macOS Security Compliance
GlobalTill

macOS 12

Security Configuration - 800GT-53R5_MODERATE_OS12

Tailored from 800-53R5_MODERATE

Monterey Guidance Revision 5.0 (2023-09-21)

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 12 system against the 800GT-53R5_MODERATE_OS12 (Tailored from 800-53R5_MODERATE) 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

Security configuration adapted by:

Michael MacKinnon	GlobalTill

Security configuration reviewed and approved by:

Douglas Wilson	GlobalTill

macOS Security Compliance Project

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
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*
NIST Special Publication 800-63	*NIST Special Publication 800-63*
NIST Special Publication 800-171	*NIST Special Publication 800-171 Rev 2*
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 8	*Apple macOS 12 (Monterey) STIG*

Table 5. 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 6. 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 7. Center for Internet Security

Document Number or Descriptor	Document Title
Apple macOS 12.0	*CIS Apple macOS 12.0 Benchmark version 2.1.0*

Chapter 6. Auditing

This section contains the configuration and enforcement of the OpenBSM settings.

The BSM Audit subsystem has been marked as deprecated by Apple.

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

6.1. Configure Audit Log Files to Not Contain Access Control Lists

The audit log files *MUST* not contain access control lists (ACLs).

This rule ensures that audit information and audit files are configured to be readable and writable only by system administrators, thereby preventing unauthorized access, modification, and deletion of files.

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

```
/bin/ls -le $(/usr/bin/grep '^dir' /etc/security/audit_control | /usr/bin/awk -F: '{print $2}') |
/usr/bin/awk '{print $1}' | /usr/bin/grep -c ":"
```

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

Remediation Description

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

```
/bin/chmod -RN $(/usr/bin/awk -F: '/^dir/{print $2}' /etc/security/audit_control)
```

ID	audit_acls_files_configure	
References	**800-53r5**	• AU-9
	800-171r2	• 3.3.8
	CCE	• CCE-90851-7

6.2. Configure Audit Log Folder to Not Contain Access Control Lists

The audit log folder *MUST* not contain access control lists (ACLs).

Audit logs contain sensitive data about the system and users. This rule ensures that the audit service is configured to create log folders that are readable and writable only by system administrators in order to prevent normal users from reading audit logs.

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

```
/bin/ls -lde $(/usr/bin/grep '^dir' /etc/security/audit_control | /usr/bin/awk -F: '{print $2}') | /usr/bin/awk '{print $1}' | /usr/bin/grep -c ":"
```

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

Remediation Description

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

```
/bin/chmod -N $(/usr/bin/grep '^dir' /etc/security/audit_control | /usr/bin/awk -F: '{print $2}')
```

ID	audit_acls_folders_configure	
References	**800-53r5**	• AU-9
	800-171r2	• 3.3.8
	CCE	• CCE-90852-5

6.3. Enable Security Auditing

The information system *MUST* be configured to generate audit records.

Audit records establish what types of events have occurred, when they occurred, and which users were involved. These records aid an organization in their efforts to establish, correlate, and investigate the events leading up to an outage or attack.

The content required to be captured in an audit record varies based on the impact level of an organization's system. Content that may be necessary to satisfy this requirement includes, for example, time stamps, source addresses, destination addresses, user identifiers, event descriptions, success/fail indications, filenames involved, and access or flow control rules invoked.

The information system initiates session audits at system start-up.

Security auditing is enabled by default on macOS.

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

```
/bin/launchctl list | /usr/bin/grep -c com.apple.auditd
```

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 load -w /System/Library/LaunchDaemons/com.apple.auditd.plist
```

ID	audit_auditd_enabled	
References	**800-53r5**	• AU-12, AU-12(1), AU-12(3) • AU-14(1) • AU-3, AU-3(1) • AU-8 • CM-5(1) • MA-4(1)
	800-171r2	• 3.3.1 • 3.3.2 • 3.3.7
	CCE	• CCE-90854-1

6.4. Configure System to Shut Down Upon Audit Failure

The audit service *MUST* be configured to shut down the computer if it is unable to audit system events.

Once audit failure occurs, user and system activity are no longer recorded, and malicious activity could go undetected. Audit processing failures can occur due to software/hardware errors, failures in the audit capturing mechanisms, and audit storage capacity being reached or exceeded.

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

```
/usr/bin/awk -F':' '/^policy/ {print $NF}' /etc/security/audit_control | /usr/bin/tr ',' '\n' |
/usr/bin/grep -Ec 'ahlt'
```

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/sed -i.bak 's/^policy.*/policy: ahlt,argv/' /etc/security/audit_control; /usr/sbin/audit
-s
```

ID	audit_failure_halt	
References	**800-53r5**	• AU-5
	800-171r2	• 3.3.4
	CCE	• CCE-90857-4

6.5. Configure Audit Log Files Group to Wheel

Audit log files *MUST* have the group set to wheel.

The audit service *MUST* be configured to create log files with the correct group ownership to prevent normal users from reading audit logs.

Audit logs contain sensitive data about the system and users. If log files are set to be readable and writable only by system administrators, the risk is mitigated.

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

```
/bin/ls -n $(/usr/bin/grep '^dir' /etc/security/audit_control | /usr/bin/awk -F: '{print $2}') |
/usr/bin/awk '{s+=$4} END {print s}'
```

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/chgrp -R wheel $(/usr/bin/grep '^dir' /etc/security/audit_control | /usr/bin/awk -F:
```

```
'{print $2}')/*
```

ID	audit_files_group_configure	
References	**800-53r5**	• AU-9
	800-171r2	• 3.3.8
	CCE	• CCE-90858-2

6.6. Configure Audit Log Files to Mode 440 or Less Permissive

The audit service *MUST* be configured to create log files that are readable only by the root user and group wheel. To achieve this, audit log files *MUST* be configured to mode 440 or less permissive; thereby preventing normal users from reading, modifying or deleting audit logs.

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

```
/bin/ls -l $(/usr/bin/grep '^dir' /etc/security/audit_control | /usr/bin/awk -F: '{print $2}') |
/usr/bin/awk '!/-r--r-----|current|total/{print $1}' | /usr/bin/wc -l | /usr/bin/tr -d ' '
```

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

Remediation Description

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

```
/bin/chmod 440 $(/usr/bin/grep '^dir' /etc/security/audit_control | /usr/bin/awk -F: '{print
$2}')/*
```

ID	audit_files_mode_configure	
References	**800-53r5**	• AU-9
	800-171r2	• 3.3.8
	CCE	• CCE-90859-0

6.7. Configure Audit Log Files to be Owned by Root

Audit log files *MUST* be owned by root.

The audit service *MUST* be configured to create log files with the correct ownership to prevent normal users from reading audit logs.

Audit logs contain sensitive data about the system and users. If log files are set to only be readable and writable by system administrators, the risk is mitigated.

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

```
/bin/ls -n $(/usr/bin/grep '^dir' /etc/security/audit_control | /usr/bin/awk -F: '{print $2}') |
/usr/bin/awk '{s+=$3} END {print s}'
```

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/chown -R root $(/usr/bin/grep '^dir' /etc/security/audit_control | /usr/bin/awk -F:
'{print $2}')/*
```

ID	audit_files_owner_configure	
References	**800-53r5**	• AU-9
	800-171r2	• 3.3.8
	CCE	• CCE-90860-8

6.8. Configure System to Audit All Authorization and Authentication Events

The auditing system *MUST* be configured to flag authorization and authentication (aa) events.

Authentication events contain information about the identity of a user, server, or client. Authorization events contain information about permissions, rights, and rules. If audit records do not include aa events, it is difficult to identify incidents and to correlate incidents to subsequent events.

Audit records can be generated from various components within the information system (e.g., via a module or policy filter).

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

```
/usr/bin/awk -F':' '/^flags/ { print $NF }' /etc/security/audit_control | /usr/bin/tr ',' '\n' |
```

```
/usr/bin/grep -Ec 'aa'
```

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/grep -qE "^flags.*[^-]aa" /etc/security/audit_control || /usr/bin/sed -i.bak '/^flags/ s/$/,aa/' /etc/security/audit_control; /usr/sbin/audit -s
```

ID	audit_flags_aa_configure	
References	**800-53r5**	• AC-2(12) • AU-12 • AU-2 • CM-5(1) • MA-4(1)
	800-171r2	• 3.3.1 • 3.3.2
	CCE	• CCE-90861-6

6.9. Configure System to Audit All Administrative Action Events

The auditing system *MUST* be configured to flag administrative action (ad) events.

Administrative action events include changes made to the system (e.g. modifying authentication policies). If audit records do not include ad events, it is difficult to identify incidents and to correlate incidents to subsequent events.

Audit records can be generated from various components within the information system (e.g., via a module or policy filter).

The information system audits the execution of privileged functions.

We recommend changing the line "43127:AUE_MAC_SYSCALL:mac_syscall(2):ad" to "43127:AUE_MAC_SYSCALL:mac_syscall(2):zz" in the file /etc/security/audit_event. This will prevent sandbox violations from being audited by the ad flag.

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

```
/usr/bin/awk -F':' '/^flags/ { print $NF }' /etc/security/audit_control | /usr/bin/tr ',' '\n' | /usr/bin/grep -Ec 'ad'
```

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/grep -qE "^flags.*[^-]ad" /etc/security/audit_control || /usr/bin/sed -i.bak '/^flags/ s/$/,ad/' /etc/security/audit_control; /usr/sbin/audit -s
```

ID	audit_flags_ad_configure	
References	**800-53r5**	• AC-2(12), AC-2(4) • AC-6(9) • AU-12 • AU-2 • CM-5(1) • MA-4(1)
	800-171r2	• 3.1.7 • 3.3.1 • 3.3.2
	CCE	• CCE-90862-4

6.10. Configure System to Audit All Failed Program Execution on the System

The audit system *MUST* be configured to record enforcement actions of access restrictions, including failed program execute (-ex) attempts.

Enforcement actions are the methods or mechanisms used to prevent unauthorized access and/or changes to configuration settings. One common and effective enforcement action method is using program execution restrictions (e.g., denying users access to execute certain processes).

This configuration ensures that audit lists include events in which program execution has failed. Without auditing the enforcement of program execution, it is difficult to identify attempted attacks, as there is no audit trail available for forensic investigation.

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

```
/usr/bin/awk -F':' '/^flags/ { print $NF }' /etc/security/audit_control | /usr/bin/tr ',' '\n' |
/usr/bin/grep -Ec '\-ex'
```

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/grep -qE "^flags.*-ex" /etc/security/audit_control || /usr/bin/sed -i.bak '/^flags/
s/$/,-ex/' /etc/security/audit_control; /usr/sbin/audit -s
```

ID	audit_flags_ex_configure	
References	**800-53r5**	• AC-2(12) • AU-12 • AU-2 • CM-5(1)
	800-171r2	• 3.3.1 • 3.3.2
	CCE	• CCE-90863-2

6.11. Configure System to Audit All Deletions of Object Attributes

The audit system *MUST* be configured to record enforcement actions of attempts to delete file attributes (fd).

***Enforcement actions are the methods or mechanisms used to prevent unauthorized changes to configuration settings. One common and effective enforcement action method is using access restrictions (i.e., denying modifications to a file by applying file permissions).

This configuration ensures that audit lists include events in which enforcement actions prevent attempts to delete a file.

Without auditing the enforcement of access restrictions, it is difficult to identify attempted attacks, as there is no audit trail available for forensic investigation.

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

```
/usr/bin/awk -F':' '/^flags/ { print $NF }' /etc/security/audit_control | /usr/bin/tr ',' '\n' |
/usr/bin/grep -Ec '\-fd'
```

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/grep -qE "^flags.*-fd" /etc/security/audit_control || /usr/bin/sed -i.bak '/^flags/
s/$/,-fd/' /etc/security/audit_control;/usr/sbin/audit -s
```

ID	audit_flags_fd_configure	
References	**800-53r5**	• AC-2(12) • AU-12 • AU-2 • AU-9 • CM-5(1) • MA-4(1)
	800-171r2	• N/A
	CCE	• CCE-90864-0

6.12. Configure System to Audit All Failed Change of Object Attributes

The audit system *MUST* be configured to record enforcement actions of failed attempts to modify file attributes (-fm).

Enforcement actions are the methods or mechanisms used to prevent unauthorized changes to configuration settings. One common and effective enforcement action method is using access restrictions (i.e., denying modifications to a file by applying file permissions).

This configuration ensures that audit lists include events in which enforcement actions prevent attempts to modify a file.

Without auditing the enforcement of access restrictions, it is difficult to identify attempted attacks, as there is no audit trail available for forensic investigation.

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

```
/usr/bin/awk -F':' '/^flags/ { print $NF }' /etc/security/audit_control | /usr/bin/tr ',' '\n' |
/usr/bin/grep -Ec '\-fm'
```

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/grep -qE "^flags.*-fm" /etc/security/audit_control || /usr/bin/sed -i.bak '/^flags/
s/$/,-fm/' /etc/security/audit_control;/usr/sbin/audit -s
```

ID	audit_flags_fm_failed_configure	
References	**800-53r5**	• AC-2(12) • AU-12 • AU-2 • AU-9 • CM-5(1) • MA-4(1)
	800-171r2	• 3.3.1 • 3.3.2 • 3.3.8
	CCE	• CCE-90865-7

6.13. Configure System to Audit All Failed Read Actions on the System

The audit system *MUST* be configured to record enforcement actions of access restrictions, including failed file read (-fr) attempts.

Enforcement actions are the methods or mechanisms used to prevent unauthorized access and/or changes to configuration settings. One common and effective enforcement action method is using access restrictions (e.g., denying access to a file by applying file permissions).

This configuration ensures that audit lists include events in which enforcement actions prevent attempts to read a file.

Without auditing the enforcement of access restrictions, it is difficult to identify attempted attacks, as there is no audit trail available for forensic investigation.

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

```
/usr/bin/awk -F':' '/^flags/ { print $NF }' /etc/security/audit_control | /usr/bin/tr ',' '\n' |
/usr/bin/grep -Ec '\-fr'
```

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/grep -qE "^flags.*-fr" /etc/security/audit_control || /usr/bin/sed -i.bak '/^flags/
s/$/,-fr/' /etc/security/audit_control;/usr/sbin/audit -s
```

ID	audit_flags_fr_configure	
References	**800-53r5**	• AC-2(12) • AU-12 • AU-2 • AU-9 • CM-5(1) • MA-4(1)
	800-171r2	• 3.3.1 • 3.3.2 • 3.3.8
	CCE	• CCE-90866-5

6.14. Configure System to Audit All Failed Write Actions on the System

The audit system *MUST* be configured to record enforcement actions of access restrictions, including failed file write (-fw) attempts.

Enforcement actions are the methods or mechanisms used to prevent unauthorized access and/or changes to configuration settings. One common and effective enforcement action method is using access restrictions (e.g., denying users access to edit a file by applying file permissions).

This configuration ensures that audit lists include events in which enforcement actions prevent attempts to change a file.

Without auditing the enforcement of access restrictions, it is difficult to identify attempted attacks, as there is no audit trail available for forensic investigation.

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

```
/usr/bin/awk -F':' '/^flags/ { print $NF }' /etc/security/audit_control | /usr/bin/tr ',' '\n' | /usr/bin/grep -Ec '\-fw'
```

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/grep -qE "^flags.*-fw" /etc/security/audit_control || /usr/bin/sed -i.bak '/^flags/ s/$/,-fw/' /etc/security/audit_control;/usr/sbin/audit -s
```

ID	audit_flags_fw_configure	
References	**800-53r5**	• AC-2(12) • AU-12 • AU-2 • AU-9 • CM-5(1) • MA-4(1)
	800-171r2	• 3.3.1 • 3.3.2 • 3.3.8
	CCE	• CCE-90867-3

6.15. Configure System to Audit All Log In and Log Out Events

The audit system *MUST* be configured to record all attempts to log in and out of the system (lo).

Frequently, an attacker that successfully gains access to a system has only gained access to an account with limited privileges, such as a guest account or a service account. The attacker must attempt to change to another user account with normal or elevated privileges in order to proceed. Auditing both successful and unsuccessful attempts to switch to another user account (by way of monitoring login and logout events) mitigates this risk.

The information system monitors login and logout events.

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

```
/usr/bin/awk -F':' '/^flags/ { print $NF }' /etc/security/audit_control | /usr/bin/tr ',' '\n' |
/usr/bin/grep -Ec '^lo'
```

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/grep -qE "^flags.*[^-]lo" /etc/security/audit_control || /usr/bin/sed -i.bak '/^flags/
s/$/,lo/' /etc/security/audit_control; /usr/sbin/audit -s
```

ID	audit_flags_lo_configure	
References	**800-53r5**	• AC-17(1) • AC-2(12) • AU-12 • AU-2 • MA-4(1)
	800-171r2	• 3.1.12 • 3.3.1 • 3.3.2
	CCE	• CCE-90868-1

6.16. Configure Audit Log Folders Group to Wheel

Audit log files *MUST* have the group set to wheel.

