



# macOS Security Compliance

macOS 26.0

## ***Security Configuration - CIS Controls Version 8***

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

# Table of Contents

1. Foreword .....	1
2. Scope .....	2
3. Authors .....	3
4. Acronyms and Definitions .....	4
5. Applicable Documents .....	6
5.1. Government Documents .....	6
5.2. Non-Government Documents .....	6
6. Auditing .....	7
6.1. Configure Audit Log Files to Not Contain Access Control Lists .....	7
6.2. Configure Audit Log Folder to Not Contain Access Control Lists .....	8
6.3. Enable Security Auditing .....	8
6.4. Configure Audit_Control to Not Contain Access Control Lists .....	10
6.5. Configure Audit_Control Group to Wheel .....	11
6.6. Configure Audit_Control Owner to Mode 440 or Less Permissive .....	11
6.7. Configure Audit_Control Owner to Root .....	12
6.8. Configure Audit Log Files Group to Wheel .....	12
6.9. Configure Audit Log Files to Mode 440 or Less Permissive .....	13
6.10. Configure Audit Log Files to be Owned by Root .....	14
6.11. Configure System to Audit All Authorization and Authentication Events .....	15
6.12. Configure System to Audit All Administrative Action Events .....	16
6.13. Configure System to Audit All Failed Program Execution on the System .....	17
6.14. Configure System to Audit All Failed Change of Object Attributes .....	18
6.15. Configure System to Audit All Failed Read Actions on the System .....	19
6.16. Configure System to Audit All Failed Write Actions on the System .....	20
6.17. Configure System to Audit All Log In and Log Out Events .....	21
6.18. Configure Audit Log Folders Group to Wheel .....	22
6.19. Configure Audit Log Folders to be Owned by Root .....	23
6.20. Configure Audit Log Folders to Mode 700 or Less Permissive .....	24
6.21. Configure Audit Retention to 7d .....	25
7. Authentication .....	26
7.1. Enforce Multifactor Authentication for Login .....	26
7.2. Enforce Multifactor Authentication for the su Command .....	27
7.3. Enforce Multifactor Authentication for Privilege Escalation Through the sudo Command ..	28
7.4. Allow Smartcard Authentication .....	29
7.5. Enforce Smartcard Authentication .....	30
7.6. Disable Password Authentication for SSH .....	31
8. iCloud .....	34
8.1. Disable iCloud Address Book .....	34

8.2. Disable the System Setting for Apple ID .....	35
8.3. Disable iCloud Bookmarks .....	36
8.4. Disable the iCloud Calendar Services .....	37
8.5. Disable iCloud Document Sync .....	38
8.6. Disable the iCloud Freeform Services .....	39
8.7. Disable iCloud Game Center .....	40
8.8. Disable iCloud Keychain Sync .....	41
8.9. Disable iCloud Mail .....	42
8.10. Disable iCloud Notes .....	43
8.11. Disable iCloud Photo Library .....	44
8.12. Disable iCloud Private Relay .....	45
8.13. Disable iCloud Reminders .....	46
8.14. Disable iCloud Desktop and Document Folder Sync .....	47
9. macOS .....	49
9.1. Disable AppleID and Internet Account Modifications .....	49
9.2. Disable AirDrop .....	50
9.3. Must Use an Approved Antivirus Program .....	51
9.4. Disable Apple ID Setup during Setup Assistant .....	52
9.5. Enable Authenticated Root .....	53
9.6. Disable Bonjour Multicast .....	54
9.7. Enforce Installation of XProtect Remediator and Gatekeeper Updates Automatically .....	55
9.8. Disable Dictation .....	56
9.9. Integrate System into a Directory Services Infrastructure .....	57
9.10. Must Use ESS .....	58
9.11. Disable FileVault Automatic Login .....	58
9.12. Enable Gatekeeper .....	59
9.13. Disable Handoff .....	60
9.14. Secure User's Home Folders .....	61
9.15. Disable the Built-in Web Server .....	62
9.16. Disable iCloud Storage Setup during Setup Assistant .....	63
9.17. Configure Install.log Retention to 365 .....	64
9.18. Disable iPhone Mirroring .....	65
9.19. Disable Infrared (IR) support .....	66
9.20. Enable Library Validation .....	67
9.21. Enforce Enrollment in Mobile Device Management .....	68
9.22. Enable Apple Mobile File Integrity .....	69
9.23. Disable Network File System Service .....	69
9.24. Enforce On Device Dictation .....	70
9.25. Remove Password Hint From User Accounts .....	71
9.26. Disable Proximity Based Password Sharing Requests .....	72
9.27. Disable Password Sharing .....	73

9.28. Disable Power Nap .....	74
9.29. Disable Privacy Setup Services During Setup Assistant .....	75
9.30. Disable Root Login .....	75
9.31. Ensure Advertising Privacy Protection in Safari Is Enabled .....	76
9.32. Disable Automatic Opening of Safe Files in Safari .....	77
9.33. Ensure Prevent Cross-site Tracking in Safari Is Enabled .....	78
9.34. Ensure Show Full Website Address in Safari Is Enabled .....	79
9.35. Ensure Show Safari shows the Status Bar is Enabled .....	80
9.36. Ensure Warn When Visiting A Fraudulent Website in Safari Is Enabled .....	80
9.37. Enforce FileVault in Setup Assistant .....	81
9.38. Ensure System Integrity Protection is Enabled .....	82
9.39. Disable Siri Setup during Setup Assistant .....	83
9.40. Disable Apple Intelligence During Setup Assistant .....	84
9.41. Disable Unlock with Apple Watch During Setup Assistant .....	85
9.42. Ensure Sleep and Display Sleep Is Enabled on Apple Silicon Devices .....	86
9.43. Enforce Software Update App Update Updates Automatically .....	87
9.44. Configure Sudo To Log Events .....	88
9.45. Configure Sudo Timeout Period to 0 .....	88
9.46. Configure Sudoers Timestamp Type .....	89
9.47. Ensure Appropriate Permissions Are Enabled for System Wide Applications .....	90
9.48. Ensure Secure Keyboard Entry Terminal.app is Enabled .....	91
9.49. Disable Trivial File Transfer Protocol Service .....	91
9.50. Enable Time Synchronization Daemon .....	92
9.51. Disable TouchID Prompt during Setup Assistant .....	93
9.52. Disable Login to Other User's Active and Locked Sessions .....	94
9.53. Disable Unix-to-Unix Copy Protocol Service .....	95
9.54. Ensure No World Writable Files Exist in the Library Folder .....	97
9.55. Ensure No World Writable Files Exist in the System Folder .....	97
10. Password Policy .....	99
10.1. Disable Accounts after 35 Days of Inactivity .....	99
10.2. Limit Consecutive Failed Login Attempts to 3 .....	100
10.3. Set Account Lockout Time to 15 Minutes .....	101
10.4. Require Passwords Contain a Minimum of One Numeric Character .....	102
10.5. Require Passwords to Match the Defined Custom Regular Expression .....	103
10.6. Prohibit Password Reuse for a Minimum of 5 Generations .....	104
10.7. Restrict Maximum Password Lifetime to 60 Days .....	105
10.8. Require a Minimum Password Length of 15 Characters .....	106
10.9. Set Minimum Password Lifetime to 24 Hours .....	107
10.10. Prohibit Repeating, Ascending, and Descending Character Sequences .....	109
10.11. Require Passwords Contain a Minimum of One Special Character .....	110
11. System Settings .....	112

11.1. Disable Airplay Receiver .....	112
11.2. Disable Unattended or Automatic Logon to the System .....	113
11.3. Disable Bluetooth When no Approved Device is Connected .....	114
11.4. Enable Bluetooth Menu .....	115
11.5. Disable the Bluetooth System Settings Pane .....	116
11.6. Disable Bluetooth Sharing .....	117
11.7. Disable Content Caching Service .....	118
11.8. Enforce Critical Security Updates to be Installed .....	118
11.9. Disable Sending Diagnostic and Usage Data to Apple .....	119
11.10. Enforce Software Update Downloads Updates Automatically using DDM. ....	120
11.11. Disable External Intelligence Integrations .....	121
11.12. Disable External Intelligence Integration Sign In .....	122
11.13. Enforce FileVault .....	123
11.14. Disable Find My Service .....	124
11.15. Enable macOS Application Firewall .....	125
11.16. Enable Firewall Stealth Mode .....	126
11.17. Disable Guest Access to Shared SMB Folders .....	127
11.18. Disable the Guest Account .....	128
11.19. Secure Hot Corners .....	129
11.20. Disable Sending Audio Recordings and Transcripts to Apple .....	130
11.21. Disable Improve Search Information to Apple .....	131
11.22. Disable Improve Siri and Dictation Information to Apple .....	132
11.23. Enforce macOS Updates are Automatically Installed .....	133
11.24. Disable the Internet Accounts System Preference Pane .....	134
11.25. Disable Internet Sharing .....	135
11.26. Enable Location Services .....	136
11.27. Configure Login Window to Show A Custom Message .....	137
11.28. Configure Login Window to Prompt for Username and Password .....	137
11.29. Disable Media Sharing .....	138
11.30. Disable Password Hints .....	139
11.31. Disable Personalized Advertising .....	140
11.32. Disable Printer Sharing .....	141
11.33. Disable Remote Apple Events .....	142
11.34. Disable Remote Management .....	143
11.35. Disable Screen Sharing and Apple Remote Desktop .....	144
11.36. Enforce Session Lock After Screen Saver is Started .....	145
11.37. Enforce Screen Saver Timeout .....	146
11.38. Enforce Automatic Installs of Available Security Updates using DDM. ....	147
11.39. Disable Siri .....	148
11.40. Ensure Siri Listen For is Disabled .....	148
11.41. Disable the System Settings Pane for Siri .....	149

11.42. Disable Server Message Block Sharing .....	150
11.43. Enforce Software Update Downloads Updates Automatically .....	151
11.44. Ensure Software Update is Updated and Current .....	152
11.45. Disable SSH Server for Remote Access Sessions .....	153
11.46. Require Administrator Password to Modify System-Wide Preferences .....	154
11.47. Configure Time Machine for Automatic Backups .....	156
11.48. Ensure Time Machine Volumes are Encrypted .....	157
11.49. Configure macOS to Use an Authorized Time Server .....	158
11.50. Enforce macOS Time Synchronization .....	158
11.51. Disable the Touch ID System Settings Pane .....	159
11.52. Ensure Wake for Network Access Is Disabled .....	160
11.53. Disable the System Settings Pane for Wallet and Apple Pay .....	161
11.54. Disable Wi-Fi Interface .....	162
11.55. Enable Wifi Menu .....	163
12. Inherent .....	164
12.1. Enforce Approved Authorization for Logical Access .....	164
12.2. Ensure the System Implements Malicious Code Protection Mechanisms .....	164
12.3. Enforce multifactor authentication for network access to privileged accounts .....	166
12.4. Obscure Passwords .....	166
12.5. Encrypt Stored Passwords .....	167
12.6. Uniquely Identify Users and Processes .....	167
12.7. Force Password Change at Next Logon .....	168
13. Permanent Findings .....	169
13.1. Off-Load Audit Records .....	169
13.2. Must Authenticate Before Establishing a Connection .....	169
13.3. Secure Name Address Resolution Service .....	170
14. Not Applicable .....	171
14.1. Access Control for Mobile Devices .....	171
15. Supplemental .....	172
15.1. FileVault Supplemental .....	172
15.2. Password Policy Supplemental .....	173

# Chapter 1. Foreword

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

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

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

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

# Chapter 2. Scope

This guide describes the actions to take when securing a macOS 26.0 system against the CIS Controls Version 8 security baseline.

