Apple Inc.

Apple macOS 26 Tahoe Common Criteria Guide

Version 1.0

2026-07-02

NIAP VID 11695

Prepared for:

Apple
One Apple Park Way
Cupertino, CA 95014

Prepared by:

atsec information security
4516 Seton Center Parkway, Suite 250
Austin, TX 78759

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

Revision History

Version	Date	Changes
1.0	2026-07-02	First version

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.

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

Table of Contents

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

List of Tables

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

1 Introduction

This guide provides instructions to configure and operate Apple macOS 26 Tahoe in the Common Criteria (CC) evaluated configuration.

1.1 Target of Evaluation

The Target of Evaluation (TOE) is Apple macOS 26 Tahoe, which is a general purpose operating system running on the Apple Mac computers with Apple silicon. The TOE hardware platforms covered by this evaluation are listed in the Security Target (ST). The tested version of the TOE is macOS 26.3.

The TOE is a Unix-based operating system built on top of the XNU kernel. The TOE implements standard Unix facilities, provides both command-line and graphical user interfaces, supports Bluetooth communication, and includes Wireless Local Area Network (WLAN) client functionality.

The Mac computers covered in this evaluation contain the Secure Enclave, a dedicated secure subsystem integrated into the Apple silicon. Utilizing a dedicated processor (Secure Enclave Processor or SEP), the Secure Enclave is isolated from the main processor to provide an extra layer of security designed to keep sensitive data secure. The SEP executes the SEP Operating System (sepOS). sepOS is included with macOS and is within the TOE boundary.

1.2 Document Purpose and Scope

This document serves as the administrative guidance for the TOE, Apple macOS 26 Tahoe. This document provides the preparative procedures and operational user guidance for the evaluated configuration. This document describes how to install, configure, and operate the TOE in the CC compliant manner.

The TOE conforms to the following Protection Profiles (PPs) and Functional Packages:

- PP-Configuration for General Purpose Operating System, Bluetooth, and Wireless Local Area Network Clients, Version 1.0 (CFG_GPOS_BT_WLANC_V1.0).
  This PP-Configuration includes the following components:
  - Base-PP: Protection Profile for General Purpose Operating Systems, Version 4.3 (PP_OS_V4.3);
  - PP-Module: PP-Module for Bluetooth, Version 1.0 (MOD_BT_V1.0); and
  - PP-Module: PP-Module for WLAN Clients, Version 1.0 (MOD_WLANC_V1.0).
- Functional Package for Transport Layer Security (TLS), Version 1.1 (PKG_TLS_V1.1).

1.3 Evaluated Configuration

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

- VPN: Virtual Private Network (VPN) technology 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 following assumptions apply when operating the TOE in the evaluated configuration:

- The OS relies upon a trustworthy computing platform for its execution. This underlying platform is out of scope of PP_OS_V4.3.

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

- The user of the OS is not willfully negligent or hostile, and uses the software in compliance with the applied enterprise security policy. At the same time, malicious software could act as the user, so requirements which confine malicious subjects are still in scope.
- The administrator of the OS is not careless, willfully negligent or hostile, and administers the OS within compliance of the applied enterprise security policy.
- Information cannot flow between the wireless client and the internal wired network without passing through the TOE.
- TOE administrators are trusted to follow and apply all administrator guidance in a trusted manner.

1.5 Common User Interface Features

There are multiple user interface terms that are important to know while using this document:

- "Apple menu" refers to the pull-down menu located in the top-left corner of the screen.
- "System Settings" is an app that provides access to and control of settings in macOS. It is a selection under the Apple menu.
- To help illustrate hierarchical menu structures, this document uses '>' to indicate menu levels. For example, to instruct the reader to click on the Apple menu and select the System Settings menu item, this document will say:

  Choose Apple menu > System Settings
- "Dock" refers to the bar normally positioned at the bottom of the screen that contains multiple application icons. As the cursor is moved across each icon in the Dock, the icon's descriptive name will appear above the icon. Below are some commonly used icons.

Table 1 - Icons

Icon	Usage
Launchpad 	By default, the Dock contains the Launchpad icon. Launchpad provides users 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 	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 and execute commands. When this document refers to command-line commands, it implies that the user must open the Terminal app and type the command into the window created by the app. Command-line commands are written on a new line and prefixed by % in this document.
System Settings 	Clicking this icon provides an alternate method to launch the System Settings app.

1.6 Obtaining TOE Hardware Platform

Customers can acquire the TOE hardware platforms (Mac computers) through various distribution channels:

- **Normal distribution channel**

  Individual customers can obtain Mac computers from the following sources:
  - The Apple Store (physical store or online at https://apple.com)

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

  - Apple retailers
  - Resellers
- **Business-specific distribution channel**

  Business customers can use the link https://www.apple.com/business/
- **Government-specific distribution channel**

  Government customers can use the link https://www.apple.com/government/
- **Apple Store Catalog for large customers**

  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

The Apple corecrypto cryptographic libraries and the Broadcom chip are the only cryptographic providers associated with the evaluated TOE configuration. The Broadcom chip, which is integrated into the TOE hardware platforms, implements the AES-CCMP-256 and AES-GCMP-256 algorithms to support the wireless LAN functionality and the AES-CCM algorithm to support the Bluetooth functionality. The Apple corecrypto cryptographic libraries support all other claimed cryptographic algorithms. These algorithms are supported by default so no further configuration is required.

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

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

2 Installation, Update, and Recovery

Apple macOS 26 Tahoe may come pre-installed on the hardware platforms. The TOE has a built-in update feature that the user can leverage to check for and install updates. Should the need arise, an administrator can download and re-install the same version of the TOE software/firmware on the supporting hardware.

The TOE recognizes two kinds of user roles: Administrator and Standard. The first user account created is automatically assigned the Administrator role.

