macOS Security Compliance

macOS 26.0

Security Configuration - US CMMC 2.0 Level 1

Tahoe Guidance, Revision 1.0 (2025-09-11)

Table of Contents

Chapter 1. Foreword

The macOS Security Compliance Project is an open source effort to provide a programmatic approach to generating security guidance. The configuration settings in this document were derived from National Institute of Standards and Technology (NIST) Special Publication (SP) 800-53, *Security and Privacy Controls for Information Systems and Organizations*, Revision 5.

This project can be used as a resource to easily create customized security baselines of technical security controls by leveraging a library of atomic actions which are mapped to the compliance requirements defined in NIST SP 800-53 (Rev. 5). It can also be used to develop customized guidance to meet the particular cybersecurity needs of any organization.

The objective of this effort was to simplify and radically accelerate the process of producing up-to-date macOS security guidance that is also accessible to any organization and tailorable to meet each organization's specific security needs.

Any and all risk based decisions to tailor the content produced by this project in order to meet the needs of a specific organization shall be approved by the responsible Information System Owner (ISO) and Authorizing Official (AO) and formally documented in their System Security Plan (SSP). While the project attempts to provide settings to meet compliance requirements, it is recommended that each rule be reviewed by your organization's Information System Security Officer (ISSO) prior to implementation.

Chapter 2. Scope

This guide describes the actions to take when securing a macOS 26.0 system against the US CMMC 2.0 Level 1 security baseline.

Information System Security Officers and benchmark creators can use this catalog of settings in order to assist them in security benchmark creation. This list is a catalog, not a checklist or benchmark, and satisfaction of every item is not likely to be possible or sensible in many operational scenarios.

Chapter 3. Authors

macOS Security Compliance Project

John Mahlman	Leidos
Bob Gendler	National Institute of Standards and Technology
Dan Brodjieski	National Aeronautics and Space Administration
Allen Golbig	Jamf

Chapter 4. Acronyms and Definitions

Table 1. Acronyms and Abbreviations

AES	Advanced Encryption Standard
ABM	Apple Business Manager
AFP	Apple Filing Protocol
ALF	Application Layer Firewall
AO	Authorizing Official
API	Application Programming Interface
ARD	Apple Remote Desktop
CA	Certificate Authority
CIS	Center for Internet Security
CMMC	Cybersecurity Maturity Model Certification
CNSSI	Committee on National Security Systems
CRL	Certificate Revocation List
DISA	Defense Information Systems Agency
DMA	Direct Memory Access
FISMA	Federal Information Security Modernization Act
FPKI	Federal Public Key Infrastructure
IR	Infrared
ISO	Information System Owner
ISSO	Information System Security Officer
MDM	Mobile Device Management
NASA	National Aeronautics and Space Administration
NFS	Network File System
NIST	National Institute of Standards and Technology
NSA	National Security Agency
OCSP	Online Certificate Status Protocol
ODV	Organization Defined Values
OS	Operating System
PF	Packet Filter
PIV	Personal Identity Verification
PIV-M	Personal Identity Verification Mandatory
PKI	Public Key Infrastructure
RBD	Risk Based Decision

SIP	System Integrity Protection
SMB	Server Message Block
SSH	Secure Shell
SSP	System Security Plan
STIG	Security Technical Implementation Guide
UAMDM	User Approved MDM
UUCP	Unix-to-Unix Copy Protocol

Table 2. Definitions

Baseline	A baseline is a predefined set of controls (also referred to as "a catalog" of settings) that address the protection needs of an organization’s information systems. A baseline serves as a starting point for the creation of security benchmarks.
Benchmark	Benchmarks are a defined list of settings with values that an organization has defined.

Chapter 5. Applicable Documents

5.1. Government Documents

Table 3. National Institute of Standards and Technology (NIST)

Document Number or Descriptor	Document Title
NIST Special Publication 800-53 Rev 5	*NIST Special Publication 800-53 Rev 5.1.1*
NIST Special Publication 800-63	*NIST Special Publication 800-63*
NIST Special Publication 800-171	*NIST Special Publication 800-171 Rev 3*
NIST Special Publication 800-219	*NIST Special Publication 800-219 Rev 1*

Table 4. Defense Information Systems Agency (DISA)

Document Number or Descriptor	Document Title
STIG Ver 1, Rel 4	*Apple macOS 15 (Sequoia) STIG*

Table 5. Cybersecurity Maturity Model Certification (CMMC)

Document Number or Descriptor	Document Title
CMMC Model Overview v2.0	*Cybersecurity Maturity Model Certification (CMMC) Model Overview v2.0*

Table 6. Committee on National Security Systems (CNSS)

Document Number or Descriptor	Document Title
CNSSI No. 1253	*Security Categorization and Control Selection for National Security Systems*

5.2. Non-Government Documents

Table 7. Apple

Document Number or Descriptor	Document Title
Apple Platform Security Guide	*Apple Platform Security*
Apple Platform Deployment	*Apple Platform Deployment*
Apple Platform Certifications	*Apple Platform Certifications*
Profile-Specific Payload Keys	*Profile-Specific Payload Keys*

Table 8. Center for Internet Security

Document Number or Descriptor	Document Title
Apple macOS 15.0	*CIS Apple macOS 15.0 Benchmark version 1.1.0*

Chapter 6. Authentication

This section contains the configuration of authentication settings, including the enforcement of smartcard authentication.

> See additional guidance in the Smartcard Supplemental.

> The check/fix commands outlined in this section must be run with elevated privileges.

6.1. Allow Smartcard Authentication

Smartcard authentication *MUST* be allowed.

The use of smartcard credentials facilitates standardization and reduces the risk of unauthorized access.

When enabled, the smartcard can be used for login, authorization, and screen saver unlocking.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.security.smartcard')\
.objectForKey('allowSmartCard').js
EOS
```

If the result is not **true**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.security.smartcard) payload type:

```
<key>allowSmartCard</key>
<true/>
```

ID	auth_smartcard_allow

References	**800-53r5**	IA-2(1), IA-2(12), IA-2(2)
	CMMC	IA.L1-3.5.1 IA.L1-3.5.2 IA.L2-3.5.3
	CCE	CCE-95135-0

6.2. Enforce Smartcard Authentication

Smartcard authentication *MUST* be enforced.

The use of smartcard credentials facilitates standardization and reduces the risk of unauthorized access.

When enforceSmartCard is set to "true", the smartcard must be used for login, authorization, and unlocking the screensaver.

enforceSmartCard will apply to the whole system. No users will be able to login with their password unless the profile is removed or a user is exempt from smartcard enforcement.

enforceSmartcard requires allowSmartcard to be set to true in order to work.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.security.smartcard')\
.objectForKey('enforceSmartCard').js
EOS
```

If the result is not **true**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.security.smartcard) payload type:

```
<key>enforceSmartCard</key>
<true/>
<key>allowSmartCard</key>
<true/>
```

ID	auth_smartcard_enforce	
References	**800-53r5**	IA-2, IA-2(1), IA-2(12), IA-2(2), IA-2(6), IA-2(8) IA-5(2)
	CMMC	IA.L1-3.5.1 IA.L1-3.5.2 IA.L2-3.5.3 IA.L2-3.5.4
	CCE	CCE-95138-4

6.3. Disable Password Authentication for SSH

If remote login through SSH is enabled, password based authentication *MUST* be disabled for user login.

All users *MUST* go through multifactor authentication to prevent unauthenticated access and potential compromise to the system.

> /etc/ssh/sshd_config will be automatically modified to its original state following any update or major upgrade to the operating system.

To check the state of the system, run the following command(s):

```
/usr/sbin/sshd -G | /usr/bin/grep -Ec
'^(passwordauthentication\s+no|kbdinteractiveauthentication\s+no)'
```

If the result is not **2**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

```
include_dir=$(/usr/bin/awk '/^Include/ {print $2}' /etc/ssh/sshd_config |
/usr/bin/tr -d '*')
if [[ -z $include_dir ]]; then
  /usr/bin/sed -i.bk "1s/.*/Include \/etc\/ssh\/sshd_config.d\/\*/"
/etc/ssh/sshd_config
fi
echo "passwordauthentication no" >> "${include_dir}01-mscp-sshd.conf"
echo "kbdinteractiveauthentication no" >> "${include_dir}01-mscp-sshd.conf"

for file in $(ls ${include_dir}); do
  if [[ "$file" == "100-macos.conf" ]]; then
      continue
  fi
  if [[ "$file" == "01-mscp-sshd.conf" ]]; then
      break
  fi
  /bin/mv ${include_dir}${file} ${include_dir}20-${file}
done
```

ID	auth_ssh_password_authentication_disable	
References	**800-53r5**	IA-2, IA-2(1), IA-2(2), IA-2(6), IA-2(8) IA-5(2) MA-4
	CMMC	IA.L1-3.5.1 IA.L1-3.5.2 IA.L2-3.5.3 IA.L2-3.5.4 MA.L2-3.7.5
	CCE	CCE-95139-2

Chapter 7. iCloud

This section contains the configuration and enforcement of iCloud and the Apple ID service settings.

> The check/fix commands outlined in this section *MUST* be run by a user with with elevated privileges.

7.1. Disable iCloud Address Book

The macOS built-in Contacts.app connection to Apple's iCloud service *MUST* be disabled.

Apple's iCloud service does not provide an organization with enough control over the storage and access of data, and, therefore, automated contact synchronization *MUST* be controlled by an organization approved service.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowCloudAddressBook').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowCloudAddressBook</key>
<false/>
```

ID	icloud_addressbook_disable

References	**800-53r5**	- AC-20, AC-20(1) - CM-7, CM-7(1) - SC-7(10)
	CMMC	- AC.L1-3.1.20 - CM.L2-3.4.6 - CM.L2-3.4.7
	CCE	- CCE-95140-0

7.2. Disable the System Setting for Apple ID

The system setting for Apple ID *MUST* be disabled.

Disabling the system setting prevents login to Apple ID and iCloud.

To check the state of the system, run the following command(s):

```
/usr/bin/profiles show -output stdout-xml | /usr/bin/xmllint --xpath
'//key[text()="DisabledSystemSettings"]/following-sibling::*[1]' - | /usr/bin/grep -c
"com.apple.systempreferences.AppleIDSettings"
```

If the result is not **1**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.systempreferences) payload type:

```
<key>DisabledSystemSettings</key>
<array>
    <string>com.apple.systempreferences.AppleIDSettings</string>
</array>
```

ID	icloud_appleid_system_settings_disable

References	**800-53r5**	- AC-20, AC-20(1) - CM-7, CM-7(1)
	CMMC	- AC.L1-3.1.20 - CM.L2-3.4.6 - CM.L2-3.4.7
	CCE	- CCE-95141-8

7.3. Disable iCloud Bookmarks

The macOS built-in Safari.app bookmark synchronization via the iCloud service *MUST* be disabled.

Apple’s iCloud service does not provide an organization with enough control over the storage and access of data and, therefore, automated bookmark synchronization *MUST* be controlled by an organization approved service.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowCloudBookmarks').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowCloudBookmarks</key>
<false/>
```

ID	icloud_bookmarks_disable

References	**800-53r5**	• AC-20, AC-20(1) • CM-7, CM-7(1) • SC-7(10)
	CMMC	• AC.L1-3.1.20 • CM.L2-3.4.6 • CM.L2-3.4.7
	CCE	• CCE-95142-6

7.4. Disable the iCloud Calendar Services

The macOS built-in Calendar.app connection to Apple’s iCloud service *MUST* be disabled.

Apple’s iCloud service does not provide an organization with enough control over the storage and access of data and, therefore, automated calendar synchronization *MUST* be controlled by an organization approved service.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowCloudCalendar').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowCloudCalendar</key>
<false/>
```

ID	icloud_calendar_disable

References	**800-53r5**	• AC-20, AC-20(1) • CM-7, CM-7(1) • SC-7(10)
	CMMC	• AC.L1-3.1.20 • CM.L2-3.4.6 • CM.L2-3.4.7
	CCE	• CCE-95143-4

7.5. Disable iCloud Document Sync

The macOS built-in iCloud document synchronization service *MUST* be disabled to prevent organizational data from being synchronized to personal or non-approved storage.

Apple’s iCloud service does not provide an organization with enough control over the storage and access of data and, therefore, automated document synchronization *MUST* be controlled by an organization approved service.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowCloudDocumentSync').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowCloudDocumentSync</key>
<false/>
```

ID	icloud_drive_disable

References	**800-53r5**	• AC-20, AC-20(1) • CM-7, CM-7(1) • SC-7(10)
	CMMC	• AC.L1-3.1.20 • CM.L2-3.4.6 • CM.L2-3.4.7
	CCE	• CCE-95144-2

7.6. Disable the iCloud Freeform Services

The macOS built-in Freeform.app connection to Apple's iCloud service *MUST* be disabled.

