



# Apple Security Baseline

## macOS 14.0 Sonoma ***Security Configuration - CIS Level 1***

Sonoma Guidance, Revision 1.0 (2023-09-21)

# Table of Contents

1. Foreword .....	1
2. Scope .....	2
3. Authors .....	3
4. Acronyms and Definitions .....	4
5. Applicable Documents .....	6
5.1. Government Documents .....	6
5.2. Non-Government Documents .....	6
6. Auditing .....	7
6.1. Configure Audit Log Files to Not Contain Access Control Lists .....	7
6.2. Configure Audit Log Folder to Not Contain Access Control Lists .....	8
6.3. Enable Security Auditing .....	8
6.4. Configure Audit_Control to Not Contain Access Control Lists .....	10
6.5. Configure Audit_Control Group to Wheel .....	10
6.6. Configure Audit_Control Owner to Mode 440 or Less Permissive .....	11
6.7. Configure Audit_Control Owner to Root .....	12
6.8. Configure Audit Log Files Group to Wheel .....	12
6.9. Configure Audit Log Files to Mode 440 or Less Permissive .....	13
6.10. Configure Audit Log Files to be Owned by Root .....	14
6.11. Configure Audit Log Folders Group to Wheel .....	14
6.12. Configure Audit Log Folders to be Owned by Root .....	15
6.13. Configure Audit Log Folders to Mode 700 or Less Permissive .....	16
6.14. Configure Audit Retention to 60d OR 5G .....	17
7. macOS .....	18
7.1. Disable AirDrop .....	18
7.2. Must Use an Approved Antivirus Program .....	19
7.3. Enable Authenticated Root .....	19
7.4. Enforce Installation of XProtect Remediator and Gatekeeper Updates Automatically .....	20
7.5. Enable Firewall Logging .....	21
7.6. Enable Gatekeeper .....	23
7.7. Remove Guest Folder if Present .....	23
7.8. Secure User's Home Folders .....	24
7.9. Disable the Built-in Web Server .....	25
7.10. Configure Install.log Retention to 365 .....	26
7.11. Enforce Enrollment in Mobile Device Management .....	27
7.12. Enable Apple Mobile File Integrity .....	28
7.13. Disable Network File System Service .....	28
7.14. Enforce On Device Dictation .....	29
7.15. Remove Password Hint From User Accounts .....	30

7.16. Disable Power Nap .....	31
7.17. Disable Root Login .....	32
7.18. Ensure Advertising Privacy Protection in Safari Is Enabled .....	32
7.19. Disable Automatic Opening of Safe Files in Safari .....	33
7.20. Ensure Pop-Up Windows are Blocked in Safari .....	34
7.21. Ensure Prevent Cross-site Tracking in Safari Is Enabled .....	35
7.22. Ensure Show Full Website Address in Safari Is Enabled .....	36
7.23. Ensure Show Safari shows the Status Bar is Enabled .....	37
7.24. Ensure Warn When Visiting A Fraudulent Website in Safari Is Enabled .....	37
7.25. Enable Show All Filename Extensions .....	38
7.26. Ensure System Integrity Protection is Enabled .....	39
7.27. Ensure Software Update Deferment Is Less Than or Equal to 30 Days .....	40
7.28. Configure Sudo Timeout Period to 0 .....	41
7.29. Configure Sudoers Timestamp Type .....	42
7.30. Ensure Appropriate Permissions Are Enabled for System Wide Applications .....	42
7.31. Ensure Secure Keyboard Entry Terminal.app is Enabled .....	43
7.32. Ensure Time Offset Within Limits .....	44
7.33. Disable Login to Other User's Active and Locked Sessions .....	45
7.34. Ensure No World Writable Files Exist in the System Folder .....	45
8. Password Policy .....	47
8.1. Limit Consecutive Failed Login Attempts to 5 .....	47
8.2. Set Account Lockout Time to 15 Minutes .....	48
8.3. Prohibit Password Reuse for a Minimum of 15 Generations .....	49
8.4. Restrict Maximum Password Lifetime to 365 Days .....	50
8.5. Require a Minimum Password Length of 15 Characters .....	51
9. System Settings .....	53
9.1. Disable Airplay Receiver .....	53
9.2. Disable Unattended or Automatic Logon to the System .....	54
9.3. Enable Bluetooth Menu .....	55
9.4. Disable Bluetooth Sharing .....	56
9.5. Disable CD/DVD Sharing .....	57
9.6. Enforce Critical Security Updates to be Installed .....	57
9.7. Enforce FileVault .....	58
9.8. Enable macOS Application Firewall .....	59
9.9. Enable Firewall Stealth Mode .....	60
9.10. Disable Guest Access to Shared SMB Folders .....	61
9.11. Disable the Guest Account .....	62
9.12. Enforce macOS Updates are Automatically Installed .....	63
9.13. Disable Internet Sharing .....	64
9.14. Configure Login Window to Show A Custom Message .....	65
9.15. Configure Login Window to Prompt for Username and Password .....	66

9.16. Disable Password Hints .....	66
9.17. Disable Personalized Advertising .....	67
9.18. Disable Printer Sharing .....	68
9.19. Disable Remote Apple Events .....	69
9.20. Disable Remote Management .....	70
9.21. Disable Screen Sharing and Apple Remote Desktop .....	70
9.22. Enforce Session Lock After Screen Saver is Started .....	71
9.23. Enforce Screen Saver Timeout .....	72
9.24. Disable Server Message Block Sharing .....	73
9.25. Enforce Software Update App Update Updates Automatically .....	74
9.26. Enforce Software Update Downloads Updates Automatically .....	75
9.27. Enforce Software Update Automatically .....	76
9.28. Ensure Software Update is Updated and Current .....	76
9.29. Disable SSH Server for Remote Access Sessions .....	77
9.30. Require Administrator Password to Modify System-Wide Preferences .....	78
9.31. Ensure Time Machine Volumes are Encrypted .....	79
9.32. Configure macOS to Use an Authorized Time Server .....	80
9.33. Enforce macOS Time Synchronization .....	81
9.34. Ensure Wake for Network Access Is Disabled .....	82
9.35. Enable Wifi Menu .....	83
10. Supplemental .....	84
10.1. CIS Manual Recommendations .....	84
10.2. Out of Scope Supplemental .....	85
10.3. FileVault Supplemental .....	86
10.4. Packet Filter (pf) Supplemental .....	88
10.5. Password Policy Supplemental .....	95
10.6. Smartcard Supplemental .....	98

# Chapter 1. Foreword

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

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

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

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

# Chapter 2. Scope

This guide describes the actions to take when securing a macOS 14.0 Sonoma system against the CIS Apple macOS 14.0 Sonoma v.1.0.0 Level 1 security benchmark.