The audit service *MUST* be configured to create log files with the correct group ownership to prevent normal users from reading audit logs.

Audit logs contain sensitive data about the system and users. If log files are set to be readable and writable only by system administrators, the risk is mitigated.

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

```
/bin/ls -dn $(/usr/bin/grep '^dir' /etc/security/audit_control | /usr/bin/awk -F: '{print $2}') |
/usr/bin/awk '{print $4}'
```

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/chgrp wheel $(/usr/bin/awk -F : '/^dir/{print $2}' /etc/security/audit_control)
```

ID	audit_folder_group_configure	
References	**800-53r5**	• AU-9
	800-171r2	• 3.3.8
	CCE	• CCE-90869-9

6.17. Configure Audit Log Folders to be Owned by Root

Audit log files *MUST* be owned by root.

The audit service *MUST* be configured to create log files with the correct ownership to prevent normal users from reading audit logs.

Audit logs contain sensitive data about the system and users. If log files are set to only be readable and writable by system administrators, the risk is mitigated.

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

```
/bin/ls -dn $(/usr/bin/grep '^dir' /etc/security/audit_control | /usr/bin/awk -F: '{print $2}') |
/usr/bin/awk '{print $3}'
```

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/chown root $(/usr/bin/awk -F : '/^dir/{print $2}' /etc/security/audit_control)
```

ID	audit_folder_owner_configure	
References	**800-53r5**	• AU-9
	800-171r2	• 3.3.8
	CCE	• CCE-90870-7

6.18. Configure Audit Log Folders to Mode 700 or Less Permissive

The audit log folder *MUST* be configured to mode 700 or less permissive so that only the root user is able to read, write, and execute changes to folders.

Because audit logs contain sensitive data about the system and users, the audit service *MUST* be configured to mode 700 or less permissive; thereby preventing normal users from reading, modifying or deleting audit logs.

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

```
/usr/bin/stat -f %A $(/usr/bin/grep '^dir' /etc/security/audit_control | /usr/bin/awk -F: '{print $2}')
```

If the result is not **700**, this is a finding.

Remediation Description

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

```
/bin/chmod 700 $(/usr/bin/grep '^dir' /etc/security/audit_control | /usr/bin/awk -F: '{print $2}')
```

ID	audit_folders_mode_configure	
References	**800-53r5**	• AU-9
	800-171r2	• 3.3.8
	CCE	• CCE-90871-5

6.19. Configure Audit Retention to 7d

The audit service *MUST* be configured to require records be kept for a organizational defined value before deletion, unless the system uses a central audit record storage facility.

When "expire-after" is set to "7d", the audit service will not delete audit logs until the log data criteria is met.

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

```
/usr/bin/awk -F: '/expire-after/{print $2}' /etc/security/audit_control
```

If the result is not **7d**, this is a finding.

Remediation Description

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

```
/usr/bin/sed -i.bak 's/^expire-after.*/expire-after:7d/' /etc/security/audit_control;
/usr/sbin/audit -s
```

ID	audit_retention_configure	
References	**800-53r5**	• AU-11 • AU-4
	800-171r2	• N/A
	CCE	• CCE-90875-6

6.20. Configure Audit Failure Notification

The audit service *MUST* be configured to immediately print messages to the console or email administrator users when an auditing failure occurs.

It is critical for the appropriate personnel to be made aware immediately if a system is at risk of failing to process audit logs as required. Without a real-time alert, security personnel may be unaware of a potentially harmful failure in the auditing system's capability, and system operation may be adversely affected.

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

```
/usr/bin/grep -c "logger -s -p" /etc/security/audit_warn
```

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/sed -i.bak 's/logger -p/logger -s -p/' /etc/security/audit_warn; /usr/sbin/audit -s
```

ID	audit_settings_failure_notify	
References	**800-53r5**	• AU-5, AU-5(2)
	800-171r2	• 3.3.4
	CCE	• CCE-90876-4

Chapter 7. 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.

7.1. Enforce Multifactor Authentication for Login

The system *MUST* be configured to enforce multifactor authentication.

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

Modification of Pluggable Authentication Modules (PAM) now require user authorization, or use of a Privacy Preferences Policy Control (PPPC) profile from MDM that authorizes modifying system administrator files or full disk access.

/etc/pam.d/login 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/bin/grep -Ec '^(auth\s+sufficient\s+pam_smartcard.so|auth\s+required\s+pam_deny.so)'
/etc/pam.d/login
```

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

Remediation Description

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

```
/bin/cat > /etc/pam.d/login << LOGIN_END
# 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
LOGIN_END


/bin/chmod 644 /etc/pam.d/login
/usr/sbin/chown root:wheel /etc/pam.d/login
```

ID	auth_pam_login_smartcard_enforce	
References	**800-53r5**	• IA-2(1), IA-2(2), IA-2(8)
	800-171r2	• 3.5.3
	CCE	• CCE-90877-2

7.2. Enforce Multifactor Authentication for the su Command

The system *MUST* be configured such that, when the su command is used, multifactor authentication is enforced.

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

Modification of Pluggable Authentication Modules (PAM) now require user authorization, or use of a Privacy Preferences Policy Control (PPPC) profile from MDM that authorizes modifying system administrator files or full disk access.

/etc/pam.d/su 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/bin/grep -Ec '^(auth\s+sufficient\s+pam_smartcard.so|auth\s+required\s+pam_rootok.so)'
/etc/pam.d/su
```

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

Remediation Description

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

```
/bin/cat > /etc/pam.d/su << SU_END
# 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
SU_END

# Fix new file ownership and permissions
/bin/chmod 644 /etc/pam.d/su
/usr/sbin/chown root:wheel /etc/pam.d/su
```

ID	auth_pam_su_smartcard_enforce	
References	**800-53r5**	• IA-2(1), IA-2(2), IA-2(8)
	800-171r2	• 3.5.3
	CCE	• CCE-90878-0

7.3. Enforce Multifactor Authentication for Privilege Escalation Through the sudo Command

The system *MUST* be configured to enforce multifactor authentication when the sudo command is used to elevate privilege.

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

Modification of Pluggable Authentication Modules (PAM) now require user authorization, or use of a Privacy Preferences Policy Control (PPPC) profile from MDM that authorizes modifying system administrator files or full disk access.

/etc/pam.d/sudo 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/bin/grep -Ec '^(auth\s+sufficient\s+pam_smartcard.so|auth\s+required\s+pam_deny.so)'
/etc/pam.d/sudo
```

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

Remediation Description

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

```
/bin/cat > /etc/pam.d/sudo << SUDO_END
# 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
SUDO_END

/bin/chmod 444 /etc/pam.d/sudo
/usr/sbin/chown root:wheel /etc/pam.d/sudo
```

ID	auth_pam_sudo_smartcard_enforce	
References	**800-53r5**	• IA-2(1), IA-2(2), IA-2(8)
	800-171r2	• 3.5.3
	CCE	• CCE-90879-8

7.4. 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)
	800-171r2	• N/A
	CCE	• CCE-90880-6

7.5. Set Smartcard Certificate Trust to Moderate

The macOS system *MUST* be configured to block access to users who are no longer authorized (i.e., users with revoked certificates).

To prevent the use of untrusted certificates, the certificates on a smartcard card *MUST* meet the following criteria: its issuer has a system-trusted certificate, the certificate is not expired, its "valid-after" date is in the past, and it passes Certificate Revocation List (CRL) and Online Certificate Status Protocol (OCSP) checking.

By setting the smartcard certificate trust level to moderate, the system will execute a soft revocation, i.e., if the OCSP/CRL server is unreachable, authentication will still succeed.

Before applying this setting, please see the smartcard supplemental guidance.

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('checkCertificateTrust').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.security.smartcard) payload type:

```
<key>checkCertificateTrust</key>
<integer>2</integer>
```

ID	auth_smartcard_certificate_trust_enforce_moderate	
References	**800-53r5**	• IA-5(2) • SC-17
	800-171r2	• N/A
	CCE	• CCE-90882-2

7.6. 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/>
```

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)
	800-171r2	• 3.5.1 • 3.5.2 • 3.5.3
	CCE	• CCE-90883-0

7.7. 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/bin/grep -Ec '^(PasswordAuthentication\s+no|ChallengeResponseAuthentication\s+no)'
/etc/ssh/sshd_config
```

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

Remediation Description

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

```
/usr/bin/sed -i.bak_$(date "+%Y-%m-%d_%H:%M") "s|#PasswordAuthentication
yes|PasswordAuthentication no|; s|#ChallengeResponseAuthentication
yes|ChallengeResponseAuthentication no|" /etc/ssh/sshd_config; /bin/launchctl kickstart
-k system/com.openssh.sshd
```

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
	800-171r2	• 3.5.1 • 3.5.2 • 3.5.3 • 3.7.5
	CCE	• CCE-90884-8

Chapter 8. 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.

8.1. 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)
	800-171r2	• 3.1.20 • 3.4.6
	CCE	• CCE-90885-5

8.2. Disable the System Preference Pane for Apple ID

The system preference pane for Apple ID *MUST* be disabled.

Disabling the system preference pane 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()="DisabledPreferencePanes"]/following-sibling::*[1]' - | /usr/bin/grep -c com.apple.preferences.AppleIDPrefPane
```

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>DisabledPreferencePanes</key>
<array>
  <string>com.apple.preferences.AppleIDPrefPane</string>
</array>
```

ID	icloud_appleid_prefpane_disable

References	**800-53r5**	• AC-20, AC-20(1) • CM-7, CM-7(1)
	800-171r2	• 3.1.20 • 3.4.6
	CCE	• CCE-90886-3

8.3. 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)
	800-171r2	• 3.1.20 • 3.4.6
	CCE	• CCE-90887-1

8.4. 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)
	800-171r2	• 3.1.20 • 3.4.6
	CCE	• CCE-90888-9

8.5. 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)
	800-171r2	• 3.1.20 • 3.4.6
	CCE	• CCE-90889-7

8.6. 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)
	800-171r2	• 3.1.20 • 3.4.6
	CCE	• CCE-90890-5

8.7. 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)
	800-171r2	• 3.1.20 • 3.4.6
	CCE	• CCE-90891-3

8.8. 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)
	800-171r2	• 3.1.20 • 3.4.6
	CCE	• CCE-90892-1

8.9. 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)
	800-171r2	• 3.1.20 • 3.4.6
	CCE	• CCE-90893-9

8.10. 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)
	800-171r2	• 3.1.20 • 3.4.6
	CCE	• CCE-90894-7

8.11. 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)
	800-171r2	• 3.1.20 • 3.4.6
	CCE	• CCE-90895-4

8.12. 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)
	800-171r2	• 3.1.20 • 3.4.6
	CCE	• CCE-90896-2

Chapter 9. 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.

9.1. 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)
	800-171r2	• 3.1.1 • 3.1.2 • 3.1.16 • 3.1.20 • 3.4.6
	CCE	• CCE-90898-8

9.2. 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 << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.SetupAssistant.managed')\
.objectForKey('SkipCloudSetup').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.SetupAssistant.managed) payload type:

```
<key>SkipCloudSetup</key>
<true/>
```

ID	os_appleid_prompt_disable

References	**800-53r5**	• AC-20
	800-171r2	• 3.1.20
	CCE	• CCE-90902-8

9.3. Configure Apple System Log Files Owned by Root and Group to Wheel

The Apple System Logs (ASL) *MUST* be owned by root.

ASL logs contain sensitive data about the system and users. If ASL log files are set to only be readable and writable by system administrators, the risk is mitigated.

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

```
/usr/bin/stat -f '%Su:%Sg:%N' $(/usr/bin/grep -e '^>' /etc/asl.conf /etc/asl/* | /usr/bin/awk '{ print $2 }') 2> /dev/null | /usr/bin/awk '!/^root:wheel:/{print $1}' | /usr/bin/wc -l | /usr/bin/tr -d '
'
```

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/chown root:wheel $(/usr/bin/stat -f '%Su:%Sg:%N' $(/usr/bin/grep -e '^>' /etc/asl.conf /etc/asl/* | /usr/bin/awk '{ print $2 }') 2> /dev/null | /usr/bin/awk '!/^root:wheel:/{print $1}' | /usr/bin/awk -F":" '!/^root:wheel:/{print $3}')
```

ID	os_asl_log_files_owner_group_configure	
References	**800-53r5**	• SI-11
	800-171r2	• N/A
	CCE	• CCE-90904-4

9.4. Configure Apple System Log Files To Mode 640 or Less Permissive

The Apple System Logs (ASL) *MUST* be configured to be writable by root and readable only by the root user and group wheel. To achieve this, ASL log files *MUST* be configured to mode 640

permissive or less; thereby preventing normal users from reading, modifying or deleting audit logs. System logs frequently contain sensitive information that could be used by an attacker. Setting the correct permissions mitigates this risk.

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

```
/usr/bin/stat -f '%A:%N' $(/usr/bin/grep -e '^>' /etc/asl.conf /etc/asl/* | /usr/bin/awk '{ print $2 }') 2> /dev/null | /usr/bin/awk '!/640/{print $1}' | /usr/bin/wc -l | /usr/bin/tr -d ' '
```

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

Remediation Description

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

```
/bin/chmod 640 $(/usr/bin/stat -f '%A:%N' $(/usr/bin/grep -e '^>' /etc/asl.conf /etc/asl/* | /usr/bin/awk '{ print $2 }') 2> /dev/null | /usr/bin/awk -F":" '!/640/{print $2}')
```

ID	os_asl_log_files_permissions_configure	
References	**800-53r5**	• SI-11
	800-171r2	• N/A
	CCE	• CCE-90905-1

9.5. 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.

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/bin/csrutil authenticated-root | /usr/bin/grep -c '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 authenticated-root enable
```

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)
	800-171r2	• 3.1.1 • 3.1.2 • 3.4.5
	CCE	• CCE-90907-7

9.6. Disable Bonjour Multicast

Bonjour multicast advertising *MUST* be disabled to prevent the system from broadcasting its presence and available services over network interfaces.

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

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.mDNSResponder')\
.objectForKey('NoMulticastAdvertisements').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.mDNSResponder) payload type:

```
<key>NoMulticastAdvertisements</key>
<true/>
```

ID	os_bonjour_disable	
References	**800-53r5**	• CM-7, CM-7(1)
	800-171r2	• 3.4.6
	CCE	• CCE-90908-5

9.7. Issue or Obtain Public Key Certificates from an Approved Service Provider

The organization *MUST* issue or obtain public key certificates from an organization-approved service provider and ensure only approved trust anchors are in the System Keychain.

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

```
/usr/bin/security dump-keychain /Library/Keychains/System.keychain | /usr/bin/awk -F'"' '/labl/ {print $4}'
```

If the result is not **a list containing approved root certificates**, this is a finding.

Remediation Description

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

Obtain the approved certificates from the appropriate authority and install them to the System Keychain.

ID	os_certificate_authority_trust	
References	**800-53r5**	• SC-17
	800-171r2	• N/A
	CCE	• CCE-90911-9

9.8. Enforce Installation of XProtect and Gatekeeper Updates Automatically

Software Update *MUST* be configured to update XProtect, MRT, and Gatekeepr automatically.

This setting enforces definition updates for XProtect 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

Software update will automatically update XProtect 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
	800-171r2	• N/A
	CCE	• CCE-90913-5

9.9. 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)
	800-171r2	• 3.1.1 • 3.1.2
	CCE	• CCE-90922-6

9.10. Control Connections to Other Systems via a Deny-All and Allow-by-Exception Firewall Policy

A deny-all and allow-by-exception firewall policy *MUST* be employed for managing connections to other systems.

Organizations *MUST* ensure the built-in packet filter firewall is configured correctly to employ the default deny rule.

Failure to restrict network connectivity to authorized systems permits inbound connections from malicious systems. It also permits outbound connections that may facilitate the exfiltration of data.

If you are using a third-party firewall solution, this setting does not apply.

Configuring the built-in packet filter firewall to employ the default deny rule has the potential to interfere with applications on the system in an unpredictable manner. Information System Security Officers (ISSOs) may make the risk-based decision not to configure the built-in packet filter firewall to employ the default deny rule to avoid losing 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):

```
/sbin/pfctl -a '*' -sr &> /dev/null | /usr/bin/grep -c "block drop in all"
```

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

Remediation Description

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

See the firewall supplemental which includes a script that has an example policy to implement this rule.

ID	os_firewall_default_deny_require	
References	**800-53r5**	• AC-4 • SC-7(5)
	800-171r2	• 3.1.3 • 3.13.6
	CCE	• CCE-90923-4

9.11. Enable Firewall Logging

Firewall logging *MUST* be enabled.

