Jamf Compliance Editor

User's Guide

Jamf Education Services
Jamf Consulting Engineering
Jamf Professional Services

Version 1.4.0
September 2024
© 2022-2024 Jamf

Warnings

1. This utility includes content from non-Jamf third parties. Always carefully inspect, understand, and test all content before installing it on production devices.
2. Some of the security baselines supported by this application include scripts and configuration profiles to require the use of smartcards. If you deploy these controls to a macOS device that has not been configured for smartcards, you will no longer be able to login with a password.

Table of Contents

Introduction

Purpose

The Jamf Compliance Editor application is based on the US National Institute of Standards and Technology's macOS Security Compliance Project (mSCP). The NIST project allows an admin to select a version of macOS, iOS/iPadOS, or visionOS and one of a number of supported compliance standards and edit the items that they want to include in their device management configuration. Once the selection process is complete, the utility can generate the configuration profiles and scripts needed to configure the required settings, remediate violations when they occur, and generate compliance reports. Additionally, the application can create documentation the admin can share with internal teams and auditors, configuration profiles to enforce settings.

The Jamf Compliance Editor app simplifies the implementation of the mSCP project in a Jamf Pro environment. Configuration choices can be made in a GUI rather than by editing configuration files and using mSCP's command line operations. Jamf Compliance Editor makes all of the same functionality available at the click of a button. The app makes it easier to browse through the different standards and select the rules that best apply to your organization and then upload the resulting profiles, scripts, and extension attributes to Jamf Pro.

Supported Standards

The NIST project currently supports the following published guides:

- NIST 800-53 - https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final
    - FISMA High/Moderate/Low
- NIST 800-171 - https://csrc.nist.gov/publications/detail/sp/800-171/rev-2/final
- DISA STIG - https://public.cyber.mil/stigs/downloads/?_dl_facet_stigs=operating-systems
- CMMC 2.0 - https://dodcio.defense.gov/CMMC/
- CNSSI-1253 - https://www.dcsa.mil/portals/91/documents/ctp/nao/CNSSI_No1253.pdf
- indigo (Base/High) - https://www.bsi.bund.de/EN/Themen/Oeffentliche-Verwaltung/Zulassung/mobile_Kommunikation/mobileKommunikation_node.html#doc919528bodyText2

Non-governmental organizations may use other standards. For example, the Center for Internet Security ("CIS") is a nonprofit that publishes its benchmarks. The following CIS projects are supported by the NIST Security Project:

- CIS Benchmarks
  - macOS - https://www.cisecurity.org/benchmark/apple_os
  - iOS/iPadOS - https://www.cisecurity.org/benchmark/apple_ios
- CIS Critical Security Controls Version 8 - https://www.cisecurity.org/controls/v8/

Security Baselines and Benchmarks

A *security baseline* is a group of controls that an organization has agreed to configure on their computing devices. Once these are in place, a process is needed to verify that the controls remain in place and that the devices remain compliant. This compliance scoring system is the "*security benchmark*". Jamf Pro provides tools for implementing baselines and running ongoing benchmark compliance reporting.

For example, an organization might wish to establish the following control set:

1. All Macs must be encrypted with FileVault
2. All Macs must turn the screen saver on no more than 5 minutes after the last user interaction and authentication will be required to wake from screen saver.
3. All Macs must disable Bluetooth Sharing

To implement the benchmark, the organization's Jamf Pro administrator can configure FileVault setup as part of their automated enrollment workflow and add a configuration profile to enforce the screen saver requirement. They can use the provided compliance script to ensure that Bluetooth Sharing is indeed disabled. Jamf Pro Extension Attributes can prove that their devices remain compliant.

The Jamf Compliance Editor app was built to make this process easier.

Why Do We Implement Baselines and Benchmarks?

Some government agencies require all computers that interact with their systems and data to be subject to a specific benchmark with few allowances for exceptions. Many regulated industries will also be required to implement a security benchmark. For example, a healthcare organization may be subject to requirements under HIPAA and a retail or e-commerce company may have systems subject to PCI because they process credit card transactions. Schools and colleges need to apply protections for ensuring

privacy of the student education records under the Family Educational Rights and Privacy Act (FERPA).

Even if an organization is not required to adhere to the entirety of a specific standard, they may find that some of the standard's settings will help keep the organization's data safe, and the application of continuous benchmark reporting ensures that configuration mistakes don't fall through the cracks unnoticed.

But one-size does not fit all, and implementing every part of a published standard is often not required or appropriate. Instead, IT and Information Security departments need to collaboratively consider the risks of greatest concern to their organization, determine the best controls to mitigate those risks, and implement them with careful thought to striking a balance between information security and user productivity.

Different devices within an organization may also be placed into different risk categories and therefore implement different control sets. For example, a device in the HR department may have a five-minute screen saver timeout while a Mac that plays a continuous slide show in the Sales Department's waiting room would have no screen saver at all.

Terms of use

Support

The Jamf Compliance Editor application is free-to-use for Jamf customers but is not a part of the Jamf Product line. It is not supported via typical Jamf Support channels and has not been tested or validated by Jamf's internal compliance, quality, or product security processes.

Warning: This utility includes content from non-Jamf third parties. Always carefully inspect, understand, and test all content before installing it on production devices.

We welcome your feedback. Comments may be sent to **compliance.editor.feedback -at- jamf.com.**

Agreement

THIS SOFTWARE IS PROVIDED "AS-IS," WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL JAMF SOFTWARE, LLC OR ANY OF ITS AFFILIATES BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT, OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OF OR OTHER DEALINGS IN THE SOFTWARE, INCLUDING BUT NOT LIMITED TO DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR PUNITIVE DAMAGES AND OTHER DAMAGES SUCH AS LOSS OF USE, PROFITS, SAVINGS, TIME OR DATA, BUSINESS INTERRUPTION, OR PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES.

Copyright notice

This document and Jamf-developed portions of the Compliance Editor app are © 2022-2024, Jamf

Included components

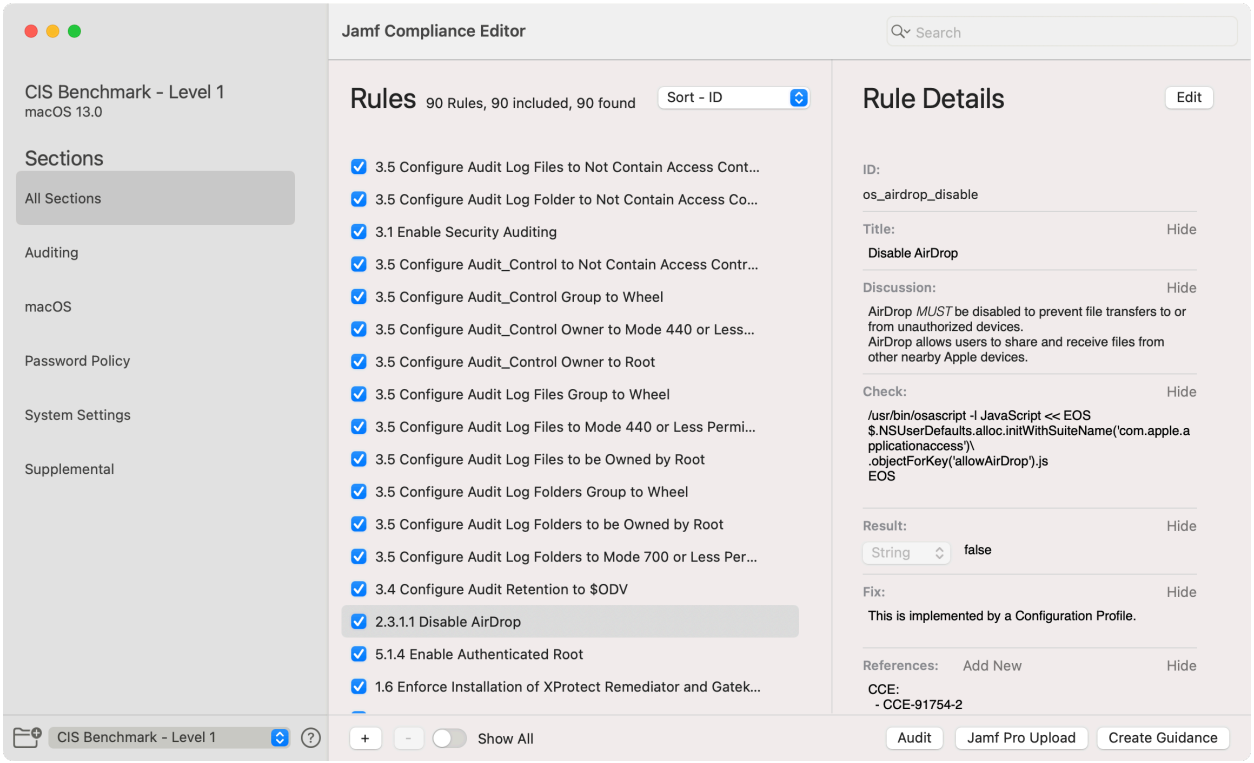
The following third-party acknowledgements and licenses are incorporated by reference:		
NIST macOS Security	Project Page	Creative Commons Attribution 4.0
Python	Project Page	Python Software Foundation License
Relocatable Python	Project Page	Apache License
AsciiDoctor	Project Page	MIT

Quick Start

Start the application and create a new project. Follow the prompts to select macOS/iOS/iPadOS/visionOS and baseline/benchmark. The application's main window will appear.

1. The selected OS and benchmark appear on the left. The different benchmark components are organized by sections.
2. The rules within the selected section are listed in the middle column. These can be turned on or off depending on the rules an organization wants to enforce and measure. The search box in the upper-right corner can be used to quickly find controls.
3. Clicking on a rule will show its details. Click **Show** to see the details about how a control will be implemented and/or enforced on the managed devices, how it will be measured by benchmark reporting, how it will be remediated if the setting falls out of compliance, and a list of the benchmark standards that include the rule.

   Some rules have Organization-defined values ("ODVs") which can be edited. For example, the screensaver timeout rule provides a recommended number of seconds after the last user interaction that macOS should wait before turning on the screen saver. This value can be modified to meet an organizations desired state.
4. Once all rules have been selected, click **Create Guidance** button in the lower right.
5. Click the **Jamf Pro Upload** button to copy the configuration profiles and scripts to Jamf Pro. See the section "Uploading to Jamf Pro" later in this document for additional required steps.



Installation

Jamf Compliance Editor can be installed using the following steps on an macOS device.