To determine the running macOS version, a user can choose Apple menu > About This Mac.

2.1 Installation

If the hardware platform was purchased prior to the release of Apple macOS 26 Tahoe, an administrator can install the TOE on the hardware platform as follows:

1. Open Software Update.
2. When Software Update opens, it automatically checks for new software. If new software is available for your Mac (i.e., macOS 26.3), click the Update, Upgrade, or Restart Now button to install it.
3. Before installation begins, you're asked to enter your administrator password.
4. During installation, your Mac might restart and show a progress bar or blank screen several times.

Alternatively, an administrator can download and install macOS 26 Tahoe via the command-line interface:

1. Open the Terminal app.
2. Enter the following command to download the installer:

   ```
   % softwareupdate --fetch-full-installer --full-installer-version 26.3
   ```

3. As macOS downloads to the Applications folder, Terminal shows the percentage installed (downloaded).
4. Open the Applications folder and double-click the macOS installer, named *Install macOS Tahoe.app*, then follow the onscreen installation instructions.

2.2 Update

An administrator can check for and install updates to the TOE as follows:

1. Choose Apple menu > System Settings, click General in the sidebar, then click Software Update.
2. The update information, if available, will be shown.
3. Follow the onscreen instructions for installation of the updates.

Signature verification of the update is performed by the TOE before the update is installed. If the signature verification fails, the update process is aborted.

2.2.1 Background Security Improvements

Starting with macOS 26.1, macOS supports Background Security Improvements feature. This feature allows for delivering important security improvements between software updates. It can be enabled/disabled under System Settings > Privacy & Security > Background Security Improvements.

2.3 Recovery

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

If macOS encounters a problem (e.g., a system integrity error), it will boot into the paired macOS hidden Recovery partition installed and updated along with each macOS update. macOS Recovery will prompt the user for an administrator's credentials and attempt to repair the problem. If macOS Recovery is able to repair the problem, macOS will boot normally.

If the repairs are unsuccessful, an administrator can use macOS Recovery to reinstall macOS, which keeps files and settings intact when reinstalling.

1. Make sure the computer is connected to the Internet.
2. Choose Apple menu > Shut Down, press and hold the power button until the system volume and the Options button appear, select Options, click Continue.
3. In the Recovery app window, select "Reinstall macOS Tahoe," then click Continue.
4. Follow the onscreen instructions.
5. When you're asked to select a disk, select your current macOS disk (in most cases, it's the only one available).

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

3 Configuration and Management

3.1 Excluded Functionality

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

1. Choose Apple menu > System Settings, then click VPN in the sidebar.
2. If there are VPN configurations shown on the right, uncheck the VPN services.

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

1. Choose Apple menu > System Settings, then click Apple Intelligence & Siri (or Siri) in the sidebar.
2. Uncheck Siri.

In the evaluated configuration, incoming connections must be blocked for the symptomsd process (/usr/libexec/symptomsd). Instructions to configure the firewall are provided in Section 3.7.

3.2 User Accounts

3.2.1 Add Users

As an administrator:

1. Choose Apple menu > System Settings, then click Users & Groups in the sidebar.
2. Click the Add User button below the list of users.
3. Click the pop-up menu next to New User, then choose a type of user.
    - Administrator: An administrator can add and manage other users, install apps, and change settings. The new user you create when you first set up your Mac is an administrator. Your Mac can have multiple administrators.
    - Standard: Standard users are set up by an administrator. Standard users can install apps and change their own settings, but can’t add other users or change other users’ settings.
    - Sharing Only: Sharing-only users can access shared files remotely, but can’t log in to the computer or change settings. Note: This user type is not evaluated and should not be used in the evaluated configuration.
4. Enter a full name for the new user. An account name is generated automatically. To use a different account name, enter it now—you can’t change it later.
5. Enter a password for the user, then enter it again to verify.
6. Click Create User.

3.2.2 Delete Users

As an administrator:

1. Choose Apple menu > System Settings, then click Users & Groups in the sidebar.
2. Click the information icon ⓘ (the letter i in a circle) next to the user to be deleted, then click Delete User.
3. Choose what to do with the user’s home folder, then click Delete User.

3.2.3 Configure Roles

Administrator privileges can be added or removed from existing users. As an administrator:

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

1. Choose Apple menu > System Settings, then click Users & Groups in the sidebar.
2. Find the existing user and click the information icon ⓘ next to the user.
3. Select “Allow user to administer this computer” to assign the user the Administrator role. Unselect that option to assign the user the Standard user role.

3.2.4 Initial User Authentication

Initial user authentication is slightly different depending on if Apple macOS 26 Tahoe is installed for the first time or macOS is updated from an earlier version to the evaluated version.

If this is the first time that the TOE is running on a Mac computer, after initial setup process (described in section 2.1), the user is required to create a new user account. This is the same as the steps described in section 3.2.1. Note that this user account is an Administrator account.

In case that the user updates macOS to the evaluated version as described in section 2.2, after the update process is completed, the user is required to enter the current account credentials to authenticate to the TOE.

3.3 Smart Card Registration

To use a smart card on macOS, follow these steps to pair it with the local user account:

1. Insert a Personal Identity Verification (PIV) smart card that includes authentication and encryption identities into the card reader attached to the Mac.
2. Select Pair at the notification dialog.
3. Provide administrator account credentials (user name/password).
4. Enter a four- to eight-digit personal identification number (PIN) for the inserted smart card.
5. Log out and use the smart card and PIN to log back in.

When you insert the smart card to authenticate, enter the associated PIN to unlock the card.