# Chapter 3. Authors

Security configuration tailored by:

Dennis Rietveld	Rietveld ICT
-----------------	--------------

**macOS Security Compliance Project**

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

Edward Byrd	Center for Internet Security
Ron Colvin	Center for Internet Security
Allen Golbig	Jamf

# Chapter 4. Acronyms and Definitions

Table 1. Acronyms and Abbreviations

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



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

*Table 2. Definitions*

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

# Chapter 5. Applicable Documents

## 5.1. Government Documents

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

Document Number or Descriptor	Document Title
<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>
<a href="#">NIST Special Publication 800-219</a>	<i>NIST Special Publication 800-219 Rev 1</i>

Table 4. Defense Information Systems Agency (DISA)

Document Number or Descriptor	Document Title
<a href="#">STIG Ver 1, Rel 2</a>	<i>Apple macOS 13 (Ventura) STIG</i>

Table 5. Cybersecurity Maturity Model Certification (CMMC)

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

Table 6. Committee on National Security Systems (CNSS)

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

## 5.2. Non-Government Documents

Table 7. Apple

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

Table 8. Center for Internet Security

Document Number or Descriptor	Document Title
<a href="#">Apple macOS 13.0</a>	<i>CIS Apple macOS 13.0 Benchmark version 1.1.0</i>

# Chapter 6. Auditing

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



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



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

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

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

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

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

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

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

**Remediation Description**

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

```
/bin/chmod -RN /var/audit
```

ID	audit_acls_files_configure	
References	800-53r5	• AU-9
	CIS Benchmark	• 3.5 (level 1)
	CIS Controls V8	• 3.3
	CCE	• CCE-92701-2

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

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

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

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

```
/bin/ls -lde /var/audit | /usr/bin/awk '{print $1}' | /usr/bin/grep -c ":"
```

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

**Remediation Description**

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

```
/bin/chmod -N /var/audit
```

ID	audit_acls_folders_configure	
References	800-53r5	• AU-9
	CIS Benchmark	• 3.5 (level 1)
	CIS Controls V8	• 3.3
	CCE	• CCE-92702-0

## 6.3. Enable Security Auditing

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

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

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

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



Security auditing is NOT enabled by default on macOS Sonoma.

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

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

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

### Remediation Description

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

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

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

ID	audit_auditd_enabled	
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</li><li>• 8.5</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-92704-6</li></ul>

# 6.4. Configure Audit\_Control to Not Contain Access Control Lists

/etc/security/audit\_control *MUST* not contain Access Control Lists (ACLs).

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

```
/bin/ls -le /etc/security/audit_control | /usr/bin/awk '{print $1}' | /usr/bin/grep -c ":",
```

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

**Remediation Description**

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

```
/bin/chmod -N /etc/security/audit_control
```

ID	audit_control_acls_configure	
References	800-53r5	• AU-9
	CIS Benchmark	• 3.5 (level 1)
	CIS Controls V8	• 3.3
	CCE	• CCE-92706-1

# 6.5. Configure Audit\_Control Group to Wheel

/etc/security/audit\_control *MUST* have the group set to wheel.

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

```
/bin/ls -dn /etc/security/audit_control | /usr/bin/awk '{print $4}'
```

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

**Remediation Description**

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

```
/usr/bin/chgrp wheel /etc/security/audit_control
```

<b>ID</b>	audit_control_group_configure	
<b>References</b>	<b>800-53r5</b>	• AU-9
	<b>CIS Benchmark</b>	• 3.5 (level 1)
	<b>CIS Controls V8</b>	• 3.3
	<b>CCE</b>	• CCE-92707-9

## 6.6. Configure Audit\_Control Owner to Mode 440 or Less Permissive

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

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

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

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

### Remediation Description

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

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

<b>ID</b>	audit_control_mode_configure	
<b>References</b>	<b>800-53r5</b>	• AU-9
	<b>CIS Benchmark</b>	• 3.5 (level 1)
	<b>CIS Controls V8</b>	• 3.3
	<b>CCE</b>	• CCE-92708-7

## 6.7. Configure Audit\_Control Owner to Root

/etc/security/audit\_control *MUST* have the owner set to root.

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

```
/bin/ls -dn /etc/security/audit_control | /usr/bin/awk '{print $3}'
```

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

**Remediation Description**

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

```
/usr/sbin/chown root /etc/security/audit_control
```

ID	audit_control_owner_configure	
References	800-53r5	• AU-9
	CIS Benchmark	• 3.5 (level 1)
	CIS Controls V8	• 3.3
	CCE	• CCE-92709-5

## 6.8. Configure Audit Log Files Group to Wheel

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

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

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

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

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

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

**Remediation Description**



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

```
/usr/bin/chgrp -R wheel /var/audit/*
```

ID	audit_files_group_configure	
References	800-53r5	• AU-9
	CIS Benchmark	• 3.5 (level 1)
	CIS Controls V8	• 3.3
	CCE	• CCE-92712-9

## 6.9. Configure Audit Log Files to Mode 440 or Less Permissive

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

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

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

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

### Remediation Description

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

```
/bin/chmod 440 /var/audit/*
```

ID	audit_files_mode_configure
----	----------------------------

<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-92713-7</li></ul>

## 6.10. Configure Audit Log Files to be Owned by Root

Audit log files *MUST* be owned by root.

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

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

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

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

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

**Remediation Description**

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

```
/usr/sbin/chown -R root /var/audit/*
```

<b>ID</b>	audit_files_owner_configure	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"><li>• AU-9</li></ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"><li>• 3.5 (level 1)</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 3.3</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-92714-5</li></ul>

## 6.11. Configure Audit Log Folders Group to Wheel

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

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

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

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

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

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

**Remediation Description**

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

```
/usr/bin/chgrp wheel /var/audit
```

ID	audit_folder_group_configure	
References	800-53r5	• AU-9
	CIS Benchmark	• 3.5 (level 1)
	CIS Controls V8	• 3.3
	CCE	• CCE-92724-4

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

Audit log folders *MUST* be owned by root.

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

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

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

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

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

### Remediation Description

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

```
/usr/sbin/chown root /var/audit
```

ID	audit_folder_owner_configure	
References	800-53r5	• AU-9
	CIS	• 3.5 (level 1)
	Benchmark	
	CIS Controls V8	• 3.3
	CCE	• CCE-92725-1

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

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

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

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

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

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

### Remediation Description

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

```
/bin/chmod 700 /var/audit
```

ID	audit_folders_mode_configure
----	------------------------------

<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-92726-9</li></ul>

## 6.14. Configure Audit Retention to 60d OR 5G

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

When "expire-after" is set to "60d OR 5G", the audit service will not delete audit logs until the log data criteria is met.

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

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

If the result is not **60d OR 5G**, 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 5G/'  
/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>• 3.4 (level 1)</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 8.1</li><li>• 8.3</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-92730-1</li></ul>

# Chapter 7. 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.

## 7.1. Disable AirDrop

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

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

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

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

**Remediation Description**

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

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

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

ID	os_airdrop_disable	
References	800-53r5	<ul style="list-style-type: none"><li>• AC-20</li><li>• AC-3</li><li>• CM-7, CM-7(1)</li><li>• 2.3.1.1 (level 1)</li></ul>
	CIS Benchmark	
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li><li>• 6.7</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-92756-6</li></ul>

## 7.2. Must Use an Approved Antivirus Program

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

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

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

```
/bin/launchctl list | /usr/bin/grep -cE "(com.apple.XprotectFramework.PluginService
$|com.apple.XProtect.daemon.scan$)"
```

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

**Remediation Description**

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

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

ID	os_anti_virus_installed	
References	800-53r5	<ul style="list-style-type: none"><li>• N/A</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 5.10 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 10.5</li><li>• 10.1</li><li>• 10.2</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-92758-2</li></ul>

## 7.3. 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.



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

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

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

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

#### Remediation Description

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

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



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

ID	os_authenticated_root_enable	
References	800-53r5	<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>
	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>• 3.6</li><li>• 3.11</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-92764-0</li></ul>

## 7.4. Enforce Installation of XProtect Remediator and Gatekeeper Updates Automatically

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

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

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





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

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

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

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

### Remediation Description

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

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

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

ID	os_config_data_install_enforce	
References	800-53r5	<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>• 1.6 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 7.3</li><li>• 7.4</li><li>• 7.7</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-92776-4</li></ul>

## 7.5. 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</li><li>• 8.2</li><li>• 8.5</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-92793-9</li></ul>

# 7.6. 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>• 2.6.5 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 10.1</li><li>• 10.2</li><li>• 10.5</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-92795-4</li></ul>

# 7.7. Remove Guest Folder if Present

The guest folder *MUST* be deleted if present.

