Apple Inc.

Apple macOS 13 Ventura Common Criteria Configuration Guide

Version 1.1

January 12, 2024

NIAP VID 11347

Prepared by:

atsec information security corporation

4516 Seton Center Parkway, Suite 250

Austin, TX 78759

www.atsec.com

Revision History

Version	Date	Changes
1.0	2023-11-22	First version
1.1	2024-01-12	Include Keychain service API functions.

Trademarks

Apple's trademarks applicable to this document are listed in
https://www.apple.com/legal/intellectual-property/trademark/appletmlist.html

Other company, product, and service names may be trademarks or service marks of others.

Contents

Figures

1 Introduction

This guide provides instructions to configure and operate Apple macOS 13 Ventura in the Common Criteria evaluated configuration.

1.1 Target of Evaluation

The Target of Evaluation (TOE) is the Apple macOS 13 Ventura general purpose operating system (GPOS). macOS is tightly integrated with hardware and runs on Apple iMac, MacBook Air, MacBook Pro, Mac mini, and Mac Pro computers. macOS is a Unix-based graphical operating system. The macOS core is a POSIX compliant operating system built on top of the XNU kernel with standard Unix facilities available from the command line interface.

Table 1 – TOE Identification

Category	Identifier
VID	11347
TOE Identifier	Apple macOS 13 Ventura
TOE Version	13.2.1
TOE Developer	Apple Inc.
Key Words	Operating System, GPOS

macOS does not have a physical boundary, because macOS is the operating system. As evaluated, macOS runs on the following hardware platforms:

Table 2 – Hardware Platforms

Marketing Name	Model	Model Identifier	Processor	Micro Architecture	Security Chip	BT version	BT Chip
2023							
MacBook Pro (16-inch, 2023)	A2780	Mac14,6	M2 Max	ARMv8.6-A	SEP v2.0	5.3	4388
		Mac14,10	M2 Pro	ARMv8.6-A	SEP v2.0	5.3	4388
MacBook Pro (14-inch, 2023)	A2779	Mac14,5	M2 Max	ARMv8.6-A	SEP v2.0	5.3	4388
		Mac14,9	M2 Pro	ARMv8.6-A	SEP v2.0	5.3	4388
Mac mini (M2 Pro, 2023)	A2816	Mac14,12	M2 Pro	ARMv8.6-A	SEP v2.0	5.3	4388
Mac mini (M2, 2023)	A2686	Mac14,3	M2	ARMv8.6-A	SEP v2.0	5.3	4388
2022							

Marketing Name	Model	Model Identifier	Processor	Micro Architecture	Security Chip	BT version	BT Chip
MacBook Pro (13-inch, M2, 2022)	A2338	Mac14,7	M2	ARMv8.6-A	SEP v2.0	5.0	4378
MacBook Air (M2, 2022)	A2861	Mac14,2	M2	ARMv8.6-A	SEP v2.0	5.0	4387
Mac Studio	A2615	Mac13,2	M1 Ultra	ARMv8.5-A	SEP v2.0	5.0	4387
	A2615	Mac13,1	M1 Max	ARMv8.5-A	SEP v2.0	5.0	4387
2021							
MacBook Pro (16-inch, 2021)	A2485	MacBookPro18,2	M1 Max	ARMv8.5-A	SEP v2.0	5.0	4387
		MacBookPro18,1	M1 Pro	ARMv8.5-A	SEP v2.0	5.0	4387
MacBook Pro (14-inch, 2021)	A2442	MacBookPro18,4	M1 Max	ARMv8.5-A	SEP v2.0	5.0	4387
		MacBookPro18,3	M1 Pro	ARMv8.5-A	SEP v2.0	5.0	4387
iMac (24-inch, M1, 2021)	A2438	iMac21,1	M1	ARMv8.5-A	SEP v2.0	5.0	4378
	A2439	iMac21,2	M1	ARMv8.5-A	SEP v2.0	5.0	4378
2020							
Mac mini (M1, 2020)	A2348	Macmini9,1	M1	ARMv8.5-A	SEP v2.0	5.0	4378
MacBook Air (M1, 2020)	A2337	MacBookAir10,1	M1	ARMv8.5-A	SEP v2.0	5.0	4378
MacBook Pro (13-inch, M1, 2020)	A2338	MacBookPro17,1	M1	ARMv8.5-A	SEP v2.0	5.0	4364
MacBook Air (Retina, 13-inch, 2020)	A2179	MacBookAir9,1	Core i5-1030NG7 Core i7-1060NG7	Ice Lake	T2	5.0	4377

Marketing Name	Model	Model Identifier	Processor	Micro Architecture	Security Chip	BT version	BT Chip
MacBook Pro (13-inch, 2020, Four Thunderbolt 3 ports)	A2251	MacBookPro16,2	Core i5-1038NG7 Core i7-1068NG7	Ice Lake	T2	5.0	4377
MacBook Pro (13-inch, 2020, Two Thunderbolt 3 ports)	A2289	MacBookPro16,3	Core i5-8257U Core i7-8557U	Coffee Lake	T2	5.0	4377
iMac (Retina 5K, 27-inch, 2020)	A2115	iMac20,1 iMac20,2	Core i5-10500 Core i5-10600 Core i7-10700K Core i9-10910	Comet Lake	T2	5.0	4364
2019							
MacBook Air (Retina, 13-inch, 2019)	A1932	MacBookAir8,2	Core i5-8210Y	Amber Lake	T2	4.2	4355
MacBook Pro (13-inch, 2019, Four Thunderbolt 3 ports)	A1989	MacBookPro15,2	Core i5-8279U Core i7-8569U	Coffee Lake	T2	5.0	4364
MacBook Pro (13-inch, 2019, Two Thunderbolt 3 ports)	A2159	MacBookPro15,4	Core i5-8257U Core i7-8557U	Coffee Lake	T2	5.0	4377
MacBook Pro (15-inch, 2019)	A1990	MacBookPro15,1 MacBookPro15,3	Core i7-9750H Core i9-9880H	Coffee Lake	T2	5.0	4364

Marketing Name	Model	Model Identifier	Processor	Micro Architecture	Security Chip	BT version	BT Chip
MacBook Pro (16-inch, 2019)	A2141	MacBookPro16,1 MacBookPro16,4	Core i7-9750H Core i9-9880H	Coffee Lake	T2	5.0	4377
Mac Pro (2019)	A1991	MacPro7,1	Xeon W-3223 Xeon W-3235 Xeon W-3245 Xeon W-3265M Xeon W-3275M	Cascade Lake	T2	5.0	4364
Mac Pro (2019 Rack)	A2304	MacPro7,1	Xeon W-3223 Xeon W-3235 Xeon W-3245 Xeon W-3265M Xeon W-3275M	Cascade Lake	T2	5.0	4364
2018							
MacBook Air (Retina, 13-inch, 2018)	A1932	MacBookAir8,1	Core i5-8210Y	Amber Lake	T2	4.2	4355
Mac mini (2018)	A1993	Macmini8,1	Core i5-8500B Core i7-8700B	Coffee Lake	T2	5.0	4364
MacBook Pro (15-inch, 2018)	A1990	MacBookPro15,1 MacBookPro15,3	Core i7-8750H Core i7-8850H Core i9-8950HK	Coffee Lake	T2	5.0	4364

Marketing Name	Model	Model Identifier	Processor	Micro Architecture	Security Chip	BT version	BT Chip
MacBook Pro (13-inch, 2018, Four Thunderbolt 3 ports)	A1989	MacBookPro15,2	Core i5-8259U Core i7-8559U	Coffee Lake	T2	5.0	4364
2017							
iMac Pro (2017)	A1862	iMacPro1,1	Xeon W-2140B Xeon W-2150B Xeon W-2170B Xeon W-2190B	Skylake	T2	5.0	4364

1.2 Document Purpose and Scope

This Common Criteria guidance document contains configuration information needed to configure and administer Apple macOS 13 Ventura. Apple macOS 13 Ventura conforms to the Protection Profile for General Purpose Operating Systems Version 4.2.1 (OS_PP_v4.2.1). The information contained in this document is intended for Administrators who would be responsible for configuration and management.

This guide will show the administrator how to install and operate macOS in a Common Criteria compliant manner.

1.3 Excluded Functionality

The CC evaluation does not cover certain product features including the following:

- Bonjour – Bonjour is Apple's standards-based, zero configuration network protocol that lets devices find services on a network. This feature is outside the scope of the evaluation.
- VPN Split Tunnel – VPN split tunnel is not included in the evaluation and must be disabled.
- Siri – Siri supports some commands related to configuration settings. This feature is not included in the evaluation and must be disabled.

1.4 Assumptions

The administrators and users must ensure the hardware computing platform is trustworthy and has not been tampered with.

The administrators must ensure that authorized human users of the OS are not willfully negligent or hostile. The users must use the software in compliance with the applied enterprise security policy.

The administrators must not be careless, willfully negligent, or hostile. They must administer the OS within compliance of the enterprise security policy.

1.5 How to Invoke Common macOS Features

If you are new to macOS 13 Ventura, there are a couple of interface terms that are important to know before using this document.

" menu" refers to the pull-down menu in the top left corner of the menu bar.

"System Settings" is an app that provides access to and control of settings in macOS. It is a selection under the menu.

To help illustrate hierarchical menu structures, this document uses '>' to indicate menu levels. For example, to invoke the System Settings app, this document will say:

> Navigating to menu > System Settings

Thus, you would click on the menu and select the System Settings menu item.

"Dock" refers to the bar, normally positioned at the bottom of the screen, that contains multiple icons. As you move your cursor across each icon in the Dock, the icon's descriptive name will appear above the icon.

System Settings icon:

By default, the Dock contains the System Settings icon shown above. Clicking this icon provides an alternate method to launch the System Settings app.

Launchpad icon:

Another icon in the Dock is the Launchpad icon shown above. Launchpad provides the user with a screen containing one or more pages of icons, each representing an app or a folder. Clicking on an app will launch the app. One of the Launchpad pages will contain the "Other" folder. When you click on this folder, Launchpad will display additional apps contained within this folder.

Terminal icon:

One of the apps in the Other folder is the "Terminal" app. The Terminal app provides a command line shell program where the user can type in commands. When this document refers to command line commands like the chmod command, it implies that the user must execute the Terminal app and type the command into the window created by the app.

1.6 Obtaining a Mac

To obtain a Mac listed in Table 2, follow the directions in the subsections below for the distribution channel that best fits your situation.

1.6.1 Normal distribution channel

The normal distribution channels for obtaining these devices include the following:

- The Apple Store (either a physical store or online at https://apple.com)
- Apple retailers
- Resellers

1.6.2 Business-specific distribution channel

There is a distinct online store for Business customers with a link from the "Apple Store" to Apple and Business: (https://www.apple.com/business/). Additionally, the following link to "Shop for Business" is provided (https://www.apple.com/retail/business/).

1.6.3 Government-specific distribution channel

Government customers can use the link: https://www.apple.com/r/store/government/

1.6.4 Additional

Large customers can have their own Apple Store Catalog for their employees to purchase devices directly from Apple under their corporate employee purchase program.

1.7 Cryptographic engine warning

Only the Apple corecrypto cryptographic modules were included in the evaluation and testing of this product. Other cryptographic modules were not evaluated or tested.

2 Installation, and Update, and Recovery

Apple macOS 13 Ventura comes pre-installed on some of the hardware platforms listed in Table 2. macOS has a built-in update feature that the user can leverage to check for and install updates. Should the need arise, the administrator can manually download, re-install, the same version or newer version of the TOE on the supporting hardware.

To determine the running macOS version, click the menu > About This Mac.

Figure 1 – macOS Version

2.1 Installation

If the hardware platform was purchased prior to the release of macOS 13 Ventura, macOS 13 Ventura can be installed:

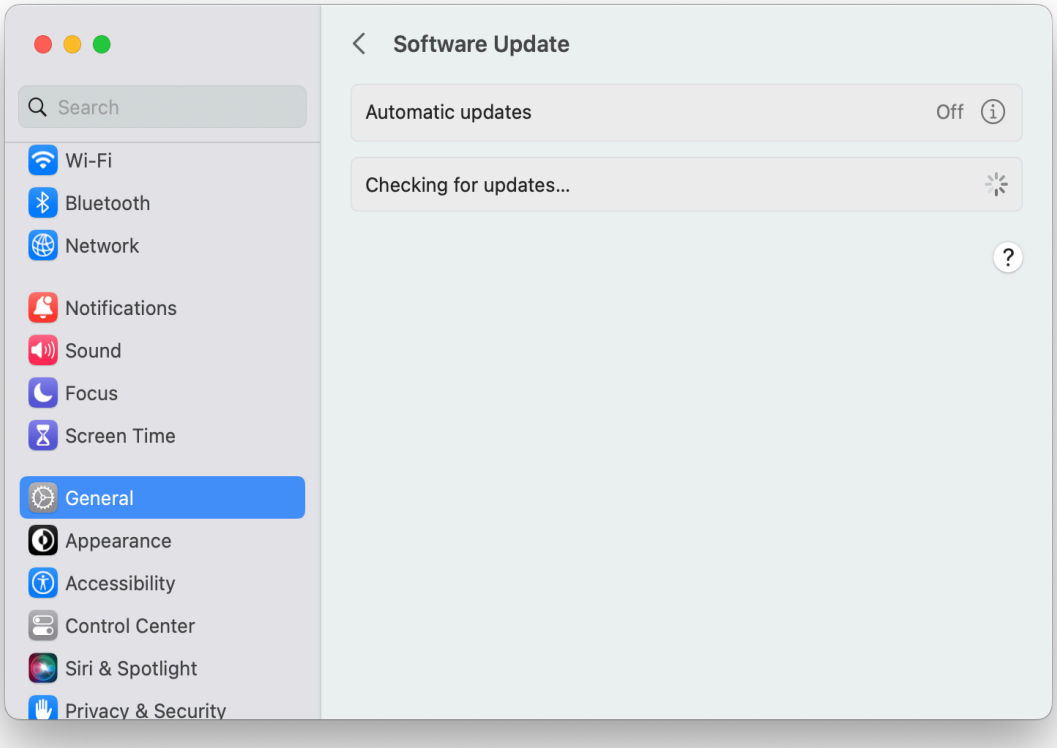
1. Open the App Store

2. Search for “macOS Ventura”
3. Click Get
4. The installer will be downloaded to your Applications folder
5. Run the installer and follow the onscreen instructions

2.2 Update

Any user can check for updates to macOS by navigating to menu > System Settings > General > Software Update or opening the System Settings app, selecting General > Software Update. If updates are available, the user will be given the option of installing updates.