Firewall logging ensures that malicious network activity will be logged to the system.

The firewall data is logged to Apple's Unified Logging with the subsystem com.apple.alf and the data is marked as private. In order to enable private data, review the com.apple.alf.private_data.mobileconfig file in the project's includes folder.

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.security.firewall')\
 .objectForKey('EnableLogging').js
 let pref2 = $.NSUserDefaults.alloc.initWithSuiteName('com.apple.security.firewall')\
 .objectForKey('LoggingOption').js
 if ( pref1 == true && pref2 == "detail" ){
  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.security.firewall) payload type:

```
<key>EnableLogging</key>
<true/>
<key>LoggingOption</key>
<string>detail</string>
```

ID	os_firewall_log_enable	
References	**800-53r5**	• AU-12 • SC-7
	800-171r2	• 3.3.1 • 3.3.2 • 3.13.1 • 3.13.2 • 3.13.5
	CCE	• CCE-90924-2

9.12. 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
```

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.

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:

See discussion on remediation and how to enable firmware password.

ID	os_firmware_password_require	
References	**800-53r5**	• AC-6
	800-171r2	• 3.1.5
	CCE	• CCE-90925-9

9.13. 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/sbin/spctl --status | /usr/bin/grep -c "assessments enabled"
```

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.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)
	800-171r2	• 3.4.5
	CCE	• CCE-90926-7

9.14. Enforce Gatekeeper 30 Day Automatic Rearm

Gatekeeper *MUST* be configured to automatically rearm after 30 days if disabled.

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

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

The following settings are in the (com.apple.ManagedClient.preferences) payload. This payload requires the additional settings to be sub-payloads within, containing their defined payload types.

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

```
<key>GKAutoRearm</key>
<true/>
```

ID	os_gatekeeper_rearm	
References	**800-53r5**	• CM-5
	800-171r2	• 3.4.5
	CCE	• CCE-90927-5

9.15. 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)
	800-171r2	• 3.1.1 • 3.1.2 • 3.1.20 • 3.4.6
	CCE	• CCE-90929-1

9.16. 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 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 -1 |
/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 $( /usr/bin/find /System/Volumes/Data/Users -mindepth 1 -maxdepth 1
-type d -perm -1 | /usr/bin/grep -v "Shared" | /usr/bin/grep -v "Guest" ); do
  /bin/chmod og-rwx "$userDirs"
done
unset IFS
```

ID	os_home_folders_secure	
References	**800-53r5**	• AC-6
	800-171r2	• 3.1.5
	CCE	• CCE-90931-7

9.17. Disable the Built-in Web Server

The built-in web server is a non-essential service built into macOS and *MUST* be disabled.

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):

```
/bin/launchctl print-disabled system | /usr/bin/grep -c '"org.apache.httpd" => true'
```

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/org.apache.httpd
```

ID	os_httpd_disable	
References	**800-53r5**	• AC-17 • AC-3
	800-171r2	• 3.1.1 • 3.1.2
	CCE	• CCE-90932-5

9.18. 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 << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.SetupAssistant.managed')\
.objectForKey('SkipiCloudStorageSetup').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.SetupAssistant.managed) payload type:

```
<key>SkipiCloudStorageSetup</key>
```

```
<true/>
```

ID	os_icloud_storage_prompt_disable	
References	**800-53r5**	• AC-20
	800-171r2	• 3.1.20
	CCE	• CCE-90933-3

9.19. Disable Infrared (IR) support

Infrared (IR) support *MUST* be disabled to prevent users from controlling the system with IR devices.

By default, if IR is enabled, the system will accept IR control from any remote device.

This is applicable only to models of Mac Mini systems earlier than Mac Mini8,1.

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

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.driver.AppleIRController')\
.objectForKey('DeviceEnabled').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:

The following settings are in the (com.apple.ManagedClient.preferences) payload. This payload requires the additional settings to be sub-payloads within, containing their defined payload types.

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

```
<key>DeviceEnabled</key>
<false/>
```

ID	os_ir_support_disable	
References	**800-53r5**	• AC-18 • CM-7, CM-7(1)
	800-171r2	• 3.1.16 • 3.4.6
	CCE	• CCE-90939-0

9.20. Enforce Enrollment in Mobile Device Management

You *MUST* enroll your Mac in a Mobile Device Management (MDM) software.

User Approved MDM (UAMDM) enrollment or enrollment via Apple Business Manager (ABM)/Apple School Manager (ASM) is required to manage certain security settings. Currently these include:

- Allowed Kernel Extensions
- Allowed Approved System Extensions
- Privacy Preferences Policy Control Payload
- ExtensibleSingleSignOn
- FDEFileVault

In macOS 11, UAMDM grants Supervised status on a Mac, unlocking the following MDM features, which were previously locked behind ABM:

- Activation Lock Bypass
- Access to Bootstrap Tokens
- Scheduling Software Updates
- Query list and delete local users

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

```
/usr/bin/profiles status -type enrollment | /usr/bin/awk -F: '/MDM enrollment/ {print $2}' | /usr/bin/grep -c "Yes (User Approved)"
```

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

Remediation Description

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

Ensure that system is enrolled via UAMDM.

ID	os_mdm_require	
References	**800-53r5**	• CM-2 • CM-6
	800-171r2	• 3.4.1 • 3.4.2
	CCE	• CCE-90950-7

9.21. Configure System Log Files Owned by Root and Group to Wheel

The system log files *MUST* be owned by root.

System logs contain sensitive data about the system and users. If log files are set to only be readable and writable by system administrators, the risk is mitigated.

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

```
/usr/bin/stat -f '%Su:%Sg:%N' $(/usr/bin/grep -v '^#' /etc/newsyslog.conf | /usr/bin/awk '{ print $1 }') 2> /dev/null | /usr/bin/awk '!/^root:wheel:/{print $1}' | /usr/bin/wc -l | /usr/bin/tr -d ' '
```

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/chown root:wheel $(/usr/bin/stat -f '%Su:%Sg:%N' $(/usr/bin/grep -v '^#' /etc/newsyslog.conf | /usr/bin/awk '{ print $1 }') 2> /dev/null | /usr/bin/awk -F":" '!/^root:wheel:/{print $3}')
```

ID	os_newsyslog_files_owner_group_configure	
References	**800-53r5**	• SI-11
	800-171r2	• N/A
	CCE	• CCE-90954-9

9.22. Configure System Log Files to Mode 640 or Less Permissive

The system logs *MUST* be configured to be writable by root and readable only by the root user and group wheel. To achieve this, system log files *MUST* be configured to mode 640 permissive or less; thereby preventing normal users from reading, modifying or deleting audit logs. System logs frequently contain sensitive information that could be used by an attacker. Setting the correct permissions mitigates this risk.

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

```
/usr/bin/stat -f '%A:%N' $(/usr/bin/grep -v '^#' /etc/newsyslog.conf | /usr/bin/awk '{ print $1 }')
2> /dev/null | /usr/bin/awk '!/640/{print $1}' | /usr/bin/wc -l | /usr/bin/tr -d ' '
```

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

Remediation Description

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

```
/bin/chmod 640 $(/usr/bin/stat -f '%A:%N' $(/usr/bin/grep -v '^#' /etc/newsyslog.conf |
/usr/bin/awk '{ print $1 }') 2> /dev/null | /usr/bin/awk '!/640/{print $1}' | awk -F":"
'!/640/{print $2}')
```

ID	os_newsyslog_files_permissions_configure	
References	**800-53r5**	• SI-11
	800-171r2	• N/A
	CCE	• CCE-90955-6

9.23. 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):

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

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.nfsd
```

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

ID	os_nfsd_disable	
References	**800-53r5**	• AC-17 • AC-3
	800-171r2	• 3.1.1 • 3.1.2
	CCE	• CCE-90956-4

9.24. Enable Parental Controls

Parental Controls *MUST* be enabled.

Control of program execution is a mechanism used to prevent program execution of unauthorized programs, which is critical to maintaining a secure system baseline.

Parental Controls on the macOS consist of many different payloads, which are set individually depending on the type of control required. Enabling parental controls allows for further configuration of these restrictions.

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

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess.new')\
.objectForKey('familyControlsEnabled').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.new) payload type:

```
<key>familyControlsEnabled</key>
<true/>
```

ID	os_parental_controls_enable	
References	**800-53r5**	• CM-7(2)
	800-171r2	• 3.4.7
	CCE	• CCE-90966-3

9.25. Disable Password Autofill

Password Autofill *MUST* be disabled.

macOS allows users to save passwords and use the Password Autofill feature in Safari and compatible apps. To protect against malicious users gaining access to the system, this feature *MUST* be disabled to prevent users from being prompted to save passwords in applications.

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('allowPasswordAutoFill').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>allowPasswordAutoFill</key>
<false/>
```

ID	os_password_autofill_disable

References	**800-53r5**	• CM-7, CM-7(1) • IA-11 • IA-5, IA-5(13)
	800-171r2	• 3.4.6 • 3.5.1 • 3.5.2
	CCE	• CCE-90967-1

9.26. Disable Proximity Based Password Sharing Requests

Proximity based password sharing requests *MUST* be disabled.

The default behavior of macOS is to allow users to request passwords from other known devices (macOS and iOS). This feature *MUST* be disabled to prevent passwords from being shared.

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('allowPasswordProximityRequests').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>allowPasswordProximityRequests</key>
<false/>
```

ID	os_password_proximity_disable

References	800-53r5	• IA-5
	800-171r2	• 3.5.1 • 3.5.2
	CCE	• CCE-90968-9

9.27. Disable Password Sharing

Password Sharing *MUST* be disabled.

The default behavior of macOS is to allow users to share a password over Airdrop between other macOS and iOS devices. This feature *MUST* be disabled to prevent passwords from being shared.

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('allowPasswordSharing').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>allowPasswordSharing</key>
<false/>
```

ID	os_password_sharing_disable	
References	800-53r5	• IA-5
	800-171r2	• 3.5.1 • 3.5.2
	CCE	• CCE-90969-7

9.28. Display Policy Banner at Login Window

Displaying a standardized and approved use notification before granting access to the operating system ensures that users are provided with privacy and security notification verbiage that is consistent with applicable federal laws, Executive Orders, directives, policies, regulations, standards, and guidance.

System use notifications are required only for access via login interfaces with human users and are not required when such human interfaces do not exist.

The policy banner will show if a "PolicyBanner.rtf" or "PolicyBanner.rtfd" exists in the "/Library/Security" folder. NOTE: The banner text of the document *MUST* read:

> "You are accessing a U.S. Government information system, which includes: 1) this computer, 2) this computer network, 3) all Government-furnished computers connected to this network, and 4) all Government-furnished devices and storage media attached to this network or to a computer on this network. You understand and consent to the following: you may access this information system for authorized use only; unauthorized use of the system is prohibited and subject to criminal and civil penalties; you have no reasonable expectation of privacy regarding any communication or data transiting or stored on this information system at any time and for any lawful Government purpose, the Government may monitor, intercept, audit, and search and seize any communication or data transiting or stored on this information system; and any communications or data transiting or stored on this information system may be disclosed or used for any lawful Government purpose. This information system may contain Controlled Unclassified Information (CUI) that is subject to safeguarding or dissemination controls in accordance with law, regulation, or Government-wide policy. Accessing and using this system indicates your understanding of this warning."

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

```
/bin/ls -ld /Library/Security/PolicyBanner.rtf* | /usr/bin/wc -l | /usr/bin/tr -d ' '
```

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

Remediation Description

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

```
bannerText="You are accessing a U.S. Government information system, which includes: 1) this computer, 2) this computer network, 3) all Government-furnished computers connected to this network, and 4) all Government-furnished devices and storage media
```

```
attached to this network or to a computer on this network. You understand and consent to
the following: you may access this information system for authorized use only;
unauthorized use of the system is prohibited and subject to criminal and civil penalties;
you have no reasonable expectation of privacy regarding any communication or data
transiting or stored on this information system at any time and for any lawful Government
purpose, the Government may monitor, intercept, audit, and search and seize any
communication or data transiting or stored on this information system; and any
communications or data transiting or stored on this information system may be disclosed
or used for any lawful Government purpose. This information system may contain
Controlled Unclassified Information (CUI) that is subject to safeguarding or dissemination
controls in accordance with law, regulation, or Government-wide policy. Accessing and
using this system indicates your understanding of this warning."
/bin/mkdir /Library/Security/PolicyBanner.rtfd
/usr/bin/textutil -convert rtf -output /Library/Security/PolicyBanner.rtfd/TXT.rtf -stdin
<<EOF
$bannerText
EOF
```

ID	os_policy_banner_loginwindow_enforce	
References	**800-53r5**	• AC-8
	800-171r2	• 3.1.9
	CCE	• CCE-90973-9

9.29. Display Policy Banner at Remote Login

Remote login service *MUST* be configured to display a policy banner at login.

Displaying a standardized and approved use notification before granting access to the operating system ensures that users are provided with privacy and security notification verbiage that is consistent with applicable federal laws, Executive Orders, directives, policies, regulations, standards, and guidance.

System use notifications are required only for access via login interfaces with human users and are not required when such human interfaces do not exist.

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

```
bannerText="You are accessing a U.S. Government (USG) Information System (IS) that is
provided for USG-authorized use only. By using this IS (which includes any device attached to
this IS), you consent to the following conditions:
```

```
-The USG routinely intercepts and monitors communications on this IS for purposes including, but not limited to, penetration testing, COMSEC monitoring, network operations and defense, personnel misconduct (PM), law enforcement (LE), and counterintelligence (CI) investigations.
-At any time, the USG may inspect and seize data stored on this IS.
-Communications using, or data stored on, this IS are not private, are subject to routine monitoring, interception, and search, and may be disclosed or used for any USG authorized purpose.
-This IS includes security measures (e.g., authentication and access controls) to protect USG interests--not for your personal benefit or privacy.
-Notwithstanding the above, using this IS does not constitute consent to PM, LE or CI investigative searching or monitoring of the content of privileged communications, or work product, related to personal representation or services by attorneys, psychotherapists, or clergy, and their assistants. Such communications and work product are private and confidential. See User Agreement for details."
test "$(cat /etc/banner)" = "$bannerText" && echo "1" || echo "0"
```

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

Remediation Description

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

```
bannerText="You are accessing a U.S. Government (USG) Information System (IS) that is provided for USG-authorized use only. By using this IS (which includes any device attached to this IS), you consent to the following conditions:
-The USG routinely intercepts and monitors communications on this IS for purposes including, but not limited to, penetration testing, COMSEC monitoring, network operations and defense, personnel misconduct (PM), law enforcement (LE), and counterintelligence (CI) investigations.
-At any time, the USG may inspect and seize data stored on this IS.
-Communications using, or data stored on, this IS are not private, are subject to routine monitoring, interception, and search, and may be disclosed or used for any USG authorized purpose.
-This IS includes security measures (e.g., authentication and access controls) to protect USG interests--not for your personal benefit or privacy.
-Notwithstanding the above, using this IS does not constitute consent to PM, LE or CI investigative searching or monitoring of the content of privileged communications, or work product, related to personal representation or services by attorneys, psychotherapists, or clergy, and their assistants. Such communications and work product are private and confidential. See User Agreement for details."
```

```
/bin/echo "${bannerText}" > /etc/banner
```

ID	os_policy_banner_ssh_configure	
References	**800-53r5**	• AC-8
	800-171r2	• 3.1.9
	CCE	• CCE-90974-7

9.30. Enforce SSH to Display Policy Banner

SSH *MUST* be configured to display a policy banner.

Displaying a standardized and approved use notification before granting access to the operating system ensures that users are provided with privacy and security notification verbiage that is consistent with applicable federal laws, Executive Orders, directives, policies, regulations, standards, and guidance.

System use notifications are required only for access via login interfaces with human users and are not required when such human interfaces do not exist

/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 -T | /usr/bin/grep -c "^banner /etc/banner"
```

If the result is not **1**, 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

/usr/bin/grep -qxF 'banner /etc/banner' "${include_dir}01-mscp-sshd.conf" 2>/dev/null ||
echo "banner /etc/banner" >> "${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	os_policy_banner_ssh_enforce	
References	800-53r5	• AC-8
	800-171r2	• 3.1.9
	CCE	• CCE-90975-4

9.31. 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 | /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.

ID	os_recovery_lock_enable	
References	**800-53r5**	• AC-6
	800-171r2	• 3.1.5
	CCE	• CCE-90989-5

9.32. 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)
	800-171r2	• 3.5.1
		• 3.5.2
	CCE	• CCE-90994-5

9.33. Enforce Screen Saver at Login Window

A default screen saver *MUST* be configured to display at the login window and *MUST* not display any sensitive information.