Apple's iCloud service does not provide an organization with enough control over the storage and access of data and, therefore, automated calendar synchronization *MUST* be controlled by an organization approved service.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowCloudFreeform').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowCloudFreeform</key>
<false/>
```

ID	icloud_freeform_disable

References	**800-53r5**	- AC-20, AC-20(1) - CM-7, CM-7(1) - SC-7(10)
	CMMC	- AC.L1-3.1.20 - CM.L2-3.4.6 - CM.L2-3.4.7
	CCE	- CCE-95145-9

7.7. Disable iCloud Game Center

This works only with supervised devices (MDM) and allows to disable Apple Game Center. The rationale is Game Center is using Apple ID and will shared data on AppleID based services, therefore, Game Center *MUST* be disabled. This setting also prohibits functionality of adding friends to Game Center.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowGameCenter').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowGameCenter</key>
<false/>
```

ID	icloud_game_center_disable

References	**800-53r5**	- AC-20, AC-20(1) - CM-7, CM-7(1) - SC-7(10)
	CMMC	- AC.L1-3.1.20 - CM.L2-3.4.6 - CM.L2-3.4.7
	CCE	- CCE-95146-7

7.8. Disable iCloud Keychain Sync

The macOS system's ability to automatically synchronize a user's passwords to their iCloud account *MUST* be disabled.

Apple's iCloud service does not provide an organization with enough control over the storage and access of data and, therefore, password management and synchronization *MUST* be controlled by an organization approved service.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowCloudKeychainSync').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowCloudKeychainSync</key>
<false/>
```

ID	icloud_keychain_disable

References	**800-53r5**	- AC-20, AC-20(1) - CM-7, CM-7(1) - SC-7(10)
	CMMC	- AC.L1-3.1.20 - CM.L2-3.4.6 - CM.L2-3.4.7
	CCE	- CCE-95147-5

7.9. Disable iCloud Mail

The macOS built-in Mail.app connection to Apple’s iCloud service *MUST* be disabled.

Apple’s iCloud service does not provide an organization with enough control over the storage and access of data and, therefore, automated mail synchronization *MUST* be controlled by an organization approved service.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowCloudMail').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowCloudMail</key>
<false/>
```

ID	icloud_mail_disable

References	**800-53r5**	- AC-20, AC-20(1) - CM-7, CM-7(1) - SC-7(10)
	CMMC	- AC.L1-3.1.20 - CM.L2-3.4.6 - CM.L2-3.4.7
	CCE	- CCE-95148-3

7.10. Disable iCloud Notes

The macOS built-in Notes.app connection to Apple's iCloud service *MUST* be disabled.

Apple's iCloud service does not provide an organization with enough control over the storage and access of data and, therefore, automated Notes synchronization *MUST* be controlled by an organization approved service.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowCloudNotes').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowCloudNotes</key>
<false/>
```

ID	icloud_notes_disable

References	**800-53r5**	• AC-20, AC-20(1) • CM-7, CM-7(1) • SC-7(10)
	CMMC	• AC.L1-3.1.20 • CM.L2-3.4.6 • CM.L2-3.4.7
	CCE	• CCE-95149-1

7.11. Disable iCloud Photo Library

The macOS built-in Photos.app connection to Apple’s iCloud service *MUST* be disabled.

Apple’s iCloud service does not provide an organization with enough control over the storage and access of data and, therefore, automated photo synchronization *MUST* be controlled by an organization approved service.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowCloudPhotoLibrary').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowCloudPhotoLibrary</key>
<false/>
```

ID	icloud_photos_disable

References	**800-53r5**	- AC-20, AC-20(1) - CM-7, CM-7(1) - SC-7(10)
	CMMC	- AC.L1-3.1.20 - CM.L2-3.4.6 - CM.L2-3.4.7
	CCE	- CCE-95150-9

7.12. Disable iCloud Private Relay

Enterprise networks may be required to audit all network traffic by policy, therefore, iCloud Private Relay *MUST* be disabled.

Network administrators can also prevent the use of this feature by blocking DNS resolution of mask.icloud.com and mask-h2.icloud.com.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowCloudPrivateRelay').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowCloudPrivateRelay</key>
<false/>
```

ID	icloud_private_relay_disable

References	**800-53r5**	AC-20, AC-20(1) CM-7, CM-7(1) SC-7(10)
	CMMC	AC.L1-3.1.20 CM.L2-3.4.6 CM.L2-3.4.7
	CCE	CCE-95151-7

7.13. Disable iCloud Reminders

The macOS built-in Reminders.app connection to Apple's iCloud service *MUST* be disabled.

Apple's iCloud service does not provide an organization with enough control over the storage and access of data and, therefore, automated reminders synchronization *MUST* be controlled by an organization approved service.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowCloudReminders').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowCloudReminders</key>
<false/>
```

ID	icloud_reminders_disable

References	**800-53r5**	• AC-20, AC-20(1) • CM-7, CM-7(1) • SC-7(10)
	CMMC	• AC.L1-3.1.20 • CM.L2-3.4.6 • CM.L2-3.4.7
	CCE	• CCE-95152-5

7.14. Disable iCloud Desktop and Document Folder Sync

The macOS system's ability to automatically synchronize a user's desktop and documents folder to their iCloud Drive *MUST* be disabled.

Apple's iCloud service does not provide an organization with enough control over the storage and access of data and, therefore, automated file synchronization *MUST* be controlled by an organization approved service.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowCloudDesktopAndDocuments').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowCloudDesktopAndDocuments</key>
<false/>
```

ID	icloud_sync_disable

References	**800-53r5**	• AC-20, AC-20(1) • CM-7, CM-7(1) • SC-7(10)
	CMMC	• AC.L1-3.1.20 • CM.L2-3.4.6 • CM.L2-3.4.7
	CCE	• CCE-95153-3

Chapter 8. macOS

This section contains the configuration and enforcement of operating system settings.

> The check/fix commands outlined in this section *MUST* be run by a user with elevated privileges.

8.1. Disable AppleID and Internet Account Modifications

The system *MUST* disable account modification.

Account modification includes adding additional or modifying internet accounts in Apple Mail, Calendar, Contacts, in the Internet Account System Setting Pane, or the AppleID System Setting Pane.

This prevents the addition of unauthorized accounts.

> Some organizations may allow the use and configuration of the built-in Mail.app, Calendar.app, and Contacts.app for organizational communication. Information System Security Officers (ISSOs) may make the risk-based decision not to disable the Internet Accounts System Preference pane to avoid losing this functionality, but they are advised to first fully weigh the potential risks posed to their organization.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowAccountModification').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowAccountModification</key>
<false/>
```

ID	os_account_modification_disable	
References	**800-53r5**	AC-20, AC-20(1) CM-7, CM-7(1)
	CMMC	AC.L1-3.1.20 CM.L2-3.4.6 CM.L2-3.4.7
	CCE	CCE-95155-8

8.2. Disable AirDrop

AirDrop *MUST* be disabled to prevent file transfers to or from unauthorized devices. AirDrop allows users to share and receive files from other nearby Apple devices.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowAirDrop').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowAirDrop</key>
<false/>
```

ID	os_airdrop_disable

References	**800-53r5**	- AC-20 - AC-3 - CM-7, CM-7(1)
	CMMC	- AC.L1-3.1.1 - AC.L1-3.1.20 - CM.L2-3.4.6 - CM.L2-3.4.7
	CCE	- CCE-95156-6

8.3. Disable Apple ID Setup during Setup Assistant

The prompt for Apple ID setup during Setup Assistant *MUST* be disabled.

macOS will automatically prompt new users to set up an Apple ID while they are going through Setup Assistant if this is not disabled, misleading new users to think they need to create Apple ID accounts upon their first login.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript 2>/dev/null << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.SetupAssistant.managed')\
.objectForKey('SkipSetupItems').containsObject("AppleID")
EOS
```

If the result is not **true**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.SetupAssistant.managed) payload type:

```
<key>SkipSetupItems</key>
<array>
    <string>AppleID</string>
</array>
```

ID	os_appleid_prompt_disable

References	**800-53r5**	• AC-20
	CMMC	• AC.L1-3.1.20
	CCE	• CCE-95159-0

8.4. Enable Authenticated Root

Authenticated Root *MUST* be enabled.

When Authenticated Root is enabled the macOS is booted from a signed volume that is cryptographically protected to prevent tampering with the system volume.

ℹ Authenticated Root is enabled by default on macOS systems.

⚠ If more than one partition with macOS is detected, the csrutil command will hang awaiting input.

To check the state of the system, run the following command(s):

```
/usr/libexec/mdmclient QuerySecurityInfo 2>/dev/null | /usr/bin/grep -c
"AuthenticatedRootVolumeEnabled = 1;"
```

If the result is not **1**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

```
/usr/bin/csrutil authenticated-root enable
```

ℹ To re-enable "Authenticated Root", boot the affected system into "Recovery" mode, launch "Terminal" from the "Utilities" menu, and run the command.

ID	os_authenticated_root_enable

References	**800-53r5**	• AC-3 • CM-5 • MA-4(1) • SC-34 • SI-7, SI-7(6)
	CMMC	• AC.L1-3.1.1 • CM.L2-3.4.5 • SC.L2-3.13.11
	CCE	• CCE-95164-0

8.5. Enforce Installation of XProtect Remediator and Gatekeeper Updates Automatically

Software Update *MUST* be configured to update XProtect Remediator and Gatekeeper automatically.

This setting enforces definition updates for XProtect Remediator and Gatekeeper; with this setting in place, new malware and adware that Apple has added to the list of malware or untrusted software will not execute. These updates do not require the computer to be restarted.

https://support.apple.com/en-us/HT207005

> ℹ Software update will automatically update XProtect Remediator and Gatekeeper by default in the macOS.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.SoftwareUpdate')\
.objectForKey('ConfigDataInstall').js
EOS
```

If the result is not **true**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.SoftwareUpdate) payload type:

```
<key>ConfigDataInstall</key>
<true/>
```

ID	os_config_data_install_enforce	
References	**800-53r5**	SI-2(5) SI-3
	CMMC	SI.L1-3.14.1 SI.L1-3.14.2 SI.L1-3.14.4
	CCE	CCE-95176-4

8.6. Disable Dictation

Dictation *MUST* be disabled on Intel based Macs as the feature On Device Dictation is only available on Apple Silicon devices.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowDictation').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowDictation</key>
<false/>
```

ID	os_dictation_disable

References	**800-53r5**	AC-20 CM-7, CM-7(1) SC-7(10)
	CMMC	AC.L1-3.1.20 CM.L2-3.4.6 CM.L2-3.4.7
	CCE	CCE-95180-6

8.7. Disable FileVault Automatic Login

If FileVault is enabled, automatic login *MUST* be disabled, so that both FileVault and login window authentication are required.

The default behavior of macOS when FileVault is enabled is to automatically log in to the computer once successfully passing your FileVault credentials.

> DisableFDEAutoLogin does not have to be set on Apple Silicon based macOS systems that are smartcard enforced as smartcards are available at pre-boot.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.loginwindow')\
.objectForKey('DisableFDEAutoLogin').js
EOS
```

If the result is not **true**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.loginwindow) payload type:

```
<key>DisableFDEAutoLogin</key>
<true/>
```

ID	os_filevault_autologin_disable

References	**800-53r5**	• AC-2(11) • AC-3 • IA-5(13)
	CMMC	• AC.L1-3.1.1
	CCE	• CCE-95192-1

8.8. Enable Firmware Password

A firmware password *MUST* be enabled and set.

Single user mode, recovery mode, the Startup Manager, and several other tools are available on macOS by holding the "Option" key down during startup. Setting a firmware password restricts access to these tools.

To set a firmware passcode use the following command:

```
/usr/sbin/firmwarepasswd -setpasswd
```

> ℹ If firmware password or passcode is forgotten, the only way to reset the forgotten password is through the use of a machine specific binary generated and provided by Apple. Schedule a support call, and provide proof of purchase before the firmware binary will be generated.

> ℹ Firmware passwords are not supported on Apple Silicon devices. This rule is only applicable to Intel devices.

To check the state of the system, run the following command(s):

```
/usr/sbin/firmwarepasswd -check | /usr/bin/grep -c "Password Enabled: Yes"
```

If the result is not **1**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

> ℹ See discussion on remediation and how to enable firmware password.

ID	os_firmware_password_require

References	800-53r5	- AC-6
	CMMC	- AC.L1-3.1.1 - AC.L2-3.1.5
	CCE	- CCE-95194-7

8.9. Enable Gatekeeper

Gatekeeper *MUST* be enabled.

Gatekeeper is a security feature that ensures that applications are digitally signed by an Apple-issued certificate before they are permitted to run. Digital signatures allow the macOS host to verify that the application has not been modified by a malicious third party.

Administrator users will still have the option to override these settings on a case-by-case basis.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.systempolicy.control')\
.objectForKey('EnableAssessment').js
EOS
```

If the result is not **true**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.systempolicy.control) payload type:

```
<key>EnableAssessment</key>
<true/>
```

ID	os_gatekeeper_enable

References	**800-53r5**	- CM-14 - CM-5 - SI-3 - SI-7(1), SI-7(15)
	CMMC	- CM.L2-3.4.5 - SI.L1-3.14.1 - SI.L1-3.14.2 - SI.L1-3.14.4
	CCE	- CCE-95195-4

8.10. Disable Genmoji AI Creation

Apple Intelligence features such as Genmoji *MUST* be disabled.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowGenmoji').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowGenmoji</key>
<false/>
```

ID	os_genmoji_disable	
References	**800-53r5**	- CM-7, CM-7(1)
	CMMC	- CM.L2-3.4.6 - CM.L2-3.4.7
	CCE	- CCE-95196-2

8.11. Disable Handoff

Handoff *MUST* be disabled.

Handoff allows you to continue working on a document or project when the user switches from one Apple device to another. Disabling Handoff prevents data transfers to unauthorized devices.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowActivityContinuation').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowActivityContinuation</key>
<false/>
```

ID	os_handoff_disable	
References	**800-53r5**	• AC-20 • AC-3 • CM-7, CM-7(1)
	CMMC	• AC.L1-3.1.1 • AC.L1-3.1.20 • CM.L2-3.4.6 • CM.L2-3.4.7
	CCE	• CCE-95199-6

8.12. Secure User’s Home Folders

The system *MUST* be configured to prevent access to other user’s home folders.

