APPLE MACOS 15 (SEQUOIA) SUPPLEMENTAL PROCEDURES

Version 1, Release 2

30 January 2025

Developed by Apple and DISA for the DOD

Trademark Information

Names, products, and services referenced within this document may be the trade names, trademarks, or service marks of their respective owners. References to commercial vendors and their products or services are provided strictly as a convenience to our users, and do not constitute or imply endorsement by the Defense Information Systems Agency (DISA) of any nonfederal entity, event, product, service, or enterprise.

TABLE OF CONTENTS

1. TECHNOLOGY IMPLEMENTATION CONSIDERATIONS

1.1 STIG development collaboration with mSCP

The macOS Security Compliance Project (mSCP) is an open-source effort to provide a programmatic approach to generating security guidance. It is authoritative through the National Institute of Standards and Technology (NIST) Special Publication 800-219, Automated Secure Configuration Guidance from the mSCP.

This STIG was developed as part of a working group collaboration with the mSCP and DISA STIG writers. The mSCP can be found at https://github.com/usnistgov/macos_security/tree/main.

2.   TECHNOLOGY CAPABILITY

2.1   Malware Protection

Apple macOS includes built-in protections against malware. Gatekeeper ensures that by default, only trusted software runs on the system. XProtect is a built-in, signature-based antivirus tool that helps protect macOS from malware infections. XProtect definition files are updated by Apple automatically, independent of OS updates. More information about these built-in tools can be found at the following links:

- https://support.apple.com/guide/security/gatekeeper-and-runtime-protection-sec5599b66df/web

- https://support.apple.com/guide/security/protecting-against-malware-sec469d47bd8/

2.2   Software Updates

Keeping macOS up to date ensures it has the latest enhancements and security controls in place. This STIG requires that all updates come from an approved source, and Apple is considered a DOD-approved source. Apple-provided updates must be installed on Apple macOS devices when available. Apple provides the capability for DOD support staff to test most updates before they are released.

Apple provides system administrators with the ability to manage macOS updates through mobile device management. Refer to: https://support.apple.com/en-us/HT211951.

2.3   Data-at-Rest Protection

Apple macOS does not support unlocking FileVault encrypted volumes using smart card-based authentication on Intel-based systems. Therefore, the use of a dedicated local full-disk encryption (FDE) unlock user is required. The unlock user is a password-based account that can only be used to unlock the FileVault encrypted volume. The "unlock" account cannot be used to log in to the operating system. Authorized users boot their systems, enter a password at the preboot screen that decrypts the boot volume, and when presented with the login window, authenticate using a smart card. For more information, refer to: https://support.apple.com/guide/security/when-filevault-is-turned-on-sec4c6dc1b6e/web.

Apple macOS, when running on Apple Silicon-based systems, transparently supports decrypting the FileVault volume for any user who is configured to use a smart card and whose account is authorized to do so. For more information, refer to:
https://support.apple.com/guide/security/managing-filevault-sec8447f5049/1/web/1.

2.4   Device Deployment and Management

2.4.1   Use of Apple Products on Enterprise Networks

Automated deployment and management of Apple devices requires access to specific network services. Apple publishes detailed information about which hosts and ports are required to use

Apple products on enterprise networks at the following link: https://support.apple.com/en-us/HT210060.

Configuration of a network using this information is approved for DOD use. If the firewall supports using hostnames, the Apple services above can be used by allowing outbound connections to *.apple.com. If the firewall can only be configured with IP addresses, allow outbound connections to 17.0.0.0/8. The entire 17.0.0.0/8 address block is assigned to Apple.

2.4.2   Apple Push Notification Service (APNs)

APNs is a platform notification system that developers use to send notification alerts to devices manufactured by Apple, Inc. In addition to app-based alerts, APNs is used by mobile device management (MDM) servers to manage enrolled devices.

APNs is an encrypted and authenticated communication protocol approved for DOD use.

2.4.3   Configuration Profiles

A configuration profile is an XML file that applies configuration information to macOS devices. Although a user cannot change settings defined by an installed configuration profile, in some cases, a user can opt to make a setting more restrictive than what is defined in the profile. For example, if a configuration profile requires the device to lock after five minutes, the user can set the device to lock immediately.