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

```
/bin/ls /Users/ | /usr/bin/grep -c "Guest"
```

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

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

```
/bin/rm -Rf /Users/Guest
```

ID	os_guest_folder_removed	
References	800-53r5	• N/A
	CIS Benchmark	• 5.9 (level 1)
	CIS Controls V8	• 4.1
	CCE	• CCE-92798-8

## 7.8. Secure User’s Home Folders

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

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

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

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

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

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

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

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

<b>ID</b>	os_home_folders_secure	
<b>References</b>	<b>800-53r5</b>	• AC-6
	<b>CIS Benchmark</b>	• 5.1.1 (level 1)
	<b>CIS Controls V8</b>	• 3.3
	<b>CCE</b>	• CCE-92804-4

## 7.9. Disable the Built-in Web Server

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



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

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

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

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

### Remediation Description

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

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

<b>ID</b>	os_httpd_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>• 4.2 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.1</li> <li>• 4.8</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-92805-1</li> </ul>

## 7.10. Configure Install.log Retention to 365

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

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

```
/usr/sbin/aslmanager -dd 2>&1 | /usr/bin/awk '/\/var\/log\/install.log/ {count++}
/Processing module com.apple.install/,/Finished/ { for (i=1;i<=NR;i++) { if ($i ==
"TTL" && $(i+2) >= 365) { ttl="True" }; if ($i == "MAX") {max="True"}}} END{if (count
> 1) { print "Multiple config files for /var/log/install, manually remove"} 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.

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

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• AU-11</li> <li>• AU-4</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 3.3 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 8.1</li> <li>• 8.3</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-92811-9</li> </ul>

## 7.11. 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>• 1.8 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.1</li><li>• 5.1</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-92824-2</li></ul>

## 7.12. 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=""
```

ID	os_mobile_file_integrity_enable	
References	800-53r5	<ul style="list-style-type: none"><li>• N/A</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 5.1.3 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 2.3</li><li>• 2.6</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-92828-3</li></ul>

## 7.13. 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" => disabled'
```

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

**Remediation Description**

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

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

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

ID	os_nfsd_disable	
References	800-53r5	<ul style="list-style-type: none"><li>• AC-17</li><li>• AC-3</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 4.3 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-92831-7</li></ul>

## 7.14. Enforce On Device Dictation

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

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

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

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

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

**Remediation Description**

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

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

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

ID	os_on_device_dictation_enforce	
References	800-53r5	<ul style="list-style-type: none"><li>• AC-20</li><li>• CM-7, CM-7(1)</li><li>• SC-7(10)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.18.1 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-92841-6</li></ul>

## 7.15. 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
```

ID	os_password_hint_remove
----	-------------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"><li>• IA-6</li></ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"><li>• 2.11.1 (level 1)</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 5.2</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-92844-0</li></ul>

## 7.16. Disable Power Nap

Power Nap *MUST* be disabled.



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

The following Macs support Power Nap:

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

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

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

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

**Remediation Description**

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

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

<b>ID</b>	os_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>• 2.9.2 (level 1)</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-92853-1</li></ul>

## 7.17. 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>• 5.4</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-92875-4</li></ul>

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

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

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

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

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

**Remediation Description**

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

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

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

ID	os_safari_advertising_privacy_protection_enable	
References	800-53r5	• N/A
	CIS Benchmark	• 6.3.6 (level 1)
	CIS Controls V8	• 9.1
	CCE	• CCE-92876-2

## 7.19. 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	• 6.3.1 (level 1)
	CIS Controls V8	• 9.1
		• 9.6
	CCE	• CCE-92877-0

## 7.20. Ensure Pop-Up Windows are Blocked in Safari

Safari *MUST* be configured to block Pop-Up windows.

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

```
/usr/bin/profiles -P -o stdout | /usr/bin/grep -c 'safariAllowPopups = 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>safariAllowPopups</key>
<false/>
```

ID	os_safari_popups_disabled
----	---------------------------

References	800-53r5	<ul style="list-style-type: none"><li>• N/A</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 6.3.9 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 9.1</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-93014-9</li></ul>

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

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

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

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

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

**Remediation Description**

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

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

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

ID	os_safari_prevent_cross-site_tracking_enable
----	----------------------------------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 6.3.4 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 9.1</li> <li>• 9.3</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-92878-8</li> </ul>

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

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

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

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

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

### Remediation Description

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

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

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

<b>ID</b>	os_safari_show_full_website_address_enable	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 6.3.7 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 9.1</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-92879-6</li> </ul>



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

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

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

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

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

### Remediation Description

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

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

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

ID	os_safari_show_status_bar_enabled	
References	800-53r5	• N/A
	CIS Benchmark	• 6.3.11 (level 1)
	CIS Controls V8	• 9.1
	CCE	• CCE-93015-6

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

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

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

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

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

### Remediation Description

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

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

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

ID	os_safari_warn_fraudulent_website_enable	
References	800-53r5	• N/A
	CIS Benchmark	• 6.3.3 (level 1)
	CIS Controls V8	• 9.1
		• 9.3
	CCE	• CCE-92880-4

## 7.25. 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=$( /usr/sbin/scutil <<< "show State:/Users/ConsoleUser" |
/usr/bin/awk '/Name :/ && ! /loginwindow/ { print $3 }' )
```

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

```
/usr/bin/sudo -u "$CURRENT_USER" /usr/bin/defaults 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	• 6.1.1 (level 1)
	CIS Controls V8	• 2.3
	CCE	• CCE-92888-7

## 7.26. Ensure System Integrity Protection is Enabled

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

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



SIP is enabled by default in macOS.

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

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

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

### Remediation Description

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

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



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

<b>ID</b>	os_sip_enable	
<b>References</b>	<b>800-53r5</b> <ul style="list-style-type: none"> <li>• AC-3</li> <li>• AU-9, AU-9(3)</li> <li>• CM-5, CM-5(6)</li> <li>• SC-4</li> <li>• SI-2</li> <li>• SI-7</li> </ul>	
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 5.1.2 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 2.3</li> <li>• 2.6</li> <li>• 10.5</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-92889-5</li> </ul>

## 7.27. Ensure Software Update Deferral Is Less Than or Equal to 30 Days

Software updates *MUST* be deferred for 30 days or less.

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

```
/usr/bin/osascript -l JavaScript << EOS
function run() {
    let timeout = ObjC.unwrap(
$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
.objectForKey('enforcedSoftwareUpdateDelay')) || 0
    if ( timeout <= 30 ) {
        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>enforcedSoftwareUpdateDelay</key>
<integer>30</integer>
```

ID	os_software_update_deferral	
References	800-53r5	<ul style="list-style-type: none"><li>• N/A</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 1.7 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 7.3</li><li>• 7.4</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-92893-7</li></ul>

## 7.28. Configure Sudo Timeout Period to 0

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

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

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

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

### Remediation Description

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

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

ID	os_sudo_timeout_configure	
References	800-53r5	<ul style="list-style-type: none"><li>• N/A</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 5.4 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.3</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-92908-3</li></ul>

## 7.29. Configure Sudoers Timestamp Type

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

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

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

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

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

### Remediation Description

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

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

ID	os_sudoers_timestamp_type_configure	
References	800-53r5	<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>• 5.5 (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-92909-1</li></ul>

## 7.30. 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	
References	800-53r5	• N/A
	CIS Benchmark	• 5.1.5 (level 1)
	CIS Controls V8	• 3.3
	CCE	• CCE-92911-7

