macOS 15.0

Security Configuration - Apple macOS 15 (Sequoia)
STIG - Ver 1, Rel 3

Sequoia Guidance, Revision 2.0 (2025-07-01)

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 15.0 system against the Apple macOS 15 (Sequoia) STIG - Ver 1, Rel 3 security baseline.

Chapter 3. Authors

macOS Security Compliance Project

Dan Brodjieski	National Aeronautics and Space Administration
Allen Golbig	Jamf
Bob Gendler	National Institute of Standards and Technology
Aaron Kegerreis	Defense Information Systems Agency

Chapter 4. Acronyms and Definitions

Table 1. Acronyms and Abbreviations

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

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

Table 2. Definitions

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

Chapter 5. Applicable Documents

5.1. Government Documents

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

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

Table 4. Defense Information Systems Agency (DISA)

Document Number or Descriptor	Document Title
STIG Ver 1, Rel 3	*Apple macOS 15 (Sequoia) STIG*

Table 5. Cybersecurity Maturity Model Certification (CMMC)

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

Table 6. Committee on National Security Systems (CNSS)

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

5.2. Non-Government Documents

Table 7. Apple

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

Table 8. Center for Internet Security

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

Chapter 6. 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 /var/audit
```

ID	audit_acls_files_configure	
References	800-53r5	¥ AU-9
	DISA STIG(s)	¥ APPL-15-000030
	CCE	¥ CCE-94101-3

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 /var/audit | /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 /var/audit
```

ID	audit_acls_folders_configure	
References	800-53r5	¥ AU-9
	DISA STIG(s)	¥ APPL-15-000031
	CCE	¥ CCE-94102-1

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 NOT enabled by default on macOS Sequoia.

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

```
LAUNCHD_RUNNING=$(/bin/launchctl list | /usr/bin/grep -c com.apple.auditd)
AUDITD_RUNNING=$(/usr/sbin/audit -c | /usr/bin/grep -c "AUC_AUDITING")
if [[ $LAUNCHD_RUNNING == 1 ]] && [[ -e /etc/security/audit_control ]] && [[
```

```
$AUDITD_RUNNING == 1 ]]; then
Ê echo "pass"
else
Ê echo "fail"
fi
```

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

Remediation Description

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

```
if [[ ! -e /etc/security/audit_control ]] && [[ -e
/etc/security/audit_control.example ]];then
Ê /bin/cp /etc/security/audit_control.example /etc/security/audit_control
fi

/bin/launchctl enable system/com.apple.auditd
/bin/launchctl bootstrap system
/System/Library/LaunchDaemons/com.apple.auditd.plist
/usr/sbin/audit -i
```

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)
	DISA STIG(s)	¥ APPL-15-001003
	CCE	¥ CCE-94104-7

6.4. Configure Audit Capacity Warning

The audit service *MUST* be configured to notify the system administrator when the amount of free disk space remaining reaches an organization defined value.

This rule ensures that the system administrator is notified in advance that action is required to free up more disk space for audit logs.

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

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

If the result is not 11, this is a finding.

Remediation Description

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

```
/usr/bin/sed -i.bak 's/.*minfree.*/minfree:11/' /etc/security/audit_control;
/usr/sbin/audit -s
```

ID	audit_configure_capacity_notify	
References	800-53r5	¥ AU-5(1)
	DISA STIG(s)	¥ APPL-15-001030
	CCE	¥ CCE-94105-4

6.5. Configure Audit_Control to Not Contain Access Control Lists

/etc/security/audit_control *MUST* not contain Access Control Lists (ACLs).

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

```
/bin/ls -le /etc/security/audit_control | /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 /etc/security/audit_control
```

ID	audit_control_acls_configure	
References	800-53r5	¥ AU-9
	DISA STIG(s)	¥ APPL-15-001140
	CCE	¥ CCE-94106-2

6.6. Configure Audit_Control Group to Wheel

/etc/security/audit_control *MUST* have the group set to wheel.

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

```
/bin/ls -dn /etc/security/audit_control | /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 /etc/security/audit_control
```

ID	audit_control_group_configure	
References	800-53r5	¥ AU-9
	DISA STIG(s)	¥ APPL-15-001110
	CCE	¥ CCE-94107-0

6.7. Configure Audit_Control Owner to Mode 440 or Less Permissive

/etc/security/audit_control *MUST* be configured so that it is readable only by the root user and group wheel.

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

```
/bin/ls -l /etc/security/audit_control | /usr/bin/awk '!/-r--[r-]-----
|current|total/{print $1}' | /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:

```
/bin/chmod 440 /etc/security/audit_control
```

ID	audit_control_mode_configure	
References	800-53r5	¥ AU-9
	DISA STIG(s)	¥ APPL-15-001130
	CCE	¥ CCE-94108-8

6.8. Configure Audit_Control Owner to Root

/etc/security/audit_control *MUST* have the owner set to root.

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

```
/bin/ls -dn /etc/security/audit_control | /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 /etc/security/audit_control
```

ID	audit_control_owner_configure	
References	800-53r5	¥ AU-9
	DISA STIG(s)	¥ APPL-15-001120
	CCE	¥ CCE-94109-6

6.9. 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
	DISA STIG(s)	¥ APPL-15-001010
	CCE	¥ CCE-94111-2

6.10. 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 /var/audit/*
```

ID	audit_files_group_configure

References	800-53r5	¥ AU-9
	DISA STIG(s)	¥ APPL-15-001014
	CCE	¥ CCE-94112-0

6.11. 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 /var/audit/*
```

ID	audit_files_mode_configure	
References	800-53r5	¥ AU-9
	DISA STIG(s)	¥ APPL-15-001016
	CCE	¥ CCE-94113-8

6.12. 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 /var/audit/*
```

ID	audit_files_owner_configure	
References	800-53r5	¥ AU-9
	DISA STIG(s)	¥ APPL-15-001012
	CCE	¥ CCE-94114-6

6.13. 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)
	DISA STIG(s)	¥ APPL-15-001044
	CCE	¥ CCE-94115-3

6.14. 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)
	DISA STIG(s)	¥ APPL-15-001001
	CCE	¥ CCE-94116-1

6.15. 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)
	DISA STIG(s)	¥ APPL-15-001024
	CCE	¥ CCE-94117-9

6.16. 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)
	DISA STIG(s)	¥ APPL-15-001020
	CCE	¥ CCE-94118-7

6.17. Configure System to Audit All Changes of Object Attributes

The audit system *MUST* be configured to record enforcement actions of 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., modifications to a file by applying file permissions).

This configuration ensures that audit lists include events in which enforcement actions 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_configure

References	800-53r5	¥ AC-2(12) ¥ AU-12 ¥ AU-2 ¥ AU-9 ¥ CM-5(1) ¥ MA-4(1)
	DISA STIG(s)	¥ APPL-15-001021
	CCE	¥ CCE-94119-5

6.18. 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)
	DISA STIG(s)	¥ APPL-15-001022
	CCE	¥ CCE-94121-1

6.19. 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)
	DISA STIG(s)	¥ APPL-15-001023
	CCE	¥ CCE-94122-9

6.20. 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)
	DISA STIG(s)	¥ APPL-15-001002
	CCE	¥ CCE-94123-7

6.21. 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 /var/audit
```

ID	audit_folder_group_configure	
References	800-53r5	¥ AU-9
	DISA STIG(s)	¥ APPL-15-001015
	CCE	¥ CCE-94124-5

6.22. Configure Audit Log Folders to be Owned by Root

Audit log folders *MUST* be owned by root.

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

Audit logs contain sensitive data about the system and users. If log folders 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 /var/audit
```

ID	audit_folder_owner_configure	
References	800-53r5	¥ AU-9
	DISA STIG(s)	¥ APPL-15-001013
	CCE	¥ CCE-94125-2

6.23. 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 /var/audit
```

ID	audit_folders_mode_configure	
References	800-53r5	¥ AU-9
	DISA STIG(s)	¥ APPL-15-001017
	CCE	¥ CCE-94126-0

6.24. Configure Audit Retention to 1d

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 "1d", 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 1d, 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:1d/'
/etc/security/audit_control; /usr/sbin/audit -s
```

ID	audit_retention_configure	
References	800-53r5	¥ AU-11
		¥ AU-4
	DISA STIG(s)	¥ APPL-15-001029
	CCE	¥ CCE-94130-2

6.25. 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)
	DISA STIG(s)	¥ APPL-15-001031
	CCE	¥ CCE-94131-0

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)
	DISA STIG(s)	¥ APPL-15-003050
	CCE	¥ CCE-94132-8

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)
	DISA STIG(s)	¥ APPL-15-003051
	CCE	¥ CCE-94133-6

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)
	DISA STIG(s)	¥ APPL-15-003052
	CCE	¥ CCE-94134-4

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)
	DISA STIG(s)	¥ APPL-15-003030
	CCE	¥ CCE-94135-1

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
	DISA STIG(s)	¥ APPL-15-001060
	CCE	¥ CCE-94137-7

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

ID	auth_smartcard_enforce	
References	800-53r5	¥ IA-2, IA-2(1), IA-2(12), IA-2(2), IA-2(6), IA-2(8) ¥ IA-5(2)
	DISA STIG(s)	¥ APPL-15-003020
	CCE	¥ CCE-94138-5

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

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

Remediation Description

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

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

for file in $(ls ${include_dir}); do
Ê if [[ "$file" == "100-macos.conf" ]]; then
```

```
Ê     continue
Ê fi
Ê if [[ "$file" == "01-mscp-sshd.conf" ]]; then
Ê     break
Ê fi
Ê /bin/mv ${include_dir}${file} ${include_dir}20-${file}
done
```

ID	auth_ssh_password_authentication_disable	
References	800-53r5	¥ IA-2, IA-2(1), IA-2(2), IA-2(6), IA-2(8) ¥ IA-5(2) ¥ MA-4
	DISA STIG(s)	¥ APPL-15-001150
	CCE	¥ CCE-94139-3

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)
	DISA STIG(s)	¥ APPL-15-002014
	CCE	¥ CCE-94140-1

8.2. 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)
	DISA STIG(s)	¥ APPL-15-002042
	CCE	¥ CCE-94142-7

8.3. 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)
	DISA STIG(s)	¥ APPL-15-002012
	CCE	¥ CCE-94143-5

8.4. 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)
	DISA STIG(s)	¥ APPL-15-002041
	CCE	¥ CCE-94144-3

8.5. Disable the iCloud Freeform Services

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

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

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

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

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

Remediation Description

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

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

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