Configuration Profiles can be installed manually or with the use of an MDM server. To install profiles manually, copy them to the target machine, double-click on the profile(s), open System Settings >> Privacy & Security >> Profiles. Click **Install** for each of the profiles to be installed. For more information, refer to: https://support.apple.com/guide/deployment/intro-to-apple-platform-deployment-dep2c1b2a43a/web.

2.4.4   Mobile Device Management (MDM)

MDM servers enable the remote management of enrolled systems. Management includes configuring restrictions, deploying credentials, monitoring compliance, or remotely wiping or locking devices. Using MDM servers to manage macOS devices is a best practice and enables some capabilities that are not possible via other means, such as enabling the Recovery Partition password on Apple Silicon-based systems and preventing local users from installing Kernel Extensions. For more information, refer to: https://support.apple.com/guide/deployment/intro-to-apple-platform-deployment-dep2c1b2a43a/web.

2.5   Firmware Password

Intel-based macOS systems include a recovery partition that can be used to reinstall the operating system, reset local user passwords, and partition the disk, among other tasks. Setting a firmware password on the system will restrict access to the recovery partition and prevent the user from

booting the computer from external media or from booting into Target Disk Mode. For more information, refer to: https://support.apple.com/en-us/HT201462.

The firmware password (https://support.apple.com/en-us/HT204455) can be set or removed from the recovery partition using either the Firmware Password Utility or Startup Security Utility. The firmware password can also be set, removed, or verified while logged in to macOS using the **firmwarepasswd** command. Once a firmware password is set, macOS will ask for the firmware password when attempting to boot from a volume other than the one set in the Startup Disk preference pane or when starting up into the Recovery partition.

Note: The only way to reset a forgotten password is with a machine-specific binary generated and provided by Apple. A user must schedule a support call and provide proof of purchase before the firmware binary will be generated.

On Apple Silicon-based systems starting with macOS 11.5, MDM administrators can set a password (using the new **SetRecoveryLock** command) that must be entered before a user can restart the system with Apple Silicon into the recoveryOS. For example, the user will not be able to modify security settings or erase the system. This password can be set only by the MDM solution. It can be removed by the MDM solution, unenrolling in MDM, or if the system is erased. MDM administrators can also verify a recoveryOS password is set by using the new **VerifyRecoveryLock** command.

2.6   Smart Cards

Apple macOS supports Personal Identity Verification (PIV)-based smart cards and has built-in support for USB CCID class-compliant smart card readers. Smart card-based authentication is supported in the following subsystems: Login Window, Screen Saver, ssh, sudo, Safari, PAM Authorization, login, su, and Finder.

Smart card-based authentication on macOS can be configured in fixed-key mapping or attribute-based mapping. Fixed-key mapping associates the hash of a public key on the users' smart card with a local account. Attribute-based matching associates certificate field values from the smart card to predefined values in a Directory Server.

By default, macOS will authenticate users using either a password or a smart card that has been bound to their account through fixed key or attribute mapping. Mandatory smart card-based authentication can be enabled using a configuration profile. Enabling mandatory smart card-based authentication without first verifying that smart card authentication is working can prevent all users from logging in to the machine. For more information, refer to: https://support.apple.com/en-us/HT208372.

2.6.1   Smart Card Enforcement Exemption

All use of exemptions to the smart card enforcement on Apple macOS must be approved and documented with the site information system security manager (ISSM) or authorizing official (AO).

2.6.1.1 Group Exemption

Starting in macOS 10.15, enforcement on a system can be granularly configured by adding a field to /private/etc/SmartcardLogin.plist. The **NotEnforcedGroup** can be added to the file to list a Directory group that will not be included in smart card enforcement. To activate this feature, **enforceSmartCard** and **allowUnmappedUsers** must be applied via a configuration profile (com.apple.security.smartcard).