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

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

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

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

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

### Remediation Description

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

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

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

ID	os_terminal_secure_keyboard_enable	
References	800-53r5	<ul style="list-style-type: none"><li>• N/A</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 6.4.1 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.8</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-92912-5</li></ul>

## 7.32. 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}')
```

ID	os_time_offset_limit_configure	
References	800-53r5	<ul style="list-style-type: none"><li>• N/A</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.3.2.2 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 8.4</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-92915-8</li></ul>



# 7.33. 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	• IA-2, IA-2(5)
	CIS Benchmark	• 5.7 (level 1)
	CIS Controls V8	• 4.3
	CCE	• CCE-92919-0

# 7.34. 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 "downloadDir" | /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 $( /usr/bin/find /System/Volumes/Data/System -type d -perm
-2 | /usr/bin/grep -v "downloadDir" ); do
  /bin/chmod -R o-w "$sysPermissions"
done
```

ID	os_world_writable_system_folder_configure	
References	800-53r5	• N/A
	CIS Benchmark	• 5.1.6 (level 1)
	CIS Controls V8	• 3.3
	CCE	• CCE-92924-0

# Chapter 8. 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.

## 8.1. Limit Consecutive Failed Login Attempts to 5

The macOS *MUST* be configured to limit the number of failed login attempts to a maximum of 5. When the maximum number of failed attempts is reached, the account *MUST* be locked for a period of time after.

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

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

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

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

### Remediation Description

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

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

```
<key>maxFailedAttempts</key>
```

```
<integer>5</integer>
```

ID	pwpolicy_account_lockout_enforce	
References	800-53r5	• AC-7
	CIS Benchmark	• 5.2.1 (level 1)
	CIS Controls V8	• 6.2
	CCE	• CCE-92927-3

## 8.2. Set Account Lockout Time to 15 Minutes

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

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

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

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

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

### Remediation Description

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

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

```
<key>minutesUntilFailedLoginReset</key>  
<integer>15</integer>
```

ID	pwpolicy_account_lockout_timeout_enforce
----	------------------------------------------

References	800-53r5	<ul style="list-style-type: none"> <li>• AC-7</li> </ul>
	CIS Benchmark	<ul style="list-style-type: none"> <li>• 5.2.1 (level 1)</li> </ul>
	CIS Controls V8	<ul style="list-style-type: none"> <li>• 6.2</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>• CCE-92928-1</li> </ul>

## 8.3. Prohibit Password Reuse for a Minimum of 15 Generations

The macOS *MUST* be configured to enforce a password history of at least 15 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 15 previous password generations.

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



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

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

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

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

### Remediation Description

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

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

```
<key>pinHistory</key>
<integer>15</integer>
```

<b>ID</b>	pwpolicy_history_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>• 5.2.8 (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-92932-3</li> </ul>

## 8.4. Restrict Maximum Password Lifetime to 365 Days

The macOS *MUST* be configured to enforce a maximum password lifetime limit of at least 365 days.

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



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

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

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

If the result is not 365, this is a finding.

### Remediation 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>365</integer>
```

<b>ID</b>	pwpolicy_max_lifetime_enforce
-----------	-------------------------------

References	800-53r5	<ul style="list-style-type: none"> <li>• IA-5</li> </ul>
	CIS Benchmark	<ul style="list-style-type: none"> <li>• 5.2.7 (level 1)</li> </ul>
	CIS Controls V8	<ul style="list-style-type: none"> <li>• 5.3</li> </ul>
	CCE	<ul style="list-style-type: none"> <li>• CCE-92935-6</li> </ul>

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

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

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



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

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

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

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

### Remediation Description

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

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

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

ID	pwpolicy_minimum_length_enforce
----	---------------------------------

<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-92936-4</li> </ul>



# Chapter 9. System Settings

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



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

## 9.1. Disable Airplay Receiver

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

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

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

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

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

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

**Remediation Description**

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

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

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

ID	system_settings_airplay_receiver_disable
----	------------------------------------------

References	800-53r5	<ul style="list-style-type: none"><li>• CM-7, CM-7(1)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.3.1.2 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-92944-8</li></ul>

## 9.2. 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	system_settings_automatic_login_disable
----	-----------------------------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"><li>• IA-2</li><li>• IA-5(13)</li></ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"><li>• 2.12.3 (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-92947-1</li></ul>

### 9.3. Enable Bluetooth Menu

The bluetooth menu *MUST* be enabled.

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

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

If the result is not **18**, this is a finding.

**Remediation Description**

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

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

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

<b>ID</b>	system_settings_bluetooth_menu_enable	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"><li>• N/A</li></ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"><li>• 2.4.2 (level 1)</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 4.8</li><li>• 13.9</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-92950-5</li></ul>

# 9.4. Disable Bluetooth Sharing

Bluetooth Sharing *MUST* be disabled.

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



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

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

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

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

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

## Remediation Description

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

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

ID	system_settings_bluetooth_sharing_disable	
References	800-53r5	<ul style="list-style-type: none"><li>AC-18(4)</li><li>AC-3</li><li>CM-7, CM-7(1)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>2.3.3.11 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>3.3</li><li>4.1</li></ul>
	CCE	<ul style="list-style-type: none"><li>CCE-92952-1</li></ul>

# 9.5. 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 ODSSAgent; /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.ODSSAgent.plist
```

ID	system_settings_cd_dvd_sharing_disable	
References	800-53r5	• CM-7, CM-7(1)
	CIS Benchmark	• 2.3.3.1 (level 1)
	CIS Controls V8	• 4.1
		• 4.8
	CCE	• CCE-92953-9

# 9.6. Enforce Critical Security Updates to be Installed

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

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

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

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

**Remediation Description**

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

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

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

ID	system_settings_critical_update_install_enforce	
References	800-53r5	• SI-2
	CIS Benchmark	• 1.6 (level 1)
	CIS Controls V8	• 7.3
		• 7.4
		• 7.7
	CCE	• CCE-92955-4

## 9.7. Enforce FileVault

FileVault *MUST* be enforced.

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

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

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

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

Remediation Description

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

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

type:

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

ID	system_settings_filevault_enforce	
References	800-53r5	• SC-28, SC-28(1)
	CIS Benchmark	• 2.6.6 (level 1)
	CIS Controls V8	• 3.6 • 3.11
	CCE	• CCE-92957-0

## 9.8. Enable macOS Application Firewall

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

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

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

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

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

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

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

### Remediation Description

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

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

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

ID	system_settings_firewall_enable	
References	800-53r5	<ul style="list-style-type: none"><li>• AC-4</li><li>• CM-7, CM-7(1)</li><li>• SC-7, SC-7(12)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.2.1 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.5</li><li>• 13.1</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-92959-6</li></ul>

## 9.9. Enable Firewall Stealth Mode

Firewall Stealth Mode *MUST* be enabled.

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



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

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

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

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

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



```
else
  echo "false"
fi
```

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

### Remediation Description

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

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

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

ID	system_settings_firewall_stealth_mode_enable	
References	800-53r5	<ul style="list-style-type: none"><li>• CM-7, CM-7(1)</li><li>• SC-7, SC-7(16)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.2.2 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.5</li><li>• 4.8</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-92960-4</li></ul>

## 9.10. Disable Guest Access to Shared SMB Folders

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

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

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

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

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

### Remediation Description

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

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

ID	system_settings_guest_access_smb_disable	
References	800-53r5	• AC-2, AC-2(9)
	CIS Benchmark	• 2.12.2 (level 1)
	CIS Controls V8	• 3.3
	CCE	• CCE-92963-8

## 9.11. Disable the Guest Account

Guest access *MUST* be disabled.