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

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

If the result is not **/System/Library/Screen Savers/Flurry.saver**, 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.screensaver) payload type:

```
<key>loginWindowModulePath</key>
<string>/System/Library/Screen Savers/Flurry.saver</string>
```

ID	os_screensaver_loginwindow_enforce	
References	**800-53r5**	• AC-11(1)
	800-171r2	• 3.1.10
	CCE	• CCE-90995-2

9.34. Ensure Secure Boot Level Set to Full

The Secure Boot security setting *MUST* be set to full.

Full security is the default Secure Boot setting in macOS. During startup, when Secure Boot is set to full security, the Mac will verify the integrity of the operating system before allowing the operating system to boot.

This will only return a proper result on a T2 or Apple Silicon Macs.

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

```
/usr/libexec/mdmclient QuerySecurityInfo | /usr/bin/grep -c "SecureBootLevel = full"
```

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

Remediation Description

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

Boot into Recovery Mode and enable Full Secure Boot

ID	os_secure_boot_verify	
References	**800-53r5**	• SI-6 • SI-7, SI-7(1), SI-7(5)
	800-171r2	• N/A
	CCE	• CCE-90996-0

9.35. 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.

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
```

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
	800-171r2	• 3.1.1 • 3.1.2 • 3.3.6 • 3.3.8 • 3.4.5 • 3.13.4
	CCE	• CCE-91000-0

9.36. 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 << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.SetupAssistant.managed')\
.objectForKey('SkipSiriSetup').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.SetupAssistant.managed) payload type:

```
<key>SkipSiriSetup</key>
<true/>
```

ID	os_siri_prompt_disable	
References	**800-53r5**	• AC-20 • CM-7, CM-7(1)
	800-171r2	• 3.1.20 • 3.4.6
	CCE	• CCE-91001-8

9.37. 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 << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.SetupAssistant.managed')\
.objectForKey('SkipUnlockWithWatch').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.SetupAssistant.managed) payload type:

```
<key>SkipUnlockWithWatch</key>
<true/>
```

ID	os_skip_unlock_with_watch_enable	
References	**800-53r5**	• AC-20
	800-171r2	• 3.1.20
	CCE	• CCE-91002-6

9.38. Limit SSH to FIPS Compliant Connections

SSH *MUST* be configured to limit the Ciphers, HostbasedAcceptedAlgorithms, HostKeyAlgorithms, KexAlgorithms, MACs, PubkeyAcceptedAlgorithms, CASignatureAlgorithms to algorithms that are FIPS 140 validated.

FIPS 140-2 is the current standard for validating that mechanisms used to access cryptographic modules utilize authentication that meet federal requirements.

Operating systems utilizing encryption *MUST* use FIPS validated mechanisms for authenticating to cryptographic modules.

For more information on FIPS compliance with the version of SSH included in the macOS, the manual page apple_ssh_and_fips has additional information.

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

```
fips_ssh_config="Host *
Ciphers aes128-gcm@openssh.com
HostbasedAcceptedAlgorithms ecdsa-sha2-nistp256,ecdsa-sha2-nistp256-cert-v01@openssh.com
HostKeyAlgorithms ecdsa-sha2-nistp256,ecdsa-sha2-nistp256-cert-v01@openssh.com
KexAlgorithms ecdh-sha2-nistp256
MACs hmac-sha2-256
PubkeyAcceptedAlgorithms ecdsa-sha2-nistp256,ecdsa-sha2-nistp256-cert-v01@openssh.com
CASignatureAlgorithms ecdsa-sha2-nistp256"
/usr/bin/grep -c "$fips_ssh_config" /etc/ssh/ssh_config.d/fips_ssh_config
```

If the result is not **8**, this is a finding.

Remediation Description

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

```
fips_ssh_config="Host *
Ciphers aes128-gcm@openssh.com
```

```
HostbasedAcceptedAlgorithms ecdsa-sha2-nistp256,ecdsa-sha2-nistp256-cert-
v01@openssh.com
HostKeyAlgorithms ecdsa-sha2-nistp256,ecdsa-sha2-nistp256-cert-v01@openssh.com
KexAlgorithms ecdh-sha2-nistp256
MACs hmac-sha2-256
PubkeyAcceptedAlgorithms ecdsa-sha2-nistp256,ecdsa-sha2-nistp256-cert-
v01@openssh.com
CASignatureAlgorithms ecdsa-sha2-nistp256"
/bin/echo "${fips_ssh_config}" > /etc/ssh/ssh_config.d/fips_ssh_config
```

ID	os_ssh_fips_compliant	
References	**800-53r5**	• AC-17(2) • IA-7 • SC-13 • SC-8(1)
	800-171r2	• 3.1.13 • 3.13.8 • 3.13.11
	CCE	• CCE-91003-4

9.39. Set SSH Active Server Alive Maximum to 0

SSH *MUST* be configured with an Active Server Alive Maximum Count set to 0. Terminating an idle session within a short time period reduces the window of opportunity for unauthorized personnel to take control of a management session enabled on the console or console port that has been left unattended. In addition, quickly terminating an idle session or an incomplete login attempt will also free up resources committed by the managed network element.

/etc/ssh/ssh_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):

```
ret="pass"
for u in $(/usr/bin/dscl . -list /Users UniqueID | /usr/bin/awk '$2 > 500 {print $1}'); do
  sshCheck=$(/usr/bin/sudo -u $u /usr/bin/ssh -G . | /usr/bin/grep -c "^serveralivecountmax 0")
  if [[ "$sshCheck" == "0" ]]; then
    ret="fail"
    break
```

```
 fi
done
/bin/echo $ret
```

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

Remediation Description

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

```
for u in $(/usr/bin/dscl . -list /Users UniqueID | /usr/bin/awk '$2 > 500 {print $1}'); do
 config=$(/usr/bin/sudo -u $u /usr/bin/ssh -Gv . 2>&1 | /usr/bin/awk '/Reading
configuration data/ {print $NF}'| /usr/bin/tr -d '\r')
 configarray=( ${(f)config} )
 for c in $configarray; do
  /usr/bin/sudo -u $u /usr/bin/grep -q '^ServerAliveCountMax' "$c" && /usr/bin/sed -i ''
's/.*ServerAliveCountMax.*/ServerAliveCountMax 0/' "$c" || /bin/echo
'ServerAliveCountMax 0' >> "$c"
 done
done
```

ID	os_ssh_server_alive_count_max_configure	
References	**800-53r5**	• SC-10
	800-171r2	• 3.13.9
	CCE	• CCE-91005-9

9.40. Configure SSH ServerAliveInterval option set to 900

SSH *MUST* be configured with an Active Server Alive Maximum Count set to 900.

Setting the Active Server Alive Maximum Count to 900 will log users out after a 900 seconds interval of inactivity.

/etc/ssh/ssh_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):

```
ret="pass"
for u in $(/usr/bin/dscl . -list /Users UniqueID | /usr/bin/awk '$2 > 500 {print $1}'); do
 sshCheck=$(/usr/bin/sudo -u $u /usr/bin/ssh -G . | /usr/bin/grep -c "^serveraliveinterval 900")
 if [[ "$sshCheck" == "0" ]]; then
  ret="fail"
  break
 fi
done
/bin/echo $ret
```

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

Remediation Description

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

```
for u in $(/usr/bin/dscl . -list /Users UniqueID | /usr/bin/awk '$2 > 500 {print $1}'); do
 config=$(/usr/bin/sudo -u $u /usr/bin/ssh -Gv . 2>&1 | /usr/bin/awk '/Reading configuration data/ {print $NF}'| /usr/bin/tr -d '\r')
 configarray=( ${(f)config} )
 for c in $configarray; do
  /usr/bin/sudo -u $u /usr/bin/grep -q '^ServerAliveInterval' "$c" && /usr/bin/sed -i '' 's/.*ServerAliveInterval.*/ServerAliveInterval 900/' "$c" || /bin/echo 'ServerAliveInterval 900' >> "$c"
 done
done
```

ID	os_ssh_server_alive_interval_configure	
References	**800-53r5**	• AC-12 • SC-10
	800-171r2	• 3.13.9
	CCE	• CCE-91006-7

9.41. Configure SSHD ClientAliveCountMax to 0

If SSHD is enabled it *MUST* be configured with the Client Alive Maximum Count set to 0.

This will set the number of client alive messages which may be sent without the SSH server receiving any messages back from the client. If this threshold is reached while client alive messages are being sent, the SSH server will disconnect the client, terminating the session. The client alive messages are sent through the encrypted channel and therefore will not be spoofable. The client alive mechanism is valuable when the client or server depend on knowing when a connection has become unresponsive.

This setting is not intended to manage idle user sessions where there is no input from the client. Its purpose is to monitor for interruptions in network connectivity and force the session to terminate after the connection appears to be broken.

/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 -T | /usr/bin/awk '/clientalivecountmax/{print $2}'
```

If the result is not **0**, 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

/usr/bin/grep -qxF 'clientalivecountmax 0' "${include_dir}01-mscp-sshd.conf" 2>/dev/null
|| echo "clientalivecountmax 0" >> "${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	os_sshd_client_alive_count_max_configure	
References	**800-53r5**	• SC-10
	800-171r2	• 3.13.9
	CCE	• CCE-91007-5

9.42. Configure SSHD ClientAliveInterval to 900

If SSHD is enabled then it *MUST* be configured with the Client Alive Interval set to 900.

Sets a timeout interval in seconds after which if no data has been received from the client, sshd(8) will send a message through the encrypted channel to request a response from the client.

This setting works in conjuction with ClientAliveCountMax to determine the termination of the connection after the threshold has been reached.

This setting is not intended to manage idle user sessions where there is no input from the client. Its purpose is to monitor for interruptions in network connectivity and force the session to terminate after the connection appears to be broken.

/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 -T | /usr/bin/awk '/clientaliveinterval/{print $2}'
```

If the result is not **900**, 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
```

```
/usr/bin/grep -qxF 'clientaliveinterval 900' "${include_dir}01-mscp-sshd.conf" 2>/dev/null
|| echo "clientaliveinterval 900" >> "${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	os_sshd_client_alive_interval_configure	
References	**800-53r5**	• AC-12 • SC-10
	800-171r2	• 3.13.9
	CCE	• CCE-91008-3

9.43. Limit SSHD to FIPS Compliant Connections

If SSHD is enabled then it *MUST* be configured to limit the Ciphers, HostbasedAcceptedAlgorithms, HostKeyAlgorithms, KexAlgorithms, MACs, PubkeyAcceptedAlgorithms, CASignatureAlgorithms to algorithms that are FIPS 140 validated.

FIPS 140-2 is the current standard for validating that mechanisms used to access cryptographic modules utilize authentication that meet federal requirements.

Operating systems utilizing encryption *MUST* use FIPS validated mechanisms for authenticating to cryptographic modules.

For more information on FIPS compliance with the version of SSHD included in the macOS, the manual page apple_ssh_and_fips has additional information.

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

```
fips_sshd_config=("Ciphers aes128-gcm@openssh.com" "HostbasedAcceptedAlgorithms ecdsa-
sha2-nistp256,ecdsa-sha2-nistp256-cert-v01@openssh.com" "HostKeyAlgorithms ecdsa-sha2-
nistp256,ecdsa-sha2-nistp256-cert-v01@openssh.com" "KexAlgorithms ecdh-sha2-nistp256"
"MACs hmac-sha2-256" "PubkeyAcceptedAlgorithms ecdsa-sha2-nistp256,ecdsa-sha2-nistp256-
```

```
cert-v01@openssh.com" "CASignatureAlgorithms ecdsa-sha2-nistp256")
total=0
for config in $fips_sshd_config; do
  total=$(expr $(/usr/sbin/sshd -T | /usr/bin/grep -i -c "$config") + $total)
done
echo $total
```

If the result is not **7**, 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

fips_sshd_config=("Ciphers aes128-gcm@openssh.com" "HostbasedAcceptedAlgorithms
ecdsa-sha2-nistp256,ecdsa-sha2-nistp256-cert-v01@openssh.com" "HostKeyAlgorithms
ecdsa-sha2-nistp256,ecdsa-sha2-nistp256-cert-v01@openssh.com" "KexAlgorithms ecdh-
sha2-nistp256" "MACs hmac-sha2-256" "PubkeyAcceptedAlgorithms ecdsa-sha2-
nistp256,ecdsa-sha2-nistp256-cert-v01@openssh.com" "CASignatureAlgorithms ecdsa-
sha2-nistp256")

for config in $fips_sshd_config; do
  /usr/bin/grep -qxF "$config" "${include_dir}01-mscp-sshd.conf" 2>/dev/null || echo
"$config" >> "${include_dir}01-mscp-sshd.conf"
done

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	os_sshd_fips_compliant	
References	800-53r5	• AC-17(2) • IA-7 • SC-13 • SC-8(1)
	800-171r2	• 3.1.13 • 3.13.8 • 3.13.11
	CCE	• CCE-91010-9

9.44. Configure Sudo Timeout Period to 0

The file /etc/sudoers *MUST* include a timestamp_timout of 0.

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

```
/usr/bin/sudo /usr/bin/sudo -V | /usr/bin/grep -c "Authentication timestamp timeout: 0.0 minutes"
```

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/find /etc/sudoers* -type f -exec sed -i '' '/timestamp_timeout/d' '{}' \;
/bin/echo "Defaults timestamp_timeout=0" >> /etc/sudoers.d/mscp
```

ID	os_sudo_timeout_configure	
References	800-53r5	• N/A
	800-171r2	• N/A
	CCE	• CCE-91116-4

9.45. Configure Sudoers Timestamp Type

The file /etc/sudoers *MUST* be configured to not include a timestamp_type of global or ppid and be configured for timestamp record types of tty.

This rule ensures that the "sudo" command will prompt for the administrator's password at least once in each newly opened terminal window. This prevents a malicious user from taking advantage of an unlocked computer or an abandoned logon session by bypassing the normal password prompt requirement.

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

```
/usr/bin/sudo /usr/bin/sudo -V | /usr/bin/awk -F": " '/Type of authentication timestamp record/{print $2}'
```

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

Remediation Description

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

```
/usr/bin/find /etc/sudoers* -type f -exec sed -i '' '/timestamp_type/d; /!tty_tickets/d' '{}' \;
```

ID	os_sudoers_timestamp_type_configure	
References	**800-53r5**	• CM-5(1) • IA-11
	800-171r2	• N/A
	CCE	• CCE-91015-8

9.46. Ensure System Volume is Read Only

The System volume *MUST* be mounted as read-only in order to ensure that configurations critical to the integrity of the macOS have not been compromised. System Integrity Protection (SIP) will prevent the system volume from being mounted as writable.

The system volume is read only by default in macOS.

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

```
/usr/sbin/system_profiler SPStorageDataType | /usr/bin/awk '/Mount Point:
```

```
\$/{x=NR+2}(NR==x){print $2}'
```

If the result is not **No**, this is a finding.

Remediation Description

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

To remount the System volume as Read Only, rebooting the computer will mount it as Read Only.

ID	os_system_read_only	
References	**800-53r5**	• MA-4(1) • SC-34 • SI-7
	800-171r2	• N/A
	CCE	• CCE-91016-6

9.47. 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.

TFTP service is disabled at startup by default macOS.

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

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

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.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)
	800-171r2	• 3.1.1 • 3.1.2
	CCE	• CCE-91018-2

9.48. Enable Time Synchronization Daemon

The macOS time synchronization daemon (timed) *MUST* be enabled for proper time synchronization to an authorized time server.

The time synchronization daemon is enabled by default on macOS.

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

```
/bin/launchctl list | /usr/bin/grep -c com.apple.timed
```

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 load -w /System/Library/LaunchDaemons/com.apple.timed.plist
```

ID	os_time_server_enabled

References	**800-53r5**	• AU-12(1) • SC-45(1)
	800-171r2	• 3.3.7
	CCE	• CCE-91019-0

9.49. Disable TouchID Prompt during Setup Assistant

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

macOS prompts new users through enabling TouchID during Setup Assistant; this is not essential and, therefore, *MUST* be disabled to prevent against the risk of individuals electing to enable TouchID to override organization-wide settings.

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

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.SetupAssistant.managed')\
.objectForKey('SkipTouchIDSetup').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.SetupAssistant.managed) payload type:

```
<key>SkipTouchIDSetup</key>
<true/>
```

ID	os_touchid_prompt_disable	
References	**800-53r5**	• CM-6
	800-171r2	• 3.4.1 • 3.4.2
	CCE	• CCE-91020-8

9.50. 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 andlocked session prevents unauthorized persons from viewing potentially sensitive and/or personal information.

Configuring this setting will disable TouchID from unlocking the screensaver.

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

```
/usr/bin/security authorizationdb read system.login.screensaver 2>&1 | /usr/bin/grep -c '<string>authenticate-session-owner</string>'
```

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/security authorizationdb write system.login.screensaver "authenticate-session-owner"
```

ID	os_unlock_active_user_session_disable	
References	**800-53r5**	• IA-2, IA-2(5)
	800-171r2	• 3.5.1 • 3.5.2
	CCE	• CCE-91022-4

9.51. 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.