To configure the **NotEnforcedGroup**, the SmartcardLogin.plist should be minimally configured as follows:

```
<?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>AttributeMapping</key>
    <dict>
        <key>fields</key>
        <array>
            <string>NT Principal Name</string>
        </array>
        <key>formatString</key>
        <string>Kerberos:$1</string>
        <key>dsAttributeString</key>
        <string>dsAttrTypeStandard:AltSecurityIdentities</string>
    </dict>
    <key>TrustedAuthorities</key>
  <array>

<string>SHA256_HASH_OF_CERTDOMAIN_1,SHA256_HASH_OF_CERTDOMAIN_2</string>
  </array>
    <key>NotEnforcedGroup</key>
    <string>EXEMPTGROUP</key>
</dict>
</plist>
```

Once a system is configured for the **NotEnforcedGroup**, a user can be added to the assigned group by running the following:

```
/usr/sbin/dseditgroup -o edit -a <exempt_user> -t user <notenforcegroup>
```

2.6.1.2 User Exemption

Alternatively, if a single user needs to be exempt for a period of time, **kDSNativeAttrTypePrefix:SmartCardEnforcement** can be set in the user's Open Directory record. The following values can be set:

> 0 - The system default is respected.

> 1 - Smartcard enforcement is enabled.

2 - Smartcard enforcement is disabled.

In Active Directory environments, the value of the **userAccountControl** attribute is respected. Run the following command to set the exemption when booted from macOS:

```
/usr/bin/dscl . -append /Users/<username> SmartCardEnforcement 2
```

Run the following command to set the exemption when booted from Recovery:

```
/usr/bin/defaults write /Volumes/Macintosh\
HD/var/db/dslocal/nodes/Default/users/<username> SmartCardEnforcement -array-
add 2
```

When booted to recovery on an Apple Silicon Mac, run the following after setting the exemption.

```
/usr/sbin/diskutil apfs updatePreboot /Volumes/Macintosh\ HD
```

2.6.1.3 Temporary Exemption

On an Apple Silicon Mac, if a temporary exemption is needed, **security filevault skip-sc-enforcement** will disable smart card enforcement on next boot only.

Run the following command to set the temporary exemption when booted from Recovery:

```
/usr/bin/security filevault skip-sc-enforcement <data volume UUID> set
```

To obtain the data volume UUID, run the following:

```
/usr/sbin/diskutil apfs listGroups | /usr/bin/awk -F: '/ Data/ { getline;
gsub(/ /,""); print $2}'
```

2.7   Kernel Extensions

Kernel Extensions (Kexts) are no longer recommended for macOS. The use of Kexts puts the performance and reliability of the system at risk. Systems administrators should select solutions that do not require extending the kernel. For more information, refer to:
https://support.apple.com/guide/security/kernel-extensions-sec8e454101b/1/web/1.

2.7.1   Endpoint Security Framework

Starting in macOS 10.15 (Catalina), all versions of macOS include an Endpoint Security Framework that provides a C-Language API, which can be used to monitor the system for malicious activity. The events that can be monitored include process executions, mounting file systems, forking processes, and raising signals. For more information, refer to:
https://developer.apple.com/documentation/endpointsecurity.

3. SUPPLEMENTAL PROCEDURES

3.1 Building a Certificate Root Trust Payload

Logging in to a macOS machine that has had a STIG applied requires that identities on the CACs used to authenticate users be trusted. Apple has not shipped DOD roots in the trust store for macOS since High Sierra. The following steps demonstrate how to build a Configuration Profile that contains the current DOD roots required to establish trust.

The root certificates are available from the DISA PKI page on DOD Cyber Exchange at:
https://cyber.mil/pki-pke/tools-configuration-files/.

After downloading and expanding the root certificate's ZIP, follow the instructions in the README.txt file to verify the certificates. Use the following command to convert the archive to PEM for use in the next step:

```
openssl pkcs7 -in Certificates_PKCS7_v5.9_DoD.pem.p7b -print_certs -out
DoD_CAs.pem
```

Convert the PEM encoded file to p12:

```
openssl pkcs12 -export -nokeys -in DoD_CAs.pem -out DoD_CAs.p12
```

Once the P12 has been created, create a new Configuration Profile and import the newly created p12 into that Profile as a certificate payload. For more information, refer to:
https://support.apple.com/guide/apple-configurator-2/create-and-edit-configuration-profiles-pmd85719196/mac.