3.4 Password Policy

To change the password policies, open the Terminal app. As an administrator, run

```
% sudo pwpolicy -setaccountpolicies <policy_plist>
```

where <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>
    <key>policyCategoryAuthentication</key>
    <array> [passwordAuthenticationPolicies] </array>
</dict>
</plist>
```

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

- Minimum Password Length:

```
<dict>
    <key>policyContent</key>
```


© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

```
        <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>
```

- Minimum Number of Numeric Characters:

```
    <dict>
        <key>policyContent</key>
        <string>policyAttributePassword matches '(.*[0-9].*){[number_of_characters],}'</string>
    </dict>
```

- Minimum Number of Uppercase Characters:

```
    <dict>
        <key>policyContent</key>
        <string>policyAttributePassword matches '(.*[A-Z].*){[number_of_characters],}'</string>
    </dict>
```

- Minimum Number of Lowercase Characters:

```
    <dict>
        <key>policyContent</key>
        <string>policyAttributePassword matches '(.*[a-z].*){[number_of_characters],}'</string>
    </dict>
```

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

- Maximum Failed Login Attempts:

```
    <dict>
        <key>policyContent</key>
        <string>policyAttributeFailedAuthentications &lt; [number_of_failures]</string>
    </dict>
```

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

3.5 Screen Lock

An administrator can configure the TOE to prevent Standard users from disabling or changing screen lock settings by deploying a Configuration Profile with the following settings:

- PayLoadType - com.apple.screensaver
- idleTime – Sets the screen lock idle time (in seconds) before activation.
- askForPassword – Forces password prompt on wake. Must be set to true to require a password when the screen lock is activated.
- askForPasswordDelay – Time delay before requiring password after screen lock starts (in seconds). Set to 0 to require the password immediately.

Example Configuration Profile:

```
<?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>PayloadContent</key>
```


© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

```
    <array>
        <dict>
            <key>PayloadDescription</key>
            <string>{human descriptive string}</string>
            <key>PayloadDisplayName</key>
            <string>{human descriptive string}</string>
            <key>PayloadIdentifier</key>
            <string>{unique identification string}</string>
            <key>PayloadType</key>
            <string>com.apple.screensaver</string>
            <key>PayloadUUID</key>
            <string>{unique UUID}</string>
            <key>PayloadVersion</key>
            <integer>1</integer>
            <key>askForPassword</key>
            <true />
            <key>askForPasswordDelay</key>
            <integer>0</integer>
            <key>idleTime</key>
            <integer>60</integer>
        </dict>
    </array>
    <key>PayloadDescription</key>
    <string>{human descriptive string}</string>
    <key>PayloadDisplayName</key>
    <string>{human descriptive string}</string>
    <key>PayloadIdentifier</key>
    <string>{unique identification string}</string>
    <key>PayloadScope</key>
    <string>System</string>
    <key>PayloadType</key>
    <string>Configuration</string>
    <key>PayloadUUID</key>
    <string>{unique UUID}</string>
    <key>PayloadVersion</key>
    <integer>1</integer>
    <key>TargetDeviceType</key>
    <integer>5</integer>
</dict>
</plist>
```

Note that in the above, the “system” scope must be targeted to ensure it applies to all users.

3.6 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. Execute the following commands:

```
% chmod 644 <filename>
```


© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