The default behavior of macOS is to allow all valid users access to the top level of every other user’s home folder while restricting access only to the Apple default folders within.

To check the state of the system, run the following command(s):

```
/usr/bin/find /System/Volumes/Data/Users -mindepth 1 -maxdepth 1 -type d ! \( -perm
700 -o -perm 711 \) | /usr/bin/grep -v "Shared" | /usr/bin/grep -v "Guest" |
/usr/bin/wc -l | /usr/bin/xargs
```

If the result is not **0**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

```
IFS=$'\n'
for userDirs in $( /usr/bin/find /System/Volumes/Data/Users -mindepth 1 -maxdepth
1 -type d ! \( -perm 700 -o -perm 711 \) | /usr/bin/grep -v "Shared" |
/usr/bin/grep -v "Guest" ); do
  /bin/chmod og-rwx "$userDirs"
done
unset IFS
```

ID	os_home_folders_secure	
References	**800-53r5**	AC-6
	CMMC	AC.L1-3.1.1 AC.L1-3.1.2 AC.L2-3.1.5 AC.L2-3.1.6
	CCE	CCE-95203-6

8.13. Disable the Built-in Web Server

The built-in web server which is managed by launchd is a non-essential service built into macOS and *MUST* be disabled and not running.

ℹ The built in web server service is disabled at startup by default macOS.

To check the state of the system, run the following command(s):

```
result="FAIL"
enabled=$(/bin/launchctl print-disabled system | /usr/bin/grep '"org.apache.httpd" =>
enabled')
running=$(/bin/launchctl print system/org.apache.httpd 2>/dev/null)
```

```
if [[ -z "$running" ]] && [[ -z "$enabled" ]]; then
  result="PASS"
elif [[ -n "$running" ]]; then
  result=result+" RUNNING"
elif [[ -n "$enabled" ]]; then
  result=result+" ENABLED"
fi
echo $result
```

If the result is not **PASS**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

```
/usr/sbin/apachectl stop 2>/dev/null
/bin/launchctl disable system/org.apache.httpd
```

ID	os_httpd_disable	
References	**800-53r5**	AC-17 AC-3
	CMMC	AC.L1-3.1.1
	CCE	CCE-95204-4

8.14. Disable iCloud Storage Setup during Setup Assistant

The prompt to set up iCloud storage services during Setup Assistant *MUST* be disabled.

The default behavior of macOS is to prompt new users to set up storage in iCloud. Disabling the iCloud storage setup prompt provides organizations more control over the storage of their data.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript 2>/dev/null << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.SetupAssistant.managed')\
.objectForKey('SkipSetupItems').containsObject("iCloudStorage")
EOS
```

If the result is not **true**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.SetupAssistant.managed) payload type:

```
<key>SkipSetupItems</key>
<array>
    <string>iCloudStorage</string>
</array>
```

ID	os_icloud_storage_prompt_disable	
References	**800-53r5**	- AC-20
	CMMC	- AC.L1-3.1.20
	CCE	- CCE-95205-1

8.15. Disable Apple Intelligence Image Playground

Apple Intelligence features such as Image Playground *MUST* be disabled.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowImagePlayground').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowImagePlayground</key>
<false/>
```

ID	os_image_playground_disable

References	**800-53r5**	CM-7, CM-7(1)
	CMMC	CM.L2-3.4.6
		CM.L2-3.4.7
	CCE	CCE-95207-7

8.16. Disable iPhone Mirroring

iPhone Mirroring *MUST* be disabled to prevent file transfers to or from unauthorized devices. Disabling iPhone Mirroring also prevents potentially unauthorized applications from appearing as if they are installed on the Mac.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowiPhoneMirroring').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowiPhoneMirroring</key>
<false/>
```

ID	os_iphone_mirroring_disable	
References	**800-53r5**	AC-20
		AC-3
		CM-7, CM-7(1)
	CMMC	AC.L1-3.1.1
		AC.L1-3.1.20
		CM.L2-3.4.6
		CM.L2-3.4.7
	CCE	CCE-95212-7

8.17. Disable Apple Intelligence Mail Smart Replies

Apple Intelligence features such as Mail Smart Replies that use off device AI *MUST* be disabled.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowMailSmartReplies').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowMailSmartReplies</key>
<false/>
```

ID	os_mail_smart_reply_disable	
References	**800-53r5**	• AC-20, AC-20(1) • CM-7, CM-7(1) • SC-7(10)
	CMMC	• AC.L1-3.1.20 • CM.L2-3.4.6 • CM.L2-3.4.7
	CCE	• CCE-95222-6

8.18. Disable Apple Intelligence Mail Summary

Apple Intelligence features such as Apple Mail Summary that use off device AI *MUST* be disabled.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowMailSummary').js
```

```
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowMailSummary</key>
<false/>
```

ID	os_mail_summary_disable	
References	**800-53r5**	- AC-20, AC-20(1) - CM-7, CM-7(1) - SC-7(10)
	CMMC	- AC.L1-3.1.20 - CM.L2-3.4.6 - CM.L2-3.4.7
	CCE	- CCE-95223-4

8.19. Disable Network File System Service

Support for Network File Systems (NFS) services is non-essential and, therefore, *MUST* be disabled.

To check the state of the system, run the following command(s):

```
isDisabled=$(/sbin/nfsd status | /usr/bin/awk '/nfsd service/ {print $NF}')
if [[ "$isDisabled" == "disabled" ]] && [[ -z $(/usr/bin/pgrep nfsd) ]]; then
  echo "pass"
else
  echo "fail"
fi
```

If the result is not **pass**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

```
/bin/launchctl disable system/com.apple.nfsd
/bin/rm -rf /etc/exports
```

The system may need to be restarted for the update to take effect.

ID	os_nfsd_disable	
References	**800-53r5**	AC-17 AC-3
	CMMC	AC.L1-3.1.1
	CCE	CCE-95235-8

8.20. Disable Apple Intelligence Notes Transcription

Apple Intelligence features such as Notes Transcription that use off device AI *MUST* be disabled.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowNotesTranscription').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowNotesTranscription</key>
<false/>
```

ID	os_notes_transcription_disable

References	**800-53r5**	- AC-20, AC-20(1) - CM-7, CM-7(1) - SC-7(10)
	CMMC	- AC.L1-3.1.20 - CM.L2-3.4.6 - CM.L2-3.4.7
	CCE	- CCE-95238-2

8.21. Disable Apple Intelligence Notes Transcription Summary

Apple Intelligence features such as Notes Transcription Summary that use off device AI *MUST* be disabled.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowNotesTranscriptionSummary').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowNotesTranscriptionSummary</key>
<false/>
```

ID	os_notes_transcription_summary_disable

References	800-53r5	- AC-20, AC-20(1) - CM-7, CM-7(1) - SC-7(10)
	CMMC	- AC.L1-3.1.20 - CM.L2-3.4.6 - CM.L2-3.4.7
	CCE	- CCE-95239-0

8.22. Enforce On Device Dictation

Dictation *MUST* be restricted to on device only to prevent potential data exfiltration.

The information system *MUST* be configured to provide only essential capabilities.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('forceOnDeviceOnlyDictation').js
EOS
```

If the result is not **true**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>forceOnDeviceOnlyDictation</key>
<true/>
```

ID	os_on_device_dictation_enforce

References		
	800-53r5	AC-20 CM-7, CM-7(1) SC-7(10)
	CMMC	AC.L1-3.1.20 CM.L2-3.4.6 CM.L2-3.4.7
	CCE	CCE-95247-3

8.23. Disable Photos Enhanced Visual Search

Enhanced Visualed Search *MUST* be disabled in the Photos app.

The information system *MUST* be configured to provide only essential capabilities. Disabling Enhanced Visual Search will mitigate the risk of unwanted data being sent to Apple.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.photos.shareddefaults')\
.objectForKey('IPXDefaultEnhancedVisualSearchEnabled').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.photos.shareddefaults) payload type:

```
<key>IPXDefaultEnhancedVisualSearchEnabled</key>
<false/>
```

ID	os_photos_enhanced_search_disable

References	**800-53r5**	AC-20, AC-20(1) CM-7, CM-7(1) SC-7(10)
	CMMC	AC.L1-3.1.20 CM.L2-3.4.6 CM.L2-3.4.7
	CCE	CCE-95254-9

8.24. Enforce Rapid Security Response Mechanism

Rapid security response mechanism *MUST* be enabled.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowRapidSecurityResponseInstallation').js
EOS
```

If the result is not **true**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowRapidSecurityResponseInstallation</key>
<true/>
```

ID	os_rapid_security_response_allow	
References	**800-53r5**	SI-2, SI-2(5) SI-3
	CMMC	SI.L1-3.14.1 SI.L1-3.14.2 SI.L1-3.14.4
	CCE	CCE-95272-1

8.25. Disable User Ability from Being Able to Undo Rapid Security Responses

Rapid security response (RSR) mechanism *MUST* be enabled and the ability for the user to disable RSR *MUST* be disabled.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowRapidSecurityResponseRemoval').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowRapidSecurityResponseRemoval</key>
<false/>
```

ID	os_rapid_security_response_removal_disable	
References	**800-53r5**	• SI-2, SI-2(5) • SI-3
	CMMC	• SI.L1-3.14.1 • SI.L1-3.14.2 • SI.L1-3.14.4
	CCE	• CCE-95273-9

8.26. Enable Recovery Lock

A recovery lock password *MUST* be enabled and set.

Single user mode, recovery mode, the Startup Manager, and several other tools are available on macOS by holding down specific key combinations during startup. Setting a recovery lock restricts access to these tools.

❗ Recovery lock passwords are not supported on Intel devices. This rule is only applicable to Apple Silicon devices.

To check the state of the system, run the following command(s):

```
/usr/libexec/mdmclient QuerySecurityInfo 2>/dev/null | /usr/bin/grep -c
"IsRecoveryLockEnabled = 1"
```

If the result is not **1**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

The SetRecoveryLock command can be used to set a Recovery Lock password and must be from your MDM.

ID	os_recovery_lock_enable	
References	**800-53r5**	AC-6
	CMMC	AC.L1-3.1.1 AC.L2-3.1.5
	CCE	CCE-95277-0

8.27. Disable Root Login

To assure individual accountability and prevent unauthorized access, logging in as root at the login window *MUST* be disabled.

The macOS system *MUST* require individuals to be authenticated with an individual authenticator prior to using a group authenticator, and administrator users *MUST* never log in directly as root.

To check the state of the system, run the following command(s):

```
/usr/bin/dscl . -read /Users/root UserShell 2>&1 | /usr/bin/grep -c "/usr/bin/false"
```

If the result is not **1**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

```
/usr/bin/dscl . -create /Users/root UserShell /usr/bin/false
```

ID	os_root_disable	
References	800-53r5	IA-2, IA-2(5)
	CMMC	IA.L1-3.5.1 IA.L1-3.5.2
	CCE	CCE-95282-0

8.28. Disable Apple Intelligence Safari Reader Summary

Apple Intelligence features such as Safari Reader Summary that use off device AI *MUST* be disabled.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowSafariSummary').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowSafariSummary</key>
<false/>
```

ID	os_safari_reader_summary_disable	
References	800-53r5	AC-20, AC-20(1) CM-7, CM-7(1) SC-7(10)
	CMMC	AC.L1-3.1.20 CM.L2-3.4.6 CM.L2-3.4.7
	CCE	CCE-95286-1

8.29. Ensure System Integrity Protection is Enabled

System Integrity Protection (SIP) *MUST* be enabled.

SIP is vital to protecting the integrity of the system as it prevents malicious users and software from making unauthorized and/or unintended modifications to protected files and folders; ensures the presence of an audit record generation capability for defined auditable events for all operating system components; protects audit tools from unauthorized access, modification, and deletion; restricts the root user account and limits the actions that the root user can perform on protected parts of the macOS; and prevents non-privileged users from granting other users direct access to the contents of their home directories and folders.

> ℹ SIP is enabled by default in macOS.

To check the state of the system, run the following command(s):

```
/usr/bin/csrutil status | /usr/bin/grep -c 'System Integrity Protection status:
enabled.'
```

If the result is not **1**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

```
/usr/bin/csrutil enable
```

> ℹ To reenable "System Integrity Protection", boot the affected system into "Recovery" mode, launch "Terminal" from the "Utilities" menu, and run the command.

ID	os_sip_enable

References	**800-53r5**	AC-3 AU-9, AU-9(3) CM-5, CM-5(6) SC-4 SI-2 SI-7
	CMMC	AC.L1-3.1.1 AU.L2-3.3.8 CM.L2-3.4.5 SC.L2-3.13.4 SI.L1-3.14.1 SI.L1-3.14.4
	CCE	CCE-95298-6

8.30. Disable Siri Setup during Setup Assistant

The prompt for Siri during Setup Assistant *MUST* be disabled.

Organizations *MUST* apply organization-wide configuration settings. The macOS Siri Assistant Setup prompt guides new users through enabling their own specific Siri settings; this is not essential and, therefore, *MUST* be disabled to prevent against the risk of individuals electing Siri settings with the potential to override organization-wide settings.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript 2>/dev/null << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.SetupAssistant.managed')\
.objectForKey('SkipSetupItems').containsObject("Siri")
EOS
```