This will produce a mobileconfig policy file that applies only to users who install the file. To make this a system policy, open the mobileconfig file with a text editor and insert the following two lines before the closing dict and plist at the end of the file **</dict></plist>:**

```
<key>PayloadScope</key>
<string>System</string>
```

3.2 Apple macOS Hard Disk Erase Procedures

A cryptographic wipe is designed to permanently delete data so it cannot be recovered. This includes email accounts, downloaded apps, media files, documents, browser bookmarks, and settings. These procedures are appropriate for macOS devices never exposed to classified data and require that FileVault Disk Encryption is enabled.

Follow the steps below when a macOS system is being retired from use:

1. Boot from the Recovery partition.
2. From the macOS Utilities window, select **Disk Utility**, and then click **Continue**.
3. Select the Hard Disk to be erased, click **Erase**, and fill in the requested fields shown below:

   - Name: Type the name the disk will have after it is erased.

UNCLASSIFIED

UNCLASSIFIED

Apple macOS 15 Supplemental Procedures, V1R2
30 January 2025

DISA
Developed by Apple and DISA for the DOD

- Format: Choose **APFS** or **Mac OS Extended (Journaled)**; Disk Utility shows a compatible format by default.
- Scheme: Choose **GUID Partition Map**.

For more information, refer to: https://support.apple.com/en-us/HT208496 and https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-88r1.pdf.

Apple Silicon or Intel-based systems with a T2 security chip running macOS 15 will allow either a local administrator or MDM administrator (if the system is enrolled) to perform an Erase All Content and Settings command. This erases all user data and any additional volumes on the system. On Apple Silicon-based systems, the system's security settings are reset to their default value, Full Security.

3.3   Example Setup Workflow

Apple macOS ships in a least-privilege configuration. The first user configured during initial setup will be an administrative user. The accepted best practice is for all subsequent users to be nonprivileged users. Some user tasks may require elevated permissions. In these cases, system administrators should configure the system to support the constrained ability of users to accomplish required tasks. The built-in command-line tools "sudo" and "security" can be used to grant additional permissions to unprivileged users. Refer to "man sudo" and the authorizationdb section of "man security" for more information.

The following workflow addresses only the simplest use case of setting up a standalone or networked machine using mandatory smart card authentication against a local account. More complicated workflows, including directory-bound or Apple Business Manager (ABM)-based enrollments, are beyond the scope of this section. The following procedure will provision a local admin account, which will be exempt from the smart card mandatory policy, and a local unprivileged user account, which is bound by the smart card mandatory policy. Using this method will verify the certificates required to establish trust are in place and the mandatory smart card policy is in place without the risk of locking the local administrator account. Once smart card login is verified as working, consider removing the smart card mandatory exemption for the administrative user.

1. Collect required equipment.
   - System running 15.x or greater.
   - STIG materials.
   - Smart card that will be paired with the local administrator account.
   - USB smart card reader.
   - Certificates required to establish smart card trust.
2. Power on the system and proceed through the setup assistant.
3. At the **Create a Computer Account** prompt:
   - This account will be the local administrative account of last resort.
   - The name of the account should follow local conventions.

   Note: When setting up this user, steps will be displayed that will be suppressed for users created after applying the STIG.

UNCLASSIFIED

4. Install the certificate roots and intermediates, which are required to validate the trust chain used for the organization's smart cards.

5. Insert the administrative smart card and follow the on-screen prompts to pair with the local account. Keyboard Setup prompts may be dismissed safely.

    - The onscreen prompts require creating a password, which will be wrapped with the private key from the smart card to secure the user's macOS keychain.

    - Verify the pairing by logging out and then back in, using the smart card to authenticate.

6. Create a new nonprivileged user, sign in as that user, and associate a smart card to the user's account.

7. Log in as the administrative user.

8. Apply the **U_Apple_macOS_15_V1R1_STIG_Security_Smartcard** configuration profile, which will enforce mandatory certificate checks.

9. Log in as the administrative user using the smart card.

    - If unable to log in as the administrator using the smart card, log in using the password and correct the certificate trust problem.

10. Verify the nonprivileged user cannot log in with a password and must use the smart card.