



# macOS Security Compliance

## macOS 12.0 *Security Configuration - CIS Controls Version 8*

Monterey Guidance, Revision 2 (2022-03-16)

# 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. Authentication .....	8
6.1. Enforce Multifactor Authentication for Login .....	8
6.2. Enforce Multifactor Authentication for the su Command .....	9
6.3. Enforce Multifactor Authentication for Privilege Escalation Through the sudo Command ..	11
6.4. Allow Smartcard Authentication .....	12
6.5. Enforce Smartcard Authentication .....	13
6.6. Disable Password Authentication for SSH .....	14
7. Auditing .....	16
7.1. Configure Audit Log Files to Not Contain Access Control Lists .....	16
7.2. Configure Audit Log Folder to Not Contain Access Control Lists .....	17
7.3. Enable Security Auditing .....	17
7.4. Configure Audit_Control to Not Contain Access Control Lists .....	18
7.5. Configure Audit_Control Group to Wheel .....	19
7.6. Configure Audit_Control Owner to Mode 440 or Less Permissive .....	20
7.7. Configure Audit_Control Owner to Root .....	21
7.8. Configure Audit Log Files Group to Wheel .....	21
7.9. Configure Audit Log Files to Mode 440 or Less Permissive .....	22
7.10. Configure Audit Log Files to be Owned by Root .....	23
7.11. Configure Audit Flags .....	24
7.12. Configure Audit Log Folders Group to Wheel .....	25
7.13. Configure Audit Log Folders to be Owned by Root .....	26
7.14. Configure Audit Log Folders to Mode 700 or Less Permissive .....	27
7.15. Configure Audit Retention to a Minimum of Seven Days .....	28
7.16. Configure Audit Retention to a Minimum of Sixty Days or One Gigabyte .....	29
8. macOS .....	31
8.1. Disable AirDrop .....	31
8.2. Disable Apple ID Setup during Setup Assistant .....	32
8.3. Enable Authenticated Root .....	33
8.4. Disable Bonjour Multicast .....	34
8.5. Disable Calendar.app .....	35
8.6. Enforce Installation of XProtect, MRT, and Gatekeeper Updates Automatically .....	36

8.7. Integrate System into a Directory Services Infrastructure .....	37
8.8. Ensure Extensible Firmware Interface Version is Valid .....	38
8.9. Must Use ESS .....	39
8.10. Disable FaceTime.app .....	40
8.11. Disable FileVault Automatic Login .....	41
8.12. Enable Firewall Logging .....	42
8.13. Enable Gatekeeper .....	43
8.14. Enforce Gatekeeper 30 Day Automatic Rearm .....	44
8.15. Disable Handoff .....	45
8.16. Disable the Built-in Web Server .....	46
8.17. Disable iCloud Storage Setup during Setup Assistant .....	47
8.18. Configure Install.log Retention to 365 Days or More .....	48
8.19. Disable Infrared (IR) support .....	49
8.20. Enable Library Validation .....	50
8.21. Disable Mail App .....	51
8.22. Enforce Enrollment in Mobile Device Management .....	53
8.23. Disable Messages App .....	54
8.24. Enable Apple Mobile File Integrity .....	55
8.25. Disable Network File System Service .....	56
8.26. Enable Parental Controls .....	57
8.27. Disable Password Autofill .....	58
8.28. Remove Password Hint From User Accounts .....	59
8.29. Disable Proximity Based Password Sharing Requests .....	60
8.30. Disable Password Sharing .....	61
8.31. Disable Privacy Setup Services During Setup Assistant .....	62
8.32. Disable Root Login .....	63
8.33. Disable Automatic Opening of Safe Files in Safari .....	64
8.34. Enable Show All Filename Extensions .....	64
8.35. Ensure System Integrity Protection is Enabled .....	65
8.36. Disable Siri Setup during Setup Assistant .....	66
8.37. Disable Unlock with Apple Watch During Setup Assistant .....	67
8.38. Configure Sudo Timeout Period to Zero .....	68
8.39. Configure Sudoers Timestamp Type .....	69
8.40. Configure Sudoers to Authenticate Users on a Per -tty Basis .....	70
8.41. Ensure Appropriate Permissions Are Enabled for System Wide Applications .....	71
8.42. Ensure Secure Keyboard Entry Terminal.app is Enabled .....	72
8.43. Disable Trivial File Transfer Protocol Service .....	73
8.44. Ensure Time Offset Within Limits .....	74
8.45. Enable Time Synchronization Daemon .....	75
8.46. Disable TouchID Prompt during Setup Assistant .....	75
8.47. Disable Login to Other User's Active and Locked Sessions .....	76

8.48. Disable Unix-to-Unix Copy Protocol Service .....	77
8.49. Ensure No World Writable Files Exist in the Library Folder .....	78
8.50. Ensure No World Writable Files Exist in the System Folder .....	79
9. Password Policy .....	81
9.1. Restrict Maximum Password Lifetime to 60 Days .....	81
9.2. Disable Accounts after 35 Days of Inactivity .....	82
9.3. Limit Consecutive Failed Login Attempts to Three .....	84
9.4. Limit Consecutive Failed Login Attempts to Five .....	84
9.5. Set Account Lockout Time to 15 Minutes .....	85
9.6. Require Passwords Contain a Minimum of One Numeric Character .....	86
9.7. Prohibit Password Reuse for a Minimum of Five Generations .....	87
9.8. Prohibit Password Reuse for a Minimum of Fifteen Generations .....	88
9.9. Require Passwords Contain a Minimum of One Lowercase Character .....	89
9.10. Require a Minimum Password Length of 15 Characters .....	91
9.11. Set Minimum Password Lifetime to 24 Hours .....	92
9.12. Prohibit Repeating, Ascending, and Descending Character Sequences .....	94
9.13. Require Passwords Contain a Minimum of One Special Character .....	95
9.14. Require Passwords Contain a Minimum of One Uppercase Character .....	96
10. iCloud .....	98
10.1. Disable iCloud Address Book .....	98
10.2. Disable the System Preference Pane for Apple ID .....	99
10.3. Disable iCloud Bookmarks .....	100
10.4. Disable the iCloud Calendar Services .....	101
10.5. Disable iCloud Document Sync .....	102
10.6. Disable iCloud Keychain Sync .....	103
10.7. Disable iCloud Mail .....	104
10.8. Disable iCloud Notes .....	105
10.9. Disable iCloud Photo Library .....	106
10.10. Disable iCloud Reminders .....	107
10.11. Disable iCloud Desktop and Document Folder Sync .....	108
11. System Preferences .....	110
11.1. Disable Airplay Receiver .....	110
11.2. Disable Unattended or Automatic Logon to the System .....	111
11.3. Disable Bluetooth When no Approved Device is Connected .....	112
11.4. Enable Bluetooth Menu .....	113
11.5. Disable Bluetooth Sharing .....	114
11.6. Disable Bluetooth When No Devices are Paired .....	115
11.7. Disable CD/DVD Sharing .....	116
11.8. Disable Content Caching Service .....	117
11.9. Enforce Critical Security Updates to be Installed .....	118
11.10. Disable Sending Diagnostic and Usage Data to Apple .....	119

11.11. Enforce FileVault .....	120
11.12. Disable Find My Service .....	121
11.13. Enable macOS Application Firewall .....	123
11.14. Enable Firewall Stealth Mode .....	124
11.15. Disable Guest Access to Shared SMB Folders .....	125
11.16. Disable the Guest Account .....	126
11.17. Secure Hot Corners .....	127
11.18. Disable Sending Siri and Dictation Information to Apple .....	128
11.19. Enforce macOS Updates are Automatically Installed .....	129
11.20. Disable the Internet Accounts System Preference Pane .....	130
11.21. Disable Internet Sharing .....	131
11.22. Audit Location Services .....	132
11.23. Enable Location Services .....	133
11.24. Configure Login Window to Prompt for Username and Password .....	134
11.25. Disable Media Sharing .....	135
11.26. Disable Password Hints .....	136
11.27. Disable Personalized Advertising .....	137
11.28. Disable Power Nap .....	138
11.29. Disable Printer Sharing .....	139
11.30. Disable Remote Apple Events .....	140
11.31. Disable Remote Management .....	141
11.32. Disable Screen Sharing and Apple Remote Desktop .....	142
11.33. Enforce Session Lock After Screen Saver is Started .....	142
11.34. Enforce Screen Saver Timeout .....	143
11.35. Disable Siri .....	144
11.36. Disable Server Message Block Sharing .....	145
11.37. Enforce Software Update App Update Updates Automatically .....	146
11.38. Enforce Software Update Downloads Updates Automatically .....	147
11.39. Enforce Software Update Automatically .....	148
11.40. Ensure Software Update is Updated and Current .....	149
11.41. Disable SSH Server for Remote Access Sessions .....	150
11.42. Require Administrator Password to Modify System-Wide Preferences .....	151
11.43. Configure Time Machine for Automatic Backups .....	152
11.44. Ensure Time Machine Volumes are Encrypted .....	153
11.45. Configure macOS to Use an Authorized Time Server .....	154
11.46. Enable macOS Time Synchronization Daemon (timed) .....	155
11.47. Ensure Wake for Network Access Is Disabled .....	156
11.48. Disable Wi-Fi Interface .....	157
11.49. Enable Wifi Menu .....	158
12. Inherent .....	159
12.1. Enforce Approved Authorization for Logical Access .....	159

12.2. Ensure the System Implements Malicious Code Protection Mechanisms .....	159
12.3. Enforce multifactor authentication for network access to privileged accounts. ....	161
12.4. Obscure Passwords .....	161
12.5. Encrypt Stored Passwords .....	162
12.6. Uniquely Identify Users and Processes .....	162
12.7. Force Password Change at Next Logon .....	163
13. Permanent Findings .....	164
13.1. Off-Load Audit Records .....	164
13.2. Must authenticate peripherals before establishing a connection .....	164
13.3. Secure Name Address Resolution Service .....	165
14. Not Applicable .....	166
14.1. Access Control for Mobile Devices .....	166
15. Supplemental .....	167
15.1. FileVault Supplemental .....	167
15.2. Packet Filter (pf) Supplemental .....	168
15.3. Password Policy Supplemental .....	173
15.4. Smartcard Supplemental .....	176

# 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.

## 2. Scope

This guide describes the actions to take when securing a macOS 12.0 system against the CIS Controls version 8 baseline.



