_field user_passwd
Auth_PG_grp_table cert_groups
Auth_PG_grp_user_field  user_name
Auth_PG_grp_group_field group_name
Auth_PG_hash_type MD5

#Auth_PG_log_table log
#Auth_PG_log_uname_field uname
#Auth_PG_log_date_field date
#Auth_PG_log_uri_field uri
#Auth_PG_log_pwd_field password

# mod_auth_pgsql must be consulted first (after SSL
# verification, anyway); it falls through to other modules by
# being non-authoritative
Auth_PG_Authoritative on
</IfModule>
</Directory>

#######
#######
#######
#######     Trac
#######
#######
#######

######################
# Subversion via https
######################

# There wasn't a <Python> :(
<Perl>
#!/usr/bin/perl

#####
##### Create a <Location> directive for each Subversion repository
##### named, for example, foo, that limits access to the svn-foo
##### group
#####

# Directory where svn repositories are, in the filesystem
my $svn_dir       = "/var/www/svn";
# Location under which they will appear, at the end of the URL
my $svn_location  = "/svn";

opendir(SVN_ROOT, $svn_dir) or die "Couldn’t open Subversion root directory[WRAP]
 ($svn_dir)";

while (my $name = readdir(SVN_ROOT)) {
# entirely alphanumeric? (i.e. not . or ..)
if($name =~ /^[[:alnum:]_]+$/) {
# Create a <Location> directive
$Location{"$svn_location/$name"} = {
# This is what goes in the <Location> directive
AuthType => "Basic",
Require => "group svn-readonly-$name svn-$name",
# http://svnbook.red-bean.com/en/1.0/ch06s04.html#svn-ch-6-sect-4.4.1
LimitExcept => {
    "GET PROPFIND OPTIONS REPORT" => {
        Require  => "group svn-$name",
    },
},
DAV      => "svn",
SVNPath  => "$svn_dir/$name",
                        # apply XSLT style that adds classification bar
                        SVNIndexXSLT => "/styles/svnindex.xsl",
                        # allow Web Folder writes to be commits
                        SVNAutoversioning => "On"
};
}
}

closedir(SVN_ROOT);
__END__
</Perl>

<Location /svn>
Options -Indexes
        # Let users do other HTTP verbs in this location, contravening the
        # global default in ../conf/httpd.conf
        <LimitExcept GET POST OPTIONS>
            Allow from all
        </LimitExcept>
</Location>


###############
```

```
# Trac
###############

# static things like pictures and CSS
Alias /trac/ /var/www/trac-shared/htdocs/common/
<Directory /var/www/trac-shared/>
# the trac htdocs are not a secret.
Satisfy any
Options -Indexes +MultiViews
AllowOverride None
Order allow,deny
Allow from all
</Directory>
<Directory /var/www/wsgi-bin>
Order allow,deny
Allow from all
</Directory>



<Perl>
#!/usr/bin/perl

## Create a <Location> directive for each Trac site
## named, for example, foo, that limits access to the trac-foo
## group

# Heavily adapted from
# http://projects.edgewall.com/trac/wiki/TracMultipleProjects?version=69
# Directory where trac configurations are, in the filesystem
my $trac_dir       = "/var/www/tracs";
# Location under which the projects will appear, at the end of the URL
my $trac_location  = "/projects";

opendir(TRAC_ROOT, $trac_dir) or die "Couldn’t open Trac root directory ($t[WRAP]
rac_dir)";

while (my $name = readdir(TRAC_ROOT)) {
# entirely alphanumeric? (i.e. not . or ..)
if($name =˜ /ˆ[[:alnum:]_]+$/) {
push @WSGIScriptAlias,
["$trac_location/$name",
"/var/www/wsgi-bin/trac.wsgi"];
# Create a <Location> directive
$Location{"$trac_location/$name"} = {
# This is what goes in the <Location> directive
AuthType => "Basic",
# require group svn-$name. same as the svn repos
Require  => "group trac-$name",
SetEnv => ["trac.env_path", "/var/www/tracs/$name"],
# http://code.google.com/p/modwsgi/wiki/IntegrationWithTrac
# look in page for this string: ’the case of hosting
# multiple sites’
WSGIApplicationGroup => "%{GLOBAL}"
};
}
}
```

```
closedir(TRAC_ROOT);
__END__
</Perl>



#######
#######
#######
#######     Static pages
#######
#######
#######






<Directory /var/www/html/pages>
AuthType Basic
Require valid-user
</Directory>

<Directory /var/www/html/styles>
Satisfy any
</Directory>
<Directory /var/www/html/images>
Satisfy any
</Directory>
<Directory /var/www/html/scripts>
Satisfy any
</Directory>

<Directory /var/www/html/Data>
AuthType Basic
Require group admins
        # Show auto-indexes
        Options +Indexes
        # We don’t want uploaders hijacking a dir by uploading index.html.
        # But there doesn’t seem to be a way to have no DirectoryIndex at a[WRAP]
ll.
        # So we’ll just set it to something obscure.
DirectoryIndex c0c751fb-200b-4b74-bbc1-b64431ca256741c68bf1-bbd7-4536-84b0[WRAP]
-0f96246db932b6a3c593-2d8f-43c8-a9e6-fa85680512a828a0ecb0-4202-4201-813d-3d[WRAP]
8540d469e6
        HeaderName /pages/files_header.html
        # Make no files special in Data
        # especially, execute nothing!
Options -ExecCGI
# do NOT execute Incoming PHP pages
        <IfModule mod_php4.c>
            php_flag engine off
        </IfModule>
</Directory>
```

```
Include conf.d/Data.perms

</VirtualHost>
```

## 13.11 sudo/

For the policy that requires files in this section, see 11.104.4.

### 13.11.1 auditable/rule.erb