```
<false/>
```

ID	icloud_freeform_disable	
References	800-53r5	¥ AC-20, AC-20(1) ¥ CM-7, CM-7(1) ¥ SC-7(10)
	DISA STIG(s)	¥ APPL-15-002270
	CCE	¥ CCE-94145-0

8.6. Disable iCloud Game Center

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

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

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

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

Remediation Description

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

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

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

ID	icloud_game_center_disable

References	800-53r5	¥ AC-20, AC-20(1) ¥ CM-7, CM-7(1) ¥ SC-7(10)
	DISA STIG(s)	¥ APPL-15-002160
	CCE	¥ CCE-94146-8

8.7. 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)
	DISA STIG(s)	¥ APPL-15-002040
	CCE	¥ CCE-94147-6

8.8. 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)
	DISA STIG(s)	¥ APPL-15-002015
	CCE	¥ CCE-94148-4

8.9. 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)
	DISA STIG(s)	¥ APPL-15-002016
	CCE	¥ CCE-94149-2

8.10. 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)
	DISA STIG(s)	¥ APPL-15-002043
	CCE	¥ CCE-94150-0

8.11. 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)
	DISA STIG(s)	¥ APPL-15-002170
	CCE	¥ CCE-94151-8

8.12. 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)
	DISA STIG(s)	¥ APPL-15-002013
	CCE	¥ CCE-94152-6

8.13. 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)
	DISA STIG(s)	¥ APPL-15-002150
	CCE	¥ CCE-94153-4

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 AppleID and Internet Account Modifications

The system *MUST* disable account modification.

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

This prevents the addition of unauthorized accounts.

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

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

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

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

Remediation Description

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

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

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

ID	os_account_modification_disable	
References	800-53r5	¥ AC-20, AC-20(1) ¥ CM-7, CM-7(1)
	DISA STIG(s)	¥ APPL-15-002120
	CCE	¥ CCE-94155-9

9.2. Disable AirDrop

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

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

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

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

Remediation Description

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

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

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

ID	os_airdrop_disable	
References	800-53r5	¥ AC-20 ¥ AC-3 ¥ CM-7, CM-7(1)
	DISA STIG(s)	¥ APPL-15-002009
	CCE	¥ CCE-94156-7

9.3. Disable Apple ID Setup during Setup Assistant

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

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

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

```
/usr/bin/osascript -l JavaScript << 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
	DISA STIG(s)	¥ APPL-15-002035
	CCE	¥ CCE-94159-1

9.4. 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
	DISA STIG(s)	¥ APPL-15-004001
	CCE	¥ CCE-94161-7

9.5. 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
	DISA STIG(s)	¥ APPL-15-004002
	CCE	¥ CCE-94162-5

9.6. 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/libexec/mdmclient QuerySecurityInfo | /usr/bin/grep -c
"AuthenticatedRootVolumeEnabled = 1;"
```

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

Remediation Description

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

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

! 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)
	DISA STIG(s)	¥ APPL-15-005070
	CCE	¥ CCE-94164-1

9.7. 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)
	DISA STIG(s)	¥ APPL-15-002005
	CCE	¥ CCE-94169-0

9.8. Disable Camera

It is detrimental for operating systems to provide, or install by default, functionality exceeding

requirements or mission objectives. These unnecessary capabilities or services are often overlooked and therefore may remain unsecured. They increase the risk to the platform by providing additional attack vectors.

Failing to disconnect from collaborative computing devices (i.e., cameras) can result in subsequent compromises of organizational information. Providing easy methods to physically disconnect from such devices after a collaborative computing session helps to ensure that participants carry out the disconnect activity without having to go through complex and tedious procedures.

This requirement is not applicable to mobile devices (smartphones and tablets), where the use of the camera is a local AO decision.

This requirement is not applicable to dedicated VTC suites located in approved VTC locations that are centrally managed.

For an external camera, if there is not a method for the operator to manually disconnect camera at the end of collaborative computing sessions, this is a finding.

For a built-in camera, the camera must be protected by a camera cover (e.g., laptop camera cover slide) when not in use. If the built-in camera is not protected with a camera cover, or is not physically disabled, this is a finding.

If the camera is not disconnected, covered, or physically disabled, the following configuration is required.

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

ID	os_camera_disable

References	800-53r5	¥ N/A
	DISA STIG(s)	¥ APPL-15-002017
	CCE	¥ CCE-94172-4

9.9. 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.

> **!** This rule is marked as manual and may not be able to be automated. It is also excluded in the compliance scan and will not report any results.

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
	DISA STIG(s)	¥ APPL-15-003001
	CCE	¥ CCE-94174-0

9.10. Enforce Installation of XProtect Remediator and Gatekeeper Updates Automatically

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

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

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

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

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

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

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

Remediation Description

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

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

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

ID	os_config_data_install_enforce	
References	800-53r5	¥ SI-2(5) ¥ SI-3
	DISA STIG(s)	¥ APPL-15-005130
	CCE	¥ CCE-94176-5

9.11. Disable Dictation

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

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

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

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

Remediation Description

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

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

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

ID	os_dictation_disable	
References	800-53r5	¥ AC-20 ¥ CM-7, CM-7(1) ¥ SC-7(10)
	DISA STIG(s)	¥ APPL-15-002230
	CCE	¥ CCE-94180-7

9.12. Disable Erase Content and Settings

Erase Content and Settings *MUST* be disabled.

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

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowEraseContentAndSettings').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>allowEraseContentAndSettings</key>
<false/>
```

ID	os_erase_content_and_settings_disable	
References	800-53r5	¥ CM-7, CM-7(1)
	DISA STIG(s)	¥ APPL-15-005061
	CCE	¥ CCE-94185-6

9.13. Must Use ESS

The approved ESS solution *MUST* be installed and configured to run.

The macOS system must employ automated mechanisms to determine the state of system components. The DoD requires the installation and use of an approved ESS solution to be implemented on the operating system. For additional information, reference all applicable ESS OPORDs and FRAGOs on SIPRNET.

! This rule is marked as manual and may not be able to be automated. It is also excluded in the compliance scan and will not report any results.

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

```
Ask the System Administrator (SA) or Information System Security Officer (ISSO) if the
approved ESS solution is loaded on the system.
If the installed components of the ESS solution are not at the DoD approved minimal
versions, 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:

Install the approved ESS solution onto the system.

ID	os_ess_installed	
References	800-53r5	¥ N/A
	DISA STIG(s)	¥ N/A
	CCE	¥ CCE-94187-2

9.14. Disable FaceTime.app

The macOS built-in FaceTime.app *MUST* be disabled.

The FaceTime.app establishes a connection to Apple's iCloud service, even when security controls

have been put in place to disable iCloud access.

> " Apple has deprecated the use of application restriction controls, using these controls may not work as expected. Third party software may be required to fulfill the compliance requirements.

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

```
/usr/bin/osascript -l JavaScript << EOS
function run() {
Ê let pref1 = ObjC.unwrap(
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess.new')\
Ê .objectForKey('familyControlsEnabled'))
Ê let pathlist =
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess.new')\
Ê .objectForKey('pathBlackList').js
Ê for ( let app in pathlist ) {
Ê     if ( ObjC.unwrap(pathlist[app]) == "/Applications/FaceTime.app" && pref1 == true
){
Ê         return("true")
Ê     }
Ê }
Ê return("false")
Ê }
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/>
<key>pathBlackList</key>
<array>
Ê   <string>/Applications/FaceTime.app</string>
</array>
```

ID	os_facetime_app_disable

References	800-53r5	¥ AC-20 ¥ CM-7, CM-7(1)
	DISA STIG(s)	¥ APPL-15-002010
	CCE	¥ CCE-94189-8

9.15. 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)
	DISA STIG(s)	¥ APPL-15-000033
	CCE	¥ CCE-94192-2

9.16. 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
	DISA STIG(s)	¥ APPL-15-003013
	CCE	¥ CCE-94194-8

9.17. Enable Gatekeeper

Gatekeeper *MUST* be enabled.

Gatekeeper is a security feature that ensures that applications are digitally signed by an Apple-issued certificate before they are permitted to run. Digital signatures allow the macOS host to verify

that the application has not been modified by a malicious third party.

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

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

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

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

Remediation Description

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

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

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

ID	os_gatekeeper_enable	
References	800-53r5	¥ CM-14
		¥ CM-5
		¥ SI-3
		¥ SI-7(1), SI-7(15)
	DISA STIG(s)	¥ APPL-15-002064
	CCE	¥ CCE-94195-5

9.18. Disable Genmoji AI Creation

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

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

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

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

Remediation Description

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

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

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

ID	os_genmoji_disable	
References	800-53r5	¥ AC-20, AC-20(1) ¥ CM-7, CM-7(1) ¥ SC-7(10)
	DISA STIG(s)	¥ APPL-15-005140
	CCE	¥ CCE-94196-3

9.19. 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)
	DISA STIG(s)	¥ APPL-15-005058
	CCE	¥ CCE-94199-7

9.20. Secure Userȏs Home Folders

The system *MUST* be configured to prevent access to other userȏs home folders.

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

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

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

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

Remediation Description

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

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

ID	os_home_folders_secure

References	800-53r5	¥ AC-6
	DISA STIG(s)	¥ APPL-15-002068
	CCE	¥ CCE-94204-5

9.21. 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" => disabled'
```

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

Remediation Description

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

```
/bin/launchctl disable system/org.apache.httpd
```

ID	os_httpd_disable	
References	800-53r5	¥ AC-17
		¥ AC-3
	DISA STIG(s)	¥ APPL-15-002008
	CCE	¥ CCE-94205-2

9.22. 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
	DISA STIG(s)	¥ APPL-15-002037
	CCE	¥ CCE-94206-0

9.23. Disable AI Image Generation

Apple Intelligence features that use off device AI *MUST* be disabled.

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

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

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

Remediation Description

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

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

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

```
<false/>
```

ID	os_image_generation_disable	
References	800-53r5	¥ AC-20, AC-20(1) ¥ CM-7, CM-7(1) ¥ SC-7(10)
	DISA STIG(s)	¥ APPL-15-005150
	CCE	¥ CCE-94208-6

9.24. Configure Install.log Retention to 300

The install.log *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.

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

```
/usr/sbin/aslmanager -dd 2>&1 | /usr/bin/awk '/\/var\/log\/install.log$/ {count++}
/Processing module com.apple.install/,/Finished/ { for (i=1;i<=NR;i++) { if ($i ==
"TTL" && $(i+2) >= 300) { ttl="True" }; if ($i == "MAX") {max="True"}}} END{if (count
> 1) { print "Multiple config files for /var/log/install, manually remove the extra
files"} else if (max == "True") { print "all_max setting is configured, must be
removed" } if (ttl != "True") { print "TTL not configured" } else { print "Yes" }}'
```

If the result is not Yes, this is a finding.

Remediation Description

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

```
/usr/bin/sed -i '' "s/\* file \/var\/log\/install.log.*/\* file \/var\/log
\/install.log format='\$\(\(Time\)\(JZ\)\) \$Host \$\(Sender\)\[\$\(PID\)\]:
\$Message' rotate=utc compress file_max=50M size_only ttl=300/g"
/etc/asl/com.apple.install
```

! If there are multiple configuration files in /etc/asl that are set to process the file /var/log/install.log, these files will have to be manually removed.

ID	os_install_log_retention_configure

References	800-53r5	¥ AU-11 ¥ AU-4
	DISA STIG(s)	¥ APPL-15-004050
	CCE	¥ CCE-94212-8

9.25. Disable iPhone Mirroring

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

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

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

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

Remediation Description

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

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

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

ID	os_iphone_mirroring_disable	
References	800-53r5	¥ AC-20 ¥ AC-3 ¥ CM-7, CM-7(1)
	DISA STIG(s)	¥ APPL-15-002271
	CCE	¥ CCE-94213-6

9.26. Prevent AdminHostInfo from Being Available at LoginWindow

The system *MUST* be configured to not display sensitive information at the LoginWindow. The key AdminHostInfo when configured will allow the HostName, IP Address, and operating system version and build to be displayed.

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

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

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

Remediation Description

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

This is implemented by a Configuration Profile.