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



# Chapter 3. Authors

## macOS Security Compliance Project

CIS Critical Security Controls® (CIS Controls®) are referenced with the permission and support of the Center for Internet Security® (CIS®)

Edward Byrd	Center for Internet Security
Bob Gendler	National Institute of Standards and Technology
Dan Brodjieski	National Aeronautics and Space Administration
Allen Golbig	Jamf

# Chapter 4. Acronyms and Definitions

Table 1. Acronyms and Abbreviations

AES	Advanced Encryption Standard
ABM	Apple Business Manager
AFP	Apple Filing Protocol
ALF	Application Layer Firewall
AO	Authorizing Official
API	Application Programming Interface
ARD	Apple Remote Desktop
CA	Certificate Authority
CIS	Center for Internet Security
CMMC	Cybersecurity Maturity Model Certification
CNSSI	Committee on National Security Systems
CRL	Certificate Revocation List
DISA	Defense Information Systems Agency
DMA	Direct Memory Access
FISMA	Federal Information Security Modernization Act
FPKI	Federal Public Key Infrastructure
IR	Infrared
ISO	Information System Owner
ISSO	Information System Security Officer
MDM	Mobile Device Management
NASA	National Aeronautics and Space Administration
NFS	Network File System
NIST	National Institute of Standards and Technology
NSA	National Security Agency
OCSF	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
<a href="#">NIST Special Publication 800-53 Rev 5</a>	<i>NIST Special Publication 800-53 Rev 5.1.1</i>
<a href="#">NIST Special Publication 800-63</a>	<i>NIST Special Publication 800-63</i>
<a href="#">NIST Special Publication 800-171</a>	<i>NIST Special Publication 800-171 Rev 3</i>
<a href="#">NIST Special Publication 800-219</a>	<i>NIST Special Publication 800-219 Rev 1</i>

Table 4. Defense Information Systems Agency (DISA)

Document Number or Descriptor	Document Title
<a href="#">STIG Ver 1, Rel 4</a>	<i>Apple macOS 15 (Sequoia) STIG</i>

Table 5. Cybersecurity Maturity Model Certification (CMMC)

Document Number or Descriptor	Document Title
<a href="#">CMMC Model Overview v2.0</a>	<i>Cybersecurity Maturity Model Certification (CMMC) Model Overview v2.0</i>

Table 6. Committee on National Security Systems (CNSS)

Document Number or Descriptor	Document Title
<a href="#">CNSSI No. 1253</a>	<i>Security Categorization and Control Selection for National Security Systems</i>

## 5.2. Non-Government Documents

Table 7. Apple

Document Number or Descriptor	Document Title
<a href="#">Apple Platform Security Guide</a>	<i>Apple Platform Security</i>
<a href="#">Apple Platform Deployment</a>	<i>Apple Platform Deployment</i>
<a href="#">Apple Platform Certifications</a>	<i>Apple Platform Certifications</i>
<a href="#">Profile-Specific Payload Keys</a>	<i>Profile-Specific Payload Keys</i>

Table 8. Center for Internet Security

Document Number or Descriptor	Document Title
<a href="#">Apple macOS 15.0</a>	<i>CIS Apple macOS 15.0 Benchmark version 1.1.0</i>

# 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
	CIS Benchmark	• 3.5 (level 1)
	CIS Controls V8	• 3.3
	CCE	• CCE-95101-2

## 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
	CIS Benchmark	• 3.5 (level 1)
	CIS Controls V8	• 3.3
	CCE	• CCE-95102-0

## 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 Tahoe.

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

```
LAUNCHD_RUNNING=$( /bin/launchctl print system | /usr/bin/grep -c -E
'\tcom.apple.auditd')
AUDITD_RUNNING=$( /usr/sbin/audit -c | /usr/bin/grep -c "AUC_AUDITING")
if [[ $LAUNCHD_RUNNING == 1 ]] && [[ -e /etc/security/audit_control ]] && [[
$AUDITD_RUNNING == 1 ]]; then
    echo "pass"
else
    echo "fail"
fi
```

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

### Remediation Description

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

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

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• AU-12, AU-12(1), AU-12(3)</li> <li>• AU-14(1)</li> <li>• AU-3, AU-3(1)</li> <li>• AU-8</li> <li>• CM-5(1)</li> <li>• MA-4(1)</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 3.1 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 8.2</li> <li>• 8.5</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95104-6</li> </ul>

## 6.4. 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
```

<b>ID</b>	audit_control_acls_configure	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• AU-9</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 3.5 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 3.3</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95106-1</li> </ul>



# 6.5. 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
	CIS Benchmark	• 3.5 (level 1)
	CIS Controls V8	• 3.3
	CCE	• CCE-95107-9

# 6.6. 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
```

<b>ID</b>	audit_control_mode_configure	
<b>References</b>	<b>800-53r5</b>	• AU-9
	<b>CIS Benchmark</b>	• 3.5 (level 1)
	<b>CIS Controls V8</b>	• 3.3
	<b>CCE</b>	• CCE-95108-7

## 6.7. 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
```

<b>ID</b>	audit_control_owner_configure	
<b>References</b>	<b>800-53r5</b>	• AU-9
	<b>CIS Benchmark</b>	• 3.5 (level 1)
	<b>CIS Controls V8</b>	• 3.3
	<b>CCE</b>	• CCE-95109-5

## 6.8. 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
	CIS Benchmark	• 3.5 (level 1)
	CIS Controls V8	• 3.3
	CCE	• CCE-95112-9

## 6.9. 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
	CIS Benchmark	• 3.5 (level 1)
	CIS Controls V8	• 3.3
	CCE	• CCE-95113-7

## 6.10. 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
----	-----------------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• AU-9</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 3.5 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 3.3</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95114-5</li> </ul>

## 6.11. 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
```

<b>ID</b>	audit_flags_aa_configure
-----------	--------------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• AC-2(12)</li> <li>• AU-12</li> <li>• AU-2</li> <li>• CM-5(1)</li> <li>• MA-4(1)</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 3.2 (level 2)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 3.14</li> <li>• 8.2</li> <li>• 8.5</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95115-2</li> </ul>

## 6.12. 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	<ul style="list-style-type: none"><li>• AC-2(12), AC-2(4)</li><li>• AC-6(9)</li><li>• AU-12</li><li>• AU-2</li><li>• CM-5(1)</li><li>• MA-4(1)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 3.2 (level 2)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 3.14</li><li>• 8.2</li><li>• 8.5</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95116-0</li></ul>

## 6.13. 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	<ul style="list-style-type: none"><li>• AC-2(12)</li><li>• AU-12</li><li>• AU-2</li><li>• CM-5(1)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 3.2 (level 2)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 3.14</li><li>• 8.2</li><li>• 8.5</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95117-8</li></ul>

## 6.14. Configure System to Audit All Failed Change of Object Attributes

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

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

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

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

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

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

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

### Remediation Description

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



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

<b>ID</b>	audit_flags_fm_failed_configure	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• AC-2(12)</li> <li>• AU-12</li> <li>• AU-2</li> <li>• AU-9</li> <li>• CM-5(1)</li> <li>• MA-4(1)</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 3.2 (level 2)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 3.14</li> <li>• 8.2</li> <li>• 8.5</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95120-2</li> </ul>

## 6.15. 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	<ul style="list-style-type: none"><li>• AC-2(12)</li><li>• AU-12</li><li>• AU-2</li><li>• AU-9</li><li>• CM-5(1)</li><li>• MA-4(1)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 3.2 (level 2)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 3.14</li><li>• 8.2</li><li>• 8.5</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95121-0</li></ul>

## 6.16. 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	<ul style="list-style-type: none"><li>• AC-2(12)</li><li>• AU-12</li><li>• AU-2</li><li>• AU-9</li><li>• CM-5(1)</li><li>• MA-4(1)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 3.2 (level 2)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 3.14</li><li>• 8.2</li><li>• 8.5</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95122-8</li></ul>

## 6.17. 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.\*[^\n]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	<ul style="list-style-type: none"><li>• AC-17(1)</li><li>• AC-2(12)</li><li>• AU-12</li><li>• AU-2</li><li>• MA-4(1)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 3.2 (level 2)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 3.14</li><li>• 8.2</li><li>• 8.5</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95123-6</li></ul>

## 6.18. 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
	CIS Benchmark	• 3.5 (level 1)
	CIS Controls V8	• 3.3
	CCE	• CCE-95124-4

## 6.19. 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
----	------------------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"><li>• AU-9</li></ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"><li>• 3.5 (level 1)</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 3.3</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-95125-1</li></ul>

## 6.20. 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
```

<b>ID</b>	audit_folders_mode_configure	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"><li>• AU-9</li></ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"><li>• 3.5 (level 1)</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 3.3</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-95126-9</li></ul>

# 6.21. Configure Audit Retention to 7d

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

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

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

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

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

**Remediation Description**

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

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

ID	audit_retention_configure	
References	800-53r5	<ul style="list-style-type: none"><li>• AU-11</li><li>• AU-4</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 3.4 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 8.1</li><li>• 8.3</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95130-1</li></ul>

# 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_ondirectory.so
password     required      pam_ondirectory.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

```

<b>ID</b>	auth_pam_login_smartcard_enforce	
<b>References</b>	<b>800-53r5</b> <ul style="list-style-type: none"> <li>• IA-2(1), IA-2(2), IA-2(8)</li> </ul>	
	<b>CIS Benchmark</b> <ul style="list-style-type: none"> <li>• N/A</li> </ul>	
	<b>CIS Controls V8</b> <ul style="list-style-type: none"> <li>• 6.3</li> <li>• 6.4</li> <li>• 6.5</li> </ul>	
	<b>CCE</b> <ul style="list-style-type: none"> <li>• CCE-95132-7</li> </ul>	

## 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+ufficient\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	<ul style="list-style-type: none"><li>IA-2(1), IA-2(2), IA-2(8)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>N/A</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>6.3</li><li>6.4</li><li>6.5</li></ul>
	CCE	<ul style="list-style-type: none"><li>CCE-95133-5</li></ul>

### 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	<ul style="list-style-type: none"><li>IA-2(1), IA-2(2), IA-2(8)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>N/A</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>6.3</li><li>6.4</li><li>6.5</li></ul>
	CCE	<ul style="list-style-type: none"><li>CCE-95134-3</li></ul>

## 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)
	CIS Benchmark	• N/A
	CIS Controls V8	• 6.3
	CCE	• 6.4
		• 6.5
		• CCE-95135-0

## 7.5. 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	<ul style="list-style-type: none"><li>• IA-2, IA-2(1), IA-2(12), IA-2(2), IA-2(6), IA-2(8)</li><li>• IA-5(2)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• N/A</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 6.3</li><li>• 6.4</li><li>• 6.5</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95138-4</li></ul>

## 7.6. Disable Password Authentication for SSH

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

All users *MUST* go through multifactor authentication to prevent unauthenticated access and

potential compromise to the system.