UUCP service is disabled at startup by default macOS.

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

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

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.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
	800-171r2	• 3.1.1 • 3.1.2
	CCE	• CCE-91024-0

Chapter 10. Password Policy

This section contains the configuration and enforcement of settings pertaining to password policies in macOS.

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

The password policy recommendations in the NIST 800-53 (Rev 5) and NIST 800-63B state that complexity rules should be organizationally defined. The values defined are based off of common complexity values. But your organization may define its own password complexity rules.

The settings outlined in this section adhere to the recommendations provided in this document for systems that utilize passwords for local accounts. If systems are integrated with a directory service, local password policies should align with domain password policies to the fullest extent feasible.

10.1. Disable Accounts after 35 Days of Inactivity

The macOS *MUST* be configured to disable accounts after 35 days of inactivity.

This rule prevents malicious users from making use of unused accounts to gain access to the system while avoiding detection.

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

```
/usr/bin/pwpolicy -getaccountpolicies 2> /dev/null | /usr/bin/tail +2 | /usr/bin/xmllint --xpath '//dict/key[text()="policyAttributeInactiveDays"]/following-sibling::integer[1]/text()' -
```

If the result is not **35**, this is a finding.

Remediation Description

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

This setting may be enforced using local policy or by a directory service.

To set local policy to disable an inactive user after 35 days, edit the current password policy to contain the following <dict> within the "policyCategoryAuthentication":

```
<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>
```

After saving the file and exiting to the command prompt, 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
```

See the password policy supplemental on more information on how to implement password policies on macOS.

ID	pwpolicy_account_inactivity_enforce	
References	**800-53r5**	• AC-2(3)
	800-171r2	• 3.5.5
		• 3.5.6
	CCE	• CCE-91028-1

10.2. Limit Consecutive Failed Login Attempts to 3

The macOS *MUST* be configured to limit the number of failed login attempts to a maximum of 3. When the maximum number of failed attempts is reached, the account *MUST* be locked for a period of time after.

This rule protects against malicious users attempting to gain access to the system via brute-force hacking methods.

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

```
/usr/bin/pwpolicy -getaccountpolicies 2> /dev/null | /usr/bin/tail +2 | /usr/bin/xmllint --xpath
'//dict/key[text()="policyAttributeMaximumFailedAuthentications"]/following-
sibling::integer[1]/text()' - | /usr/bin/awk '{ if ($1 <= 3) {print "yes"} else {print "no"}}'
```

If the result is not **yes**, 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.mobiledevice.passwordpolicy) payload type:

```
<key>maxFailedAttempts</key>
<integer>3</integer>
```

ID	pwpolicy_account_lockout_enforce	
References	**800-53r5**	• AC-7
	800-171r2	• 3.1.8
	CCE	• CCE-91029-9

10.3. Set Account Lockout Time to 15 Minutes

The macOS *MUST* be configured to enforce a lockout time period of at least 15 minutes when the maximum number of failed logon attempts is reached.

This rule protects against malicious users attempting to gain access to the system via brute-force hacking methods.

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

```
/usr/bin/pwpolicy -getaccountpolicies 2> /dev/null | /usr/bin/tail +2 | /usr/bin/xmllint --xpath '//dict/key[text()="autoEnableInSeconds"]/following-sibling::integer[1]/text()' - | /usr/bin/awk '{ if ($1/60 >= 15 ) {print "yes"} else {print "no"}}'
```

If the result is not **yes**, 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.mobiledevice.passwordpolicy) payload type:

```
<key>minutesUntilFailedLoginReset</key>
```

```
<integer>15</integer>
```

ID	pwpolicy_account_lockout_timeout_enforce	
References	**800-53r5**	• AC-7
	800-171r2	• 3.1.8
	CCE	• CCE-91030-7

10.4. Require Passwords Contain a Minimum of One Numeric Character

The macOS *MUST* be configured to require at least one numeric character be used when a password is created.

This rule enforces password complexity by requiring users to set passwords that are less vulnerable to malicious users.

The guidance for password based authentication in NIST 800-53 (Rev 5) and NIST 800-63B state that complexity rules should be organizationally defined. The values defined are based off of common complexity values. But your organization may define its own password complexity rules.

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

```
/usr/bin/pwpolicy -getaccountpolicies 2> /dev/null | /usr/bin/tail +2 | /usr/bin/xmllint --xpath '//dict/key[text()="policyIdentifier"]/following-sibling::*[1]/text()' - | /usr/bin/grep "requireAlphanumeric" -c
```

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.mobiledevice.passwordpolicy) payload type:

```
<key>requireAlphanumeric</key>
<true/>
```

ID	pwpolicy_alpha_numeric_enforce	
References	**800-53r5**	• IA-5(1)
	800-171r2	• 3.5.1 • 3.5.2 • 3.5.7 • 3.5.8 • 3.5.9 • 3.5.10
	CCE	• CCE-91031-5

10.5. Prohibit Password Reuse for a Minimum of 5 Generations

The macOS *MUST* be configured to enforce a password history of at least 5 previous passwords when a password is created.

This rule ensures that users are not allowed to re-use a password that was used in any of the 5 previous password generations.

Limiting password reuse protects against malicious users attempting to gain access to the system via brute-force hacking methods.

The guidance for password based authentication in NIST 800-53 (Rev 5) and NIST 800-63B state that complexity rules should be organizationally defined. The values defined are based off of common complexity values. But your organization may define its own password complexity rules.

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

```
/usr/bin/pwpolicy -getaccountpolicies 2> /dev/null | /usr/bin/tail +2 | /usr/bin/xmllint --xpath '//dict/key[text()="policyAttributePasswordHistoryDepth"]/following-sibling::*[1]/text()' - | /usr/bin/awk '{ if ($1 >= 5 ) {print "yes"} else {print "no"}}'
```

If the result is not **yes**, 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.mobiledevice.passwordpolicy) payload type:

```
<key>pinHistory</key>
<integer>5</integer>
```

ID	pwpolicy_history_enforce	
References	**800-53r5**	• IA-5(1)
	800-171r2	• 3.5.7 • 3.5.8 • 3.5.9 • 3.5.10
	CCE	• CCE-91034-9

10.6. Require Passwords Contain a Minimum of One Lowercase Character

The macOS *MUST* be configured to require at least one lower-case character be used when a password is created.

This rule enforces password complexity by requiring users to set passwords that are less vulnerable to malicious users.

The guidance for password based authentication in NIST 800-53 (Rev 5) and NIST 800-63B state that complexity rules should be organizationally defined. The values defined are based off of common complexity values. But your organization may define its own password complexity rules.

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

```
/usr/bin/pwpolicy -getaccountpolicies 2> /dev/null | /usr/bin/tail +2 | /usr/bin/xmllint --xpath '//dict/key[text()="minimumAlphaCharactersLowerCase"]/following-sibling::integer[1]/text()' - | /usr/bin/awk '{ if ($1 >= 1 ) {print "yes"} else {print "no"}}'
```

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

Remediation Description

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

This setting may be enforced using local policy or by a directory service.

To set local policy to require at least 1 lowercase letter, edit the current password policy to

contain the following <dict> within the "policyCategoryPasswordContent":

```
<dict>
<key>policyContent</key>
<string>policyAttributePassword matches &apos;(.*[a-z].*){1,}+&apos;</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>
```

After saving the file and exiting to the command prompt, 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
```

See the password policy supplemental on more information on how to implement password policies on macOS.

ID	pwpolicy_lower_case_character_enforce	
References	**800-53r5**	• IA-5(1)
	800-171r2	• 3.5.1 • 3.5.2 • 3.5.7 • 3.5.8 • 3.5.9 • 3.5.10
	CCE	• CCE-91035-6

10.7. Restrict Maximum Password Lifetime to 60 Days

The macOS *MUST* be configured to enforce a maximum password lifetime limit of at least 60 days.

This rule ensures that users are forced to change their passwords frequently enough to prevent malicious users from gaining and maintaining access to the system.

The guidance for password based authentication in NIST 800-53 (Rev 5) and NIST 800-63B state that complexity rules should be organizationally defined. The values defined are based off of common complexity values. But your organization may define its own password complexity rules.

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

```
/usr/bin/pwpolicy -getaccountpolicies 2> /dev/null | /usr/bin/tail +2 | /usr/bin/xmllint --xpath '//dict/key[text()="policyAttributeExpiresEveryNDays"]/following-sibling::*[1]/text()' -
```

If the result is not **60**, 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.mobiledevice.passwordpolicy) payload type:

```
<key>maxPINAgeInDays</key>
<integer>60</integer>
```

ID	pwpolicy_max_lifetime_enforce	
References	**800-53r5**	• IA-5
	800-171r2	• 3.5.1 • 3.5.2 • 3.5.7 • 3.5.8 • 3.5.9 • 3.5.10
	CCE	• CCE-91027-3

10.8. Require a Minimum Password Length of 15 Characters

The macOS *MUST* be configured to require a minimum of 15 characters be used when a password is created.

This rule enforces password complexity by requiring users to set passwords that are less

vulnerable to malicious users.

The guidance for password based authentication in NIST 800-53 (Rev 5) and NIST 800-63B state that complexity rules should be organizationally defined. The values defined are based off of common complexity values. But your organization may define its own password complexity rules.

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

```
/usr/bin/pwpolicy -getaccountpolicies 2> /dev/null | /usr/bin/tail +2 | /usr/bin/xmllint --xpath 'boolean(//*[contains(text(),"policyAttributePassword matches '\''.{15,}'\''")])' -
```

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.mobiledevice.passwordpolicy) payload type:

```
<key>minLength</key>
<integer>15</integer>
```

ID	pwpolicy_minimum_length_enforce	
References	**800-53r5**	• IA-5(1)
	800-171r2	• 3.5.1 • 3.5.2 • 3.5.7 • 3.5.8 • 3.5.9 • 3.5.10
	CCE	• CCE-91036-4

10.9. Set Minimum Password Lifetime to 24 Hours

The macOS *MUST* be configured to enforce a minimum password lifetime limit of 24 hours.

This rule discourages users from cycling through their previous passwords to get back to a preferred one.

The guidance for password based authentication in NIST 800-53 (Rev 5) and NIST 800-63B state that complexity rules should be organizationally defined. The values defined are based off of common complexity values. But your organization may define its own password complexity rules.

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

```
/usr/bin/pwpolicy -getaccountpolicies 2> /dev/null | /usr/bin/tail +2 | /usr/bin/xmllint --xpath '//dict/key[text()="policyAttributeMinimumLifetimeHours"]/following-sibling::integer[1]/text()' - | /usr/bin/awk '{ if ($1 >= 24 ) {print "yes"} else {print "no"}}'
```

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

Remediation Description

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

This setting may be enforced using local policy or by a directory service.

To set local policy to require a minimum password lifetime, edit the current password policy to contain the following <dict> within the "policyCategoryPasswordContent":

```
<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>
```

After saving the file and exiting to the command prompt, 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
```

See the password policy supplemental on more information on how to

implement password policies on macOS.

ID	pwpolicy_minimum_lifetime_enforce	
References	**800-53r5**	• IA-5
	800-171r2	• 3.5.7 • 3.5.8 • 3.5.9 • 3.5.10
	CCE	• CCE-91037-2

10.10. Prohibit Repeating, Ascending, and Descending Character Sequences

The macOS *MUST* be configured to prohibit the use of repeating, ascending, and descending character sequences when a password is created.

This rule enforces password complexity by requiring users to set passwords that are less vulnerable to malicious users.

The guidance for password based authentication in NIST 800-53 (Rev 5) and NIST 800-63B state that complexity rules should be organizationally defined. The values defined are based off of common complexity values. But your organization may define its own password complexity rules.

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

```
/usr/bin/pwpolicy -getaccountpolicies 2> /dev/null | /usr/bin/tail +2 | /usr/bin/xmllint --xpath '//dict/key[text()="policyIdentifier"]/following-sibling::*[1]/text()' - | /usr/bin/grep "allowSimple" -c
```

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.mobiledevice.passwordpolicy) payload type:

```
<key>allowSimple</key>
```

```
<false/>
```

ID	pwpolicy_simple_sequence_disable	
References	**800-53r5**	• IA-5(1)
	800-171r2	• 3.5.1 • 3.5.2 • 3.5.7 • 3.5.8 • 3.5.9 • 3.5.10
	CCE	• CCE-91039-8

10.11. Require Passwords Contain a Minimum of One Special Character

The macOS *MUST* be configured to require at least one special character be used when a password is created.

Special characters are those characters that are not alphanumeric. Examples include: ~ ! @ # $ % ^ *.

This rule enforces password complexity by requiring users to set passwords that are less vulnerable to malicious users.

The guidance for password based authentication in NIST 800-53 (Rev 5) and NIST 800-63B state that complexity rules should be organizationally defined. The values defined are based off of common complexity values. But your organization may define its own password complexity rules.

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

```
/usr/bin/pwpolicy -getaccountpolicies 2> /dev/null | /usr/bin/tail +2 | /usr/bin/xmllint --xpath 'boolean(//*[contains(text(),"policyAttributePassword matches '\''(.*[^a-zA-Z0-9].*){1,}'\''")])' -
```

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.mobiledevice.passwordpolicy) payload type:

```
<key>minComplexChars</key>
<integer>1</integer>
```

ID	pwpolicy_special_character_enforce	
References	**800-53r5**	• IA-5(1)
	800-171r2	• 3.5.1 • 3.5.2 • 3.5.7 • 3.5.8 • 3.5.9 • 3.5.10
	CCE	• CCE-91040-6

10.12. Automatically Remove or Disable Temporary or Emergency User Accounts within 72 Hours

The macOS is able to be configured to set an automated termination for 72 hours or less for all temporary or emergency accounts upon account creation.

Emergency administrator accounts are privileged accounts established in response to crisis situations where the need for rapid account activation is required. Therefore, emergency account activation may bypass normal account authorization processes. If these accounts are disabled, system maintenance during emergencies may not be possible, thus adversely affecting system availability.

Although the ability to create and use emergency administrator accounts is necessary for performing system maintenance during emergencies, these accounts present vulnerabilities to the system if they are not disabled and removed when they are no longer needed. Configuring the macOS to automatically remove or disable emergency accounts within 72 hours of creation mitigates the risks posed if one were to be created and accidentally left active once the crisis is resolved.

Emergency administrator accounts are different from infrequently used accounts (i.e., local logon accounts used by system administrators when network or normal logon is not available). Infrequently used accounts also remain available and are not subject to automatic termination dates. However, an emergency administrator account is normally a different account created for use by vendors or system maintainers.

To address access requirements, many operating systems can be integrated with enterprise-level

authentication/access mechanisms that meet or exceed access control policy requirements.

If temporary or emergency user accounts remain active when no longer needed or for an excessive period, these accounts may be targeted by attackers to gain unauthorized access. To mitigate this risk, automated termination of all temporary or emergency accounts *MUST* be set to 72 hours (or less) when the temporary or emergency account is created.

If no policy is enforced by a directory service, a password policy can be set with the "pwpolicy" utility. The variable names may vary depending on how the policy was set.

If there are no temporary or emergency accounts defined on the system, this is Not Applicable.

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

Verify **if** a password policy is enforced by a directory service by asking the System Administrator (SA) or Information System Security Officer (ISSO).

If no policy is enforced by a directory service, a password policy can be set with the "pwpolicy" utility. The variable names may vary depending on how the policy was set.

If there are no temporary or emergency accounts defined on the system, this is Not Applicable.

To check **if** the password policy is configured to disable a temporary or emergency account after 72 hours, run the following command to output the password policy to the screen, substituting the correct user name **in** place of username:

/usr/bin/pwpolicy **-u** username getaccountpolicies | tail **-n** +2

If there is no output, and password policy is not controlled by a directory service, this is a finding.

Otherwise, look **for** the line "<key>policyCategoryAuthentication</key>".

In the array that follows, there should be a <dict> section that contains a check <string> that allows users to log **in if** "policyAttributeCurrentTime" is less than the result of adding "policyAttributeCreationTime" to 72 hours (259299 seconds). The check might use a variable defined **in** its "policyParameters" section.

If the check does not exist or **if** the check adds too great an amount of time to "policyAttributeCreationTime", this is a finding.

If the result is not **N/A**, this is a finding.

Remediation Description

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

This setting may be enforced using local policy or by a directory service.

To set local policy to disable a temporary or emergency user, create a plain text file containing the following:

<dict> <key>policyCategoryAuthentication</key> <array> <dict> <key>policyContent</key> <string>policyAttributeCurrentTime < policyAttributeCreationTime+259299</string> <key>policyIdentifier</key> <string>Disable Tmp Accounts </string> </dict> </array> </dict>

After saving the file and exiting to the command prompt, run the following command to load the new policy file, substituting the correct user name in place of "username" and the path to the file in place of "/path/to/file".

/usr/bin/pwpolicy -u username setaccountpolicies /path/to/file

ID	pwpolicy_temporary_or_emergency_accounts_disable	
References	**800-53r5**	• AC-2(2)
	800-171r2	• N/A
	CCE	• CCE-91042-2

10.13. Require Passwords Contain a Minimum of One Uppercase Character

The macOS *MUST* be configured to require at least one uppercase character be used when a password is created.

This rule enforces password complexity by requiring users to set passwords that are less vulnerable to malicious users.

The guidance for password based authentication in NIST 800-53 (Rev 5) and NIST 800-63B state that complexity rules should be organizationally defined. The values defined are based off of common complexity values. But your organization may define its own password complexity rules.

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

```
/usr/bin/pwpolicy -getaccountpolicies 2> /dev/null | /usr/bin/tail +2 | /usr/bin/xmllint --xpath
'//dict/key[text()="minimumAlphaCharactersUpperCase"]/following-sibling::integer[1]/text()' - |
```

```
/usr/bin/awk '{ if ($1 >= 1 ) {print "yes"} else {print "no"}}'
```

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

Remediation Description

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

This setting may be enforced using local policy or by a directory service.

To set local policy to require at least 1 lowercase letter, edit the current password policy to contain the following <dict> within the "policyCategoryPasswordContent":

```
<dict>
<key>policyContent</key>
<string>policyAttributePassword matches &apos;(.*[A-Z].*){1,}+&apos;</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>
```

After saving the file and exiting to the command prompt, 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
```

See the password policy supplemental on more information on how to implement password policies on macOS.