Figure 2 – Software Updates



Users can also check for updates from the command line by issuing the `softwareupdate -l` command. The user must be an authorized user (i.e., successfully logged in) to initiate an update.

2.2.1 Rapid Security Response (RSR)

Starting with macOS 13.2 Ventura, macOS supports Apple’s Rapid Security Response feature. This feature allows Apple to provide security fixes to users more frequently. RSR can be enabled/disabled in the Software Update dialog. See section 3.10 for information on how to enable/disable automatic RSR updates.

2.3 Recovery

If macOS encounters a problem (e.g., a system integrity error), it will boot into the paired macOS Recovery. macOS Recovery will prompt the user for an administrators' credentials and attempt to repair the problem. If macOS Recovery can repair the problem, macOS will boot normally.

If the repairs are unsuccessful, the administrator can reinstall macOS:

1. Boot into macOS Recovery
    a. On a Mac with Apple silicon, press and hold the power button on your Mac until you see "Loading startup options."
    b. On an Intel-based Mac, press the power button. Press and hold *Command-R* until you see the startup screen
2. Make sure you're connected to the Internet
3. Click Reinstall macOS Ventura, then click Continue
4. Follow the onscreen instructions

2.4 Installation/Verification Process

The above sections describe how to obtain macOS updates and installation images. macOS will automatically prompt you to initiate the installation process once the image has been downloaded.

Signature verification of the TOE image is performed when the system reboots. As the kernel boots, each stage of the boot process validates the digital signatures of the next stage using the public key that is embedded in the Mac's Boot ROM during manufacturing. Any verification failures will cause the Mac to not boot. The Mac will provide a black screen with a question mark or place you into the Recovery mode. If you cannot fix the computer from Recovery, you will need to contact Apple Support.

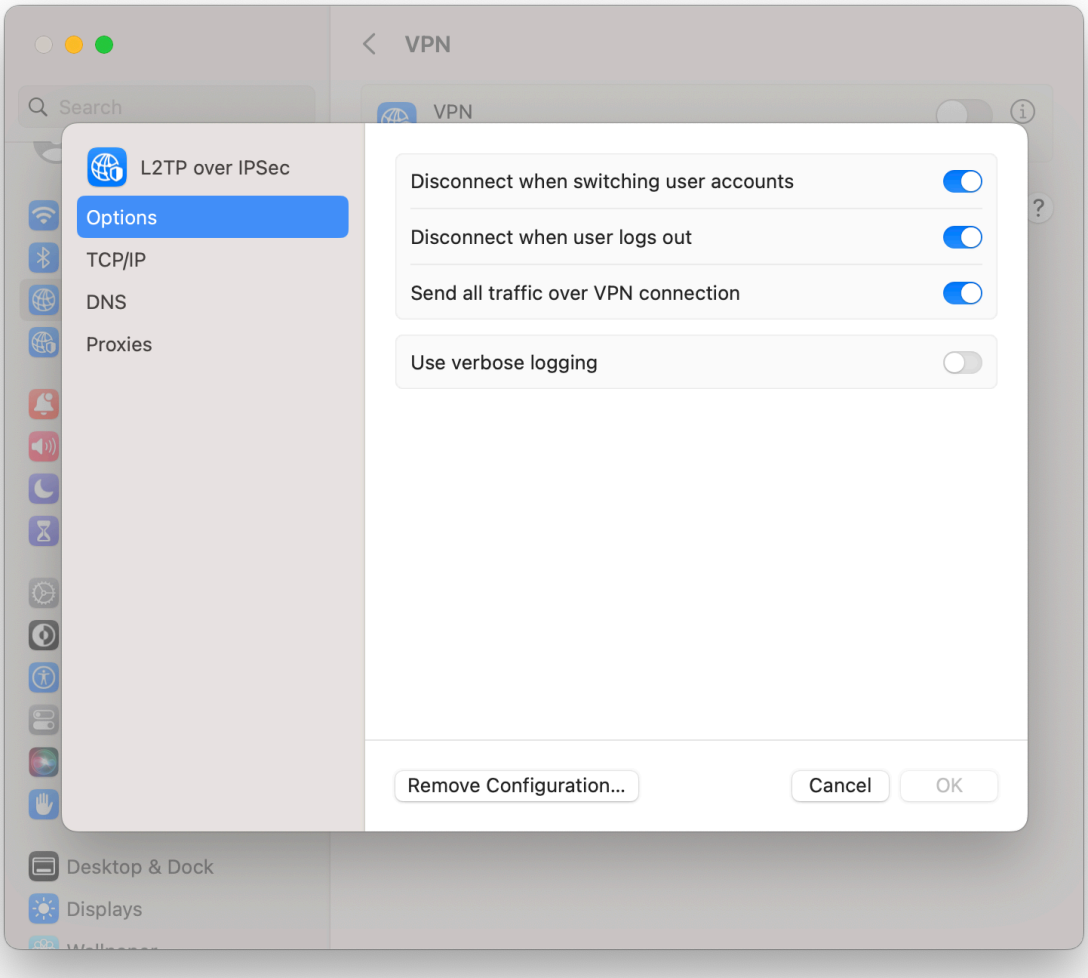
3 Configuration and Management

3.1 Excluded Functionality

Protection of traffic using a VPN is outside the scope of this evaluation. If a VPN is configured, VPN split tunnel must be disabled. As an administrator:

1. Open the System Settings app
2. Select Network > <VPN-Identifier>
3. Click the information icon (the letter i in a circle) on the righthand side of the selected VPN
4. Select Options on the leftside of the dialog that appears
5. Enable "Send all traffic over VPN connection"
6. Click OK

Figure 3 - VPN Options

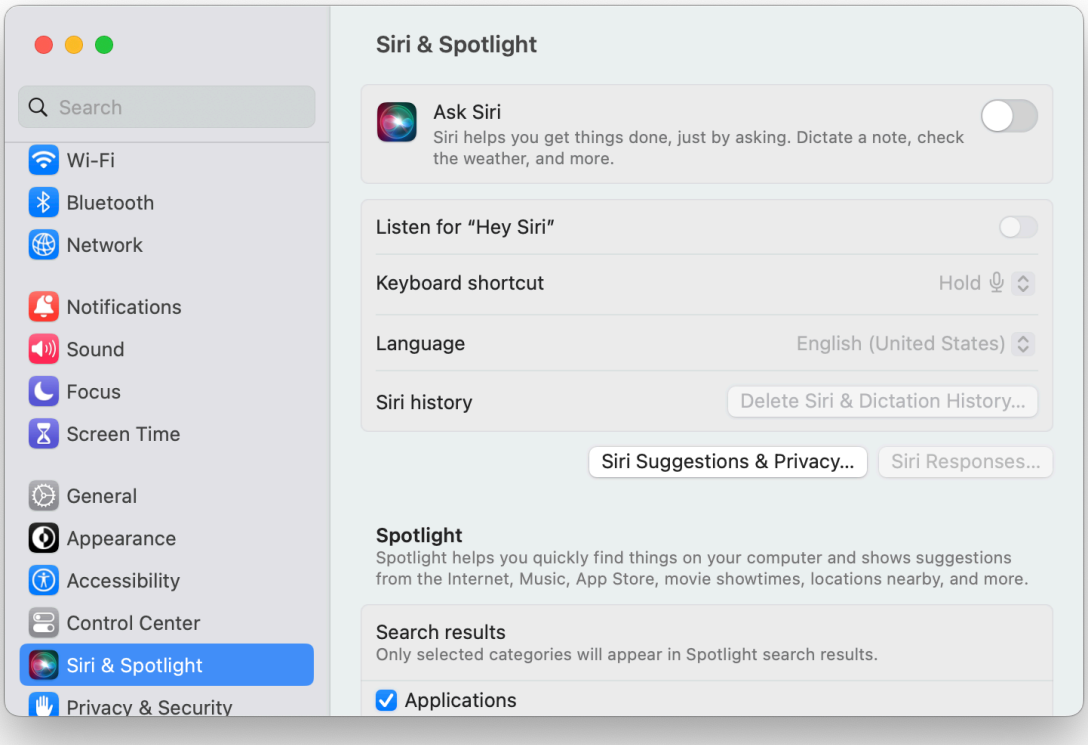


Siri must be disabled since it sends voice commands and queries to remote Apple servers for processing. As an administrator:

1. Open the System Settings app

2. Select Siri
3. Uncheck "Enable ask Siri"

Figure 4 – Disable Siri



3.2 User Accounts

macOS supports two roles, Standard and Administrator. The first account created is assigned the administrator role.

3.2.1 Add Users

As an administrator:

1. Open the System Settings app
2. Select Users & Groups
3. Click Add Account...
4. Authenticate as an administrator when prompted
5. Enter the details for the new user:
    a. New Account – This specifies the role for the user. The user can be assigned the Administrator role by selecting Administrator from the drop down list
    b. Full Name – This is a display
    c. Account name – This will be used as the name in audit logs and for the user's home directory

   d. Password – This is the user's initial password. Passwords may be up to 255 characters in length and composed of printable ASCII characters (i.e., character codes 0x20 to 0x7E inclusive)
6. Click Create User

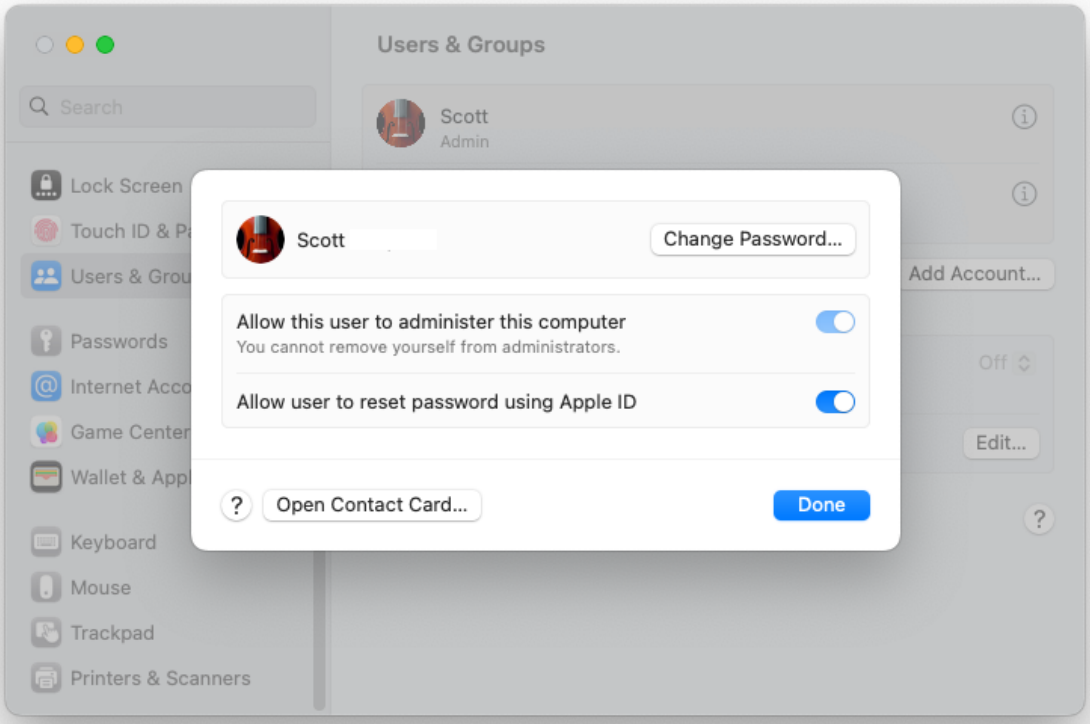
3.2.2 Delete Users

As an administrator:
1. Open the System Settings app
2. Select Users & Groups
3. Click the information icon (the letter i in a circle) on the righthand side of the user to be deleted
4. Click "Delete Account…"
5. Authenticate as an administrator when prompted
6. On the "Are you sure …" dialog:
   a. Choose what to do with the user's home folder
   b. Click Delete Account

3.2.3 Configure Roles

Administrator privileges can be added or removed from existing users. As an administrator:
1. Open the System Settings app
2. Select Users & Groups
3. Find the existing user and click the information icon (the letter i in a circle) of the user
4. Select "Allow user to administer this computer" to assign the user the Administrator role. Unselect "Allow user to administer this computer" to assign the user the Standard user role
5. Authenticate as an administrator when prompted

Figure 5 – Administrator Role



3.2.4 Smart Card Authentication

As the user that will be paired with the smart card:

1. Connect the smart card reader to the Mac
2. Insert the smart card
3. A new window will appear to pair the user with the smart card
4. Select "Certificate For PIV Authentication (<Name>)" from the Card Identity drop down list
5. Click Pair
6. Authenticate as the user to approve the pairing
7. Enter the PIN for the smart card
8. Enter the password for the user to add the smart card to the Keychain

The next time the user logs in, the user can authenticate by inserting their smart card and typing their PIN on the login screen. If the PIN verification fails, the authentication is unsuccessful and one authentication attempt is consumed.