If the result is not **true**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.SetupAssistant.managed) payload type:

```
<key>SkipSetupItems</key>
<array>
    <string>Siri</string>
```

```
</array>
```

ID	os_siri_prompt_disable	
References	800-53r5	- AC-20 - CM-7, CM-7(1)
	CMMC	- AC.L1-3.1.20 - CM.L2-3.4.6 - CM.L2-3.4.7
	CCE	- CCE-95299-4

8.31. Disable Apple Intelligence During Setup Assistant

The prompt for setting up Apple Intelligence during Setup Assistant *MUST* be disabled.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript 2>/dev/null << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.SetupAssistant.managed')\
.objectForKey('SkipSetupItems').containsObject("Intelligence")
EOS
```

If the result is not **true**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.SetupAssistant.managed) payload type:

```
<key>SkipSetupItems</key>
<array>
    <string>Intelligence</string>
</array>
```

ID	os_skip_apple_intelligence_enable

References	**800-53r5**	- AC-20 - AC-4 - CM-7
	CMMC	- AC.L1-3.1.20
	CCE	- CCE-95603-7

8.32. Disable Unlock with Apple Watch During Setup Assistant

The prompt for Apple Watch unlock setup during Setup Assistant *MUST* be disabled.

Disabling Apple watches is a necessary step to ensuring that the information system retains a session lock until the user reestablishes access using an authorized identification and authentication procedures.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript 2>/dev/null << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.SetupAssistant.managed')\
.objectForKey('SkipSetupItems').containsObject("WatchMigration")
EOS
```

If the result is not **true**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.SetupAssistant.managed) payload type:

```
<key>SkipSetupItems</key>
<array>
    <string>WatchMigration</string>
</array>
```

ID	os_skip_unlock_with_watch_enable	
References	**800-53r5**	- AC-20
	CMMC	- AC.L1-3.1.20
	CCE	- CCE-95301-8

8.33. Disable Trivial File Transfer Protocol Service

If the system does not require Trivial File Transfer Protocol (TFTP), support it is non-essential and *MUST* be disabled.

The information system *MUST* be configured to provide only essential capabilities. Disabling TFTP helps prevent the unauthorized connection of devices and the unauthorized transfer of information.

ℹ TFTP service is disabled at startup by default macOS.

To check the state of the system, run the following command(s):

```
result="FAIL"
enabled=$(/bin/launchctl print-disabled system | /usr/bin/grep '"com.apple.tftpd" =>
enabled')
running=$(/bin/launchctl print system/com.apple.tftpd 2>/dev/null)

if [[ -z "$running" ]] && [[ -z "$enabled" ]]; then
  result="PASS"
elif [[ -n "$running" ]]; then
  result=result+" RUNNING"
elif [[ -n "$enabled" ]]; then
  result=result+" ENABLED"
fi
echo $result
```

If the result is not **PASS**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

```
/bin/launchctl bootout system/com.apple.tftpd
/bin/launchctl disable system/com.apple.tftpd
```

The system may need to be restarted for the update to take effect.

ID	os_tftpd_disable

References	**800-53r5**	AC-17 AC-3 IA-5(1)
	CMMC	AC.L1-3.1.1 IA.L2-3.5.7 IA.L2-3.5.8 IA.L2-3.5.9
	CCE	CCE-95323-2

8.34. Disable Login to Other User's Active and Locked Sessions

The ability to log in to another user's active or locked session *MUST* be disabled.

macOS has a privilege that can be granted to any user that will allow that user to unlock active user's sessions. Disabling the admins and/or user's ability to log into another user's active and locked session prevents unauthorized persons from viewing potentially sensitive and/or personal information.

Configuring this setting will change the user experience and disable TouchID from unlocking the screensaver. A configuration profile will be generated to include the setting that restores the expected behavior. You can also apply the settings using `/usr/bin/sudo /usr/bin/defaults write /Library/Preferences/com.apple.loginwindow screenUnlockMode -int 1`.

This rule may cause issues when platformSSO is configured.

To check the state of the system, run the following command(s):

```
RESULT="FAIL"
SS_RULE=$(/usr/bin/security -q authorizationdb read system.login.screensaver  2>&1 |
/usr/bin/xmllint --xpath "//dict/key[.='rule']/following-
sibling::array[1]/string/text()" -)

if [[ "${SS_RULE}" == "authenticate-session-owner" ]]; then
    RESULT="PASS"
else
    PSSO_CHECK=$(/usr/bin/security -q authorizationdb read "$SS_RULE"  2>&1 |
/usr/bin/xmllint --xpath '//key[.="rule"]/following-sibling::array[1]/string/text()' -
)
    if /usr/bin/grep -Fxq "authenticate-session-owner" <<<"$PSSO_CHECK"; then
        RESULT="PASS"
    fi
fi
```

```
echo $RESULT
```

If the result is not **PASS**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.loginwindow) payload type:

```
<key>screenUnlockMode</key>
<integer>1</integer>
```

ID	os_unlock_active_user_session_disable	
References	**800-53r5**	- IA-2, IA-2(5)
	CMMC	- IA.L1-3.5.1 - IA.L1-3.5.2
	CCE	- CCE-95328-1

8.35. Disable Unix-to-Unix Copy Protocol Service

The system *MUST* not have the Unix-to-Unix Copy Protocol (UUCP) service active.

UUCP, a set of programs that enable the sending of files between different UNIX systems as well as sending commands to be executed on another system, is not essential and *MUST* be disabled in order to prevent the unauthorized connection of devices, transfer of information, and tunneling.

ℹ UUCP service is disabled at startup by default macOS.

To check the state of the system, run the following command(s):

```
result="FAIL"
enabled=$(/bin/launchctl print-disabled system | /usr/bin/grep '"com.apple.uucp" =>
enabled')
running=$(/bin/launchctl print system/com.apple.uucp 2>/dev/null)

if [[ -z "$running" ]] && [[ -z "$enabled" ]]; then
  result="PASS"
elif [[ -n "$running" ]]; then
  result=result+" RUNNING"
elif [[ -n "$enabled" ]]; then
  result=result+" ENABLED"
```

```
fi
echo $result
```

If the result is not **PASS**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

```
/bin/launchctl bootout system/com.apple.uucp
/bin/launchctl disable system/com.apple.uucp
```

The system may need to be restarted for the update to take effect.

ID	os_uucp_disable	
References	**800-53r5**	- AC-17 - AC-3
	CMMC	- AC.L1-3.1.1
	CCE	- CCE-95330-7

8.36. Disable Apple Intelligence Writing Tools

Apple Intelligence features such as writing tools that use off device AI *MUST* be disabled.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowWritingTools').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowWritingTools</key>
<false/>
```

ID	os_writing_tools_disable	
References	**800-53r5**	AC-20, AC-20(1) CM-7, CM-7(1) SC-7(10)
	CMMC	AC.L1-3.1.20 CM.L2-3.4.6 CM.L2-3.4.7
	CCE	CCE-95334-9

Chapter 9. System Settings

This section contains the configuration and enforcement of the settings within the macOS System Settings application.

The check/fix commands outlined in this section *MUST* be run by a user with elevated privileges.

9.1. Disable Unattended or Automatic Logon to the System

Automatic logon *MUST* be disabled.

When automatic logons are enabled, the default user account is automatically logged on at boot time without prompting the user for a password. Even if the screen is later locked, a malicious user would be able to reboot the computer and find it already logged in. Disabling automatic logons mitigates this risk.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.loginwindow')\
.objectForKey('com.apple.login.mcx.DisableAutoLoginClient').js
EOS
```

If the result is not **true**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.loginwindow) payload type:

```
<key>com.apple.login.mcx.DisableAutoLoginClient</key>
<true/>
```

ID	system_settings_automatic_login_disable

References	**800-53r5**	- IA-2 - IA-5(13)
	CMMC	- IA.L1-3.5.1 - IA.L1-3.5.2
	CCE	- CCE-95356-2

9.2. Disable Bluetooth Sharing

Bluetooth Sharing *MUST* be disabled.

Bluetooth Sharing allows users to wirelessly transmit files between the macOS and Bluetooth-enabled devices, including personally owned cellphones and tablets. A malicious user might introduce viruses or malware onto the system or extract sensitive files via Bluetooth Sharing. When Bluetooth Sharing is disabled, this risk is mitigated.

> ℹ The check and fix are for the last logged in user. To get the last logged in user, run the following.
>
> ```
> CURRENT_USER=$( /usr/bin/defaults read
> /Library/Preferences/com.apple.loginwindow lastUserName )
> ```

To check the state of the system, run the following command(s):

```
/usr/bin/sudo -u "$CURRENT_USER" /usr/bin/defaults -currentHost read
com.apple.Bluetooth PrefKeyServicesEnabled
```

If the result is not **0**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

```
/usr/bin/sudo -u "$CURRENT_USER" /usr/bin/defaults -currentHost write
com.apple.Bluetooth PrefKeyServicesEnabled -bool false
```

ID	system_settings_bluetooth_sharing_disable

References	**800-53r5**	• AC-18(4) • AC-3 • CM-7, CM-7(1)
	CMMC	• AC.L1-3.1.1 • CM.L2-3.4.6 • CM.L2-3.4.7
	CCE	• CCE-95361-2

9.3. Enforce Critical Security Updates to be Installed

Ensure that security updates are installed as soon as they are available from Apple.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.SoftwareUpdate')\
.objectForKey('CriticalUpdateInstall').js
EOS
```

If the result is not **true**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.SoftwareUpdate) payload type:

```
<key>CriticalUpdateInstall</key>
<true/>
```

ID	system_settings_critical_update_install_enforce	
References	**800-53r5**	• SI-2
	CMMC	• SI.L1-3.14.1 • SI.L1-3.14.4
	CCE	• CCE-95363-8

9.4. Disable Sending Diagnostic and Usage Data to Apple

The ability to submit diagnostic data to Apple *MUST* be disabled.

The information system *MUST* be configured to provide only essential capabilities. Disabling the submission of diagnostic and usage information will mitigate the risk of unwanted data being sent to Apple.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
function run() {
let pref1 = $.NSUserDefaults.alloc.initWithSuiteName('com.apple.SubmitDiagInfo')\
.objectForKey('AutoSubmit').js
let pref2 = $.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowDiagnosticSubmission').js
if ( pref1 == false && pref2 == false ){
    return("true")
} else {
    return("false")
}
}
EOS
```

If the result is not **true**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.SubmitDiagInfo) payload type:

```
<key>AutoSubmit</key>
<false/>
```

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowDiagnosticSubmission</key>
<false/>
```

ID	system_settings_diagnostics_reports_disable

References	**800-53r5**	- AC-20 - SC-7(10) - SI-11
	CMMC	- AC.L1-3.1.20
	CCE	- CCE-95364-6

9.5. Disable External Intelligence Integrations

Integration with external intelligence systems *MUST* be disabled unless approved by the organization. Disabling external intelligence integration will mitigate the risk of data being sent to unapproved third party.

The information system *MUST* be configured to provide only essential capabilities.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowExternalIntelligenceIntegrations').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowExternalIntelligenceIntegrations</key>
<false/>
```

ID	system_settings_external_intelligence_disable	
References	**800-53r5**	- AC-20 - CM-7, CM-7(1)
	CMMC	- AC.L1-3.1.20 - CM.L2-3.4.6 - CM.L2-3.4.7
	CCE	- CCE-95365-3

9.6. Disable External Intelligence Integration Sign In

The ability to sign into an external intelligence systems *MUST* be disabled unless approved by the organiztion. Disabling external intelligence integration will mitigate the risk of data being sent to unapproved third party.

The information system *MUST* be configured to provide only essential capabilities.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowExternalIntelligenceIntegrationsSignIn').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowExternalIntelligenceIntegrationsSignIn</key>
<false/>
```

ID	system_settings_external_intelligence_sign_in_disable	
References	**800-53r5**	• AC-20 • CM-7, CM-7(1)
	CMMC	• AC.L1-3.1.20 • CM.L2-3.4.6 • CM.L2-3.4.7
	CCE	• CCE-95366-1

9.7. Disable Find My Service

The Find My service *MUST* be disabled.

A Mobile Device Management (MDM) solution *MUST* be used to carry out remote locking and wiping instead of Apple’s Find My service.

Apple’s Find My service uses a personal AppleID for authentication. Organizations should rely on

MDM solutions, which have much more secure authentication requirements, to perform remote lock and remote wipe.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
function run() {
  let pref1 = ObjC.unwrap(
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowFindMyDevice'))
  let pref2 = ObjC.unwrap(
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowFindMyFriends'))
  let pref3 = ObjC.unwrap(
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.icloud.managed')\
.objectForKey('DisableFMMiCloudSetting'))
  if ( pref1 == false && pref2 == false && pref3 == true ) {
    return("true")
  } else {
    return("false")
  }
}
EOS
```

If the result is not **true**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowFindMyDevice</key>
<false/>
<key>allowFindMyFriends</key>
<false/>
```

Create a configuration profile containing the following keys in the (com.apple.icloud.managed) payload type:

```
<key>DisableFMMiCloudSetting</key>
<true/>
```

ID	system_settings_find_my_disable

References	**800-53r5**	• AC-20 • CM-7, CM-7(1)
	CMMC	• AC.L1-3.1.20 • CM.L2-3.4.6 • CM.L2-3.4.7
	CCE	• CCE-95368-7

9.8. Enable macOS Application Firewall

The macOS Application Firewall is the built-in firewall that comes with macOS, and it *MUST* be enabled.

When the macOS Application Firewall is enabled, the flow of information within the information system and between interconnected systems will be controlled by approved authorizations.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.security.firewall')\
.objectForKey('EnableFirewall').js
EOS
```

If the result is not **true**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.security.firewall) payload type:

```
<key>EnableFirewall</key>
<true/>
```

ID	system_settings_firewall_enable

References	800-53r5	- AC-4 - CM-7, CM-7(1) - SC-7, SC-7(12)
	CMMC	- AC.L2-3.1.3 - CM.L2-3.4.6 - CM.L2-3.4.7 - SC.L1-3.13.1
	CCE	- CCE-95369-5

9.9. Enable Firewall Stealth Mode

Firewall Stealth Mode *MUST* be enabled.