Requirements

The Jamf Compliance Editor has the following requirements:

- A Mac computer running macOS 13 or higher
- An internet connection is needed to download the mSCP project from NIST's GitHub

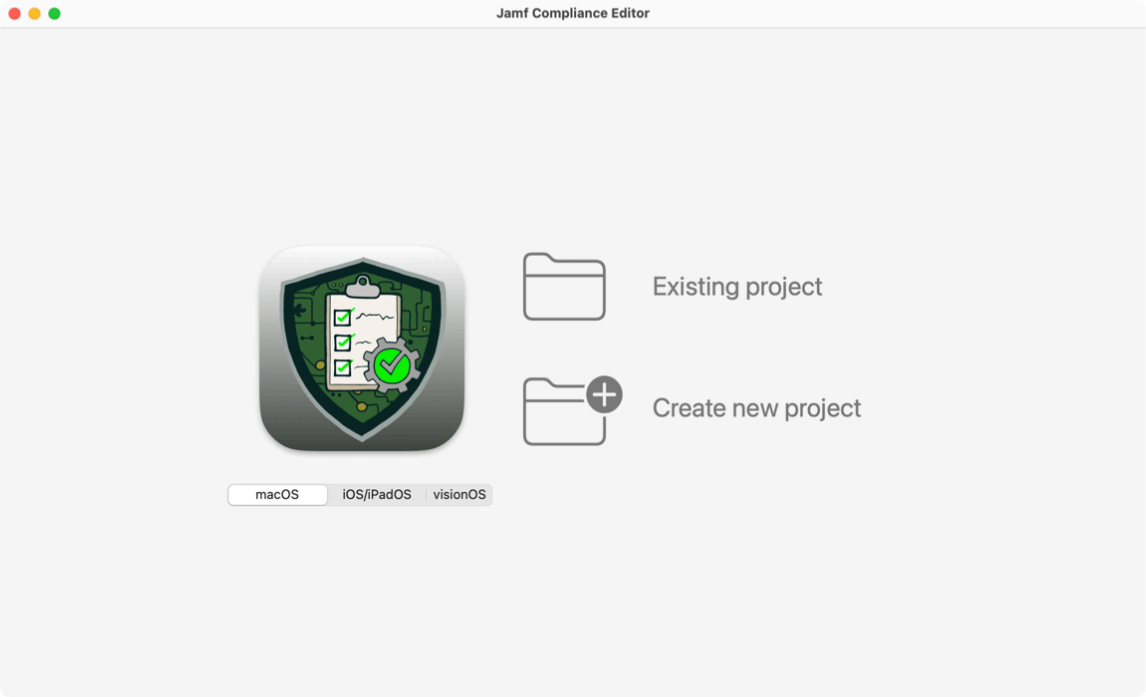
Download

The application can be downloaded from Jamf Trusted Access website:

https://trusted.jamf.com/docs/establishing-compliance-baselines

Beginning Steps

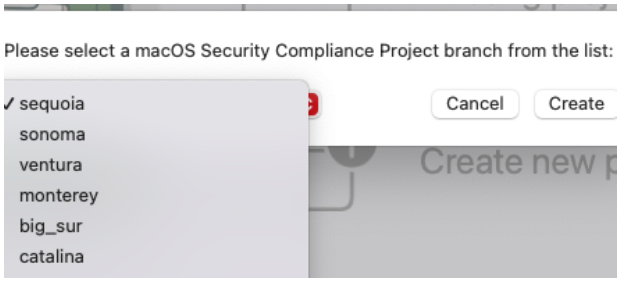
When launching the Jamf Compliance Editor, a wizard will prompt to select which type of operating system, then an “Existing project” created previously or “Create new project” from mSCP.



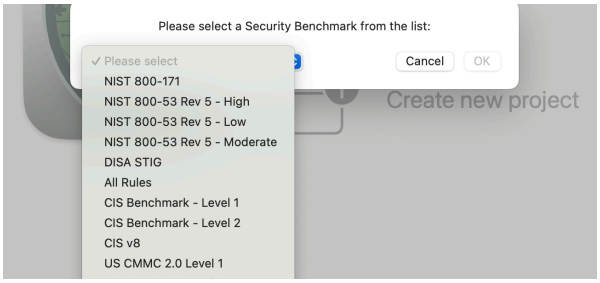
New Project

When creating a new project select a version of operating system this is for and then the security baseline/benchmark to use. Different operating system versions require different projects because each OS version has variations in how a rule will be implemented and measured.

1st: Select operating system



2nd: Select baseline/benchmark



Existing Project

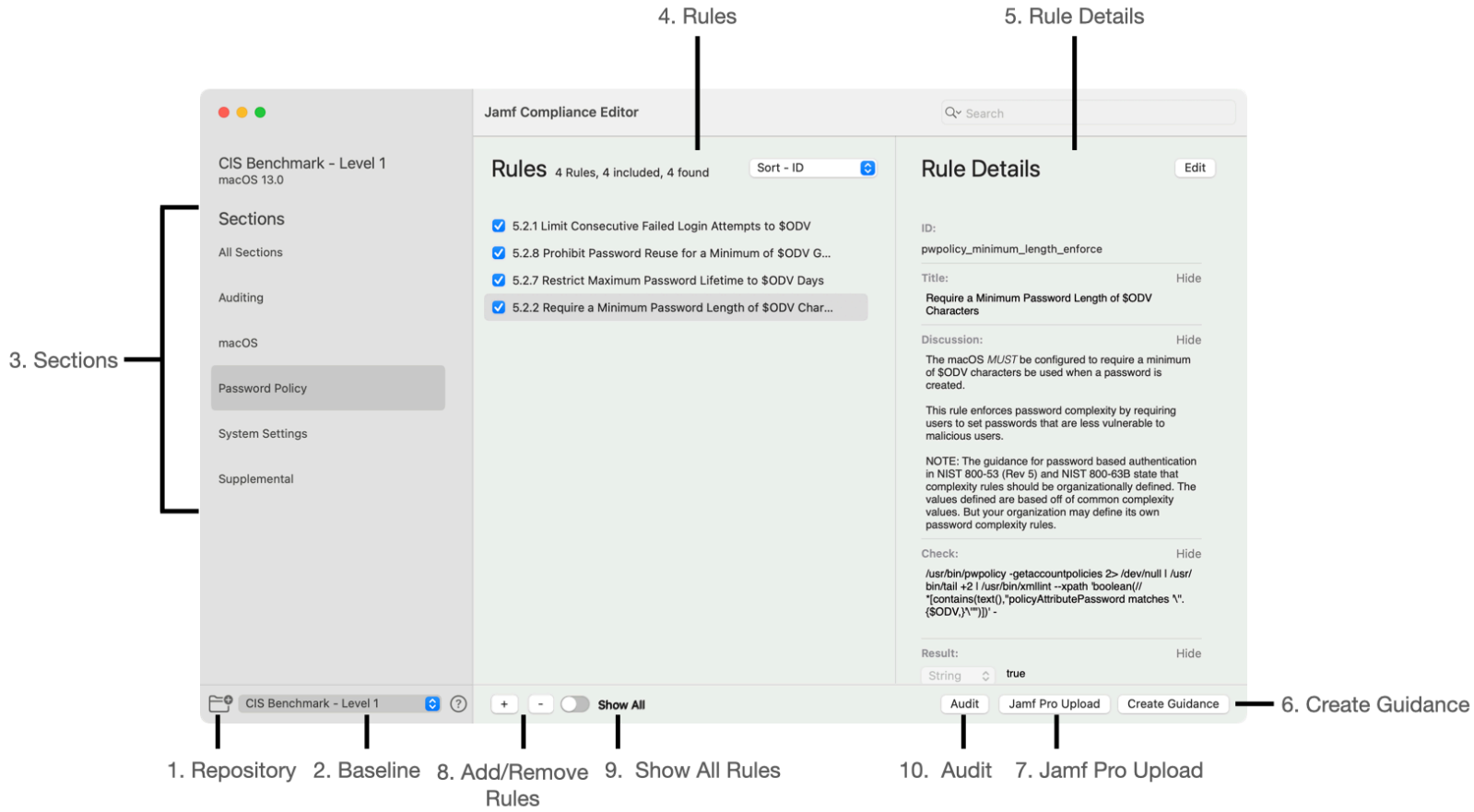
When selecting an existing project, select the directory where a previously saved project is located and then choose the security baseline/benchmark to use.

NOTE: We suggest only using folders created by this app. Any customizations that were built using the mSCP project directly will be overwritten if the folder is later used with Compliance Editor.

Application Overview

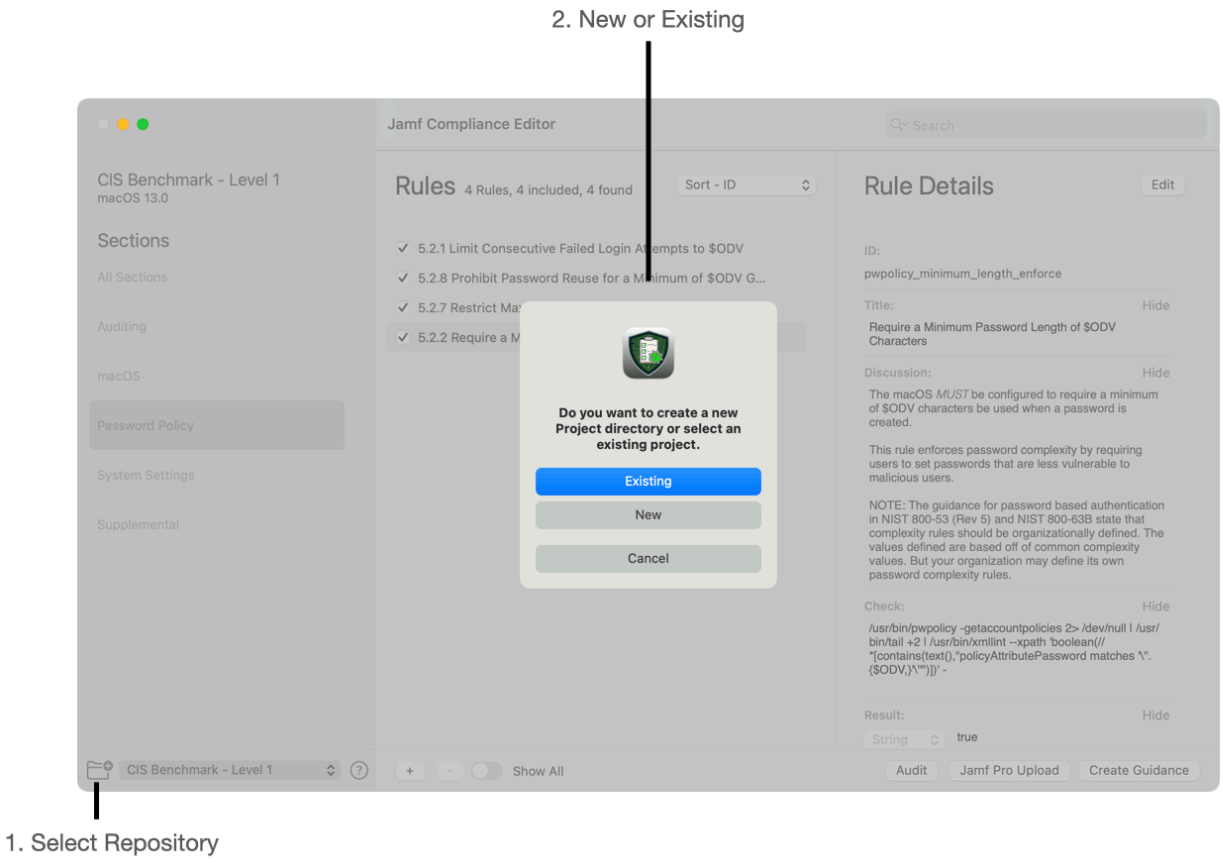
The Jamf Compliance Editor window is divided into the following areas.