```
% diskutil apfs updatePreboot /
```

3. Restart macOS for the changes to take effect.

3.7 Firewall

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

1. Choose Apple menu > System Settings, click Network in the sidebar, then click Firewall.
2. Use the Firewall switch to enable and disable the firewall.
3. Click Options to configure specific application-based firewall rules.
    - An application can be added to the firewall rules by clicking “+”. A path can be entered manually by hitting Shift-Command-G.
4. Click OK to save your changes.

3.8 Audit

The TOE does not generate security audits by default using auditd. Users need to enable and configure the auditing rules to start the audit function.

3.8.1 Rules

No rules are required to maintain the Apple Unified Logs.

To edit audit rules for auditd, as an administrator:

1. Edit the /etc/security/audit_control file. E.g.,

    ```
    % sudo nano /etc/security/audit_control
    ```

2. Add/edit lines in the following format (spaces between the parameter, the colon, and the value are not permitted): `parameter:value`
3. Restart macOS for the changes to take effect.

To configure the audit system to generate required audit records:

1. Run the following command

    ```
    % sudo launchctl enable system/com.apple.auditd
    ```

2. Open the audit_control file:

    ```
    % sudo nano /etc/security/audit_control
    ```

3. Set the dir parameter to specify the directory where audit log files are stored:

    ```
    dir:/var/audit
    ```

4. Configure the flags parameter to specify which audit event classes are audited for all users.

    ```
    flags:lo,aa,ad,fw,fd,fc,ex,io
    ```

    Where:

    - lo for login/logout events
    - aa for authentication and authorization events
    - ad for administrative events
    - fw, fd, fc for file write, delete, and create events, respectively
    - ex for execve events
    - io for ioctl events

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

5. Set the naflags parameter to define what classes of events are audited when an action cannot be attributed to a specific user:

   naflags:lo,aa,ad,fw,fd,fc,ex,io

6. Configure the minfree parameter to set the minimum free space (in percent of overall storage space) required on the file system where audit logs are being written:

   minfree:20

7. Set the policy parameter to specify global audit policy flags. At a minimum, argv is required. Based on your requirements, you might want to include:

   policy:argv,arge,trail

   Where:

   - argv to audit command line arguments
   - arge to audit environmental variable arguments
   - trail to append a trailer token to each audit record

8. Set the filesz parameter to specify the maximum trail size in bytes:

   filesz:5M

9. Configure the expire-after parameter to specify when audit log files will expire and be removed:

   expire-after:7d OR 500M

10. Save and close the file.
11. Restart macOS for the changes to take effect.
12. After the system has restarted, check if auditd is enabled:

    % sudo launchctl list | grep auditd

3.8.2 auditd Review

macOS includes the auditreduce and praudit command-line utilities for the administrator to review local auditd audit logs.

auditreduce selects records from audit trail files. The auditreduce command outputs data in a binary format. Typically, its output should be piped to praudit and the praudit command prints the contents of the audit trail.

3.8.3 Apple Unified Log Review

macOS includes the log command-line utility for the administrator to review the local Apple Unified Logging audit logs. This tool is invoked as follows:

% log command [options]

The command can be any of the following:

- show: output previously stored logs.
- stream: emit logs to the console as they occur (note that logs can be generated at a rate far exceeding the ability to post to the console, and therefore this is not recommended).

3.9 Network Time Server

To configure the name or address of the network time server, an administrator can perform the following steps:

1. Choose Apple menu > System Settings, click General in the sidebar, then click Date & Time.
2. Enable “Set time and date automatically”.

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

3. Click the Set button, then type the address (DNS name or IP address) of the network time server in the “Time Server” field.

3.10 Automatic Software Updates

To configure automatic downloading and installation of software updates, an administrator can perform the following steps:

1. Choose Apple menu > System Settings, click General in the sidebar, then click Software Update.
2. Click the information icon ⓘ next to Automatic Updates, then enable/disable the following options:
   - Download new updates when available: Automatically download macOS updates without asking.
   - Install macOS updates: Install macOS updates automatically.
   - Install system data files and security updates: Automatically install system data files and security updates.
3. Click Done.
4. Choose Apple menu > System Settings. Then click Privacy & Security.
5. Go to Background Security Improvements.
6. Enable/disable the “Automatically Install” option.

3.11 Wi-Fi

3.11.1 Enable/Disable Wi-Fi

As a user:

1. Choose Apple menu > System Settings, then click Wi-Fi in the sidebar.
2. Use the Wi-Fi switch to enable and disable Wi-Fi.

3.11.2 Join Networks

As a user:

1. Choose Apple menu > System Settings, then click Wi-Fi in the sidebar.
2. Select the Wi-Fi network name.

If a Wi-Fi connection would be unintentionally broken, the user can reconnect to the Wi-Fi network by following the same steps. If the auto-join option for the network is turned on, macOS will automatically connect when it's available.

3.11.3 Restrict Administrative Access to Wireless Interface

To prevent Standard users from configuring the wireless interface, an administrator can perform the following steps:

1. Choose Apple menu > System Settings, then click Wi-Fi in the sidebar.
2. Click the Advanced button below the list of Wi-Fi networks.
3. In the pop-up window, select “Change networks” and “Turn Wi-Fi on or off”.
4. If needed, set the auto-join option for a network from the “Known Networks” list in the pop-up window:
   - Click the More button (an ellipsis in a circle) next to the network you want to set to auto-join, then choose Auto-Join.

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

3.11.4 Enable/Disable Wireless Network Bridging

To share the Internet connection on the Mac with other devices on the local network, an administrator can perform the following steps:

1. Choose Apple menu > System Settings, click General in the sidebar, then click Sharing.
2. Turn on Internet Sharing, then click Configure.
3. Click the “Share your connection from” pop-up menu, then choose the Internet connection you want to share. For example, if you’re connected to the Internet over Ethernet, choose Ethernet.
4. Below “To devices using,” turn on the port other devices can use to access the shared Internet connection. For example, if you want to share your Internet connection over Wi-Fi, select Wi-Fi.
5. If you’re sharing to devices using Wi-Fi, configure the Internet-sharing network, then click OK.
    - Network Name: Enter a name for the shared connection.
    - Channel: If you don’t want to use the default channel, click the Channel pop-up menu, then choose another channel.
    - Security: If available, click the Security pop-up menu, then choose an option.
        - Choose WPA3 Personal if all computers that use the shared connection support WPA3.
        - Choose WPA2/WPA3 Personal if some computers that use the shared connection support only WPA2.
    - Password: Enter a password. If you want to see the current password, select the “Show password” checkbox below the password.
6. Click Done.

3.11.5 Enable/Disable AirDrop

To enable/disable AirDrop for sharing files between the Mac and other devices on the local network, an administrator can perform the following steps:

1. Choose Apple menu > System Settings, click General in the sidebar, then click AirDrop & Handoff.
2. Turn AirDrop on or off:
    - To turn on, click the arrow next to AirDrop, then select "Contacts Only" or "Everyone".
    - To turn off, click the arrow next to AirDrop, then select "No One".

Note: To ensure a successful transfer, both the Mac and the other device should have Wi-Fi and Bluetooth enabled and be within 10 meters of each other.

3.12 Bluetooth

3.12.1 Enable/Disable Bluetooth

As a user:

1. Choose Apple menu > System Settings, then click Bluetooth in the sidebar.
2. Use the Bluetooth switch to turn Bluetooth on or off.

3.12.2 Connect/Pair

Ensure the other Bluetooth device is in discoverable mode.

As a user:

1. Choose Apple menu > System Settings, then click Bluetooth in the sidebar.

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

2. Ensure Bluetooth is enabled.
3. The TOE displays a list of nearby Bluetooth devices. Hold the pointer over the device in the list, then click Connect.
4. If prompted, click Accept or enter the pairing code, then press Enter.

3.13 Configuration Profiles

A configuration profile allows an administrator to consistently set configuration options for devices based on organizational and business needs. Depending on the specific configuration setting enforced, this may restrict the ability for non-administrative users to adjust the setting.

Configuration profiles are structured in XML format and follow Apple's Property List (plist) specification. Each configuration profile can include one or more payloads, which define specific settings for a particular feature or service. The scope of a configuration profile can range from device-level configurations to user-specific settings, depending on how it is deployed. Within the XML, the individual settings are grouped into categories (often referred to as “payload types”), such as com.apple.system.logging for audit settings or com.apple.screensaver for screen lock settings. Each category contains its own set of keys and values that control how the feature behaves.

Multiple profiles can be independently installed in order to permit modular configuration as needed. Configuration profile settings that are pertinent to the scope of the TOE are described in their relevant sections.

Of note, each configuration profile must have a unique identifier provided in the PayloadIdentifier key/value as well as a unique UUID in the PayloadUUID key/value. These are used to help track individual profiles and to track updates and resolve conflicts. Payloads can be versioned using the PayloadVersion key/value. By default, these should start at 1. Payloads should target macOS using a key TargetDeviceType and value of “5”. Optionally, they can target the system by using the key PayloadScope with “system” (otherwise they apply only to the user who loads it).

To manage configuration profiles, an administrator can perform the following steps:

1. Choose Apple menu > System Settings, click General in the sidebar, then click Device Management.
2. Perform the following steps as needed:
   - To install a configuration profile, click the “+” button, then follow the onscreen instructions.
   - To remove a configuration profile, select the configuration profile from the list, then click the “-” button.
   - To view an installed configuration profile, double-click the profile to view information about it.

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

4 Secure Communications

The TOE supports Transport Layer Security version 1.2 (TLS 1.2) for secure communication and TLS 1.1 and TLS 1.2 for WLAN. No additional configuration is required to ensure proper usage of TLS.

The TOE enforces encryption when transmitting data over Bluetooth and terminates the connection if the connected device stops encrypting. The Bluetooth data communication between the TOE and Bluetooth devices is always encrypted using 128-bit AES Counter with CBC-MAC (AES-CCM-128). No additional configuration is needed for Bluetooth.

- The encryption algorithm is not configurable.
- The encryption key size for BR/EDR and LE encryption is not configurable.

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

5 Storage of Credentials

The TOE offers an encrypted database, called keychain, to securely store the following sensitive data:

- Trusted certificate authorities for establishing TLS sessions.
- Private keys for establishing TLS sessions.
- Bluetooth Link Keys to manage the pairing and secure communication with Bluetooth devices.
- Wi-Fi passwords to authenticate the TOE itself when connecting to wireless access points.

Keychain items are encrypted. No configuration is necessary. The Key Encryption Key used to protect Keychain keys is stored in the Secure Enclave, which is dedicated security hardware, so there are no instances where key destruction may be delayed.

Users can access the keychain by opening the Keychain Access app in /Applications/Utilities/. 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. The developer documentation on the API is available at

https://developer.apple.com/documentation/security/keychain-services

5.1 Digital Certificates

The Keychain is used to manage X.509v3 Certification Authority (CA) certificates used to validate the identity of TLS servers, EAP-TLS servers, 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.

To load an X.509v3 certificate into the Keychain (e.g., a client-side certificate for TLS mutual authentication), an administrator can perform the following steps:

1. Locate and open the Keychain Access app in /Applications/Utilities/.
2. Select the login keychain for client-side certificates or the System keychain for CA certificates.
3. Drag the certificate file onto the Keychain Access app.
4. If importing a CA certificate, there is an additional step to change the trust settings:
    - Double-click the certificate. Next to Trust, click the arrow to set the certificate to be "Always Trust".

To remove a certificate from the Keychain, an administrator can perform the following steps:

1. Locate and open the Keychain Access app in /Applications/Utilities/.
2. Select the keychain where the certificate is stored, then locate the certificate you wish to remove.
3. Right-click on the selected item and choose "Delete" from the context menu.

Each certificate stored in the keychain is associated with a trust policy, which determines whether the certificate is considered trustworthy and defines the conditions under which it may be used.

To change the trust settings of a certificate, an administrator can perform the following steps:

1. Locate and open the Keychain Access app in /Applications/Utilities/.
2. Select the keychain where the certificate is stored, then double-click a certificate.
3. Next to Trust, click the arrow to display the trust policies for the certificate.

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

4. To override the trust policies, choose new trust settings from the pop-up menus.

5.2 Keychain Services

The contents of this section are taken from the Apple Developer Documentation on the Keychain Services API. The following API functions pertaining to the Keychain Services 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 *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 the entry for the item's class in "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 Return 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 * result);
```