ID	os_loginwindow_adminhostinfo_undefined	
References	800-53r5	¥ AC-11(1)
	DISA STIG(s)	¥ APPL-15-000009
	CCE	¥ CCE-94221-9

9.27. 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
	DISA STIG(s)	- APPL-15-005110
	CCE	- CCE-94227-6

9.28. 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
	DISA STIG(s)	¥ APPL-15-004030
	CCE	¥ CCE-94233-4

9.29. 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
	DISA STIG(s)	¥ APPL-15-004040
	CCE	¥ CCE-94234-2

9.30. Disable Network File System Service

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

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

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

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

Remediation Description

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

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

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

ID	os_nfsd_disable	
References	800-53r5	¥ AC-17
		¥ AC-3
	DISA STIG(s)	¥ APPL-15-002003
	CCE	¥ CCE-94235-9

9.31. Enforce On Device Dictation

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

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

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

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

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

Remediation Description

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

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

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

ID	os_on_device_dictation_enforce	
References	800-53r5	¥ AC-20
		¥ CM-7, CM-7(1)
		¥ SC-7(10)
	DISA STIG(s)	¥ APPL-15-002220
	CCE	¥ CCE-94245-8

9.32. Remove Password Hint From User Accounts

User accounts *MUST* not contain password hints.

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

```
HINT=$(/usr/bin/dscl . -list /Users hint | /usr/bin/awk '{ print $2 }')

if [ -z "$HINT" ]; then
Ê echo "PASS"
else
Ê echo "FAIL"
fi
```

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

Remediation Description

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

```
for u in $(/usr/bin/dscl . -list /Users UniqueID | /usr/bin/awk '$2 > 500 {print $1}'); do
Ê /usr/bin/dscl . -delete /Users/$u hint
done
```

ID	os_password_hint_remove	
References	800-53r5	¥ IA-6
	DISA STIG(s)	¥ APPL-15-003014
	CCE	¥ CCE-94248-2

9.33. 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
	DISA STIG(s)	¥ APPL-15-005060
	CCE	¥ CCE-94249-0

9.34. 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.

The banner text of the document *MUST* read:

```
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.
```

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 (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/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
	DISA STIG(s)	¥ APPL-15-000025
	CCE	¥ CCE-94254-0

9.35. 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
	DISA STIG(s)	¥ APPL-15-000023
	CCE	¥ CCE-94255-7

9.36. 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 -G | /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
Ê /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 ${include_dir}); do
Ê if [[ "$file" == "100-macos.conf" ]]; then
Ê     continue
Ê fi
Ê if [[ "$file" == "01-mscp-sshd.conf" ]]; then
Ê     break
Ê fi
Ê /bin/mv ${include_dir}${file} ${include_dir}20-${file}
done
```

ID	os_policy_banner_ssh_enforce	
References	800-53r5	¥ AC-8
	DISA STIG(s)	¥ APPL-15-000024
	CCE	¥ CCE-94256-5

9.37. Disable Privacy Setup Services During Setup Assistant

The prompt for Privacy Setup services during Setup Assistant *MUST* be disabled.

Organizations *MUST* apply organization-wide configuration settings. The macOS Privacy Setup services prompt guides new users through enabling their own specific privacy settings; this is not essential and, therefore, *MUST* be disabled to prevent against the risk of individuals electing privacy 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('SkipPrivacySetup').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>SkipPrivacySetup</key>
<true/>
```

ID	os_privacy_setup_prompt_disable	
References	800-53r5	¥ CM-7, CM-7(1)
	DISA STIG(s)	¥ APPL-15-002036
	CCE	¥ CCE-94264-9

9.38. 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

and must be from your MDM.

ID	os_recovery_lock_enable	
References	800-53r5	¥ AC-6
	DISA STIG(s)	¥ APPL-15-005120
	CCE	¥ CCE-94274-8

9.39. 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)
	DISA STIG(s)	¥ APPL-15-000100
	CCE	¥ CCE-94279-7

9.40. 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)
	DISA STIG(s)	¥ APPL-15-005100
	CCE	¥ CCE-94288-8

9.41. 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
	DISA STIG(s)	¥ APPL-15-005001
	CCE	¥ CCE-94294-6

9.42. 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)
	DISA STIG(s)	¥ APPL-15-002039
	CCE	¥ CCE-94295-3

9.43. Disable Screen Time Prompt During Setup Assistant

The prompt for Screen Time setup during Setup Assistant *MUST* be disabled.

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

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.SetupAssistant.managed')\
.objectForKey('SkipScreenTime').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>SkipScreenTime</key>
<true/>
```

ID	os_skip_screen_time_prompt_enable	
References	800-53r5	¥ CM-7, CM-7(1)
	DISA STIG(s)	¥ APPL-15-005055
	CCE	¥ CCE-94296-1

9.44. 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
	DISA STIG(s)	¥ APPL-15-005056
	CCE	¥ CCE-94297-9

9.45. 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 140-3 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.

! On macOS 15.2 and higher the SSH configuration can be reset to the macOS default by running /usr/libexec/reset-ssh-configuration.

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

```
fips_ssh_config=("Ciphers aes128-gcm@openssh.com" "HostbasedAcceptedAlgorithms ecdsa-
sha2-nistp256,ecdsa-sha2-nistp256-cert-v01@openssh.com" "HostKeyAlgorithms ecdsa-sha2-
nistp256-cert-v01@openssh.com,sk-ecdsa-sha2-nistp256-cert-v01@openssh.com,ecdsa-sha2-
nistp256,sk-ecdsa-sha2-nistp256@openssh.com" "KexAlgorithms ecdh-sha2-nistp256" "MACs
hmac-sha2-256-etm@openssh.com,hmac-sha2-256" "PubkeyAcceptedAlgorithms ecdsa-sha2-
nistp256,ecdsa-sha2-nistp256-cert-v01@openssh.com,sk-ecdsa-sha2-nistp256-cert-
v01@openssh.com" "CASignatureAlgorithms ecdsa-sha2-nistp256,sk-ecdsa-sha2-
nistp256@openssh.com")
total=0
ret="pass"
for config in $fips_ssh_config; do
Ê if [[ "$ret" == "fail" ]]; then
Ê   break
Ê fi
Ê for u in $(/usr/bin/dscl . list /users shell | /usr/bin/egrep -v
'(^_)|(root)|(/usr/bin/false)' | /usr/bin/awk '{print $1}'); do
Ê   sshCheck=$(/usr/bin/sudo -u $u /usr/bin/ssh -G . | /usr/bin/grep -ci "$config")
Ê   if [[ "$sshCheck" == "0" ]]; then
Ê     ret="fail"
Ê     break
Ê   fi
Ê done
done
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:

```
if [ -f /etc/ssh/crypto.conf ] && /usr/bin/grep -q "Include /etc/ssh/crypto.conf"
/etc/ssh/ssh_config.d/100-macos.conf 2>/dev/null; then
Ê /bin/ln -fs /etc/ssh/crypto/fips.conf /etc/ssh/crypto.conf
fi
include_dir=$(/usr/bin/awk '/^Include/ {print $2}' /etc/ssh/ssh_config |
/usr/bin/tr -d '*')
```

```
fips_ssh_config=("Ciphers aes128-gcm@openssh.com" "HostbasedAcceptedAlgorithms
ecdsa-sha2-nistp256,ecdsa-sha2-nistp256-cert-v01@openssh.com" "HostKeyAlgorithms
ecdsa-sha2-nistp256-cert-v01@openssh.com,sk-ecdsa-sha2-nistp256-cert-
v01@openssh.com,ecdsa-sha2-nistp256,sk-ecdsa-sha2-nistp256@openssh.com"
"KexAlgorithms ecdh-sha2-nistp256" "MACs hmac-sha2-256-etm@openssh.com,hmac-sha2-
256" "PubkeyAcceptedAlgorithms ecdsa-sha2-nistp256,ecdsa-sha2-nistp256-cert-
v01@openssh.com,sk-ecdsa-sha2-nistp256-cert-v01@openssh.com"
"CASignatureAlgorithms ecdsa-sha2-nistp256,sk-ecdsa-sha2-nistp256@openssh.com")
for ssh_config in $fips_ssh_config; do
Ê ssh_setting=$(echo $ssh_config | /usr/bin/cut -d " " -f1)
Ê /usr/bin/grep -qEi "^$ssh_setting" "${include_dir}01-mscp-ssh.conf" &&
/usr/bin/sed -i "" "s/^$ssh_setting.*/${ssh_config}/" "${include_dir}01-mscp-
ssh.conf" || echo "$ssh_config" >> "${include_dir}01-mscp-ssh.conf"
Ê for u in $(/usr/bin/dscl . list /users shell | /usr/bin/egrep -v
'(^_)|(root)|(/usr/bin/false)' | /usr/bin/awk '{print $1}'); do
Ê   config=$(/usr/bin/sudo -u $u /usr/bin/ssh -Gv . 2>&1)
Ê   configfiles=$(echo "$config" | /usr/bin/awk '/Reading configuration data/
{print $NF}'| /usr/bin/tr -d '\r')
Ê   configarray=( ${(f)configfiles} )
Ê   if ! echo $config | /usr/bin/grep -q -i "$ssh_config" ; then
Ê     for c in $configarray; do
Ê       if [[ "$c" == "/etc/ssh/crypto.conf" ]]; then
Ê         continue
Ê       fi

Ê       /usr/bin/sudo -u $u /usr/bin/grep -qEi "^$ssh_setting" "$c" &&
/usr/bin/sed -i "" "s/^$ssh_setting.*/${ssh_config}/I" "$c"
Ê       if [[ "$c" =~ ".ssh/config" ]]; then
Ê         if /usr/bin/grep -qEi "$ssh_setting" "$c" 2> /dev/null; then
Ê           old_file=$(cat ~$u/.ssh/config)
Ê           echo "$ssh_config" > ~$u/.ssh/config
Ê           echo "$old_file" >> ~$u/.ssh/config
Ê         fi
Ê       fi
Ê     done
Ê   fi
Ê done
done
```

ID	os_ssh_fips_compliant	
References	800-53r5	¥ AC-17(2)
		¥ IA-7
		¥ SC-13
		¥ SC-8(1)
	DISA STIG(s)	¥ APPL-15-000057
	CCE	¥ CCE-94299-5

9.46. 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.

! On macOS 15.2 and higher the SSH configuration can be reset to the macOS default by running /usr/libexec/reset-ssh-configuration.