Turning off guest access prevents anonymous users from accessing files.

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

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

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

### Remediation Description

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

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

type:

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

ID	system_settings_guest_account_disable	
References	800-53r5	• AC-2, AC-2(9)
	CIS Benchmark	• 2.12.1 (level 1)
	CIS Controls V8	• 5.2
		• 6.2
		• 6.8
	CCE	• CCE-92964-6

## 9.12. 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/>
```

<b>ID</b>	system_settings_install_macos_updates_enforce	
<b>References</b>	<b>800-53r5</b> <ul style="list-style-type: none"> <li>• N/A</li> </ul>	<b>CIS Benchmark</b> <ul style="list-style-type: none"> <li>• 1.4 (level 1)</li> </ul>
	<b>CIS Controls V8</b> <ul style="list-style-type: none"> <li>• 7.3</li> <li>• 7.4</li> </ul>	
	<b>CCE</b> <ul style="list-style-type: none"> <li>• CCE-92968-7</li> </ul>	

## 9.13. 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/>
```

<b>ID</b>	system_settings_internet_sharing_disable
-----------	------------------------------------------

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"><li>• AC-20</li><li>• AC-4</li></ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"><li>• 2.3.3.8 (level 1)</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-92971-1</li></ul>

## 9.14. Configure Login Window to Show A Custom Message

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

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

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

If the result is not **Your org message here**, this is a finding.

**Remediation Description**

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

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

```
<key>LoginwindowText</key>
<string>Your org message here</string>
```

<b>ID</b>	system_settings_loginwindow_loginwindowtext_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>• 2.10.3 (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-92975-2</li></ul>

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

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

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

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

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

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

**Remediation Description**

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

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

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

ID	system_settings_loginwindow_prompt_username_password_enforce	
References	800-53r5	• IA-2
	CIS Benchmark	• 2.10.4 (level 1)
	CIS Controls V8	• 4.1
	CCE	• CCE-92976-0

# 9.16. Disable Password Hints

Password hints *MUST* be disabled.

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

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

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

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

**Remediation Description**

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

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

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

ID	system_settings_password_hints_disable	
References	800-53r5	• IA-6
	CIS Benchmark	• 2.10.5 (level 1)
	CIS Controls V8	• 4.1
	CCE	• CCE-92978-6

## 9.17. Disable Personalized Advertising

Ad tracking and targeted ads *MUST* be disabled.

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

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

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

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

### Remediation Description

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

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

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

ID	system_settings_personalized_advertising_disable	
References	800-53r5	<ul style="list-style-type: none"><li>• AC-20</li><li>• CM-7, CM-7(1)</li><li>• SC-7(10)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.6.4 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.8</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-92979-4</li></ul>

## 9.18. Disable Printer Sharing

Printer Sharing *MUST* be disabled.

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

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

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

### Remediation Description

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

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

ID	system_settings_printer_sharing_disable
----	-----------------------------------------



<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"><li>• CM-7, CM-7(1)</li></ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"><li>• 2.3.3.4 (level 1)</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-92980-2</li></ul>

## 9.19. Disable Remote Apple Events

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

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

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


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

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

**Remediation Description**

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

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



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

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

<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"><li>• AC-17</li><li>• AC-3</li></ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"><li>• 2.3.3.7 (level 1)</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-92981-0</li></ul>

## 9.20. 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>	system_settings_remote_management_disable	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"><li>• CM-7, CM-7(1)</li></ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"><li>• 2.3.3.6 (level 1)</li></ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li><li>• 5.4</li></ul>
	<b>CCE</b>	<ul style="list-style-type: none"><li>• CCE-92982-8</li></ul>

## 9.21. Disable Screen Sharing and Apple Remote Desktop

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

disabled.

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

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

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

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

**Remediation Description**

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

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

NOTE - This will apply to the whole system

ID	system_settings_screen_sharing_disable	
References	800-53r5	<ul style="list-style-type: none"><li>• AC-17</li><li>• AC-3</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.3.3.2 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-92983-6</li></ul>

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

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

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

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

```
/usr/bin/osascript -l JavaScript << EOS
function run() {
```

```

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

```

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

### Remediation Description

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

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

```

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

```

<b>ID</b>	system_settings_screensaver_ask_for_password_delay_enforce	
<b>References</b>	<b>800-53r5</b>	<ul style="list-style-type: none"> <li>• AC-11</li> </ul>
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 2.10.2 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 4.7</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-92984-4</li> </ul>

## 9.23. Enforce Screen Saver Timeout

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

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

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

```

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

```

```
if ( timeout <= 1200 ) {  
    return("true")  
} else {  
    return("false")  
}  
}  
EOS
```

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

### Remediation Description

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

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

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

ID	system_settings_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>• 2.10.1 (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-92986-9</li></ul>

## 9.24. Disable Server Message Block Sharing

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

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

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

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

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

### Remediation Description

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

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

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

ID	system_settings_smbd_disable	
References	800-53r5	<ul style="list-style-type: none"><li>• AC-17</li><li>• AC-3</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.3.3.3 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li><li>• 5.4</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-92989-3</li></ul>

## 9.25. Enforce Software Update App Update Updates Automatically

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

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

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

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

### Remediation Description

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

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

```
<key>AutomaticallyInstallAppUpdates</key>
<true/>
```

<b>ID</b>	system_settings_software_update_app_update_enforce	
<b>References</b>	<b>800-53r5</b> <ul style="list-style-type: none"> <li>• N/A</li> </ul>	
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 1.5 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 7.3</li> <li>• 7.4</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-92990-1</li> </ul>

## 9.26. 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/>
```

<b>ID</b>	system_settings_software_update_download_enforce	
<b>References</b>	<b>800-53r5</b> <ul style="list-style-type: none"> <li>• N/A</li> </ul>	
	<b>CIS Benchmark</b>	<ul style="list-style-type: none"> <li>• 1.3 (level 1)</li> </ul>
	<b>CIS Controls V8</b>	<ul style="list-style-type: none"> <li>• 7.3</li> <li>• 7.4</li> </ul>
	<b>CCE</b>	<ul style="list-style-type: none"> <li>• CCE-92991-9</li> </ul>

# 9.27. 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	system_settings_software_update_enforce	
References	800-53r5	• SI-2(5)
	CIS Benchmark	• 1.2 (level 1)
	CIS Controls V8	• 7.3
		• 7.4
	CCE	• CCE-92992-7

# 9.28. Ensure Software Update is Updated and Current

Make sure Software Update is updated and current.



Automatic fix can cause unplanned restarts and may lose work.

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

```
softwareupdate_date_epoch=$(/bin/date -j -f "%Y-%m-%d" "$(/usr/bin/defaults read /Library/Preferences/com.apple.SoftwareUpdate.plist LastFullSuccessfulDate | /usr/bin/awk '{print $1}')" "+%s")
```



```
thirty_days_epoch=$(/bin/date -v -30d "+%s")
if [[ $softwareupdate_date_epoch -lt $thirty_days_epoch ]]; then
  /bin/echo "0"
else
  /bin/echo "1"
fi
```

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

#### Remediation Description

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

```
/usr/sbin/softwareupdate -i -a
```

NOTE - This will apply to the whole system

ID	system_settings_softwareupdate_current	
References	800-53r5	• N/A
	CIS Benchmark	• 1.1 (level 1)
	CIS Controls V8	• 7.3
		• 7.4
	CCE	• CCE-92993-5

## 9.29. Disable SSH Server for Remote Access Sessions

SSH service *MUST* be disabled for remote access.

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

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

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

#### Remediation Description

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