1. Repository button - Used to select an existing repository or download a new one
2. Baseline popup menu - Switch between the baselines/benchmarks available
3. Sections - Displays all sections available from the selected baseline/benchmark
4. Rules - Displays rules from the selected Section
5. Rule Details - Allows editing of the various rule details including ODV values
6. Create Guidance - Generates output from mSCP plus files for Jamf Pro
7. Jamf Pro Upload - Uploads configuration profiles, compliance script, and Extension Attributes to a Jamf Pro server (Greyed out until Create Guidance completes)
8. Add/Remove Rules - Add/Remove custom rules
9. Show All Rules - Shows rules not in current baseline
10. Audit - Run audit against generated baseline (Greyed out until Create Guidance completes)



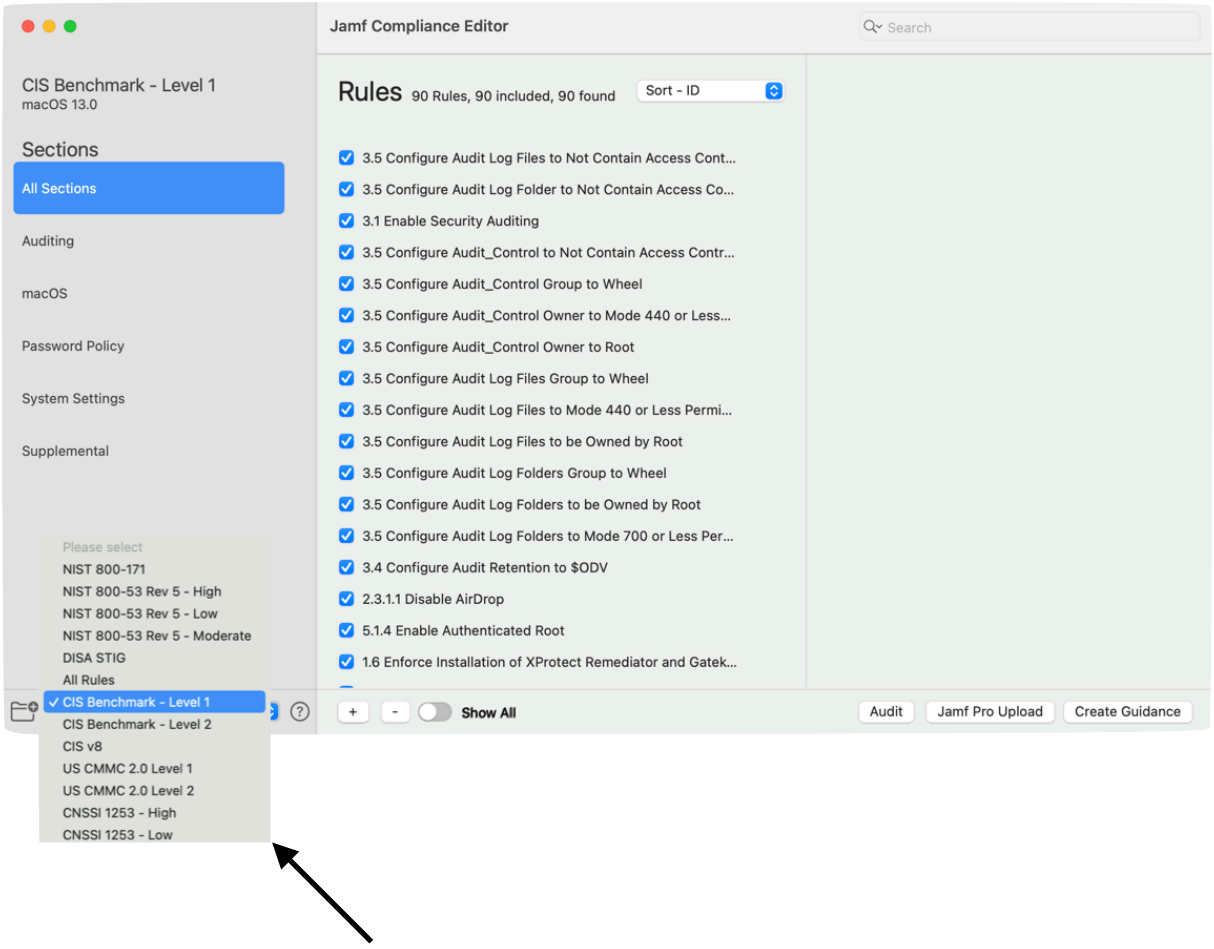
Repository Selection

When in the main app window, clicking offers the option to either select another existing local repository or download a new macOS repository from the mSCP.



Baseline Selection

Once an Operating System version has been selected, there is the option to switch between the other baselines and benchmarks from mSCP.

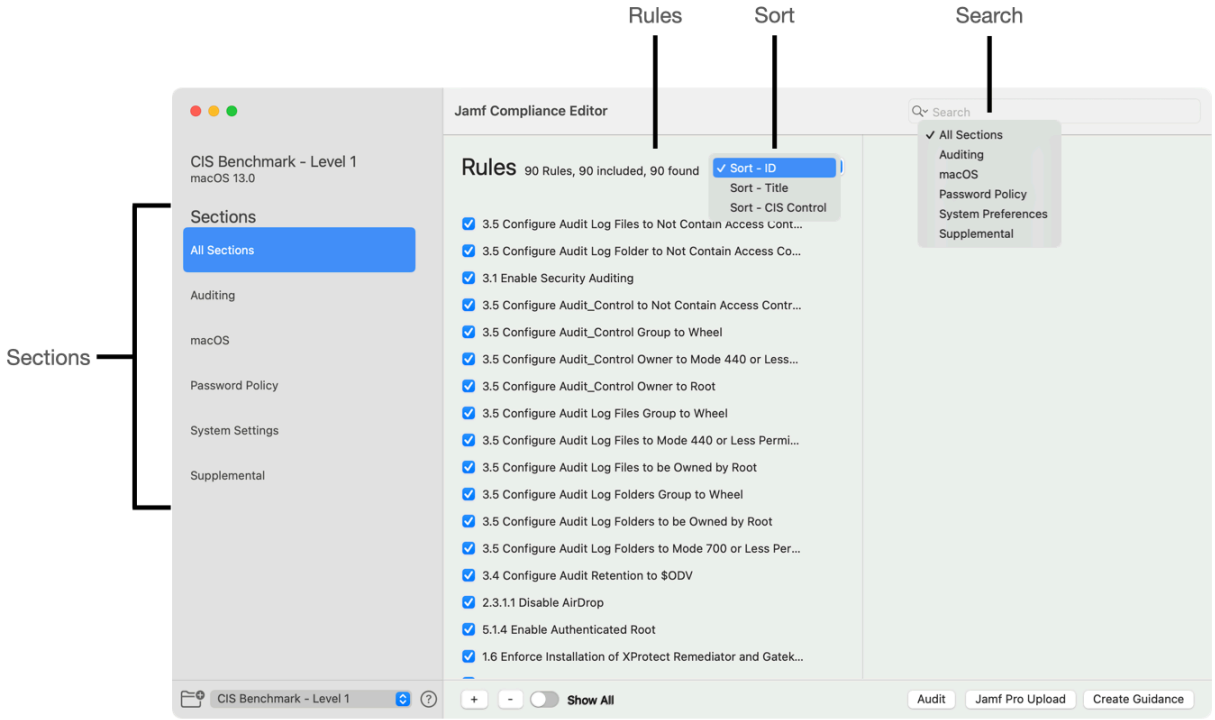


Sections and Rules

Once a baseline has been selected, the sections included in the selected baseline will appear in the sidebar. Selecting a specific section will then present a list of rules that relate to that section in the middle column.

Rules can be sorted by ID, Title or CIS control, if applicable.

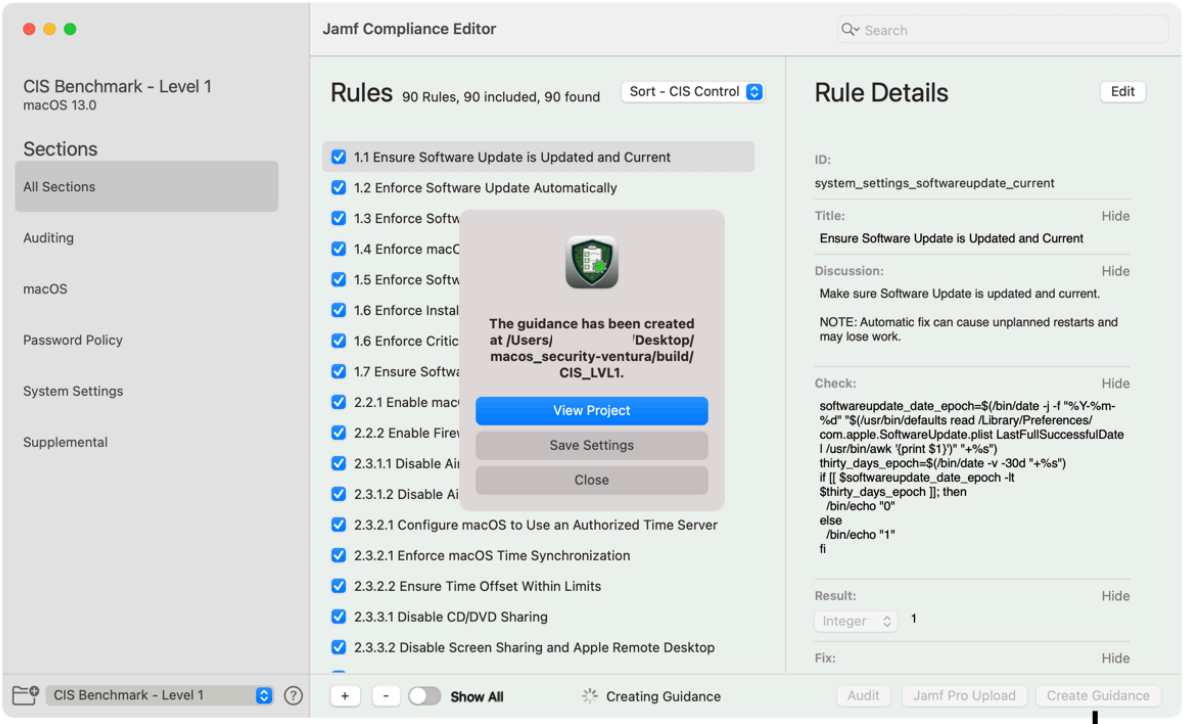
The search field is available to search for rules either via a specific section or across all sections. Enter *“odv”* to see all rules which have organization defined values (like how many minutes to wait before turning on the screen saver) or *“mobileconfig”* to show any rule that enforceable via a configuration profile.