Parameters

query

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

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

- **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 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.
- **One or more return types.** Use the keys found in "Item Return 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 Return 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 the entry for the item's class in "Item Class Keys and Values".

Return Value

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

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 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.

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

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

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 Return 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.

Item result keys

Keys you use to specify the type of results to return from a keychain item search or add operation.

kSecReturnData

A key whose value is a Boolean that indicates whether or not to return item data.

kSecReturnAttributes

A key whose value is a Boolean indicating whether or not to return item attributes.

kSecReturnRef

A key whose value is a Boolean indicating whether or not to return a reference to an item.

kSecReturnPersistentRef

A key whose value is a Boolean indicating whether or not to return a persistent reference to an item.

Item value type keys

Keys that appear in the result dictionary when you specify more than one search result key.

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

kSecValueData

A key whose value is the item’s data.

kSecValueRef

A key whose value is a reference to the item.

kSecValuePersistentRef

A key whose value is a persistent reference to the item.

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 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.

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

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.

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.

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

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

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

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.

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

6 Audit Logs

6.1 Operating System Logs

The TOE generates audit records 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
    - Security Changes (auditreduce -m AUE_modify_password)
    - Audit Changes (auditreduce -m AUE_OPEN_WC -m AUE_auth_user)
    - Configuration Changes (auditreduce -m AUE_ssauthorize)
- Privilege or role escalation events:
    - Command Line (auditreduce -m AUE_auth_user)
    - System Settings (auditreduce -m AUE_ssauthorize)
- Administrator or root-level access events (same as Privilege or role escalation events)

The praudit command is used to make the binary output of the auditreduce command readable. The praudit command reads audit records in binary format from standard input and displays the records in multiple formats, such as XML.

The audit record XML format is:

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

The following table describes the audit record XML tags.