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

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

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

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

### Remediation Description

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

```
include_dir=$(/usr/bin/awk '/^Include/ {print $2}' /etc/ssh/sshd_config |  
/usr/bin/tr -d '*')  
if [[ -z $include_dir ]]; then  
    /usr/bin/sed -i.bk "1s/.*/Include \/etc\/ssh\/sshd_config.d\/\*/"  
/etc/ssh/sshd_config  
fi  
echo "passwordauthentication no" >> "${include_dir}01-mscp-sshd.conf"  
echo "kbdinteractiveauthentication no" >> "${include_dir}01-mscp-sshd.conf"  
  
for file in $(ls ${include_dir}); do  
    if [[ "$file" == "100-macos.conf" ]]; then  
        continue  
    fi  
    if [[ "$file" == "01-mscp-sshd.conf" ]]; then  
        break  
    fi  
    /bin/mv ${include_dir}${file} ${include_dir}20-${file}  
done
```

ID	auth_ssh_password_authentication_disable
----	--

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• IA-2, IA-2(1), IA-2(2), IA-2(6), IA-2(8)</li> <li>• IA-5(2)</li> <li>• MA-4</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 6.3</li> <li>• 6.4</li> <li>• 6.5</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95139-2</li> </ul>

# 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
----	----------------------------



<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• AC-20, AC-20(1)</li> <li>• CM-7, CM-7(1)</li> <li>• SC-7(10)</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> <li>• 15.3</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95140-0</li> </ul>

## 8.2. Disable the System Setting for Apple ID

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

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

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

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

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

### Remediation Description

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

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

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

<b>ID</b>	icloud_appleid_system_settings_disable
-----------	--

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• AC-20, AC-20(1)</li> <li>• CM-7, CM-7(1)</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95141-8</li> </ul>

## 8.3. Disable iCloud Bookmarks

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

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

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

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

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

### Remediation Description

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

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

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

<b>ID</b>	icloud_bookmarks_disable
-----------	--------------------------

References	800-53r5	<ul style="list-style-type: none"> <li>• AC-20, AC-20(1)</li> <li>• CM-7, CM-7(1)</li> <li>• SC-7(10)</li> </ul>
	CIS Benchmark	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	CIS Controls V8	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> <li>• 15.3</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>• CCE-95142-6</li> </ul>

## 8.4. Disable the iCloud Calendar Services

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

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

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

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

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

### Remediation Description

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

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

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

ID	icloud_calendar_disable
----	-------------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• AC-20, AC-20(1)</li> <li>• CM-7, CM-7(1)</li> <li>• SC-7(10)</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> <li>• 15.3</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95143-4</li> </ul>

## 8.5. Disable iCloud Document Sync

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

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

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

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

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

### Remediation Description

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

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

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

<b>ID</b>	icloud_drive_disable
-----------	----------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• AC-20, AC-20(1)</li> <li>• CM-7, CM-7(1)</li> <li>• SC-7(10)</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> <li>• 15.3</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95144-2</li> </ul>

## 8.6. Disable the iCloud Freeform Services

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

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

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

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

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

### Remediation Description

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

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

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

<b>ID</b>	icloud_freeform_disable
-----------	-------------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• AC-20, AC-20(1)</li> <li>• CM-7, CM-7(1)</li> <li>• SC-7(10)</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> <li>• 15.3</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95145-9</li> </ul>

## 8.7. Disable iCloud Game Center

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

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

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

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

### Remediation Description

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

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

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

<b>ID</b>	icloud_game_center_disable
-----------	----------------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• AC-20, AC-20(1)</li> <li>• CM-7, CM-7(1)</li> <li>• SC-7(10)</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> <li>• 15.3</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95146-7</li> </ul>

## 8.8. Disable iCloud Keychain Sync

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

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

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

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

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

### Remediation Description

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

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

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

<b>ID</b>	icloud_keychain_disable
-----------	-------------------------

References	800-53r5	<ul style="list-style-type: none"> <li>• AC-20, AC-20(1)</li> <li>• CM-7, CM-7(1)</li> <li>• SC-7(10)</li> </ul>
	CIS Benchmark	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	CIS Controls V8	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> <li>• 15.3</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>• CCE-95147-5</li> </ul>

## 8.9. Disable iCloud Mail

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

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

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

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

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

### Remediation Description

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

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

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

ID	icloud_mail_disable
----	---------------------



<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• AC-20, AC-20(1)</li> <li>• CM-7, CM-7(1)</li> <li>• SC-7(10)</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> <li>• 15.3</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95148-3</li> </ul>

## 8.10. Disable iCloud Notes

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

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

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

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

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

### Remediation Description

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

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

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

<b>ID</b>	icloud_notes_disable
-----------	----------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• AC-20, AC-20(1)</li> <li>• CM-7, CM-7(1)</li> <li>• SC-7(10)</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> <li>• 15.3</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95149-1</li> </ul>

## 8.11. Disable iCloud Photo Library

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

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

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

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

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

### Remediation Description

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

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

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

<b>ID</b>	icloud_photos_disable
-----------	-----------------------

References	800-53r5	<ul style="list-style-type: none"> <li>• AC-20, AC-20(1)</li> <li>• CM-7, CM-7(1)</li> <li>• SC-7(10)</li> </ul>
	CIS Benchmark	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	CIS Controls V8	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> <li>• 15.3</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>• CCE-95150-9</li> </ul>

## 8.12. Disable iCloud Private Relay

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

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

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

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

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

### Remediation Description

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

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

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

ID	icloud_private_relay_disable
----	------------------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• AC-20, AC-20(1)</li> <li>• CM-7, CM-7(1)</li> <li>• SC-7(10)</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> <li>• 15.3</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95151-7</li> </ul>

## 8.13. Disable iCloud Reminders

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

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

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

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

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

### Remediation Description

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

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

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

<b>ID</b>	icloud_reminders_disable
-----------	--------------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• AC-20, AC-20(1)</li> <li>• CM-7, CM-7(1)</li> <li>• SC-7(10)</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> <li>• 15.3</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95152-5</li> </ul>

## 8.14. Disable iCloud Desktop and Document Folder Sync

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

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

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

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

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

### Remediation Description

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

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

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

<b>ID</b>	icloud_sync_disable
-----------	---------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• AC-20, AC-20(1)</li> <li>• CM-7, CM-7(1)</li> <li>• SC-7(10)</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 2.1.1.3 (level 2)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> <li>• 15.3</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95153-3</li> </ul>

# 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/>
```

<b>ID</b>	os_account_modification_disable	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• AC-20, AC-20(1)</li> <li>• CM-7, CM-7(1)</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95155-8</li> </ul>

## 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/>
```

<b>ID</b>	os_airdrop_disable
-----------	--------------------



<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• AC-20</li> <li>• AC-3</li> <li>• CM-7, CM-7(1)</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 2.3.1.1 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> <li>• 6.7</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95156-6</li> </ul>

## 9.3. Must Use an Approved Antivirus Program

An approved antivirus product *MUST* be installed and configured to run.

Malicious software can establish a base on individual desktops and servers. Employing an automated mechanism to detect this type of software will aid in elimination of the software from the operating system.'

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

```
/usr/bin/xprotect status | /usr/bin/grep -cE "(launch scans: enabled|background scans: enabled)"
```

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

### Remediation Description

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

```
/bin/launchctl load -w
/Library/Apple/System/Library/LaunchDaemons/com.apple.XProtect.daemon.scan.plist
/bin/launchctl load -w
/Library/Apple/System/Library/LaunchDaemons/com.apple.XprotectFramework.PluginService.plist
```



These services cannot be unloaded or loaded while System Integrity Protection (SIP) is enabled.

<b>ID</b>	os_anti_virus_installed
-----------	-------------------------

References	800-53r5	<ul style="list-style-type: none"><li>• N/A</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 5.10 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 10.5</li><li>• 10.1</li><li>• 10.2</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95158-2</li></ul>

## 9.4. Disable Apple ID Setup during Setup Assistant

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

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

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

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

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

**Remediation Description**

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

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

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

ID	os_appleid_prompt_disable
----	---------------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• AC-20</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95159-0</li> </ul>

## 9.5. Enable Authenticated Root

Authenticated Root *MUST* be enabled.

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



Authenticated Root is enabled by default on macOS systems.



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

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

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

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

### Remediation Description

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

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



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

<b>ID</b>	os_authenticated_root_enable
-----------	------------------------------

References	800-53r5	<ul style="list-style-type: none"><li>• AC-3</li><li>• CM-5</li><li>• MA-4(1)</li><li>• SC-34</li><li>• SI-7, SI-7(6)</li><li>• 5.1.4 (level 1)</li></ul>
	CIS Benchmark	
	CIS Controls V8	<ul style="list-style-type: none"><li>• 3.6</li><li>• 3.11</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95164-0</li></ul>

## 9.6. Disable Bonjour Multicast

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

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

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

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

**Remediation Description**

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

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

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

ID	os_bonjour_disable
----	--------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• CM-7, CM-7(1)</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 4.1 (level 2)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95169-9</li> </ul>

## 9.7. 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/>
```

<b>ID</b>	os_config_data_install_enforce
-----------	--------------------------------

References	800-53r5	<ul style="list-style-type: none"><li>• SI-2(5)</li><li>• SI-3</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 1.5 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 7.3</li><li>• 7.4</li><li>• 7.7</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95176-4</li></ul>

## 9.8. 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
----	----------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• AC-20</li> <li>• CM-7, CM-7(1)</li> <li>• SC-7(10)</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95180-6</li> </ul>

## 9.9. Integrate System into a Directory Services Infrastructure

The macOS system *MUST* be integrated into a directory services infrastructure.

A directory service infrastructure enables centralized user and rights management, as well as centralized control over computer and user configurations. Integrating the macOS systems used throughout an organization into a directory services infrastructure ensures more administrator oversight and security than allowing distinct user account databases to exist on each separate system.

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

```
/usr/bin/dscl localhost -list . | /usr/bin/grep -qvE '(Contact|Search|Local|^$)';  
/bin/echo $?
```

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

### Remediation Description

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

Integrate the system into an existing directory services infrastructure.

<b>ID</b>	os_directory_services_configured	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 6.7</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95181-4</li> </ul>

## 9.10. 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 OPODs 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.

<b>ID</b>	os_ess_installed	
<b>References</b>	<b>800-53r5</b>	• N/A
	<b>CCE</b>	• CCE-95187-1

## 9.11. 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	<ul style="list-style-type: none"> <li>• AC-2(11)</li> <li>• AC-3</li> <li>• IA-5(13)</li> </ul>
	CIS Benchmark	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	CIS Controls V8	<ul style="list-style-type: none"> <li>• 3.3</li> <li>• 6.7</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>• CCE-95192-1</li> </ul>

## 9.12. 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
```

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	<ul style="list-style-type: none"><li>• CM-14</li><li>• CM-5</li><li>• SI-3</li><li>• SI-7(1), SI-7(15)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.6.5 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 10.1</li><li>• 10.2</li><li>• 10.5</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95195-4</li></ul>

9.13. 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	<ul style="list-style-type: none"><li>• AC-20</li><li>• AC-3</li><li>• CM-7, CM-7(1)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• N/A</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95199-6</li></ul>

## 9.14. Secure User's Home Folders

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

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

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

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

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

### Remediation Description

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

```
IFS=$'\n'
for userDirs in $( /usr/bin/find /System/Volumes/Data/Users -mindepth 1 -maxdepth
```

```
1 -type d ! \( -perm 700 -o -perm 711 \) | /usr/bin/grep -v "Shared" |
/usr/bin/grep -v "Guest" ); do
/bin/chmod og-rwx "$userDirs"
done
unset IFS
```

ID	os_home_folders_secure	
References	<div>800-53r5</div> <div>CIS Benchmark</div> <div>CIS Controls V8</div> <div>CCE</div>	<ul style="list-style-type: none"> <li>AC-6</li> <li>5.1.1 (level 1)</li> <li>3.3</li> <li>CCE-95203-6</li> </ul>

## 9.15. Disable the Built-in Web Server

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



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

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

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

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

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

### Remediation Description

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

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

ID	os_httpd_disable	
References	800-53r5	<ul style="list-style-type: none"><li>• AC-17</li><li>• AC-3</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 4.2 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95204-4</li></ul>

## 9.16. Disable iCloud Storage Setup during Setup Assistant

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

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

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

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

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

### Remediation Description

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

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

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

<b>ID</b>	os_icloud_storage_prompt_disable	
<b>References</b>	<b>800-53r5</b> <ul style="list-style-type: none"> <li>• AC-20</li> </ul>	
	<b>CIS Benchmark</b> <ul style="list-style-type: none"> <li>• N/A</li> </ul>	
	<b>CIS Controls V8</b> <ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> </ul>	
	<b>CCE</b> <ul style="list-style-type: none"> <li>• CCE-95205-1</li> </ul>	

## 9.17. Configure Install.log Retention to 365

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" && $(i+2) >= 365) { ttl="True" }; if ($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=365/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.

<b>ID</b>	os_install_log_retention_configure
-----------	------------------------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• AU-11</li> <li>• AU-4</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 3.3 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 8.1</li> <li>• 8.3</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95211-9</li> </ul>

## 9.18. Disable iPhone Mirroring

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

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

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

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

### Remediation Description

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

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

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

<b>ID</b>	os_iphone_mirroring_disable
-----------	-----------------------------

References	800-53r5	<ul style="list-style-type: none"> <li>• AC-20</li> <li>• AC-3</li> <li>• CM-7, CM-7(1)</li> </ul>
	CIS Benchmark	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	CIS Controls V8	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> <li>• 6.7</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>• CCE-95212-7</li> </ul>

## 9.19. Disable Infrared (IR) support

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

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



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

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

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

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

### Remediation Description

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



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

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

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



<b>ID</b>	os_ir_support_disable	
<b>References</b>	<b>800-53r5</b> <ul style="list-style-type: none"> <li>• AC-18</li> <li>• CM-7, CM-7(1)</li> </ul>	
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> <li>• 12.6</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95213-5</li> </ul>

## 9.20. Enable Library Validation

Library validation *MUST* be enabled.