Creating Guidance

To create the content associated with the selected baseline/benchmark, click **Create Guidance** button. When complete, a pop-up will be presented to either:

- Click "**View Project**" to open the folder displaying all the outputs
- Click "**Save Settings**" to save your configuration to a .jce file
- Click "**Close**" to close and continue using the app

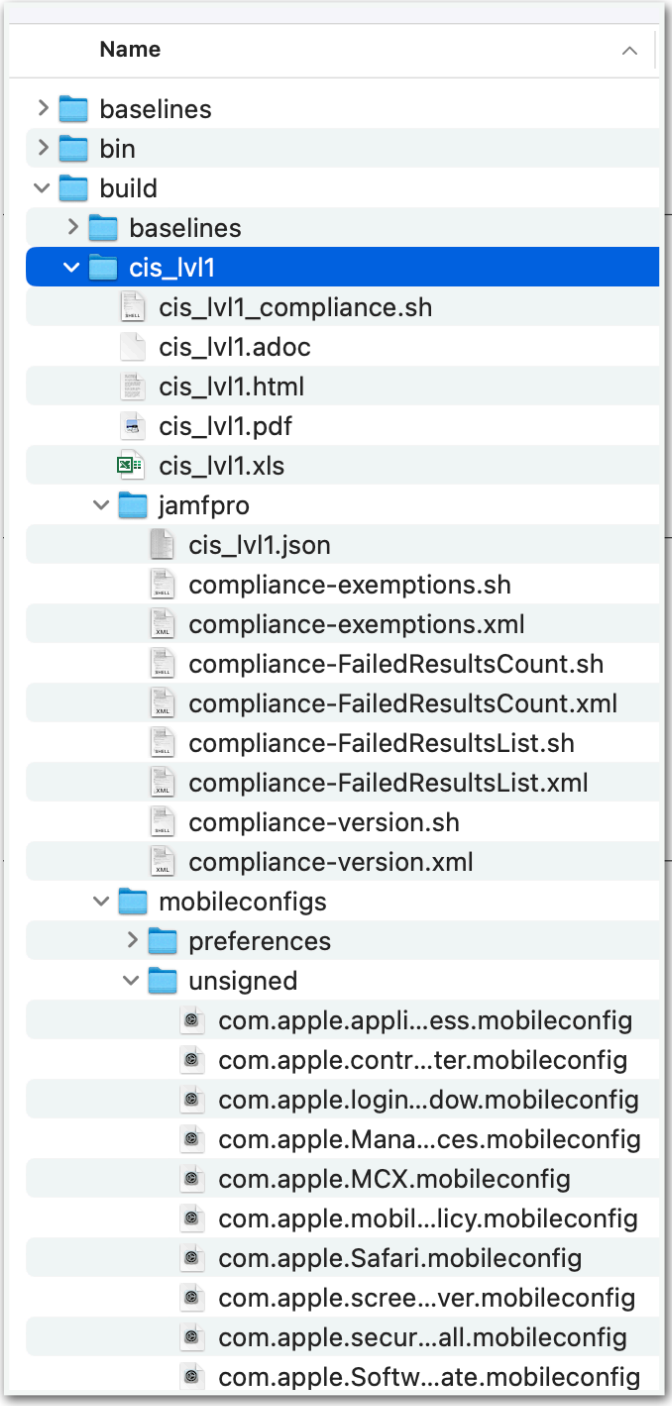


Note: The .jce file is a json file which contains any work that has been applied to a selected baseline, including any customizations. When launching the application by double-clicking the .jce file, all previous edits will appear.

Create Guidance Output

A folder containing the different scripts and other components of the mSCP project was saved to the location you specified when creating a project. A **build** folder is added to that directory when you click the **Create Guidance** button.

Generated Content



In the example shown here, the admin has generated the CIS Level 1 baseline.

Documentation & Script

Audit & Remediation Script. Output of selected rules in PDF, HTML, and other formats.

Jamf Pro Folder

The JSON schema for setting exemptions and the EAs to gather results.

Configuration Profiles

MDM profiles that configure and enforce the compliance baseline. These are provided as plists for use as custom profiles (the “preferences” folder) and in .mobileconfig format (the “unsigned” folder).

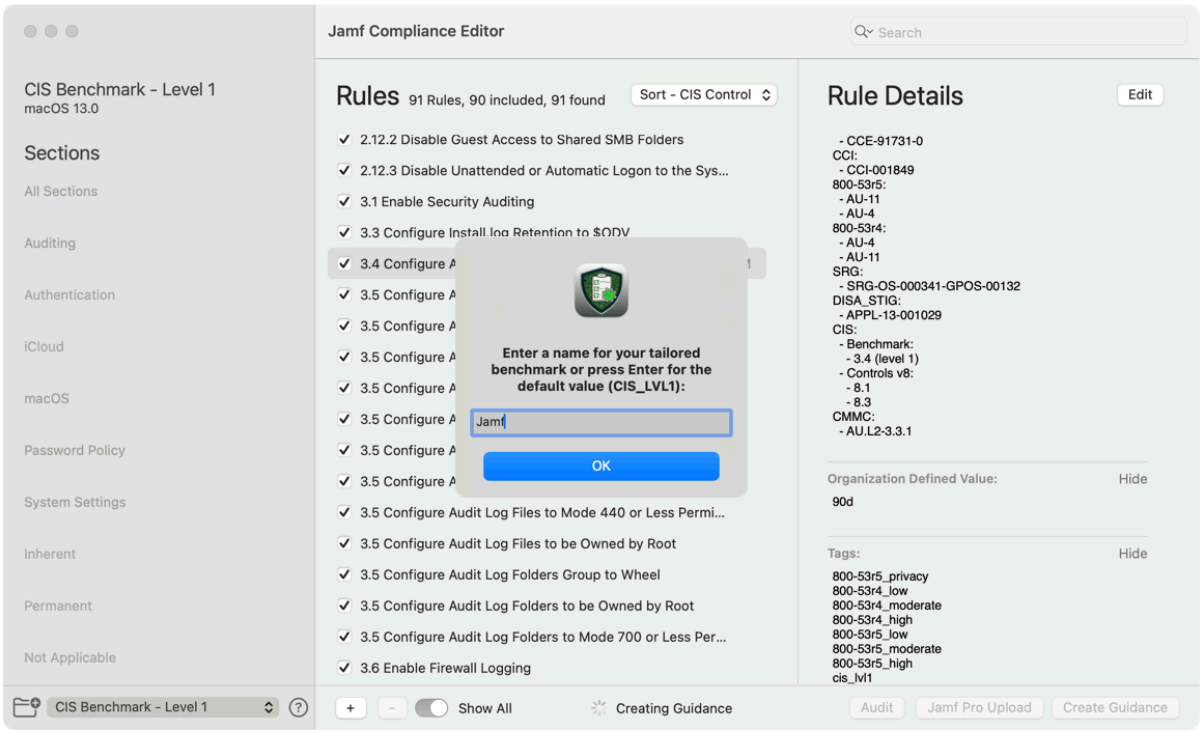
These are uploaded to Jamf Pro when using the Compliance Editor’s Upload button

Create Custom Guidance

Many organizations will want to customize a selected baseline/benchmark to conform to their organization's requirements. Jamf Compliance Editor allows you to customize baselines/benchmarks in the following ways:

- Enable/Disable rules
- Add rules not included in a selected baseline/benchmark
- Add rules not included in mSCP
- Edit rules contents
- Add authors
- Add custom banners

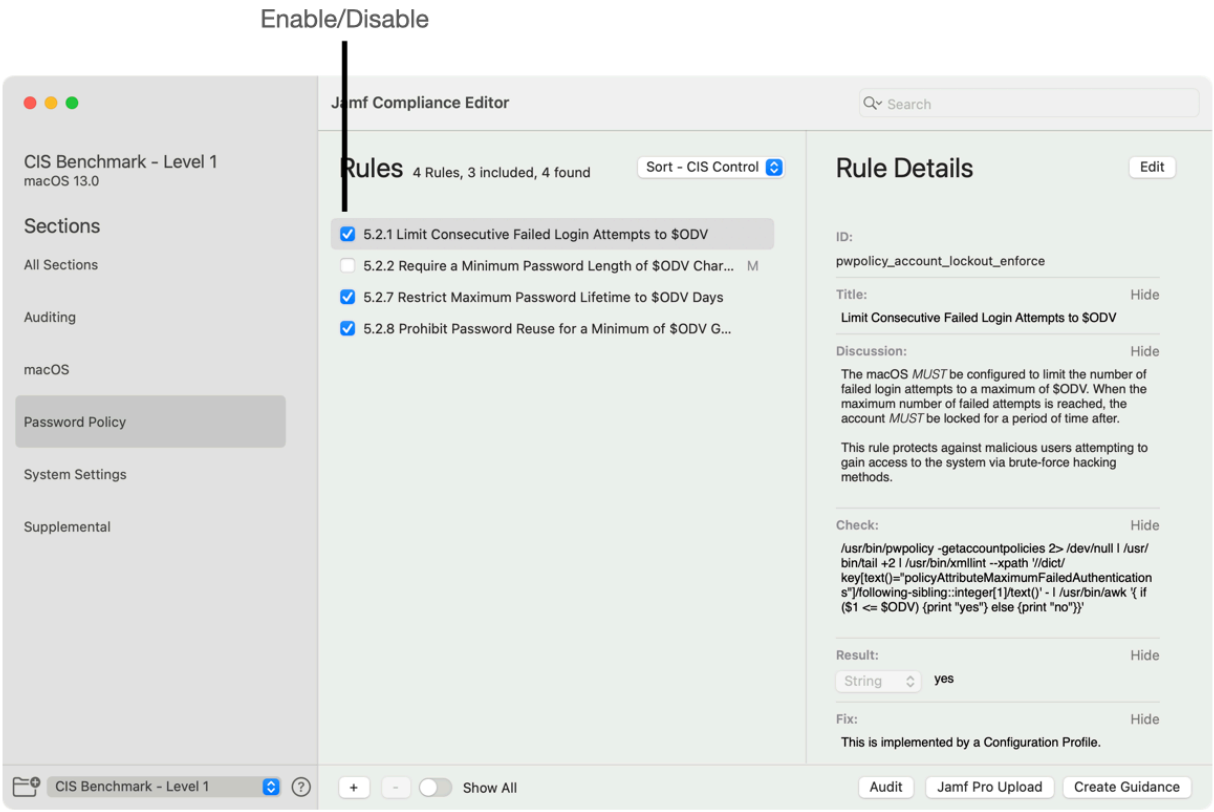
When modifications are made to the content and you click the **Generate Guidance** button, a pop-up will appear asking to enter new name for tailored benchmark. This will create a new build folder with the name you entered and a new baseline file which can be reopened at a later time.



Enable/Disable Rules

Any rule can be included or excluded from a baseline by selecting or deselecting the checkbox beside each rule.