Table 2 – Audit Record XML Tags

Field	Description	Attributes
<record>	Specifies the beginning and end of an audit record.	• version – audit trail version • event – event name • modifier – specify additional flags or options • time – event date and time • msec – millisecond offset
<argument> (zero or more <argument> tags may exist in a record)	Specifies the command line arguments	• arg-num – argument position • value – argument value • desc – argument description
<path> (zero or more <path> tags may exist in a record)	Filesystem object path name	None

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

Field	Description	Attributes
<attribute> (zero or more <attribute> tags may exist in a record)	Access control attributes	• mode – Unix mode bits • uid – user ID • gid – group ID • fsid – filesystem ID • modeid – filesystem's node ID • device – device ID
<subject>	The subject (e.g., user) triggering the audit event	• audit-uid – logged in user name • uid – effective user name • gid – effective group name • ruid – real user name • rgid – real group name • sid – session identity • tid – thread identity
<text>	Audit event description	None
<return>	Outcome of the audited event	• errval – success/failure of the audited event • retval – return value of the function (e.g., syscall)
<identity>	Code-signing information of the component generating the audit event	• 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

The following example audit records are in the XML format.

Start-up of the audit function

```
<record version="11" event="audit startup" modifier="0" time="Fri Sep 19 13:46:35 2025" msec=" + 620 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="0xa74b562a95b98b78bda61e068fbaff1cc1bb90ce" />
</record>
```

Shut-down of the audit function

```
<record version="11" event="audit shutdown" modifier="0" time="Fri Sep 19 13:46:39 2025" msec=" + 196 msec" >
<text>auditd::Audit shutdown</text>
<return errval="success" retval="0" />
```


© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

<identity signer-type="1" signing-id="com.apple.auditd" signing-id-truncated="no" team-id="" team-id-truncated="no" cdhash="0xa74b562a95b98b78bda61e068fbaff1cc1bb90ce" />
</record>

Authentication events

<record version="11" event="user authentication" modifier="0" time="Fri Sep 19 13:49:54 2025" msec=" + 197 msec" >
<subject audit-uid="atsec" uid="root" gid="staff" ruid="atsec" rgid="staff" pid="1354" sid="100004" tid="3702 0.0.0.0" />
<text>Verify password for record type Users 'atsec' node '/Local/Default'</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="0x4a21fe7d1588f7dea2909b5d490d45dbd6d61ed7" />
</record>

<record version="11" event="user authentication" modifier="0" time="Fri Sep 19 13:49:48 2025" msec=" + 578 msec" >
<subject audit-uid="atsec" uid="root" gid="staff" ruid="atsec" rgid="staff" pid="1352" sid="100004" tid="3697 0.0.0.0" />
<text>Verify password for record type Users 'atsec' node '/Local/Default'</text>
<return errval="failure: Unknown error: 255" retval="5000" />
<identity signer-type="1" signing-id="com.apple.opendirectoryd" signing-id-truncated="no" team-id="" team-id-truncated="no" cdhash="0x4a21fe7d1588f7dea2909b5d490d45dbd6d61ed7" />
</record>

Use of privileged/special rights events

<record version="11" event="modify password" modifier="0" time="Fri Sep 19 13:51:55 2025" msec=" + 976 msec" >
<subject audit-uid="atsec" uid="atsec" gid="staff" ruid="atsec" rgid="staff" pid="1433" sid="100004" tid="3872 0.0.0.0" />
<text>Modify password for record type Users 'atsec' node '/Local/Default'</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="0x4a21fe7d1588f7dea2909b5d490d45dbd6d61ed7" />
</record>

<record version="11" event="modify password" modifier="0" time="Fri Sep 19 13:51:40 2025" msec=" + 276 msec" >
<subject audit-uid="atsec" uid="atsec" gid="staff" ruid="atsec" rgid="staff" pid="1432" sid="100004" tid="3870 0.0.0.0" />
<text>Modify password for record type Users 'atsec' node '/Local/Default'</text>
<return errval="failure: Unknown error: 255" retval="5000" />
<identity signer-type="1" signing-id="com.apple.opendirectoryd" signing-id-truncated="no" team-id="" team-id-truncated="no" cdhash="0x4a21fe7d1588f7dea2909b5d490d45dbd6d61ed7" />
</record>

<record version="11" event="open(2) - write,creat" modifier="0" time="Fri Sep 19 15:06:05 2025" msec=" + 936 msec" >
<argument arg-num="3" value="0x1a4" desc="mode" />
<argument arg-num="2" value="0x201" desc="flags" />
<path>/private/etc/security/audit_control</path>
<subject audit-uid="atsec" uid="root" gid="wheel" ruid="root" rgid="wheel" pid="2640" sid="100115" tid="50331650 0.0.0.0" />
<return errval="success" retval="3" />
<identity signer-type="1" signing-id="com.apple.vim" signing-id-truncated="no" team-id="" team-id-truncated="no" cdhash="0x2bb769ba57d0a1c37968916a3552016c6ad5920c" />
</record>

<record version="11" event="open(2) - write,creat" modifier="0" time="Fri Sep 19 15:05:51 2025" msec=" + 727 msec" >
<argument arg-num="3" value="0x1a4" desc="mode" />
<argument arg-num="2" value="0xb01" desc="flags" />
<path>/private/etc/security/audit_control~</path>
<subject audit-uid="atsec" uid="atsec" gid="staff" ruid="atsec" rgid="staff" pid="2637" sid="100115" tid="50331650 0.0.0.0" />
<return errval="failure : Permission denied" retval="4294967295" />

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

<identity signer-type="1" signing-id="com.apple.vim" signing-id-truncated="no" team-id="" team-id-truncated="no" cdhash="0x2bb769ba57d0a1c37968916a3552016c6ad5920c" />
</record>