### 3. Authors

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

## 4. Acronyms and Definitions

Table 1. Acronyms and Abbreviations

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

## 5. Applicable Documents

### 5.1. Government Documents

Table 2. 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</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 2</i>

Table 3. Defense Information Systems Agency (DISA)

Document Number or Descriptor	Document Title
<a href="#">STIG Ver 1, Rel 1</a>	<i>Apple macOS 12 (Monterey) STIG</i>

Table 4. 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 5. Apple

Document Number or Descriptor	Document Title
<a href="#">Apple Platform Security Guide</a>	<i>Apple Platform Security</i>
<a href="#">Deployment Reference for Mac</a>	<i>Deployment Reference</i>
<a href="#">Mobile Device Management Settings</a>	<i>Mobile Device Management Settings</i>
<a href="#">Profile-Specific Payload Keys</a>	<i>Profile-Specific Payload Keys</i>
<a href="#">Security Certifications and Compliance Center</a>	<i>Security Certifications and Compliance Center</i>

Table 6. Center for Internet Security

Document Number or Descriptor	Document Title
<a href="#">Apple macOS 12.0</a>	<i>CIS Apple macOS 12.0 Benchmark version 1.0</i>

## 6. Authentication

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



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

### 6.1. Enforce Multifactor Authentication for Login

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

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



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



/etc/pam.d/login will be automatically modified to its original state following any update or major upgrade to the operating system.

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

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

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

## Remediation Description

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

```
/bin/cat > /etc/pam.d/login << LOGIN_END
# login: auth account password session
auth      sufficient    pam_smartcard.so
auth      optional      pam_krb5.so use_kcminit
auth      optional      pam_ntlm.so try_first_pass
auth      optional      pam_mount.so try_first_pass
auth      required      pam_opendirectory.so try_first_pass
auth      required      pam_deny.so
account    required      pam_nologin.so
account    required      pam_opendirectory.so
password   required      pam_opendirectory.so
session    required      pam_launchd.so
session    required      pam_uwtmp.so
session    optional     pam_mount.so
LOGIN_END

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

<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, 6.4, 6.5</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-90877-2</li></ul>

## 6.2. Enforce Multifactor Authentication for the su Command

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

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



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



/etc/pam.d/su will be automatically modified to its original state following any update or major upgrade to the operating system.

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

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

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

### Remediation Description

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

```
/bin/cat > /etc/pam.d/su << SU_END
# su: auth account password session
auth      sufficient    pam_smartcard.so
auth      required      pam_rootok.so
auth      required      pam_group.so no_warn group=admin,wheel ruser root_only
fail_safe
account    required      pam_permit.so
account    required      pam_opendirectory.so no_check_shell
password   required      pam_opendirectory.so
session    required      pam_launchd.so
SU_END

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

ID

auth\_pam\_su\_smartcard\_enforce



References	800-53r5	<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, 6.4, 6.5</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>CCE-90878-0</li> </ul>

## 6.3. Enforce Multifactor Authentication for Privilege Escalation Through the sudo Command

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

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



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



/etc/pam.d/sudo will be automatically modified to its original state following any update or major upgrade to the operating system.

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

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

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

## Remediation Description

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

```
/bin/cat > /etc/pam.d/sudo << SUDO_END
# sudo: auth account password session
auth      sufficient    pam_smartcard.so
auth      required      pam_opendirectory.so
auth      required      pam_deny.so
account    required      pam_permit.so
password   required      pam_deny.so
session    required      pam_permit.so
SUDO_END

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

ID	auth_pam_sudo_smartcard_enforce	
References	800-53r5	• IA-2(1), IA-2(2), IA-2(8)
	CIS Benchmark	• N/A
	CIS Controls V8	• 6.3, 6.4, 6.5
	CCE	• CCE-90879-8

## 6.4. 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, 6.4, 6.5
	CCE	• CCE-90880-6

## 6.5. 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/>
```

<b>ID</b>	auth_smartcard_enforce	
<b>References</b>	<b>800-53r5</b>	<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>
	<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, 6.4, 6.5</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-90883-0</li> </ul>

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

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

### Remediation Description

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

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

ID	auth_ssh_password_authentication_disable	
References	800-53r5	<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>
	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, 6.4, 6.5</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>• CCE-90884-8</li> </ul>

# 7. 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.

## 7.1. Configure Audit Log Files to Not Contain Access Control Lists

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

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

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

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

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

### Remediation Description

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

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

ID	audit_acls_files_configure	
References	800-53r5	<ul style="list-style-type: none"><li>• AU-9</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 3.5 (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-90851-7</li></ul>

## 7.2. Configure Audit Log Folder to Not Contain Access Control Lists

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

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

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

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

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

**Remediation Description**

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

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

ID	audit_acls_folders_configure	
References	800-53r5	• AU-9
	CIS Benchmark	• 3.5 (level 1)
	CIS Controls V8	• 3.3
	CCE	• CCE-90852-5

## 7.3. Enable Security Auditing

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

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

The content required to be captured in an audit record varies based on the impact level of an

organization’s system. Content that may be necessary to satisfy this requirement includes, for example, time stamps, source addresses, destination addresses, user identifiers, event descriptions, success/fail indications, filenames involved, and access or flow control rules invoked.

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



Security auditing is enabled by default on macOS.

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

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

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

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

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

ID	audit_auditd_enabled	
References	800-53r5	<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>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 3.1 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 8.2, 8.5</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-90854-1</li></ul>

## 7.4. 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-91088-5</li></ul>

## 7.5. 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
```

<b>ID</b>	audit_control_group_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-91089-3</li> </ul>	

## 7.6. 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 | 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> <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-91090-1</li> </ul>	

## 7.7. 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
```

ID	audit_control_owner_configure	
References	800-53r5	• AU-9
	CIS Benchmark	• 3.5 (level 1)
	CIS Controls V8	• 3.3
	CCE	• CCE-91091-9

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

<b>ID</b>	audit_files_group_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-90858-2</li></ul>

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

ID	audit_files_mode_configure	
References	800-53r5	<ul style="list-style-type: none"><li>• AU-9</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 3.5 (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-90859-0</li></ul>

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

<b>ID</b>	audit_files_owner_configure	
<b>References</b>	<div>800-53r5</div> <div>CIS Benchmark</div> <div>CIS Controls V8</div> <div>CCE</div>	<ul style="list-style-type: none"> <li>• AU-9</li> <li>• 3.5 (level 1)</li> <li>• 3.3</li> <li>• CCE-90860-8</li> </ul>

## 7.11. Configure Audit Flags

The auditing system *MUST* be configured with at least the minimal flags of fm, ad, -ex, aa, -fr, lo, and -fw.

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

```
/usr/bin/sed -n 's/^flags: //p' /etc/security/audit_control | /usr/bin/grep -ce 'fm,ad,\-ex,aa,\-fr,lo,\-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/sed -i '' 's/^flags: .*/flags:fm,ad,\-ex,aa,\-fr,lo,\-fw/' /etc/security/audit_control; /usr/sbin/audit -s
```



NOTE: This fix will replace the contents of the flags: line in `/etc/security/audit_control`, if you have customized the flags, your changes may be overwritten.

<b>ID</b>	audit_flags_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, 8.2, 8.5</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-91092-7</li> </ul>

## 7.12. Configure Audit Log Folders Group to Wheel

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

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

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

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

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

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

### Remediation Description

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

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

<b>ID</b>	audit_folder_group_configure
-----------	------------------------------

References	800-53r5	<ul style="list-style-type: none"> <li>• AU-9</li> </ul>
	CIS Benchmark	<ul style="list-style-type: none"> <li>• 3.5 (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-90869-9</li> </ul>

## 7.13. Configure Audit Log Folders to be Owned by Root

Audit log files *MUST* be owned by root.

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

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

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

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

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

### Remediation Description

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

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

ID	audit_folder_owner_configure
----	------------------------------



<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-90870-7</li> </ul>

## 7.14. Configure Audit Log Folders to Mode 700 or Less Permissive

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

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

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

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

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

### Remediation Description

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

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

<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-90871-5</li> </ul>

## 7.15. Configure Audit Retention to a Minimum of Seven Days

The audit service *MUST* be configured to require records be kept for seven days or longer 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 is at least seven days old.

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
```

<b>ID</b>	audit_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>• N/A</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 8.3, 8.1</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-90875-6</li> </ul>

## 7.16. Configure Audit Retention to a Minimum of Sixty Days or One Gigabyte

The audit service *MUST* be configured to require records be kept for sixty days or longer before deletion, unless the system uses a central audit record storage facility.

When "expire-after" is set to "60d", the audit service will not delete audit logs until the log data is at least sixty days old.

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 **60d or 1G**, 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:60d or 1G/'
/etc/security/audit_control; /usr/sbin/audit -s
```

<b>ID</b>	audit_retention_configure_sixty_days
-----------	--------------------------------------

<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>• 3.4 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 8.3, 8.1</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-91093-5</li> </ul>

# 8. macOS

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



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

## 8.1. Disable AirDrop

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

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

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

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

**Remediation Description**

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

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

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

ID	os_airdrop_disable
----	--------------------

<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>• 2.4.11 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1, 4.8, 6.7</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-90898-8</li> </ul>

## 8.2. Disable Apple ID Setup during Setup Assistant

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

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

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

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

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

### Remediation Description

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

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

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

<b>ID</b>	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, 4.8</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-90902-8</li> </ul>

## 8.3. 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.

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

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

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

### Remediation Description

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

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



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

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

<b>References</b>	<b>800-53r5</b>	<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> </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-90907-7</li> </ul>

## 8.4. 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/>
```

<b>ID</b>	os_bonjour_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>• 4.1 (level 2)</li> </ul>
	CIS Controls V8	<ul style="list-style-type: none"> <li>• 4.1, 4.8</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>• CCE-90908-5</li> </ul>

## 8.5. Disable Calendar.app

The macOS built-in Calendar.app *MUST* be disabled as this application can establish a connection to non-approved services. This rule is in place to prevent inadvertent data transfers.



Some organizations allow the use of the built-in Calendar.app for organizational communication. Information System Security Officers (ISSOs) may make the risk-based decision not to disable the macOS built-in Mail.app 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
function run() {
  let pref1 = ObjC.unwrap(
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess.new')\
.objectForKey('familyControlsEnabled'))
  let pathlist =
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess.new')\
.objectForKey('pathBlackList').js
  for ( let app in pathlist ) {
    if ( ObjC.unwrap(pathlist[app]) == "/Applications/Calendar.app" && pref1 == true
){
      return("true")
    }
  }
  return("false")
}
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.new) payload type:

```
<key>familyControlsEnabled</key>
<true/>
<key>pathBlackList</key>
<array>
  <string>/Applications/Calendar.app</string>
</array>
```

<b>ID</b>	os_calendar_app_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></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, 4.8</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-90909-3</li></ul>

## 8.6. Enforce Installation of XProtect, MRT, and Gatekeeper Updates Automatically

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

This setting enforces definition updates for XProtect, MRT, 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, MRT, and Gatekeeper by default in the macOS.

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

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

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

**Remediation Description**

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

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

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

ID	os_config_data_install_enforce	
References	800-53r5	<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>• 10.1, 10.2, 10.4</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-90913-5</li></ul>

## 8.7. 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>	• N/A
	<b>CIS Benchmark</b>	• N/A
	<b>CIS Controls V8</b>	• 6.7
	<b>CCE</b>	• CCE-91087-7

## 8.8. Ensure Extensible Firmware Interface Version is Valid

The macOS Extensible Firmware Interface (EFI) *MUST* be checked to ensure it is a known good version from Apple.

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

```
if /usr/sbin/ioreg -w 0 -c AppleSEPManager | /usr/bin/grep -q AppleSEPManager; then  
echo "1"; else /usr/libexec/firmwarecheckers/eficheck/eficheck --integrity-check |  
/usr/bin/grep -c "No changes detected"; fi
```

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

#### Remediation Description

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

Install a known good version of macOS.

<b>ID</b>	os_efi_integrity_validated	
<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>• 2.11 (level 1)</li> </ul>
	<b>CIS Controls V8</b> <ul style="list-style-type: none"> <li>• 2.2</li> </ul>	
	<b>CCE</b> <ul style="list-style-type: none"> <li>• CCE-91102-4</li> </ul>	

## 8.9. 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.

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> <ul style="list-style-type: none"> <li>• N/A</li> <li>◦ N/A</li> </ul>	
	<b>CCE</b> <ul style="list-style-type: none"> <li>• CCE-90930-9</li> </ul>	

## 8.10. Disable FaceTime.app

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

The FaceTime.app establishes a connection to Apple's iCloud service, even when security controls have been put in place to disable iCloud access.

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.new')\
  .objectForKey('familyControlsEnabled'))
  let pathlist =
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess.new')\
  .objectForKey('pathBlackList').js
  for ( let app in pathlist ) {
    if ( ObjC.unwrap(pathlist[app]) == "/Applications/FaceTime.app" && pref1 == true
){
      return("true")
    }
  }
  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.new) payload type:

```
<key>familyControlsEnabled</key>
<true/>
<key>pathBlackList</key>
<array>
  <string>/Applications/FaceTime.app</string>
</array>
```

ID	os_facetime_app_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, 4.8</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>• CCE-90919-2</li> </ul>

## 8.11. 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, 6.7</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>• CCE-90922-6</li> </ul>

## 8.12. Enable Firewall Logging

Firewall logging *MUST* be enabled.

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



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

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

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

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



### Remediation Description

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

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

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

ID	os_firewall_log_enable	
References	800-53r5	<ul style="list-style-type: none"><li>• AU-12</li><li>• SC-7</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 3.6 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.5, 8.2, 8.5</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-90924-2</li></ul>

## 8.13. Enable Gatekeeper

Gatekeeper *MUST* be enabled.

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

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

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

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

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

### Remediation Description

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

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

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

ID	os_gatekeeper_enable	
References	800-53r5	<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>• 2.5.2.1 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 10.1, 10.2, 10.5</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-90926-7</li></ul>

## 8.14. Enforce Gatekeeper 30 Day Automatic Rearm

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

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

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

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

## Remediation Description

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



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

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

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

ID	os_gatekeeper_rearm	
References	<b>800-53r5</b>	<ul style="list-style-type: none"><li>• CM-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>• 10.5</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-90927-5</li></ul>

## 8.15. Disable Handoff

Handoff *MUST* be disabled.

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

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

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

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

## Remediation Description

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

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

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

<b>ID</b>	os_handoff_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>• N/A</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 4.1, 4.8</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-90929-1</li></ul>

## 8.16. Disable the Built-in Web Server

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



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

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

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

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

## Remediation Description

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

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

ID	os_httpd_disable	
References	800-53r5	<ul style="list-style-type: none"><li>• AC-17</li><li>• AC-3</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 4.4 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 3.3, 6.7</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-90932-5</li></ul>

## 8.17. Disable iCloud Storage Setup during Setup Assistant

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

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

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

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

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

## Remediation Description

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

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

```
<key>Skip iCloud Storage Setup</key>
<true/>
```

ID	os_icloud_storage_prompt_disable	
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, 4.8</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-90933-3</li></ul>

## 8.18. Configure Install.log Retention to 365 Days or More

The install.log *MUST* be configured to require records be kept for 365 days or longer 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"} else if
(ttl != "True") { print "TTL not configured" } else if (max == "True") { print "Max
Size is configured, must be removed" } 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.

ID	os_install_log_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>• 3.3 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 8.1, 8.3</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-91107-3</li></ul>

## 8.19. 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 their defined payload types.

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

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

ID	os_ir_support_disable	
References	<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, 4.8, 12.6</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-90939-0</li></ul>

## 8.20. 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/>
```

ID	os_library_validation_enabled	
References	800-53r5	<ul style="list-style-type: none"><li>• N/A</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 5.1.4 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 2.3, 2.6</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-91108-1</li></ul>

## 8.21. Disable Mail App

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

The Mail.app contains functionality that can establish connections to Apple's iCloud, even when security controls to disable iCloud access have been put in place.



Some organizations allow the use of the built-in Mail.app for organizational communication. Information System Security Officers (ISSOs) may make the risk-based decision not to disable the macOS built-in Mail.app 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
function run() {
  let pref1 = ObjC.unwrap(
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess.new')\
  .objectForKey('familyControlsEnabled'))
  let pathlist =
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess.new')\
  .objectForKey('pathBlackList').js
  for ( let app in pathlist ) {
    if ( ObjC.unwrap(pathlist[app]) == "/Applications/Mail.app" && pref1 == true ){
      return("true")
    }
  }
  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.new) payload type:

```

<key>familyControlsEnabled</key>
<true/>
<key>pathBlackList</key>
<array>
  <string>/Applications/Mail.app</string>
</array>

```

<b>ID</b>	os_mail_app_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> </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, 4.8</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-90946-5</li> </ul>

## 8.22. Enforce Enrollment in Mobile Device Management

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

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

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

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

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

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

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

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

## Remediation Description

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

Ensure that system is enrolled via UAMDM.

ID	os_mdm_require	
References	800-53r5	<ul style="list-style-type: none"><li>• CM-2</li><li>• CM-6</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, 5.1</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-90950-7</li></ul>

## 8.23. Disable Messages App

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

The Messages.app establishes a connection to Apple's iCloud service, even when security controls to disable iCloud access have been put in place.

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.new')\
  .objectForKey('familyControlsEnabled'))
  let pathlist =
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess.new')\
  .objectForKey('pathBlackList').js
  for ( let app in pathlist ) {
    if ( ObjC.unwrap(pathlist[app]) == "/Applications/Messages.app" && pref1 == true
){
      return("true")
    }
  }
  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.new) payload type:

```
<key>familyControlsEnabled</key>
<true/>
<key>pathBlackList</key>
<array>
  <string>/Applications/Messages.app</string>
</array>
```

ID	os_messages_app_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, 4.8</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-90951-5</li></ul>

## 8.24. 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, 2.6</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-91109-9</li></ul>

## 8.25. Disable Network File System Service

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

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

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

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

### Remediation Description

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

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

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

<b>ID</b>	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>• 4.5 (level 1)</li> </ul>
	CIS Controls V8	<ul style="list-style-type: none"> <li>• 3.3, 6.7</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>• CCE-90956-4</li> </ul>

## 8.26. Enable Parental Controls

Parental Controls *MUST* be enabled.

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

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

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

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

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

### Remediation Description

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

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

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

ID	os_parental_controls_enable
----	-----------------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• CM-7(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>• 4.8</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-90966-3</li> </ul>

## 8.27. Disable Password Autofill

Password Autofill *MUST* be disabled.

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

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

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

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

### Remediation Description

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

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

```
<key>allowPasswordAutoFill</key>
<false/>
```

<b>ID</b>	os_password_autofill_disable
-----------	------------------------------



<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• CM-7, CM-7(1)</li> <li>• IA-11</li> <li>• IA-5, IA-5(13)</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, 4.8</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-90967-1</li> </ul>

## 8.28. 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):

```
/usr/bin/dscl . -list /Users hint | /usr/bin/awk '{print $2}' | /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:

```
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
```

<b>ID</b>	os_password_hint_remove
-----------	-------------------------

References	800-53r5	<ul style="list-style-type: none"> <li>• IA-6</li> </ul>
	CIS Benchmark	<ul style="list-style-type: none"> <li>• 5.14 (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-91110-7</li> </ul>

## 8.29. Disable Proximity Based Password Sharing Requests

Proximity based password sharing requests *MUST* be disabled.

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

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

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

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

### Remediation Description

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

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

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

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

References	800-53r5	<ul style="list-style-type: none"> <li>• IA-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>• 4.1, 4.8</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>• CCE-90968-9</li> </ul>

## 8.30. Disable Password Sharing

Password Sharing *MUST* be disabled.

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

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

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

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

### Remediation Description

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

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

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

ID	os_password_sharing_disable
----	-----------------------------

References	800-53r5	<ul style="list-style-type: none"> <li>• IA-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>• 4.1, 4.8</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>• CCE-90969-7</li> </ul>

## 8.31. Disable Privacy Setup Services During Setup Assistant

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

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

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

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

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

### Remediation Description

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

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

```
<key>SkipPrivacySetup</key>
<true/>
```

ID	os_privacy_setup_prompt_disable
----	---------------------------------

<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, 4.8</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-90981-2</li> </ul>

## 8.32. Disable Root Login

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

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

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

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

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

### Remediation Description

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

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

<b>ID</b>	os_root_disable	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• IA-2, IA-2(5)</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 5.6 (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-90994-5</li> </ul>

## 8.33. 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	
References	800-53r5	• N/A
	CIS Benchmark	• 6.3 (level 1)
	CIS Controls V8	• 9
	CCE	• CCE-91111-5

## 8.34. Enable Show All Filename Extensions

Show all filename extensions *MUST* be enabled in the Finder.



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

```
CURRENT_USER=$( scutil <<< "show State:/Users/ConsoleUser" \& awk  
'/Name :/ && ! /loginwindow/ { print $3 }' )
```

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

```
/usr/bin/sudo -u "$CURRENT_USER" /usr/bin/defaults read .GlobalPreferences
AppleShowAllExtensions 2>/dev/null
```

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/sudo -u "$CURRENT_USER" /usr/bin/defaults write /Users/"$CURRENT_USER"
"/Library/Preferences/.GlobalPreferences AppleShowAllExtensions -bool true
```

ID	os_show_filename_extensions_enable	
References	800-53r5	• N/A
	CIS Benchmark	• 6.2 (level 1)
	CIS Controls V8	• 2.3
	CCE	• CCE-91112-3

### 8.35. Ensure System Integrity Protection is Enabled

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

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



SIP is enabled by default in macOS.