When stealth mode is enabled, the Mac will not respond to any probing requests, and only requests from authorized applications will still be authorized.

> ❗ Enabling firewall stealth mode may prevent certain remote mechanisms used for maintenance and compliance scanning from properly functioning. Information System Security Officers (ISSOs) are advised to first fully weigh the potential risks posed to their organization before opting not to enable stealth mode.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.security.firewall')\
.objectForKey('EnableStealthMode').js
EOS
```

If the result is not **true**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.security.firewall) payload type:

```
<key>EnableStealthMode</key>
<true/>
<key>EnableFirewall</key>
<true/>
```

ID	system_settings_firewall_stealth_mode_enable	
References	**800-53r5**	CM-7, CM-7(1) SC-7, SC-7(16)
	CMMC	CM.L2-3.4.6 CM.L2-3.4.7 SC.L1-3.13.1
	CCE	CCE-95370-3

9.10. Disable Guest Access to Shared SMB Folders

Guest access to shared Server Message Block (SMB) folders *MUST* be disabled.

Turning off guest access prevents anonymous users from accessing files shared via SMB.

To check the state of the system, run the following command(s):

```
/usr/bin/defaults read /Library/Preferences/SystemConfiguration/com.apple.smb.server
AllowGuestAccess
```

If the result is not **0**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

```
/usr/sbin/sysadminctl -smbGuestAccess off
```

ID	system_settings_guest_access_smb_disable	
References	**800-53r5**	AC-2, AC-2(9)
	CMMC	AC.L1-3.1.2
	CCE	CCE-95373-7

9.11. Disable the Guest Account

Guest access *MUST* be disabled.

Turning off guest access prevents anonymous users from accessing files.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
function run() {
  let pref1 = ObjC.unwrap($.NSUserDefaults.alloc.initWithSuiteName('com.apple.MCX')\
.objectForKey('DisableGuestAccount'))
  let pref2 = ObjC.unwrap($.NSUserDefaults.alloc.initWithSuiteName('com.apple.MCX')\
.objectForKey('EnableGuestAccount'))
  if ( pref1 == true && pref2 == false ) {
    return("true")
  } else {
    return("false")
  }
}
EOS
```

If the result is not **true**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.MCX) payload type:

```
<key>DisableGuestAccount</key>
<true/>
<key>EnableGuestAccount</key>
<false/>
```

ID	system_settings_guest_account_disable	
References	**800-53r5**	• AC-2, AC-2(9)
	CMMC	• AC.L1-3.1.2
	CCE	• CCE-95374-5

9.12. Disable Sending Audio Recordings and Transcripts to Apple

The ability for Apple to store and review audio of your audio recordings and transcripts of your vocal shortcuts and voice control interactions *MUST* be disabled. This will disable "Improve Assistive Voice Features" in Privacy & Security within System Settings.

The information system *MUST* be configured to provide only essential capabilities. Disabling the submission of this information will mitigate the risk of unwanted data being sent to Apple.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.Accessibility')\
.objectForKey('AXSAudioDonationSiriImprovementEnabled').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.Accessibility) payload type:

```
<key>AXSAudioDonationSiriImprovementEnabled</key>
<false/>
```

ID	system_settings_improve_assistive_voice_disable	
References	**800-53r5**	- AC-20 - CM-7, CM-7(1) - SC-7(10)
	CMMC	- AC.L1-3.1.20 - CM.L2-3.4.6 - CM.L2-3.4.7
	CCE	- CCE-95377-8

9.13. Disable Improve Search Information to Apple

Sending data to Apple to help improve search *MUST* be disabled. This will disable "Improve Search" within Spotlight in System Settings.

The information system *MUST* be configured to provide only essential capabilities. Disabling the submission of search data will mitigate the risk of unwanted data being sent to Apple.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.assistant.support')\
.objectForKey('Search Queries Data Sharing Status').js
EOS
```

If the result is not **2**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.assistant.support) payload type:

```
<key>Search Queries Data Sharing Status</key>
<integer>2</integer>
```

ID	system_settings_improve_search_disable	
References	**800-53r5**	AC-20 CM-7, CM-7(1) SC-7(10)
	CMMC	AC.L1-3.1.20 CM.L2-3.4.6 CM.L2-3.4.7
	CCE	CCE-95378-6

9.14. Disable Improve Siri and Dictation Information to Apple

The ability for Apple to store and review audio of your Siri and Dictation interactions *MUST* be disabled.

The information system *MUST* be configured to provide only essential capabilities. Disabling the submission of Siri and Dictation information will mitigate the risk of unwanted data being sent to Apple.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.assistant.support')\
.objectForKey('Siri Data Sharing Opt-In Status').js
EOS
```

If the result is not **2**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.assistant.support) payload type:

```
<key>Siri Data Sharing Opt-In Status</key>
<integer>2</integer>
```

ID	system_settings_improve_siri_dictation_disable	
References	**800-53r5**	- AC-20 - CM-7, CM-7(1) - SC-7(10)
	CMMC	- AC.L1-3.1.20 - CM.L2-3.4.6 - CM.L2-3.4.7
	CCE	- CCE-95379-4

9.15. Disable the Internet Accounts System Preference Pane

The Internet Accounts System Setting *MUST* be disabled to prevent the addition of unauthorized internet accounts.

> Some organizations may allow the use and configuration of the built-in Mail.app, Calendar.app, and Contacts.app for organizational communication. Information System Security Officers (ISSOs) may make the risk-based decision not to disable the Internet Accounts System Preference pane to avoid losing this functionality, but they are advised to first fully weigh the potential risks posed to their organization.

To check the state of the system, run the following command(s):

```
/usr/bin/profiles show -output stdout-xml | /usr/bin/xmllint --xpath
'//key[text()="DisabledSystemSettings"]/following-sibling::*[1]' - | /usr/bin/grep -c
com.apple.Internet-Accounts-Settings.extension
```

If the result is not **1**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.systempreferences) payload type:

```
<key>DisabledSystemSettings</key>
<array>
    <string>com.apple.Internet-Accounts-Settings.extension</string>
</array>
```

ID	system_settings_internet_accounts_disable	
References	**800-53r5**	- AC-20 - CM-7, CM-7(1), CM-7(5)
	CMMC	- AC.L1-3.1.20 - CM.L2-3.4.8
	CCE	- CCE-95381-0

9.16. Disable Internet Sharing

If the system does not require Internet sharing, support for it is non-essential and *MUST* be disabled.

The information system *MUST* be configured to provide only essential capabilities. Disabling Internet sharing helps prevent the unauthorized connection of devices, unauthorized transfer of information, and unauthorized tunneling.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.MCX')\
.objectForKey('forceInternetSharingOff').js
EOS
```

If the result is not **true**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.MCX) payload type:

```
<key>forceInternetSharingOff</key>
<true/>
```

ID	system_settings_internet_sharing_disable	
References	**800-53r5**	- AC-20 - AC-4
	CMMC	- AC.L1-3.1.20 - AC.L2-3.1.3
	CCE	- CCE-95382-8

9.17. Configure Login Window to Prompt for Username and Password

The login window *MUST* be configured to prompt all users for both a username and a password.

By default, the system displays a list of known users on the login window, which can make it easier for a malicious user to gain access to someone else’s account. Requiring users to type in both their username and password mitigates the risk of unauthorized users gaining access to the information system.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.loginwindow')\
.objectForKey('SHOWFULLNAME').js
EOS
```

If the result is not **true**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.loginwindow) payload type:

```
<key>SHOWFULLNAME</key>
<true/>
```

ID	system_settings_loginwindow_prompt_username_password_enforce

References	**800-53r5**	- IA-2
	CMMC	- IA.L1-3.5.1
		- IA.L1-3.5.2
	CCE	- CCE-95387-7

9.18. Disable Media Sharing

Media sharing *MUST* be disabled.

When Media Sharing is enabled, the computer starts a network listening service that shares the contents of the user’s music collection with other users in the same subnet.

The information system *MUST* be configured to provide only essential capabilities. Disabling Media Sharing helps prevent the unauthorized connection of devices and the unauthorized transfer of information. Disabling Media Sharing mitigates this risk.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
function run() {
  let pref1 = ObjC.unwrap(
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowMediaSharing'))
  let pref2 = ObjC.unwrap(
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowMediaSharingModification'))
  if ( pref1 == false && pref2 == false ) {
    return("true")
  } else {
    return("false")
  }
}
EOS
```

If the result is not **true**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowMediaSharing</key>
<false/>
<key>allowMediaSharingModification</key>
```

```
<false/>
```

ID	system_settings_media_sharing_disabled	
References	**800-53r5**	• AC-17 • AC-3
	CMMC	• AC.L1-3.1.1
	CCE	• CCE-95388-5

9.19. Disable Personalized Advertising

Ad tracking and targeted ads *MUST* be disabled.

The information system *MUST* be configured to provide only essential capabilities. Disabling ad tracking ensures that applications and advertisers are unable to track users' interests and deliver targeted advertisements.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowApplePersonalizedAdvertising').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowApplePersonalizedAdvertising</key>
<false/>
```

ID	system_settings_personalized_advertising_disable

References	**800-53r5**	• AC-20 • CM-7, CM-7(1) • SC-7(10)
	CMMC	• AC.L1-3.1.20 • CM.L2-3.4.6 • CM.L2-3.4.7
	CCE	• CCE-95390-1

9.20. Disable Remote Apple Events

If the system does not require Remote Apple Events, support for Apple Remote Events is non-essential and *MUST* be disabled.

The information system *MUST* be configured to provide only essential capabilities. Disabling Remote Apple Events helps prevent the unauthorized connection of devices, the unauthorized transfer of information, and unauthorized tunneling.

To check the state of the system, run the following command(s):

```
/bin/launchctl print-disabled system | /usr/bin/grep -c '"com.apple.AEServer" =>
disabled'
```

If the result is not **1**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

```
/usr/sbin/systemsetup -setremoteappleevents off
/bin/launchctl disable system/com.apple.AEServer
```

Systemsetup with -setremoteappleevents flag will fail unless you grant Full Disk Access to systemsetup or its parent process. Requires supervision.

ID	system_settings_rae_disable	
References	**800-53r5**	• AC-17 • AC-3
	CMMC	• AC.L1-3.1.1
	CCE	• CCE-95392-7

9.21. Disable Screen Sharing and Apple Remote Desktop

Support for both Screen Sharing and Apple Remote Desktop (ARD) is non-essential and *MUST* be disabled.

The information system *MUST* be configured to provide only essential capabilities. Disabling screen sharing and ARD helps prevent the unauthorized connection of devices, the unauthorized transfer of information, and unauthorized tunneling.

To check the state of the system, run the following command(s):

```
result="FAIL"
enabled=$(/bin/launchctl print-disabled system | /usr/bin/grep
'"com.apple.screensharing" => enabled')
running=$(/bin/launchctl print system/com.apple.screensharing 2>/dev/null)

if [[ -z "$running" ]] && [[ -z "$enabled" ]]; then
  result="PASS"
elif [[ -n "$running" ]]; then
  result=result+" RUNNING"
elif [[ -n "$enabled" ]]; then
  result=result+" ENABLED"
fi
echo $result
```

If the result is not **PASS**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

```
/bin/launchctl bootout system/com.apple.screensharing
/bin/launchctl disable system/com.apple.screensharing
```

NOTE - This will apply to the whole system

ID	system_settings_screen_sharing_disable	
References	**800-53r5**	• AC-17 • AC-3
	CMMC	• AC.L1-3.1.1
	CCE	• CCE-95394-3

9.22. Enforce Automatic Installs of Available Security Updates using DDM.

Ensure that available security updates are installed as soon as they are available from Apple and that the user cannot modify the setting within System Settings.

To check the state of the system, run the following command(s):

```
/usr/bin/plutil -convert json
/var/db/softwareupdate/SoftwareUpdateDDMStatePersistence.plist -o - | /usr/bin/jq
--raw-output
.'SUCorePersistedStatePolicyFields.SUCoreDDMDeclarationGlobalSettings.automaticallyIns
tallSystemAndSecurityUpdates'
```

If the result is not **1**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

This is implemented by Declarative Device Management (DDM).

ID	system_settings_security_update_install	
References	**800-53r5**	- SI-2
	CMMC	- SI.L1-3.14.1 - SI.L1-3.14.4
	CCE	- CCE-95602-9

9.23. Disable Siri

Support for Siri is non-essential and *MUST* be disabled.

The information system *MUST* be configured to provide only essential capabilities.

To check the state of the system, run the following command(s):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowAssistant').js
EOS
```

If the result is not **false**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

Create a configuration profile containing the following keys in the (com.apple.applicationaccess) payload type:

```
<key>allowAssistant</key>
<false/>
```

ID	system_settings_siri_disable	
References	800-53r5	• AC-20 • CM-7, CM-7(1) • SC-7(10)
	CMMC	• AC.L1-3.1.20 • CM.L2-3.4.6 • CM.L2-3.4.7
	CCE	• CCE-95398-4

9.24. Disable Server Message Block Sharing

Support for Server Message Block (SMB) file sharing is non-essential and *MUST* be disabled.