<record version="11" event="SecSrvr AuthEngine" modifier="0" time="Fri Sep 19 14:33:09 2025" msec=" + 791 msec" >
<subject audit-uid="atsec" uid="atsec" gid="staff" ruid="atsec" rgid="staff" pid="2157" sid="100115" tid="5864 0.0.0.0" />
<text>system.preferences</text>
<text>client /System/Library/ExtensionKit/Extensions/PowerPreferences.appex</text>
<text>creator /System/Library/ExtensionKit/Extensions/PowerPreferences.appex</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="0x5dd18ad68711749f8a2be847d5623cba85b03c4a" />
</record>

<record version="11" event="SecSrvr AuthEngine" modifier="0" time="Fri Sep 19 14:33:03 2025" msec=" + 117 msec" >
<subject audit-uid="atsec" uid="atsec" gid="staff" ruid="atsec" rgid="staff" pid="2157" sid="100115" tid="5864 0.0.0.0" />
<text>system.preferences</text>
<text>client /System/Library/ExtensionKit/Extensions/PowerPreferences.appex</text>
<text>creator /System/Library/ExtensionKit/Extensions/PowerPreferences.appex</text>
<return errval="failure : Operation not permitted" retval="4294907289" />
<identity signer-type="1" signing-id="com.apple.authd" signing-id-truncated="no" team-id="" team-id-truncated="no" cdhash="0x5dd18ad68711749f8a2be847d5623cba85b03c4a" />
</record>

Privilege or role escalation events

<record version="11" event="user authentication" modifier="0" time="Fri Sep 19 15:37:03 2025" msec=" + 576 msec" >
<subject audit-uid="atsec" uid="root" gid="staff" ruid="atsec" rgid="staff" pid="2900" sid="100115" tid="7781 0.0.0.0" />
<text>Verify password for record type Users 'atsec' node '/Local/Default'</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="0x4a21fe7d1588f7dea2909b5d490d45dbd6d61ed7" />
</record>

<record version="11" event="user authentication" modifier="0" time="Fri Sep 19 15:36:52 2025" msec=" + 179 msec" >
<subject audit-uid="atsec" uid="root" gid="staff" ruid="atsec" rgid="staff" pid="2898" sid="100115" tid="7776 0.0.0.0" />
<text>Verify password for record type Users 'atsec' node '/Local/Default'</text>
<return errval="failure: Unknown error: 255" retval="5000" />
<identity signer-type="1" signing-id="com.apple.opendirectoryd" signing-id-truncated="no" team-id="" team-id-truncated="no" cdhash="0x4a21fe7d1588f7dea2909b5d490d45dbd6d61ed7" />
</record>

<record version="11" event="SecSrvr AuthEngine" modifier="0" time="Fri Sep 19 14:38:35 2025" msec=" + 898 msec" >
<subject audit-uid="atsec" uid="atsec" gid="staff" ruid="atsec" rgid="staff" pid="2331" sid="100115" tid="6274 0.0.0.0" />
<text>system.preferences.network</text>
<text>client /System/Library/ExtensionKit/Extensions/Network.appex</text>
<text>creator /System/Library/ExtensionKit/Extensions/Network.appex</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="0x5dd18ad68711749f8a2be847d5623cba85b03c4a" />
</record>

<record version="11" event="SecSrvr AuthEngine" modifier="0" time="Fri Sep 19 14:38:18 2025" msec=" + 398 msec" >
<subject audit-uid="atsec" uid="atsec" gid="staff" ruid="atsec" rgid="staff" pid="2331" sid="100115" tid="6274 0.0.0.0" />
<text>system.preferences.network</text>
<text>client /System/Library/ExtensionKit/Extensions/Network.appex</text>
<text>creator /System/Library/ExtensionKit/Extensions/Network.appex</text>

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

<return errval="failure : Operation not permitted" retval="4294907289" />
<identity signer-type="1" signing-id="com.apple.authd" signing-id-truncated="no" team-id="" team-id-truncated="no" cdhash="0x5dd18ad68711749f8a2be847d5623cba85b03c4a" />
</record>

Administrator or root-level access events

Same as “Privilege or role escalation events” above.

6.2 Apple Unified Logging

Some subsystems require the use of Apple Unified Logging to locate specific records. Once logging is enabled, Apple Unified Logging has a distinct format from the audit daemon logs described earlier. The following table identifies Apple Unified logging field descriptions.

Table 3 - Apple Unified Logging Records

Field	Description	
Timestamp	ISO 8601 format datetime with millisecond precision and timezone offset	
Thread ID	Hexadecimal identifier (e.g., 0x23efd5) representing the specific thread that generated the log message. Useful for tracking execution flow across multiple processes.	
Log Level (type)	Severity indicator (Default, Info, Debug, Error, Fault). Represents the importance and nature of the logged information.	
Activity ID	Hexadecimal identifier (e.g., 0x0) linking related log entries across process boundaries. Helps track operations that span multiple processes or subsystems.	
Process ID (PID)	Numeric identifier (e.g., 15937) of the process that generated the log message. Useful for filtering logs by specific processes.	
TTL	Time To Live value (e.g., 0). Determines how long the log entry should be retained in the system's log store before being purged.	
Message	The message contains a series of optional information that can be used to further isolate events.	
	Process Name	Name of the process (e.g., eapolclient) that generated the log entry. The executable name of the running process.
	(Library/Module)	Contained in parentheses. Library or module involved
	[Subsystem]	Contained in square brackets. Hierarchical identifier (e.g., com.apple.eapol) representing the functional area within the OS or application. Often follows reverse DNS notation and helps categorize logs.
	[Category]	Following the subsystem and separated by a colon. Sub-classification (e.g., Client) within the subsystem. Provides finer granularity for filtering and organizing related log messages.
	Message Contents	The actual log message text containing the specific information being logged. May contain redacted private information (displayed as <private> unless private logging is enabled).