```
/usr/sbin/systemsetup -f -setremotelogin off >/dev/null
```

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



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

ID	system_settings_ssh_disable	
References	800-53r5	<ul style="list-style-type: none"><li>• AC-17</li><li>• CM-7, CM-7(1)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.3.3.5 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.1</li><li>• 4.8</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-92994-3</li></ul>

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

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

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

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

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

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

## Remediation Description

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

```
authDBs=("system.preferences" "system.preferences.energysaver"  
"system.preferences.network" "system.preferences.printing"  
"system.preferences.sharing" "system.preferences.softwareupdate"  
"system.preferences.startupdisk" "system.preferences.timemachine")  
  
for section in ${authDBs[@]}; do  
/usr/bin/security -q authorizationdb read "$section" > "/tmp/$section.plist"  
key_value=$(/usr/libexec/PlistBuddy -c "Print :shared" "/tmp/$section.plist" 2>&1)  
  if [[ "$key_value" == *"Does Not Exist"* ]]; then  
    /usr/libexec/PlistBuddy -c "Add :shared bool false" "/tmp/$section.plist"  
  else  
    /usr/libexec/PlistBuddy -c "Set :shared false" "/tmp/$section.plist"  
  fi  
/usr/bin/security -q authorizationdb write "$section" < "/tmp/$section.plist"  
done
```

ID	system_settings_system_wide_preferences_configure	
References	800-53r5	<ul style="list-style-type: none"><li>AC-6, AC-6(1), AC-6(2)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>2.6.8 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>4.1</li></ul>
	CCE	<ul style="list-style-type: none"><li>CCE-92996-8</li></ul>

## 9.31. 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': ' '{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
```

```
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 Settings → Time Machine
2. Click **Select Disk**
3. Select existing Backup Disk under **Available Disks**
4. Click **Encrypt Backups**
5. Click **Use Disk**

ID	system_settings_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>• 2.3.4.2 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 3.6</li><li>• 3.11</li><li>• 11.3</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-92998-4</li></ul>

## 9.32. 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.apple.com**, this is a finding.

### Remediation 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.apple.com</string>
```

ID	system_settings_time_server_configure	
References	800-53r5	<ul style="list-style-type: none"><li>• AU-12(1)</li><li>• SC-45(1)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.3.2.1 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 8.4</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-92999-2</li></ul>

## 9.33. Enforce macOS Time Synchronization

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

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

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

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

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

### Remediation Description

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



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

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

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

ID	system_settings_time_server_enforce	
References	800-53r5	<ul style="list-style-type: none"><li>• AU-12(1)</li><li>• SC-45(1)</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.3.2.1 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 8.4</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-93000-8</li></ul>

## 9.34. Ensure Wake for Network Access Is Disabled

Wake for network access *MUST* be disabled.

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

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

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

### Remediation Description

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

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

ID	system_settings_wake_network_access_disable	
References	800-53r5	<ul style="list-style-type: none"><li>• N/A</li></ul>
	CIS Benchmark	<ul style="list-style-type: none"><li>• 2.9.3 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.8</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-93005-7</li></ul>

# 9.35. 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	system_settings_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>• 2.4.1 (level 1)</li></ul>
	CIS Controls V8	<ul style="list-style-type: none"><li>• 4.8</li><li>• 12.6</li></ul>
	CCE	<ul style="list-style-type: none"><li>• CCE-93010-7</li></ul>

# Chapter 10. Supplemental

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

## 10.1. CIS Manual Recommendations

List of CIS recommendations that are manual check in the CIS macOS Benchmark.

Section	System Settings
<b>Recommendations</b>	2.1.1.1 Audit iCloud Keychain 2.1.1.2 Audit iCloud Drive 2.1.1.4 Audit Security Keys Used With AppleIDs 2.1.2 Audit App Store Password Settings 2.3.3.12 Ensure Computer Name Does Not Contain PII or Protected Organizational Information 2.5.1 Audit Siri Settings 2.6.1.3 Audit Location Services Access 2.6.2.1 Audit Full Disk Access for Applications 2.6.7 Audit Lockdown Mode 2.8.1 Audit Universal Control Settings 2.11.2 Audit Touch ID 2.13.1 Audit Passwords System Preference Setting 2.14.1 Audit Game Center Settings 2.15.1 Audit Notification & Focus Settings 2.16.1 Audit Wallet & Apple Pay Settings 2.17.1 Audit Internet Accounts for Authorized Use

Section	Logging and Auditing
<b>Recommendations</b>	3.7 Audit Software Inventory

Section	System Access, Authentication and Authorization
<b>Recommendations</b>	5.2.3 Ensure Complex Password Must Contain Alphabetic Characters Is Configured 5.2.4 Ensure Complex Password Must Contain Numeric Character Is Configured 5.2.5 Ensure Complex Password Must Contain Special Character Is Configured 5.2.6 Ensure Complex Password Must Contain Uppercase and Lowercase Characters Is Configured 5.3.1 Ensure All User Storage APFS Volumes are Encrypted 5.3.2 Ensure All User Storage CoreStorage Volumes are Encrypted

Section	Applications
---------	--------------



## 10.2. Out of Scope Supplemental

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

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

<b>Family</b>	Access Control (AC)
<b>Controls</b>	<a href="#">AC-1</a> , <a href="#">AC-2</a> , <a href="#">AC-3(14)</a> , <a href="#">AC-14</a> , <a href="#">AC-17(4)</a> , <a href="#">AC-22</a>
<b>Family</b>	Awareness and Training (AT)
<b>Controls</b>	<a href="#">AT-1</a> , <a href="#">AT-2</a> , <a href="#">AT-3</a> , <a href="#">AT-4</a>
<b>Family</b>	Audit and Accountability (AU)
<b>Controls</b>	<a href="#">AU-1</a> , <a href="#">AU-6</a> , <a href="#">AU-9(2)</a>
<b>Family</b>	Security Assessment and Authorization (CA)
<b>Controls</b>	<a href="#">CA-1</a> , <a href="#">CA-2</a> , <a href="#">CA-3</a> , <a href="#">CA-3(6)</a> , <a href="#">CA-5</a> , <a href="#">CA-6</a> , <a href="#">CA-7</a> , <a href="#">CA-7(4)</a> , <a href="#">CA-9</a>
<b>Family</b>	Configuration Management (CM)
<b>Controls</b>	<a href="#">CM-1</a> , <a href="#">CM-4</a> , <a href="#">CM-8</a> , <a href="#">CM-10</a> , <a href="#">CM-11</a>
<b>Family</b>	Contingency Planning (CP)
<b>Controls</b>	<a href="#">CP-1</a> , <a href="#">CP-2</a> , <a href="#">CP-3</a> , <a href="#">CP-4</a> , <a href="#">CP-9</a> , <a href="#">CP-10</a>
<b>Family</b>	Identification and Authentication (IA)
<b>Controls</b>	<a href="#">IA-1</a> , <a href="#">IA-8(1)</a> , <a href="#">IA-8(2)</a> , <a href="#">IA-8(3)</a> , <a href="#">IA-8(4)</a>
<b>Family</b>	Incident Response (IR)
<b>Controls</b>	<a href="#">IR-1</a> , <a href="#">IR-2</a> , <a href="#">IR-4</a> , <a href="#">IR-5</a> , <a href="#">IR-6</a> , <a href="#">IR-7</a> , <a href="#">IR-8</a>
<b>Family</b>	Maintenance (MA)
<b>Controls</b>	<a href="#">MA-1</a> , <a href="#">MA-2</a> , <a href="#">MA-5</a>