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 $1}'
); do
Ê sshCheck=$(/usr/bin/sudo -u $u /usr/bin/ssh -G . | /usr/bin/grep -c
"^serveralivecountmax 0")
Ê if [[ "$sshCheck" == "0" ]]; then
Ê   ret="fail"
Ê   break
Ê 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:

```
include_dir=$(/usr/bin/awk '/^Include/ {print $2}' /etc/ssh/ssh_config |
/usr/bin/tr -d '*')

ssh_config=("ServerAliveCountMax 0")

ssh_setting=$(echo $ssh_config | /usr/bin/cut -d " " -f1)
/usr/bin/grep -qEi "^$ssh_setting" "${include_dir}01-mscp-ssh.conf" &&
/usr/bin/sed -i "" "s/^$ssh_setting.*/${ssh_config}/" "${include_dir}01-mscp-
ssh.conf" || echo "$ssh_config" >> "${include_dir}01-mscp-ssh.conf"
for u in $(/usr/bin/dscl . list /users shell | /usr/bin/egrep -v
'(^_)|(root)|(/usr/bin/false)' | /usr/bin/awk '{print $1}'); do
Ê config=$(/usr/bin/sudo -u $u /usr/bin/ssh -Gv . 2>&1)
Ê configfiles=$(echo "$config" | /usr/bin/awk '/Reading configuration data/ {print
```

```
$NF}'| /usr/bin/tr -d '\r')
Ê configarray=( ${(f)configfiles} )
Ê if ! echo $config | /usr/bin/grep -q -i "$ssh_config" ; then
Ê   for c in $configarray; do
Ê     if [[ "$c" == "/etc/ssh/crypto.conf" ]]; then
Ê       continue
Ê     fi

Ê     /usr/bin/sudo -u $u /usr/bin/grep -qEi "^$ssh_setting" "$c" && /usr/bin/sed
-i "" "s/^$ssh_setting.*/${ssh_config}/I" "$c"
Ê     if [[ "$c" =~ ".ssh/config" ]]; then
Ê       if /usr/bin/grep -qEi "$ssh_setting" "$c" 2> /dev/null; then
Ê         old_file=$(cat ~$u/.ssh/config)
Ê         echo "$ssh_config" > ~$u/.ssh/config
Ê         echo "$old_file" >> ~$u/.ssh/config
Ê       fi
Ê     fi
Ê   done
Ê fi
done
```

ID	os_ssh_server_alive_count_max_configure	
References	800-53r5	¥ SC-10
	DISA STIG(s)	¥ APPL-15-000140
	CCE	¥ CCE-94300-1

9.47. 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.

! On macOS 15.2 and higher the SSH configuration can be reset to the macOS default by running /usr/libexec/reset-ssh-configuration.

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
```

```
Ê sshCheck=$(/usr/bin/sudo -u $u /usr/bin/ssh -G . | /usr/bin/grep -c
"^serveraliveinterval 900")
Ê if [[ "$sshCheck" == "0" ]]; then
Ê   ret="fail"
Ê   break
Ê 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:

```
include_dir=$(/usr/bin/awk '/^Include/ {print $2}' /etc/ssh/ssh_config |
/usr/bin/tr -d '*')

ssh_config_string=("ServerAliveInterval 900")
for ssh_config in $ssh_config_string; do
Ê ssh_setting=$(echo $ssh_config | /usr/bin/cut -d " " -f1)
Ê /usr/bin/grep -qEi "^$ssh_setting" "${include_dir}01-mscp-ssh.conf" &&
/usr/bin/sed -i "" "s/^$ssh_setting.*/${ssh_config}/" "${include_dir}01-mscp-
ssh.conf" || echo "$ssh_config" >> "${include_dir}01-mscp-ssh.conf"
Ê for u in $(/usr/bin/dscl . list /users shell | /usr/bin/egrep -v
'(^_)|(root)|(/usr/bin/false)' | /usr/bin/awk '{print $1}'); do
Ê   config=$(/usr/bin/sudo -u $u /usr/bin/ssh -Gv . 2>&1)
Ê   configfiles=$(echo "$config" | /usr/bin/awk '/Reading configuration data/
{print $NF}'| /usr/bin/tr -d '\r')
Ê   configarray=( ${(f)configfiles} )
Ê   if ! echo $config | /usr/bin/grep -q -i "$ssh_config" ; then
Ê     for c in $configarray; do
Ê       if [[ "$c" == "/etc/ssh/crypto.conf" ]]; then
Ê         continue
Ê       fi

Ê       /usr/bin/sudo -u $u /usr/bin/grep -qEi "^$ssh_setting" "$c" &&
/usr/bin/sed -i "" "s/^$ssh_setting.*/${ssh_config}/I" "$c"
Ê       if [[ "$c" =~ ".ssh/config" ]]; then
Ê         if /usr/bin/grep -qEi "$ssh_setting" "$c" 2> /dev/null; then
Ê           old_file=$(cat ~$u/.ssh/config)
Ê           echo "$ssh_config" > ~$u/.ssh/config
Ê           echo "$old_file" >> ~$u/.ssh/config
Ê         fi
Ê       fi
Ê     done
Ê   fi
Ê done
```

```
done
```

ID	os_ssh_server_alive_interval_configure	
References	800-53r5	¥ AC-12 ¥ SC-10
	DISA STIG(s)	¥ APPL-15-000110
	CCE	¥ CCE-94301-9

9.48. Configure SSHD Channel Timeout to session:*=900

If SSHD is enabled it *MUST* be configured with session ChannelTime out set to session:*=900.

This will set the time out when the session is inactive.

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

! On macOS 15.2 and higher the SSH configuration can be reset to the macOS default by running /usr/libexec/reset-ssh-configuration.

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

```
/usr/sbin/sshd -G | /usr/bin/awk '/channeltimeout/{print $2}'
```

If the result is not session:=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
Ê /usr/bin/sed -i.bk "1s/.*/Include \/etc\/ssh\/sshd_config.d\/\*/"
/etc/ssh/sshd_config
fi

/usr/bin/grep -qxF 'channeltimeout session:*=900' "${include_dir}01-mscp-
sshd.conf" 2>/dev/null || echo "channeltimeout session:*=900" >> "${include_dir
}01-mscp-sshd.conf"
```

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

ID	os_sshd_channel_timeout_configure	
References	800-53r5	¥ AC-12
		¥ SC-10
	DISA STIG(s)	¥ APPL-15-000120
	CCE	¥ CCE-94302-7

9.49. Configure SSHD ClientAliveCountMax to 1

If SSHD is enabled it *MUST* be configured with the Client Alive Maximum Count set to 1.

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.

! On macOS 15.2 and higher the SSH configuration can be reset to the macOS default by running /usr/libexec/reset-ssh-configuration.

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

```
/usr/sbin/sshd -G | /usr/bin/awk '/clientalivecountmax/{print $2}'
```

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
Ê /usr/bin/sed -i.bk "1s/.*/Include \/etc\/ssh\/sshd_config.d\/\*/"
/etc/ssh/sshd_config
fi

/usr/bin/grep -qxF 'clientalivecountmax 1' "${include_dir}01-mscp-sshd.conf"
2>/dev/null || echo "clientalivecountmax 1" >> "${include_dir}01-mscp-sshd.conf"

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

ID	os_sshd_client_alive_count_max_configure	
References	800-53r5	¥ SC-10
	DISA STIG(s)	¥ APPL-15-000052
	CCE	¥ CCE-94303-5

9.50. 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 conjunction 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.

! On macOS 15.2 and higher the SSH configuration can be reset to the macOS default by running /usr/libexec/reset-ssh-configuration.

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

```
/usr/sbin/sshd -G | /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
Ê /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 ${include_dir}); do
Ê if [[ "$file" == "100-macos.conf" ]]; then
Ê     continue
Ê fi
Ê if [[ "$file" == "01-mscp-sshd.conf" ]]; then
Ê     break
Ê fi
Ê /bin/mv ${include_dir}${file} ${include_dir}20-${file}
done
```

ID	os_sshd_client_alive_interval_configure	
References	800-53r5	¥ AC-12
		¥ SC-10
	DISA STIG(s)	¥ APPL-15-000051
	CCE	¥ CCE-94304-3

9.51. 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 140-3 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.

! On macOS 15.2 and higher the SSH configuration can be reset to the macOS default by running /usr/libexec/reset-ssh-configuration.

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-cert-v01@openssh.com,sk-ecdsa-sha2-nistp256-cert-v01@openssh.com,ecdsa-sha2-
nistp256,sk-ecdsa-sha2-nistp256@openssh.com" "KexAlgorithms ecdh-sha2-nistp256" "MACs
hmac-sha2-256-etm@openssh.com,hmac-sha2-256" "PubkeyAcceptedAlgorithms ecdsa-sha2-
nistp256,ecdsa-sha2-nistp256-cert-v01@openssh.com,sk-ecdsa-sha2-nistp256-cert-
v01@openssh.com" "CASignatureAlgorithms ecdsa-sha2-nistp256,sk-ecdsa-sha2-
nistp256@openssh.com")
total=0
for config in $fips_sshd_config; do
Ê total=$(expr $(/usr/sbin/sshd -G | /usr/bin/grep -i -c "$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:

```
if [ -f /etc/ssh/crypto.conf ] && /usr/bin/grep -q "Include /etc/ssh/crypto.conf"
/etc/ssh/sshd_config.d/100-macos.conf 2>/bin/null; then
Ê /bin/ln -fs /etc/ssh/crypto/fips.conf /etc/ssh/crypto.conf
fi

include_dir=$(/usr/bin/awk '/^Include/ {print $2}' /etc/ssh/sshd_config |
/usr/bin/tr -d '*')

if [[ -z $include_dir ]]; then
Ê /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-cert-v01@openssh.com,sk-ecdsa-sha2-nistp256-cert-
v01@openssh.com,ecdsa-sha2-nistp256,sk-ecdsa-sha2-nistp256@openssh.com"
"KexAlgorithms ecdh-sha2-nistp256" "MACs hmac-sha2-256-etm@openssh.com,hmac-sha2-
256" "PubkeyAcceptedAlgorithms ecdsa-sha2-nistp256,ecdsa-sha2-nistp256-cert-
v01@openssh.com,sk-ecdsa-sha2-nistp256-cert-v01@openssh.com"
"CASignatureAlgorithms ecdsa-sha2-nistp256,sk-ecdsa-sha2-nistp256@openssh.com")
sshd_config=$(/usr/sbin/sshd -G)
for config in $fips_sshd_config; do
Ê if ! echo $sshd_config | /usr/bin/grep -q -i "$config" 2>/dev/null; then
Ê   /usr/bin/grep -qxF "$config" "${include_dir}01-mscp-sshd.conf" 2>/dev/null ||
echo "$config" >> "${include_dir}01-mscp-sshd.conf"
Ê fi
done

for file in $(ls ${include_dir}); do
Ê if [[ "$file" == "100-macos.conf" ]]; then
Ê     continue
Ê fi
Ê if [[ "$file" == "01-mscp-sshd.conf" ]]; then
Ê     break
Ê fi
Ê /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)
	DISA STIG(s)	¥ APPL-15-000054
	CCE	¥ CCE-94305-0

9.52. Set Login Grace Time to 30

If SSHD is enabled then it *MUST* be configured to wait only 30 seconds before timing out logon attempts.

> **!** On macOS 15.2 and higher the SSH configuration can be reset to the macOS default by running /usr/libexec/reset-ssh-configuration.

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

```
/usr/sbin/sshd -G | /usr/bin/awk '/logingracetime/{print $2}'
```

If the result is not 30, 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
Ê /usr/bin/sed -i.bk "1s/.*/Include \/etc\/ssh\/sshd_config.d\/\*/"
/etc/ssh/sshd_config
fi

/usr/bin/grep -qxF 'logingracetime 30' "${include_dir}01-mscp-sshd.conf"
2>/dev/null || echo "logingracetime 30" >> "${include_dir}01-mscp-sshd.conf"

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

ID	os_sshd_login_grace_time_configure	
References	800-53r5	¥ SC-10
	DISA STIG(s)	¥ APPL-15-000053
	CCE	¥ CCE-94306-8

9.53. Disable Root Login for SSH

If SSH is enabled to assure individual accountability and prevent unauthorized access, logging in as root via SSH *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.

> **!** On macOS 15.2 and higher the SSH configuration can be reset to the macOS default by running /usr/libexec/reset-ssh-configuration.