The information system *MUST* be configured to provide only essential capabilities.

To check the state of the system, run the following command(s):

```
/bin/launchctl print-disabled system | /usr/bin/grep -c '"com.apple.smbd" => disabled'
```

If the result is not **1**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

```
/bin/launchctl disable system/com.apple.smbd
```

The system may need to be restarted for the update to take effect.

ID	system_settings_smbd_disable

References	**800-53r5**	• AC-17 • AC-3
	CMMC	• AC.L1-3.1.1
	CCE	• CCE-95401-6

9.25. Disable SSH Server for Remote Access Sessions

SSH service *MUST* be disabled for remote access.

To check the state of the system, run the following command(s):

```
result="FAIL"
enabled=$(/bin/launchctl print-disabled system | /usr/bin/grep '"com.openssh.sshd" =>
enabled')
running=$(/bin/launchctl print system/com.openssh.sshd 2>/dev/null)

if [[ -z "$running" ]] && [[ -z "$enabled" ]]; then
  result="PASS"
elif [[ -n "$running" ]]; then
  result=result+" RUNNING"
elif [[ -n "$enabled" ]]; then
  result=result+" ENABLED"
fi
echo $result
```

If the result is not **PASS**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

```
/usr/sbin/systemsetup -f -setremotelogin off >/dev/null
/bin/launchctl disable system/com.openssh.sshd
```

Systemsetup with -setremotelogin flag will fail unless you grant Full Disk Access to systemsetup or its parent process. Requires supervision.

ID	system_settings_ssh_disable

References	**800-53r5**	- AC-17 - CM-7, CM-7(1)
	CMMC	- AC.L1-3.1.1 - CM.L2-3.4.6 - CM.L2-3.4.7
	CCE	- CCE-95406-5

9.26. Enable SSH Server for Remote Access Sessions

Remote access sessions *MUST* use encrypted methods to protect unauthorized individuals from gaining access.

To check the state of the system, run the following command(s):

```
/bin/launchctl print-disabled system | /usr/bin/grep -c '"com.openssh.sshd" =>
enabled'
```

If the result is not **1**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

```
/bin/launchctl enable system/com.openssh.sshd
```

ID	system_settings_ssh_enable	
References	**800-53r5**	- AC-17 - AC-3 - CM-7, CM-7(1) - IA-2(8)
	CMMC	- AC.L1-3.1.1 - CM.L2-3.4.6 - CM.L2-3.4.7 - IA.L2-3.5.4
	CCE	- CCE-95407-3

9.27. Require Administrator Password to Modify System-Wide Preferences

The system *MUST* be configured to require an administrator password in order to modify the system-wide preferences in System Settings.

Some Preference Panes in System Settings contain settings that affect the entire system. Requiring a password to unlock these system-wide settings reduces the risk of a non-authorized user modifying system configurations.

To check the state of the system, run the following command(s):

```
authDBs=("system.preferences" "system.preferences.energysaver"
"system.preferences.network" "system.preferences.printing"
"system.preferences.sharing" "system.preferences.softwareupdate"
"system.preferences.startupdisk" "system.preferences.timemachine")
result="1"
for section in ${authDBs[@]}; do
  if [[ $(/usr/bin/security -q authorizationdb read "$section" | /usr/bin/xmllint
-xpath 'name(//*[contains(text(), "shared")]/following-sibling::*[1])' -) != "false"
]]; then
    result="0"
  fi
  if [[ $(/usr/bin/security -q authorizationdb read "$section" | /usr/bin/xmllint
-xpath '//*[contains(text(), "group")]/following-sibling::*[1]/text()' - ) != "admin"
]]; then
    result="0"
  fi
  if [[ $(/usr/bin/security -q authorizationdb read "$section" | /usr/bin/xmllint
-xpath 'name(//*[contains(text(), "authenticate-user")]/following-sibling::*[1])' -)
!= "true" ]]; then
    result="0"
  fi
  if [[ $(/usr/bin/security -q authorizationdb read "$section" | /usr/bin/xmllint
-xpath 'name(//*[contains(text(), "session-owner")]/following-sibling::*[1])' -) !=
"false" ]]; then
    result="0"
  fi
done
echo $result
```

If the result is not **1**, this is a finding.

Remediation Description

Perform the following to configure the system to meet the requirements:

```
authDBs=("system.preferences" "system.preferences.energysaver"
"system.preferences.network" "system.preferences.printing"
"system.preferences.sharing" "system.preferences.softwareupdate"
"system.preferences.startupdisk" "system.preferences.timemachine")

for section in ${authDBs[@]}; do
  /usr/bin/security -q authorizationdb read "$section" > "/tmp/$section.plist"

  class_key_value=$(/usr/libexec/PlistBuddy -c "Print :class" "/tmp/
$section.plist" 2>&1)
  if [[ "$class_key_value" == *"Does Not Exist"* ]]; then
    /usr/libexec/PlistBuddy -c "Add :class string user" "/tmp/$section.plist"
  else
    /usr/libexec/PlistBuddy -c "Set :class user" "/tmp/$section.plist"
  fi

  key_value=$(/usr/libexec/PlistBuddy -c "Print :shared" "/tmp/$section.plist"
2>&1)
  if [[ "$key_value" == *"Does Not Exist"* ]]; then
    /usr/libexec/PlistBuddy -c "Add :shared bool false" "/tmp/$section.plist"
  else
    /usr/libexec/PlistBuddy -c "Set :shared false" "/tmp/$section.plist"
  fi

  auth_user_key=$(/usr/libexec/PlistBuddy -c "Print :authenticate-user"
"/tmp/$section.plist" 2>&1)
  if [[ "$auth_user_key" == *"Does Not Exist"* ]]; then
    /usr/libexec/PlistBuddy -c "Add :authenticate-user bool true" "/tmp/
$section.plist"
  else
    /usr/libexec/PlistBuddy -c "Set :authenticate-user true" "/tmp/$section.plist"
  fi

  session_owner_key=$(/usr/libexec/PlistBuddy -c "Print :session-owner"
"/tmp/$section.plist" 2>&1)
  if [[ "$session_owner_key" == *"Does Not Exist"* ]]; then
    /usr/libexec/PlistBuddy -c "Add :session-owner bool false" "/tmp/
$section.plist"
  else
    /usr/libexec/PlistBuddy -c "Set :session-owner false" "/tmp/$section.plist"
  fi

  group_key=$(/usr/libexec/PlistBuddy -c "Print :group" "/tmp/$section.plist"
2>&1)
  if [[ "$group_key" == *"Does Not Exist"* ]]; then
    /usr/libexec/PlistBuddy -c "Add :group string admin" "/tmp/$section.plist"
  else
    /usr/libexec/PlistBuddy -c "Set :group admin" "/tmp/$section.plist"
  fi

  /usr/bin/security -q authorizationdb write "$section" < "/tmp/$section.plist"
```

```
done
```

ID	system_settings_system_wide_preferences_configure	
References	**800-53r5**	• AC-6, AC-6(1), AC-6(2)
	CMMC	• AC.L1-3.1.1 • AC.L2-3.1.5 • AC.L2-3.1.6
	CCE	• CCE-95408-1

Chapter 10. Inherent

This section reviews the controls that are built-in to macOS, and cannot be configured out of compliance.

10.1. Enforce Approved Authorization for Logical Access

The information system *IS* configured to enforce an approved authorization process before granting users logical access.

The inherent configuration of the macOS does not grant users logical access without authorization. Authorization is achieved on the macOS through permissions, which are controlled at many levels, from the Mach and BSD components of the kernel, through higher levels of the operating system and, for networked applications, through the networking protocols. Permissions can be granted at the level of directories, subdirectories, files or applications, or specific data within files or functions within applications.

https://developer.apple.com/library/archive/documentation/Security/Conceptual/AuthenticationAndAuthorizationGuide/Permissions/Permissions.html

The technology supports this requirement and cannot be configured to be out of compliance. The technology inherently meets this requirement.

ID	os_logical_access	
References	**800-53r5**	• AC-3
	CMMC	• AC.L1-3.1.1

10.2. Ensure the System Implements Malicious Code Protection Mechanisms

The inherent configuration of the macOS *IS* in compliance as Apple has designed the system with three layers of protection against malware. Each layer of protection is comprised of one or more malicious code protection mechanisms, which are automatically implemented and which, collectively, meet the requirements of all applicable federal laws, Executive Orders, directives, policies, regulations, standards, and guidance for malicious code prevention.

1. This first layer of defense targets the distribution of malware; the aim is to prevent malware from ever launching. The following mechanisms are inherent to the macOS design and constitute the first layer of protection against malicious code:
   - The Apple App Store: the safest way to add new applications to a Mac is by downloading them from the App Store; all apps available for download from the App Store have been reviewed for signs of tampering and signed by Apple to indicate that the app meets security requirements and does not contain malware.

   - XProtect: a built-in, signature-based, anti-virus, anti-malware technology inherent to all Macs. XProtect automatically detects and blocks the execution of known malware.
   - In macOS 10.15 and all subsequent releases, XProtect checks for known malicious content when:
   - an app is first launched,
   - an app has been changed (in the file system), and
   - XProtect signatures are updated.
   - YARA: another built-in tool (inherent to all Macs), which conducts signature-based detection of malware. Apple updates YARA rules regularly.
   - Gatekeeper: a security feature inherent to all Macs; Gatekeeper scans apps to detect malware and/or revocations of a developer’s signing certificate and prevents unsafe apps from running.
   - Notarization: Apple performs regular, automated scans to detect signs of malicious content and to verify developer ID-signed software; when no issues are found, Apple notarizes the software and delivers the results of scans to the system owner.

2. The second layer of defense targets malware that manages to appear on a Mac before it runs; the aim is to quickly identify and block any malware present on a Mac in order to prevent the malware from running and further spreading. The following mechanisms are inherent to the macOS design and constitute the second layer of protection against malicious code:
   - XProtect (defined above).
   - Gatekeeper (defined above).
   - Notarization (defined above).
3. The third layer of defense targets infected Mac system(s); the aim is to remediate Macs on which malware has managed to successfully execute. The following mechanism is inherent to the macOS design and constitutes the third layer of protection against malicious code:
   - Apple’s XProtect: a technology included on all macOS systems. XProtect will remediate infections upon receiving updated information delivered and when infections are detected

https://support.apple.com/guide/security/protecting-against-malware-sec469d47bd8/1/web/1

https://support.apple.com/guide/security/app-security-overview-sec35dd877d0/web

The technology supports this requirement and cannot be configured to be out of compliance. The technology inherently meets this requirement.

ID	os_malicious_code_prevention	
References	**800-53r5**	SI-3
	CMMC	SI.L1-3.14.1 SI.L1-3.14.2 SI.L1-3.14.4

Chapter 11. Permanent Findings

This section contains the controls that are defined in NIST 800-53 revision 5 but are unable to be configured natively within macOS. It is recommended to implement a third-party solution to meet the controls in this section.

11.1. Must Authenticate Before Establishing a Connection

Organizational devices requiring unique device-to-device identification and authentication may be defined by type, by device, or by a combination of type/device. Information systems typically use either shared known information (e.g., Media Access Control [MAC] or Transmission Control Protocol/Internet Protocol [TCP/IP] addresses) for device identification or organizational authentication solutions (e.g., IEEE 802.1x and Extensible Authentication Protocol [EAP], Radius server with EAP-Transport Layer Security [TLS] authentication, Kerberos) to identify/authenticate devices on local and/or wide area networks. Organizations determine the required strength of authentication mechanisms by the security categories of information systems. Because of the challenges of applying this control on large scale, organizations are encouraged to only apply the control to those limited number (and type) of devices that truly need to support this capability.

The technology does support this requirement, however, third party solutions are required to implement at an infrastructure level.

ID	os_auth_peripherals	
References	**800-53r5**	• IA-3
	CMMC	• IA.L1-3.5.2

Chapter 12. Supplemental

This section provides additional information to support the guidance provided by the baselines.

12.1. Out of Scope Supplemental

There are several requirements defined in National Institute of Standards and Technology (NIST) Special Publication (SP) 800-53, Security and Privacy Controls for Information Systems and Organizations, Revision 5 that can be met by making configuration changes to the operating system. However, NIST SP 800-53 (Rev. 5) contains a broad set of guidelines that attempt to address all aspects of an information system or systems within an organization. Because the macOS Security Compliance Project is tailored specifically to macOS, some requirements defined in NIST SP 800-53 (Rev. 5) are not applicable.

This supplemental contains those controls that are assigned to a baseline in NIST SP 800-53 (Rev. 5) which cannot be addressed with a technical configuration for macOS. These controls can be accomplished though administrative or procedural processes within an organization or via integration of the macOS system into enterprise information systems which are configured to protect the systems within.