*Shortcut: **Option** + **Click** one checkbox to Select All (⌘ **R**) or Deselect All (⇧ ⌘ **R**)*

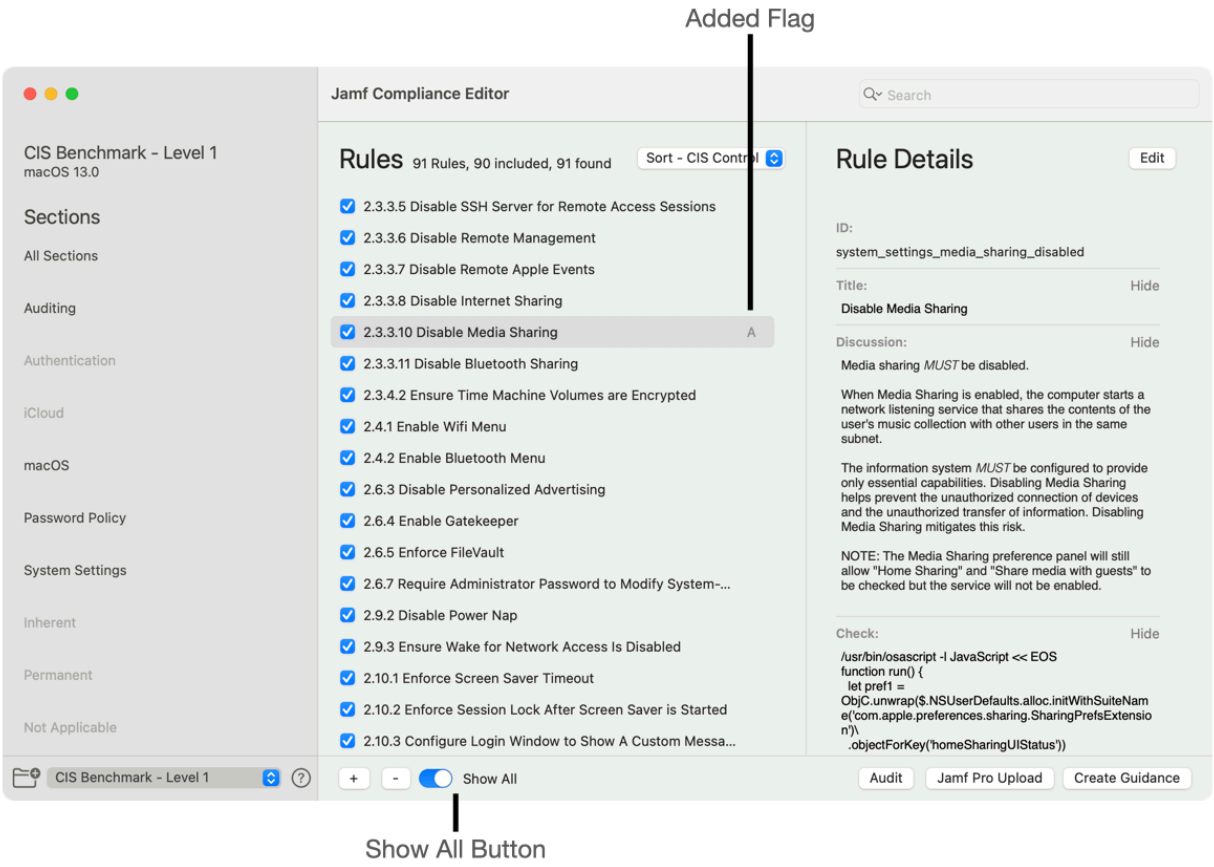


Show All Rules

There might be cases where you're implementing one level of a baseline but you want the option of incorporating some rules from another level. The Show All toggle makes them accessible. For example, if you're implementing CIS Level 1 but you want to also include "Disable Media Sharing" from Level 2, you would:

- Click  **Show All**
- Click on **System Settings**
- Check **2.3.3.10 Disable Media Sharing**
- Click  **Show All**

Disable Media Sharing will now appear in the System Settings section with an "A" appearing next to the rule to signify it's been added to the custom benchmark.



Adding/Removing Rules

Organizations can add additional rules to a custom benchmark. The yaml files must follow the same format as rules in the project (https://github.com/usnistgov/macos_security/wiki/Rules) and must be prepended with one of the following:

- audit
- auth
- icloud
- os
- pwpolicy
- supplemental
- sysprefs (Pre-macOS Ventura)
- system_settings (macOS Ventura and later)

To add a new rule, do the following.

- Click ⊞. *Alternatively, click* ***Rules -> Import Rule*** *(⌘ I)*
- Browse to the rule to be imported and click **Open**

Once a rule is imported, the main project window will refresh making the rule and its contents available. The imported file can also be found under rules -> other in the project folder.

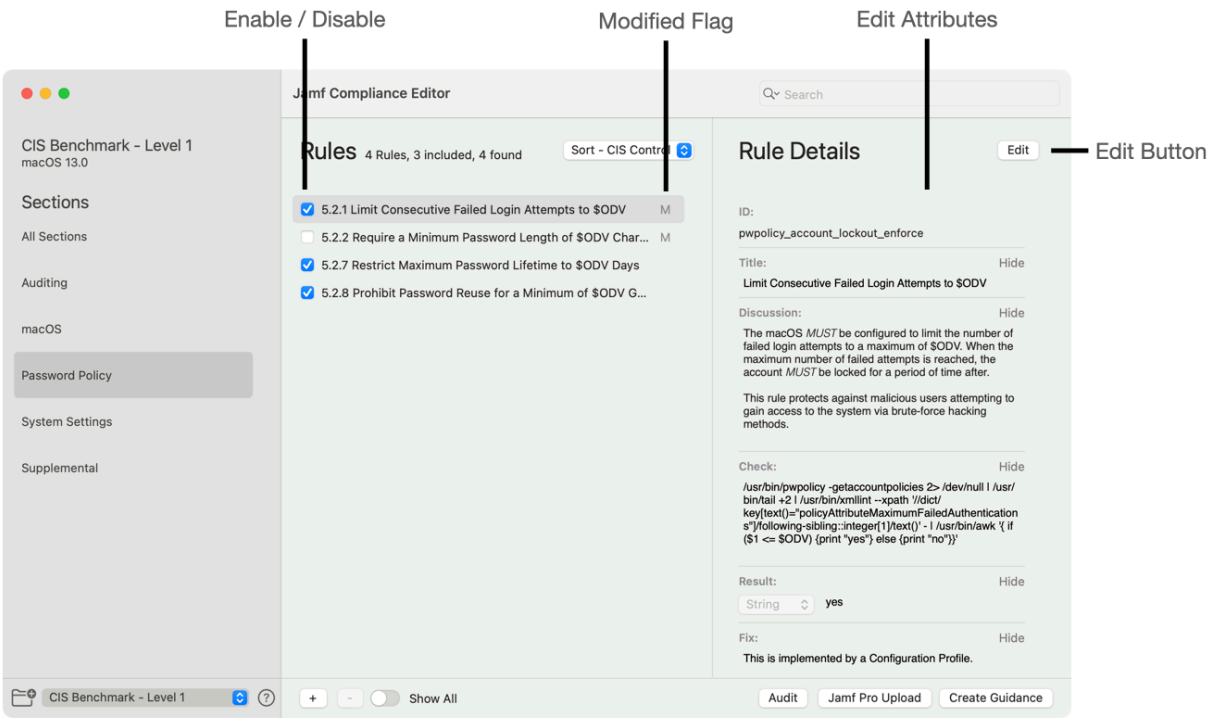
To remove a rule which was previously added, do the following.

- Click ⊟. *Alternatively, click* ***Rules -> Delete Rule*** *(⌘ D)*

Editing Rules

Each rule can also be edited, allowing an organization to override the default values. When a rule is modified an “M” will appear next to the rule to signify there are changes. To edit a rule do the following.

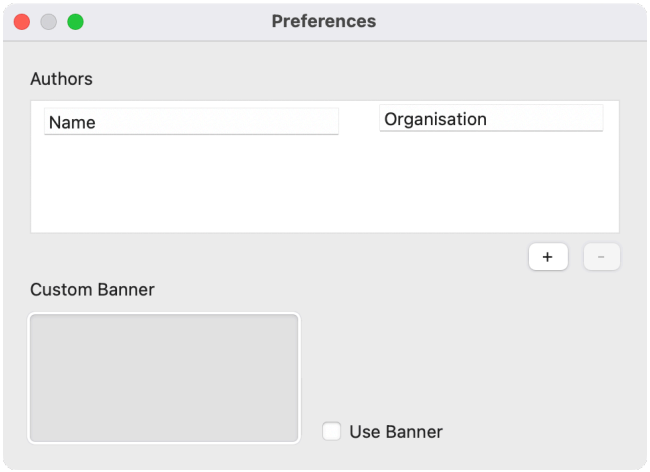
- Click **Edit**
- Click **Show** next to the field that will be customized**.** Alternatively hit ⇧ ⌘ **D** on the keyboard to expand all the rule details.
- Make any necessary changes
- Click **Done**



Adding Authors

If any changes are made to a baseline/benchmark an additional authors field can be added to the documentation. To add additional authors to the documentation do the following.

- Select **Jamf Compliance Editor** from the menu bar
- Click on **Settings**
- Under Authors, Click ⊞
    - Add **Name** & **Organization**



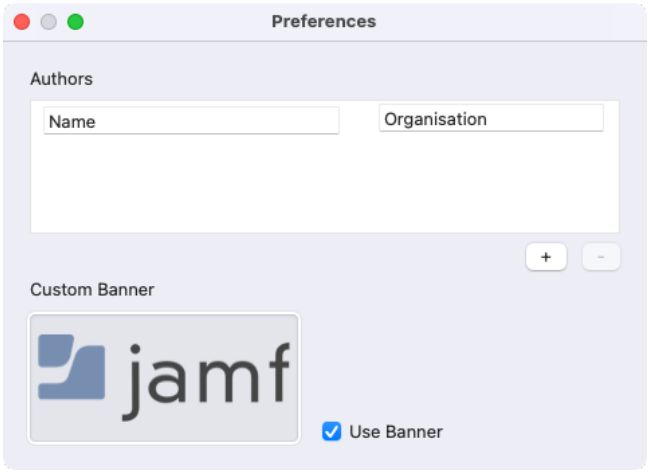
The additional authors will be added to the HTML & PDF documents upon clicking **Generate Guidance.**

NOTE: Disabling a rule from a baseline/benchmark will not modify the authors field in the generated documents.

Custom Banner

Organizations can replace the default mSCP banner on the HTML and PDF documents with your own graphic. To replace the banner, follow these steps:

- Prepare a .png picture you want to use. The graphic should not be more than 500 pixels in height.
- Select **Jamf Compliance Editor** from the menu bar
- Click on **Settings**
- Drag the replacement banner into the Custom Banner field.
- Click **Use Banner**



This banner will now be used when you Create Guidance.

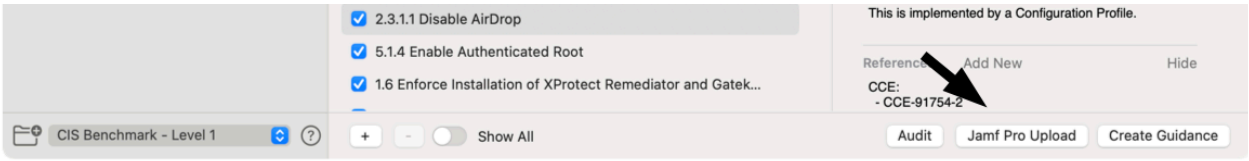
Uploading to Jamf Pro

Use Safe Practices!

Supply chain attacks have happened to even the most security-conscious organizations. Caution should be exercised when using this app just as with any content sourced from the internet. Be aware that this application incorporates content from a number of Jamf and non-Jamf sources. Initial implementation of a new process in a test environment improves the chances you'll catch problems before they can effect something important. Always do a careful inspection of any content uploaded to Jamf Pro before scoping it to user devices to make sure you fully understand what it's doing. Deploy and test new content gradually, initially scoping it to a small number of non-production test devices, then expanding the scope over time to increasingly larger groups of user devices.