<b>Family</b>	Media Protection (MP)
<b>Controls</b>	<a href="#">MP-1</a> , <a href="#">MP-2</a> , <a href="#">MP-6</a> , <a href="#">MP-7</a>

<b>Family</b>	Physical and Environmental Protection (PE)
<b>Controls</b>	<a href="#">PE-1</a> , <a href="#">PE-2</a> , <a href="#">PE-3</a> , <a href="#">PE-6</a> , <a href="#">PE-8</a> , <a href="#">PE-12</a> , <a href="#">PE-13</a> , <a href="#">PE-14</a> , <a href="#">PE-15</a> , <a href="#">PE-16</a>

<b>Family</b>	Planning (PL)
<b>Controls</b>	<a href="#">PL-1</a> , <a href="#">PL-2</a> , <a href="#">PL-4</a>

<b>Family</b>	Personnel Security (PS)
<b>Controls</b>	<a href="#">PS-1</a> , <a href="#">PS-2</a> , <a href="#">PS-3</a> , <a href="#">PS-4</a> , <a href="#">PS-5</a> , <a href="#">PS-6</a> , <a href="#">PS-7</a> , <a href="#">PS-8</a>

<b>Family</b>	Risk Assessment (RA)
<b>Controls</b>	<a href="#">RA-1</a> , <a href="#">RA-2</a> , <a href="#">RA-3</a> , <a href="#">RA-5</a>

<b>Family</b>	System and Services Acquisition (SA)
<b>Controls</b>	<a href="#">SA-1</a> , <a href="#">SA-2</a> , <a href="#">SA-3</a> , <a href="#">SA-4</a> , <a href="#">SA-4(10)</a> , <a href="#">SA-5</a> , <a href="#">SA-9</a>

<b>Family</b>	System and Communications Protection (SC)
<b>Controls</b>	<a href="#">SC-1</a> , <a href="#">SC-7(3)</a> , <a href="#">SC-7(7)</a> , <a href="#">SC-7(8)</a> , <a href="#">SC-7(18)</a> , <a href="#">SC-7(21)</a> , <a href="#">SC-12</a> , <a href="#">SC-12(1)</a> , <a href="#">SC-20</a> , <a href="#">SC-22</a> , <a href="#">SC-23</a>

<b>Family</b>	System and Information Integrity (SI)
<b>Controls</b>	<a href="#">SI-1</a> , <a href="#">SI-4</a> , <a href="#">SI-4(2)</a> , <a href="#">SI-4(4)</a> , <a href="#">SI-4(5)</a> , <a href="#">SI-4(12)</a> , <a href="#">SI-4(14)</a> , <a href="#">SI-4(20)</a> , <a href="#">SI-4(22)</a> , <a href="#">SI-5</a> , <a href="#">SI-7(2)</a> , <a href="#">SI-8(2)</a> , <a href="#">SI-12</a>

## 10.3. FileVault Supplemental

The supplemental guidance found in this section is applicable for the following rules: \*  
system\_settings\_filevault\_enforce

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



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

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

FileVault can be enabled in two ways within the macOS. It can be managed using the `fdsetup` command or by a Configuration Profile. When enabling FileVault via either of the aforementioned

methods, you will be required to enter a username and password, which must be a local Open Directory account with a valid SecureToken password.

## Using the `fdsetup` Command

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

```
/usr/bin/fdsetup enable
```

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



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

## Using a Configuration Profile

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

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

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

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

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

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

It's recommended that you use a Personal Recovery key instead of an Institutional key as it will generate a specific key for each device. You can find more guidance on choosing a recover key here:



FileVault currently only uses password-based authentication and cannot be done using a smartcard or any other type of multi-factor authentication.

## 10.4. Packet Filter (pf) Supplemental

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

- `os_firewall_default_deny_require`

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

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

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

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

The ruleset will block connections on the following ports:

Port	Service
548	Apple File Protocol (AFP)
1900	Bonjour
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

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

For more on configuring the PF firewall check out the man pages on [pf.conf](#) and [pfctl](#).

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

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

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

mdm_managed=$(/usr/bin/osascript -l JavaScript -e "
```

```

$.NSUserDefaults.alloc.initWithSuiteName('com.apple.security.firewall').objectIsForced
ForKey('EnableFirewall'))

#### Functions ####

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

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

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

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

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

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

# append the mscp anchors to pf.conf
configure_pf_config_add_mscp_anchors () {
    echo "Adding the MSCP anchors to /etc/pf.conf"
    # check to see if mscp anchors exists
    anchors_exist=$(grep -c '^anchor "mscp_pf_anchors"' /etc/pf.conf)

```

```

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

# Create /etc/pf.anchors/msep_pf_anchors
create_msep_pf_anchors () {
    echo "Creating the MSCP anchor configuration file"
    if [[ -e /etc/pf.anchors/msep_pf_anchors ]]; then
        echo "msep Anchor file exists, deleting and recreating..."
        rm -f /etc/pf.anchors/msep_pf_anchors
    fi

cat > /etc/pf.anchors/msep_pf_anchors <<'ENDCONFIG'

anchor msep_pf_anchors

#default deny all in, allow all out and keep state
block in all
pass out all keep state

#pass in all packets from localhost
pass in from 127.0.0.1

## Allow DHCP
pass in inet proto udp from port 67 to port 68
pass in inet6 proto udp from port 547 to port 546

## Allow incoming SSH
pass in proto tcp to any port 22

#apple file service --port 548-- pf firewall rule
block in log proto tcp to any port { 548 }

#bonjour component SSDP --port 1900-- pf firewall rule
block log proto udp to any port 1900

#finger --port 79-- pf firewall rule
block log proto tcp to any port 79

#ftp --ports 20 21-- pf firewall rule
block in log proto { tcp udp } to any port { 20 21 }

#http --port 80-- pf firewall rule

```

```

block in log proto { tcp udp } to any port 80

#icmp pf firewall rule
block in log proto icmp

#imap --port 143-- pf firewall rule
block in log proto tcp to any port 143

#imaps --port 993-- pf firewall rule
block in log proto tcp to any port 993

#iTunes sharing --port 3689-- pf firewall rule
block log proto tcp to any port 3689

#mDNSResponder --port 5353-- pf firewall rule
block log proto udp to any port 5353

#nfs --port 2049-- pf firewall rule
block log proto tcp to any port 2049

#optical drive sharing --port 49152-- pf firewall rule
block log proto tcp to any port 49152

#pop3 --port 110-- pf firewall rule
block in log proto tcp to any port 110

#pop3s --port 995-- pf firewall rule
block in log proto tcp to any port 995

#remote apple events --port 3031-- pf firewall rule
block in log proto tcp to any port 3031

#screen_sharing --port 5900-- pf firewall rule
block in log proto tcp to any port 5900
#allow screen sharing from localhost while tunneled via SSH
pass in quick on lo0 proto tcp from any to any port 5900

#smb --ports 139 445 137 138-- pf firewall rule
block in log proto tcp to any port { 139 445 }
block in log proto udp to any port { 137 138 }

#smtp --port 25-- pf firewall rule
block in log proto tcp to any port 25

#telnet --port 23-- pf firewall rule
block in log proto { tcp udp } to any port 23

#tftp --port 69-- pf firewall rule
block log proto { tcp udp } to any port 69

#uucp --port 540-- pf firewall rule

```



```

block log proto tcp to any port 540

ENDCONFIG
}

# function to remove legacy setup if exists
remove_macsec_setup() {
    echo "References to macsec appear to exist, removing..."

    launchctl disable system/macsec.pfctl
    launchctl bootout system $legacy_launchd_plist
    rm -rf $legacy_launchd_plist