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

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

```
<key>DisableLibraryValidation</key>
<false/>
```

<b>ID</b>	os_library_validation_enabled
-----------	-------------------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 2.3</li> <li>• 2.6</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95215-0</li> </ul>

## 9.21. 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.

<b>ID</b>	os_mdm_require	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• CM-2</li> <li>• CM-6</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 5.1</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95227-5</li> </ul>

## 9.22. Enable Apple Mobile File Integrity

Mobile file integrity *MUST* be enabled.

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

```
/usr/sbin/nvram -p | /usr/bin/grep -c "amfi_get_out_of_my_way=1"
```

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/nvram boot-args=""
```

<b>ID</b>	os_mobile_file_integrity_enable	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 5.1.3 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 2.3</li> <li>• 2.6</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95231-7</li> </ul>

## 9.23. Disable Network File System Service

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

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

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

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

### Remediation Description

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

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

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

ID	os_nfsd_disable	
References	800-53r5	<ul style="list-style-type: none"> <li>• AC-17</li> <li>• AC-3</li> </ul>
	CIS Benchmark	<ul style="list-style-type: none"> <li>• 4.3 (level 1)</li> </ul>
	CIS Controls V8	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>• CCE-95235-8</li> </ul>

## 9.24. 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	<ul style="list-style-type: none"><li>• AC-20</li><li>• CM-7, CM-7(1)</li><li>• SC-7(10)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.18.1 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95247-3</li></ul>

## 9.25. 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
    echo "PASS"
else
    echo "FAIL"
fi
```

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

### Remediation Description

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

```
for u in $( /usr/bin/dscl . -list /Users UniqueID | /usr/bin/awk '$2 > 500 {print
```

```
$1}'); do
/usr/bin/dscl . -delete /Users/$u hint
done
```

ID	os_password_hint_remove	
References	<div>800-53r5</div> <div>CIS Benchmark</div> <div>CIS Controls V8</div> <div>CCE</div>	<ul style="list-style-type: none"> <li>IA-6</li> <li>2.12.1 (level 1)</li> <li>5.2</li> <li>CCE-95250-7</li> </ul>

## 9.26. Disable Proximity Based Password Sharing Requests

Proximity based password sharing requests *MUST* be disabled.

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

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

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

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

### Remediation Description

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

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

```
<key>allowPasswordProximityRequests</key>
<false/>
```

ID	os_password_proximity_disable
----	-------------------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"><li>• IA-5</li></ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"><li>• N/A</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-95251-5</li></ul>

## 9.27. Disable Password Sharing

Password Sharing *MUST* be disabled.

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

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

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

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

**Remediation Description**

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

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

```
<key>allowPasswordSharing</key>
<false/>
```

<b>ID</b>	os_password_sharing_disable	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"><li>• IA-5</li></ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"><li>• N/A</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-95252-3</li></ul>

# 9.28. Disable Power Nap

Power Nap *MUST* be disabled.



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

The following Macs support Power Nap:

- MacBook (Early 2015 and later)
- MacBook Air (Late 2010 and later)
- MacBook Pro (all models with Retina display)
- Mac mini (Late 2012 and later)
- iMac (Late 2012 and later)
- Mac Pro (Late 2013 and later)

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

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

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

**Remediation Description**  
Perform the following to configure the system to meet the requirements:  

```
/usr/bin/pmset -a powernap 0
```

ID	os_power_nap_disable	
References	800-53r5	<ul style="list-style-type: none"><li>• CM-7, CM-7(1)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.10.2 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95260-6</li></ul>



# 9.29. 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 2>/dev/null << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.SetupAssistant.managed')\
.objectForKey('SkipSetupItems').containsObject("Privacy")
EOS
```

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

**Remediation Description**

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

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

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

ID	os_privacy_setup_prompt_disable	
References	800-53r5	• CM-7, CM-7(1)
	CIS Benchmark	• N/A
	CIS Controls V8	• 4.1
		• 4.8
	CCE	• CCE-95267-1

# 9.30. 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)
	CIS Benchmark	• 5.6 (level 1)
	CIS Controls V8	• 5.4
	CCE	• CCE-95282-0

### 9.31. Ensure Advertising Privacy Protection in Safari Is Enabled

Allow privacy-preserving measurement of ad effectiveness *MUST* be enabled in Safari.

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

```
/usr/bin/profiles -P -o stdout | /usr/bin/grep -c  
'"WebKitPreferences.privateClickMeasurementEnabled" = 1' | /usr/bin/awk '{ if ($1 >=  
1) {print "1"} else {print "0"}}'
```

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.Safari) payload type:

```
<key>WebKitPreferences.privateClickMeasurementEnabled</key>
<true/>
```

ID	os_safari_advertising_privacy_protection_enable	
References	800-53r5	• N/A
	CIS Benchmark	• 6.3.6 (level 1)
	CIS Controls V8	• 9.1
	CCE	• CCE-95283-8

## 9.32. Disable Automatic Opening of Safe Files in Safari

Open "safe" files after downloading *MUST* be disabled in Safari.

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

```
/usr/bin/profiles -P -o stdout | /usr/bin/grep -c 'AutoOpenSafeDownloads = 0' |
/usr/bin/awk '{ if ($1 >= 1) {print "1"} else {print "0"}}'
```

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.Safari) payload type:

```
<key>AutoOpenSafeDownloads</key>
<false/>
```

ID	os_safari_open_safe_downloads_disable
----	---------------------------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 6.3.1 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 9.1</li> <li>• 9.6</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95284-6</li> </ul>

## 9.33. Ensure Prevent Cross-site Tracking in Safari Is Enabled

Prevent cross-site tracking *MUST* be enabled in Safari.

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

```
/usr/bin/profiles -P -o stdout | /usr/bin/grep -cE
'"WebKitPreferences.storageBlockingPolicy" = 1|"WebKitStorageBlockingPolicy" =
1|"BlockStoragePolicy" =2' | /usr/bin/awk '{ if ($1 >= 1) {print "1"} else {print
"0"}}'
```

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.Safari) payload type:

```
<key>WebKitPreferences.storageBlockingPolicy</key>
<integer>1</integer>
<key>WebKitStorageBlockingPolicy</key>
<integer>1</integer>
<key>BlockStoragePolicy</key>
<integer>2</integer>
```

<b>ID</b>	os_safari_prevent_cross-site_tracking_enable
-----------	--

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 6.3.4 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 9.1</li> <li>• 9.3</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95285-3</li> </ul>

## 9.34. Ensure Show Full Website Address in Safari Is Enabled

Show full website address *MUST* be enabled in Safari.

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

```
/usr/bin/profiles -P -o stdout | /usr/bin/grep -c 'ShowFullURLInSmartSearchField = 1'
| /usr/bin/awk '{ if ($1 >= 1) {print "1"} else {print "0"}}'
```

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.Safari) payload type:

```
<key>ShowFullURLInSmartSearchField</key>
<true/>
```

<b>ID</b>	os_safari_show_full_website_address_enable	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 6.3.7 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 9.1</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95287-9</li> </ul>

## 9.35. Ensure Show Safari shows the Status Bar is Enabled

Safari *MUST* be configured to show the status bar.

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

```
/usr/bin/profiles -P -o stdout | /usr/bin/grep -c 'ShowOverlayStatusBar = 1' |  
/usr/bin/awk '{ if ($1 >= 1) {print "1"} else {print "0"}}'
```

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.Safari) payload type:

```
<key>ShowOverlayStatusBar</key>  
<true/>
```

ID	os_safari_show_status_bar_enabled	
References	800-53r5	• N/A
	CIS Benchmark	• 6.3.10 (level 1)
	CIS Controls V8	• 9.1
	CCE	• CCE-95288-7

## 9.36. Ensure Warn When Visiting A Fraudulent Website in Safari Is Enabled

Warn when visiting a fraudulent website *MUST* be enabled in Safari.

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

```
/usr/bin/profiles -P -o stdout | /usr/bin/grep -c 'WarnAboutFraudulentWebsites = 1' |  
/usr/bin/awk '{ if ($1 >= 1) {print "1"} else {print "0"}}'
```

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.Safari) payload type:

```
<key>WarnAboutFraudulentWebsites</key>
<true/>
```

ID	os_safari_warn_fraudulent_website_enable	
References	800-53r5	• N/A
	CIS Benchmark	• 6.3.3 (level 1)
	CIS Controls V8	• 9.1
		• 9.3
	CCE	• CCE-95289-5

## 9.37. Enforce FileVault in Setup Assistant

FileVault *MUST* be enforced in Setup Assistant.

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):

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.MCX.FileVault2')\
.objectForKey('ForceEnableInSetupAssistant')
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.MCX.FileVault2) payload type:

```
<key>ForceEnableInSetupAssistant</key>
```

<true/>

ID	os_setup_assistant_filevault_enforce	
References	800-53r5	<ul style="list-style-type: none"><li>• SC-28, SC-28(1)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• N/A</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 3.6</li><li>• 3.11</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95296-0</li></ul>

## 9.38. 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 reenble "System Integrity Protection", boot the affected system into "Recovery" mode, launch "Terminal" from the "Utilities" menu, and run the command.



<b>ID</b>	os_sip_enable	
<b>References</b>	<b>800-53r5</b> <ul style="list-style-type: none"> <li>• AC-3</li> <li>• AU-9, AU-9(3)</li> <li>• CM-5, CM-5(6)</li> <li>• SC-4</li> <li>• SI-2</li> <li>• SI-7</li> </ul>	
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 5.1.2 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 2.3</li> <li>• 2.6</li> <li>• 10.5</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95298-6</li> </ul>

## 9.39. Disable Siri Setup during Setup Assistant

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

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

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

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

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

### Remediation Description

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

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

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

```
</array>
```

ID	os_siri_prompt_disable	
References	800-53r5	<ul style="list-style-type: none"><li>• AC-20</li><li>• CM-7, CM-7(1)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• N/A</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95299-4</li></ul>

## 9.40. Disable Apple Intelligence During Setup Assistant

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

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

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

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

### Remediation Description

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

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

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

ID	os_skip_apple_intelligence_enable
----	-----------------------------------

References	800-53r5	<ul style="list-style-type: none"><li>• AC-20</li><li>• AC-4</li><li>• CM-7</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• N/A</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.1</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95603-7</li></ul>

## 9.41. Disable Unlock with Apple Watch During Setup Assistant

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

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

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

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

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

**Remediation Description**

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

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

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

ID	os_skip_unlock_with_watch_enable
----	----------------------------------

References	800-53r5	<ul style="list-style-type: none"> <li>• AC-20</li> </ul>
	CIS Benchmark	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	CIS Controls V8	<ul style="list-style-type: none"> <li>• 4.1</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>• CCE-95301-8</li> </ul>

## 9.42. Ensure Sleep and Display Sleep Is Enabled on Apple Silicon Devices

Apple Silicon MacBooks should set sleep timeout to 15 minutes (900 seconds) or less and the display sleep timeout should be 10 minutes (600 seconds) or less but less than the sleep setting.

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

```
error_count=0
if /usr/sbin/ioreg -rd1 -c IOPlatformExpertDevice 2>&1 | /usr/bin/grep -q "MacBook";
then
    sleepMode=$(/usr/bin/pmset -b -g | /usr/bin/grep '^s*sleep' 2>&1 | /usr/bin/awk
'{print $2}')
    displaysleepMode=$(/usr/bin/pmset -b -g | /usr/bin/grep displaysleep 2>&1 |
/usr/bin/awk '{print $2}')

    if [[ "$sleepMode" == "" ]] || [[ "$sleepMode" -gt 15 ]]; then
        ((error_count++))
    fi
    if [[ "$displaysleepMode" == "" ]] || [[ "$displaysleepMode" -gt 10 ]] || [[
"$displaysleepMode" -gt "$sleepMode" ]]; then
        ((error_count++))
    fi
fi
echo "$error_count"
```

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

### Remediation Description

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

```
/usr/bin/pmset -a sleep 15
/usr/bin/pmset -a displaysleep 10
```

ID	os_sleep_and_display_sleep_apple_silicon_enable
----	---

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"><li>• N/A</li></ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"><li>• 2.10.1.2 (level 2)</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 4.1</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-95302-6</li></ul>

## 9.43. Enforce Software Update App Update Updates Automatically

Software Update *MUST* be configured to enforce automatic updates of App Updates is enabled.

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('AutomaticallyInstallAppUpdates').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>AutomaticallyInstallAppUpdates</key>
<true/>
```

<b>ID</b>	os_software_update_app_update_enforce	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"><li>• N/A</li></ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"><li>• 1.5 (level 1)</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 7.3</li><li>• 7.4</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-95402-4</li></ul>

## 9.44. 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)
	CIS Benchmark	• 5.11 (level 1)
	CIS Controls V8	• N/A
	CCE	• CCE-95316-6

## 9.45. 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	<ul style="list-style-type: none"><li>• N/A</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 5.4 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.3</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95317-4</li></ul>

## 9.46. 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
----	-------------------------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• CM-5(1)</li> <li>• IA-11</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 5.5 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.3</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95318-2</li> </ul>

## 9.47. Ensure Appropriate Permissions Are Enabled for System Wide Applications

Applications in the System Applications Directory (/Applications) *MUST* not be world-writable.