3.3 Password Policy

As an administrator, run `sudo pwpolicy –setaccountpolicies <policy_plist>`

<policy_plist> is a plist file that specifies the password policy. Its basic structure is:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version ="1.0">
<dict>
  <key>policyCategoryPasswordContent</key>
  <array>
   [passwordContentPolicies]
  </array>
</dict>
</plist>
```

[passwordContentPolicies] may be one or more of the following password policy rules:

- Minimum Password Length:

```
<dict>
  <key>policyContent</key>
  <string>policyAttributePassword matches
'.{[minimum_length],}+'</string>
</dict>
```

- Minimum Number of Special Characters:

```
<dict>
  <key>policyContent</key>
  <string>policyAttributePassword matches '(.*[^a-zA-Z0-
9].*){[number_of_characters],}+'</string>
</dict>
```

- Maximum Failed Login Attempts:

```
<dict>
  <key>autoEnableInSeconds</key>
  <integer>$LOCKOUT</integer>        # Lockout period in
seconds (300 == 5 minutes)
  <key>policyAttributeMaximumFailedAuthentications</key>
  <integer>$MAX_FAILED</integer>     # Example: 5 max failed
logins before locking
</dict>
```

For additional details and options, please see the pwpolicy man page.

Note: Additional policy categories and policy content are unevaluated but may be included.

3.4 Screen Lock

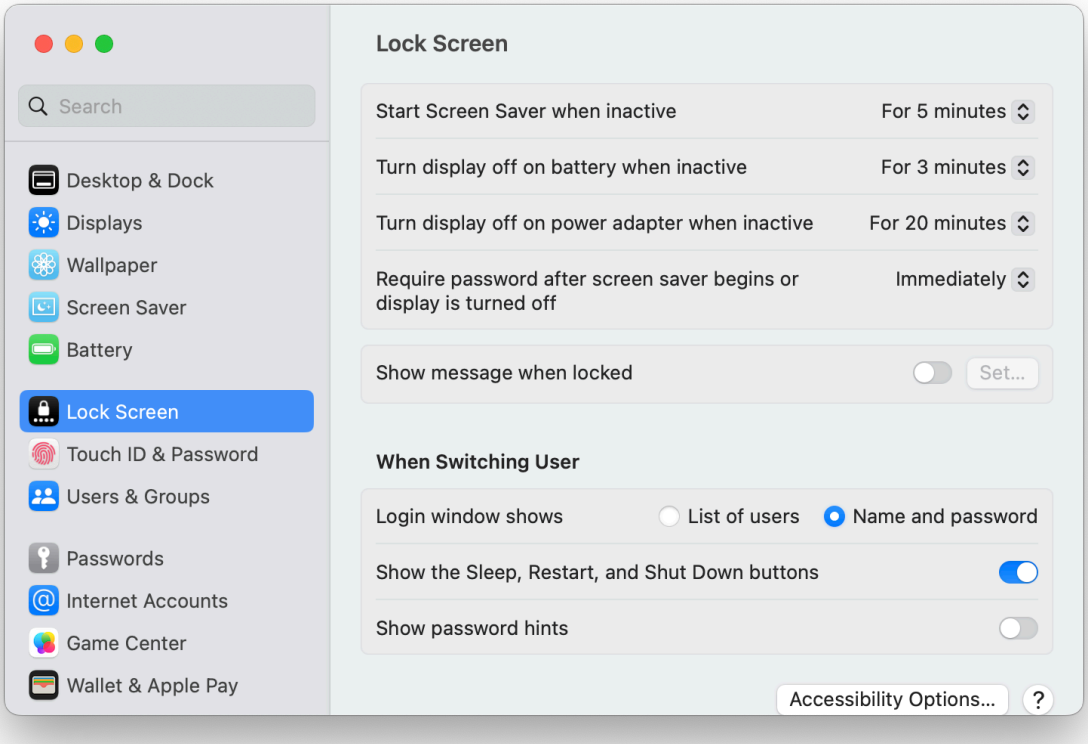
3.4.1 Enable/Disable

As a user:

1. Open the System Settings app
2. Select Lock Screen
3. Select a time for “Require password after screensaver beings or display is turned off” to enable the screen lock. Select Never to disable the screen lock.

4. Authenticate as the user to approve the change

Figure 6 – Enable/Disable Screen Lock



3.4.2 Inactivity Timeout

The inactivity timeout is the combination of the "Require password after screensaver begins or display is turned off" and the lesser of the "Start Screen Saver when inactive" and "Turn display off …" settings.

Configure "Require password after screensaver beings or display is turned off" as a user:

1. Open the System Settings app
2. Select Lock Screen
3. Select a time from the "Require password after screensaver beings or display is turned off" drop down list
4. Authenticate as the user to approve the change

Configure "Start Screen Saver when inactive" as a user:

1. Open the System Settings app
2. Select Lock Screen
3. Select a time from the "Start Screen Saver when inactive" drop down list

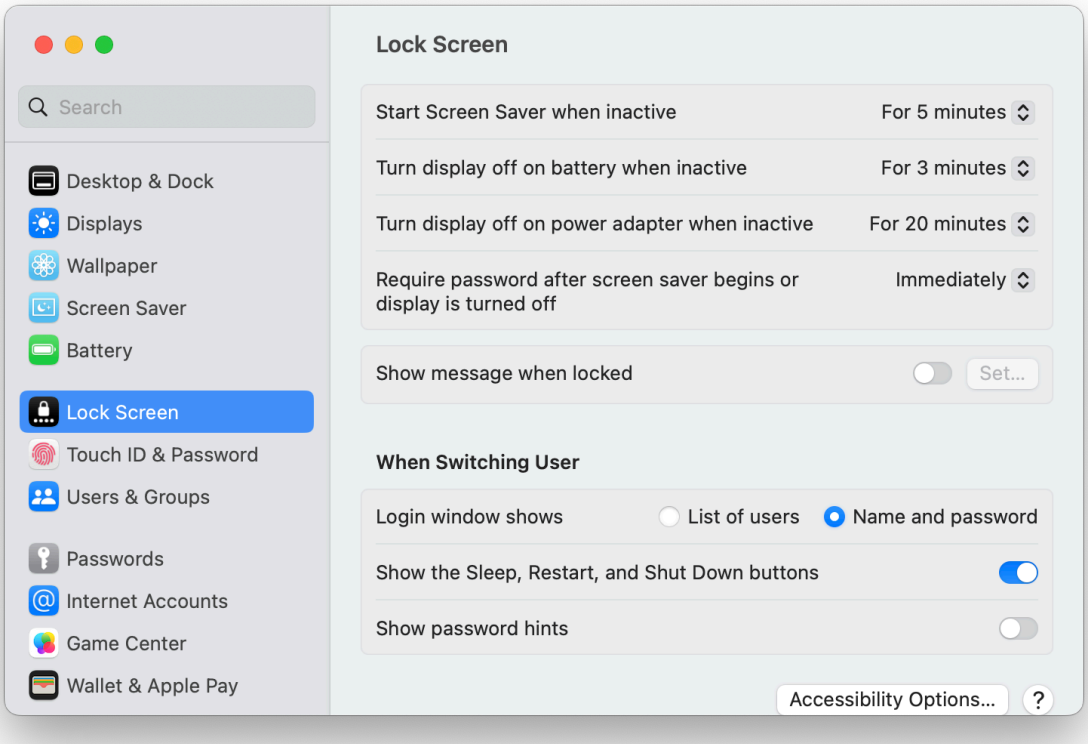
Configure "Turn display off on power adapter when inactive" as an administrator:

1. Open the System Settings app
2. Select Lock Screen
3. Select a time from the "Turn display off on power adapter when inactive" drop down list

If available, configure "Turn display off on battery when inactive" as an administrator:

1. Open the System Settings app
2. Select Lock Screen
3. Select a time from the "Turn display off on battery when inactive" drop down list

Figure 7 – Screen Saver



3.5 Warning Banner

As an administrator:

1. In the */Library/Security* folder, create either a plain-text (.txt) or rich-text (.rtf) file named *PolicyBanner.txt* or *PolicyBanner.rtf*, respectively, containing the warning banner text
2. Change the file's access rights to 644: chmod 644 <filename>

As the user:

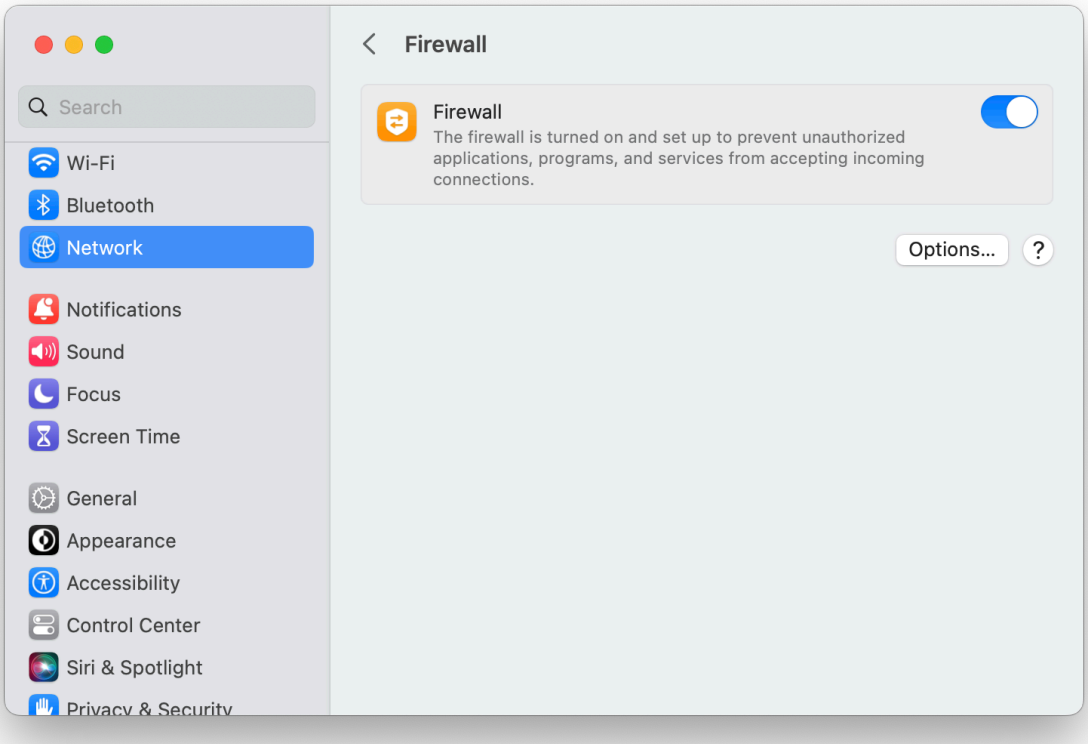
1. Execute: diskutil apfs updatePreboot /
2. Restart macOS for the changes to take effect

3.6 Firewall

The host-based firewall can be managed by an administrator:

1. Open the System Settings app
2. Select Network
3. Select Firewall
4. Use the Firewall switch to enable and disable the firewall. Click “Options…” to configure specific application-based firewall rules
5. Authenticate as an administrator when prompted

Figure 8 – Firewall



3.7 Management Server

To configure the DNS name or IP address of an MDM server, as an administrator:

1. Obtain an Enrollment Profile from the MDM administrator. The Enrollment Profile must contain the DNS name or IP address of the MDM server
2. Copy the Enrollment Profile onto the Mac
3. If a “Profile installation notification" does not appear (automatic recognition and processing of the Enrollment Profile), open the Enrollment Profile
4. Open the System Settings app
5. Select Profiles
6. Select the New Enrollment Profile
7. Click Install…

8. Confirm the installation and authenticate as an administrator

3.8 Audit

3.8.1 Server

To specify the remote syslog server to receive audit logs, as an administrator:

1. Edit /etc/syslog.conf
   Note: sudo must be used to elevate privileges
2. Add/edit lines in the following format:
   `<facility>.<level>  @<server>[:<port>]`

`<facility>` matches against the facility specified by generator of the log. The generator may specify any arbitrary string as the facility. macOS components use a "reverse ICANN" naming convention, for example "com.apple.system.syslog" and other users of the audit system are encouraged to use the same naming conventions. Because these facility names may contain dot characters, the names may be enclosed in either single quote or double quote characters. If facility is terminated by an asterisk ("*"), then facility names are matched using the prefix characters preceding the asterisk.

`<level>` matches the specified level or higher. An asterisk ("*") matches all levels.

`<server>` is the DNS name or IP address of the remote syslog server.

`<port>` optionally specifies the destination port. If `<port>` is omitted, 514 is used.

The audit command line utility controls the state of the audit system. Only an administrator can perform audit functions. The audit utility is invoked as follows:

audit -e | -i | -n | -s | -t

The options are as follows:

- -e Forces the audit system to immediately remove audit log files that meet the expiration criteria specified in the audit control file without doing a log rotation.
- -i Initializes and starts auditing. This option is currently for Mac OS X only and requires auditd(8) to be configured to run under launchd(8).
- -n Forces the audit system to close the existing audit log file and rotate to a new log file in a location specified in the audit control file. Also, audit log files that meet the expiration criteria specified in the audit control file will be removed.
- -s Specifies that the audit system should [re]synchronize its configuration from the audit control file. A new log file will be created. The attributable flags parameter from the audit_control(5) configuration file is set at login time and is not synchronized with this flag.
- -t Specifies that the audit system should terminate. Log files are closed and renamed to indicate the time of the shutdown.

3.8.2 Rules

To edit audit rules, as an administrator:

1. Edit `/etc/security/audit_control`
   Note: sudo must be used to elevate privileges
2. Add/edit lines in the following format:
   `parameter:value`
3. Run `audit -s` or restart the system for changes to take effect

The parameters may be:

`dir` The directory where audit log files are stored. There may be more than one of these entries. Changes to this entry can only be enacted by restarting the audit system.

`flags` Specifies which audit event classes are audited for all users. flags is a comma-delimited list of audit classes with optional prefixes.

Classes are:

- `no` invalid class
- `fr` file read
- `fw` file write
- `fa` file attribute access
- `fm` file attribute modify
- `fc` file create
- `fd` file delete
- `cl` file close
- `pc` process
- `nt` network
- `ip` ipc
- `na` non attributable
- `ad` administrative
- `lo` login_logout
- `aa` authentication and authorization
- `ap` application
- `res` reserved for internal use
- `io` ioctl
- `ex` exec
- `ot` miscellaneous
- `all` all flags set

Prefixes are:

- (none) Record both successful and failed events.
- `+` Record successful events.
- `-` Record failed events.
- `^` Record neither successful nor failed events.
- `^+` Do not record successful events.
- `^-` Do not record failed events.

By default, macOS is configured to log lo and aa classes. To log configuration changes fr and fw classes must be enabled; however, this causes macOS to generate a large number of logs.

`host` Specify the hostname or IP address to be used when setting the local system's audit host information. This hostname will be converted into an IP or IPv6 address and will be

included in the header of each audit record. Due to the possibility of transient errors coupled with the security issues in the DNS protocol itself, the use of DNS should be avoided. Instead, it is strongly recommended that the hostname be specified in the /etc/hosts file.

`naflags` Contains the audit flags that define what classes of events are audited when an action cannot be attributed to a specific user.

`minfree` The minimum free space required on the file system audit logs are being written to. When the free space falls below this limit a warning will be issued. If no value for the minimum free space is set, the default of 20 percent is applied by the kernel.

`policy` A list of global audit policy flags specifying various behaviors, such as fail stop, auditing of paths and arguments, etc.

The policy flags field is a comma-delimited list of policy flags from the following list:

- `cnt` Allow processes to continue running even though events are not being audited. If not set, processes will be suspended when the audit store space is exhausted. Currently, this is not a recoverable state.
- `ahlt` Fail stop the system if unable to audit an event--this consists of first draining pending records to disk, and then halting the operating system.
- `argv` Audit command line arguments.
- `arge` Audit environmental variable arguments.
- `seq` Include a unique audit sequence number token in generated audit records (not implemented on FreeBSD or Darwin).
- `group` Include supplementary groups list in generated audit records (not implemented on FreeBSD or Darwin; supplementary groups are never included in records on these systems).
- `trail` Append a trailer token to each audit record (not implemented on FreeBSD or Darwin; trailers are always included in records on these systems).
- `path` Include secondary file paths in audit records (not implemented on FreeBSD or Darwin; secondary paths are never included in records on these systems).
- `zonename` Include a zone ID token with each audit record (not implemented on FreeBSD or Darwin; FreeBSD audit records do not currently include the jail ID or name).
- `perzone` Enable auditing for each local zone (not implemented on FreeBSD or Darwin; on FreeBSD, audit records are collected from all jails and placed in a single global trail, and only limited audit controls are permitted within a jail).

`filesz` Maximum trail size in bytes; if set to a non-0 value, the audit daemon will rotate the audit trail file at around this size. Sizes less than the minimum trail size (default of 512K) will be rejected as invalid. If 0, trail files will not be automatically rotated based on file size. For convenience, the trail size may be expressed with suffix letters: B (Bytes), K (Kilobytes), M (Megabytes), or G (Gigabytes). For example, 2M is the same as 2097152.

`expire-after` Specifies when audit log files will expire and be removed. This may be after a time period has passed since the file was last written to or when the aggregate of all the trail files have reached a specified size or a combination of both. If no expire-after parameter is given then audit log files will not expire and be removed by the audit control system.

The expiration specification can be one value or two values with the logical conjunction of AND/OR between them. Values for the audit log file age are numbers with the following suffixes:

- s Log file age in seconds.
- h Log file age in hours.
- d Log file age in days.
- y Log file age in years.

Values for the disk space used are numbers with the following suffixes:

- (none) or
- B Disk space used in Bytes.
- K Disk space used in Kilobytes.
- M Disk space used in Megabytes.
- G Disk space used in Gigabytes.

The suffixes on the values are case sensitive. If both an age and disk space value are used, they are separated by AND or OR and both values are used to determine when audit log files expire. In the case of AND, both the age and disk space conditions must be met before the log file is removed. In the case of OR, either condition may expire the log file.

3.8.3 Review

macOS includes the auditreduce and praudit command line utilities for the administrator to review local audit logs. Any user can use the auditreduce and praudit utilities, but macOS restricts access to audit logs, so an administrator must use sudo to access the audit logs stored in `/var/log`.

auditreduce select records from audit trail files. The auditreduce utility is invoked as follows:

```
auditreduce [-A] [-a YYYYMMDD[HH[MM[SS]]]] [-b
YYYYMMDD[HH[MM[SS]]]] [-c flags] [-d YYYYMMDD] [-e euid] [-f
egid] [-g rgid] [-j id] [-m event] [-o object=value] [-r ruid]
[-u auid] [-v] <file>
```

The options are as follows:

- `-A` Select all records.
- `-a YYYYMMDD[HH[MM[SS]]]`
  Select records that occurred after or on the given datetime.
- `-b YYYYMMDD[HH[MM[SS]]]`
  Select records that occurred before the given datetime.
- `-c <flags>`
  Select records matching the given audit classes specified as a comma separated list of audit flags. See Section 3.8.2 for a description of audit flags.
- `-d YYYYMMDD`
  Select records that occurred on a given date. This option cannot be used with -a or -b.
- `-e euid`
  Select records with the given effective user ID or name.
- `-f egid`
  Select records with the given effective group ID or name.

- `-g rgid`
  Select records with the given real group ID or name.
- `-j id`
  Select records having a subject token with matching ID.
- `-m event`
  Select records with the given event name or number. This option can be used more then once to select records of multiple event types.
- `-o object=value`
  - `file` Select records containing path tokens, where the pathname matches one of the comma delimited extended regular expression contained in given specification. Regular expressions which are prefixed with a tilde ('~') are excluded from the search results. These extended regular expressions are processed from left to right, and a path will either be selected or deselected based on the first match.
    Since commas are used to delimit the regular expressions, a backslash ('\') character should be used to escape the comma if it is a part of the search pattern.
  - `msgqid` Select records containing the given message queue ID.
  - `pid` Select records containing the given process ID.
  - `semid` Select records containing the given semaphore ID.
  - `shmid` Select records containing the given shared memory ID.
- `-r ruid`
  Select records with the given real user ID or name.
- `-u auid`
  Select records with the given audit ID.
- `-v` Invert sense of matching, to select records that do not match.
- `<file>`
  The audit log file(s) to search and output. Typically, this parameter should be `/var/audit/*` to search all of the audit log files.

auditreduce outputs data in a binary format. Typically, its output should be piped to praudit.

praudit prints the contents of the audit trail. The praudit utility is invoked as follows:

```
praudit [-lnpx] [-r | -s] [-d del]
```

The options are as follows:

- `-d del` Specifies the delimiter. The default delimiter is the comma.
- `-l` Prints the entire record on the same line. If this option is not specified, every token is displayed on a different line.
- `-n` Do not convert user and group IDs to their names but leave in their numeric forms.
- `-p` Specify this option if input to praudit is piped from the tail(1) utility. This causes praudit to sync to the start of the next record.
- `-r` Prints the records in their raw form. Show records and event types in a numeric form (also known as raw form). This option is exclusive from `-s`.
- `-s` Prints the records in their short form. Show records and events in a short textual representation. This option is exclusive from `-r`.

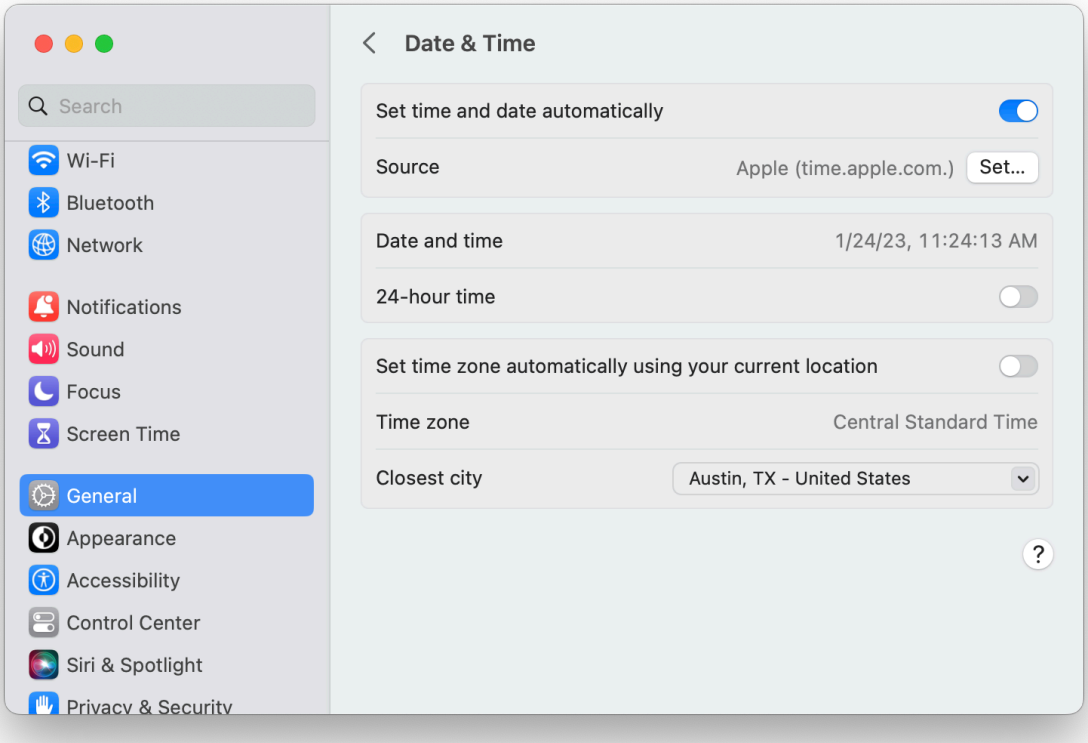
- –x Print audit records in the XML output format.

3.9 Network Time Server

As an administrator:

1. Open the System Settings app
2. Select General
3. Select Date & Time
4. Click Set… in the Source field
5. Authenticate as an administrator when prompted
6. Enter the network time server DNS name or IP address in the “Time Server” field

Figure 9 – Network Time Server



3.10 Automatic Software Updates

As an administrator:

1. Open the System Settings app
2. Select General
3. Select Software Update
4. Click the information icon (the letter i in the circle) in the “Automatic updates” field
5. Enable/disable “Check for updates”
6. Enable/disable “Download new updates when available”
7. Enable/disable “Install macOS updates”

8. Enable/disable “Install application updates from the App Store”
9. Enable/disable “Install Security Responses and system files” to enable/disable the automatic application of Apple’s Rapid Security Responses (RSR) updates
10. Select Done
11. Authenticate as an administrator

Figure 10 – Software Updates

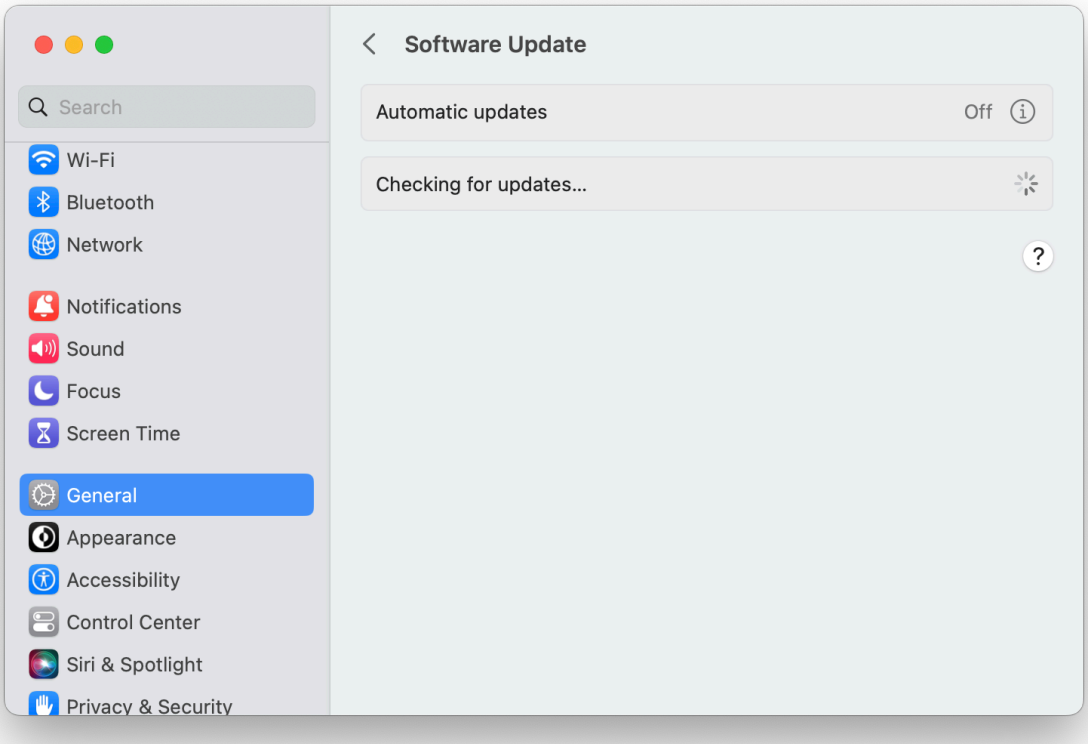
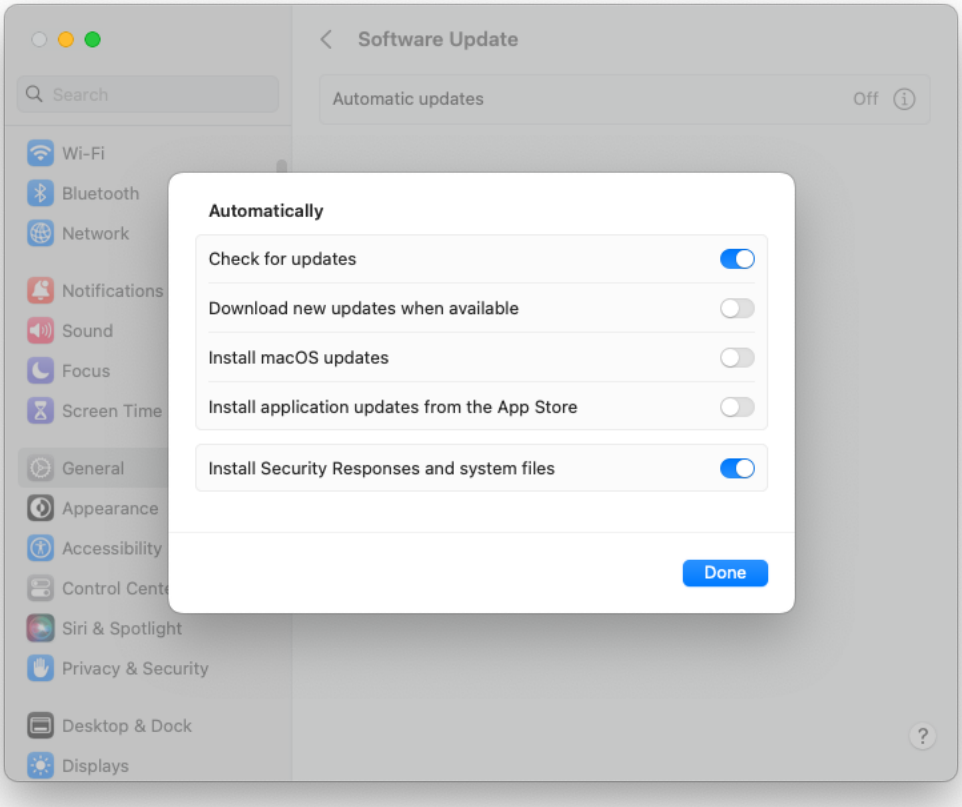


Figure 11 - Automatic Software Updates



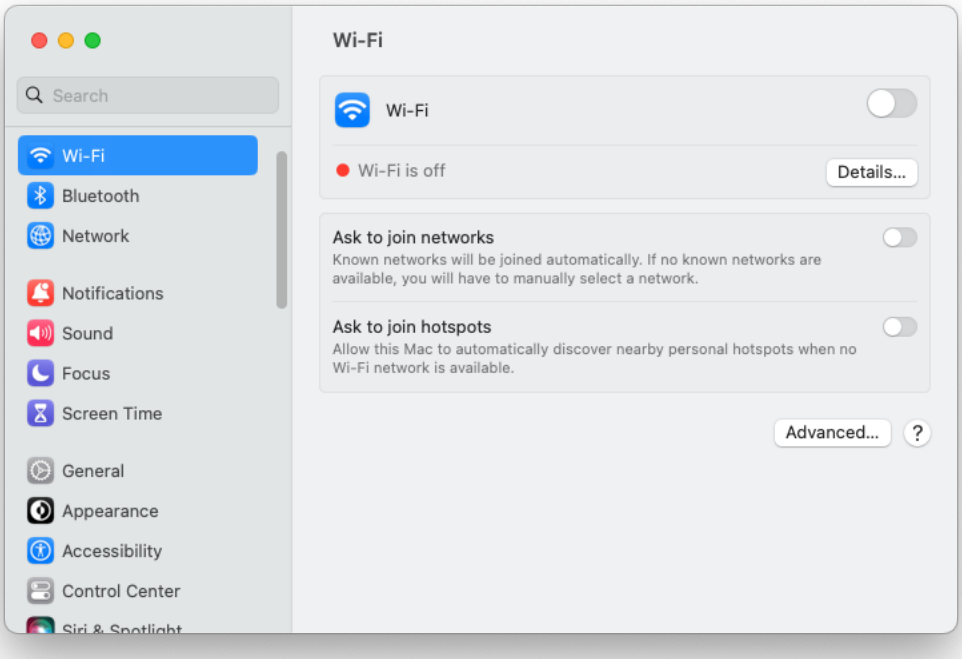
3.11 Wi-Fi

3.11.1 Enable/Disable Wi-Fi

As a user:

1. Open the System Settings app
2. Select Wi-Fi
3. Use the Wi-Fi switch to enable and disable Wi-Fi

Figure 12 – Enable/Disable Wi-Fi



3.11.2 Join Networks

As a user:

1. Open the System Settings app
2. Select Wi-Fi
3. Select the Wi-Fi network name

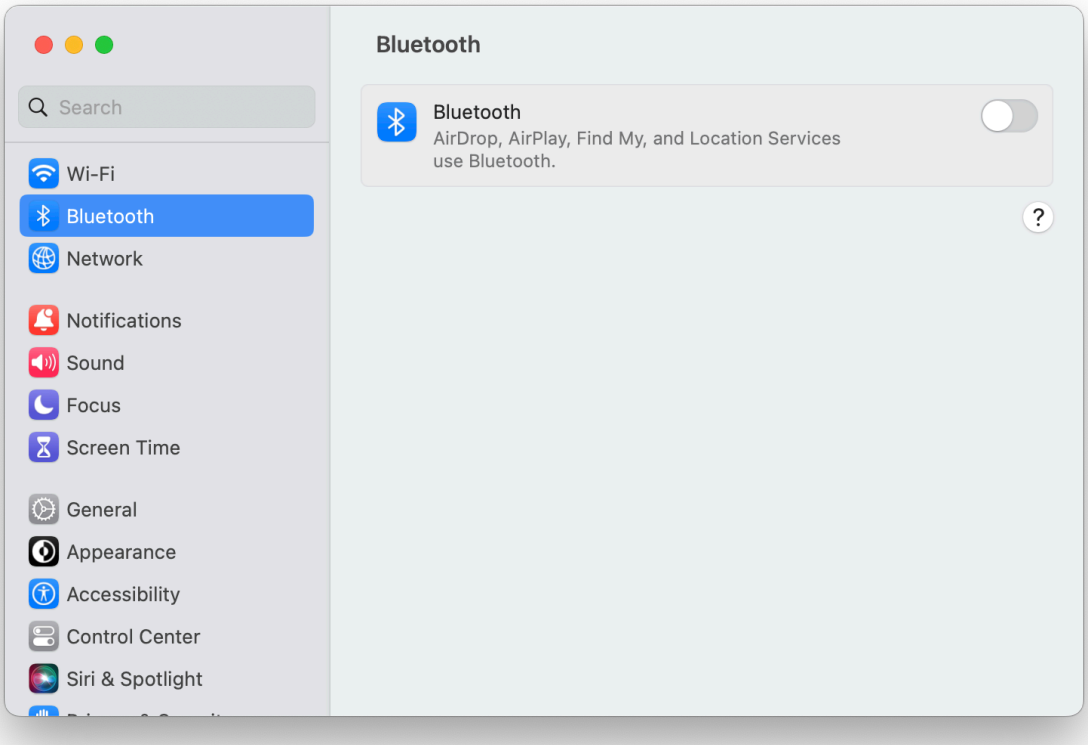
3.12 Bluetooth

3.12.1 Enable/Disable Bluetooth

As a user:

1. Open the System Settings app
2. Select Bluetooth
3. Use the Bluetooth switch to enable and disable Bluetooth.

Figure 13 – Enable/Disable Bluetooth



3.12.2 Add/Pair

As a user:

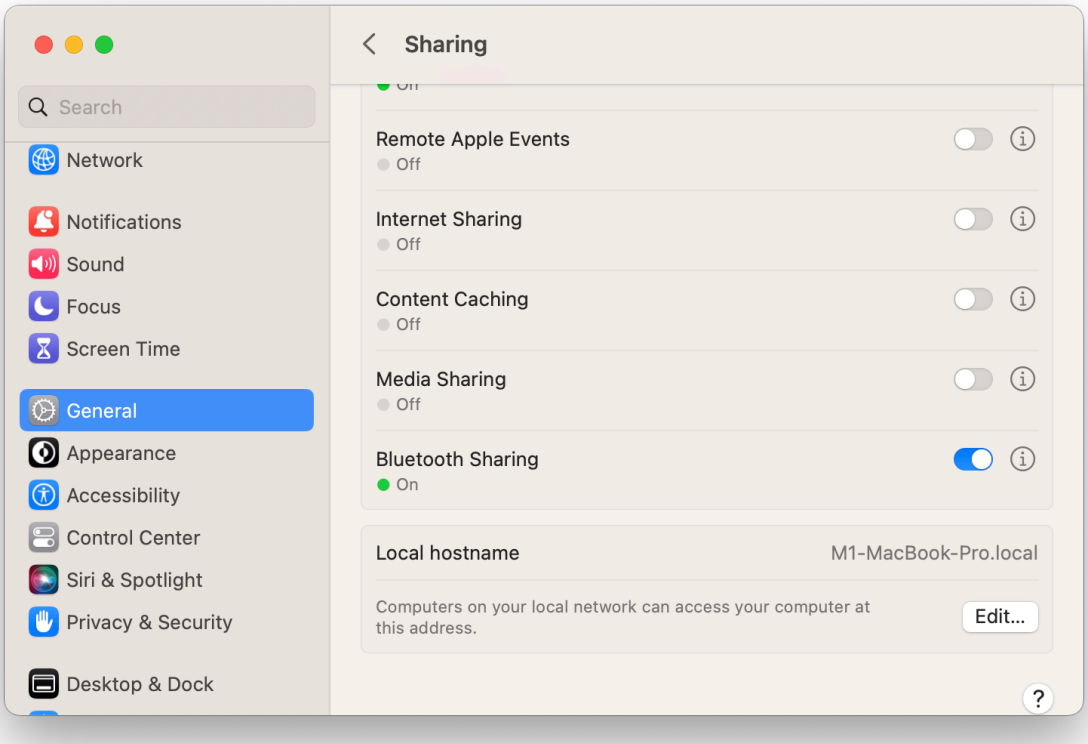
1. Open the System Settings app
2. Select Bluetooth
3. Place the other device in discoverable mode
4. Select the other device from the list of devices. Enter the PIN, if required

3.12.3 Enable/Disable Discoverable/Advertising Mode

When Bluetooth is enabled (see section 3.12.1 for instructions on how to enable Bluetooth), the Discoverable/Advertising mode can be enabled/disabled by performing the following actions:

1. Open the System Settings app
2. Select General
3. Select Sharing
4. Enable or disable Discoverable/Advertising mode by enabling or disabling Bluetooth Sharing, respectively.

Figure 14 - Enable/Disable Bluetooth Sharing



3.13 Access Control

3.13.1 Sandbox Entitlement

The TOE supports the use of app sandbox to limit an app's ability to access files. These limits override any permissions the app might otherwise have. Sandbox limits are subtractive, not additive. Therefore, the file system permissions represent the maximum access an app might be allowed if its sandbox also permits that access.

To enable App Sandbox on an app, in Xcode, select the project where your app resides, select the app's Target, then select Signing & Capabilities. If "App Sandbox" does not appear, click "Capability and add "App Sandbox." A set of checkboxes and selection lists will appear. Enable access to one or more resources by checking one or more checkboxes and/or selecting from one or more selection lists.

3.13.2 POSIX ACLs

POSIX ACLs can be managed using the chmod command.

POSIX ACLs are manipulated using extensions to the symbolic mode grammar. Each file has one ACL, containing an ordered list of entries. Each entry refers to a user or group, and grants or denies a set of permissions. In cases where a user and a group exist with the same name,

the user/group name can be prefixed with "user:" or "group:" in order to specify the type of name.

If the user or group name contains spaces, you can use ':' as the delimiter between name and permission.

The following permissions are applicable to all filesystem objects:

Delete
: Delete the item. Deletion may be granted by either this permission on an object or the delete_child right on the containing directory.

readattr
: Read an object's basic attributes. This is implicitly granted if the object can be looked up and not explicitly denied.

writeattr
: Write an object's basic attributes.

readextattr
: Read extended attributes.

writeextattr
: Write extended attributes.

readsecurity
: Read an object's extended security information (ACL).

writesecurity
: Write an object's security information (ownership, mode, ACL).

chown
: Change an object's ownership.

The following permissions are applicable to directories:

list
: List entries.

search
: Look up files by name.

add_file
: Add a file.

add_subdirectory
: Add a subdirectory.

delete_child
: Delete a contained object. See the file delete permission above.

The following permissions are applicable to non-directory filesystem objects:

read
: Open for reading.

write
: Open for writing.

append
: Open for writing, but in a fashion that only allows writes into areas of the file not previously written.

execute
: Execute the file as a script or program.

ACL inheritance is controlled with the following permissions words, which may only be applied to directories:

file_inherit
: Inherit to files.

directory_inherit
: Inherit to directories.

limit_inherit
This flag is only relevant to entries inherited by subdirectories; it causes the directory_inherit flag to be cleared in the entry that is inherited, preventing further nested subdirectories from also inheriting the entry.

only_inherit
The entry is inherited by created items but not considered when processing the ACL.

The ACL manipulation options are as follows:

+a The +a mode parses a new ACL entry from the next argument on the command line and inserts it into the canonical location in the ACL. If the supplied entry refers to an identity already listed, the two entries are combined.

Examples