The Jamf Pro Upload Button

Once you've finished creating your customized benchmark and used the Create Guidance button, you can then upload the configuration profiles, the compliance script, and Extension Attributes to Jamf Pro.



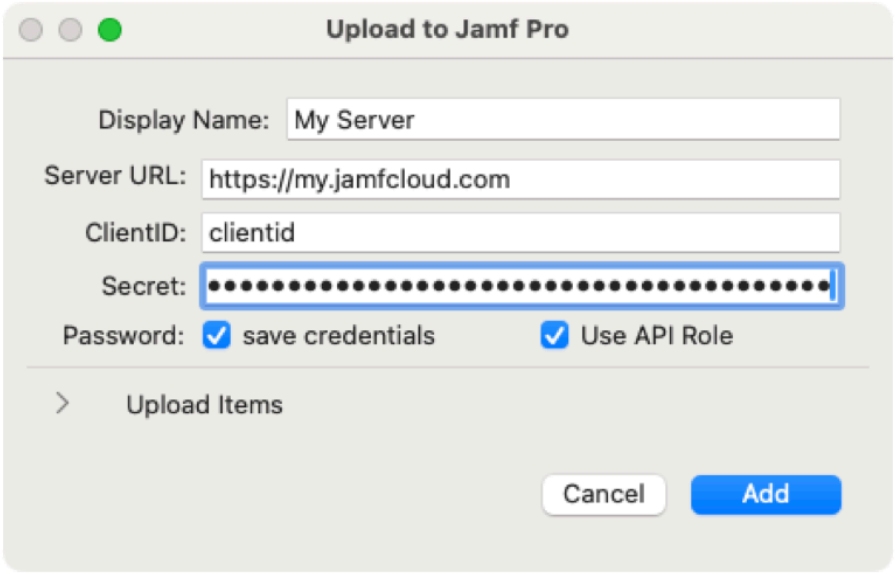
The following Jamf Pro API role privileges are required to utilize this feature:

Jamf Pro Server Object	Create	Read	Update	Delete
Categories	x			
Computer Extension Attributes	x	x	x	
macOS Configuration Profiles	x	x	x	
iOS Configuration Profiles	x	x	x	
Scripts	x	x	x	

In addition to the content that gets uploaded automatically, there's also a custom profile schema definition .json file that needs to be uploaded manually via the Jamf Pro console. Instructions for uploading this file are in the "Custom JSON Schema" section of the Jamf Pro setup instructions that follow.

Procedure

1. Click **Jamf Pro Upload**
2. Enter **Jamf Pro Server URL** and **ClientID/Secret**. This information will be saved to your macOS login keychain if you check “save credentials”.
3. Click **Continue**



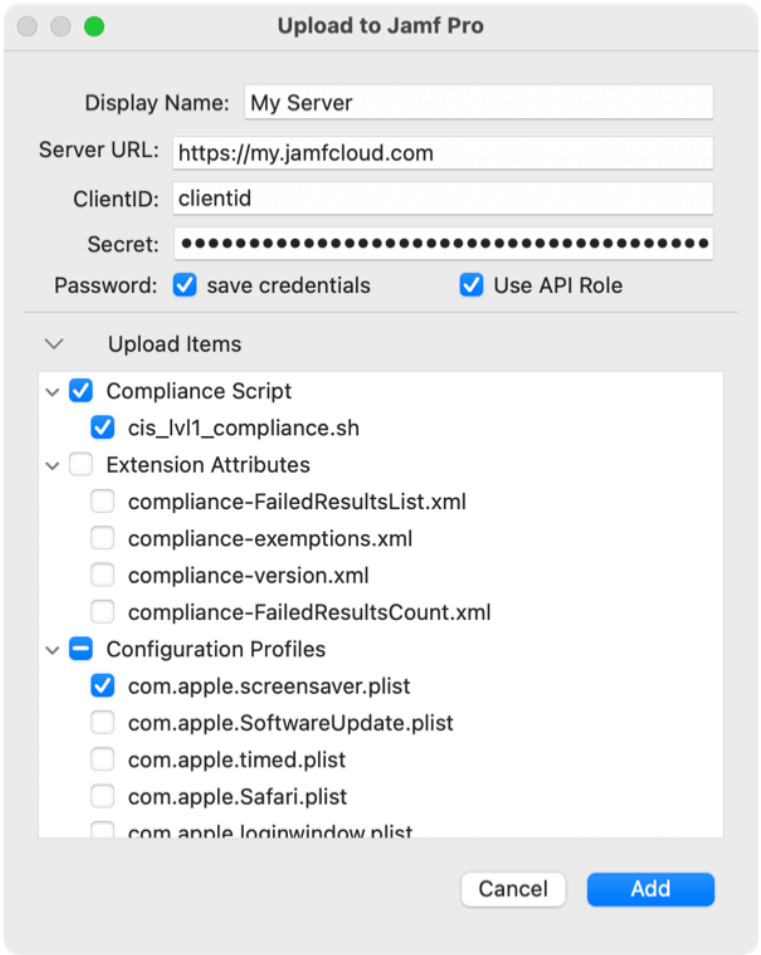
Note: If running Jamf Pro 10.49 or earlier, a Jamf API/Username can be used by unchecking **Use API Role.**

Uploaded Items

When using the “Jamf Pro Upload” button the following actions are performed:

1. A category is created in Jamf Pro with the name of the <Operating System>_<baseline>, for example, “Ventura_cis_lvl1”
2. Generated profiles are uploaded to Jamf Pro and labeled with the category
3. The compliance script is uploaded to Jamf Pro and labeled with the category
4. Four Extension Attributes are created:
   a. *Compliance - Exemptions*
   b. *Compliance - Failed Result List*
   c. *Compliance - Failed Results Count*
   d. *Compliance - Version*

Note: Jamf Compliance Editor 1.2 and later allows you to choose what to upload to Jamf Pro.



Scoping Content

The Configuration Profiles and Compliance script which are uploaded via the “Jamf Pro Upload” button will be un-scoped. Take the opportunity to inspect them before starting any test deployments.

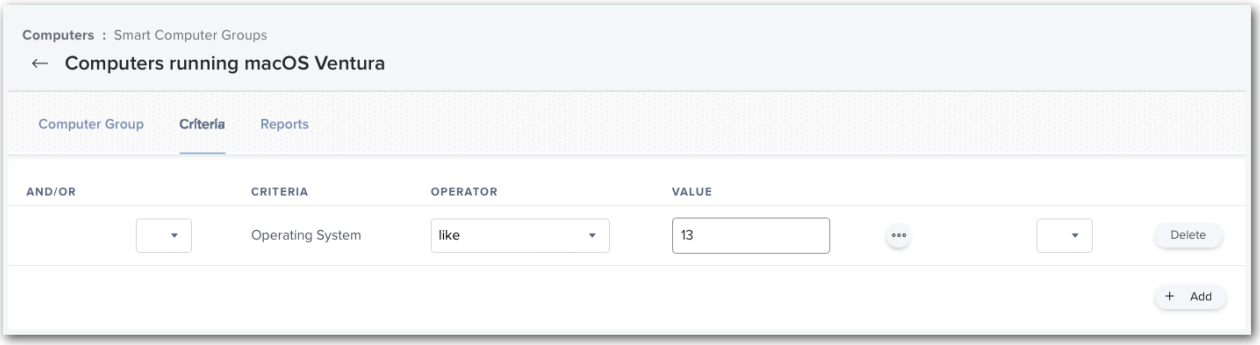
Jamf Pro Configuration (macOS)

We're using Ventura/CIS Benchmark Level 1 as an example in this section but the process is the same for other versions of macOS and baselines/benchmarks.

Smart Computer Groups

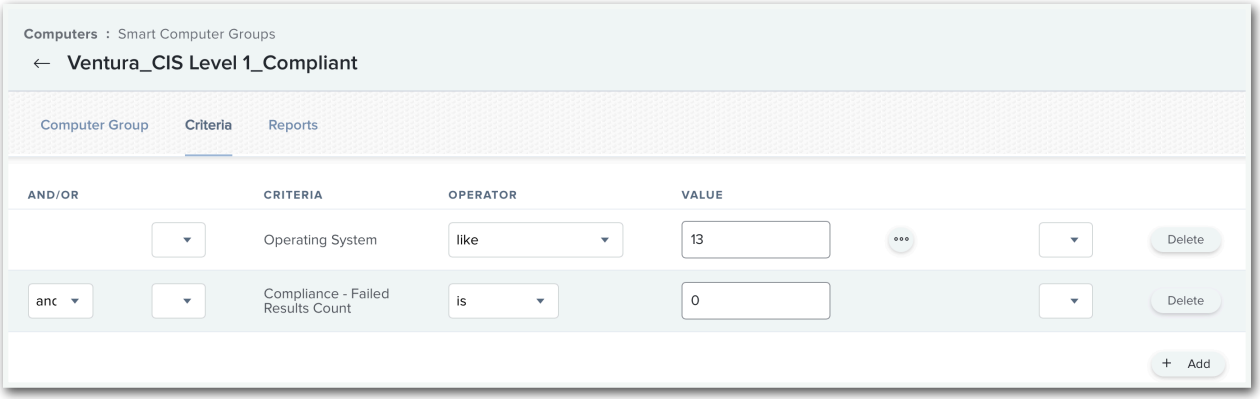
We recommend three Smart Computer Groups be created before scoping the configuration profiles and compliance script.

1st Group - **Computers running macOS Ventura**



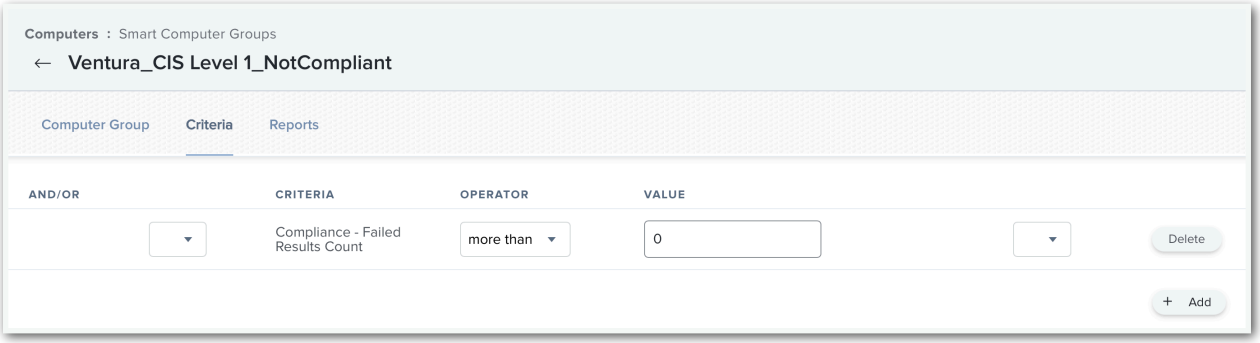
2nd Group - **Ventura_CIS_LVL1_Compliant**

The extension attribute *Compliance - Failed Results Count* will be used to calculate the membership of the following smart group. (Enter the count threshold deemed as compliant for an organization in the Value field.)