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


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

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

**Remediation Description**

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

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



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

ID	os_sip_enable	
References	<div>800-53r5</div> <div>CIS Benchmark</div> <div>CIS Controls V8</div> <div>CCE</div>	<div><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></div> <div><ul style="list-style-type: none"><li>5.18 (level 1)</li></ul></div> <div><ul style="list-style-type: none"><li>2.6, 3.3, 10.5</li></ul></div> <div><ul style="list-style-type: none"><li>CCE-91000-0</li></ul></div>

### 8.36. Disable Siri Setup during Setup Assistant

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

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



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

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

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

**Remediation Description**

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

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

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

ID	os_siri_prompt_disable	
References	800-53r5	<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, 4.8</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-91001-8</li></ul>

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

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

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

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

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

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

### Remediation Description

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

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

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

<b>ID</b>	os_skip_unlock_with_watch_enable	
<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> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-91002-6</li> </ul>

## 8.38. Configure Sudo Timeout Period to Zero

The file /etc/sudoers *MUST* include a timestamp\_timeout of zero.

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

```
/usr/bin/find /etc/sudoers* -type f -exec /usr/bin/grep -E "^Defaults
\s+timestamp_timeout=0" '{}' \; | /usr/bin/wc -l | /usr/bin/xargs
```

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	<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>• 5.3 (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-91116-4</li></ul>	

## 8.39. Configure Sudoers Timestamp Type

The file `/etc/sudoers` *MUST* be configured to not include a `timestamp_type` of `global` or `ppid`.

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/find /etc/sudoers* -type f -exec /usr/bin/grep -E  
'(^Defaults\s+timestamp_type=global|^Defaults\s+timestamp_type=ppid)' '{}' \; |  
/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:

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

ID	os_sudoers_timestamp_type_configure	
References	800-53r5	<ul style="list-style-type: none"><li>• CM-5(1)</li><li>• IA-11</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-91015-8</li></ul>

## 8.40. Configure Sudoers to Authenticate Users on a Per-tty Basis

The file `/etc/sudoers` *MUST* be configured to include `tty_tickets`.

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. Without the "tty\_tickets" option, all open local and remote logon sessions would be authenticated to use sudo without a password for the duration of the configured password timeout window.

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

```
/usr/bin/find /etc/sudoers* -type f -exec /usr/bin/grep -E "^Defaults\s+\\!tty_tickets"
'{}' \; | /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:

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

<b>ID</b>	os_sudoers_tty_configure
-----------	--------------------------

References	800-53r5	<ul style="list-style-type: none"> <li>• CM-5(1)</li> <li>• IA-11</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-91015-8</li> </ul>

## 8.41. 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
```

ID	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>• 5.1.6 (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-91117-2</li> </ul>

## 8.42. 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/>
```

<b>ID</b>	os_terminal_secure_keyboard_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 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.8</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-91118-0</li> </ul>

## 8.43. Disable Trivial File Transfer Protocol Service

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

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



TFTP service is disabled at startup by default macOS.

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

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

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

### Remediation Description

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

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

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

<b>ID</b>	os_tftpd_disable
-----------	------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• AC-17</li> <li>• AC-3</li> <li>• IA-5(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>• 3.3, 3.1, 5.2</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-91018-2</li> </ul>

## 8.44. Ensure Time Offset Within Limits

The macOS system time *MUST* be monitored to not drift more than four minutes and thirty seconds.

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

```
/usr/bin/sntp $(/usr/sbin/systemsetup -getnetworktimeserver | /usr/bin/awk '{print $4}') | /usr/bin/awk -F'. ' '/\+\/\+\/-/{if (substr($1,2) >= 270) {print "No"} 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/sntp -Ss $(/usr/sbin/systemsetup -getnetworktimeserver | /usr/bin/awk '{print $4}')
```

<b>ID</b>	os_time_offset_limit_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>• 2.2.2 (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-91119-8</li> </ul>



# 8.45. Enable Time Synchronization Daemon

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



The time synchronization daemon is enabled by default on macOS.

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

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

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

**Remediation Description**

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

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

ID	os_time_server_enabled	
References	800-53r5	<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>• N/A</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-91019-0</li></ul>

# 8.46. Disable TouchID Prompt during Setup Assistant

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

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

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

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

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

**Remediation Description**

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

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

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

ID	os_touchid_prompt_disable	
References	800-53r5	• CM-6
	CIS Benchmark	• N/A
	CIS Controls V8	• 4.1
	CCE	• CCE-91020-8

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

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

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

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

```
/usr/bin/security authorizationdb read system.login.screensaver 2>&1 | /usr/bin/grep -c 'use-login-window-ui'
```

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

### Remediation Description

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

```
/usr/bin/security authorizationdb write system.login.screensaver "use-login-window-ui"
```

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>• 5.11 (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-91022-4</li></ul>

## 8.48. Disable Unix-to-Unix Copy Protocol Service

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

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



UUCP service is disabled at startup by default macOS.

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

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

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

### Remediation Description

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

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

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

<b>ID</b>	os_uucp_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>• N/A</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 3.3, 4.1, 4.8</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-91024-0</li></ul>

## 8.49. 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 /System/Volumes/Data/Library -type d -perm -2 -ls | /usr/bin/grep -v  
Caches | /usr/bin/grep -v /Preferences/Audio/Data | /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 $( /usr/bin/find /System/Volumes/Data/Library -type d -perm
-2 | /usr/bin/grep -v Caches | /usr/bin/grep -v /Preferences/Audio/Data ); do
  /bin/chmod -R o-w "$libPermissions"
done
```

ID	os_world_writable_library_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>• 5.1.8 (level 2)</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-91120-6</li></ul>

## 8.50. 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 -v
"Drop Box" | /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 $( /usr/bin/find /System/Volumes/Data/System -type d -perm
-2 | /usr/bin/grep -v "Drop Box" ); do
    /bin/chmod -R o-w "$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>• 5.1.7 (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-91121-4</li></ul>

## 9. 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.

### 9.1. Restrict Maximum Password Lifetime to 60 Days

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

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



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

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

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

If the result is not **60**, this is a finding.

## Remediation Description

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

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

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

ID	pwpolicy_60_day_enforce	
References	800-53r5	• IA-5
	CIS Benchmark	• N/A
	CIS Controls V8	• 4.7
	CCE	• CCE-91027-3

## 9.2. 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 | /usr/bin/grep -v "Getting global account policies" | /usr/bin/xmllint --xpath '/plist/dict/array/dict/dict[key="policyAttributeInactiveDays"]/integer' - | /usr/bin/awk -F ' [<>]' '{print $3}'
```

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-91028-1</li></ul>

### 9.3. Limit Consecutive Failed Login Attempts to Three

The macOS *MUST* be configured to limit the number of failed login attempts to a maximum of three. 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/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.mobiledevice.passwordpolicy')\
.objectForKey('maxFailedAttempts').js
EOS
```

If the result is not 3, this is a finding.

**Remediation 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	• N/A
	CIS Controls V8	• 6.2
	CCE	• CCE-91029-9

### 9.4. Limit Consecutive Failed Login Attempts to Five

The macOS *MUST* be configured to limit the number of failed login attempts to a maximum of five. 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/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.mobiledevice.passwordpolicy')\
.objectForKey('maxFailedAttempts').js
EOS
```

If the result is not 5, this is a finding.

**Remediation 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>5</integer>
```

ID	pwpolicy_account_lockout_enforce_five	
References	800-53r5	• AC-7
	CIS Benchmark	• 5.2.1 (level 1)
	CIS Controls V8	• 6.2
	CCE	• CCE-91122-2

## 9.5. 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/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.mobiledevice.passwordpolicy')\
.objectForKey('minutesUntilFailedLoginReset').js
EOS
```

If the result is not 15, this is a finding.

**Remediation 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	• N/A
	CIS Controls V8	• 4.1
	CCE	• CCE-91030-7

## 9.6. Require Passwords Contain a Minimum of One Numeric Character

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

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



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

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

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

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

ID	pwpolicy_alpha_numeric_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>• 5.2.3 (level 2), 5.2.4 (level 2)</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-91031-5</li></ul>

## 9.7. Prohibit Password Reuse for a Minimum of Five Generations

The macOS *MUST* be configured to enforce a password history of at least five 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 five 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/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.mobiledevice.passwordpolicy')\
.objectForKey('pinHistory').js
EOS
```

If the result is not 5, this is a finding.

### Remediation Description

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

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

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

ID	pwpolicy_history_enforce	
References	800-53r5	• IA-5(1)
	CIS Benchmark	• N/A
	CIS Controls V8	• 5.2
	CCE	• CCE-91034-9

## 9.8. Prohibit Password Reuse for a Minimum of Fifteen Generations

The macOS *MUST* be configured to enforce a password history of at least fifteen 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 fifteen

previous password generations.

Limiting password reuse 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/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.mobiledevice.passwordpolicy')\
.objectForKey('pinHistory').js
EOS
```

If the result is not 15, this is a finding.

**Remediation 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>15</integer>
```

ID	pwpolicy_history_enforce_fifteen	
References	800-53r5	• IA-5(1)
	CIS Benchmark	• 5.2.8 (level 1)
	CIS Controls V8	• 5.2
	CCE	• CCE-91123-0

## 9.9. Require Passwords Contain a Minimum of One Lowercase Character

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

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



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

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

```
/usr/bin/pwpolicy getaccountpolicies | /usr/bin/grep -v "Getting global account policies" | /usr/bin/xmllint --xpath '/plist/dict/array/dict/dict[key="minimumAlphaCharactersLowerCase"]/integer' - | /usr/bin/awk -F ' [<>]' '{print $3}'
```

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

### Remediation Description

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

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

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

```
<dict>
<key>policyContent</key>
<string>policyAttributePassword matches &apos;(.*[a-z].*){1,}&apos;;</string>
<key>policyIdentifier</key>
<string>Must have at least 1 lowercase letter</string>
<key>policyParameters</key>
<dict>
<key>minimumAlphaCharactersLowerCase</key>
<integer>1</integer>
</dict>
</dict>
```

After saving the file and exiting to the command prompt, run the following command to load the new policy file, substituting the path to the file in place of "\$pwpolicy\_file".

```
/usr/bin/pwpolicy setaccountpolicies $pwpolicy_file
```



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

ID	pwpolicy_lower_case_character_enforce
----	---------------------------------------



References	800-53r5	<ul style="list-style-type: none"> <li>• IA-5(1)</li> </ul>
	CIS Benchmark	<ul style="list-style-type: none"> <li>• 5.2.6 (level 2)</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-91035-6</li> </ul>

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

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

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



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

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

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

If the result is not 15, this is a finding.