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

```
/usr/sbin/sshd -G | /usr/bin/awk '/permitrootlogin/{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:

```
include_dir=$(/usr/bin/awk '/^Include/ {print $2}' /etc/ssh/sshd_config |
/usr/bin/tr -d '*')

if [[ -z $include_dir ]]; then
Ê /usr/bin/sed -i.bk "1s/.*/Include \/etc\/ssh\/sshd_config.d\/\*/"
/etc/ssh/sshd_config
fi

/usr/bin/grep -qxF 'permitrootlogin no' "${include_dir}01-mscp-sshd.conf"
2>/dev/null || echo "permitrootlogin no" >> "${include_dir}01-mscp-sshd.conf"

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

ID	os_sshd_permit_root_login_configure	
References	800-53r5	¥ IA-2(5)
	DISA STIG(s)	¥ APPL-15-001100
	CCE	¥ CCE-94307-6

9.54. Configure SSHD Unused Connection Timeout to 900

If SSHD is enabled it *MUST* be configured with unused connection timeout set to 900.

This will set the time out when there are no open channels within an session.

> **!** On macOS 15.2 and higher the SSH configuration can be reset to the macOS default by running /usr/libexec/reset-ssh-configuration.

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

```
/usr/sbin/sshd -G | /usr/bin/awk '/unusedconnectiontimeout/{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
Ê /usr/bin/sed -i.bk "1s/.*/Include \/etc\/ssh\/sshd_config.d\/\*/"
/etc/ssh/sshd_config
fi

/usr/bin/grep -qxF 'unusedconnectiontimeout 900' "${include_dir}01-mscp-sshd.conf"
2>/dev/null || echo "unusedconnectiontimeout 900" >> "${include_dir}01-mscp-
sshd.conf"

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

ID	os_sshd_unused_connection_timeout_configure	
References	800-53r5	¥ AC-12
		¥ SC-10
	DISA STIG(s)	¥ APPL-15-000130
	CCE	¥ CCE-94308-4

9.55. Configure Sudo To Log Events

Sudo *MUST* be configured to log privilege escalation.

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

```
/usr/bin/sudo /usr/bin/sudo -V | /usr/bin/grep -c "Log when a command is allowed by sudoers"
```

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 ''
'/^Defaults[[:blank:]]*\!log_allowed/s/^/# /' '{}' \;
/bin/echo "Defaults log_allowed" >> /etc/sudoers.d/mscp
```

ID	os_sudo_log_enforce	
References	800-53r5	¥ AC-6(9)
	DISA STIG(s)	¥ APPL-15-000190
	CCE	¥ CCE-94310-0

9.56. Configure Sudo Timeout Period to 0

The file /etc/sudoers *MUST* include a timestamp_timeout 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
	DISA STIG(s)	¥ APPL-15-004022
	CCE	¥ CCE-94311-8

9.57. 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
	DISA STIG(s)	¥ APPL-15-004060
	CCE	¥ CCE-94312-6

9.58. 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" =>
disabled'
```

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

Remediation Description

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

```
/bin/launchctl disable system/com.apple.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)
	DISA STIG(s)	¥ APPL-15-002038
	CCE	¥ CCE-94317-5

9.59. 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
```

! The service timed cannot be unloaded or loaded while System Integrity Protection (SIP) is enabled.

ID	os_time_server_enabled	
References	800-53r5	¥ AU-12(1) ¥ SC-45(1)
	DISA STIG(s)	¥ APPL-15-000180
	CCE	¥ CCE-94319-1

9.60. 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
	DISA STIG(s)	¥ APPL-15-005054
	CCE	¥ CCE-94320-9

9.61. Disable Login to Other Userôs Active and Locked Sessions

The ability to log in to another userôs 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 userôs sessions. Disabling the admins and/or userôs ability to log into another userôs active and locked session prevents unauthorized persons from viewing potentially sensitive and/or personal information.

! Configuring this setting will change the user experience and disable TouchID from unlocking the screensaver. To restore the user experience and allow TouchID to unlock the screensaver, you can run `/usr/bin/sudo /usr/bin/defaults write /Library/Preferences/com.apple.loginwindow screenUnlockMode -int 1`. This setting can also be deployed with a configuration profile.

$ This rule may cause issues when platformSSO is configured.

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)
	DISA STIG(s)	¥ APPL-15-000090
	CCE	¥ CCE-94322-5

9.62. Prohibit User Installation of Software into /Users/

Users *MUST* not be allowed to install software into /Users/.

Allowing regular users to install software, without explicit privileges, presents the risk of untested and potentially malicious software being installed on the system. Explicit privileges (escalated or administrative privileges) provide the regular user with explicit capabilities and control that exceeds the rights of a regular user.

> Apple has deprecated the use of application restriction controls, using these controls may not work as expected. Third party software may be required to fulfill the compliance requirements.

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

```
/usr/bin/osascript -l JavaScript << EOS
function run() {
Ê let pref1 = ObjC.unwrap(
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess.new')\
Ê .objectForKey('familyControlsEnabled'))
Ê let pathlist =
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess.new')\
Ê .objectForKey('pathBlackList').js
Ê for ( let app in pathlist ) {
Ê     if ( ObjC.unwrap(pathlist[app]) == "/Users/" && pref1 == true ){
Ê         return("true")
Ê     }
Ê }
Ê return("false")
Ê }
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/>
<key>pathBlackList</key>
<array>
Ê   <string>/Users/</string>
</array>
```

ID	os_user_app_installation_prohibit	
References	800-53r5	¥ CM-11(2)
	DISA STIG(s)	¥ APPL-15-005080
	CCE	¥ CCE-94323-3

9.63. 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" => disabled'
```

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

Remediation Description

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

```
/bin/launchctl disable system/com.apple.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
	DISA STIG(s)	¥ APPL-15-002006
	CCE	¥ CCE-94324-1

9.64. Disable Apple Intelligence Writing Tools

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

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

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

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

Remediation Description

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

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

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

ID	os_writing_tools_disable	
References	800-53r5	¥ AC-20, AC-20(1) ¥ CM-7, CM-7(1) ¥ SC-7(10)
	DISA STIG(s)	¥ APPL-15-005160
	CCE	¥ CCE-94328-2

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)
	DISA STIG(s)	¥ APPL-15-003080
	CCE	¥ CCE-94330-8

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"}}' | /usr/bin/uniq
```

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
	DISA STIG(s)	¥ APPL-15-000022
	CCE	¥ CCE-94331-6

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"}}' | /usr/bin/uniq
```

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
	DISA STIG(s)	¥ APPL-15-000060
	CCE	¥ CCE-94332-4

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.

> **!** To comply with Executive Order 14028, ÒImproving the NationÔs CybersecurityÓ, OMB M-22-09, ÒMoving the U.S. Government Toward Zero Trust Cybersecurity PrinciplesÓ, and NIST SP-800-63b, ÒDigital Identity Guidelines: Authentication and Lifecycle ManagementÓ federal, military, and intelligence communities must adopt the following configuration settings. Password policies must not require the use of complexity policies such as upper characters, lower characters, or special characters. Password policies must also not require the use of regular rotation. Password policies should define a minimum length. Multifactor authentication should be used where ever possible.

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)
	DISA STIG(s)	¥ APPL-15-003007
	CCE	¥ CCE-94333-2

10.5. Require Passwords to Match the Defined Custom Regular Expression

The macOS *MUST* be configured to meet complexity requirements defined in ^(?=.[A-Z])(?=.[a-z])(?=.[0-9]).$.

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

! To comply with Executive Order 14028, ÒImproving the NationÔs CybersecurityÓ, OMB M-22-09, ÒMoving the U.S. Government Toward Zero Trust Cybersecurity PrinciplesÓ, and NIST SP-800-63b, ÒDigital Identity Guidelines: Authentication and Lifecycle ManagementÓ federal, military, and intelligence communities must adopt the following configuration settings. Password policies must not require the use of complexity policies such as upper characters, lower characters, or special characters. Password policies must also not require the use of regular rotation. Password policies should define a minimum length. Multifactor authentication should be used where ever possible.

! The configuration profile generated must be installed from an MDM server.

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-Z])(?=.*[a-z])(?=.*[0-9]).*$'\''")])' -
```

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>customRegex</key>
<dict>
Ê   <key>passwordContentRegex</key>
Ê   <string>^(?=.*[A-Z])(?=.*[a-z])(?=.*[0-9]).*$</string>
```

```
Ê   <key>passwordContentDescription</key>
Ê   <dict>
Ê     <key>default</key>
Ê     <string>Password must match custom regex.</string>
Ê   </dict>
</dict>
```

ID	pwpolicy_custom_regex_enforce	
References	800-53r5	¥ IA-5(1)
	DISA STIG(s)	¥ APPL-15-003060
	CCE	¥ CCE-94334-0

10.6. 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"}}' | /usr/bin/uniq
```

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)
	DISA STIG(s)	¥ N/A
	CCE	¥ CCE-94337-3

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.

> **!** To comply with Executive Order 14028, ÒImproving the NationÕs CybersecurityÓ, OMB M-22-09, ÒMoving the U.S. Government Toward Zero Trust Cybersecurity PrinciplesÓ, and NIST SP-800-63b, ÒDigital Identity Guidelines: Authentication and Lifecycle ManagementÓ federal, military, and intelligence communities must adopt the following configuration settings. Password policies must not require the use of complexity policies such as upper characters, lower characters, or special characters. Password policies must also not require the use of regular rotation. Password policies should define a minimum length. Multifactor authentication should be used where ever possible.

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
	DISA STIG(s)	¥ APPL-15-003008
	CCE	¥ CCE-94339-9

10.8. Require a Minimum Password Length of 14 Characters

The macOS *MUST* be configured to require a minimum of 14 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.

> **!** To comply with Executive Order 14028, ÒImproving the NationÕs CybersecurityÓ, OMB M-22-09, ÒMoving the U.S. Government Toward Zero Trust CybersecurityÓ Principles Ó, and NIST SP-800-63b, ÒDigital Identity Guidelines: Authentication and Lifecycle ManagementÓ federal, military, and intelligence communities must adopt the following configuration settings. Password policies must not require the use of complexity policies such as upper characters, lower characters, or special characters. Password policies must also not require the use of regular rotation. Password policies should define a minimum length. Multifactor authentication should be used where ever possible.

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
'\''.{14,}'\''")])' -
```

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>14</integer>
```

ID	pwpolicy_minimum_length_enforce	
References	800-53r5	¥ IA-5(1)
	DISA STIG(s)	¥ APPL-15-003010
	CCE	¥ CCE-94340-7

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.

> **!** To comply with Executive Order 14028, ÒImproving the NationÔs CybersecurityÓ, OMB M-22-09, ÒMoving the U.S. Government Toward Zero Trust Cybersecurity PrinciplesÓ, and NIST SP-800-63b, ÒDigital Identity Guidelines: Authentication and Lifecycle ManagementÓ federal, military, and intelligence communities must adopt the following configuration settings. Password policies must not require the use of complexity policies such as upper characters, lower characters, or special characters. Password policies must also not require the use of regular rotation. Password policies should define a minimum length. Multifactor authentication should be used where ever possible.

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
	DISA STIG(s)	¥ APPL-15-003070
	CCE	¥ CCE-94341-5

10.10. 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.

! To comply with Executive Order 14028, ÒImproving the NationÕs CybersecurityÓ, OMB M-22-09, ÒMoving the U.S. Government Toward Zero Trust Cybersecurity PrinciplesÓ, and NIST SP-800-63b, ÒDigital Identity Guidelines: Authentication and Lifecycle ManagementÓ federal, military, and intelligence communities must adopt the following configuration settings. Password policies must not require the use of

complexity policies such as upper characters, lower characters, or special characters. Password policies must also not require the use of regular rotation. Password policies should define a minimum length. Multifactor authentication should be used where ever possible.