3rd Group - **Ventura_CIS_LVL1_NotCompliant**

The extension attribute *Compliance - Failed Results Count* will be used to calculate the membership of the following smart group. (Enter the count threshold deemed as non-compliant by your organization in the Value field.)

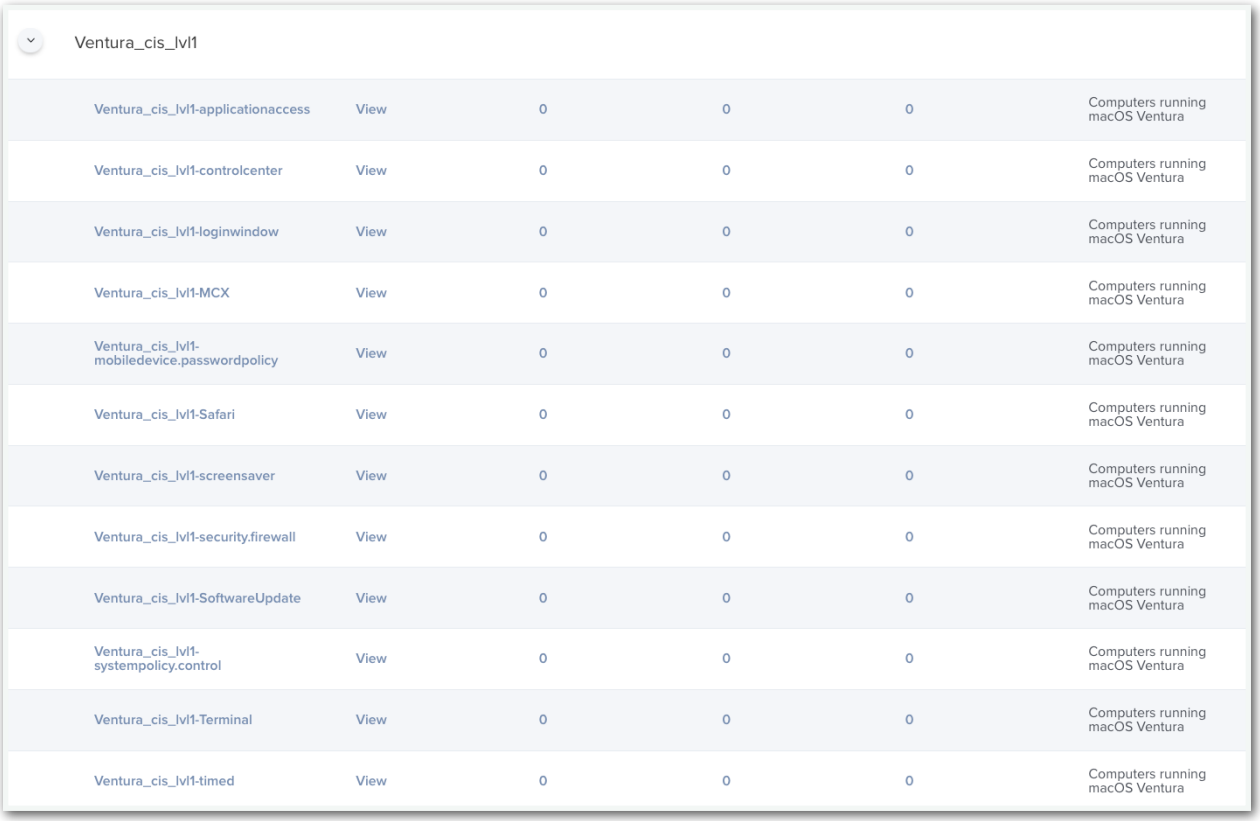


Once the Smart Computer Groups have been configured, the next step is to scope the uploaded configuration profiles and create the compliance script policies.

Configuration Profiles

To ensure that the devices running macOS Ventura get the appropriate configuration profiles applied, follow these steps.

1. In Jamf Pro, click **Computers** at the top of the sidebar
2. Click **Configuration Profiles** in the left pane
3. Scope each profile under **Ventura_cis_lvl1** to the smart group **Computers running macOS Ventura**.



Ventura_cis_lvl1

Ventura_cis_lvl1-applicationaccess	View	0	0	0	Computers running macOS Ventura
Ventura_cis_lvl1-controlcenter	View	0	0	0	Computers running macOS Ventura
Ventura_cis_lvl1-loginwindow	View	0	0	0	Computers running macOS Ventura
Ventura_cis_lvl1-MCX	View	0	0	0	Computers running macOS Ventura
Ventura_cis_lvl1-mobiledevice.passwordpolicy	View	0	0	0	Computers running macOS Ventura
Ventura_cis_lvl1-Safari	View	0	0	0	Computers running macOS Ventura
Ventura_cis_lvl1-screensaver	View	0	0	0	Computers running macOS Ventura
Ventura_cis_lvl1-security.firewall	View	0	0	0	Computers running macOS Ventura
Ventura_cis_lvl1-SoftwareUpdate	View	0	0	0	Computers running macOS Ventura
Ventura_cis_lvl1-systempolicy.control	View	0	0	0	Computers running macOS Ventura
Ventura_cis_lvl1-Terminal	View	0	0	0	Computers running macOS Ventura
Ventura_cis_lvl1-timed	View	0	0	0	Computers running macOS Ventura

Policies

The compliance script provided by mSCP will audit all the rules within the chosen baseline/benchmark and remediates any which are **not** controlled by a configuration profile. Below are example policies that can be created within Jamf pro.

Audit Policy

1. In Jamf Pro, click Computers at the top of the sidebar
2. Click **Policies** in the sidebar
3. Click **New +**
4. Configure the following:

General	• **Display Name:** Ventura_CIS Level 1_Audit • Click **Enabled** • **Trigger:** Recurring Check-in • **Execution Frequency:** Once every day

Scripts	• Click **Configure**
	  • Find **Ventura_cis_lvl1_compliance.sh**
	  • Click **Add +**
	• **Parameter Values**
	  • **Parameter 4:** --check

Maintenance	• Click **Configure** • Check **Update Inventory**

5. Click the **Scope** tab and configure the scope.
   a. The Selected Deployment Targets is **Computers running macOS Ventura.**
6. Click **Save.**

Remediation Policy

1. In Jamf Pro, click Computers at the top of the sidebar
2. Click **Policies** in the sidebar
3. Click **New +**
4. Configure the following:

General	• **Display Name:** Ventura_CIS Level 1_Remediation • Click **Enabled** • **Trigger:** Recurring Check-in • **Execution Frequency:** Ongoing

Scripts	• Click **Configure**
	  • Find **Ventura_cis_lvl1_compliance.sh**
	  • Click **Add +**
	• **Parameter Values**
	  • **Parameter 4:** --check
	  • **Parameter 5:** --fix

Maintenance	• Click **Configure** • Check **Update Inventory**

5. Click the Scope tab and configure the scope.
    a. The Selected Deployment Targets is **Ventura_CIS_LVL1_NotCompliant.**
6. Click **Save.**

Reset Policy

When testing your initial configuration you may make changes before settling a final baseline. During this time you might need to reset the plist which the EAs use to calculate compliance. The `--reset` flag can be used to reset the state of the plist between scans.

1. In Jamf Pro, click Computers at the top of the sidebar
2. Click **Policies** in the sidebar
3. Click **New +**
4. Configure the following:

General	• **Display Name:** Reset Baseline • Click **Enabled** • **Trigger:** Custom • **Custom Event:** cis_reset • **Execution Frequency:** Ongoing

Scripts	• Click **Configure**
	  • Find **Ventura_cis_lvl1_compliance.sh**
	  • Click **Add +**
	• **Parameter Values**
	  • **Parameter 4:** --check
	  • **Parameter 5:** —reset

Maintenance	• Click **Configure** • Check **Update Inventory**

5. Click the Scope tab and configure the scope.
    a. The Selected Deployment Targets is **All Managed Computers.**
6. Click **Save.**

This policy can then be offered in Self Service or by running the following on the endpoint:

```
sudo jamf policy –event cis_reset
```

Custom JSON Schema

As mentioned previously, there is one item that the Jamf Pro Upload button does not cover, therefore manual upload is needed. The <baseline name>.json custom schema file can be used to create a custom application settings profile. This profile will be read by the compliance script and Extension Attributes to determine any exemptions to rules that a user in an organization has approval for. This ensures that the compliance checks can succeed without the result count going up.

Once the app has created the guidance files, locate the `jamfpro` folder within the exported build folder. For example, if the user "`Admin`" saved "`CIS Benchmark – Level 1`" for Ventura to the Documents folder, the path would be:

```
/Users/Admin/Documents/macos_security-ventura/build/cis_lvl1/jamfpro/cis_lvl1.json
```

Follow these steps to upload the custom schema into Jamf Pro. For this example, os_airdrop_disable will be set to exempt.

1. In Jamf Pro, click **Computers** at the top of the sidebar
2. Click **Configuration Profiles** in the sidebar
3. Click **New** +

General	**Name:** Ventura_cis_lvl1_AirDrop_Exemption **Category:** Ventura_cis_lvl1 **Level:** Computer **Distribution Method:** Install Automatically

Application & Custom Settings	• Click **External Applications**
	  • Click **Add +**
	• **Source:** Custom Schema
	• **Preference Domain:** org.cis_lvl1.audit
	• Click **+ Add Schema**
	  • Click **Upload**
	  • Navigate to the jamfpro build folder, choose **cis_lvl1.json** and click **Upload**
	  • Click **Save**
	• Find **os_airdrop_disable***
	  • Set to **Configured**
	  • Set exempt to **true**
	  • Set exempt_reason to **“AirDrop is necessary for this user’s job”**