```
# ls -le
-rw-r--r--+ 1 juser  wheel  0 Apr 28 14:06 file1
# chmod +a "admin allow write" file1
# ls -le
-rw-r--r--+ 1 juser  wheel  0 Apr 28 14:06 file1
 owner: juser
 1: admin allow write
# chmod +a "guest deny read" file1
# ls -le
-rw-r--r--+ 1 juser  wheel  0 Apr 28 14:06 file1
 owner: juser
 1: guest deny read
 2: admin allow write
# chmod +a "admin allow delete" file1
# ls -le
-rw-r--r--+ 1 juser  wheel  0 Apr 28 14:06 file1
 owner: juser
 1: guest deny read
 2: admin allow write,delete
# chmod +a "User 1:allow:read" file
# ls -le
-rw-r--r--+ 1 juser  wheel  0 Apr 28 14:06 file1
 owner: juser
 1: guest deny read
 2: User 1 allow read
 3: admin allow write,delete
```

The +a mode strives to maintain correct canonical form for the ACL.

local deny
local allow
inherited deny
inherited allow

By default, chmod adds entries to the top of the local deny and local allow lists. Inherited entries are added by using the +ai mode.

Examples
```
# ls -le
-rw-r--r--+ 1 juser  wheel  0 Apr 28 14:06 file1
  owner: juser
  1: guest deny read
  2: admin allow write,delete
  3: juser inherited deny delete
  4: admin inherited allow delete
  5: backup inherited deny read
  6: admin inherited allow write-security
# chmod +ai "others allow read" file1
# ls -le
-rw-r--r--+ 1 juser  wheel  0 Apr 28 14:06 file1
  owner: juser
  1: guest deny read
  2: admin allow write,delete
  3: juser inherited deny delete
  4: others inherited allow read
  5: admin inherited allow delete
  6: backup inherited deny read
  7: admin inherited allow write-security
```

+a# When a specific ordering is required, the exact location at which an entry will be inserted is specified with the +a# mode.

Examples
```
# ls -le
-rw-r--r--+ 1 juser  wheel  0 Apr 28 14:06 file1
  owner: juser
  1: guest deny read
  2: admin allow write
# chmod +a# 2 "others deny read" file1
# ls -le
-rw-r--r--+ 1 juser  wheel  0 Apr 28 14:06 file1
  owner: juser
  1: guest deny read
  2: others deny read
  3: admin allow write
```

The +ai# mode may be used to insert inherited entries at a specific location. Note that these modes allow non-canonical ACL ordering to be constructed.

-a The -a mode is used to delete ACL entries. All entries exactly matching the supplied entry will be deleted. If the entry lists a subset of rights granted by an entry, only the rights listed are removed. Entries may also be deleted by index using the -a# mode.

Examples
```
# ls -le
```

```
-rw-r--r--+ 1 juser  wheel  0 Apr 28 14:06 file1
  owner: juser
  1: guest deny read
  2: admin allow write,delete
# chmod -a# 1 file1
# ls -le
-rw-r--r--+ 1 juser  wheel  0 Apr 28 14:06 file1
  owner: juser
  1: admin allow write,delete
# chmod -a "admin allow write" file1
# ls -le
-rw-r--r--+ 1 juser  wheel  0 Apr 28 14:06 file1
  owner: juser
  1: admin allow delete
```

Inheritance is not considered when processing the -a mode; rights and entries will be removed regardless of their inherited state.

If the user or group name contains spaces you can use ':' as the delimiter.

Example

```
# chmod +a "User 1:allow:read" file
```

=a# Individual entries are rewritten using the =a# mode.

Examples

```
# ls -le
-rw-r--r--+ 1 juser  wheel  0 Apr 28 14:06 file1
  owner: juser
  1: admin allow delete
# chmod =a# 1 "admin allow write,chown"
# ls -le
-rw-r--r--+ 1 juser  wheel  0 Apr 28 14:06 file1
  owner: juser
  1: admin allow write,chown
```

This mode may not be used to add new entries.

-E Reads the ACL information from stdin, as a sequential list of ACEs, separated by newlines. If the information parses correctly, the existing information is replaced.

-C Returns false if any of the named files have ACLs in non-canonical order.

-i Removes the 'inherited' bit from all entries in the named file(s) ACLs.

-I Removes all inherited entries from the named file(s) ACL(s).

-N Removes the ACL from the named file(s).

3.13.3 Unix Permissions

The Unix permissions of a file system object can be managed using the chown, chgrp, and chmod commands. For file system objects: the chown command changes the owner (user ID) and/or group ID; the chgrp command changes the group ID; and the chmod command changes the permission sets.

3.13.3.1 chown

The chown command line format is as follows:

```
chown [-fhnv] [-R [-H | -L | -P]] owner[:group] file ...
chown [-fhnv] [-R [-H | -L | -P]] :group file ...
```

The chown utility changes the user ID and/or the group ID of the specified files. Symbolic links named by arguments are silently left unchanged unless -h is used.

The options are as follows:

-f Don't report any failure to change file owner or group, nor modify the exit status to reflect such failures.

-H If the -R option is specified, symbolic links on the command line are followed. (Symbolic links encountered in the tree traversal are not followed.)

-h If the file is a symbolic link, change the user ID and/or the group ID of the link itself.

-L If the -R option is specified, all symbolic links are followed.

-P If the -R option is specified, no symbolic links are followed. Instead, the user and/or group ID of the link itself are modified. This is the default. Use -h to change the user ID and/or the group of symbolic links.

-R Change the user ID and/or the group ID for the file hierarchies rooted in the files instead of just the files themselves.

-n Interpret user ID and group ID as numeric, avoiding name lookups.

-v Cause chown to be verbose, showing files as the owner is modified.

The -H, -L and -P options are ignored unless the -R option is specified. In addition, these options override each other and the command's actions are determined by the last one specified.

The owner and group operands are both optional; however, at least one must be specified. If the group operand is specified, it must be preceded by a colon (“:”) character.

The owner may be either a numeric user ID or a user name. If a user name is also a numeric user ID, the operand is used as a user name. The group may be either a numeric group ID or a group name. If a group name is also a numeric group ID, the operand is used as a group name.

The ownership of a file may only be altered by a super-user. Similarly, only a member of a group can change a file's group ID to that group.

3.13.3.2 chgrp

The chgrp command line format is as follows:

chgrp [-fhnv] [-R [-H | -L | -P]] group file ...

The chgrp utility sets the group ID of the file named by each file operand to the group ID specified by the group operand.

The following options are available:

-f The force option ignores errors, except for usage errors and doesn't query about strange modes (unless the user does not have proper permissions).

-H If the -R option is specified, symbolic links on the command line are followed. (Symbolic links encountered in the tree traversal are not followed).

-h If the file is a symbolic link, the group ID of the link itself is changed rather than the file that is pointed to.

-L If the -R option is specified, all symbolic links are followed.

-P If the -R option is specified, no symbolic links are followed. This is the default. Use -h to change the group ID of a symbolic link.

-R Change the group ID for the file hierarchies rooted in the files instead of just the files themselves.

-n Interpret the group ID as numeric, avoiding the name lookup.

-v Cause chgrp to be verbose, showing files as the group is modified.

The -H, -L and -P options are ignored unless the -R option is specified. In addition, these options override each other and the command's actions are determined by the last one specified.

The group operand can be either a group name from the group database, or a numeric group ID. If a group name is also a numeric group ID, the operand is used as a group name.

The user invoking chgrp must belong to the specified group and be the owner of the file, or be the super-user.

3.13.3.3 chown

The chmod command line format is as follows:

chmod [-fv] [-R [-H | -L | -P]] mode file ...

The chmod utility modifies the file mode bits of the listed files as specified by the mode operand. It may also be used to modify the Access Control Lists (ACLs) associated with the listed files.

The generic options are as follows:

-f Do not display a diagnostic message if chmod could not modify the mode for file, nor modify the exit status to reflect such failures.

-H If the -R option is specified, symbolic links on the command line are followed and hence unaffected by the command. (Symbolic links encountered during tree traversal are not followed.)

-h If the file is a symbolic link, change the mode of the link itself rather than the file that the link points to.

-L If the -R option is specified, all symbolic links are followed.

-P If the -R option is specified, no symbolic links are followed. This is the default.

-R Change the modes of the file hierarchies rooted in the files, instead of just the files themselves. Beware of unintentionally matching the ".." hard link to the parent directory when using wildcards like ".*".

-v Cause chmod to be verbose, showing filenames as the mode is modified. If the -v flag is specified more than once, the old and new modes of the file will also be printed, in both octal and symbolic notation.

The -H, -L and -P options are ignored unless the -R option is specified. In addition, these options override each other and the command's actions are determined by the last one specified.

Only the owner of a file or the super-user is permitted to change the mode of a file.

Modes may be absolute or symbolic. An absolute mode is an octal number constructed from the sum of one or more of the following values:

4000 (the setuid bit). Executable files with this bit set will run with effective uid set to the uid of the file owner. Directories with this bit set will force all files and subdirectories created in them to be owned by the directory owner and not by the uid of the creating process, if the underlying file system supports this feature: see chmod(2) and the suiddir option to mount(8).
2000 (the setgid bit). Executable files with this bit set will run with effective gid set to the gid of the file owner.
1000 (the sticky bit). See chmod(2) and sticky(7).
0400 Allow read by owner.

0200 Allow write by owner.
0100 For files, allow execution by owner. For directories, allow the owner to search in the directory.
0040 Allow read by group members.
0020 Allow write by group members.
0010 For files, allow execution by group members. For directories, allow group members to search in the directory.
0004 Allow read by others.
0002 Allow write by others.
0001 For files, allow execution by others. For directories allow others to search in the directory.

For example, the absolute mode that permits read, write and execute by the owner, read and execute by group members, read and execute by others, and no set-uid or set-gid behaviour is 755 (400+200+100+040+010+004+001).

The symbolic mode is described by the following grammar:

```
mode        ::= clause [, clause ...]
clause      ::= [who ...] [action ...] action
action      ::= op [perm ...]
who         ::= a | u | g | o
op          ::= + | - | =
perm        ::= r | s | t | w | x | X | u | g | o
```

The who symbols “u”, “g”, and “o” specify the user, group, and other parts of the mode bits, respectively. The who symbol “a” is equivalent to “ugo”.

The perm symbols represent the portions of the mode bits as follows:

r The read bits.
s The set-user-ID-on-execution and set-group-ID-on-execution bits.
t The sticky bit.
w The write bits.
x The execute/search bits.
X The execute/search bits if the file is a directory or any of the execute/search bits are set in the original (unmodified) mode. Operations with the perm symbol “X” are only meaningful in conjunction with the op symbol “+”, and are ignored in all other cases.
u The user permission bits in the original mode of the file.
g The group permission bits in the original mode of the file.
o The other permission bits in the original mode of the file.

The op symbols represent the operation performed, as follows:

+ If no value is supplied for perm, the “+” operation has no effect. If no value is supplied for who, each permission bit specified in perm, for which the corresponding bit in the file mode creation mask (see umask(2)) is clear, is set. Otherwise, the mode bits represented by the specified who and perm values are set.

- If no value is supplied for perm, the "-" operation has no effect. If no value is supplied for who, each permission bit specified in perm, for which the corresponding bit in the file mode creation mask is set, is cleared. Otherwise, the mode bits represented by the specified who and perm values are cleared.

= The mode bits specified by the who value are cleared, or, if no who value is specified, the owner, group and other mode bits are cleared. Then, if no value is supplied for who, each permission bit specified in perm, for which the corresponding bit in the file mode creation mask is clear, is set. Otherwise, the mode bits represented by the specified who and perm values are set.

Each clause specifies one or more operations to be performed on the mode bits, and each operation is applied to the mode bits in the order specified.

Operations upon the other permissions only (specified by the symbol "o" by itself), in combination with the perm symbols "s" or "t", are ignored.

The "w" permission on directories will permit file creation, relocation, and copy into that directory. Files created within the directory itself will inherit its group ID.

3.13.4 BSD File Flags

The BSD File Flags can be managed using the chflags command.

The chflags utility modifies the file flags of the listed files as specified by the flags operand.

The options are as follows:

-f Do not display a diagnostic message if chflags could not modify the flags for file, nor modify the exit status to reflect such failures.

-H If the -R option is specified, symbolic links on the command line are followed. (Symbolic links encountered in the tree traversal are not followed.)

-h If the file is a symbolic link, change the file flags of the link itself rather than the file to which it points.

-L If the -R option is specified, all symbolic links are followed.

-P If the -R option is specified, no symbolic links are followed. This is the default.

-R Change the file flags for the file hierarchies rooted in the files instead of just the files themselves.

-v Cause chflags to be verbose, showing filenames as the flags are modified. If the -v option is specified more than once, the old and new flags of the file will also be printed, in octal notation.

The flags are specified as an octal number or a comma separated list of keywords. The following keywords are currently defined:

arch, archived
set the archived flag (super-user only)

opaque
set the opaque flag (owner or super-user only). [Directory is opaque when viewed through a union mount]

nodump set the nodump flag (owner or super-user only)

sappnd, sappend
set the system append-only flag (super-user only)

schg, schange, simmutable
set the system immutable flag (super-user only)

uappnd, uappend
set the user append-only flag (owner or super-user only)

uchg, uchange, uimmutable
set the user immutable flag (owner or super-user only)

hidden
set the hidden flag [Hide item from GUI]

Putting the letters “no” before or removing the letters “no” from a keyword causes the flag to be cleared. For example:

nouchg clear the user immutable flag (owner or super-user only)
dump clear the nodump flag (owner or super-user only)

Unless the -H or -L options are given, chflags on a symbolic link always succeeds and has no effect. The -H, -L and -P options are ignored unless the -R option is specified. In addition, these options override each other and the command's actions are determined by the last one specified.

You can use "ls -lO" to see the flags of existing files.

4 Secure Communications

macOS supports secure communications using TLS. TLS is used by macOS to communicate with the Apple Update Server and by applications to communicate with TLS servers specified by the applications.

Applications invoke TLS using the URLSession class with the “https” protocol. The reference identifiers are automatically generated from the host portion of the URL. macOS only generates reference identifiers that will match the SAN. When the host is a DNS name that contains at least three parts, macOS generates a reference identifier with the left-most position replaced with a wildcard.

Applications can perform TLS client authentication using an X.509 certificate by specifying the certificate in a URLCredential instance.

No additional configuration is required to ensure proper usage.

5 Storage of Credentials

macOS offers a repository, called Keychain, that securely stores the following sensitive data:

- X.509 Certificates used for TLS client authentication
- Private Keys associated with client certificates

Keychain items are encrypted. No configuration is necessary. The Key Encrypting Key used to protect Keychain keys is stored in the Secure Enclave Processor (SEP) which is part of the Security Chip in all hardware platforms. The SEP is specialized security hardware, so there are no instances when where key destruction may be delayed.

Users can manage keychains using the Keychain Access app. Standard users have permission to manage their own keychains, while administrators can also manage system keychains.

Applications can store and access credentials in a keychain using the Keychain Services API. Please see *section 5.2 "Keychain Services"* for additional details.

TOE usernames used for authentication, also considered as sensitive data, are maintained by the local directory services. The TOE provides a GUI for managing Users; see *section 3.2 "User Accounts"*.

5.1 Digital Certificates

The Keychain is also used to manage X.509 Certification Authority certificates used to validate the identity of TLS server and software developers. macOS leverages "Trusted" root digital certificates that are pre-installed in a keychain. No configuration is required to facilitate the usage of these digital certificates. Additional information regarding the trusted root certificates may be found at: https://support.apple.com/en-us/HT212140

5.2 Keychain Services

The following API functions pertaining to the Keychain Service can be used by applications to store and access credentials in a keychain.

5.2.1 SecItemAdd

Adds one or more items to a keychain.