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

```
/usr/bin/pwpolicy -getaccountpolicies 2>/dev/null | /usr/bin/tail -n +2 |
/usr/bin/xmllint --xpath "//string[contains(text(), \"policyAttributePassword matches
'(.*[^a-zA-Z0-9].*){\")]" - 2>/dev/null | /usr/bin/awk -F"{|}" '{if ($2 >= 1) {print
"true"} else {print "false"}}'
```

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)
	DISA STIG(s)	¥ APPL-15-003011
	CCE	¥ CCE-94344-9

10.11. 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.

> **!** This rule is marked as manual and may not be able to be automated. It is also excluded in the compliance scan and will not report any results.

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)
	DISA STIG(s)	¥ APPL-15-000012
	CCE	¥ CCE-94346-4

Chapter 11. System Settings

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

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

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	system_settings_airplay_receiver_disable	
References	800-53r5	¥ CM-7, CM-7(1)
	DISA STIG(s)	¥ APPL-15-002080
	CCE	¥ CCE-94348-0

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.

> **!** Unlocking the system with an Apple Watch is not an approved authenticator for US Federal Government usage as it has not been verified to meet the strength requirements outlined in NIST SP 800-63.

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	system_settings_apple_watch_unlock_disable	
References	800-53r5	¥ IA-5
	DISA STIG(s)	¥ APPL-15-000001
	CCE	¥ CCE-94349-8

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	system_settings_automatic_login_disable	
References	800-53r5	¥ IA-2
		¥ IA-5(13)
	DISA STIG(s)	¥ APPL-15-002066
	CCE	¥ CCE-94350-6

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	system_settings_automatic_logout_enforce	
References	800-53r5	¥ AC-12
		¥ AC-2(5)
	DISA STIG(s)	¥ APPL-15-000160
	CCE	¥ CCE-94351-4

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	system_settings_bluetooth_disable	
References	800-53r5	¥ AC-18, AC-18(3) ¥ SC-8
	DISA STIG(s)	¥ APPL-15-002062
	CCE	¥ CCE-94352-2

11.6. Disable the Bluetooth System Settings Pane

The Bluetooth System Setting pane *MUST* be disabled to prevent access to the bluetooth configuration.

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

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

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

Remediation Description

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

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

```
<key>DisabledSystemSettings</key>
<array>
Ê    <string>com.apple.BluetoothSettings</string>
</array>
```

ID	system_settings_bluetooth_settings_disable	
References	800-53r5	¥ CM-7, CM-7(1)
	DISA STIG(s)	¥ APPL-15-002260
	CCE	¥ CCE-94354-8

11.7. 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	system_settings_bluetooth_sharing_disable

References	800-53r5	¥ AC-18(4) ¥ AC-3 ¥ CM-7, CM-7(1)
	DISA STIG(s)	¥ APPL-15-002110
	CCE	¥ CCE-94355-5

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	system_settings_content_caching_disable	
References	800-53r5	¥ CM-7, CM-7(1)
	DISA STIG(s)	¥ APPL-15-002140
	CCE	¥ CCE-94357-1

11.9. 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 ){
Ê   return("true")
} else {
Ê   return("false")
}
}
EOS
```

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

Remediation Description

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

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

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

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

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

ID	system_settings_diagnostics_reports_disable

References	800-53r5	¥ AC-20 ¥ SC-7(10) ¥ SI-11
	DISA STIG(s)	¥ APPL-15-002021
	CCE	¥ CCE-94359-7

11.10. 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
Ê echo "1"
else
Ê 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	system_settings_filevault_enforce

References	800-53r5	¥ SC-28, SC-28(1)
	DISA STIG(s)	¥ APPL-15-005020
	CCE	¥ CCE-94360-5

11.11. 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() {
Ê let pref1 = ObjC.unwrap(
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowFindMyDevice'))
Ê let pref2 = ObjC.unwrap(
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowFindMyFriends'))
Ê let pref3 = ObjC.unwrap(
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.icloud.managed')\
.objectForKey('DisableFMMiCloudSetting'))
Ê if ( pref1 == false && pref2 == false && pref3 == true ) {
Ê   return("true")
Ê } else {
Ê   return("false")
Ê }
}
EOS
```

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

Remediation Description

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

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

```
<key>allowFindMyDevice</key>
```

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

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

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

ID	system_settings_find_my_disable	
References	800-53r5	¥ AC-20 ¥ CM-7, CM-7(1)
	DISA STIG(s)	¥ APPL-15-002180
	CCE	¥ CCE-94361-3

11.12. Enable macOS Application Firewall

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

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

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

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

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

Remediation Description

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

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

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

ID	system_settings_firewall_enable	
References	800-53r5	¥ AC-4 ¥ CM-7, CM-7(1) ¥ SC-7, SC-7(12)
	DISA STIG(s)	¥ APPL-15-005050
	CCE	¥ CCE-94362-1

11.13. 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/bin/osascript -l JavaScript << EOS
function run() {
Ê let pref1 = ObjC.unwrap(
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.systempolicy.control')\
.objectForKey('AllowIdentifiedDevelopers'))
Ê let pref2 = ObjC.unwrap(
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.systempolicy.control')\
.objectForKey('EnableAssessment'))
Ê if ( pref1 == true && pref2 == true ) {
Ê   return("true")
Ê } else {
Ê   return("false")
Ê }
}
EOS
```

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

Remediation Description

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

Create a configuration profile containing the following keys in the

(com.apple.systempolicy.control) payload type:

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

ID	system_settings_gatekeeper_identified_developers_allowed	
References	800-53r5	¥ CM-14
		¥ CM-5
		¥ SI-7(1), SI-7(15)
	DISA STIG(s)	¥ APPL-15-002060
	CCE	¥ CCE-94364-7

11.14. 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() {
Ê let pref1 = ObjC.unwrap($.NSUserDefaults.alloc.initWithSuiteName('com.apple.MCX')\
.objectForKey('DisableGuestAccount'))
Ê let pref2 = ObjC.unwrap($.NSUserDefaults.alloc.initWithSuiteName('com.apple.MCX')\
.objectForKey('EnableGuestAccount'))
Ê if ( pref1 == true && pref2 == false ) {
Ê   return("true")
Ê } else {
Ê   return("false")
Ê }
}
EOS
```

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

Remediation Description

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

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

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

ID	system_settings_guest_account_disable	
References	800-53r5	¥ AC-2, AC-2(9)
	DISA STIG(s)	¥ APPL-15-002063
	CCE	¥ CCE-94367-0

11.15. 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	system_settings_hot_corners_disable	
References	800-53r5	¥ AC-11(1)
	DISA STIG(s)	¥ APPL-15-000007
	CCE	¥ CCE-94368-8

11.16. Disable Sending Audio Recordings and Transcripts to Apple

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

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

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

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

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

Remediation Description

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

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

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

ID	system_settings_improve_assistive_voice_disable

References	800-53r5	¥ AC-20 ¥ CM-7, CM-7(1) ¥ SC-7(10)
	DISA STIG(s)	¥ APPL-15-002023
	CCE	¥ CCE-94370-4

11.17. Disable Improve Search Information to Apple

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

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

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

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

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

Remediation Description

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

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

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

ID	system_settings_improve_search_disable	
References	800-53r5	¥ AC-20 ¥ CM-7, CM-7(1) ¥ SC-7(10)
	DISA STIG(s)	¥ APPL-15-002024
	CCE	¥ CCE-94371-2

11.18. Disable Improve Siri and Dictation Information to Apple

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

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

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

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

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

Remediation Description

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

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

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

ID	system_settings_improve_siri_dictation_disable	
References	800-53r5	¥ AC-20
		¥ CM-7, CM-7(1)
		¥ SC-7(10)
	DISA STIG(s)	¥ APPL-15-002210
	CCE	¥ CCE-94372-0

11.19. Disable Internet Sharing

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

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

Internet sharing helps prevent the unauthorized connection of devices, unauthorized transfer of information, and unauthorized tunneling.

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

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

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

Remediation Description

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

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

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

ID	system_settings_internet_sharing_disable	
References	800-53r5	¥ AC-20
		¥ AC-4
	DISA STIG(s)	¥ APPL-15-002007
	CCE	¥ CCE-94375-3

11.20. 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/sudo -u _locationd /usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.locationd')\
.objectForKey('LocationServicesEnabled').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:

```
/usr/bin/defaults write
/var/db/locationd/Library/Preferences/ByHost/com.apple.locationd
LocationServicesEnabled -bool false;
pid=$(/bin/launchctl list | /usr/bin/awk '/com.apple.locationd/ { print $1 }')
kill -9 $pid
```

ID	system_settings_location_services_disable	
References	800-53r5	¥ CM-7, CM-7(1)
		¥ SC-7(10)
	DISA STIG(s)	¥ APPL-15-002004
	CCE	¥ CCE-94376-1

11.21. Configure Login Window to Prompt for Username and Password

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

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

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

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

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

Remediation Description

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

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

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

ID	system_settings_loginwindow_prompt_username_password_enforce	
References	800-53r5	¥ IA-2
	DISA STIG(s)	¥ APPL-15-005052
	CCE	¥ CCE-94380-3

11.22. Disable Media Sharing

Media sharing *MUST* be disabled.

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

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

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

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

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

Remediation Description

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

Create a configuration profile containing the following keys in the

(com.apple.applicationaccess) payload type:

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

ID	system_settings_media_sharing_disabled	
References	800-53r5	¥ AC-17
		¥ AC-3
	DISA STIG(s)	¥ APPL-15-002100
	CCE	¥ CCE-94381-1

11.23. 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	system_settings_password_hints_disable

References	800-53r5	¥ IA-6
	DISA STIG(s)	¥ APPL-15-003012
	CCE	¥ CCE-94382-9

11.24. Disable Personalized Advertising

Ad tracking and targeted ads *MUST* be disabled.