ID	pwpolicy_upper_case_character_enforce

References	**800-53r5**	• IA-5(1)
	800-171r2	• 3.5.1 • 3.5.2 • 3.5.7 • 3.5.8 • 3.5.9 • 3.5.10
	CCE	• CCE-91043-0

Chapter 11. System Preferences

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

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

11.1. Disable Airplay Receiver

Airplay Receiver allows you to send content from another Apple device to be displayed on the screen as it's being played from your other device.

Support for Airplay Receiver 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('allowAirPlayIncomingRequests').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>allowAirPlayIncomingRequests</key>
<false/>
```

ID	sysprefs_airplay_receiver_disable	
References	**800-53r5**	• CM-7, CM-7(1)
	800-171r2	• 3.4.6
	CCE	• CCE-91044-8

11.2. Prevent Apple Watch from Terminating a Session Lock

Apple Watches are not an approved authenticator and their use *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 << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowAutoUnlock').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>allowAutoUnlock</key>
<false/>
```

ID	sysprefs_apple_watch_unlock_disable	
References	**800-53r5**	• AC-11
	800-171r2	• 3.1.10
	CCE	• CCE-91045-5

11.3. 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	sysprefs_automatic_login_disable	
References	**800-53r5**	• IA-2 • IA-5(13)
	800-171r2	• 3.5.1 • 3.5.2
	CCE	• CCE-91046-3

11.4. Enforce Auto Logout After 86400 Seconds of Inactivity

Auto logout *MUST* be configured to automatically terminate a user session and log out the after 86400 seconds of inactivity.

NOTE:The maximum that macOS can be configured for autologoff is 86400 seconds.

The automatic logout may cause disruptions to an organization's workflow and/or loss of data. Information System Security Officers (ISSOs) are advised to first fully weigh the potential risks posed to their organization before opting to disable the

automatic logout setting.

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

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

If the result is not **86400**, 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 (.GlobalPreferences) payload type:

```
<key>com.apple.autologout.AutoLogOutDelay</key>
<integer>86400</integer>
```

ID	sysprefs_automatic_logout_enforce	
References	**800-53r5**	• AC-12 • AC-2(5)
	800-171r2	• 3.1.11
	CCE	• CCE-91047-1

11.5. Disable Bluetooth When no Approved Device is Connected

The macOS system *MUST* be configured to disable Bluetooth unless there is an approved device connected.

Information System Security Officers (ISSOs) may make the risk-based decision not to disable Bluetooth, so as to maintain necessary 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.MCXBluetooth')\
.objectForKey('DisableBluetooth').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:

The following settings are in the (com.apple.ManagedClient.preferences) payload. This payload requires the additional settings to be sub-payloads within, containing their defined payload types.

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

```
<key>DisableBluetooth</key>
<true/>
```

ID	sysprefs_bluetooth_disable	
References	**800-53r5**	• AC-18, AC-18(3) • SC-8
	800-171r2	• 3.13.8
	CCE	• CCE-91048-9

11.6. 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.

The check and fix are for the currently logged in user. To get the currently logged in user, run the following.

```
CURRENT_USER=$( /usr/sbin/scutil <<< "show State:/Users/ConsoleUser" |
```

```
/usr/bin/awk '/Name :/ && ! /loginwindow/ { print $3 }' )
```

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	sysprefs_bluetooth_sharing_disable	
References	**800-53r5**	• AC-18(4) • AC-3 • CM-7, CM-7(1)
	800-171r2	• 3.1.1 • 3.1.2 • 3.1.16 • 3.4.7
	CCE	• CCE-91049-7

11.7. Disable CD/DVD Sharing

CD/DVD Sharing *MUST* be disabled.

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

```
/usr/bin/pgrep -q ODSAgent; /bin/echo $?
```

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 unload /System/Library/LaunchDaemons/com.apple.ODSAgent.plist
```

ID	sysprefs_cd_dvd_sharing_disable	
References	**800-53r5**	• CM-7, CM-7(1)
	800-171r2	• N/A
	CCE	• CCE-91127-1

11.8. Disable Content Caching Service

Content caching *MUST* be disabled.

Content caching is a macOS service that helps reduce Internet data usage and speed up software installation on Mac computers. It is not recommended for devices furnished to employees to act as a caching server.

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('allowContentCaching').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>allowContentCaching</key>
<false/>
```

ID	sysprefs_content_caching_disable

References	**800-53r5**	• CM-7, CM-7(1)
	800-171r2	• 3.4.6
	CCE	• CCE-91050-5

11.9. 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	sysprefs_critical_update_install_enforce	
References	**800-53r5**	• SI-2
	800-171r2	• N/A
	CCE	• CCE-91051-3

11.10. 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	sysprefs_diagnostics_reports_disable

References	800-53r5	• AC-20 • SC-7(10) • SI-11
	800-171r2	• 3.1.20
	CCE	• CCE-91052-1

11.11. Enforce FileVault

FileVault *MUST* be enforced.

The information system implements cryptographic mechanisms to protect the confidentiality and integrity of information stored on digital media during transport outside of controlled areas.

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

```
dontAllowDisable=$(/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.MCX')\
.objectForKey('dontAllowFDEDisable').js
EOS
)
fileVault=$(/usr/bin/fdesetup status | /usr/bin/grep -c "FileVault is On.")
if [[ "$dontAllowDisable" == "true" ]] && [[ "$fileVault" == 1 ]]; then
  echo "1"
else
  echo "0"
fi
```

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.MCX) payload type:

```
<key>dontAllowFDEDisable</key>
<true/>
```

ID	sysprefs_filevault_enforce	
References	**800-53r5**	• SC-28, SC-28(1)
	800-171r2	• 3.13.16
	CCE	• CCE-91053-9

11.12. 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	sysprefs_find_my_disable	
References	**800-53r5**	• AC-20 • CM-7, CM-7(1)
	800-171r2	• 3.1.20 • 3.4.6
	CCE	• CCE-91054-7

11.13. 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):

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

```
plist="$(/usr/bin/defaults read /Library/Preferences/com.apple.alf globalstate 2>/dev/null)"

if [[ "$profile" == "true" ]] && [[ "$plist" =~ [1,2] ]]; then
  echo "true"
else
  echo "false"
fi
```

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	sysprefs_firewall_enable	
References	**800-53r5**	• AC-4 • CM-7, CM-7(1) • SC-7, SC-7(12)
	800-171r2	• 3.1.3 • 3.1.5 • 3.1.18 • 3.4.6 • 3.13.1 • 3.13.2 • 3.13.5
	CCE	• CCE-91055-4

11.14. 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):

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

plist=$(/usr/bin/defaults read /Library/Preferences/com.apple.alf stealthenabled 2>/dev/null)

if [[ "$profile" == "true" ]] && [[ $plist == 1 ]]; then
  echo "true"
else
  echo "false"
fi
```

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/>
```

ID	sysprefs_firewall_stealth_mode_enable

References	**800-53r5**	• CM-7, CM-7(1) • SC-7, SC-7(16)
	800-171r2	• 3.4.6 • 3.13.1 • 3.13.2 • 3.13.5
	CCE	• CCE-91056-2

11.15. Apply Gatekeeper Settings to Block Applications from Unidentified Developers

The information system implements cryptographic mechanisms to authenticate software prior to installation.

Gatekeeper settings must be configured correctly to only allow the system to run applications downloaded from the Mac App Store or applications signed with a valid Apple Developer ID code. Administrator users will still have the option to override these settings on a per-app basis. Gatekeeper is a security feature that ensures that applications must be digitally signed by an Apple-issued certificate in order to run. Digital signatures allow the macOS to verify that the application has not been modified by a malicious third party.

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

```
/usr/sbin/spctl --status --verbose | /usr/bin/grep -c "developer id enabled"
```

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.systempolicy.control) payload type:

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

ID	sysprefs_gatekeeper_identified_developers_allowed

References	**800-53r5**	• CM-14 • CM-5 • SI-7(1), SI-7(15)
	800-171r2	• 3.4.5
	CCE	• CCE-91057-0

11.16. Configure Gatekeeper to Disallow End User Override

Gatekeeper *MUST* be configured with a configuration profile to prevent normal users from overriding its settings.

If users are allowed to disable Gatekeeper or set it to a less restrictive setting, malware could be introduced into the system.

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

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.systempolicy.managed')\
.objectForKey('DisableOverride').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.managed) payload type:

```
<key>DisableOverride</key>
<true/>
```

ID	sysprefs_gatekeeper_override_disallow

References	**800-53r5**	• CM-5 • SI-7(15)
	800-171r2	• 3.4.5
	CCE	• CCE-91058-8

11.17. 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	sysprefs_guest_access_smb_disable	
References	**800-53r5**	• AC-2, AC-2(9)
	800-171r2	• 3.5.1 • 3.5.2
	CCE	• CCE-91059-6

11.18. 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	sysprefs_guest_account_disable	
References	**800-53r5**	• AC-2, AC-2(9)
	800-171r2	• 3.5.1 • 3.5.2
	CCE	• CCE-91060-4

11.19. Disable Hot Corners

Hot corners *MUST* be disabled.

The information system conceals, via the session lock, information previously visible on the display with a publicly viewable image. Although hot comers can be used to initiate a session lock or to launch useful applications, they can also be configured to disable an automatic session lock from initiating. Such a configuration introduces the risk that a user might forget to manually lock the screen before stepping away from the computer.

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

```
/usr/bin/profiles -P -o stdout | /usr/bin/grep -Ec '"wvous-bl-corner" = 0|"wvous-br-corner" = 0|"wvous-tl-corner" = 0|"wvous-tr-corner" = 0'
```

If the result is not **4**, this is a finding.

Remediation Description

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

The following settings are in the (com.apple.ManagedClient.preferences) payload. This payload requires the additional settings to be sub-payloads within, containing their defined payload types.

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

```
<key>wvous-bl-corner</key>
<integer>0</integer>
<key>wvous-br-corner</key>
<integer>0</integer>
<key>wvous-tr-corner</key>
<integer>0</integer>
<key>wvous-tl-corner</key>
<integer>0</integer>
```

ID	sysprefs_hot_corners_disable	
References	**800-53r5**	• AC-11(1)
	800-171r2	• 3.1.10
	CCE	• CCE-91061-2

11.20. Disable Sending 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	sysprefs_improve_siri_dictation_disable	
References	**800-53r5**	• AC-20 • CM-7, CM-7(1) • SC-7(10)
	800-171r2	• 3.1.20 • 3.4.6
	CCE	• CCE-91062-0

11.21. Disable the Internet Accounts System Preference Pane

The Internet Accounts System Preference pane *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()="DisabledPreferencePanes"]/following-sibling::*[1]' - | /usr/bin/grep -c
com.apple.preferences.internetaccounts
```

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>DisabledPreferencePanes</key>
<array>
  <string>com.apple.preferences.internetaccounts</string>
</array>
```

ID	sysprefs_internet_accounts_prefpane_disable	
References	**800-53r5**	• AC-20 • CM-7(5)
	800-171r2	• 3.1.20
	CCE	• CCE-90938-2

11.22. 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	sysprefs_internet_sharing_disable	
References	**800-53r5**	• AC-20 • AC-4
	800-171r2	• 3.1.3 • 3.1.20
	CCE	• CCE-91063-8

11.23. Disable Location Services

Location Services *MUST* be disabled.

The information system *MUST* be configured to provide only essential capabilities. Disabling Location Services 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/defaults read /var/db/locationd/Library/Preferences/ByHost/com.apple.locationd.plist
LocationServicesEnabled
```

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/defaults write /var/db/locationd/Library/Preferences/ByHost/com.apple.locationd
LocationServicesEnabled -bool false; /bin/launchctl kickstart -k
system/com.apple.locationd
```

ID	sysprefs_location_services_disable	
References	**800-53r5**	• CM-7, CM-7(1) • SC-7(10)
	800-171r2	• 3.4.6
	CCE	• CCE-91064-6

11.24. 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	sysprefs_loginwindow_prompt_username_password_enforce	
References	**800-53r5**	• IA-2
	800-171r2	• 3.5.1 • 3.5.2
	CCE	• CCE-91065-3

11.25. 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.

The Media Sharing preference panel will still allow "Home Sharing" and "Share media with guests" to be checked but the service will not be enabled.

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.preferences.sharing.SharingPrefsExtension')\
  .objectForKey('homeSharingUIStatus'))
  let pref2 = ObjC.unwrap(
```

```
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.preferences.sharing.SharingPrefsExtension')\
  .objectForKey('legacySharingUIStatus'))
  let pref3 = ObjC.unwrap(
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.preferences.sharing.SharingPrefsExtension')\
  .objectForKey('mediaSharingUIStatus'))
  if ( pref1 == 0 && pref2 == 0 && pref3 == 0 ) {
    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.preferences.sharing.SharingPrefsExtension) payload type:

```
<key>homeSharingUIStatus</key>
<integer>0</integer>
<key>legacySharingUIStatus</key>
<integer>0</integer>
<key>mediaSharingUIStatus</key>
<integer>0</integer>
```

ID	sysprefs_media_sharing_disabled	
References	**800-53r5**	• AC-17 • AC-3
	800-171r2	• 3.1.1 • 3.1.2
	CCE	• CCE-91066-1

11.26. Disable Password Hints

Password hints *MUST* be disabled.

Password hints leak information about passwords that are currently in use and can lead to loss of confidentiality.

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('RetriesUntilHint').js
EOS
```

If the result is not **0**, 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>RetriesUntilHint</key>
<integer>0</integer>
```

ID	sysprefs_password_hints_disable	
References	**800-53r5**	• IA-6
	800-171r2	• 3.5.11
	CCE	• CCE-91067-9

11.27. 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	sysprefs_personalized_advertising_disable	
References	**800-53r5**	• AC-20 • CM-7, CM-7(1) • SC-7(10)
	800-171r2	• 3.1.20 • 3.4.6
	CCE	• CCE-91068-7

11.28. Disable Power Nap

Power Nap *MUST* be disabled.

Power Nap allows your Mac to perform actions while a Mac is asleep. This can interfere with USB power and may cause devices to stop functioning until a reboot and must therefore be disabled on all applicable systems.