```
OSStatus SecItemAdd(CFDictionaryRef attributes, CFTypeRef  _Nullable *result);
```

Parameters

`attributes`
A dictionary that describes the item to add. A typical attributes dictionary consists of:

- **The item's class.** Different attributes and behaviors apply to different classes of items. You use the kSecClass key with a suitable value to tell keychain services whether the data you want to store represents a password, a certificate, a cryptographic key, or something else. See *"Item class keys and values"*.

- **The data.** Use the kSecValueData key to indicate the data you want to store. Keychain services takes care of encrypting this data if the item is secret, namely when it's one of the password types or involves a private key.

- **Optional attributes.** Include attribute keys that help you find the item later, indicate how your app uses the data, and how the system shares the data. You can add any number of attributes, although many are specific to a particular class of item. For the attributes applicable to the keychain item you add, see *"Item class keys and values"*.

- **Optional return types.** Include return type keys to indicate what data, if any, you want returned upon successful completion. You often ignore the return data from a SecItemAdd call, in which case you don't need to include any return result key. See *"Item Result Keys"* for more information.

`result`
On return, a reference to the newly added items. The exact type of the result is based on the values supplied in attributes, as discussed in Item return result keys. Pass nil if you don't need the result. Otherwise, your app becomes responsible for releasing the referenced object.

Return Value

A result code. See *"Keychain Result Codes"*.

5.2.2 SecItemCopyMatching

Returns one or more keychain items that match a search query, or copies attributes of specific keychain items.

```
OSStatus SecItemCopyMatching(CFDictionaryRef query, CFTypeRef  _Nullable *result);
```

Parameters

`query`
A dictionary that describes the search. A typical query dictionary consists of:

- **The item's class.** Specify the kind of item you want, for example a password, a certificate, or a cryptographic key, using one of the class values in *"Item class keys and values"*.

- **Attributes.** Narrow the search by indicating the attributes that the found item or items should have. The more attributes you specify, the more refined the results, but not all attributes apply to all item classes. For the attributes applicable to the keychain item you're searching for, see *"Item class keys and values"*.

- **Search parameters.** Condition the search in a variety of ways. For example, you can limit the results to a specific number of items, control case sensitivity when matching string attributes, or search only among a particular set of items. See *"Search attribute keys and values"* for the complete list of possible search parameters.

- **One or more return types.** Use the keys found in *"Item Result Keys"* to indicate whether you seek the item's attributes, the item's data, a reference to the data, a persistent reference to the data, or a combination of these. When you specify more than one return type, the search returns a dictionary containing each of the types you request. When your search allows multiple results, they're all returned together in an array of items.

`result`

On return, a reference to the found items. The exact type of the result depends on the return type values supplied in query, as discussed in *"Item Result Keys"*.

Return Value

A result code. See *"Keychain Result Codes"*.

5.2.3 SecItemUpdate

Modifies items that match a search query.

```
OSStatus SecItemUpdate(CFDictionaryRef query, CFDictionaryRef attributesToUpdate);
```

Parameters

`query`
A dictionary that describes the search for the keychain items you want to update. A typical query dictionary consists of:

- **The item's class.** Specify the kind of item you want, for example a password, a certificate, or a cryptographic key, using one of the class values in *"Item class keys and values"*.
- **Attributes.** Narrow the search by indicating the attributes that the found item or items should have. The more attributes you specify, the more refined the results, but not all attributes apply to all item classes. For the attributes applicable to the keychain item you're updating, see the entry for the item's class in *"Item class keys and values"*.
- **Search parameters.** Condition the search in a variety of ways. For example, you can limit the results to a specific number of items, control case sensitivity when matching string attributes, or search only among a particular set of items. See "*Search attribute keys and values*" for the complete list of possible search parameters.

`attributesToUpdate`
A dictionary containing the attributes whose values should update, along with the new values. Only real keychain attributes are permitted in this dictionary (no "meta" attributes are allowed.) For the attributes applicable to the keychain item you're updating, see *"Item class keys and values"*.

Return Value

A result code. See *"Keychain Result Codes"*.

5.2.4 SecItemDelete

Deletes items that match a search query.

```
OSStatus SecItemDelete(CFDictionaryRef query);
```

Parameters

`query`
A dictionary that describes the search for the keychain items you want to delete. A typical query dictionary consists of:

- **The item's class.** Specify the kind of item you want, for example a password, a certificate, or a cryptographic key, using one of the class values in *"Item class keys and values"*.

- **Attributes.** Narrow the search by indicating the attributes that the found item or items should have. The more attributes you specify, the more refined the results, but not all attributes apply to all item classes. For the attributes applicable to the keychain item you're deleting, see *"Item class keys and values"*.

- **Search parameters.** Condition the search in a variety of ways. For example, you can limit the results to a specific number of items, control case sensitivity when matching string attributes, or search only among a particular set of items. See "*Search attribute keys and values*" for the complete list of possible search parameters.

Return Value

A result code. See *"Keychain Result Codes"*.

5.2.5 Item class keys and values

Keychain items come in a variety of classes according to the kind of data they hold, such as passwords, cryptographic keys, and certificates. The item's class dictates which attributes apply and enables the system to decide whether to encrypt the data. For example, the system encrypts passwords, but not certificates because they aren't secret.

Use the key and one of the corresponding values listed here to specify the class for a new item you create with a call to the SecItemAdd function by placing the key/value pair in the attributes dictionary.

Later, use this same pair in the query dictionary when searching for an item with one of the SecItemCopyMatching, SecItemUpdate, or SecItemDelete functions.

Item class keys

Specify the class of a keychain item

`kSecClass`
A dictionary key whose value is the item's class.

Item class values

Values you use with the kSecClass key.

`kSecClassGenericPassword`
The value that indicates a generic password item.
`kSecClassInternetPassword`
The value that indicates an Internet password item.
`kSecClassCertificate`
The value that indicates a certificate item.
`kSecClassKey`
The value that indicates a cryptographic key item.
`kSecClassIdentity`
The value that indicates an identity item.

5.2.6 Item Result Keys

When you use one of the SecItemAdd or SecItemCopyMatching functions to add or search for keychain items, these functions return the item's data and attributes through the result parameter to which you provide a pointer. Use the item result keys described below in the corresponding query dictionary to indicate how those results should be formatted:

- If you request a data reference with kSecReturnRef, the search returns a reference of type SecKeychainItemRef, SecKeyRef, SecCertificateRef, SecIdentityRef, or CFData, depending on the class of the item.

- If you request a persistent data reference using kSecReturnPersistentRef, the search returns an item reference of type CFData that you can store on disk or pass between processes. You later convert persistent references to regular references with another call to the SecItemCopyMatching function, using an array of persistent references (of the same item class) as the value for the kSecMatchItemList key.

- If you ask for the data itself with kSecReturnData, the search returns a CFData instance that holds the actual data. This is typically what you want for password items. To undo the encryption it added prior to storing the item, keychain services decrypts the data before returning it to you. Don’t use kSecReturnData for cryptographic key or identity items, as the key material may not be extractable. Instead, call SecKeyCopyExternalRepresentation, and check the error parameter if it returns nil.

- If you request the item’s attributes using kSecReturnAttributes or more than one return type, the search returns a dictionary. Item attributes are represented directly as key-value pairs in this dictionary, while the item’s data appears in one or more of the previously mentioned forms, and is associated with the appropriate item value type key from Item return result keys.

- When you specify a match limit greater than one, the search produces an array. Each element of the array is itself a search result formatted according to the previous rules.

5.2.7 Search attribute keys and values

When looking for items using any of the SecItemCopyMatching, SecItemUpdate, or SecItemDelete functions, you specify a query dictionary containing both the item attributes to look for (see *“Item class keys and values”*) and additional search attributes that condition the search. For example, you can use the matching key kSecMatchLimit with value kSecMatchLimitOne to restrict the output to include only the first result even when more than one item matches.

Item search matching keys
Keys used to condition a keychain item search.

`kSecMatchPolicy`
A key whose value indicates a policy with which a matching certificate or identity must verify.
`kSecMatchItemList`
A key whose value indicates a list of items to search.
`kSecMatchSearchList`

A key whose value indicates a list of items to search.

`kSecMatchIssuers`
A key whose value is a string to match against a certificate or identity's issuers.

`kSecMatchEmailAddressIfPresent`
A key whose value is a string to match against a certificate or identity's email address.

`kSecMatchSubjectContains`
A key whose value is a string to look for in a certificate or identity's subject.

`kSecMatchSubjectStartsWith`
A key whose value is a string to match against the beginning of a certificate or identity's subject.

`kSecMatchSubjectEndsWith`
A key whose value is a string to match against the end of a certificate or identity's subject.

`kSecMatchSubjectWholeString`
A key whose value is a string to exactly match a certificate or identity's subject.

`kSecMatchCaseInsensitive`
A key whose value is a Boolean indicating whether case-insensitive matching is performed.

`kSecMatchDiacriticInsensitive`
A key whose value is a Boolean indicating whether diacritic-insensitive matching is performed.

`kSecMatchWidthInsensitive`
A key whose value is a Boolean indicating whether width-insensitive matching is performed.

`kSecMatchTrustedOnly`
A key whose value is a Boolean indicating whether untrusted certificates should be returned.

`kSecMatchValidOnDate`
A key whose value indicates the validity date.

`kSecMatchLimit`
A key whose value indicates the match limit.

Match limit values
Keys used to limit the number of results returned.

`kSecMatchLimitOne`
A value that corresponds to matching exactly one item.

`kSecMatchLimitAll`
A value that corresponds to matching an unlimited number of items.

Additional item search keys
Keys used to specify additional keychain item search options.

`kSecUseItemList`
A key whose value is an array of items to search.

`kSecUseKeychain`
A key whose value is a keychain to operate on.

`kSecUseAuthenticationUI`
A key whose value indicates whether the user is prompted for authentication.

`kSecUseAuthenticationContext`
A key whose value indicates a local authentication context to use.

`kSecUseDataProtectionKeychain`
A key whose value indicates whether to treat macOS keychain items like iOS keychain items.

UI authentication values
Values you use to indicate whether to allow UI authentication.

`kSecUseAuthenticationUISkip`

A value that indicates items requiring user authentication should be skipped.

5.2.8 Keychain Result Codes

`errSecNotAvailable`
No trust results are available.

`errSecReadOnly`
Read-only error.

`errSecAuthFailed`
Authorization and/or authentication failed.

`errSecNoSuchKeychain`
The keychain does not exist.

`errSecInvalidKeychain`
The keychain is not valid.

`errSecDuplicateKeychain`
A keychain with the same name already exists.

`errSecDuplicateCallback`
More than one callback of the same name exists.

`errSecInvalidCallback`
The callback is not valid.

`errSecDuplicateItem`
The item already exists.

`errSecItemNotFound`
The item cannot be found.

`errSecBufferTooSmall`
The buffer is too small.

`errSecDataTooLarge`
The data is too large for the particular data type.

`errSecNoSuchAttr`
The attribute does not exist.

`errSecInvalidItemRef`
The item reference is invalid.

`errSecInvalidSearchRef`
The search reference is invalid.

`errSecNoSuchClass`
The keychain item class does not exist.

`errSecNoDefaultKeychain`
A default keychain does not exist.

`errSecInteractionNotAllowed`

Interaction with the Security Server is not allowed.

`errSecReadOnlyAttr`
The attribute is read-only.

`errSecWrongSecVersion`
The version is incorrect.

`errSecKeySizeNotAllowed`
The key size is not allowed.

`errSecNoStorageModule`
There is no storage module available.

`errSecNoCertificateModule`
There is no certificate module available.

`errSecNoPolicyModule`
There is no policy module available.

`errSecInteractionRequired`
User interaction is required.

`errSecDataNotAvailable`
The data is not available.

`errSecDataNotModifiable`
The data is not modifiable.

`errSecCreateChainFailed`
The attempt to create a certificate chain failed.

`errSecInvalidPrefsDomain`
The preference domain specified is invalid.

`errSecInDarkWake`
The user interface cannot be displayed because the system is in a dark wake state.

6 Audit Logs

macOS generates audit logs for the following events (auditreduce options specified in parenthesis):

- Start-up of the audit function (`auditreduce –m AUE_audit_startup`)
- Shut-down of the audit function (`auditreduce –m AUE_audit_shutdown`)
- Authentication events (`auditreduce –m AUE_auth_user`)
- Use of privileged/special rights events for configuration changes (`auditreduce –u '–1' –v <audit file(s)> | auditreduce –e root –m AUE_OPEN_W –m AUE_OPEN_WC –m AUE_OPEN_WTC –m AUE_OPEN_RW –m AUE_OPEN_RWC –m AUE_OPEN_RWT –m AUE_OPEN_RWTC`)
- Privilege or role escalation events:
  - Command Line (`auditreduce –u '–1' –v <audit file(s)> | auditreduce –m AUE_auth_user –e root`)
  - System Settings (`auditreduce –m AUE_ssauthorize`)
- Administrator or root-level access events (same as Privilege or role escalation events)

The audit record XML format is:

<record><argument/><path/><attribute/><subject/><text/><return/><identity/></record>

<record>
Description: Specifies the beginning and end of an audit record.
Attributes:
version – Audit trail version
event – Event name
modifier - ?
time – Event date and time
msec – Millisecond offset

<argument> (zero or more <argument> tags may exist in a record)
Description: Specifies the command line arguments.
Attributes:
arg-num – Argument position
value – Argument value
desc – Argument description

<path> (zero or more <path> tags may exist in a record)
Description: File system object path name
Attributes: None

<attribute> (zero or more <attribute> tags may exist in a record)
Description: Access control attributes.
Attributes:
mode – Unix mode bits
uid – User ID
gid – Group ID
fsid – File system ID
nodeid – File system's node ID

device – Device ID

<subject>
Description: The subject (e.g., user) triggering the audit event.
Attributes:
audit-uid – Logged in user name
uid – Effective user name
gid – Effective group name
ruid – Real user name
rgid – Real group name
pid – Process identity
sid – Session identity
tid – Thread identity

<text>
Description: Audit event description
Attributes: None

<return>
Description: The outcome of the audited event.
Attributes:
errval – Success/failure of the audited event
retval – Return value of the function (e.g., syscall)

<identity>
Description: Code-signing information of the component generating the audit event.
Attributes:
signer-type – Number indicating the signer type
signing-id – The identifier used to sign the process.
signing-id-truncated – Yes/no if the signing ID is truncated
team-id – The development team identifier used to sign the process
team-id-truncated – Yes/no if the development team identity is truncated
cdhash – Specifies the code directory hash value

praudit formats logs into text and can be configured to output multiple formats. The following example audit logs are in the XML output format. The example audit logs use usernames admin_user and regular_user.

Start-up of the audit function

