FIPS 140-2

Some of the cryptographic modules supporting Apple macOS 10.15 have not been FIPS 140-2- validated as of the date of this publication but are in process. DoD organizations using macOS 10.15 devices should visit the following website to obtain updates on validation status:
http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401vend.htm

The cryptographic modules supporting Apple macOS 10.15 are being revalidated to accommodate new features in Apple macOS 10.15. The previously validated cryptographic modules are:

- Apple CoreCrypto Module v9.0 for Intel, which is a software cryptographic module running on a multi-chip standalone hardware device and providing services intended to protect data in transit and at rest. (FIPS Certificate number 3402)
- Apple CoreCrypto Kernel Module v9.0 for Intel, which is used by the kernel for low-level services intended to protect data in transit and at rest. (FIPS Certificate number 3431)
- Apple Secure Key Store Cryptographic Module v9.0, which provides services intended to protect data in transit and at rest on T2-equipped Macs.(FIPS Certificate number 3523)

Software Updates

Keeping Apple macOS up to date ensures that it has the latest enhancements and security controls in place. This STIG requires that all updates come from an approved source. 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 Push Notification Service (APNS)

APNS are encrypted and authenticated communication tools approved for DoD use.

Firmware Password

Systems running macOS 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 as well as prevent the user from booting the computer from external media or from booting into Target Disk Mode (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. You can also set, remove, or verify the firmware password 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.

Please note that if you cannot remember your firmware password or passcode, the only way to reset the forgotten password is through the use of a machine-specific binary generated and provided by Apple. You will need to schedule a support call and provide proof of purchase before the firmware binary will be generated.

Full Disk Encryption

FileVault full-disk encryption (FileVault 2) uses XTS-AES-128 encryption with a 256-bit key to help prevent unauthorized access to the information on your startup disk. macOS does not support unlocking FileVault encrypted volumes using smart card-based authentication. Therefore, the use of a dedicated local FDE unlock user is recommended when a mandatory smart card authentication policy is enforced. 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 Mac. Authorized users boot their Macs, enter a password at the pre-boot screen, which decrypts the boot volume, and then are presented with a login window where they can authenticate using a smart card.

Smart Cards

macOS supports Personal Identity Verification (PIV)-based smart cards. macOS has built-in support for USB CCID class-compliant SmartCard readers.

Mandatory Smart Card Enforcement

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 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 through the use of a configuration profile. Enabling mandatory smart card-based authentication without first verifying that smart card authentication is working can prevent all users from logging into the machine. See https://support.apple.com/en-us/HT208372 for more information.

Building a Certificate Root Trust Payload

Logging in to a macOS machine that has been STIG'd 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 will 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:
https://public.cyber.mil/pki-pke/end-users/getting-started/

After downloading and expanding the ZIP, follow the instructions in the README.txt 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_v5.5_DoD.pem.p7b -print_certs -out DoD_CAs.pem
```

Finally, 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.

See https://support.apple.com/guide/apple-configurator-2/create-and-edit-configuration-profiles-pmd85719196/mac for more information.

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>
```

Suggested Setup Workflow

The following workflow only addresses 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 f(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 that the certificates required to establish trust are in place and the mandatory smart card policy is in place without risking locking the local administrator account. Once it is verified that smart card login works, consider removing the smart card mandatory exemption for the administrative user.

1. Collect required equipment
   Mac running 10.13.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 Mac and proceed through the setup assistant.

3. When you reach “Create a Computer Account”:
   a. This account will be the local administrative account of last resort.
   b. The name of the account should follow local conventions.
   c. Note that you will see steps when setting up this user that will be suppressed for users that are created after applying the STIG.

4. Install the certificate roots and intermediates that are required to validate the trust chain used for your organization’s smart cards.

5. Insert administrative smart card and follow the on-screen prompts to pair with the local account. You may safely dismiss Keyboard Setup prompts.
   a. The onscreen prompts require creating a password, which will be wrapped with the private key from the smart card to secure the users’ macOS keychain.
   b. Verify the pairing by logging out and then back in using the smart card to authenticate.

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

7. Log in as the administrative user.

8. Create /etc/SmartcardLogin.plist with the following content, owned by root:wheel, and with permissions set to 644.

```
            <?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>NotEnforcedGroup</key>
                    <string>admin</string>
                    </dict>
           </plist>
```

9. Apply the U_Apple_OS_X_10-15_V1R1_STIG_Smart_Card_Policy, mobileconfig policy, which will enforce mandatory smart card authentication.

10. Log in as the administrative user using the smart card.
    a. If you are unable to log in as the administrator using the smart card, log in using the password and correct the certificate trust problem.
    b. Optional: Verify that the administrative user can log in with a password.

11. Verify that the unprivileged user cannot login with a password and must use the smart card.

12. To disable the mandatory smart card exemption for the admin group, remove /etc/SmartcardLogin.plist and reboot.