The following Macs support Power Nap:

- MacBook (Early 2015 and later)
- MacBook Air (Late 2010 and later)
- MacBook Pro (all models with Retina display)
- Mac mini (Late 2012 and later)

- iMac (Late 2012 and later)
- Mac Pro (Late 2013 and later)

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

```
/usr/bin/pmset -g custom | /usr/bin/awk '/powernap/ { sum+=$2 } END {print sum}'
```

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/pmset -a powernap 0
```

ID	sysprefs_power_nap_disable	
References	**800-53r5**	• CM-7, CM-7(1)
	800-171r2	• 3.4.6
	CCE	• CCE-91069-5

11.29. Disable Printer Sharing

Printer Sharing *MUST* be disabled.

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

```
/usr/sbin/cupsctl | /usr/bin/grep -c "_share_printers=0"
```

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/cupsctl --no-share-printers
/usr/bin/lpstat -p | awk '{print $2}'| /usr/bin/xargs -I{} lpadmin -p {} -o printer-is-shared
=false
```

ID	sysprefs_printer_sharing_disable	
References	**800-53r5**	• CM-7, CM-7(1)
	800-171r2	• N/A
	CCE	• CCE-91134-7

11.30. 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" => true'
```

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 it's parent process. Requires supervision.

ID	sysprefs_rae_disable	
References	**800-53r5**	• AC-17
		• AC-3
	800-171r2	• 3.1.1
		• 3.1.2
	CCE	• CCE-91070-3

11.31. Disable Remote Management

Remote Management *MUST* be disabled.

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

```
/usr/libexec/mdmclient QuerySecurityInfo | /usr/bin/grep -c "RemoteDesktopEnabled = 0"
```

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

Remediation Description

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

```
/System/Library/CoreServices/RemoteManagement/ARDAgent.app/Contents/Resources/kickstart -deactivate -stop
```

ID	sysprefs_remote_management_disable	
References	**800-53r5**	• CM-7, CM-7(1)
	800-171r2	• N/A
	CCE	• CCE-91135-4

11.32. 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):

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

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.screensharing
```

NOTE - This will apply to the whole system

ID	sysprefs_screen_sharing_disable	
References	**800-53r5**	• AC-17 • AC-3
	800-171r2	• 3.1.1 • 3.1.2
	CCE	• CCE-91071-1

11.33. Enforce Session Lock After Screen Saver is Started

A screen saver *MUST* be enabled and the system *MUST* be configured to require a password to unlock once the screensaver has been on for a maximum of 5 seconds.

An unattended system with an excessive grace period is vulnerable to a malicious user.

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

```
/usr/bin/osascript -l JavaScript << EOS
function run() {
  let delay = ObjC.unwrap($.NSUserDefaults.alloc.initWithSuiteName('com.apple.screensaver')\
.objectForKey('askForPasswordDelay'))
  if ( delay <= 5 ) {
    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.screensaver) payload type:

```
<key>askForPasswordDelay</key>
<integer>5</integer>
```

ID	sysprefs_screensaver_ask_for_password_delay_enforce	
References	**800-53r5**	• AC-11
	800-171r2	• 3.1.10
	CCE	• CCE-91072-9

11.34. Enforce Screen Saver Password

Users *MUST* authenticate when unlocking the screen saver.

The screen saver acts as a session lock and prevents unauthorized users from accessing the current user's account.

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

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.screensaver')\
.objectForKey('askForPassword').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.screensaver) payload type:

```
<key>askForPassword</key>
<true/>
```

ID	sysprefs_screensaver_password_enforce

References	800-53r5	• AC-11
	800-171r2	• 3.1.10
	CCE	• CCE-91073-7

11.35. Enforce Screen Saver Timeout

The screen saver timeout *MUST* be set to 1200 seconds or a shorter length of time.

This rule ensures that a full session lock is triggered within no more than 1200 seconds of inactivity.

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

```
/usr/bin/osascript -l JavaScript << EOS
function run() {
  let timeout = ObjC.unwrap($.
NSUserDefaults.alloc.initWithSuiteName('com.apple.screensaver')\
.objectForKey('idleTime'))
  if ( timeout <= 1200 ) {
    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.screensaver) payload type:

```
<key>idleTime</key>
<integer>1200</integer>
```

ID	sysprefs_screensaver_timeout_enforce

References	**800-53r5**	• AC-11 • IA-11
	800-171r2	• 3.1.10
	CCE	• CCE-91074-5

11.36. 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.ironwood.support')\
.objectForKey('Ironwood Allowed').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.ironwood.support) payload type:

```
<key>Ironwood Allowed</key>
<false/>
```

ID	sysprefs_siri_disable	
References	**800-53r5**	• AC-20 • CM-7, CM-7(1) • SC-7(10)
	800-171r2	• 3.1.20 • 3.4.6
	CCE	• CCE-91075-2

11.37. 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" => true'
```

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	sysprefs_smbd_disable	
References	**800-53r5**	• AC-17 • AC-3
	800-171r2	• 3.1.1 • 3.1.2
	CCE	• CCE-91076-0

11.38. 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" => false'
```

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	sysprefs_ssh_enable	
References	**800-53r5**	• AC-17 • AC-3 • CM-7, CM-7(1) • IA-2(8)
	800-171r2	• 3.1.1 • 3.1.2 • 3.4.6 • 3.5.4
	CCE	• CCE-91078-6

11.39. 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 Preferences.

Some Preference Panes in System Preferences 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):

```
/usr/bin/security authorizationdb read system.preferences 2> /dev/null |  /usr/bin/grep -A 1 "<key>shared</key>" | /usr/bin/grep -c "<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/security authorizationdb read system.preferences > /tmp/system.preferences.plist
key_value=$(/usr/libexec/PlistBuddy -c "Print :shared" /tmp/system.preferences.plist 2>&1)
```

```
if [[ "$key_value" == *"Does Not Exist"* ]]; then
  /usr/libexec/PlistBuddy -c "Add :shared bool false" /tmp/system.preferences.plist
else
  /usr/libexec/PlistBuddy -c "Set :shared false" /tmp/system.preferences.plist
fi
/usr/bin/security authorizationdb write system.preferences <
/tmp/system.preferences.plist
```

ID	sysprefs_system_wide_preferences_configure	
References	800-53r5	• AC-6, AC-6(1), AC-6(2)
	800-171r2	• 3.1.5 • 3.1.6
	CCE	• CCE-91079-4

11.40. Configure macOS to Use an Authorized Time Server

Approved time server *MUST* be the only servers configured for use. As of macOS 10.13 only one time server is supported.

This rule ensures the uniformity of time stamps for information systems with multiple system clocks and systems connected over a network.

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('timeServer').js
EOS
```

If the result is not **time.nist.gov**, 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>timeServer</key>
<string>time.nist.gov</string>
```

ID	sysprefs_time_server_configure	
References	**800-53r5**	• AU-12(1) • SC-45(1)
	800-171r2	• 3.3.7
	CCE	• CCE-91080-2

11.41. Enable macOS Time Synchronization Daemon (timed)

The timed service *MUST* be enabled on all networked systems and configured to set time automatically from the approved time server.

This rule ensures the uniformity of time stamps for information systems with multiple system clocks and systems connected over a network.

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

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.timed')\
.objectForKey('TMAutomaticTimeOnlyEnabled').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:

The following settings are in the (com.apple.ManagedClient.preferences) payload. This payload requires the additional settings to be sub-payloads within, containing their defined payload types.

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

```
<key>TMAutomaticTimeOnlyEnabled</key>
```

```
<true/>
```

ID	sysprefs_time_server_enforce	
References	**800-53r5**	• AU-12(1) • SC-45(1)
	800-171r2	• 3.3.7
	CCE	• CCE-91081-0

11.42. Configure User Session Lock When a Smart Token is Removed

The screen lock *MUST* be configured to initiate automatically when the smart token is removed from the system.

Session locks are temporary actions taken when users stop work and move away from the immediate vicinity of the information system but do not want to log out because of the temporary nature of their absences. While a session lock is not an acceptable substitute for logging out of an information system for longer periods of time, they prevent a malicious user from accessing the information system when a user has removed their smart token.

Information System Security Officers (ISSOs) may make the risk-based decision not to enforce a session lock when a smart token is removed, so as to maintain necessary workflow capabilities, 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.security.smartcard')\
.objectForKey('tokenRemovalAction').js
EOS
```

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.security.smartcard) payload type:

```
<key>tokenRemovalAction</key>
<integer>1</integer>
```

ID	sysprefs_token_removal_enforce	
References	**800-53r5**	• AC-11
	800-171r2	• 3.1.10
	CCE	• CCE-91082-8

11.43. Disable TouchID for Unlocking the Device

TouchID enables the ability to unlock a Mac system with a user's fingerprint.

TouchID *MUST* be disabled for "Unlocking your Mac" on all macOS devices that are capable of using Touch ID.

The system *MUST* remain locked until the user establishes access using an authorized identification and authentication method.

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('allowFingerprintForUnlock').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>allowFingerprintForUnlock</key>
<false/>
```

ID	sysprefs_touchid_unlock_disable

References	**800-53r5**	• AC-11
	800-171r2	• 3.1.10
	CCE	• CCE-91083-6

11.44. Disable Wi-Fi Interface

The macOS system must be configured with Wi-Fi support software disabled if not connected to an authorized trusted network.

Allowing devices and users to connect to or from the system without first authenticating them allows untrusted access and can lead to a compromise or attack. Since wireless communications can be intercepted it is necessary to use encryption to protect the confidentiality of information in transit.Wireless technologies include for example microwave packet radio (UHF/VHF) 802.11x and Bluetooth. Wireless networks use authentication protocols (e.g. EAP/TLS PEAP) which provide credential protection and mutual authentication.

If the system requires Wi-Fi to connect to an authorized network, this is not applicable.

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

```
/usr/sbin/networksetup -listallnetworkservices | /usr/bin/grep -c "*Wi-Fi"
```

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

Remediation Description

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

To disable Wi-Fi on a macOS system, run the following command.

```
/usr/sbin/networksetup -setnetworkserviceenabled "Wi-Fi" off
```

ID	sysprefs_wifi_disable	
References	**800-53r5**	• AC-18, AC-18(1), AC-18(3)
		• AC-4
	800-171r2	• N/A
	CCE	• CCE-91084-4

Chapter 12. Inherent

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

12.1. Audit Record Reduction and Report Generation

The system *IS* configured with the ability provide and implement an audit record reduction and report generation capability.

Audit record reduction is a process that manipulates collected audit log information and organizes it into a summary format that is more meaningful to analysts. Audit record reduction and report generation capabilities do not always emanate from the same system or from the same organizational entities that conduct audit logging activities. The audit record reduction capability includes modern data mining techniques with advanced data filters to identify anomalous behavior in audit records. The report generation capability provided by the system can generate customizable reports. Time ordering of audit records can be an issue if the granularity of the timestamp in the record is insufficient.

Audit record reduction and report generation can be done with tools built into macOS such as auditreduce and praudit. These tools are protected by System Integrity Protection (SIP).

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

ID	audit_record_reduction_report_generation	
References	**800-53r5**	• AU-7
	800-171r2	• N/A

12.2. Ensure Seperate Execution Domain for Processes

The inherent configuration of the macOS *IS* in compliance as Apple has implemented multiple features Mandatory access controls (MAC), System Integrity Protection (SIP), and application sandboxing.

https://support.apple.com/guide/security/system-integrity-protection-secb7ea06b49/web

https://developer.apple.com/library/archive/documentation/Security/Conceptual/AppSandboxDesignGuide/AboutAppSandbox/AboutAppSandbox.html

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

ID	os_application_sandboxing

References	**800-53r5**	• SC-39
	800-171r2	• N/A

12.3. Configure the System to Implement Approved Cryptography to Protect Information

The information system *IS* configured to implement approved cryptography to protect information.

Use of weak or untested encryption algorithms undermines the purposes of utilizing encryption to protect data. The operating system must implement cryptographic modules that adhere to the higher standards that have been tested, validated, and approved by the federal government.

macOS Monterey has been submitted to the National Institute of Standards and Technology (NIST) and is in review for the cryptographic module for FIPS 140-3 validation.

https://csrc.nist.gov/Projects/cryptographic-module-validation-program/modules-in-process/Modules-In-Process-List

https://support.apple.com/en-us/HT201159

The technology supports this requirement and cannot be configured to be out of compliance. The technology inherently meets this requirement using FIPS Validated Cryptographic Modules.

ID	os_implement_cryptography	
References	**800-53r5**	• SC-13
	800-171r2	• 3.13.11

12.4. Configure the System to Protect Memory from Unauthorized Code Execution

The information system *IS* configured to implement non-executable data to protect memory from code execution.

Some adversaries launch attacks with the intent of executing code in non-executable regions of memory or in memory locations that are prohibited (e.g., buffer overflow attacks). Security safeguards (e.g., data execution prevention and address space layout randomization) can be employed to protect non-executable regions of memory. Data execution prevention safeguards can either be hardware-enforced or software-enforced; hardware-enforced methods provide the greater strength of mechanism.

macOS supports address space layout randomization (ASLR), position-independent executable (PIE), Stack Canaries, and NX stack and heap protection.

https://developer.apple.com/library/archive/documentation/Darwin/Conceptual/64bitPorting/

transition/transition.html

https://developer.apple.com/library/archive/qa/qa1788/_index.html

https://www.apple.com/macos/security/

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

ID	os_implement_memory_protection	
References	**800-53r5**	• SI-16
	800-171r2	• N/A

12.5. 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
	800-171r2	• 3.1.1
		• 3.1.2

12.6. 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 Malware Removal Tool (MRT): a technology included on all macOS systems. MRT is an agent that remediates based on automatic updates delivered from Apple. MRT will remove the malware upon receiving updated information and check for malware on restart and login.

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
	800-171r2	• N/A

12.7. Obscure Passwords

The information system *IS* configured to obscure feedback of authentication information during the authentication process to protect the information from possible exploitation by unauthorized individuals.

The inherent configuration of a macOS uses NSSecureTextField for any text field that receives a password, which automatically obscures text which is entered.

https://developer.apple.com/documentation/appkit/nssecuretextfield

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

ID	os_obscure_password	
References	**800-53r5**	• IA-5
		• IA-6
	800-171r2	• 3.5.1
		• 3.5.2
		• 3.5.11

12.8. Configure the System to Block Non-Privileged Users from Executing Privileged Functions

The information system *IS* configured to block standard users from executing privileged functions.

Privileged functions include disabling, circumventing, or altering implemented security safeguards and countermeasures.

The inherent configuration of the macOS does not allow for non-privileged users to be able to execute functions requiring privilege.

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

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

ID	os_prevent_priv_functions

References	**800-53r5**	• AC-6(10)
	800-171r2	• 3.1.7

12.9. Configure the System to Prevent the Unauthorized Disclosure of Data via Shared Resources

The information system *IS* configured to ensure that the unauthorized disclosure of data does not occur when resources are shared.

The inherent configuration of the macOS does not allow for resources to be shared between users without authorization.

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_prevent_unauthorized_disclosure	
References	**800-53r5**	• SC-4
	800-171r2	• 3.13.4

12.10. Prohibit Remote Activation of Collaborative Computing Devices

The inherent configuration of the macOS *IS* in compliance.

Apple has implemented a green light physically next to your camera that will glow when the camera is activated. There is an orange dot indicator by the Control Center pull down menu item to indicate when the system's microphone is listening or activated.

The macOS has built into the system, the ability to grant or deny access to the camera and microphone which requires the application to have an entitlement to use the device.

https://support.apple.com/guide/mac-help/use-the-built-in-camera-mchlp2980/mac

https://support.apple.com/guide/mac-help/control-access-to-your-camera-mchlf6d108da/mac

https://support.apple.com/guide/mac-help/control-access-to-your-microphone-on-mac-mchla1b1e1fe/12.0/mac/12.0

The technology partially supports this requirement and cannot be configured to be in full compliance.

ID	os_prohibit_remote_activation_collab_devices

References	**800-53r5**	• SC-15
	800-171r2	• N/A

12.11. Require users to reauthenticate when changing authenticators

Without reauthentication, users may access resources or perform tasks for which they do not have authorization. When operating systems provide the capability to change user authenticators, it is critical the user reauthenticate.

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

ID	os_reauth_users_change_authenticators	
References	**800-53r5**	• IA-11
	800-171r2	• N/A

12.12. Ensure all Federal Laws, Executive Orders, Directives, Policies, Regulations, Standards, and Guidance for Authentication to a Cryptographic Module are Met

The inherent configuration of the macOS *IS* in compliance by implementing mechanisms for authentication to a cryptographic module that meet the requirements of all applicable federal laws, Executive Orders, directives, policies, regulations, standards, and guidance for such authentication

macOS contains many open source projects that may use their own cryptographic libraries typically for the purposes of maintaining platform independence. These services are not covered by the Apple FIPS Validation of the CoreCrypto and CoreCrypto Kernel modules.

macOS Monterey has been submitted to the National Institute of Standards and Technology (NIST) and is in review for the cryptographic module for FIPS 140-3 validation.

https://csrc.nist.gov/Projects/cryptographic-module-validation-program/modules-in-process/IUT-List

https://support.apple.com/en-us/HT201159

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

ID	os_required_crypto_module

References	**800-53r5**	• IA-7
	800-171r2	• N/A

12.13. Configure the System to Separate User and System Functionality

The information system *IS* configured to separate user and system functionality.

Operating system management functionality includes functions necessary for administration and requires privileged user access. Allowing non-privileged users to access operating system management functionality capabilities increases the risk that non-privileged users may obtain elevated privileges. Operating system management functionality includes functions necessary to administer console, network components, workstations, or servers and typically requires privileged user access.