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

```
/usr/bin/find /Applications -iname "*.app" -type d -perm -2 -ls | /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 apps in $( /usr/bin/find /Applications -iname "*.app" -type d -perm -2 ); do
  /bin/chmod -R o-w "$apps"
done
```

<b>ID</b>	os_system_wide_applications_configure	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 5.1.5 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 3.3</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95320-8</li> </ul>



# 9.48. Ensure Secure Keyboard Entry Terminal.app is Enabled

Secure keyboard entry *MUST* be enabled in Terminal.app.

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

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

```
<key>SecureKeyboardEntry</key>
<true/>
```

ID	os_terminal_secure_keyboard_enable	
References	800-53r5	• N/A
	CIS Benchmark	• 6.4.1 (level 1)
	CIS Controls V8	• 4.8
	CCE	• CCE-95321-6

# 9.49. 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):

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

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

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

**Remediation Description**

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

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

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

ID	os_tftpd_disable	
References	800-53r5	<ul style="list-style-type: none"><li>• AC-17</li><li>• AC-3</li><li>• IA-5(1)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• N/A</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 3.3</li><li>• 3.1</li><li>• 5.2</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95323-2</li></ul>

## 9.50. 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 print system | /usr/bin/grep -c -E '\tcom.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	<ul style="list-style-type: none"><li>• AU-12(1)</li><li>• SC-45(1)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.3.2.2 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 8.4</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95325-7</li></ul>

## 9.51. 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 2>/dev/null << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.SetupAssistant.managed')\
.objectForKey('SkipSetupItems').containsObject("Biometric")
EOS
```

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

**Remediation Description**

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

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


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

ID	os_touchid_prompt_disable	
References	800-53r5	• CM-6
	CIS Benchmark	• N/A
	CIS Controls V8	• 4.1
	CCE	• CCE-95326-5


## 9.52. Disable Login to Other User’s Active and Locked Sessions

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

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



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



This rule may cause issues when platformSSO is configured.

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

RESULT="FAIL"

```

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

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

echo $RESULT

```

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

### Remediation Description

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

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

```

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

```

ID	os_unlock_active_user_session_disable	
References	800-53r5	<ul style="list-style-type: none"> <li>IA-2, IA-2(5)</li> </ul>
	CIS Benchmark	<ul style="list-style-type: none"> <li>5.7 (level 1)</li> </ul>
	CIS Controls V8	<ul style="list-style-type: none"> <li>4.3</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>CCE-95328-1</li> </ul>

## 9.53. 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):

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

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

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

### Remediation Description

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

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

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

ID	os_uucp_disable	
References	800-53r5	<ul style="list-style-type: none"><li>• AC-17</li><li>• AC-3</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• N/A</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 3.3</li><li>• 4.1</li><li>• 4.8</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95330-7</li></ul>

## 9.54. Ensure No World Writable Files Exist in the Library Folder

Folders in /System/Volumes/Data/Library *MUST* not be world-writable.



Some vendors are known to create world-writable folders to the System Library folder. You may need to add more exclusions to this check and fix to match your environment.

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

```
/usr/bin/find /Library -type d -perm -002 ! -perm -1000 ! -xattrname  
com.apple.rootless 2>/dev/null | /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 libPermissions in $(/usr/bin/find /Library -type d -perm -002 ! -perm -1000 !  
-xattrname com.apple.rootless 2>/dev/null); do  
  /bin/chmod -R o-w "$libPermissions"  
done
```

ID	os_world_writable_library_folder_configure	
References	800-53r5	• N/A
	CIS Benchmark	• 5.1.7 (level 2)
	CIS Controls V8	• 3.3
	CCE	• CCE-95332-3

## 9.55. Ensure No World Writable Files Exist in the System Folder

Folders in /System/Volumes/Data/System *MUST* not be world-writable.

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

```
/usr/bin/find /System/Volumes/Data/System -type d -perm -2 -ls | /usr/bin/grep -vE "downloadDir|locks" | /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 sysPermissions in $( /usr/bin/find /System/Volumes/Data/System -type d -perm -2 | /usr/bin/grep -vE "downloadDir|locks" ); do
  /bin/chmod -R o-w "$sysPermissions"
done
```

ID	os_world_writable_system_folder_configure	
References	800-53r5	<ul style="list-style-type: none"><li>N/A</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>5.1.6 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>3.3</li></ul>
	CCE	<ul style="list-style-type: none"><li>CCE-95333-1</li></ul>



# 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.

<b>ID</b>	pwpolicy_account_inactivity_enforce	
<b>References</b>	<b>800-53r5</b> <ul style="list-style-type: none"> <li>• AC-2(3)</li> </ul>	
	<b>CIS Benchmark</b> <ul style="list-style-type: none"> <li>• N/A</li> </ul>	
	<b>CIS Controls V8</b> <ul style="list-style-type: none"> <li>• 5.3</li> </ul>	
	<b>CCE</b> <ul style="list-style-type: none"> <li>• CCE-95336-4</li> </ul>	

## 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 "pass"} else
{print "fail"}}' | /usr/bin/uniq
```

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

### Remediation Description

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

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

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

ID	pwpolicy_account_lockout_enforce	
References	800-53r5	• AC-7
	CIS Benchmark	• 5.2.1 (level 1)
	CIS Controls V8	• 6.2
	CCE	• CCE-95337-2

## 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 "pass"} else
{print "fail"}}' | /usr/bin/uniq
```

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

### Remediation Description

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

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

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

<integer>15</integer>

ID	pwpolicy_account_lockout_timeout_enforce	
References	800-53r5	• AC-7
	CIS Benchmark	• 5.2.1 (level 1)
	CIS Controls V8	• 6.2
	CCE	• CCE-95338-0

## 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, “Improving the Nation’s Cybersecurity”, OMB M-22-09, “Moving the U.S. Government Toward Zero Trust Cybersecurity Principles”, and NIST SP-800-63b, “Digital Identity Guidelines: Authentication and Lifecycle Management” federal, military, and intelligence communities must adopt the following configuration settings. Password policies must not require the use of complexity policies such as upper characters, lower characters, or special characters. Password policies must also not require the use of regular rotation. 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)
	CIS	• 5.2.3 (level 2)
	Benchmark	• 5.2.4 (level 2)
	CIS Controls V8	• 5.2
	CCE	• CCE-95339-8

## 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, “Improving the Nation’s Cybersecurity”, OMB M-22-09, “Moving the U.S. Government Toward Zero Trust Cybersecurity Principles”, and NIST SP-800-63b, “Digital Identity Guidelines: Authentication and Lifecycle Management” federal, military, and intelligence communities must adopt the following configuration settings. Password policies must not require the use of complexity policies such as upper characters, lower characters, or special characters. Password policies must also not require the use of regular rotation. 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>
  <key>passwordContentRegex</key>
  <string>^(?=.*[A-Z])(?=.*[a-z])(?=.*[0-9]).*$</string>
  <key>passwordContentDescription</key>
  <dict>
    <key>default</key>
    <string>Password must match custom regex.</string>
  </dict>
</dict>
```

ID	pwpolicy_custom_regex_enforce	
References	800-53r5	• IA-5(1)
	CIS Benchmark	• 5.2.6 (level 2)
	CIS Controls V8	• 5.2
	CCE	• CCE-95340-6

## 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 "pass"} else {print
"fail"}}' | /usr/bin/uniq
```

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

### Remediation Description

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

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

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

ID	pwpolicy_history_enforce	
References	800-53r5	<ul style="list-style-type: none"> <li>IA-5(1)</li> </ul>
	CIS Benchmark	<ul style="list-style-type: none"> <li>5.2.8 (level 1)</li> </ul>
	CIS Controls V8	<ul style="list-style-type: none"> <li>5.2</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>CCE-95343-0</li> </ul>

## 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, “Improving the Nation’s Cybersecurity”, OMB M-22-09, “Moving the U.S. Government Toward Zero Trust Cybersecurity Principles”, and NIST SP-800-63b, “Digital Identity Guidelines: Authentication and Lifecycle Management” federal, military, and intelligence communities must adopt the following configuration settings. Password policies must not require the use of complexity policies such as upper characters, lower characters, or special characters. Password policies must also not require the use of regular rotation. 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()'
- | /usr/bin/awk '{ if ($1 <= 60 ) {print "pass"} else {print "fail"}}' |
/usr/bin/uniq
```

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

### Remediation Description

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

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

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

ID	pwpolicy_max_lifetime_enforce	
References	800-53r5	<ul style="list-style-type: none"> <li>IA-5</li> </ul>
	CIS Benchmark	<ul style="list-style-type: none"> <li>5.2.7 (level 1)</li> </ul>
	CIS Controls V8	<ul style="list-style-type: none"> <li>5.3</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>CCE-95345-5</li> </ul>

## 10.8. Require a Minimum Password Length of 15 Characters

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

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



To comply with Executive Order 14028, “Improving the Nation’s Cybersecurity”, OMB M-22-09, “Moving the U.S. Government Toward Zero Trust Cybersecurity Principles”, and NIST SP-800-63b, “Digital Identity Guidelines: Authentication and Lifecycle Management” federal, military, and intelligence communities must adopt the following configuration settings. Password policies must not require the use of complexity policies such as upper characters, lower characters, or special characters. Password policies must also not require the use of regular rotation.



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 | tail +2 | grep -oE  
"policyAttributePassword matches '.\{[0-9]+\,' | awk -F'[{}]' -v ODV=15 '{if ($2 > max)  
max=$2} END {print (max >= ODV) ? "pass" : "fail"}'
```

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

### Remediation Description

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

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

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

ID	pwpolicy_minimum_length_enforce	
References	800-53r5	• IA-5(1)
	CIS Benchmark	• 5.2.2 (level 1)
	CIS Controls V8	• 5.2
	CCE	• CCE-95346-3

## 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, “Improving the Nation’s Cybersecurity”, OMB M-22-09, “Moving the U.S. Government Toward Zero Trust Cybersecurity Principles”, and NIST SP-800-63b, “Digital Identity Guidelines: Authentication and Lifecycle Management” federal, military, and intelligence communities must adopt the following configuration settings. Password policies must not require the use of complexity policies such as upper characters, lower characters, or special characters. Password policies must also not require the use of regular rotation.

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 "pass"} else  
{print "fail"}}'
```

If the result is not **pass**, 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
----	-----------------------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• IA-5</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.7</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95347-1</li> </ul>

## 10.10. Prohibit Repeating, Ascending, and Descending Character Sequences

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

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



To comply with Executive Order 14028, “Improving the Nation’s Cybersecurity”, OMB M-22-09, “Moving the U.S. Government Toward Zero Trust Cybersecurity Principles”, and NIST SP-800-63b, “Digital Identity Guidelines: Authentication and Lifecycle Management” federal, military, and intelligence communities must adopt the following configuration settings. Password policies must not require the use of complexity policies such as upper characters, lower characters, or special characters. Password policies must also not require the use of regular rotation. Password policies should define a minimum length. Multifactor authentication should be used where ever possible.