### Remediation 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>
```

<b>ID</b>	pwpolicy_minimum_length_enforce	
<b>References</b>	<b>800-53r5</b> <ul style="list-style-type: none"> <li>• IA-5(1)</li> </ul>	<b>CIS Benchmark</b> <ul style="list-style-type: none"> <li>• 5.2.2 (level 1)</li> </ul>
	<b>CIS Controls V8</b> <ul style="list-style-type: none"> <li>• 5.2</li> </ul>	
	<b>CCE</b> <ul style="list-style-type: none"> <li>• CCE-91036-4</li> </ul>	

## 9.11. Set Minimum Password Lifetime to 24 Hours

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

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



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

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

```
/usr/bin/pwpolicy getaccountpolicies | /usr/bin/grep -v "Getting global account policies" | /usr/bin/xmllint --xpath '/plist/dict/array/dict/dict[key="policyAttributeMinimumLifetimeHours"]/integer' - | /usr/bin/awk -F ' [<>] ' '{print $3}'
```

If the result is not 24, this is a finding.

## Remediation Description

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

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

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

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

After saving the file and exiting to the command prompt, run the following command to load the new policy file, substituting the path to the file in place of "\$pwpolicy\_file".

```
/usr/bin/pwpolicy setaccountpolicies $pwpolicy_file
```



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

ID	pwpolicy_minimum_lifetime_enforce	
References	800-53r5	<ul style="list-style-type: none"><li>• IA-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>• 4.7</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-91037-2</li></ul>

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

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

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



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

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

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

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

ID	pwpolicy_simple_sequence_disable
----	----------------------------------

References	800-53r5	<ul style="list-style-type: none"> <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>• 5.2</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>• CCE-91039-8</li> </ul>

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

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

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

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



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

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

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

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

## Remediation Description

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

Create a configuration profile containing the following keys in the (com.apple.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-91040-6

## 9.14. Require Passwords Contain a Minimum of One Uppercase Character

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

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



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

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

```
/usr/bin/pwpolicy getaccountpolicies | /usr/bin/grep -v "Getting global account policies" | /usr/bin/xmllint --xpath '/plist/dict/array/dict/dict[key="minimumAlphaCharactersUpperCase"]/integer' - | /usr/bin/awk -F '[<>]' '{print $3}'
```

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

**Remediation Description**

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

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

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

```
<dict>
<key>policyContent</key>
<string>policyAttributePassword matches &apos;(.*[A-Z].*){1,}&apos;;</string>
<key>policyIdentifier</key>
<string>Must have at least 1 uppercase letter</string>
<key>policyParameters</key>
<dict>
<key>minimumAlphaCharactersUpperCase</key>
<integer>1</integer>
</dict>
</dict>
```

After saving the file and exiting to the command prompt, run the following command to load the new policy file, substituting the path to the file in place of "\$pwpolicy\_file".

```
/usr/bin/pwpolicy setaccountpolicies $pwpolicy_file
```



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

ID	pwpolicy_upper_case_character_enforce	
References	800-53r5	• IA-5(1)
	CIS Benchmark	• 5.2.6 (level 2)
	CIS Controls V8	• 5.2
	CCE	• CCE-91043-0

# 10. 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.

## 10.1. 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, 4.8, 15.3</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-90885-5</li> </ul>

## 10.2. Disable the System Preference Pane for Apple ID

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

Disabling the system preference pane prevents login to Apple ID and iCloud.

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

```
/usr/bin/profiles -P -o stdout | /usr/bin/grep -c 'com.apple.preferences.AppleID' | /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.systempreferences) payload type:

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

<b>ID</b>	icloud_appleid_prefpane_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, 4.8</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-90886-3</li> </ul>

## 10.3. 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
-----------	--------------------------

<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, 4.8, 15.3</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-90887-1</li> </ul>

## 10.4. 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/>
```

<b>ID</b>	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, 4.8, 15.3</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-90888-9</li> </ul>

## 10.5. 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, 4.8, 15.3</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-90889-7</li> </ul>

## 10.6. Disable iCloud Keychain Sync

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

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

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

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

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

### Remediation Description

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

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

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

<b>ID</b>	icloud_keychain_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, 4.8, 15.3</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-90890-5</li> </ul>

## 10.7. Disable iCloud Mail

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

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

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

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

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

### Remediation Description

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

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

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

<b>ID</b>	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, 4.8, 15.3</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-90891-3</li> </ul>

## 10.8. Disable iCloud Notes

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

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

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

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

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

### Remediation Description

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

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

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

<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, 4.8, 15.3</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-90892-1</li> </ul>

## 10.9. Disable iCloud Photo Library

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

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

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

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

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

### Remediation Description

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

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

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

<b>ID</b>	icloud_photos_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, 4.8, 15.3</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-90893-9</li> </ul>

## 10.10. 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, 4.8, 15.3</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-90895-4</li> </ul>

## 10.11. 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>• 2.6.1.4 (level 2)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1, 4.8, 15.3</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-90896-2</li> </ul>

# 11. System Preferences

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



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

## 11.1. Disable Airplay Receiver

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

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

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

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

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

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

**Remediation Description**

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

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

```
<key>allowAirPlayIncomingRequests</key>
<false/>
```

ID	sysprefs_airplay_receiver_disable
----	-----------------------------------

References	800-53r5	<ul style="list-style-type: none"> <li>• CM-7, CM-7(1)</li> </ul>
	CIS Benchmark	<ul style="list-style-type: none"> <li>• 2.4.13 (level 1)</li> </ul>
	CIS Controls V8	<ul style="list-style-type: none"> <li>• 4.1, 4.8</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>• CCE-91044-8</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/>
```

ID	sysprefs_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>• 5.7 (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-91046-3</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 their defined payload types.

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

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

ID	sysprefs_bluetooth_disable	
References	<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, 12.6, 13.9</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-91048-9</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>
```

ID	sysprefs_bluetooth_menu_enable	
References	800-53r5	• N/A
	CIS Benchmark	• 2.1.2 (level 1)
	CIS Controls V8	• 4.8, 13.9
	CCE	• CCE-91124-8

## 11.5. Disable Bluetooth Sharing

Bluetooth Sharing *MUST* be disabled.

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



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

```
CURRENT_USER=$( scutil <<< "show State:/Users/ConsoleUser" \& awk  
'/Name :/ 8& ! /loginwindow/ { print $3 }' )
```

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

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

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



## Remediation Description

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

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

ID	sysprefs_bluetooth_sharing_disable	
References	800-53r5	<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>• 2.4.7 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 3.3, 4.1, 4.8</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-91049-7</li></ul>

## 11.6. Disable Bluetooth When No Devices are Paired

Bluetooth *MUST* be disabled when no devices are paired.

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

```
isPaired=$(/usr/sbin/system_profiler SPBluetoothDataType 2>/dev/null | /usr/bin/grep  
-c 'Connected: Yes')  
if [[ "$isPaired" = "0" ]]; then  
    powerState=$(/usr/sbin/system_profiler SPBluetoothDataType 2>/dev/null |  
/usr/bin/grep -c 'State: On')  
    /bin/echo "$powerState"  
else  
    /bin/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/defaults write  
/private/var/root/Library/Preferences/com.apple.BTServer.plist defaultPoweredState  
off  
/usr/bin/killall -HUP bluetoothd
```

ID	sysprefs_bluetooth_unpaired_disable	
References	800-53r5	<ul style="list-style-type: none"><li>• AC-18, AC-18(3)</li><li>• SC-8</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.1.1 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.8, 12.6, 13.9</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-91126-3</li></ul>

## 11.7. Disable CD/DVD Sharing

CD/DVD Sharing *MUST* be disabled.

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

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

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

### Remediation Description

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

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

ID	sysprefs_cd_dvd_sharing_disable
----	---------------------------------

<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>• 2.4.6 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1, 4.8</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-91127-1</li> </ul>

## 11.8. Disable Content Caching Service

Content caching *MUST* be disabled.

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

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

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

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

### Remediation Description

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

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

```
<key>allowContentCaching</key>
<false/>
```

<b>ID</b>	sysprefs_content_caching_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>• 2.4.10 (level 2)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1, 4.8</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-91050-5</li> </ul>

## 11.9. Enforce Critical Security Updates to be Installed

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

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

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

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

### Remediation Description

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

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

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

<b>ID</b>	sysprefs_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, 7.4, 7.7</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>• CCE-91051-3</li> </ul>

## 11.10. Disable Sending Diagnostic and Usage Data to Apple

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

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

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

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

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

## Remediation Description

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

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

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

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

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

<b>ID</b>	sysprefs_diagnostics_reports_disable	
<b>References</b>	<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>• 2.5.5 (level 2)</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 4.1, 4.8</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-91052-1</li></ul>

## 11.11. Enforce FileVault

FileVault *MUST* be enforced.

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

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

```
/usr/bin/fdesetup status | /usr/bin/grep -c "FileVault is On."
```

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

### Remediation Description

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



See the FileVault supplemental to implement this rule.

<b>ID</b>	sysprefs_filevault_enforce	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"><li>• SC-28, SC-28(1)</li></ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"><li>• 2.5.5.1 (level 1)</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 3.6, 3.11</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-91053-9</li></ul>

## 11.12. Disable Find My Service

The Find My service *MUST* be disabled.

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

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

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

```

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

```

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

### Remediation Description

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

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

```

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

```

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

```

<key>DisableFMiCloudSetting</key>
<true/>

```

<b>ID</b>	sysprefs_find_my_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> </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, 4.8, 15.3</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-91054-7</li> </ul>

## 11.13. Enable macOS Application Firewall

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

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

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

```
/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/>
```

<b>ID</b>	sysprefs_firewall_enable
-----------	--------------------------

<b>References</b>	<b>800-53r5</b>	<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>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 2.5.2.2 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1, 4.5, 13.1</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-91055-4</li> </ul>

## 11.14. Enable Firewall Stealth Mode

Firewall Stealth Mode *MUST* be enabled.

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



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

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

```
/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/>
```

<b>ID</b>	sysprefs_firewall_stealth_mode_enable	
<b>References</b>	<div>800-53r5</div> <ul style="list-style-type: none"> <li>• CM-7, CM-7(1)</li> <li>• SC-7, SC-7(16)</li> </ul>	<div>CIS Benchmark</div> <ul style="list-style-type: none"> <li>• 2.5.2.3 (level 1)</li> </ul>
	CIS Controls V8	<ul style="list-style-type: none"> <li>• 4.1, 4.5, 4.8</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>• CCE-91056-2</li> </ul>

## 11.15. 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
```