```
<%=@user_spec%> ALL=(<%=@run_as%>) \
    <%=@modifiers%>NOEXEC:        AUDITABLE_NOEXEC, \
    <%=@modifiers%>EXEC:          AUDITABLE_EXEC,   \
    <%=@modifiers%>SETENV:NOEXEC: AUDITABLE_SETENV_NOEXEC, \
    <%=@modifiers%>SETENV:EXEC:   AUDITABLE_SETENV_EXEC
```

### 13.11.2 auditable/whole.erb

```
<% [’noexec’, ’exec’, ’setenv_noexec’, ’setenv_exec’].each do |t|
   items = []
   items += (@data[t] || []).sort.uniq
   items += (@data[’DISALLOW_’+t] || []).sort.uniq.map {|x| ’!’+x}
   if items.any?
%>
Cmnd_Alias AUDITABLE_<%=t.upcase%> = \
    <%=items.join(", \\\n    ")%>
<% end; end %>
```

### 13.11.3 unlimited.erb

```
<%=user_spec%> ALL=(<%=run_as%>) NOPASSWD:ALL
```

## 13.12 usb/

For the policy that requires files in this section, see 11.111.1.

### 13.12.1 mass_storage/group-udisks.pkla

```
[allow disk actions for group]
Identity=unix-group:<%= group %>
Action=org.freedesktop.udisks.filesystem-mount;org.freedesktop.udisks.drive[WRAP]
-eject;org.freedesktop.udisks.drive-detach
ResultAny=yes
ResultActive=yes
ResultInactive=auth_admin
```

# Chapter 14

# External Requirements

This chapter discusses requirements passed on to other components of the network, which are not configured by this policy.

## 14.1 DHCP services

The DHCP server(s) must render to their clients DHCP options that result in the configuration of two or more DNS servers. GEN001375 M6

# Chapter 15

# Bibliography

[1] [ms-sntp]: Network time protocol (ntp) authentication extensions. 2011. http://msdn.microsoft.com/en-us/library/cc246877%28v=PROT.13%29.aspx.

[2] Defense Information Services Agency [DISA]. *Sharing Peripherals Across the Network Security Technical Implementation Guide, Version 1, Release 1*. DISA, 2005. http://iase.disa.mil/stigs/stig/span-stig-v1r1.pdf.

[3] Defense Information Services Agency [DISA]. *Generic Database STIG, Version 8, Release 1*. DISA, 2007. http://iase.disa.mil/stigs/downloads/zip/database-stig-v8r1.zip.

[4] Defense Information Services Agency [DISA]. *Apache Server 2.2 for Unix Draft STIG, Version 0, Release 0.2*. DISA, 2011. http://iase.disa.mil/stigs/downloads/zip/u_apache_2.2_unix_v0r2_idraft_stig.zip.

[5] Defense Information Services Agency [DISA]. *Apache Site 2.2 for Unix Draft STIG, Version 0, Release 0.2*. DISA, 2011. http://iase.disa.mil/stigs/downloads/zip/u_apache_2.2_unix_v0r2_idraft_stig.zip.

[6] Defense Information Services Agency [DISA]. *UNIX System Requirements Guide, Version 1, Release 2*. DISA, 2012. http://iase.disa.mil/stigs/os/unix/u_unix_os_policy-v1r2_srg_manual.zip.

[7] The PostgreSQL Global Development Group. *PostgreSQL 8.4.x Documentation*. 2009. Available as part of the postgresql RHEL package.

[8] B. Haberman and D. Mills. Rfc 5906: Network time protocol version 4: Autokey specification. 2010. http://www.ietf.org/rfc/rfc5906.txt.

[9] Puppet Labs. *Generated references*. http://docs.puppetlabs.com/references/2.7.6/.

[10] Daniel Macpherson, Lana Brindley, and Athene Chan. *Red Hat Network Satellite 5.4 Installation Guide.* Red Hat, Inc., 2010. http://docs.redhat.com/docs/en-US/Red_Hat_Network_Satellite/5.4/html/Installation_Guide/index.html.

[11] D. Mills, J. Martin, J. Burbank, and W. Kasch. Rfc 5905: Network time protocol version 4. 2010. http://www.ietf.org/rfc/rfc5905.txt.

[12] David L. Mills. Rfc 1305: Network time protocol (version 3). 1992. http://www.ietf.org/rfc/rfc1305.txt.

[13] Department of Defense CIO. *Department of Defense Instruction 8500.2, Information Assurance Implementation.* DTIC, 2005. http://www.dtic.mil/whs/directives/corres/pdf/850002p.pdf.

[14] National Institute of Standards and Technology (NIST). *Federal Information Processing Standard 140-2.* National Institute of Standards and Technology (NIST), 2001. http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf.

[15] Douglas Silas, Martin Prpic, et al. *Red Hat Enterprise Linux 6 Deployment Guide.* Red Hat, Inc., 2010. http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Deployment_Guide/index.html.

[16] Sun Microsystems Inc. and Red Hat, Inc. *NSS Cryptographic Module Version 3.12 FIPS 140-2 Non-Proprietary Security Policy, Version 1.21.* 2009. http://www.mozilla.org/projects/security/pki/nss/fips/secpolicy.pdf.

[17] James Turnbull and Jeffrey McCune. *Pro Puppet.* Apress, 2011. http://www.apress.com/9781430230571.

# Chapter 16

# Indices

# Compliance

## Apache 2.2 STIG

## Application Security & Development STIG

## Database STIG

## IA Controls

## Apple OS X 10.8 STIG

## RHEL 5 STIG

## RHEL 6 STIG

## SPAN STIG

## UNIX SRG

# Classes

# Defined Resource Types

# Files

# General Index