`pwpolicy_simple_sequence_disable` prevents use of passwords which are regularly found in compromised password lists.

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

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

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

### Remediation Description

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

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

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

ID	pwpolicy_simple_sequence_disable	
References	800-53r5	• IA-5(1)
	CIS Benchmark	• N/A
	CIS Controls V8	• 5.2
	CCE	• CCE-95349-7

## 10.11. Require Passwords Contain a Minimum of One Special Character

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

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

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



To comply with Executive Order 14028, “Improving the Nation’s Cybersecurity”, OMB M-22-09, “Moving the U.S. Government Toward Zero Trust Cybersecurity Principles”, and NIST SP-800-63b, “Digital Identity Guidelines: Authentication and Lifecycle Management” federal, military, and intelligence communities must adopt the following configuration settings. Password policies must not require the use of complexity policies such as upper characters, lower characters, or special characters. Password policies must also not require the use of regular rotation. 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
"pass"} else {print "fail"}}'
```

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

Remediation Description

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

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

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

ID	pwpolicy_special_character_enforce	
References	800-53r5	• IA-5(1)
	CIS Benchmark	• 5.2.5 (level 2)
	CIS Controls V8	• 5.2
	CCE	• CCE-95350-5

# 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
----	--

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• CM-7, CM-7(1)</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 2.3.1.2 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95354-7</li> </ul>

## 11.2. 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/>
```

<b>ID</b>	system_settings_automatic_login_disable
-----------	---

References	800-53r5	<ul style="list-style-type: none"> <li>• IA-2</li> <li>• IA-5(13)</li> </ul>
	CIS Benchmark	<ul style="list-style-type: none"> <li>• 2.13.3 (level 1)</li> </ul>
	CIS Controls V8	<ul style="list-style-type: none"> <li>• 4.7</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>• CCE-95356-2</li> </ul>

## 11.3. 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/>
```



<b>ID</b>	system_settings_bluetooth_disable	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• AC-18, AC-18(3)</li> <li>• SC-8</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.8</li> <li>• 12.6</li> <li>• 13.9</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95358-8</li> </ul>

## 11.4. Enable Bluetooth Menu

The bluetooth menu *MUST* be enabled.

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

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

If the result is not **18**, 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.controlcenter) payload type:

```
<key>Bluetooth</key>
<integer>18</integer>
```

<b>ID</b>	system_settings_bluetooth_menu_enable
-----------	---------------------------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"><li>• N/A</li></ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"><li>• N/A</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 4.8</li><li>• 13.9</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-95359-6</li></ul>

## 11.5. 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 '//*[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>
  <string>com.apple.BluetoothSettings</string>
</array>
```

<b>ID</b>	system_settings_bluetooth_settings_disable	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"><li>• CM-7, CM-7(1)</li></ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"><li>• N/A</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-95360-4</li></ul>

# 11.6. Disable Bluetooth Sharing

Bluetooth Sharing *MUST* be disabled.

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



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

```
CURRENT_USER=$( /usr/bin/defaults read
/Library/Preferences/com.apple.loginwindow lastUserName )
```

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

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

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

## Remediation Description

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

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

ID	system_settings_bluetooth_sharing_disable	
References	800-53r5	<ul style="list-style-type: none"><li>AC-18(4)</li><li>AC-3</li><li>CM-7, CM-7(1)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>2.3.3.10 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>3.3</li><li>4.1</li></ul>
	CCE	<ul style="list-style-type: none"><li>CCE-95361-2</li></ul>

# 11.7. 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)
	CIS Benchmark	• 2.3.3.8 (level 2)
	CIS Controls V8	• 4.8
	CCE	• CCE-95362-0

# 11.8. Enforce Critical Security Updates to be Installed

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

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

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

```
.objectForKey('CriticalUpdateInstall').js  
EOS
```

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

### Remediation Description

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

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

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

ID	system_settings_critical_update_install_enforce	
References	800-53r5	<ul style="list-style-type: none"><li>• SI-2</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 1.5 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 7.3</li><li>• 7.4</li><li>• 7.7</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95363-8</li></ul>

## 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 ){
```

```
        return("true")
    } else {
        return("false")
    }
}
EOS
```

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

### Remediation Description

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

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

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

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

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

ID	system_settings_diagnostics_reports_disable	
References	<b>800-53r5</b>	<ul style="list-style-type: none"><li>• AC-20</li><li>• SC-7(10)</li><li>• SI-11</li></ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"><li>• 2.6.3.1 (level 1)</li><li>• 2.6.3.4 (level 1)</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-95364-6</li></ul>

## 11.10. Enforce Software Update Downloads Updates Automatically using DDM.

Software Update *MUST* be configured to enforce automatic downloads of updates from Apple and that the user cannot modify the setting within System Settings.

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

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

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

Remediation Description

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

This is implemented by Declarative Device Management (DDM).

ID	system_settings_download_software_update_enforce	
References	800-53r5	• N/A
	CIS Benchmark	• N/A
	CIS Controls V8	• 7.3
		• 7.4
	CCE	• CCE-95403-2

# 11.11. Disable External Intelligence Integrations

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

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

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

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

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

Remediation Description

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

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

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

ID	system_settings_external_intelligence_disable	
References	800-53r5	<ul style="list-style-type: none"><li>• AC-20</li><li>• CM-7, CM-7(1)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.5.1.1 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li><li>• 15.3</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95365-3</li></ul>

# 11.12. Disable External Intelligence Integration Sign In

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

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

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

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

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

**Remediation Description**

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

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



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

ID	system_settings_external_intelligence_sign_in_disable	
References	800-53r5	<ul style="list-style-type: none"><li>• AC-20</li><li>• CM-7, CM-7(1)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.5.1.1 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li><li>• 15.3</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95366-1</li></ul>

## 11.13. 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.



See the FileVault supplemental to implement this rule.

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

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

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

### Remediation Description

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

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

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

ID	system_settings_filevault_enforce	
References	800-53r5	• SC-28, SC-28(1)
	CIS Benchmark	• 2.6.6 (level 1)
	CIS Controls V8	• 3.6
	CCE	• 3.11
		• CCE-95367-9

## 11.14. Disable Find My Service

The Find My service *MUST* be disabled.

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

Apple's Find My service uses a personal AppleID for authentication. Organizations should rely on MDM solutions, which have much more secure authentication requirements, to perform remote lock and remote wipe.

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

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

ID	system_settings_find_my_disable	
References	800-53r5	<ul style="list-style-type: none"> <li>• AC-20</li> <li>• CM-7, CM-7(1)</li> </ul>
	CIS Benchmark	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	CIS Controls V8	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> <li>• 15.3</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>• CCE-95368-7</li> </ul>

## 11.15. 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	<ul style="list-style-type: none"><li>• AC-4</li><li>• CM-7, CM-7(1)</li><li>• SC-7, SC-7(12)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.2.1 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.5</li><li>• 13.1</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95369-5</li></ul>

# 11.16. Enable Firewall Stealth Mode

Firewall Stealth Mode *MUST* be enabled.

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



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

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

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

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

### Remediation Description

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

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

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

ID	system_settings_firewall_stealth_mode_enable	
References	800-53r5	<ul style="list-style-type: none"> <li>• CM-7, CM-7(1)</li> <li>• SC-7, SC-7(16)</li> </ul>
	CIS Benchmark	<ul style="list-style-type: none"> <li>• 2.2.2 (level 1)</li> </ul>
	CIS Controls V8	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.5</li> <li>• 4.8</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>• CCE-95370-3</li> </ul>

## 11.17. Disable Guest Access to Shared SMB Folders

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

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

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

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

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

### Remediation Description

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

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

ID	system_settings_guest_access_smb_disable	
References	800-53r5	• AC-2, AC-2(9)
	CIS Benchmark	• 2.13.2 (level 1)
	CIS Controls V8	• 3.3
	CCE	• CCE-95373-7

## 11.18. Disable the Guest Account

Guest access *MUST* be disabled.

Turning off guest access prevents anonymous users from accessing files.

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

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

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

### Remediation Description

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

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

type:

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

ID	system_settings_guest_account_disable	
References	800-53r5	<ul style="list-style-type: none"><li>• AC-2, AC-2(9)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.13.1 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 5.2</li><li>• 6.2</li><li>• 6.8</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95374-5</li></ul>

## 11.19. Secure Hot Corners

Hot corners *MUST* be secured.

The information system conceals, via the session lock, information previously visible on the display with a publicly viewable image. Although hot corners 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.



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

```
CURRENT_USER=$( /usr/bin/defaults read
/Library/Preferences/com.apple.loginwindow lastUserName )
```

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

```
bl_corner="$(/usr/bin/defaults read /Users/"$CURRENT_USER
/Library/Preferences/com.apple.dock wvous-bl-corner 2>/dev/null)"
tl_corner="$(/usr/bin/defaults read /Users/"$CURRENT_USER
/Library/Preferences/com.apple.dock wvous-tl-corner 2>/dev/null)"
tr_corner="$(/usr/bin/defaults read /Users/"$CURRENT_USER
/Library/Preferences/com.apple.dock wvous-tr-corner 2>/dev/null)"
br_corner="$(/usr/bin/defaults read /Users/"$CURRENT_USER
/Library/Preferences/com.apple.dock wvous-br-corner 2>/dev/null)"
```

```
if [[ "$bl_corner" != "6" ]] && [[ "$tl_corner" != "6" ]] && [[ "$tr_corner" != "6" ]]
&& [[ "$br_corner" != "6" ]]; then
    echo "0"
fi
```

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 "$CURRENT_USER" /usr/bin/defaults delete /Users/"$CURRENT_USER"
/Library/Preferences/com.apple.dock wvous-bl-corner 2>/dev/null
/usr/bin/sudo -u "$CURRENT_USER" /usr/bin/defaults delete /Users/"$CURRENT_USER"
/Library/Preferences/com.apple.dock wvous-tl-corner 2>/dev/null
/usr/bin/sudo -u "$CURRENT_USER" /usr/bin/defaults delete /Users/"$CURRENT_USER"
/Library/Preferences/com.apple.dock wvous-tr-corner 2>/dev/null
/usr/bin/sudo -u "$CURRENT_USER" /usr/bin/defaults delete /Users/"$CURRENT_USER"
/Library/Preferences/com.apple.dock wvous-br-corner 2>/dev/null
```

<b>ID</b>	system_settings_hot_corners_secure	
<b>References</b>	<b>800-53r5</b>	• AC-11(1)
	<b>CIS Benchmark</b>	• 2.7.1 (level 2)
	<b>CIS Controls V8</b>	• 4.3
	<b>CCE</b>	• CCE-95376-0

## 11.20. 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	<ul style="list-style-type: none"><li>• AC-20</li><li>• CM-7, CM-7(1)</li><li>• SC-7(10)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.6.3.3 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95377-8</li></ul>

## 11.21. 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	<ul style="list-style-type: none"><li>• AC-20</li><li>• CM-7, CM-7(1)</li><li>• SC-7(10)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.9.1</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95378-6</li></ul>

11.22. 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	<ul style="list-style-type: none"><li>• AC-20</li><li>• CM-7, CM-7(1)</li><li>• SC-7(10)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.6.3.2 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95379-4</li></ul>

## 11.23. Enforce macOS Updates are Automatically Installed

Software Update *MUST* be configured to enforce automatic installation of macOS updates is enabled.

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('AutomaticallyInstallMacOSUpdates').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>AutomaticallyInstallMacOSUpdates</key>
```

<true/>

ID	system_settings_install_macos_updates_enforce	
References	800-53r5	• N/A
	CIS Benchmark	• 1.3 (level 1)
	CIS Controls V8	• 7.3
		• 7.4
	CCE	• CCE-95380-2

## 11.24. Disable the Internet Accounts System Preference Pane

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



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

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

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

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

### Remediation Description

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

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

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

</array>

ID	system_settings_internet_accounts_disable	
References	800-53r5	<ul style="list-style-type: none"><li>• AC-20</li><li>• CM-7, CM-7(1), CM-7(5)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• N/A</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.8</li><li>• 15.2</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95381-0</li></ul>

## 11.25. 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
----	--

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"><li>• AC-20</li><li>• AC-4</li></ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"><li>• 2.3.3.7 (level 1)</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-95382-8</li></ul>

# 11.26. Enable Location Services

Location Services *MUST* be enabled.

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 **true**, 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 true;
pid=$(/bin/launchctl print system | /usr/bin/awk '/\tcom.apple.locationd/ {print $1}')
kill -9 $pid
```

<b>ID</b>	system_settings_location_services_enable	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"><li>• N/A</li></ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"><li>• 2.6.1.1 (level 2)</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-95384-4</li></ul>

# 11.27. Configure Login Window to Show A Custom Message

The login window *MUST* be configured to show a custom access warning message.