Family	Access Control (AC)
Controls	AC-1, AC-2, AC-3(14), AC-14, AC-17(4), AC-22

Family	Awareness and Training (AT)
Controls	AT-1, AT-2, AT-3, AT-4

Family	Audit and Accountability (AU)
Controls	AU-1, AU-6, AU-9(2)

Family	Security Assessment and Authorization (CA)
Controls	CA-1, CA-2, CA-3, CA-3(6), CA-5, CA-6, CA-7, CA-7(4), CA-9

Family	Configuration Management (CM)
Controls	CM-1, CM-4, CM-8, CM-10, CM-11

Family	Contingency Planning (CP)
Controls	CP-1, CP-2, CP-3, CP-4, CP-9, CP-10

Family	Identification and Authentication (IA)
Controls	IA-1, IA-8(1), IA-8(2), IA-8(3), IA-8(4)

Family	Incident Response (IR)
Controls	IR-1, IR-2, IR-4, IR-5, IR-6, IR-7, IR-8

Family	Maintenance (MA)
Controls	MA-1, MA-2, MA-5

Family	Media Protection (MP)
Controls	MP-1, MP-2, MP-6, MP-7

Family	Physical and Environmental Protection (PE)
Controls	PE-1, PE-2, PE-3, PE-6, PE-8, PE-12, PE-13, PE-14, PE-15, PE-16

Family	Planning (PL)
Controls	PL-1, PL-2, PL-4

Family	Personnel Security (PS)
Controls	PS-1, PS-2, PS-3, PS-4, PS-5, PS-6, PS-7, PS-8

Family	Risk Assessment (RA)
Controls	RA-1, RA-2, RA-3, RA-5

Family	System and Services Acquisition (SA)
Controls	SA-1, SA-2, SA-3, SA-4, SA-4(10), SA-5, SA-9

Family	System and Communications Protection (SC)
Controls	SC-1, SC-7(3), SC-7(7), SC-7(8), SC-7(18), SC-7(21), SC-12, SC-12(1), SC-20, SC-22, SC-23

Family	System and Information Integrity (SI)
Controls	SI-1, SI-4, SI-4(2), SI-4(4), SI-4(5), SI-4(12), SI-4(14), SI-4(20), SI-4(22), SI-5, SI-7(2), SI-8(2), SI-12

12.2. FileVault Supplemental

The supplemental guidance found in this section is applicable for the following rules: * system_settings_filevault_enforce

In macOS the internal Apple File System (APFS) data volume can be protected by FileVault. The system volume is always cryptographically protected (T2 and Apple Silicon) and is a read-only volume.

> FileVault uses an AES-XTS data encryption algorithm to protect full volumes of internal and external storage. Macs with a secure enclave (T2 and Apple Silicon) utilize the hardware security features of the architecture.

FileVault is described in detail here: https://support.apple.com/guide/security/volume-encryption-with-filevault-sec4c6dc1b6e/web.

FileVault can be enabled in two ways within the macOS. It can be managed using the fdesetup command or by a Configuration Profile. When enabling FileVault via either of the aforementioned methods, you will be required to enter a username and password, which must be a local Open Directory account with a valid SecureToken password.

Using the fdesetup Command

When enabling FileVault via the command line in the Terminal application, you can run the following command.

```
/usr/bin/fdesetup enable
```

Running this command will prompt you for a username and password and then enable FileVault and return the personal recovery key. There are a number of management features available when managing FileVault via the command line that are not available when using a configuration profile. More information on these management features is available in the man page for `fdesetup`.

> Apple has deprecated `fdesetup` command line tool from recognizing user name and password for security reasons and may remove the ability in future versions of macOS.

Using a Configuration Profile

When managing FileVault with a configuration profile, you must deploy a profile with the payload type `com.apple.MCX.FileVault2`. When using the Enable key to enable FileVault with a configuration profile, you must include 1 of the following:

```
<key>Enable</key>
<string>On</string>
<key>Defer</key>
<true/>
```

```
<key>Enable</key>
<string>On</string>
<key>UserEntersMissingInfo</key>
<true/>
```

If using the Defer key it will prompt for the user name and password at logout.

The `UserEntersMissingInfo` key will only work if installed through manual installation, and it will prompt for the username and password immediately.

When using a configuration profile, you can escrow the Recovery key to a Mobile Device Management (MDM) server. Documentation for that can be found on Apple’s Developer site: https://developer.apple.com/documentation/devicemanagement/fderecoverykeyescrow.

It's recommended that you use a Personal Recovery key instead of an Institutional key as it will generate a specific key for each device. You can find more guidance on choosing a recover key here: https://docs.jamf.com/technical-papers/jamf-pro/administering-filevault-macos/10.7.1/Choosing_a_Recovery_Key.html.

> On Intel Macs, FileVault only supports password-based unlock and cannot be done using a smartcard. Smartcard unlock for FileVault is supported on Apple Silicon Macs.

12.3. Packet Filter (pf) Supplemental

The supplemental guidance found in this section is applicable for the following rules:

- os_firewall_default_deny_require

macOS contains an application layer firewall (ALF) and a packet filter (PF) firewall.

- The ALF can block incoming traffic on a per-application basis and prevent applications from gaining control of network ports, but it cannot be configured to block outgoing traffic.
  - More information on the ALF can be found here: https://support.apple.com/en-ca/HT201642
- The PF firewall can manipulate virtually any packet data and is highly configurable.
  - More information on the BF firewall can be found here: https://www.openbsd.org/faq/pf/index.html

Below is a script that configures ALF and the PF firewall to meet the requirements defined in NIST SP 800-53 (Rev. 5). The script will make sure the application layer firewall is enabled, set logging to "detailed", set built-in signed applications to automatically receive incoming connections, and set downloaded signed applications to automatically receive incoming connections. It will then create a custom rule set and copy `com.apple.pfctl.plis` from `/System/Library/LaunchDaemons/` into the `/Library/LaunchDaemons` folder and name it `800-53.pfctl.plist`. This is done to not conflict with the system's pf ruleset.

The custom pf rules are created at `/etc/pf.anchors/800_53_pf_anchors`.

The ruleset will block connections on the following ports:

Port	Service
548	Apple File Protocol (AFP)
1900	Bonjour
79	Finger
20, 21	File Transfer Protocol (FTP)
80	HTTP
icmp	ping
143	Internet Message Access Protocol (IMAP)

Port	Service
993	Internet Message Access Protocol over SSL (IMAPS)
3689	Music Sharing
5353	mDNSResponder
2049	Network File System (NFS)
49152	Optical Media Sharing
110	Post Office Protocol (POP3)
995	Post Office Protocol Secure (POP3S)
631	Printer Sharing
3031	Remote Apple Events
5900	Screen Sharing
137, 138, 138, 445	Samba (SMB)
25	Simple Mail Transfer Protocol (SMTP)
22	Secure Shell (SSH)
23	Telnet
69	Trivial File Transfer Protocol (TFTP)
540	Unix-to-Unix Copy (UUCP)

For more on configuring the PF firewall check out the man pages on `pf.conf` and `pfctl`.

```
#!/bin/bash
# Title          : enablePF-mscp.sh
# Description    : This script will configure the packet filter `pf` with the settings recommended by the macOS Security Compliance Project (MSCP)
# Author         : Dan Brodjieski
# Date           : 2023-10-05
# Version        : 1.0
# Usage          : enablePF-mscp.sh [--uninstall]
# Notes          : Script must be run with privileges
#                : Configuring `pf` with a content filter installed may have unexpected results
# Changelog      : 2023-10-05 - Added --uninstall parameter, refactored script for better functionality

#### verify running as root
if [[ $EUID -ne 0 ]]; then
    echo "This script must be run as root or with sudo, exiting..."
    exit 1
fi

#### Setup environment
launchd_pfctl_plist="/Library/LaunchDaemons/mscp.pfctl.plist"
```

```
legacy_launchd_plist="/Library/LaunchDaemons/macsec.pfctl.plist"

mdm_managed=$(/usr/bin/osascript -l JavaScript -e "
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.security.firewall').objectIsForced
ForKey('EnableFirewall')")

#### Functions ####

#enabling macos application firewall
enable_macos_application_firewall () {
    echo "The macOS application firewall is not managed by a profile, enabling from
CLI"
    /usr/libexec/ApplicationFirewall/socketfilterfw --setglobalstate on
    /usr/libexec/ApplicationFirewall/socketfilterfw --setloggingopt detail
    /usr/libexec/ApplicationFirewall/socketfilterfw --setallowsigned on
    /usr/libexec/ApplicationFirewall/socketfilterfw --setallowsignedapp on

}

#enabling pf firewall with mscp rules
enable_pf_firewall_with_mscp_rules () {
    echo "Creating LaunchDeamon to load the MSCP rules"
    if [[ -e "$launchd_pfctl_plist" ]]; then
        echo "LaunchDaemon already exists, flushing and reloading rules..."
        pfctl -e 2> /dev/null
        pfctl -f /etc/pf.conf 2> /dev/null
        return 0
    fi

    # copy system provided launchd for custom ruleset
    cp "/System/Library/LaunchDaemons/com.apple.pfctl.plist" "$launchd_pfctl_plist"
    #allow pf to be enabled when the job is loaded
    /usr/libexec/PlistBuddy -c "Add :ProgramArguments:1 string -e"
$launchd_pfctl_plist
    #use new label to not conflict with System's pfctl
    /usr/libexec/PlistBuddy -c "Set :Label mscp.pfctl" $launchd_pfctl_plist

    # enable the firewall
    pfctl -e 2> /dev/null

    #make pf run at system startup
    launchctl enable system/mscp.pfctl
    launchctl bootstrap system $launchd_pfctl_plist

    pfctl -f /etc/pf.conf 2> /dev/null #flush the pf ruleset (reload the rules)

}

# append the mscp anchors to pf.conf
configure_pf_config_add_mscp_anchors () {
    echo "Adding the MSCP anchors to /etc/pf.conf"
    # check to see if mscp anchors exists
    anchors_exist=$(grep -c '^anchor "mscp_pf_anchors"' /etc/pf.conf)

    if [[ $anchors_exist == "0" ]];then
        echo 'anchor "mscp_pf_anchors"' >> /etc/pf.conf
        echo 'load anchor "mscp_pf_anchors" from "/etc/pf.anchors/mscp_pf_anchors"' >>
/etc/pf.conf
    else
        echo "mscp anchors exist, continuing..."
    fi

}


# Create /etc/pf.anchors/mscp_pf_anchors
create_mscp_pf_anchors () {
    echo "Creating the MSCP anchor configuration file"
if [[ -e /etc/pf.anchors/mscp_pf_anchors ]]; then
    echo "mscp Anchor file exists, deleting and recreating..."
    rm -f /etc/pf.anchors/mscp_pf_anchors
fi


cat > /etc/pf.anchors/mscp_pf_anchors <<'ENDCONFIG'

anchor mscp_pf_anchors

#default deny all in, allow all out and keep state
block in all
pass out all keep state

#pass in all packets from localhost
pass in from 127.0.0.1

## Allow DHCP
pass in inet proto udp from port 67 to port 68
pass in inet6 proto udp from port 547 to port 546

## Allow incoming SSH
pass in proto tcp to any port 22

#apple file service --port 548-- pf firewall rule
block in log proto tcp to any port { 548 }

#bonjour component SSDP --port 1900-- pf firewall rule
block log proto udp to any port 1900

#finger --port 79-- pf firewall rule
block log proto tcp to any port 79

#ftp --ports 20 21-- pf firewall rule
```

```
block in log proto { tcp udp } to any port { 20 21 }

#http --port 80-- pf firewall rule
block in log proto { tcp udp } to any port 80

#icmp pf firewall rule
block in log proto icmp

#imap --port 143-- pf firewall rule
block in log proto tcp to any port 143

#imaps --port 993-- pf firewall rule
block in log proto tcp to any port 993

#iTunes sharing --port 3689-- pf firewall rule
block log proto tcp to any port 3689

#mDNSResponder --port 5353-- pf firewall rule
block log proto udp to any port 5353

#nfs --port 2049-- pf firewall rule
block log proto tcp to any port 2049

#optical drive sharing --port 49152-- pf firewall rule
block log proto tcp to any port 49152

#pop3 --port 110-- pf firewall rule
block in log proto tcp to any port 110

#pop3s --port 995-- pf firewall rule
block in log proto tcp to any port 995

#remote apple events --port 3031-- pf firewall rule
block in log proto tcp to any port 3031

#screen_sharing --port 5900-- pf firewall rule
block in log proto tcp to any port 5900
#allow screen sharing from localhost while tunneled via SSH
pass in quick on lo0 proto tcp from any to any port 5900

#smb --ports 139 445 137 138-- pf firewall rule
block in log proto tcp to any port { 139 445 }
block in log proto udp to any port { 137 138 }

#smtp --port 25-- pf firewall rule
block in log proto tcp to any port 25

#telnet --port 23-- pf firewall rule
block in log proto { tcp udp } to any port 23

#tftp --port 69-- pf firewall rule
```

```
block log proto { tcp udp } to any port 69

#uucp --port 540-- pf firewall rule
block log proto tcp to any port 540

ENDCONFIG
}

# function to remove legacy setup if exists
remove_macsec_setup() {
    echo "References to macsec appear to exist, removing..."

    launchctl disable system/macsec.pfctl
    launchctl bootout system $legacy_launchd_plist
    rm -rf $legacy_launchd_plist

    # check to see if macsec anchors exists
    anchors_exist=$(grep -c '^anchor "macsec_pf_anchors"' /etc/pf.conf)

    if [[ ! $anchors_exist == "0" ]];then
        sed -i "" '/macsec/d' /etc/pf.conf
    else
        echo "macsec anchors do not exist, continuing..."
    fi

    rm -f /etc/pf.anchors/macsec_pf_anchors
}

uninstall_mscp_pf(){
    echo "Removing MSCP configuration files from pf"
    if [[ -e "$launchd_pfctl_plist" ]]; then
        echo "LaunchDaemon exists, unloading and removing"
        #remove mscp pf components from launchd
        launchctl disable system/mscp.pfctl
        launchctl bootout system $launchd_pfctl_plist
        rm -rf $launchd_pfctl_plist
    fi

    # check to see if mscp anchors exists
    anchors_exist=$(grep -c '^anchor "mscp_pf_anchors"' /etc/pf.conf)

    if [[ ! $anchors_exist == "0" ]];then
        sed -i "" '/mscp/d' /etc/pf.conf
    else
        echo "mscp anchors do not exist, continuing..."
    fi

    rm -f /etc/pf.anchors/mscp_pf_anchors

    # flush rules and reload pf
    echo "Flushing rules and reloading pf"
    pfctl -f /etc/pf.conf 2> /dev/null #flush the pf ruleset (reload the rules)

}

#### Main Script ####

POSITIONAL_ARGS=()

while [[ $# -gt 0 ]]; do
  case $1 in
    -u|--uninstall)
      UNINSTALL="true"
      shift # past argument
      shift # past value
      ;;
    -*|--*)
      echo "Unknown option $1"
      exit 1
      ;;
    *)
      POSITIONAL_ARGS+=("$1") # save positional arg
      shift # past argument
      ;;
  esac
done

set -- "${POSITIONAL_ARGS[@]}" # restore positional parameters

if [[ $UNINSTALL == "true" ]]; then
    if [[ -e "$legacy_launchd_plist" ]]; then
        remove_macsec_setup
    fi
    uninstall_mscp_pf
    exit 0
fi

# check to see if a profile has enabled the firewall.  If it hasn't, then CLI can be used to enable
if [[ "$mdm_managed" == "false" ]];then
     enable_macos_application_firewall
fi

# clean up any legacy configurations
if [[ -e "$legacy_launchd_plist" ]]; then
    echo "References to macsec appear to exist, removing..."
    remove_macsec_setup
fi

# create mscp anchors file
create_mscp_pf_anchors
```

```
# add the anchors to the /etc/pf.conf file
configure_pf_config_add_mscp_anchors

# create specific launch daemon for mscp configuration
enable_pf_firewall_with_mscp_rules
```