```
<record version="11" event="audit startup" modifier="0" time="Tue
Jun 28 09:48:04 2022" msec=" + 130 msec" ><text>auditd::Audit
startup</text><return errval="success" retval="0" /><identity
signer-type="1" signing-id="com.apple.auditd" signing-id-
truncated="no" team-id="" team-id-truncated="no"
cdhash="0x75f7dd0ae3a70d946b6ebd0c290f14afbf0d07a8" /></record>
```

Shut-down of the audit function

```
<record version="11" event="audit shutdown" modifier="0" time="Tue
Jun 28 09:47:04 2022" msec=" + 329 msec" ><text>auditd::Audit
shutdown</text><return errval="success" retval="0" /><identity
signer-type="1" signing-id="com.apple.auditd" signing-id-
truncated="no" team-id="" team-id-truncated="no"
cdhash="0x75f7dd0ae3a70d946b6ebd0c290f14afbf0d07a8" /></record>
```

Authentication events

```
<record version="11" event="user authentication" modifier="0"
time="Tue Jun 28 15:05:05 2022" msec=" + 414 msec" ><subject audit-
uid="-1" uid="root" gid="wheel" ruid="root" rgid="wheel" pid="4353"
sid="100119" tid="9945 0.0.0.0" /><text>Verify password for record
type Users &apos;regular_user&apos; node
&apos;/Local/Default&apos;</text><return errval="success" retval="0"
/><identity signer-type="1" signing-id="com.apple.opendirectoryd"
signing-id-truncated="no" team-id="" team-id-truncated="no"
cdhash="0x11afb4e3b04698ef7c71086d1b8fd771ee6aae70" /></record>
```

Use of privileged/special rights events for configuration changes

Command Line

```
<record version="11" event="open(2) - write,creat" modifier="0"
time="Tue Jun 28 12:54:09 2022" msec=" + 324 msec" ><argument arg-
num="3" value="0x180" desc="mode" /><argument arg-num="2"
value="0x201" desc="flags" /><path>audit_control</path><subject
audit-uid="admin_user" uid="root" gid="wheel" ruid="root"
rgid="wheel" pid="2191" sid="100025" tid="50331650 0.0.0.0"
/><return errval="success" retval="4" /><identity signer-type="1"
signing-id="com.apple.vim" signing-id-truncated="no" team-id=""
team-id-truncated="no"
cdhash="0xf76dc0b2b96b2a96d69cd90566e6ade122546209" /></record>
```

System Settings

```
<record version="11" event="open(2) - read,write" modifier="0"
time="Tue Jun 28 16:03:44 2022" msec=" + 778 msec" ><argument arg-
num="2" value="0x2" desc="flags"
/><path>/dev/dtracehelper</path><path>/dev/dtracehelper</path><attri
bute mode="20666" uid="root" gid="wheel" fsid="2199351650"
nodeid="619" device="419430400" /><subject audit-uid="admin_user"
uid="admin_user" gid="staff" ruid="admin_user" rgid="staff"
pid="4853" sid="100024" tid="50331650 0.0.0.0" /><return
errval="success" retval="3" /><identity signer-type="1" signing-
id="com.apple.preference.battery.remoteservice" signing-id-
truncated="no" team-id="" team-id-truncated="no"
cdhash="0xbcf556a793acc7b8186b2f8b07702780e0103b62" /></record>
```

Privilege or role escalation events

Command Line

```
<record version="11" event="user authentication" modifier="0"
time="Tue Jun 28 15:08:19 2022" msec=" + 609 msec" ><subject audit-
uid="admin_user" uid="root" gid="staff" ruid="root" rgid="staff"
```

```
pid="4563" sid="100024" tid="10520 0.0.0.0" /><text>Verify password
for record type Users &apos;admin_user&apos; node
&apos;/Local/Default&apos;</text><return errval="success" retval="0"
/><identity signer-type="1" signing-id="com.apple.opendirectoryd"
signing-id-truncated="no" team-id="" team-id-truncated="no"
cdhash="0x11afb4e3b04698ef7c71086d1b8fd771ee6aae70" /></record>
```

System Settings

```
<record version="11" event="SecSrvr AuthEngine" modifier="0"
time="Tue Jun 28 13:03:45 2022" msec=" + 522 msec" ><subject audit-
uid="admin_user" uid="admin_user" gid="staff" ruid="admin_user"
rgid="staff" pid="1124" sid="100025" tid="2640 0.0.0.0"
/><text>com.alf</text><text>client
/usr/libexec/ApplicationFirewall/Firewall</text><text>creator
/usr/libexec/ApplicationFirewall/Firewall</text><return
errval="success" retval="0" /><identity signer-type="1" signing-
id="com.apple.authd" signing-id-truncated="no" team-id="" team-id-
truncated="no" cdhash="0x47225d901be7b10ddad43577d7614fdf892824f6"
/></record>
```

Administrator or root-level access events

Same events as Privilege or role escalation events above.

Bluetooth: Failed user authorization of Bluetooth device

```
2023-04-04 14:49:20.300707-0500 0x6315     Default 0x9ddc
225    0    bluetoothd: (CoreUtils)
[com.apple.bluetooth:CBStackController] Pairing agent event:
AttemptComplete, device 00:01:02:03:04:05, result 310161/0x4BB91
BT_ERROR_PAIRING_CANCELLED
2023-04-04 14:49:20.315891-0500 0x4ac4     Default 0x9ddc
225    0    bluetoothd: (CoreUtils)
[com.apple.bluetooth:CBDaemonXPCConnection] CBConnection pairing
continue: CBConnection: Peer E9D82898-309E-C18C-25B9-5267BF5E2ABE,
CF 0x12 < WaitConnectedServices UserInitiated >, SF 0xFFFFD7BF < HFP
PhoneBook GPS AVRCP A2DP HID WirelessiAP NetSharing MAP Passthrough
NetworkConsumer PassiveMultiStream LEGATT LEA WirelessiAPSink
CarPlay AACP GATT SerialPort BLE ACL SCO >, CnTO 30.000 secs,
CBPairingInfo: device CBDevice E9D82898-309E-C18C-25B9-5267BF5E2ABE,
BDA 00:01:02:03:04:05, Nm 'linux-system', PID 0x0246 (?), VID
0x1D6B, VS 2, DsFl 0x4000000000000 < ClassicScan >, DvF 0x11000 <
Hidden HIDGoodBehavior >, DvT LaptopComputer, RSSI -58, SupS
0x100000 < GATT >, ClkH L NoiseManagement R NoiseManagement, ECC 2,
Color 0, FV '5.4.2', LsnM Normal, LsMC 0x6 < ANC Transparency >,
MicM Auto, Plcm M Enabled, srMd Disabled, spAM ContentDriven, CF
0x4080A00200000 < Connections RSSI DiscoveryFlags Attributes
ClassicScan >, PIN NULL, Flags 0x0 < >, Type JustWorks, Error
kCanceledErr 'User canceled pairing', kNoErr
```

Bluetooth: Failed user authorization for local Bluetooth Service

```
Same as above. The devices authenticate and authorize at the same
time.

Bluetooth: Initiation of Bluetooth connection
default    2023-04-04 15:09:25.684880 -0500    bluetoothd    ACL
connected aclConnectCfm for device linux-system status 0
default    2023-04-04 15:09:25.684937 -0500    bluetoothd    ACL
connected aclConnectCfm for device linux-system status 0 incoming 0
default    2023-04-04 15:09:25.684962 -0500    bluetoothd    ACL
connected: 00:01:02:03:04:05, result 0

Bluetooth: Failure of Bluetooth connection
2023-04-04 14:26:04.388260-0500 0x8d4      Default 0x0
225    0    bluetoothd: [com.apple.bluetooth:Stack.SDP]
00:01:02:03:04:05 disconnected with reason  STATUS 722
2023-04-04 14:49:23.310594-0500 0x8d4      Default 0x0
225    0    bluetoothd: [com.apple.bluetooth:Stack.SDP]
00:01:02:03:04:05 disconnected with reason  STATUS 734
```

6.1 Enabling Bluetooth Logging

The TOE generates Bluetooth audit records after the "Bluetooth for macOS" configuration profile is installed on the TOE. This configuration profile is obtainable from the Apple Developer website under "Profiles and Logs » macOS" along with instructions. The online Apple Developer documentation on “Profiles and Logs” for macOS provides instructions on how to enable Bluetooth (and other types of) logging and provides this configuration profile. The configuration profile requires the use of the Apple Configurator app, which is a free download in the Apple Store. The following is the “Profiles and Logs” URL for macOS:
https://developer.apple.com/bug-reporting/profiles-and-logs/?platform=macos

The administrator can use the macOS sysdiagnose feature to obtain these records in the form of log files. The /private/var/tmp folder will contain the log files.

The instructions for enabling logging is as follows:

1. Download the logging profile and install it
2. Click Install when prompted.
3. Enter your administrator password, if prompted.
4. Reboot system.
5. Trigger a sysdiagnose (See below).

Triggering a sysdiagnose from the computer keyboard:

1. Briefly press the following keys simultaneously to trigger a sysdiagnose from the Finder: Command + Option + Shift + Control + Period (.)
   Note: The sysdiagnose process can take 10 minutes to complete. Once finished, the folder “/private/var/tmp/“ should appear automatically in the Finder and the sysdiagnose file there will look similar to this:
   “sysdiagnose_2017.08.17_07-30-12-0700_10169.tar.gz”

2. The sysdiagnose file should appear in /private/var/tmp.

Trigger a sysdiagnose from Terminal:

1. Launch Terminal (/Applications/Utilities/Terminal.app).
2. Enter this command, followed by the return key, at the Terminal command prompt:
```
sudo sysdiagnose
```
3. Enter your administrator password when prompted (for sudo access)
4. Press return again (or enter) to proceed with the sysdiagnose capture.
5. The syslogdiagnose should appear in /private/var/tmp/.

7 Acronyms

Table 3 – Acronyms

Acronym	Definition
ACL	Access Control List
AES	Advanced Encryption Standard
API	Application Programming Interface
ASLR	Address Space Layout Randomization
BSD	Berkeley Software Distribution
BSM	Basic Security Module
CA	Certificate Authority
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CC	Common Criteria
CRL	Certificate Revocation List
CTR	Counter
CVE	Common Vulnerabilities and Exposures
DRBG	Deterministic Random Bit Generator
DSS	Digital Signature Standard
ECC	Elliptic-Curve Cryptography
ECDSA	Elliptic-Curve Digital Signature Algorithm
ECDH	Elliptic-Curve Diffie-Hellman
EKU	Extended Key Usage
FIPS	Federal Information Processing Standards
GCM	Galois/Counter Mode
GPOS	General Purpose Operating System
HMAC	Keyed-Hash Message Authentication Code
HTTPS	Hypertext Transfer Protocol Secure
IP	Internet Protocol
KEK	Key Encrypting Key
MDM	Mobile Device Management
NIAP	Nation Information Assurance Partnership
NIST	National Institute of Standards and Technology
OCSP	Online Certificate Status Protocol
OID	Object Identifier
OS	Operating System
OSP	Organizational Security Policy
PKCS	Public-Key Cryptography Standards
POSIX	Portable Operating System Interface
PP	Protection Profile
RFC	Request for Comments

Acronym	Definition
RSA	Rivest, Shamir, & Adleman
RSR	Rapid Security Response
SEP	Secure Enclave Processor
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SP	Special Publication
ST	Security Target
TD	Technical Decision
TOE	Target of Evaluation
TLS	Transport Layer Security
USB	Universal Serial Bus
VID	Validation Identifier

End of Document