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

```
/usr/bin/osascript -l JavaScript << EOS | /usr/bin/base64
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.loginwindow')\
.objectForKey('LoginwindowText').js
EOS
```

If the result is not **Center for Internet Security Test Message**, 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>LoginwindowText</key>
<string>Center for Internet Security Test Message</string>
```

ID	system_settings_loginwindow_loginwindowtext_enable	
References	800-53r5	• N/A
	CIS Benchmark	• 2.11.3 (level 1)
	CIS Controls V8	• 4.1
	CCE	• CCE-95386-9

# 11.28. 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
	CIS Benchmark	• 2.11.4 (level 1)
	CIS Controls V8	• 4.1
	CCE	• CCE-95387-7

## 11.29. Disable Media Sharing

Media sharing *MUST* be disabled.

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

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

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

```
/usr/bin/osascript -l JavaScript << EOS
function run() {
  let pref1 = ObjC.unwrap(
```



```
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowMediaSharing'))
let pref2 = ObjC.unwrap(
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('allowMediaSharingModification'))
if ( pref1 == false && pref2 == false ) {
    return("true")
} else {
    return("false")
}
}
EOS
```

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

### Remediation Description

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

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

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

<b>ID</b>	system_settings_media_sharing_disabled	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• AC-17</li> <li>• AC-3</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 2.3.3.9 (level 2)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95388-5</li> </ul>

## 11.30. 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
	CIS Benchmark	• 2.11.5 (level 1)
	CIS Controls V8	• 4.1
	CCE	• CCE-95389-3

# 11.31. 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	<ul style="list-style-type: none"><li>• AC-20</li><li>• CM-7, CM-7(1)</li><li>• SC-7(10)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.6.4 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.8</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95390-1</li></ul>

## 11.32. 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
----	---

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"><li>• CM-7, CM-7(1)</li></ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"><li>• 2.3.3.3 (level 1)</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-95391-9</li></ul>

## 11.33. 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.

<b>ID</b>	system_settings_rae_disable
-----------	-----------------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• AC-17</li> <li>• AC-3</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 2.3.3.6 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95392-7</li> </ul>

## 11.34. Disable Remote Management

Remote Management *MUST* be disabled.

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

```
/usr/libexec/mdmclient QuerySecurityInfo 2>/dev/null | /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/kick
start -deactivate -stop
```

<b>ID</b>	system_settings_remote_management_disable	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• CM-7, CM-7(1)</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 2.3.3.5 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> <li>• 5.4</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95393-5</li> </ul>

## 11.35. Disable Screen Sharing and Apple Remote Desktop

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

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

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

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

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

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

### Remediation Description

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

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

NOTE - This will apply to the whole system

ID	system_settings_screen_sharing_disable
----	--

References	800-53r5	<ul style="list-style-type: none"> <li>• AC-17</li> <li>• AC-3</li> </ul>
	CIS Benchmark	<ul style="list-style-type: none"> <li>• 2.3.3.1 (level 1)</li> </ul>
	CIS Controls V8	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>• CCE-95394-3</li> </ul>

## 11.36. Enforce Session Lock After Screen Saver is Started

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

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

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

```
/usr/bin/osascript -l JavaScript << EOS
function run() {
  let delay = ObjC.unwrap(
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.screensaver')\
.objectForKey('askForPasswordDelay'))
  if ( delay <= 5 ) {
    return("true")
  } else {
    return("false")
  }
}
EOS
```

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

### Remediation Description

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

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

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

<b>ID</b>	system_settings_screensaver_ask_for_password_delay_enforce	
<b>References</b>	<b>800-53r5</b> <ul style="list-style-type: none"> <li>• AC-11</li> </ul>	
	<b>CIS Benchmark</b> <ul style="list-style-type: none"> <li>• 2.11.2 (level 1)</li> </ul>	
	<b>CIS Controls V8</b> <ul style="list-style-type: none"> <li>• 4.7</li> </ul>	
	<b>CCE</b> <ul style="list-style-type: none"> <li>• CCE-95395-0</li> </ul>	

## 11.37. Enforce Screen Saver Timeout

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

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

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

```
/usr/bin/osascript -l JavaScript << EOS
function run() {
  let timeout = ObjC.unwrap(
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.screensaver')\
.objectForKey('idleTime'))
  if ( timeout <= 1200 ) {
    return("true")
  } else {
    return("false")
  }
}
EOS
```

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

### Remediation Description

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

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

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

<b>ID</b>	system_settings_screensaver_timeout_enforce
-----------	---



<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"><li>• AC-11</li><li>• IA-11</li></ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"><li>• 2.11.1 (level 1)</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 4.3</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-95397-6</li></ul>

## 11.38. Enforce Automatic Installs of Available Security Updates using DDM.

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

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

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

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

**Remediation Description**

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

This is implemented by Declarative Device Management (DDM).

<b>ID</b>	system_settings_security_update_install	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"><li>• SI-2</li></ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"><li>• N/A</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 7.3</li><li>• 7.4</li><li>• 7.7</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-95602-9</li></ul>

# 11.39. 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	<ul style="list-style-type: none"><li>• AC-20</li><li>• CM-7, CM-7(1)</li><li>• SC-7(10)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.5.2.1 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95398-4</li></ul>

# 11.40. Ensure Siri Listen For is Disabled

Siri has the ability to listen for "Hey Siri" or "Siri". Listen for *MUST* be disabled.

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

```
/usr/bin/sudo /usr/bin/osascript -l JavaScript << EOS
```

```
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.Siri')\
.objectForKey('VoiceTriggerUserEnabled').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.Siri) payload type:

```
<key>VoiceTriggerUserEnabled</key>
<false/>
```

ID	system_settings_siri_listen_disable	
References	800-53r5	• N/A
	CIS Benchmark	• N/A
	CIS Controls V8	• 4.1 • 4.8
	CCE	• CCE-95399-2

## 11.41. 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>
  <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)
	CIS Benchmark	• N/A
	CIS Controls V8	• 4.1
		• 4.8
	CCE	• CCE-95400-8

# 11.42. 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	<ul style="list-style-type: none"> <li>• AC-17</li> <li>• AC-3</li> </ul>
	CIS Benchmark	<ul style="list-style-type: none"> <li>• 2.3.3.2 (level 1)</li> </ul>
	CIS Controls V8	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> <li>• 5.4</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>• CCE-95401-6</li> </ul>

## 11.43. Enforce Software Update Downloads Updates Automatically

Software Update *MUST* be configured to enforce automatic downloads of updates is enabled.

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('AutomaticDownload').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>AutomaticDownload</key>
<true/>
```

ID	system_settings_software_update_download_enforce
----	--

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 1.2 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 7.3</li> <li>• 7.4</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95403-2</li> </ul>

## 11.44. Ensure Software Update is Updated and Current

Make sure Software Update is updated and current.



Automatic fix can cause unplanned restarts and may lose work.

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

```
softwareupdate_date_epoch=$(/bin/date -j -f "%Y-%m-%d" "$( /usr/bin/defaults read /Library/Preferences/com.apple.SoftwareUpdate.plist LastFullSuccessfulDate | /usr/bin/awk '{print $1}' )" "+%s")
thirty_days_epoch=$(/bin/date -v -30d "+%s")
if [[ $softwareupdate_date_epoch -lt $thirty_days_epoch ]]; then
  /bin/echo "0"
else
  /bin/echo "1"
fi
```

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/softwareupdate -i -a
```

NOTE - This will apply to the whole system

<b>ID</b>	system_settings_softwareupdate_current
-----------	--

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 1.1 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 7.3</li> <li>• 7.4</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95405-7</li> </ul>

## 11.45. Disable SSH Server for Remote Access Sessions

SSH service *MUST* be disabled for remote access.

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

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

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

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

### Remediation Description

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

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



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

<b>ID</b>	system_settings_ssh_disable
-----------	-----------------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• AC-17</li> <li>• CM-7, CM-7(1)</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 2.3.3.4 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95406-5</li> </ul>

## 11.46. Require Administrator Password to Modify System-Wide Preferences

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

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

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

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



```
done
echo $result
```

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

## Remediation Description

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

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

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

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

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

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

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

```

fi

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

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

```

<b>ID</b>	system_settings_system_wide_preferences_configure	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• AC-6, AC-6(1), AC-6(2)</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 2.6.8 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-95408-1</li> </ul>

## 11.47. Configure Time Machine for Automatic Backups

Automatic backups *MUST* be enabled when using Time Machine.

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

```

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

```

<key>AutoBackup</key>
<true/>

```

<b>ID</b>	system_settings_time_machine_auto_backup_enable	
<b>References</b>	<b>800-53r5</b> <ul style="list-style-type: none"> <li>• N/A</li> </ul>	<b>CIS Benchmark</b> <ul style="list-style-type: none"> <li>• 2.3.4.1 (level 2)</li> </ul>
	<b>CIS Controls V8</b> <ul style="list-style-type: none"> <li>• 11.2</li> </ul>	
	<b>CCE</b> <ul style="list-style-type: none"> <li>• CCE-95409-9</li> </ul>	

## 11.48. Ensure Time Machine Volumes are Encrypted

Time Machine volumes *MUST* be encrypted.

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

```
/usr/bin/sudo /usr/bin/defaults read /Library/Preferences/com.apple.TimeMachine.plist
| grep -c NotEncrypted
```

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

### Remediation Description

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

1. Go to System Settings → Time Machine
2. Click **Select Disk**
3. Select existing Backup Disk under **Available Disks**
4. Click **Encrypt Backups**
5. Click **Use Disk**

<b>ID</b>	system_settings_time_machine_encrypted_configure	
<b>References</b>	<b>800-53r5</b> <ul style="list-style-type: none"> <li>• N/A</li> </ul>	<b>CIS Benchmark</b> <ul style="list-style-type: none"> <li>• 2.3.4.2 (level 1)</li> </ul>
	<b>CIS Controls V8</b> <ul style="list-style-type: none"> <li>• 3.6</li> <li>• 3.11</li> <li>• 11.3</li> </ul>	
	<b>CCE</b> <ul style="list-style-type: none"> <li>• CCE-95410-7</li> </ul>	

## 11.49. 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	<b>800-53r5</b>	<ul style="list-style-type: none"><li>• AU-12(1)</li><li>• SC-45(1)</li></ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"><li>• 2.3.2.1 (level 1)</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 8.4</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-95411-5</li></ul>

## 11.50. 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	<ul style="list-style-type: none"><li>• AU-12(1)</li><li>• SC-45(1)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.3.2.1 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 8.4</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-95412-3</li></ul>

## 11.51. Disable the Touch ID System Settings Pane

The System Settings pane for Touch ID *MUST* be disabled.

Disabling the System Settings pane prevents the users from configuring Touch ID.

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.Touch-ID-Settings.extension"
```

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

### Remediation Description

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

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

```
<key>DisabledSystemSettings</key>
<array>
  <string>com.apple.Touch-ID-Settings.extension</string>
</array>
```

<b>ID</b>	system_settings_touch_id_settings_disable	
<b>References</b>	<b>800-53r5</b> <ul style="list-style-type: none"> <li>• CM-7, CM-7(1), CM-7(5)</li> </ul>	
	<b>CIS Benchmark</b> <ul style="list-style-type: none"> <li>• N/A</li> </ul>	
	<b>CIS Controls V8</b> <ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> </ul>	
	<b>CCE</b> <ul style="list-style-type: none"> <li>• CCE-95414-9</li> </ul>	

## 11.52. Ensure Wake for Network Access Is Disabled

Wake for network access *MUST* be disabled.