<b>ID</b>	sysprefs_guest_access_smb_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>• 6.1.4 (level 1)</li> </ul>
	CIS Controls V8	<ul style="list-style-type: none"> <li>• 5.2, 6.2, 6.8</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>• CCE-91059-6</li> </ul>

## 11.16. 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
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.MCX')\
.objectForKey('DisableGuestAccount').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>DisableGuestAccount</key>
<true/>
```

ID	sysprefs_guest_account_disable
----	--------------------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• AC-2, AC-2(9)</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 6.1.3 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 5.2, 5.3, 6.8</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-91060-4</li> </ul>

## 11.17. 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.

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>	sysprefs_hot_corners_secure	
<b>References</b>	<b>800-53r5</b> <ul style="list-style-type: none"><li>• AC-11(1)</li></ul> <b>CIS Benchmark</b> <ul style="list-style-type: none"><li>• 2.3.2 (level 2)</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-91128-9</li></ul>	

## 11.18. Disable Sending Siri and Dictation Information to Apple

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

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

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

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

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

## Remediation Description

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

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

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

ID	sysprefs_improve_siri_dictation_disable	
References	800-53r5	<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>• N/A</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.1, 4.8</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-91062-0</li></ul>

## 11.19. 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	sysprefs_install_macos_updates_enforce	
References	800-53r5	<ul style="list-style-type: none"><li>• N/A</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 1.6 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 7.3, 7.4</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-91129-7</li></ul>

## 11.20. Disable the Internet Accounts System Preference Pane

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



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

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

```
/usr/bin/profiles -P -o stdout | /usr/bin/grep -c
'com.apple.preferences.internetaccounts' | /usr/bin/awk '{ if ($1 >= 2) {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.systempreferences) payload type:

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

<b>ID</b>	sysprefs_internet_accounts_prefpane_disable	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"><li>• AC-20</li><li>• 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>• 4.8, 15.2</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-90938-2</li></ul>

## 11.21. Disable Internet Sharing

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

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

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

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

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

### Remediation Description

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

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

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

ID	sysprefs_internet_sharing_disable	
References	800-53r5	<ul style="list-style-type: none"><li>• AC-20</li><li>• AC-4</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.4.2 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.1, 4.8</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-91063-8</li></ul>

## 11.22. Audit Location Services

The organization *MUST* audit which applications have access to location services.

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

```
sudo /usr/libexec/PlistBuddy -c print /var/db/locationd/clients.plist | grep Dict |
awk '(NR>1) { print $1 }'
```

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

### Remediation Description

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

Review the list of applications and remove any unauthorized applications from System Preferences → Security & Privacy → Privacy → Location Services.

<b>ID</b>	sysprefs_location_services_audit	
<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>• 2.5.4 (level 2)</li> </ul>
	<b>CIS Controls V8</b> <ul style="list-style-type: none"> <li>• 2.3, 4.1</li> </ul>	
	<b>CCE</b> <ul style="list-style-type: none"> <li>• CCE-91131-3</li> </ul>	

## 11.23. Enable Location Services

Location Services *MUST* be enabled.

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

```
/usr/bin/defaults read
/var/db/locationd/Library/Preferences/ByHost/com.apple.locationd.plist
LocationServicesEnabled
```

If the result is not **1**, 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; /bin/launchctl kickstart -k
system/com.apple.locationd
```

<b>ID</b>	sysprefs_location_services_enable
-----------	-----------------------------------

References	800-53r5	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	CIS Benchmark	<ul style="list-style-type: none"> <li>• 2.5.3 (level 2)</li> </ul>
	CIS Controls V8	<ul style="list-style-type: none"> <li>• 4.1, 4.8</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>• CCE-91132-1</li> </ul>

## 11.24. Configure Login Window to Prompt for Username and Password

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

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

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

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

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

### Remediation Description

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

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

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

ID	sysprefs_loginwindow_prompt_username_password_enforce
----	---

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• IA-2</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 6.1.1 (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-91065-3</li> </ul>

## 11.25. Disable Media Sharing

Media sharing *MUST* be disabled.

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

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



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

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

```
/usr/bin/osascript -l JavaScript << EOS
function run() {
  let pref1 = ObjC.unwrap(
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.preferences.sharing.SharingPrefsEx
tension')\
.objectForKey('homeSharingUIStatus'))
  let pref2 = ObjC.unwrap(
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.preferences.sharing.SharingPrefsEx
tension')\
.objectForKey('legacySharingUIStatus'))
  let pref3 = ObjC.unwrap(
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.preferences.sharing.SharingPrefsEx
tension')\
.objectForKey('mediaSharingUIStatus'))
  if ( pref1 == 0 && pref2 == 0 && pref3 == 0 ) {
    return("true")
  } else {
    return("false")
  }
}
EOS
```

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

**Remediation Description**

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

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

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

ID	sysprefs_media_sharing_disabled	
References	800-53r5	<ul style="list-style-type: none"><li>• AC-17</li><li>• AC-3</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.4.12 (level 2)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.1, 4.8</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-91066-1</li></ul>

## 11.26. Disable Password Hints

Password hints *MUST* be disabled.

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

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

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

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

## Remediation Description

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

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

```
<key>RetriesUntilHint</key>
<integer>0</integer>
```

ID	sysprefs_password_hints_disable	
References	800-53r5	• IA-6
	CIS Benchmark	• 6.1.2 (level 1)
	CIS Controls V8	• 4.1
	CCE	• CCE-91067-9

## 11.27. Disable Personalized Advertising

Ad tracking and targeted ads *MUST* be disabled.

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

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

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

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

<b>ID</b>	sysprefs_personalized_advertising_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>• 2.5.6 (level 1)</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 4.1, 4.8</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-91068-7</li></ul>

## 11.28. Disable Power Nap

Power Nap *MUST* be disabled.

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

The following Macs support Power Nap:

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

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



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

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

### Remediation Description

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

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

<b>ID</b>	sysprefs_power_nap_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>• 2.9 (level 1)</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 4.1, 4.8</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-91069-5</li></ul>

## 11.29. Disable Printer Sharing

Printer Sharing *MUST* be disabled.

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

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

<b>ID</b>	sysprefs_printer_sharing_disable	
<b>References</b>	<div>800-53r5</div> <div>CIS Benchmark</div> <div>CIS Controls V8</div> <div>CCE</div>	<ul style="list-style-type: none"> <li>• CM-7, CM-7(1)</li> <li>• 2.4.4 (level 1)</li> <li>• 4.1, 4.8</li> <li>• CCE-91134-7</li> </ul>

## 11.30. Disable Remote Apple Events

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

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

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

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

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

### Remediation Description

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

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



Systemsetup with -setremoteappleevents flag will fail unless you grant Full Disk Access to systemsetup or it's parent process. Requires UAMDM.

<b>ID</b>	sysprefs_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>• 2.4.1 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1, 4.8</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-91070-3</li> </ul>

## 11.31. Disable Remote Management

Remote Management *MUST* be disabled.

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

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

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

### Remediation Description

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

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

<b>ID</b>	sysprefs_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>• 2.4.3 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1, 4.8</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-91135-4</li> </ul>

# 11.32. Disable Screen Sharing and Apple Remote Desktop

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

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

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

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

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

**Remediation Description**

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

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

NOTE - This will apply to the whole system

ID	sysprefs_screen_sharing_disable	
References	800-53r5	<ul style="list-style-type: none"><li>• AC-17</li><li>• AC-3</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.4.3 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.1, 4.8</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-91071-1</li></ul>

# 11.33. Enforce Session Lock After Screen Saver is Started

A screen saver *MUST* be enabled and the system *MUST* be configured to require a password to

unlock once the screensaver has been on for a maximum of five 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
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.screensaver')\
.objectForKey('askForPasswordDelay').js
EOS
```

If the result is not 5, this is a finding.

**Remediation Description**

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

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

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

ID	sysprefs_screensaver_ask_for_password_delay_enforce	
References	800-53r5	<ul style="list-style-type: none"><li>• AC-11</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 5.8 (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-91072-9</li></ul>

## 11.34. Enforce Screen Saver Timeout

The screen saver timeout *MUST* be set to 20 minutes or a shorter length of time.

This rule ensures that a full session lock is triggered within no more than 20 minutes of inactivity.

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

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

If the result is not **1200**, this is a finding.

**Remediation Description**

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

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

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

ID	sysprefs_screensaver_timeout_enforce	
References	800-53r5	<ul style="list-style-type: none"><li>• AC-11</li><li>• IA-11</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.3.1 (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-91074-5</li></ul>

## 11.35. Disable Siri

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

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

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

```
/usr/bin/osascript -l JavaScript << EOS
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.ironwood.support')\
.objectForKey('Ironwood Allowed').js
EOS
```

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

**Remediation Description**

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

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

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

ID	sysprefs_siri_disable	
References	800-53r5	<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>• N/A</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.1, 4.8</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-91075-2</li></ul>

## 11.36. Disable Server Message Block Sharing

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

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

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

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

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

### Remediation Description

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

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

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

<b>ID</b>	sysprefs_smbd_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>• 2.4.8 (level 1)</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 4.1, 4.8</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-91076-0</li></ul>

## 11.37. 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/>
```

ID	sysprefs_software_update_app_update_enforce	
References	800-53r5	• N/A
	CIS Benchmark	• 1.4 (level 1)
	CIS Controls V8	• 7.3, 7.4
	CCE	• CCE-91138-8

## 11.38. 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	sysprefs_software_update_download_enforce	
References	800-53r5	<ul style="list-style-type: none"><li>• N/A</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 1.3 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 7.3, 7.4</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-91139-6</li></ul>

## 11.39. Enforce Software Update Automatically

Software Update *MUST* be configured to enforce automatic update 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('AutomaticCheckEnabled').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>AutomaticCheckEnabled</key>
<true/>
```

ID	sysprefs_software_update_enforce	
References	800-53r5	<ul style="list-style-type: none"><li>• SI-2(5)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 1.2 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 7.3, 7.4</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-91140-4</li></ul>

## 11.40. 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 -R
```

NOTE - This will apply to the whole system

ID	sysprefs_softwareupdate_current	
References	800-53r5	<ul style="list-style-type: none"><li>• N/A</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 1.1 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 7.3, 7.4</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-91141-2</li></ul>

## 11.41. Disable SSH Server for Remote Access Sessions

SSH service *MUST* be disabled for remote access.

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

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

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

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

### Remediation Description

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

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

ID	sysprefs_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>• 2.4.5 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1, 4.8</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-91077-8</li> </ul>

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

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

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

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