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

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

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

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

Remediation Description

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

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

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

ID	system_settings_personalized_advertising_disable	
References	800-53r5	¥ AC-20
		¥ CM-7, CM-7(1)
		¥ SC-7(10)
	DISA STIG(s)	¥ APPL-15-002200
	CCE	¥ CCE-94383-7

11.25. 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	system_settings_printer_sharing_disable	
References	800-53r5	¥ CM-7, CM-7(1)
	DISA STIG(s)	¥ APPL-15-002240
	CCE	¥ CCE-94384-5

11.26. Disable Remote Apple Events

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

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

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

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

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

Remediation Description

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

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

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

ID	system_settings_rae_disable	
References	800-53r5	¥ AC-17 ¥ AC-3
	DISA STIG(s)	¥ APPL-15-002022
	CCE	¥ CCE-94385-2

11.27. 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	system_settings_remote_management_disable	
References	800-53r5	¥ CM-7, CM-7(1)
	DISA STIG(s)	¥ APPL-15-002250
	CCE	¥ CCE-94386-0

11.28. 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" => disabled'
```

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

Remediation Description

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

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

NOTE - This will apply to the whole system

ID	system_settings_screen_sharing_disable	
References	800-53r5	¥ AC-17
		¥ AC-3
	DISA STIG(s)	¥ APPL-15-002050
	CCE	¥ CCE-94387-8

11.29. 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() {
Ê let delay = ObjC.unwrap(
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.screensaver')\
.objectForKey('askForPasswordDelay'))
Ê if ( delay <= 5 ) {
Ê   return("true")
Ê } else {
Ê   return("false")
Ê }
```

```
}
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	system_settings_screensaver_ask_for_password_delay_enforce	
References	800-53r5	¥ AC-11
	DISA STIG(s)	¥ APPL-15-000003
	CCE	¥ CCE-94388-6

11.30. 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	system_settings_screensaver_password_enforce	
References	800-53r5	¥ AC-11
	DISA STIG(s)	¥ APPL-15-000002
	CCE	¥ CCE-94389-4

11.31. Enforce Screen Saver Timeout

The screen saver timeout *MUST* be set to 900 seconds or a shorter length of time.

This rule ensures that a full session lock is triggered within no more than 900 seconds of inactivity.

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

```
/usr/bin/osascript -l JavaScript << EOS
function run() {
Ê let timeout = ObjC.unwrap(
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.screensaver')\
.objectForKey('idleTime'))
Ê if ( timeout <= 900 ) {
Ê   return("true")
Ê } else {
Ê   return("false")
Ê }
}
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>900</integer>
```

ID	system_settings_screensaver_timeout_enforce

References	800-53r5	¥ AC-11 ¥ IA-11
	DISA STIG(s)	¥ APPL-15-000070
	CCE	¥ CCE-94390-2

11.32. Disable Siri

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

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

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

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

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

Remediation Description

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

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

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

ID	system_settings_siri_disable	
References	800-53r5	¥ AC-20 ¥ CM-7, CM-7(1) ¥ SC-7(10)
	DISA STIG(s)	¥ APPL-15-002020
	CCE	¥ CCE-94391-0

11.33. Disable the System Settings Pane for Siri

The System Settings pane for Siri *MUST* be hidden.

Hiding the System Settings pane prevents the users from configuring Siri.

> **!** Disabling the Siri System Settings pane blocks the user from opting into Apple Intelligence.

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

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

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

Remediation Description

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

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

```
<key>DisabledSystemSettings</key>
<array>
Ê   <string>com.apple.Siri-Settings.extension</string>
</array>
```

ID	system_settings_siri_settings_disable	
References	800-53r5	¥ CM-7, CM-7(1), CM-7(5)
	DISA STIG(s)	¥ APPL-15-002053
	CCE	¥ CCE-94393-6

11.34. Disable Server Message Block Sharing

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

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

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

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

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

Remediation Description

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

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

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

ID	system_settings_smbd_disable	
References	800-53r5	¥ AC-17
		¥ AC-3
	DISA STIG(s)	¥ APPL-15-002001
	CCE	¥ CCE-94394-4

11.35. Require Administrator Password to Modify System-Wide Preferences

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

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

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

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

```
!= "true" ]]; then
Ê   result="0"
Ê fi
Ê if [[ $(/usr/bin/security -q authorizationdb read "$section" | /usr/bin/xmllint
-xpath 'name(//*[contains(text(), "session-owner")]/following-sibling::*[1])' -) !=
"false" ]]; then
Ê   result="0"
Ê fi
done
echo $result
```

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

Remediation Description

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

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

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

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

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

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

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

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

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

ID	system_settings_system_wide_preferences_configure	
References	800-53r5	¥ AC-6, AC-6(1), AC-6(2)
	DISA STIG(s)	¥ APPL-15-002069
	CCE	¥ CCE-94401-7

11.36. Configure macOS to Use an Authorized Time Server

Approved time server *MUST* be the only server 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	system_settings_time_server_configure	
References	800-53r5	¥ AU-12(1) ¥ SC-45(1)
	DISA STIG(s)	¥ APPL-15-000170
	CCE	¥ CCE-94404-1

11.37. Enforce macOS Time Synchronization

Time synchronization *MUST* be enforced on all networked systems.

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	system_settings_time_server_enforce	
References	800-53r5	¥ AU-12(1) ¥ SC-45(1)
	DISA STIG(s)	¥ APPL-15-000014
	CCE	¥ CCE-94405-8

11.38. 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	system_settings_token_removal_enforce	
References	800-53r5	¥ AC-11
	DISA STIG(s)	¥ APPL-15-000005
	CCE	¥ CCE-94406-6

11.39. 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.

> **!** TouchID is not an approved biometric authenticator for US Federal Government usage as it has not been verified to meet the strength requirements outlined in NIST SP 800-63.

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	system_settings_touchid_unlock_disable	
References	800-53r5	¥ IA-5
	DISA STIG(s)	¥ APPL-15-002090
	CCE	¥ CCE-94408-2

11.40. USB Devices Must be Authorized Before Allowing

USB devices connected to a Mac *MUST* be authorized.

> " This feature is removed if a smartcard is paired or smartcard attribute mapping is configured.

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

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

ID	system_settings_usb_restricted_mode

References	800-53r5	¥ MP-7
		¥ SC-41
	DISA STIG(s)	¥ APPL-15-005090
	CCE	¥ CCE-94409-0

11.41. Disable the System Settings Pane for Wallet and Apple Pay

The System Settings pane for Wallet and Apple Pay *MUST* be disabled.

Disabling the System Settings pane prevents the users from configuring Wallet and Apple Pay.

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

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

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

Remediation Description

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

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

```
<key>DisabledSystemSettings</key>
<array>
Ê   <string>com.apple.WalletSettingsExtension</string>
</array>
```

ID	system_settings_wallet_applepay_settings_disable	
References	800-53r5	¥ CM-7, CM-7(1), CM-7(5)
	DISA STIG(s)	¥ APPL-15-002052
	CCE	¥ CCE-94411-6

Chapter 12. Supplemental

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

12.1. Out of Scope Supplemental

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

12.2. FileVault Supplemental

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

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

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

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

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

Using the fdesetup Command

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

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

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

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

Using a Configuration Profile

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

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

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

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

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

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

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

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

12.3. Packet Filter (pf) Supplemental

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

- os_firewall_default_deny_require

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

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

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

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

The ruleset will block connections on the following ports:

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

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

For more on configuring the PF firewall check out the man pages on `pf.conf` and `pfctl`.

```
#!/bin/bash
# Title          : enablePF-mscp.sh
# Description    : This script will configure the packet filter `pf` with the settings
recommended by the macOS Security Compliance Project (MSCP)
# Author         : Dan Brodjieski
# Date           : 2023-10-05
# Version        : 1.0
# Usage          : enablePF-mscp.sh [--uninstall]
# Notes          : Script must be run with privileges
#                : Configuring `pf` with a content filter installed may have
unexpected results
# Changelog      : 2023-10-05 - Added --uninstall parameter, refactored script for
better functionality

#### verify running as root
if [[ $EUID -ne 0 ]]; then
Ê   echo "This script must be run as root or with sudo, exiting..."
Ê   exit 1
fi

#### Setup environment
launchd_pfctl_plist="/Library/LaunchDaemons/mscp.pfctl.plist"
```

```
legacy_launchd_plist="/Library/LaunchDaemons/macsec.pfctl.plist"

mdm_managed=$(/usr/bin/osascript -l JavaScript -e "
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.security.firewall').objectIsForced
ForKey('EnableFirewall')")

#### Functions ####

#enabling macos application firewall
enable_macos_application_firewall () {
Ê   echo "The macOS application firewall is not managed by a profile, enabling from
CLI"
Ê   /usr/libexec/ApplicationFirewall/socketfilterfw --setglobalstate on
Ê   /usr/libexec/ApplicationFirewall/socketfilterfw --setloggingopt detail
Ê   /usr/libexec/ApplicationFirewall/socketfilterfw --setallowsigned on
Ê   /usr/libexec/ApplicationFirewall/socketfilterfw --setallowsignedapp on

}

#enabling pf firewall with mscp rules
enable_pf_firewall_with_mscp_rules () {
Ê   echo "Creating LaunchDeamon to load the MSCP rules"
Ê   if [[ -e "$launchd_pfctl_plist" ]]; then
Ê       echo "LaunchDaemon already exists, flushing and reloading rules..."
Ê       pfctl -e 2> /dev/null
Ê       pfctl -f /etc/pf.conf 2> /dev/null
Ê       return 0
Ê   fi

Ê   # copy system provided launchd for custom ruleset
Ê   cp "/System/Library/LaunchDaemons/com.apple.pfctl.plist" "$launchd_pfctl_plist"
Ê   #allow pf to be enabled when the job is loaded
Ê   /usr/libexec/PlistBuddy -c "Add :ProgramArguments:1 string -e"
$launchd_pfctl_plist
Ê   #use new label to not conflict with System's pfctl
Ê   /usr/libexec/PlistBuddy -c "Set :Label mscp.pfctl" $launchd_pfctl_plist

Ê   # enable the firewall
Ê   pfctl -e 2> /dev/null

Ê   #make pf run at system startup
Ê   launchctl enable system/mscp.pfctl
Ê   launchctl bootstrap system $launchd_pfctl_plist

Ê   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 () {
Ê   echo "Adding the MSCP anchors to /etc/pf.conf"
```

```
Ê   # check to see if mscp anchors exists
Ê   anchors_exist=$(grep -c '^anchor "mscp_pf_anchors"' /etc/pf.conf)

Ê   if [[ $anchors_exist == "0" ]];then
Ê       echo 'anchor "mscp_pf_anchors"' >> /etc/pf.conf
Ê       echo 'load anchor "mscp_pf_anchors" from "/etc/pf.anchors/mscp_pf_anchors"' >>
/etc/pf.conf
Ê   else
Ê       echo "mscp anchors exist, continuing..."
Ê   fi

}


# Create /etc/pf.anchors/mscp_pf_anchors
create_mscp_pf_anchors () {
Ê   echo "Creating the MSCP anchor configuration file"
if [[ -e /etc/pf.anchors/mscp_pf_anchors ]]; then
Ê   echo "mscp Anchor file exists, deleting and recreating..."
Ê   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() {
Ê   echo "References to macsec appear to exist, removing..."