12.4. Password Policy Supplemental

To comply with Executive Order 14028, “Improving the Nation’s Cybersecurity”, OMB M-22-09, “Moving the U.S. Government Toward Zero Trust Cybersecurity Principles”, and NIST SP-800-63b, “Digital Identity Guidelines: Authentication and Lifecycle Management” federal, military, and intelligence communities must adopt the following configuration settings:

- Password policies must not require the use of complexity policies such as upper characters, lower characters, or special characters.
- Password policies must also not require the use of regular rotation.

In accordance with these requirements, the following rules, while they remain on specific benchmarks, have been removed from any of the NIST 800-53r5 baselines as recommendations.

- pwpolicy_alpha_numeric_enforce
- pwpolicy_custom_regex_enforce
- pwpolicy_lower_case_character_enforce.yaml
- pwpolicy_max_lifetime_enforce
- pwpolicy_minimum_lifetime_enforce
- pwpolicy_prevent_dictionary_words
- pwpolicy_simple_sequence_disable
- pwpolicy_special_character_enforce
- pwpolicy_upper_case_character_enforce.yaml

If an organization has requirements to implement additional password policies, the remainder of this supplemental discusses the following password policy rules:

- pwpolicy_lower_case_character_enforce
- pwpolicy_upper_case_character_enforce
- pwpolicy_account_inactivity_enforce
- pwpolicy_minimum_lifetime_enforce

Password policies should be enforced as much as possible via Configuration Profiles. However, the following policies are currently not enforceable via Configuration Profiles, and must therefore be enabled using the `pwpolicy` command:

- Enforcing at least 1 lowercase character
- Enforcing at least 1 uppercase character

- Disabling an account after 35 days of inactivity
- Password minimum lifetime

To set the local policy to meet these requirements, save the following XML password policy to a file.

```
  <?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>policyCategoryAuthentication</key>
    <array>
      <dict>
        <key>policyContent</key>
        <string>policyAttributeLastAuthenticationTime &gt; policyAttributeCurrentTime
- (policyAttributeInactiveDays * 24 * 60 * 60)</string>
        <key>policyIdentifier</key>
        <string>Inactive Account</string>
        <key>policyParameters</key>
        <dict>
          <key>policyAttributeInactiveDays</key>
          <integer>35</integer>
        </dict>
      </dict>
    </array>
    <key>policyCategoryPasswordContent</key>
    <array>
      <dict>
        <key>policyContent</key>
        <string>policyAttributePassword matches '(.*[A-Z].*){1,}+'</string>
        <key>policyIdentifier</key>
        <string>Must have at least 1 uppercase letter</string>
        <key>policyParameters</key>
        <dict>
          <key>minimumAlphaCharactersUpperCase</key>
          <integer>1</integer>
        </dict>
      </dict>
      <dict>
        <key>policyContent</key>
        <string>policyAttributePassword matches '(.*[a-z].*){1,}+'</string>
        <key>policyIdentifier</key>
        <string>Must have at least 1 lowercase letter</string>
        <key>policyParameters</key>
        <dict>
          <key>minimumAlphaCharactersLowerCase</key>
          <integer>1</integer>
        </dict>
      </dict>
      <dict>
```

```
        <key>policyContent</key>
        <string>policyAttributeLastPasswordChangeTime &lt; policyAttributeCurrentTime
- (policyAttributeMinimumLifetimeHours * 60 * 60)</string>
        <key>policyIdentifier</key>
        <string>Minimum Password Lifetime</string>
        <key>policyParameters</key>
        <dict>
          <key>policyAttributeMinimumLifetimeHours</key>
          <integer>24</integer>
        </dict>
      </dict>
    </array>
  </dict>
</plist>
```

Run the following command to load the new policy file, substituting the path to the file in place of "$pwpolicy_file".

```
/usr/bin/pwpolicy setaccountpolicies $pwpolicy_file
```

If directory services is being utilized, password policies should come from the domain.

In order to apply any password policy, the `allowPasscodeModification` setting in `com.apple.applicationaccess` must not be set to `false`.

12.5. Smartcard Supplemental

The supplemental guidance found in this section is applicable for the following rules:

- auth_ssh_password_authentication_disable
- auth_smartcard_enforce
- auth_smartcard_certificate_trust_enforce_moderate
- auth_smartcard_certificate_trust_enforce_high
- auth_smartcard_allow
- auth_pam_sudo_smartcard_enforce
- auth_pam_su_smartcard_enforce
- auth_pam_login_smartcard_enforce

macOS supports smartcards, such as U.S. Personal Identity Verification (PIV) cards and U.S. Department of Defense Common Access Cards (CAC). Smartcards can be used on a macOS for the following:

- Authentication (Loginwindow, Screensaver, SSH, PKINIT, Safari, Finder, and PAM Authorization (`sudo`, `login`, and `su`))

- Digital Encryption
- Digital Signing
- Remote Access (VPN:L2TP)
- Port-based Network Access Control (802.1X)
- Keychain Unlock

macOS has built-in support for USB CCID class-compliant smartcard readers.

Smartcard Pairing

The default method for using smartcards in macOS is a method called "local account pairing". Local account pairing is automatically initiated when a user inserts a smartcard into the Mac. The user is prompted to pair their smartcard with their account. If a user receives a new smartcard, the previous card must be unpaired, and the new card paired to the account. Local account pairing employs fixed key mapping with the hash of a public key on the user's smartcard with a local account.

Smartcard Attribute Mapping

Smartcards can be used to authenticate against a directory via attribute mapping configured in `/private/etc/SmartcardLogin.plist`. This file takes precedence over local account pairing. Attribute mapping matches the configured certificate field values from the smart card to the value in a directory. This may be used with network accounts, mobile accounts, or local accounts.

Smartcard Management in macOS

The following settings are available to manage smartcards (com.apple.security.smartcard):

Key	Type	Value
userPairing	bool	If false, users will not get the pairing dialog, although existing pairings will still work.
allowSmartCard	bool	If false, the SmartCard is disabled for logins, authorizations, and screensaver unlocking. It is still allowed for other functions, such as signing emails and web access. A restart is required for a change of setting to take effect.

Key	Type	Value
checkCertificateTrust	int	Valid values are 0-3: • 0: certificate trust check is turned off • 1: certificate trust check is turned on. Standard validity check is being performed but this does not include additional revocation checks. • 2: certificate trust check is turned on, and a soft revocation check is performed. Until the certificate is explicitly rejected by CRL/OCSP, it is considered valid. This implies that unavailable/unreachable CRL/OCSP allows this check to succeed. • 3: certificate trust check is turned on, plus a hard revocation check is performed. Unless CRL/OCSP explicitly states that "this certificate is OK", the certificate is considered invalid. This is the most secure value for this setting.
oneCardPerUser	bool	If true, a user can pair with only one smartcard, although existing pairings will be allowed if already set up.
enforceSmartCard	bool	If true, a user can only login or authenticate with a smartcard.
tokenRemovalAction	int	If 1, the screen saver will automatically when the smartcard is removed.
allowUnmappedUsers	int	If 1, allows users who are in a directory group to be exempt from smartcard-only enforcement. The group allowed for exemption is defined in /private/etc/SmartcardLogin.plist

A custom configuration profile (`com.apple.loginwindow`) should be created to disable automatic login when FileVault is enabled. This ensures that authorized users boot their Macs, enter a password at the pre-boot screen (which decrypts the boot volume), and are then presented with a login window where they can authenticate with a smartcard.

Key	Type	Value
DisableFDEAutoLogin	bool	If true, both Extensible Firmware Interface (EFI) login password and loginwindow PIN are required.

> DisableFDEAutoLogin does not have to be set on Apple Silicon based macOS systems that are smartcard enforced as smartcards are available at pre-boot.

Trusted Authorities

The macOS allows users to specify which certificate authorities (CA) can be used for trust evaluation during smartcard authentication. Only CAs listed in the TrustedAuthorities section of the SmartcardLogin.plist will be evaluated as trusted. This setting only works if `checkCertificateTrust` is set to either 1, 2, or 3 in `com.apple.security.smartcard`.

To get the SHA-256 hash in the correct format, run the following command within terminal:

```
/usr/bin/openssl x509 -noout -fingerprint -sha256 -inform pem -in <issuer cert> |
/usr/bin/awk -F '=' '{print $2}' |  /usr/bin/sed 's/://g'
```

To configure Trusted Authorities, the SmartcardLogin.plist should be minimally configured as below:

```
<?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>
</dict>
</plist>
```

Smartcard Enforcement Exemption

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 smartcard enforcement. In order 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>
```

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`

Temporary Exemption

On an Apple Silicon Mac, if a temporary exemption is needed, `security filevault skip-sc-enforcement` will disable smartcard 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}'
```

Pluggable Authentication Module (PAM)

Terminal sessions in macOS can be configured for smartcard enforcement by modifying the PAM modules for `sudo`, `su`, and `login`.

```
/etc/pam.d/sudo
# sudo: auth account password session
auth        sufficient    pam_smartcard.so
auth        required      pam_opendirectory.so
auth        required      pam_deny.so
account     required      pam_permit.so
password    required      pam_deny.so
session     required      pam_permit.so
```

```
/etc/pam.d/su
# su: auth account password session
auth        sufficient    pam_smartcard.so
auth        required      pam_rootok.so
auth        required      pam_group.so no_warn group=admin,wheel ruser root_only fail_safe
account     required      pam_permit.so
account     required      pam_opendirectory.so no_check_shell
password    required      pam_opendirectory.so
session     required      pam_launchd.so
```

```
/etc/pam.d/login
# login: auth account password session
auth        sufficient    pam_smartcard.so
auth        optional      pam_krb5.so use_kcminit
auth        optional      pam_ntlm.so try_first_pass
auth        optional      pam_mount.so try_first_pass
auth        required      pam_opendirectory.so try_first_pass
auth        required      pam_deny.so
account     required      pam_nologin.so
account     required      pam_opendirectory.so
password    required      pam_opendirectory.so
session     required      pam_launchd.so
session     required      pam_uwtmp.so
session     optional      pam_mount.so
```

Screen Sharing and Screen Recording

macOS will disable support for TouchID, Watch, or Smartcard authentication when being watched or recorded. This can cause certain portions of the system to not recognize your smartcard.

In Unified Logging you'll notice an entry such as

```
2022-07-14 16:45:46.880038-0400 0x2F97 Info 0xC8D2 1600 SecurityAgent: (SecurityAgent)
[com.apple.Authorization:SecurityAgent] Screen is being watched, no Touch ID, Watch or
SmartCard support is allowed
```

This can be remediated by writing the preference domain com.apple.authorization with the key ignoreARD.

`defaults write com.apple.Authorization ignoreARD -bool true`

Or applied system wide with a configuration profile named `com.apple.security.authorization.mobileconfig` in the project's `includes` folder.

```
<key>PayloadType</key>
<string>com.apple.security.authorization</string>
<key>ignoreArd</key>
<true/>
```