```
/usr/bin/security authorizationdb read system.preferences 2> /dev/null |
/usr/bin/grep -A 1 "<key>shared</key>" | /usr/bin/grep -c "<false/>"
```

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

### Remediation Description

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

```
/usr/bin/security authorizationdb read system.preferences >
/tmp/system.preferences.plist
/usr/libexec/PlistBuddy -c "Set :shared false" /tmp/system.preferences.plist
/usr/bin/security authorizationdb write system.preferences <
/tmp/system.preferences.plist
```

<b>ID</b>	sysprefs_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>• 5.10 (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-91079-4</li> </ul>

## 11.43. 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>	sysprefs_time_machine_auto_backup_enable
-----------	--

References	800-53r5	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	CIS Benchmark	<ul style="list-style-type: none"> <li>• 2.7.1 (level 2)</li> </ul>
	CIS Controls V8	<ul style="list-style-type: none"> <li>• 11.2</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>• CCE-91142-0</li> </ul>

## 11.44. Ensure Time Machine Volumes are Encrypted

Time Machine volumes *MUST* be encrypted.

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

```
error_count=0
for tm in $(/usr/bin/tmutil destinationinfo 2>/dev/null | /usr/bin/awk -F': ' '
'/Name/{print $2}'); do
    tmMounted=$(/usr/sbin/diskutil info "${tm}" 2>/dev/null | /usr/bin/awk
'/Mounted/{print $2}')
    tmEncrypted=$(/usr/sbin/diskutil info "${tm}" 2>/dev/null | /usr/bin/awk
'/FileVault/{print $2}')
    if [[ "$tmMounted" = "Yes" && "$tmEncrypted" = "No" ]]; then
        ((error_count++))
    fi
done
echo "$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:

1. Go to System Preferences → Time Machine
2. Click **Select Disk**
3. Select existing Backup Disk under **Available Disks**
4. Click **Encrypt Backups**
5. Click **Use Disk**

ID	sysprefs_time_machine_encrypted_configure
----	---

References	800-53r5	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	CIS Benchmark	<ul style="list-style-type: none"> <li>• 2.7.2 (level 2)</li> </ul>
	CIS Controls V8	<ul style="list-style-type: none"> <li>• 3.6, 3.11, 11.2</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>• CCE-91143-8</li> </ul>

## 11.45. Configure macOS to Use an Authorized Time Server

Approved time servers *MUST* be the only servers configured for use.

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-a.nist.gov,time-b.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-a.nist.gov,time-b.nist.gov</string>
```

ID	sysprefs_time_server_configure
----	--------------------------------



<b>References</b>	<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>• 2.2.1 (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-91080-2</li> </ul>

## 11.46. Enable macOS Time Synchronization Daemon (timed)

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

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

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

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

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

### Remediation Description

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



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

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

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

<b>ID</b>	sysprefs_time_server_enforce	
<b>References</b>	<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>• 2.2.1 (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-91081-0</li> </ul>	

## 11.47. 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/{print $2}'
```

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

### Remediation Description

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

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

<b>ID</b>	sysprefs_wake_network_access_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>• 2.8 (level 1)</li> </ul> <b>CIS Controls V8</b> <ul style="list-style-type: none"> <li>• 4.8</li> </ul> <b>CCE</b> <ul style="list-style-type: none"> <li>• CCE-91146-1</li> </ul>	

# 11.48. Disable Wi-Fi Interface

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

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



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

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

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

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

**Remediation Description**

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

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

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

ID	sysprefs_wifi_disable	
References	800-53r5	<ul style="list-style-type: none"><li>AC-18, AC-18(1), AC-18(3)</li><li>AC-4</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>4.2, 12.6</li></ul>
	CCE	<ul style="list-style-type: none"><li>CCE-91084-4</li></ul>

# 11.49. 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>
```

ID	sysprefs_wifi_menu_enable	
References	800-53r5	<ul style="list-style-type: none"><li>• N/A</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>• 4.8, 12.6</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-91149-5</li></ul>

## 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 Malware Removal Tool (MRT): a technology included on all macOS systems. MRT is an agent that remediates based on automatic updates delivered from Apple. MRT will remove the malware upon receiving updated information and check for malware on restart and login.

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

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

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

<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, 10.2, 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_obscore_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-library/kodattributetypeauthenticationauthority>

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, 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.

<b>ID</b>	pwpolicy_force_password_change	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• IA-5(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>• 5.2</li> </ul>

## 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 peripherals before establishing a connection

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

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

<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>	

# 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

# 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: \*  
sysprefs\_filevault\_enforce

In macOS 11 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).



FileVault currently only uses password-based authentication and cannot be done using a smartcard or any other type of multi-factor authentication.

## 15.2. Packet Filter (pf) Supplemental

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

- `os_firewall_default_deny_require`

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

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

Below is a script that configures ALF and the PF firewall to meet the requirements defined in NIST SP 800-53 (Rev. 5). The script will make sure the application layer firewall is enabled, set logging to "detailed", set built-in signed applications to automatically receive incoming connections, and set downloaded signed applications to automatically receive incoming connections. It will then create

a custom rule set and copy `com.apple.pfctl.plist` from `/System/Library/LaunchDaemons/` into the `/Library/LaunchDaemons` folder and name it `800-53.pfctl.plist`. This is done to not conflict with the system's pf ruleset.

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

The ruleset will block connections on the following ports:

Port	Service
548	Apple File Protocol (AFP)
1900	Bonjour
79	Finger
20, 21	File Transfer Protocol (FTP)
80	HTTP
icmp	ping
143	Internet Message Access Protocol (IMAP)
993	Internet Message Access Protocol over SSL (IMAPS)
3689	Music Sharing
5353	mDNSResponder
2049	Network File System (NFS)
49152	Optical Media Sharing
110	Post Office Protocol (POP3)
995	Post Office Protocol Secure (POP3S)
631	Printer Sharing
3031	Remote Apple Events
5900	Screen Sharing
137, 138, 138, 445	Samba (SMB)
25	Simple Mail Transfer Protocol (SMTP)
22	Secure Shell (SSH)
23	Telnet
69	Trivial File Transfer Protocol (TFTP)
540	Unix-to-Unix Copy (UUCP)

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

```
#!/bin/bash

#enabling macos application firewall
enable_macos_application_firewall () {
```

```

/usr/libexec/ApplicationFirewall/socketfilterfw --setglobalstate on
/usr/libexec/ApplicationFirewall/socketfilterfw --setloggingopt detail
/usr/libexec/ApplicationFirewall/socketfilterfw --setallowsigned on
/usr/libexec/ApplicationFirewall/socketfilterfw --setallowsignedapp on
}

#enabling pf firewall with macsec rules
enable_pf_firewall_with_macsec_rules () {
    macsec_pfctl_plist="/Library/LaunchDaemons/macsec.pfctl.plist"

    if [[ -e "$macsec_pfctl_plist" ]]; then
        echo "LaunchDaemon already exists, flushing and reloading rules..."
        pfctl -e 2> /dev/null
        pfctl -f /etc/pf.conf 2> /dev/null
        return 0
    fi

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

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

    #make pf run at system startup
    launchctl enable system/macsec.pfctl
    launchctl bootstrap system $macsec_pfctl_plist

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

# append the macsec anchors to pf.conf
configure_pf_config_add_macsec_anchors () {

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

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

```



```
}
```

```
# Create /etc/pf.anchors/macsec_pf_anchors
create_macsec_pf_anchors () {
if [[ -e /etc/pf.anchors/macsec_pf_anchors ]]; then
    echo "macsec Anchor file exists, deleting and recreating..."
    rm -f /etc/pf.anchors/macsec_pf_anchors
fi
```

```
cat > /etc/pf.anchors/macsec_pf_anchors <<'ENDCONFIG'
```

```
anchor macsec_pf_anchors
```

```
#default deny all in, allow all out and keep state
block in all
pass out all keep state
```

```
## Allow DHCP
pass in inet proto udp from port 67 to port 68
pass in inet6 proto udp from port 547 to port 546
```

```
## Allow incoming SSH
pass in proto tcp to any port 22
```

```
#apple file service --port 548-- pf firewall rule
block in log proto tcp to any port { 548 }
```

```
#bonjour component SSDP --port 1900-- pf firewall rule
block log proto udp to any port 1900
```

```
#finger --port 79-- pf firewall rule
block log proto tcp to any port 79
```

```
#ftp --ports 20 21-- pf firewall rule
block in log proto { tcp udp } to any port { 20 21 }
```

```
#http --port 80-- pf firewall rule
block in log proto { tcp udp } to any port 80
```

```
#icmp pf firewall rule
block in log proto icmp
```

```
#imap --port 143-- pf firewall rule
block in log proto tcp to any port 143
```

```
#imaps --port 993-- pf firewall rule
block in log proto tcp to any port 993
```

```
#iTunes sharing --port 3689-- pf firewall rule
```

```

block log proto tcp to any port 3689

#mDNSResponder --port 5353-- pf firewall rule
block log proto udp to any port 5353

#nfs --port 2049-- pf firewall rule
block log proto tcp to any port 2049

#optical drive sharing --port 49152-- pf firewall rule
block log proto tcp to any port 49152

#pop3 --port 110-- pf firewall rule
block in log proto tcp to any port 110

#pop3s --port 995-- pf firewall rule
block in log proto tcp to any port 995

#remote apple events --port 3031-- pf firewall rule
block in log proto tcp to any port 3031

#screen_sharing --port 5900-- pf firewall rule
block in log proto tcp to any port 5900
#allow screen sharing from localhost while tunneled via SSH
pass in quick on lo0 proto tcp from any to any port 5900

#smb --ports 139 445 137 138-- pf firewall rule
block in log proto tcp to any port { 139 445 }
block in log proto udp to any port { 137 138 }

#smtp --port 25-- pf firewall rule
block in log proto tcp to any port 25

#telnet --port 23-- pf firewall rule
block in log proto { tcp udp } to any port 23