The inherent configuration of the macOS allows only privileged users to access operating system management functionalities.

https://developer.apple.com/library/archive/documentation/MacOSX/Conceptual/BPSystemStartup/Chapters/DesigningDaemons.html

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

ID	os_separate_functionality	
References	**800-53r5**	• MA-4(1) • SC-2
	800-171r2	• 3.13.3

12.14. Encrypt Stored Passwords

The information system *IS* configured to encrypt stored passwords.

Passwords need to be protected at all times, and encryption is the standard method for protecting passwords. If passwords are not encrypted, they can be plainly read (i.e., clear text) and easily compromised.

https://developer.apple.com/documentation/opendirectory/kodattributetypeauthenticationauthority

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

ID	os_store_encrypted_passwords

References	**800-53r5** **800-171r2**	• IA-5(1), IA-5(1)(c) • 3.5.7 • 3.5.8 • 3.5.9 • 3.5.10

12.15. Uniquely Identify Users and Processes

The macOS is a UNIX 03-compliant operating system. The system uniquely identifies and authenticates organizational users or processes.

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

ID	os_unique_identification	
References	**800-53r5** **800-171r2**	• IA-4 • N/A

12.16. Automatically Remove or Disable Emergency Accounts within 72 Hours

The macOS is able to be configured to automatically remove or disable emergency accounts within 72 hours or less.

Emergency administrator accounts are privileged accounts established in response to crisis situations where the need for rapid account activation is required. Therefore, emergency account activation may bypass normal account authorization processes. If these accounts are disabled, system maintenance during emergencies may not be possible, thus adversely affecting system availability.

Although the ability to create and use emergency administrator accounts is necessary for performing system maintenance during emergencies, these accounts present vulnerabilities to the system if they are not disabled and removed when they are no longer needed. Configuring the macOS to automatically remove or disable emergency accounts within 72 hours of creation mitigates the risks posed if one were to be created and accidentally left active once the crisis is resolved.

Emergency administrator accounts are different from infrequently used accounts (i.e., local logon accounts used by system administrators when network or normal logon is not available). Infrequently used accounts also remain available and are not subject to automatic termination dates. However, an emergency administrator account is normally a different account created for use by vendors or system maintainers.

To address access requirements, many operating systems can be integrated with enterprise-level

authentication/access mechanisms that meet or exceed access control policy requirements.

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

ID	pwpolicy_emergency_accounts_disable	
References	**800-53r5**	• AC-2(2)
	800-171r2	• N/A

12.17. Force Password Change at Next Logon

The macOS is able to be configured to force users to change their password at next logon.

Temporary passwords are often used for new users when accounts are created. However, once logged in to the system, users must be immediately prompted to change to a permanent password of their creation.

For a user to change their password at next logon, run the following command:

```
/usr/bin/pwpolicy -u [USER] -setpolicy "newPasswordRequired=1"
```

Replace [USER] with the username that must change the password at next logon

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

ID	pwpolicy_force_password_change	
References	**800-53r5**	• IA-5(1)
	800-171r2	• 3.5.1 • 3.5.2 • 3.5.7 • 3.5.8 • 3.5.9 • 3.5.10

12.18. Automatically Remove or Disable Temporary User Accounts within 72 Hours

The macOS is able to be configured to set an automated termination for 72 hours or less for all temporary accounts upon account creation.

If temporary user accounts remain active when no longer needed or for an excessive period, these accounts may be targeted by attackers to gain unauthorized access. To mitigate this risk, automated termination of all temporary accounts *MUST* be set to 72 hours (or less) when the temporary account is created.

If no policy is enforced by a directory service, a password policy can be set with the "pwpolicy" utility. The variable names may vary depending on how the policy was set.

If there are no temporary accounts defined on the system, this is Not Applicable.

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

ID	pwpolicy_temporary_accounts_disable	
References	**800-53r5**	• AC-2(2)
	800-171r2	• N/A

Chapter 13. 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.

13.1. Audit Record Reduction and Report Generation

The macOS should be configured to provide and implement the capability to process, sort, and search audit records for events of interest based on organizationally defined fields.

Events of interest can be identified by the content of audit records, including system resources involved, information objects accessed, identities of individuals, event types, event locations, event dates and times, Internet Protocol addresses involved, or event success or failure. Organizations may define event criteria to any degree of granularity required, such as locations selectable by a general networking location or by specific system component.

The technology does not support this requirement. This is an applicable-does not meet finding.

ID	audit_records_processing	
References	**800-53r5**	• AU-7(1)
	800-171r2	• N/A

13.2. Must authenticate peripherals 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
	800-171r2	• 3.5.1
		• 3.5.2

13.3. Configure Automated Flaw Remediation

The macOS system *MUST* be configured to determine the state of system components with regard to flaw remediation.

The technology does not support this requirement. This is an applicable-does not meet finding.

ID	os_continuous_monitoring	
References	**800-53r5**	• SI-2(2)
	800-171r2	• N/A

13.4. Protect Against Denial of Service Attacks by Ensuring Rate-Limiting Measures on Network Interfaces

The macOS should be configured to prevent Denial of Service (DoS) attacks by enforcing rate-limiting measures on network interfaces.

DoS attacks leave authorized users unable to access information systems, devices, or other network resources due to the actions of a malicious cyber threat actor. When this occurs, the organization must operate at degraded capacity; often resulting in an inability to accomplish its mission.

To prevent DoS attacks by ensuring rate-limiting measures on network interfaces, many operating systems can be integrated with enterprise-level firewalls and networking equipment that meet or exceed this requirement.

The technology does not support this requirement. This is an applicable-does not meet finding.

ID	os_protect_dos_attacks	
References	**800-53r5**	• SC-5
	800-171r2	• N/A

13.5. Employ Automated Mechanisms for Account Management Functions

The organization should employ automated mechanisms to support the management of information system accounts.

The use of automated mechanisms prevents against human error and provide a faster and more efficient means of relaying time-sensitive information and account management.

To employ automated mechanisms for account management functions, many operating systems can be integrated with an enterprise-level directory service that meets or exceeds this requirement.

The technology does not support this requirement. This is an applicable-does not meet finding.

ID	os_provide_automated_account_management	
References	**800-53r5**	• AC-2(1)
	800-171r2	• N/A

13.6. Require Devices to Reauthenticate when Changing Authenticators

The macOS should be configured to require users to reauthenticate when the device authenticator is changed.

Without reauthentication, users may access resources or perform tasks for which they are not authorization. When operating systems provide the capability to change device authenticators, it is critical the device reauthenticate.

The technology does not support this requirement. This is an applicable-does not meet finding.

ID	os_reauth_devices_change_authenticators	
References	**800-53r5**	• IA-11
	800-171r2	• N/A

13.7. Secure Name Address Resolution Service

The information system requests and performs data origin authentication and data integrity verification on the name/address resolution responses the system receives from authoritative sources.

macOS supports encrypted DNS settings with the com.apple.dnsSettings.managed payload, however, the system must be integrated with a DNS server that supports encrypted DNS. https://developer.apple.com/documentation/devicemanagement/dnssettings

The technology does not support this requirement. This is an applicable-does not meet finding.

ID	os_secure_name_resolution	
References	**800-53r5**	• SC-21
	800-171r2	• N/A

13.8. Disable Wi-Fi When Connected to Ethernet

The macOS should be configured to automatically disable Wi-Fi when connected to ethernet.

The use of Wi-Fi to connect to unauthorized networks may facilitate the exfiltration of mission data. Therefore, wireless networking capabilities internally embedded within information system components should be disabled when not intended to be used.

If the system requires Wi-Fi to connect to an authorized network, this is not applicable.

The technology does not support this requirement. This is an applicable-does not meet finding.

ID	sysprefs_wifi_disable_when_connected_to_ethernet	
References	**800-53r5**	• AC-18(1), AC-18(3) • AC-4
	800-171r2	• 3.1.3 • 3.1.17

Chapter 14. Not Applicable

This section contains the controls that are defined in the NIST 800-53 revision 5 but are not applicable when configuring a macOS system.

14.1. Access Control for Mobile Devices

A mobile device is a computing device that has a small form factor such that it can easily be carried by a single individual; is designed to operate without a physical connection; possesses local, non-removable or removable data storage; and includes a self-contained power source. Mobile device functionality may also include voice communication capabilities, on-board sensors that allow the device to capture information, and/or built-in features for synchronizing local data with remote locations. Examples include smart phones and tablets. Mobile devices are typically associated with a single individual. The processing, storage, and transmission capability of the mobile device may be comparable to or merely a subset of notebook/desktop systems, depending on the nature and intended purpose of the device. Protection and control of mobile devices is behavior or policy-based and requires users to take physical action to protect and control such devices when outside of controlled areas. Controlled areas are spaces for which organizations provide physical or procedural controls to meet the requirements established for protecting information and systems.

Due to the large variety of mobile devices with different characteristics and capabilities, organizational restrictions may vary for the different classes or types of such devices. Usage restrictions and specific implementation guidance for mobile devices include configuration management, device identification and authentication, implementation of mandatory protective software, scanning devices for malicious code, updating virus protection software, scanning for critical software updates and patches, conducting primary operating system (and possibly other resident software) integrity checks, and disabling unnecessary hardware.

Usage restrictions and authorization to connect may vary among organizational systems. For example, the organization may authorize the connection of mobile devices to its network and impose a set of usage restrictions, while a system owner may withhold authorization for mobile device connection to specific applications or impose additional usage restrictions before allowing mobile device connections to a system.

The technology does not support this requirement. This is an applicable-does not meet finding.

ID	os_access_control_mobile_devices	
References	**800-53r5**	• AC-19
	800-171r2	• N/A

14.2. Configure the System to Uniquely Identify and Authenticate Non-Organizational Users

The information system uniquely identifies and authenticates non-organizational users (or processes acting on behalf of non-organizational users).

This requirement is NA for this technology.

ID	os_identify_non-org_users	
References	**800-53r5**	• IA-8
	800-171r2	• N/A

14.3. Information Input Validation

Check the validity of the following information inputs: organization-defined information inputs to the systems.

Checking the valid syntax and semantics of system inputs—including character set, length, numerical range, and acceptable values—verifies that inputs match specified definitions for format and content. For example, if the organization specifies that numerical values between 1-100 are the only acceptable inputs for a field in a given application, inputs of "387," "abc," or "%K%" are invalid inputs and are not accepted as input to the system. Valid inputs are likely to vary from field to field within a software application. Applications typically follow well-defined protocols that use structured messages (i.e., commands or queries) to communicate between software modules or system components. Structured messages can contain raw or unstructured data interspersed with metadata or control information. If software applications use attacker-supplied inputs to construct structured messages without properly encoding such messages, then the attacker could insert malicious commands or special characters that can cause the data to be interpreted as control information or metadata. Consequently, the module or component that receives the corrupted output will perform the wrong operations or otherwise interpret the data incorrectly. Prescreening inputs prior to passing them to interpreters prevents the content from being unintentionally interpreted as commands. Input validation ensures accurate and correct inputs and prevents attacks such as cross-site scripting and a variety of injection attacks.

This requirement is NA for this technology.

ID	os_information_validation	
References	**800-53r5**	• SI-10
	800-171r2	• N/A

14.4. Managed Access Control Points

Route remote accesses through authorized and managed network access control points.

Organizations consider the Trusted Internet Connections (TIC) initiative requirements for external network connections since limiting the number of access control points for remote access reduces attack surfaces.

This requirement is NA for this technology.

ID	os_managed_access_control_points

References	**800-53r5**	• AC-17(3)
	800-171r2	• N/A

14.5. Configure the System for Nonlocal Maintenance

Nonlocal maintenance and diagnostic activities are those activities conducted by individuals communicating through a network, either an external network or an internal network.

This requirement is NA for this technology.

ID	os_nonlocal_maintenance	
References	**800-53r5**	• MA-4
	800-171r2	• 3.7.5

Chapter 15. Supplemental

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

15.1. CIS Manual Recommendations

List of CIS recommendations that are manual check in the CIS macOS Benchmark.

Section	Install Updates, Patches and Additional Security Software
Recommendations	1.8 Ensure Computer Name Does Not Contain PII or Protected Organizational Information

Section	System Preferences
Recommendations	2.5.1.2 Ensure all user storage APFS volumes are encrypted 2.5.1.3 Ensure all user storage CoreStorage volumes are encrypted 2.5.4 Audit Location Services Access 2.6.1.1 Audit iCloud Keychain 2.6.1.2 Audit iCloud Drive 2.6.2 Audit App Store Password Settings 2.11 Audit Siri Settings 2.12 Audit Universal Control Settings 2.13 Audit Touch ID 2.14 Audit Notification & Focus Settings 2.15 Audit Passwords System Preference Setting

Section	Logging and Auditing
Recommendations	3.7 Audit Software Inventory

Section	System Access, Authentication and Authorization
Recommendations	5.2.3 Ensure Complex Password Must Contain Alphabetic Characters Is Configured 5.2.4 Ensure Complex Password Must Contain Numeric Character Is Configured 5.2.5 Ensure Complex Password Must Contain Special Character Is Configured 5.2.6 Ensure Complex Password Must Contain Uppercase and Lowercase Characters Is Configured

Section	Applications
Recommendations	7.1.1 Ensure Protect Mail Activity in Mail is Enabled 7.2.2 Audit History and Remove History Items 7.2.3 Audit Passwords System Preference Setting 7.2.6 Audit Hide IP Address in Safari Setting

15.2. 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

15.3. FileVault Supplemental

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

FileVault currently only uses password-based authentication and cannot be done using a smartcard or any other type of multi-factor authentication.

15.4. 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

Port	Service
79	Finger
20, 21	File Transfer Protocol (FTP)
80	HTTP
icmp	ping
143	Internet Message Access Protocol (IMAP)
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').objectIsForcedForKey('E
nableFirewall')")

#### 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
```

15.5. Password Policy Supplemental

The supplemental guidance found in this section is applicable for the following 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>(policyAttributeFailedAuthentications &lt;
policyAttributeMaximumFailedAuthentications) OR (policyAttributeCurrentTime &gt;
(policyAttributeLastFailedAuthenticationTime + autoEnableInSeconds))</string>
      <key>policyIdentifier</key>
      <string>Authentication Lockout</string>
      <key>policyParameters</key>
      <dict>
        <key>autoEnableInSeconds</key>
        <integer>300</integer>
        <key>policyAttributeMaximumFailedAuthentications</key>
        <integer>3</integer>
      </dict>
    </dict>
    <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>policyCategoryPasswordChange</key>
    <array>
     <dict>
      <key>policyContent</key>
      <string>policyAttributeCurrentTime &gt; policyAttributeLastPasswordChangeTime +
(policyAttributeExpiresEveryNDays * 24 * 60 * 60)</string>
      <key>policyIdentifier</key>
      <string>Password Expires after 60 days</string>
      <key>policyParameters</key>
      <dict>
       <key>policyAttributeExpiresEveryNDays</key>
       <integer>60</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>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>
 <dict>
  <key>policyContent</key>
  <string>policyAttributePassword matches '.{15,}+'</string>
  <key>policyIdentifier</key>
  <string>Must be at least 15 characters</string>
  <key>policyParameters</key>
  <dict>
   <key>minimumLength</key>
   <integer>15</integer>
  </dict>
 </dict>
 <dict>
  <key>policyContent</key>
  <string>policyAttributePassword matches '(.*[0-9].*){1,}+'</string>
  <key>policyIdentifier</key>
  <string>Must have at least 1 numeric value</string>
  <key>policyParameters</key>
  <dict>
   <key>minimumNumericCharacters</key>
   <integer>2</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>policyAttributePassword matches '(.*[A-Za-z].*){1,}+'</string>
    <key>policyIdentifier</key>
    <string>Must have at least 1 Letter</string>
    <key>policyParameters</key>
    <dict>
     <key>minimumAlphaCharacters</key>
     <integer>1</integer>
    </dict>
   </dict>
   <dict>
    <key>policyContent</key>
    <string>policyAttributePassword matches '(.*[^a-zA-Z0-9].*){1,}+'</string>
    <key>policyIdentifier</key>
    <string>Must have at least 1 special characters</string>
    <key>policyParameters</key>
    <dict>
     <key>minimumSymbols</key>
     <integer>1</integer>
    </dict>
   </dict>
   <dict>
    <key>policyContent</key>
    <string>none policyAttributePasswordHashes in policyAttributePasswordHistory</string>
    <key>policyIdentifier</key>
    <string>Cannot match the last 5 passwords</string>
    <key>policyParameters</key>
    <dict>
     <key>policyAttributePasswordHistoryDepth</key>
     <integer>5</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.

15.6. 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.
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/>
```