Ê   launchctl disable system/macsec.pfctl
Ê   launchctl bootout system $legacy_launchd_plist
Ê   rm -rf $legacy_launchd_plist
Ê
Ê   # check to see if macsec anchors exists
Ê   anchors_exist=$(grep -c '^anchor "macsec_pf_anchors"' /etc/pf.conf)

Ê   if [[ ! $anchors_exist == "0" ]];then
Ê       sed -i "" '/macsec/d' /etc/pf.conf
Ê   else
Ê       echo "macsec anchors do not exist, continuing..."
Ê   fi

Ê   rm -f /etc/pf.anchors/macsec_pf_anchors
}

uninstall_mscp_pf(){
Ê   echo "Removing MSCP configuration files from pf"
Ê   if [[ -e "$launchd_pfctl_plist" ]]; then
Ê       echo "LaunchDaemon exists, unloading and removing"
Ê       #remove mscp pf components from launchd
Ê       launchctl disable system/mscp.pfctl
Ê       launchctl bootout system $launchd_pfctl_plist
Ê       rm -rf $launchd_pfctl_plist
Ê   fi
Ê
Ê   # check to see if mscp anchors exists
Ê   anchors_exist=$(grep -c '^anchor "mscp_pf_anchors"' /etc/pf.conf)

Ê   if [[ ! $anchors_exist == "0" ]];then
Ê       sed -i "" '/mscp/d' /etc/pf.conf
Ê   else
Ê       echo "mscp anchors do not exist, continuing..."
Ê   fi

Ê   rm -f /etc/pf.anchors/mscp_pf_anchors

Ê   # flush rules and reload pf
Ê   echo "Flushing rules and reloading pf"
```

```
Ê   pfctl -f /etc/pf.conf 2> /dev/null #flush the pf ruleset (reload the rules)

}

#### Main Script ####

POSITIONAL_ARGS=()

while [[ $# -gt 0 ]]; do
Ê case $1 in
Ê   -u|--uninstall)
Ê     UNINSTALL="true"
Ê     shift # past argument
Ê     shift # past value
Ê     ;;
Ê   -*|--*)
Ê     echo "Unknown option $1"
Ê     exit 1
Ê     ;;
Ê   *)
Ê     POSITIONAL_ARGS+=("$1") # save positional arg
Ê     shift # past argument
Ê     ;;
Ê esac
done

set -- "${POSITIONAL_ARGS[@]}" # restore positional parameters

if [[ $UNINSTALL == "true" ]]; then
Ê   if [[ -e "$legacy_launchd_plist" ]]; then
Ê       remove_macsec_setup
Ê   fi
Ê   uninstall_mscp_pf
Ê   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
Ê    enable_macos_application_firewall
fi

# clean up any legacy configurations
if [[ -e "$legacy_launchd_plist" ]]; then
Ê   echo "References to macsec appear to exist, removing..."
Ê   remove_macsec_setup
fi

# create mscp anchors file
create_mscp_pf_anchors
```

```
# add the anchors to the /etc/pf.conf file
configure_pf_config_add_mscp_anchors

# create specific launch daemon for mscp configuration
enable_pf_firewall_with_mscp_rules
```

12.4. Password Policy Supplemental

To comply with Executive Order 14028, ÒImproving the NationÕs CybersecurityÓ, OMB M-22-09, ÒMoving the U.S. Government Toward Zero Trust Cybersecurity PrinciplesÓ, and NIST SP-800-63b, ÒDigital Identity Guidelines: Authentication and Lifecycle ManagementÓ federal, military, and intelligence communities must adopt the following configuration settings:

- Password policies must not require the use of complexity policies such as upper characters, lower characters, or special characters.
- Password policies must also not require the use of regular rotation.

In accordance with these requirements, the following rules, while they remain on specific benchmarks, have been removed from any of the NIST 800-53r5 baselines as recommendations.

- pwpolicy_alpha_numeric_enforce
- pwpolicy_custom_regex_enforce
- pwpolicy_lower_case_character_enforce.yaml
- pwpolicy_max_lifetime_enforce
- pwpolicy_minimum_lifetime_enforce
- pwpolicy_prevent_dictionary_words
- pwpolicy_simple_sequence_disable
- pwpolicy_special_character_enforce
- pwpolicy_upper_case_character_enforce.yaml

If an organization has requirements to implement additional password policies, the remainder of this supplemental discusses the following password policy rules:

- pwpolicy_lower_case_character_enforce
- pwpolicy_upper_case_character_enforce
- pwpolicy_account_inactivity_enforce
- pwpolicy_minimum_lifetime_enforce

Password policies should be enforced as much as possible via Configuration Profiles. However, the following policies are currently not enforceable via Configuration Profiles, and must therefore be enabled using the `pwpolicy` command:

- Enforcing at least 1 lowercase character
- Enforcing at least 1 uppercase character

¥ Disabling an account after 35 days of inactivity

¥ Password minimum lifetime

To set the local policy to meet these requirements, save the following XML password policy to a file.

```
Ê <?xml version="1.0" encoding="UTF-8"?>
Ê <!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
Ê <plist version="1.0">
Ê <dict>
Ê   <key>policyCategoryAuthentication</key>
Ê   <array>
Ê     <dict>
Ê       <key>policyContent</key>
Ê       <string>policyAttributeLastAuthenticationTime &gt; policyAttributeCurrentTime
- (policyAttributeInactiveDays * 24 * 60 * 60)</string>
Ê       <key>policyIdentifier</key>
Ê       <string>Inactive Account</string>
Ê       <key>policyParameters</key>
Ê       <dict>
Ê         <key>policyAttributeInactiveDays</key>
Ê         <integer>35</integer>
Ê       </dict>
Ê     </dict>
Ê   </array>
Ê   <key>policyCategoryPasswordContent</key>
Ê   <array>
Ê     <dict>
Ê       <key>policyContent</key>
Ê       <string>policyAttributePassword matches '(.*[A-Z].*){1,}+'</string>
Ê       <key>policyIdentifier</key>
Ê       <string>Must have at least 1 uppercase letter</string>
Ê       <key>policyParameters</key>
Ê       <dict>
Ê         <key>minimumAlphaCharactersUpperCase</key>
Ê         <integer>1</integer>
Ê       </dict>
Ê     </dict>
Ê     <dict>
Ê       <key>policyContent</key>
Ê       <string>policyAttributePassword matches '(.*[a-z].*){1,}+'</string>
Ê       <key>policyIdentifier</key>
Ê       <string>Must have at least 1 lowercase letter</string>
Ê       <key>policyParameters</key>
Ê       <dict>
Ê         <key>minimumAlphaCharactersLowerCase</key>
Ê         <integer>1</integer>
Ê       </dict>
Ê     </dict>
Ê     <dict>
```

```
Ê       <key>policyContent</key>
Ê       <string>policyAttributeLastPasswordChangeTime &lt; policyAttributeCurrentTime
- (policyAttributeMinimumLifetimeHours * 60 * 60)</string>
Ê       <key>policyIdentifier</key>
Ê       <string>Minimum Password Lifetime</string>
Ê       <key>policyParameters</key>
Ê       <dict>
Ê         <key>policyAttributeMinimumLifetimeHours</key>
Ê         <integer>24</integer>
Ê       </dict>
Ê     </dict>
Ê   </array>
Ê </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.

12.5. Smartcard Supplemental

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

- auth_ssh_password_authentication_disable
- auth_smartcard_enforce
- auth_smartcard_certificate_trust_enforce_moderate
- auth_smartcard_certificate_trust_enforce_high
- auth_smartcard_allow
- auth_pam_sudo_smartcard_enforce
- auth_pam_su_smartcard_enforce
- auth_pam_login_smartcard_enforce

macOS supports smartcards, such as U.S. Personal Identity Verification (PIV) cards and U.S. Department of Defense Common Access Cards (CAC). Smartcards can be used on a macOS for the following:

- Authentication (Loginwindow, Screensaver, SSH, PKINIT, Safari, Finder, and PAM Authorization (`sudo`, `login`, and `su`))
- Digital Encryption
- Digital Signing

- Remote Access (VPN:L2TP)
- Port-based Network Access Control (802.1X)
- Keychain Unlock

macOS has built-in support for USB CCID class-compliant smartcard readers.

Smartcard Pairing

The default method for using smartcards in macOS is a method called "local account pairing". Local account pairing is automatically initiated when a user inserts a smartcard into the Mac. The user is prompted to pair their smartcard with their account. If a user receives a new smartcard, the previous card must be unpaired, and the new card paired to the account. Local account pairing employs fixed key mapping with the hash of a public key on the user's smartcard with a local account.

Smartcard Attribute Mapping

Smartcards can be used to authenticate against a directory via attribute mapping configured in `/private/etc/SmartcardLogin.plist`. This file takes precedence over local account pairing. Attribute mapping matches the configured certificate field values from the smart card to the value in a directory. This may be used with network accounts, mobile accounts, or local accounts.

Smartcard Management in macOS

The following settings are available to manage smartcards (com.apple.security.smartcard):

Key	Type	Value
userPairing	bool	If false, users will not get the pairing dialog, although existing pairings will still work.
allowSmartCard	bool	If false, the SmartCard is disabled for logins, authorizations, and screensaver unlocking. It is still allowed for other functions, such as signing emails and web access. A restart is required for a change of setting to take effect.

Key	Type	Value
checkCertificateTrust	int	Valid values are 0-3: ¥ 0: certificate trust check is turned off ¥ 1: certificate trust check is turned on. Standard validity check is being performed but this does not include additional revocation checks. ¥ 2: certificate trust check is turned on, and a soft revocation check is performed. Until the certificate is explicitly rejected by CRL/OCSP, it is considered valid. This implies that unavailable/unreachable CRL/OCSP allows this check to succeed. ¥ 3: certificate trust check is turned on, plus a hard revocation check is performed. Unless CRL/OCSP explicitly states that "this certificate is OK", the certificate is considered invalid. This is the most secure value for this setting.
oneCardPerUser	bool	If true, a user can pair with only one smartcard, although existing pairings will be allowed if already set up.
enforceSmartCard	bool	If true, a user can only login or authenticate with a smartcard.
tokenRemovalAction	int	If 1, the screen saver will automatically when the smartcard is removed.
allowUnmappedUsers	int	If 1, allows users who are in a directory group to be exempt from smartcard-only enforcement. The group allowed for exemption is defined in /private/etc/SmartcardLogin.plist

A custom configuration profile (`com.apple.loginwindow`) should be created to disable automatic login when FileVault is enabled. This ensures that authorized users boot their Macs, enter a password at the pre-boot screen (which decrypts the boot volume), and are then presented with a login window where they can authenticate with a smartcard.

Key	Type	Value
DisableFDEAutoLogin	bool	If true, both Extensible Firmware Interface (EFI) login password and loginwindow PIN are required.

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

Trusted Authorities

The macOS allows users to specify which certificate authorities (CA) can be used for trust evaluation during smartcard authentication. Only CAs listed in the TrustedAuthorities section of the SmartcardLogin.plist will be evaluated as trusted. This setting only works if `checkCertificateTrust` is set to either 1, 2, or 3 in `com.apple.security.smartcard`.

To get the SHA-256 hash in the correct format, run the following command within terminal:

```
/usr/bin/openssl x509 -noout -fingerprint -sha256 -inform pem -in <issuer cert> |
/usr/bin/awk -F '=' '{print $2}' |  /usr/bin/sed 's/://g'
```

To configure Trusted Authorities, the SmartcardLogin.plist should be minimally configured as below:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
Ê   <key>AttributeMapping</key>
Ê   <dict>
Ê         <key>fields</key>
Ê         <array>
Ê             <string>NT Principal Name</string>
Ê         </array>
Ê         <key>formatString</key>
Ê         <string>Kerberos:$1</string>
Ê         <key>dsAttributeString</key>
Ê         <string>dsAttrTypeStandard:AltSecurityIdentities</string>
Ê   </dict>
Ê   <key>TrustedAuthorities</key>
Ê <array>
Ê     <string>SHA256_HASH_OF_CERTDOMAIN_1,SHA256_HASH_OF_CERTDOMAIN_2</string>
Ê </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>
```

```
Ê   <key>AttributeMapping</key>
Ê   <dict>
Ê         <key>fields</key>
Ê         <array>
Ê             <string>NT Principal Name</string>
Ê         </array>
Ê         <key>formatString</key>
Ê         <string>Kerberos:$1</string>
Ê         <key>dsAttributeString</key>
Ê         <string>dsAttrTypeStandard:AltSecurityIdentities</string>
Ê   </dict>
Ê   <key>TrustedAuthorities</key>
Ê <array>
Ê     <string>SHA256_HASH_OF_CERTDOMAIN_1,SHA256_HASH_OF_CERTDOMAIN_2</string>
Ê </array>
Ê   <key>NotEnforcedGroup</key>
Ê   <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/>
```