#tftp --port 69-- pf firewall rule
block log proto { tcp udp } to any port 69

#uucp --port 540-- pf firewall rule
block log proto tcp to any port 540

ENDCONFIG
}

####

enable_macos_application_firewall
create_macsec_pf_anchors
configure_pf_config_add_macsec_anchors
enable_pf_firewall_with_macsec_rules

```

## 15.3. Password Policy Supplemental

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

- `pwpolicy_lower_case_character_enforce`
- `pwpolicy_upper_case_character_enforce`
- `pwpolicy_account_inactivity_enforce`
- `pwpolicy_minimum_lifetime_enforce`

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

- Enforcing at least 1 lowercase character
- Enforcing at least 1 uppercase character
- Disabling an account after 35 days of inactivity
- Password minimum lifetime

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

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>policyCategoryAuthentication</key>
  <array>
    <dict>
      <key>policyContent</key>
      <string>(policyAttributeFailedAuthentications &lt;
policyAttributeMaximumFailedAuthentications) OR (policyAttributeCurrentTime &gt;
(policyAttributeLastFailedAuthenticationTime + autoEnableInSeconds))</string>
      <key>policyIdentifier</key>
      <string>Authentication Lockout</string>
      <key>policyParameters</key>
      <dict>
        <key>autoEnableInSeconds</key>
        <integer>300</integer>
        <key>policyAttributeMaximumFailedAuthentications</key>
        <integer>3</integer>
      </dict>
    </dict>
  </array>
</dict>
<dict>
  <key>policyContent</key>
  <string>policyAttributeLastAuthenticationTime &gt; policyAttributeCurrentTime
- (policyAttributeInactiveDays * 24 * 60 * 60)</string>
  <key>policyIdentifier</key>
```

```

    <string>Inactive Account</string>
    <key>policyParameters</key>
    <dict>
      <key>policyAttributeInactiveDays</key>
      <integer>35</integer>
    </dict>
  </dict>
</array>
<key>policyCategoryPasswordChange</key>
<array>
  <dict>
    <key>policyContent</key>
    <string>policyAttributeCurrentTime &gt; policyAttributeLastPasswordChangeTime
+ (policyAttributeExpiresEveryNDays * 24 * 60 * 60)</string>
    <key>policyIdentifier</key>
    <string>Password Expires after 60 days</string>
    <key>policyParameters</key>
    <dict>
      <key>policyAttributeExpiresEveryNDays</key>
      <integer>60</integer>
    </dict>
  </dict>
</array>
<key>policyCategoryPasswordContent</key>
<array>
  <dict>
    <key>policyContent</key>
    <string>policyAttributePassword matches '(.*[A-Z].*){1,}+'</string>
    <key>policyIdentifier</key>
    <string>Must have at least 1 uppercase letter</string>
    <key>policyParameters</key>
    <dict>
      <key>minimumAlphaCharactersUpperCase</key>
      <integer>1</integer>
    </dict>
  </dict>
  <dict>
    <key>policyContent</key>
    <string>policyAttributeLastPasswordChangeTime &lt; policyAttributeCurrentTime
- (policyAttributeMinimumLifetimeHours * 60 * 60)</string>
    <key>policyIdentifier</key>
    <string>Minimum Password Lifetime</string>
    <key>policyParameters</key>
    <dict>
      <key>policyAttributeMinimumLifetimeHours</key>
      <integer>24</integer>
    </dict>
  </dict>
</array>
<dict>
  <key>policyContent</key>
  <string>policyAttributePassword matches '.{15,}+'</string>

```

```

<key>policyIdentifier</key>
<string>Must be at least 15 characters</string>
<key>policyParameters</key>
<dict>
  <key>minimumLength</key>
  <integer>15</integer>
</dict>
</dict>
<dict>
  <key>policyContent</key>
  <string>policyAttributePassword matches '(.*[0-9].*){1,}+'</string>
  <key>policyIdentifier</key>
  <string>Must have at least 1 numeric value</string>
  <key>policyParameters</key>
  <dict>
    <key>minimumNumericCharacters</key>
    <integer>2</integer>
  </dict>
</dict>
<dict>
  <key>policyContent</key>
  <string>policyAttributePassword matches '(.*[a-z].*){1,}+'</string>
  <key>policyIdentifier</key>
  <string>Must have at least 1 lowercase letter</string>
  <key>policyParameters</key>
  <dict>
    <key>minimumAlphaCharactersLowerCase</key>
    <integer>1</integer>
  </dict>
</dict>
<dict>
  <key>policyContent</key>
  <string>policyAttributePassword matches '(.*[A-Za-z].*){1,}+'</string>
  <key>policyIdentifier</key>
  <string>Must have at least 1 Letter</string>
  <key>policyParameters</key>
  <dict>
    <key>minimumAlphaCharacters</key>
    <integer>1</integer>
  </dict>
</dict>
<dict>
  <key>policyContent</key>
  <string>policyAttributePassword matches '(.^[a-zA-Z0-9].*){1,}+'</string>
  <key>policyIdentifier</key>
  <string>Must have at least 1 special characters</string>
  <key>policyParameters</key>
  <dict>
    <key>minimumSymbols</key>
    <integer>1</integer>
  </dict>
</dict>

```

```

    </dict>
    <dict>
      <key>policyContent</key>
      <string>none policyAttributePasswordHashes in
policyAttributePasswordHistory</string>
      <key>policyIdentifier</key>
      <string>Cannot match the last 5 passwords</string>
      <key>policyParameters</key>
      <dict>
        <key>policyAttributePasswordHistoryDepth</key>
        <integer>5</integer>
      </dict>
    </dict>
  </array>
</dict>
</plist>

```

Run the following command to load the new policy file, substituting the path to the file in place of "\$pwpolicy\_file".

```
/usr/bin/pwpolicy setaccountpolicies $pwpolicy_file
```



If directory services is being utilized, password policies should come from the domain.

## 15.4. Smartcard Supplemental

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

- auth\_ssh\_password\_authentication\_disable
- auth\_smartcard\_enforce
- auth\_smartcard\_certificate\_trust\_enforce\_moderate
- auth\_smartcard\_certificate\_trust\_enforce\_high
- auth\_smartcard\_allow
- auth\_pam\_sudo\_smartcard\_enforce
- auth\_pam\_su\_smartcard\_enforce
- auth\_pam\_login\_smartcard\_enforce

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

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

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

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

## Smartcard Pairing

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

## Smartcard Attribute Mapping

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

## Smartcard Management in macOS

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

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

Key	Type	Value
checkCertificateTrust	int	Valid values are 0-3: <ul style="list-style-type: none"> <li>• 0: certificate trust check is turned off</li> <li>• 1: certificate trust check is turned on. Standard validity check is being performed but this does not include additional revocation checks.</li> <li>• 2: certificate trust check is turned on, and a soft revocation check is performed. Until the certificate is explicitly rejected by CRL/OCSP, it is considered valid. This implies that unavailable/unreachable CRL/OCSP allows this check to succeed.</li> <li>• 3: certificate trust check is turned on, plus a hard revocation check is performed. Unless CRL/OCSP explicitly states that "this certificate is OK", the certificate is considered invalid. This is the most secure value for this setting.</li> </ul>
oneCardPerUser	bool	If true, a user can pair with only one smartcard, although existing pairings will be allowed if already set up.
enforceSmartCard	bool	If true, a user can only login or authenticate with a smartcard.
tokenRemovalAction	int	If 1, the screen saver will automatically when the smartcard is removed.
allowUnmappedUsers	int	If 1, allows users who are in a directory group to be exempt from smartcard-only enforcement. The group allowed for exemption is defined in /private/etc/SmartcardLogin.plist

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

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



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

## Trusted Authorities

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



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

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

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

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

## Smartcard Enforcement Exemption

### Group Exemption

Starting in macOS 10.15, enforcement on a system can be granularly configured by adding a field to `/private/etc/SmartcardLogin.plist`. The `NotEnforcedGroup` can be added to the file to list a Directory group that will not be included in smartcard enforcement. In order to activate this feature, `enforceSmartCard` and `allowUnmappedUsers` must be applied via a configuration profile (`com.apple.security.smartcard`).

To configure the `NotEnforcedGroup`, the `SmartcardLogin.plist` should be minimally configured as follows:

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

Once a system is configured for the `NotEnforcedGroup` a user can be added to the assigned group by running the following:

```
/usr/sbin/dseditgroup -o edit -a <exempt_user> -t user <notenforcegroup>
```

## User Exemption

Alternatively, if a single user needs to be exempt for a period of time, `kDSNativeAttrTypePrefix:SmartCardEnforcement` can be set in the user's Open Directory record. The following values can be set:

- 0 - The system default is respected.
- 1 - Smartcard enforcement is enabled.
- 2 - Smartcard enforcement is disabled.



In Active Directory environments, the value of the `userAccountControl` attribute is respected.

Run the following command to set the exemption when booted from macOS:

```
/usr/bin/dscl . -append /Users/<username> SmartCardEnforcement 2
```

Run the following command to set the exemption when booted from Recovery:

```
/usr/bin/defaults write /Volumes/Macintosh\
HD/var/db/dslocal/nodes/Default/users/<username> SmartCardEnforcement -array-add 2
```



When booted to recovery on an Apple Silicon Mac, run the following after setting the exemption. `/usr/sbin/diskutil apfs updatePreboot /Volumes/Macintosh\ HD`

## Temporary Exemption

On an Apple Silicon Mac, if a temporary exemption is needed, `security filevault skip-sc-enforcement` will disable smartcard enforcement on next boot only.

Run the following command to set the temporary exemption when booted from Recovery:

```
/usr/bin/security filevault skip-sc-enforcement <data volume UUID> set
```

To obtain the `data volume UUID` run the following:

```
/usr/sbin/diskutil apfs listGroups | /usr/bin/awk -F: '/ Data/ { getline; gsub(/
/, ""); print $2}'
```

## Pluggable Authentication Module (PAM)

Terminal sessions in macOS can be configured for smartcard enforcement by modifying the PAM modules for `sudo`, `su`, and `login`.

```
/etc/pam.d/sudo
# sudo: auth account password session
auth      sufficient      pam_smartcard.so
auth      required        pam_opendirectory.so
auth      required        pam_deny.so
account    required        pam_permit.so
password   required        pam_deny.so
session    required        pam_permit.so
```

```
/etc/pam.d/su
```

```
# su: auth account password session
```

```
auth      sufficient    pam_smartcard.so
```

```
auth      required      pam_rootok.so
```

```
auth      required      pam_group.so no_warn group=admin,wheel ruser root_only
```

```
fail_safe
```

```
account   required      pam_permit.so
```

```
account   required      pam_opendirectory.so no_check_shell
```

```
password  required      pam_opendirectory.so
```

```
session   required      pam_launchd.so
```

```
/etc/pam.d/login
```

```
# login: auth account password session
```

```
auth      sufficient    pam_smartcard.so
```

```
auth      optional      pam_krb5.so use_kcminit
```

```
auth      optional      pam_ntlm.so try_first_pass
```

```
auth      optional      pam_mount.so try_first_pass
```

```
auth      required      pam_opendirectory.so try_first_pass
```

```
auth      required      pam_deny.so
```

```
account   required      pam_nologin.so
```

```
account   required      pam_opendirectory.so
```

```
password  required      pam_opendirectory.so
```

```
session   required      pam_launchd.so
```

```
session   required      pam_uwtmp.so
```

```
session   optional      pam_mount.so
```