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

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

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

### Remediation Description

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

```
/usr/bin/pmset -a womp 0
```

ID	system_settings_wake_network_access_disable	
References	800-53r5	• N/A
	CIS Benchmark	• 2.10.3 (level 1)
	CIS Controls V8	• 4.8
	CCE	• CCE-95417-2

## 11.53. 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>  
  <string>com.apple.WalletSettingsExtension</string>  
</array>
```

ID	system_settings_wallet_applepay_settings_disable
----	--

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"><li>• CM-7, CM-7(1), CM-7(5)</li></ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"><li>• N/A</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-95418-0</li></ul>

# 11.54. Disable Wi-Fi Interface

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

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



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



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

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

**Remediation Description**

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

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

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

<b>ID</b>	system_settings_wifi_disable
-----------	------------------------------



<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"><li>• AC-18, AC-18(1), AC-18(3)</li><li>• AC-4</li></ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"><li>• N/A</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 4.2</li><li>• 12.6</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-95419-8</li></ul>

# 11.55. Enable Wifi Menu

The WiFi menu *MUST* be enabled.

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

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

If the result is not **18**, 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.controlcenter) payload type:

```
<key>WiFi</key>
<integer>18</integer>
```

<b>ID</b>	system_settings_wifi_menu_enable	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"><li>• N/A</li></ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"><li>• N/A</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 4.8</li><li>• 12.6</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-95421-4</li></ul>

# Chapter 12. Inherent

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

## 12.1. Enforce Approved Authorization for Logical Access

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

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

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

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

ID	os_logical_access	
References	800-53r5	• AC-3
	CIS Benchmark	• N/A
	CIS Controls V8	• 3.3
		• 6.7

## 12.2. Ensure the System Implements Malicious Code Protection Mechanisms

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

1. This first layer of defense targets the distribution of malware; the aim is to prevent malware from ever launching. The following mechanisms are inherent to the macOS design and constitute the first layer of protection against malicious code:
  - The Apple App Store: the safest way to add new applications to a Mac is by downloading

them from the App Store; all apps available for download from the App Store have been reviewed for signs of tampering and signed by Apple to indicate that the app meets security requirements and does not contain malware.

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

2. The second layer of defense targets malware that manages to appear on a Mac before it runs; the aim is to quickly identify and block any malware present on a Mac in order to prevent the malware from running and further spreading. The following mechanisms are inherent to the macOS design and constitute the second layer of protection against malicious code:

- XProtect (defined above).
- Gatekeeper (defined above).
- Notarization (defined above).

3. The third layer of defense targets infected Mac system(s); the aim is to remediate Macs on which malware has managed to successfully execute. The following mechanism is inherent to the macOS design and constitutes the third layer of protection against malicious code:

- Apple's XProtect: a technology included on all macOS systems. XProtect will remediate infections upon receiving updated information delivered and when infections are detected

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

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

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

<b>ID</b>	os_malicious_code_prevention
-----------	------------------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• SI-3</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 10.1</li> <li>• 10.2</li> <li>• 10.5</li> </ul>

## 12.3. Enforce multifactor authentication for network access to privileged accounts

The information system implements multifactor authentication for network access to privileged accounts.

For directory bound systems: The technology supports this requirement and cannot be configured to be out of compliance. The technology inherently meets this requirement.

<b>ID</b>	os_mfa_network_access	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 5.6</li> </ul>

## 12.4. Obscure Passwords

The information system *IS* configured to obscure feedback of authentication information during the authentication process to protect the information from possible exploitation by unauthorized individuals.

The inherent configuration of a macOS uses NSSecureTextField for any text field that receives a password, which automatically obscures text which is entered.

<https://developer.apple.com/documentation/appkit/nssecuretextfield>

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

<b>ID</b>	os_obscure_password
-----------	---------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• IA-5</li> <li>• IA-6</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1</li> </ul>

## 12.5. Encrypt Stored Passwords

The information system *IS* configured to encrypt stored passwords.

Passwords need to be protected at all times, and encryption is the standard method for protecting passwords. If passwords are not encrypted, they can be plainly read (i.e., clear text) and easily compromised.

<https://developer.apple.com/documentation/openssh/using-key-authentication>

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

<b>ID</b>	os_store_encrypted_passwords	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• IA-5(1), IA-5(1)(c)</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 3.11</li> </ul>

## 12.6. Uniquely Identify Users and Processes

The macOS is a UNIX 03-compliant operating system. The system uniquely identifies and authenticates organizational users or processes.

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

<b>ID</b>	os_unique_identification	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• IA-4</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 5.1</li> <li>• 6.1</li> </ul>

# 12.7. Force Password Change at Next Logon

The macOS is able to be configured to force users to change their password at next logon.

Temporary passwords are often used for new users when accounts are created. However, once logged in to the system, users must be immediately prompted to change to a permanent password of their creation.

For a user to change their password at next logon, run the following command:

```
/usr/bin/pwpolicy -u [USER] -setpolicy "newPasswordRequired=1"
```



Replace [USER] with the username that must change the password at next logon

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

ID	pwpolicy_force_password_change	
References	800-53r5	• IA-5(1)
	CIS Benchmark	• N/A
	CIS Controls V8	• 5.2

# Chapter 13. Permanent Findings

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

## 13.1. Off-Load Audit Records

Audit records should be off-loaded onto a different system or media from the system being audited.

Information stored in only one location is vulnerable to accidental or incidental deletion or alteration. Off-loading is a common process in information systems with limited audit storage capacity.

To secure audit records by off-loading, many operating systems can be integrated with enterprise-level auditing mechanisms that meet or exceed this requirement.

The technology does not support this requirement. This is an applicable-does not meet finding.

<b>ID</b>	audit_off_load_records	
<b>References</b>	<b>800-53r5</b> <ul style="list-style-type: none"><li>• AU-4(1)</li></ul> <b>CIS Benchmark</b> <ul style="list-style-type: none"><li>• N/A</li></ul> <b>CIS Controls V8</b> <ul style="list-style-type: none"><li>• 8.9</li></ul>	

## 13.2. Must Authenticate Before Establishing a Connection

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

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

<b>ID</b>	os_auth_peripherals
-----------	---------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• IA-3</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 13.9</li> </ul>

## 13.3. Secure Name Address Resolution Service

The information system requests and performs data origin authentication and data integrity verification on the name/address resolution responses the system receives from authoritative sources.



macOS supports encrypted DNS settings with the com.apple.dnsSettings.managed payload, however, the system must be integrated with a DNS server that supports encrypted DNS. <https://developer.apple.com/documentation/devicemanagement/dnssettings>

The technology does not support this requirement. This is an applicable-does not meet finding.

<b>ID</b>	os_secure_name_resolution	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• SC-21</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.9</li> </ul>



# Chapter 14. Not Applicable

This section contains the controls that are defined in the NIST 800-53 revision 5 but are not applicable when configuring a macOS system.

## 14.1. Access Control for Mobile Devices

A mobile device is a computing device that has a small form factor such that it can easily be carried by a single individual; is designed to operate without a physical connection; possesses local, non-removable or removable data storage; and includes a self-contained power source. Mobile device functionality may also include voice communication capabilities, on-board sensors that allow the device to capture information, and/or built-in features for synchronizing local data with remote locations. Examples include smart phones and tablets. Mobile devices are typically associated with a single individual. The processing, storage, and transmission capability of the mobile device may be comparable to or merely a subset of notebook/desktop systems, depending on the nature and intended purpose of the device. Protection and control of mobile devices is behavior or policy-based and requires users to take physical action to protect and control such devices when outside of controlled areas. Controlled areas are spaces for which organizations provide physical or procedural controls to meet the requirements established for protecting information and systems.

Due to the large variety of mobile devices with different characteristics and capabilities, organizational restrictions may vary for the different classes or types of such devices. Usage restrictions and specific implementation guidance for mobile devices include configuration management, device identification and authentication, implementation of mandatory protective software, scanning devices for malicious code, updating virus protection software, scanning for critical software updates and patches, conducting primary operating system (and possibly other resident software) integrity checks, and disabling unnecessary hardware.

Usage restrictions and authorization to connect may vary among organizational systems. For example, the organization may authorize the connection of mobile devices to its network and impose a set of usage restrictions, while a system owner may withhold authorization for mobile device connection to specific applications or impose additional usage restrictions before allowing mobile device connections to a system.

The technology does not support this requirement. This is an applicable-does not meet finding.

ID	os_access_control_mobile_devices	
References	800-53r5	• AC-19
	CIS Benchmark	• N/A
	CIS Controls V8	• 6.4

# Chapter 15. Supplemental

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

## 15.1. 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 `fdsetup` 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 `fdsetup` Command

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

```
/usr/bin/fdsetup 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 `fdsetup`.



Apple has deprecated `fdsetup` 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](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.

## 15.2. Password Policy Supplemental

To comply with Executive Order 14028, “Improving the Nation’s Cybersecurity”, OMB M-22-09, “Moving the U.S. Government Toward Zero Trust Cybersecurity Principles”, and NIST SP-800-63b, “Digital Identity Guidelines: Authentication and Lifecycle Management” federal, military, and intelligence communities must adopt the following configuration settings:

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

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

- `pwpolicy_alpha_numeric_enforce`
- `pwpolicy_custom_regex_enforce`
- `pwpolicy_lower_case_character_enforce.yaml`

- pwpolicy\_max\_lifetime\_enforce
- pwpolicy\_minimum\_lifetime\_enforce
- pwpolicy\_prevent\_dictionary\_words
- pwpolicy\_simple\_sequence\_disable
- pwpolicy\_special\_character\_enforce
- pwpolicy\_upper\_case\_character\_enforce.yaml

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

- pwpolicy\_lower\_case\_character\_enforce
- pwpolicy\_upper\_case\_character\_enforce
- pwpolicy\_account\_inactivity\_enforce
- pwpolicy\_minimum\_lifetime\_enforce

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

- Enforcing at least 1 lowercase character
- Enforcing at least 1 uppercase character
- Disabling an account after 35 days of inactivity
- Password minimum lifetime

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

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>policyCategoryAuthentication</key>
  <array>
    <dict>
      <key>policyContent</key>
      <string>policyAttributeLastAuthenticationTime > policyAttributeCurrentTime
- (policyAttributeInactiveDays * 24 * 60 * 60)</string>
      <key>policyIdentifier</key>
      <string>Inactive Account</string>
      <key>policyParameters</key>
      <dict>
        <key>policyAttributeInactiveDays</key>
        <integer>35</integer>
      </dict>
    </dict>
  </array>
</dict>
```

```

</array>
<key>policyCategoryPasswordContent</key>
<array>
  <dict>
    <key>policyContent</key>
    <string>policyAttributePassword matches '(.*[A-Z].*){1,}+'</string>
    <key>policyIdentifier</key>
    <string>Must have at least 1 uppercase letter</string>
    <key>policyParameters</key>
    <dict>
      <key>minimumAlphaCharactersUpperCase</key>
      <integer>1</integer>
    </dict>
  </dict>
  <dict>
    <key>policyContent</key>
    <string>policyAttributePassword matches '(.*[a-z].*){1,}+'</string>
    <key>policyIdentifier</key>
    <string>Must have at least 1 lowercase letter</string>
    <key>policyParameters</key>
    <dict>
      <key>minimumAlphaCharactersLowerCase</key>
      <integer>1</integer>
    </dict>
  </dict>
  <dict>
    <key>policyContent</key>
    <string>policyAttributeLastPasswordChangeTime &lt; policyAttributeCurrentTime
- (policyAttributeMinimumLifetimeHours * 60 * 60)</string>
    <key>policyIdentifier</key>
    <string>Minimum Password Lifetime</string>
    <key>policyParameters</key>
    <dict>
      <key>policyAttributeMinimumLifetimeHours</key>
      <integer>24</integer>
    </dict>
  </dict>
</array>
</dict>
</plist>

```

Run the following command to load the new policy file, substituting the path to the file in place of "\$pwpolicy\_file".

```
/usr/bin/pwpolicy setaccountpolicies $pwpolicy_file
```



If directory services is being utilized, password policies should come from the domain.



In order to apply any password policy, the `allowPasscodeModification` setting in `com.apple.applicationaccess` must not be set to `false`.