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

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

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

uninstall_mscp_pf(){
    echo "Removing MSCP configuration files from pf"
    if [[ -e "$launchd_pfctl_plist" ]]; then
        echo "LaunchDaemon exists, unloading and removing"
        #remove mscp pf components from launchd
        launchctl disable system/mscp.pfctl
        launchctl bootout system $launchd_pfctl_plist
        rm -rf $launchd_pfctl_plist
    fi

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

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

    rm -f /etc/pf.anchors/mscp_pf_anchors

    # flush rules and reload pf
    echo "Flushing rules and reloading pf"
    pfctl -f /etc/pf.conf 2> /dev/null #flush the pf ruleset (reload the rules)
}

```

```

#### Main Script ####

POSITIONAL_ARGS=()

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

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

if [[ $UNINSTALL == "true" ]]; then
    if [[ -e "$legacy_launchd_plist" ]]; then
        remove_macsec_setup
    fi
    uninstall_mscp_pf
    exit 0
fi

# check to see if a profile has enabled the firewall. If it hasn't, then CLI can be
used to enable
if [[ "$mdm_managed" == "false" ]]; then
    enable_macos_application_firewall
fi

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

# create mscp anchors file
create_mscp_pf_anchors

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

```

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

## 10.5. Password Policy Supplemental

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

- pwpolicy\_lower\_case\_character\_enforce
- pwpolicy\_upper\_case\_character\_enforce
- pwpolicy\_account\_inactivity\_enforce
- pwpolicy\_minimum\_lifetime\_enforce

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

- Enforcing at least 1 lowercase character
- Enforcing at least 1 uppercase character
- Disabling an account after 35 days of inactivity
- Password minimum lifetime

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

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>policyCategoryAuthentication</key>
  <array>
    <dict>
      <key>policyContent</key>
      <string>(policyAttributeFailedAuthentications &lt;
policyAttributeMaximumFailedAuthentications) OR (policyAttributeCurrentTime &gt;
(policyAttributeLastFailedAuthenticationTime + autoEnableInSeconds))</string>
      <key>policyIdentifier</key>
      <string>Authentication Lockout</string>
      <key>policyParameters</key>
      <dict>
        <key>autoEnableInSeconds</key>
        <integer>300</integer>
        <key>policyAttributeMaximumFailedAuthentications</key>
        <integer>3</integer>
      </dict>
    </dict>
  </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>
</dict>

```

```

<dict>
  <key>policyContent</key>
  <string>policyAttributePassword matches '.{15,}+'</string>
  <key>policyIdentifier</key>
  <string>Must be at least 15 characters</string>
  <key>policyParameters</key>
  <dict>
    <key>minimumLength</key>
    <integer>15</integer>
  </dict>
</dict>
<dict>
  <key>policyContent</key>
  <string>policyAttributePassword matches '(.*[0-9].*){1,}+'</string>
  <key>policyIdentifier</key>
  <string>Must have at least 1 numeric value</string>
  <key>policyParameters</key>
  <dict>
    <key>minimumNumericCharacters</key>
    <integer>2</integer>
  </dict>
</dict>
<dict>
  <key>policyContent</key>
  <string>policyAttributePassword matches '(.*[a-z].*){1,}+'</string>
  <key>policyIdentifier</key>
  <string>Must have at least 1 lowercase letter</string>
  <key>policyParameters</key>
  <dict>
    <key>minimumAlphaCharactersLowerCase</key>
    <integer>1</integer>
  </dict>
</dict>
<dict>
  <key>policyContent</key>
  <string>policyAttributePassword matches '(.*[A-Za-z].*){1,}+'</string>
  <key>policyIdentifier</key>
  <string>Must have at least 1 Letter</string>
  <key>policyParameters</key>
  <dict>
    <key>minimumAlphaCharacters</key>
    <integer>1</integer>
  </dict>
</dict>
<dict>
  <key>policyContent</key>
  <string>policyAttributePassword matches '(.^[a-zA-Z0-9].*){1,}+'</string>
  <key>policyIdentifier</key>
  <string>Must have at least 1 special characters</string>
  <key>policyParameters</key>
  <dict>

```

```

        <key>minimumSymbols</key>
        <integer>1</integer>
    </dict>
</dict>
<dict>
    <key>policyContent</key>
    <string>none policyAttributePasswordHashes in
policyAttributePasswordHistory</string>
    <key>policyIdentifier</key>
    <string>Cannot match the last 5 passwords</string>
    <key>policyParameters</key>
    <dict>
        <key>policyAttributePasswordHistoryDepth</key>
        <integer>5</integer>
    </dict>
</dict>
</array>
</dict>
</plist>

```

Run the following command to load the new policy file, substituting the path to the file in place of "\$pwpolicy\_file".

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



If directory services is being utilized, password policies should come from the domain.

## 10.6. Smartcard Supplemental

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

- auth\_ssh\_password\_authentication\_disable
- auth\_smartcard\_enforce
- auth\_smartcard\_certificate\_trust\_enforce\_moderate
- auth\_smartcard\_certificate\_trust\_enforce\_high
- auth\_smartcard\_allow
- auth\_pam\_sudo\_smartcard\_enforce
- auth\_pam\_su\_smartcard\_enforce
- auth\_pam\_login\_smartcard\_enforce

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

- Authentication (Loginwindow, Screensaver, SSH, PKINIT, Safari, Finder, and PAM Authorization (`sudo`, `login`, and `su`) )
- Digital Encryption
- Digital Signing
- Remote Access (VPN:L2TP)
- Port-based Network Access Control (802.1X)
- Keychain Unlock

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

## Smartcard Pairing

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

## Smartcard Attribute Mapping

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

## Smartcard Management in macOS

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

Key	Type	Value
<code>userPairing</code>	bool	If false, users will not get the pairing dialog, although existing pairings will still work.
<code>allowSmartCard</code>	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</string>
</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\
```



When booted to recovery on an Apple Silicon Mac, run the following after setting the exemption. `/usr/sbin/diskutil apfs updatePreboot /Volumes/Macintosh\ HD`

## Temporary Exemption

On an Apple Silicon Mac, if a temporary exemption is needed, `security filevault skip-sc-enforcement` will disable smartcard enforcement on next boot only.

Run the following command to set the temporary exemption when booted from Recovery:

```
/usr/bin/security filevault skip-sc-enforcement <data volume UUID> set
```

To obtain the `data volume UUID` run the following:

```
/usr/sbin/diskutil apfs listGroups | /usr/bin/awk -F: '/ Data/ { getline; gsub(/,,""); print $2}'
```

## Pluggable Authentication Module (PAM)

Terminal sessions in macOS can be configured for smartcard enforcement by modifying the PAM modules for `sudo`, `su`, and `login`.

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

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

```

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

```

## Screen Sharing and Screen Recording

macOS will disable support for TouchID, Watch, or Smartcard authentication when being watched or recorded. This can cause certain portions of the system to not recognize your smartcard.

In Unified Logging you'll notice an entry such as

```

2022-07-14 16:45:46.880038-0400 0x2F97 Info 0xC8D2 1600 SecurityAgent: (SecurityAgent)
[com.apple.Authorization:SecurityAgent] Screen is being watched, no Touch ID, Watch or
SmartCard support is allowed

```

This can be remediated by writing the preference domain `com.apple.authorization` with the key `ignoreARD`.

```
defaults write com.apple.Authorization ignoreARD -bool true
```

Or applied system wide with a configuration profile named `com.apple.security.authorization.mobileconfig` in the project's `includes` folder.

```

<key>PayloadType</key>
<string>com.apple.security.authorization</string>
<key>ignoreArd</key>
<true/>

```