Using the --predicate option with 'log stream' or 'log show' allows specific logs to be filtered on the contents of the fields listed in Table 3. These filters can be combined using boolean operations (AND / OR / NOT) to construct more complex filters.

For example:

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

log stream --predicate "process == 'bluetoothd' and composedMessage CONTAINS[c] '00:02:5B:00:00:00' and composedMessage CONTAINS[c] 'with accept 0'

This filter only captures messages from the bluetoothd process that contain the two specified strings.

6.3 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" along with instructions. The following is the “Profiles and Logs” URL for macOS:

https://developer.apple.com/bug-reporting/profiles-and-logs/?platform=macos

An administrator can use the Apple Unified Logging system to obtain the audit records once the profile is installed.

The following are example Bluetooth audit records.

Failed user authorization of Bluetooth device

2025-09-24 15:19:49.596294-0500 0x1ebf Default 0x143c8 150 0 bluetoothd: [com.apple.bluetooth:Server.BTUIClient] Device Address: 00:02:5B:00:00:00 with accept 0, ignore button 1

Failed user authorization for local Bluetooth Service

Same as “Failed user authorization of Bluetooth device” above. The TOE authenticates and authorizes at the same time.

Initiation of Bluetooth connection

2025-09-24 16:28:37.052184-0500 0x2962 Default 0x0 150 0 bluetoothd: (CoreUtils) [com.apple.bluetooth:CBStackDeviceMonitor] ACL connected: 00:02:5B:00:00:00 "" - "", result 0

Failure of Bluetooth connection

2025-09-24 15:36:30.608564-0500 0x2834 Default 0x0 150 0 bluetoothd: [com.apple.bluetooth:Stack.SDP] 00:02:5B:00:00:00 disconnected with reason STATUS 705

6.4 WLAN Logging

The TOE is capable of generating logs for the WLAN client, including 802.1X EAPOL events. An administrator can use the Apple Unified Logging system to obtain the audit records.

The following are example WLAN audit records.

Failure to establish an EAP-TLS session

2025-10-01 10:13:29.302797-0500 0x2708af Default 0x0 71229 0 eapolclient: [com.apple.eapol:Client] State=Held Status=SecurityError (1001): errSSLXCertChainInvalid (-9807):
2025-10-01 10:13:29.310151-0500 0x2708af Info 0x0 71229 0 eapolclient: [com.apple.eapol:Client] Receive Packet Size 22: EAP Failure (4) From 1a:6:4b:b8:42:a9

Establishment of an EAP-TLS session

2025-09-26 14:54:06.552125-0500 0x1f17 Info 0x0 767 0 eapolclient: [com.apple.eapol:Client] Receive Packet Size 22: EAP Success (3) From 1a:6:4b:b8:42:a9

Termination of an EAP-TLS session

2025-09-26 14:54:17.000872-0500 0x1f17 Info 0x0 767 0 eapolclient: [com.apple.eapol:Client] Transmit Packet Size 4: EAPOL Logoff (2) To 1a:6:4b:b8:42:a9

Failure to validate X.509v3 certificate

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

2025-09-26 14:58:55.7874840-0500 0x381b Default 0x0 997 0 eapolclient: [com.apple.eapol:Client] State=Held Status=SecurityError (1001): errSSLXCertChainInvalid (-9807):

Attempts to load certificates

2025-09-29 11:13:24.110026-0500 0x8efa6 Default 0x1e483e 16389 0 Keychain Access: (Security) [com.apple.securityd:truststore] Set TrustSettings for <cert(0x145c5ce40) s: CA i: CA>

Attempts to revoke certificates

2025-09-29 11:13:53.212834-0500 0x8efa6 Default 0x1e5cb2 16389 0 Keychain Access: (Security) [com.apple.securityd:truststore] Set TrustSettings for <cert(0x123d665d0) s: CA i: CA>

Execution of this set of TSF self-tests

2025-09-26 14:52:26.183317-0500 0x65 Default 0x0 0 0 kernel: (corecrypto) FIPSPOST_KEXT [112007804] fipspost_post:211 all tests PASSED (17 ms)

All attempts to connect to access points

2025-09-26 16:56:46.886705-0500 0xcf0c Default 0x0 169 0 airportd: (IO80211) [com.apple.WiFiManager:] Assoc: <airport[169]> Successfully associated to Wi-Fi network fta_wse_ext.1-allowed on interface en1

2025-09-26 16:56:53.685570-0500 0xcf23 Default 0x0 169 0 airportd: (IO80211) [com.apple.WiFiManager:] AutoJoin: <airport[169]> - [CWXPCSubsystem_associatedToWiFiNetwork:tetherDevice:password:is8021X:passpointDomain:remember:possiblyHidden:updateUserKeychain:interface:processName:suspendAWDLToken:isAutoJoin:error]: Failed to associated to Wi-Fi network fta_wse_ext.1-disallowed on interface en1, triggering auto-join

All attempts to establish a trusted channel

Same as “All attempts to connect to access points events” above.

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

7 Acronyms

ACL
: Access Control List

AES
: Advanced Encryption Standard

API
: Application Programming Interface

CA
: Certificate Authority

CBC
: Cipher Block Chaining

CC
: Common Criteria

CCM
: Counter with CBC-MAC

EAPOL
: Extensible Authentication Protocol over LAN

GPOS
: General Purpose Operating System

IP
: Internet Protocol

MAC
: Message Authentication Code

NIAP
: National Information Assurance Partnership

OS
: Operating System

PP
: Protection Profile

SEP
: Secure Enclave Processor

ST
: Security Target

TLS
: Transport Layer Security

TOE
: Target of Evaluation

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.

UUID
: Universally Unique Identifier

VID
: Validation Identifier

© 2026 Apple Inc. All rights reserved.
This document may be reproduced and distributed only in its original entirety without revision.