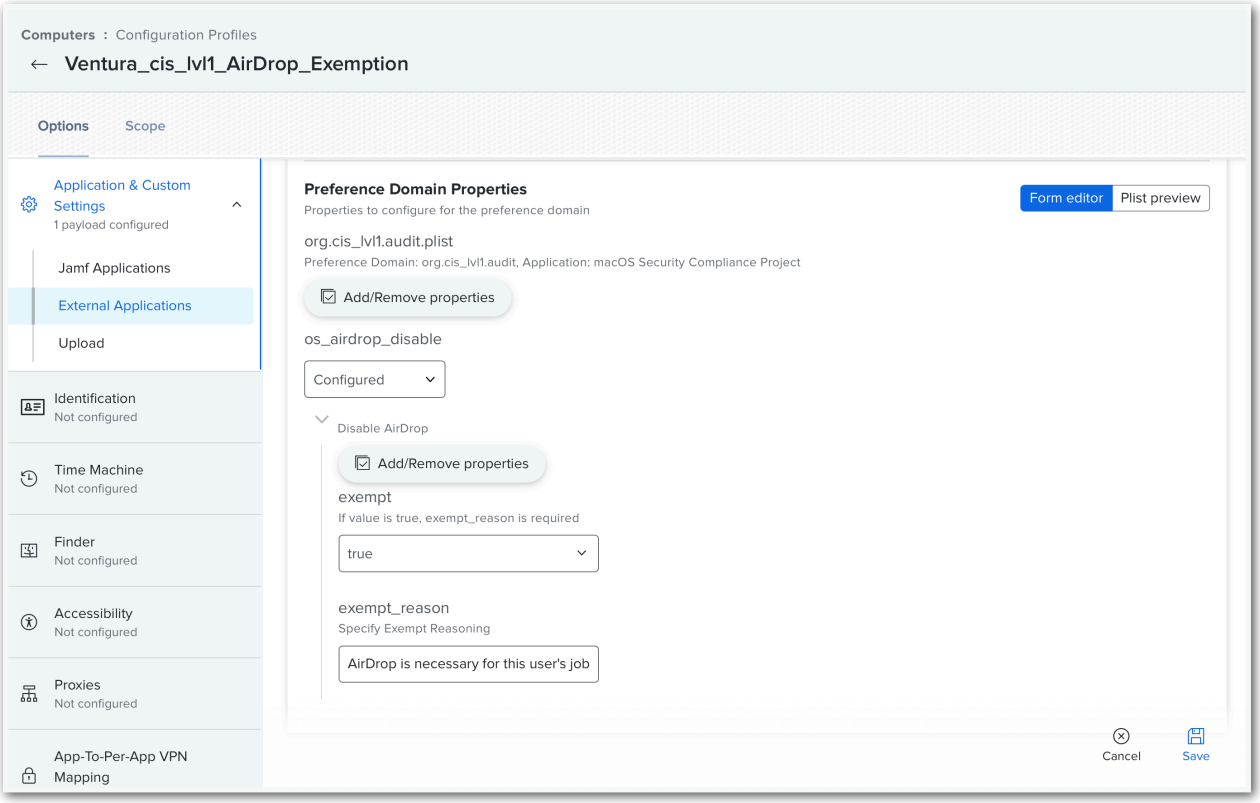
* This is one example of many that could be configured for exceptions

4. Click the Scope tab and configure the scope.

    a. The Selected Deployment Targets should be the system identified as requiring AirDrop.

5. Click **Save**.

The next time the targeted system's compliance is calculated os_airdrop_disable will show up as a finding but will not be added to the *Failed Results Count* EA.



Extension Attributes

Once the configuration profiles and scripts are scoped, the included Extension Attributes (EA) can be used to measure an organization's state of compliance.

Compliance - Exemptions

The custom JSON schema which needs to be manually uploaded to Jamf Pro can be used to determine any exemptions that an organization needs when calculating compliance. When using the custom schema, the exemptions are written to /Library/Managed Preferences/org.<baseline>.audit.plist. This EA will display which exemptions are in place on a given system.

Compliance - Failed Result List

The failed result list EA displays which rules in a given baseline/benchmark are not being met. One thing to note, rules that are exempt will still show up in this list. Being exempt from a rule is still considered a finding and needs to be reported.

Compliance - Failed Results Count

The failed results count EA calculates what rules are being met, but also calculates exemptions. Exemptions are not counted towards the total results count. For example, if a system failed five rules, but had an exemption in place for two rules, the EA would be three. This EA is used for calculating membership to the smart group used for remediation policy.

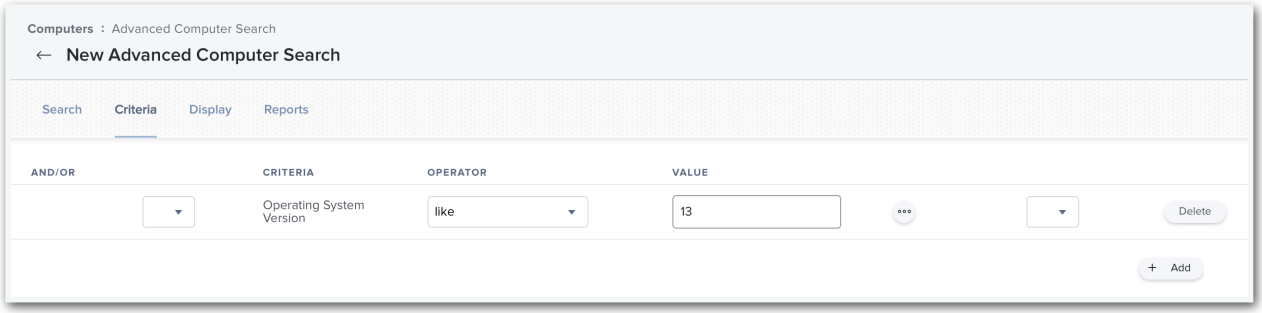
Compliance - Version

The version EA displays which version of the compliance script is currently auditing a system. This is useful when preparing for macOS major upgrades or if an organization is planning to make changes to what baseline/benchmark they are trying to align with.

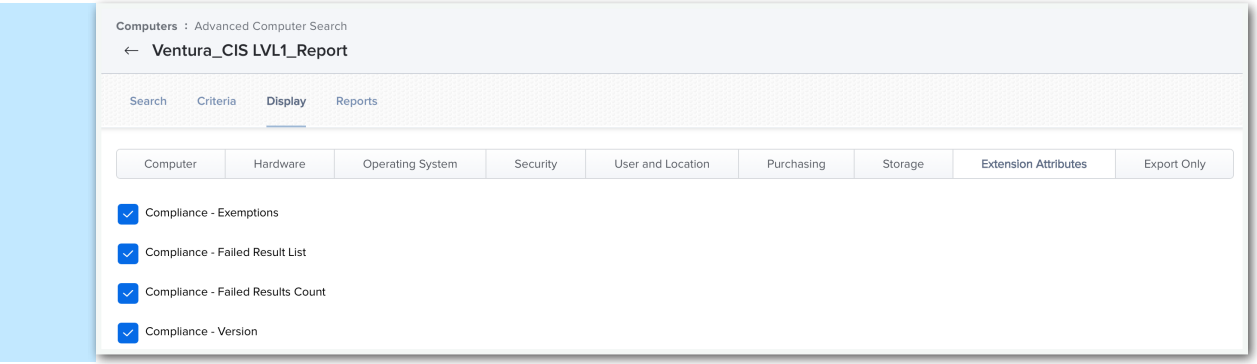
Jamf Pro Reporting

Jamf Pro administrators can generate a report detailing what computers in their environment are compliant and what computers require remediation. In these cases, an admin can run a Report from a saved Advanced Computer Search in Jamf Pro.

1. In Jamf Pro, click **Computers** at the top of the sidebar
2. Under **Advanced Computer Searches,** click **+ New**
3. Check **Save this search** and name it **Ventura CIS Benchmarks Level 1 Report**
4. Click **Criteria** and add **Operating System Version,** set **Operator** to **like** and the **Value** to **13** (reflecting macOS Ventura version 13.0)



5. Click **Display** and select the **Extension Attribute** tab
    1. Add each of the compliance extension attributes:
    Compliance - Exemptions
    Compliance - Failed Result List
    Compliance - Failed Results Count
    Compliance - Version



    2. Add any additional inventory information under **Display**, to be included in the report.

6. Click **Save**

7. Click **View**

Once an advanced search is created, click on **Reports** and export a report in .csv, .tsv, or .xml format. This report can now be used as needed to retrieve reporting from Jamf Pro on the compliance of each of an organization's endpoints. If you have SMTP configured in your Jamf Pro you can request that the report be emailed periodically.

Jamf Pro Configuration (iOS/iPadOS/visionOS)

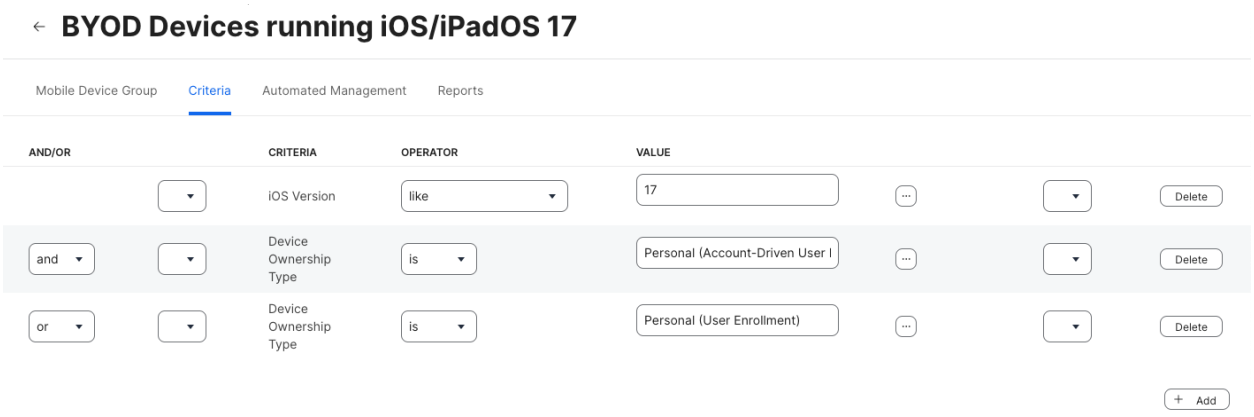
We're using iOS17/CIS Benchmark Level 1 BYOD as an example in this section but the process is the same for other versions of iOS/iPadOS and baselines/benchmarks.

Smart Mobile Device Groups

We recommend only one Smart Mobile Device Group be created before scoping the configuration profiles.

BYOD - iOS/iPadOS devices running iOS 17

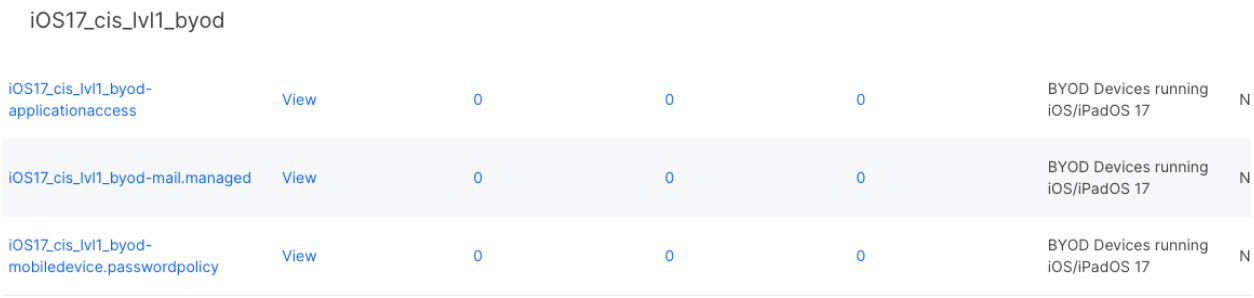
* Recognizing the "OR" Status for the following criteria of Device Ownership Type



Configuration Profiles

To ensure that the BYOD devices running iOS/iPadOS 17 get the appropriate configuration profiles applied, follow these steps.

4. In Jamf Pro, click **Devices** at the top of the sidebar
5. Click **Configuration Profiles** in the left pane
6. Scope each profile under **iOS17_cis_lvl1_byod** to the smart group **Devices running iOS/iPadOS 17**



Remediation/Scripts for iOS/iPadOS/visionOS

The ability to audit or remediate does not exist for iOS/iPadOS/visionOS. Once the configuration profile has been validated as deployed by the MDM server it is considered compliant. There are no scripts that can audit or remediate an iOS/iPadOS/visionOS device, nor are Jamf Pro Extension Attributes available.

Audit (macOS Only)

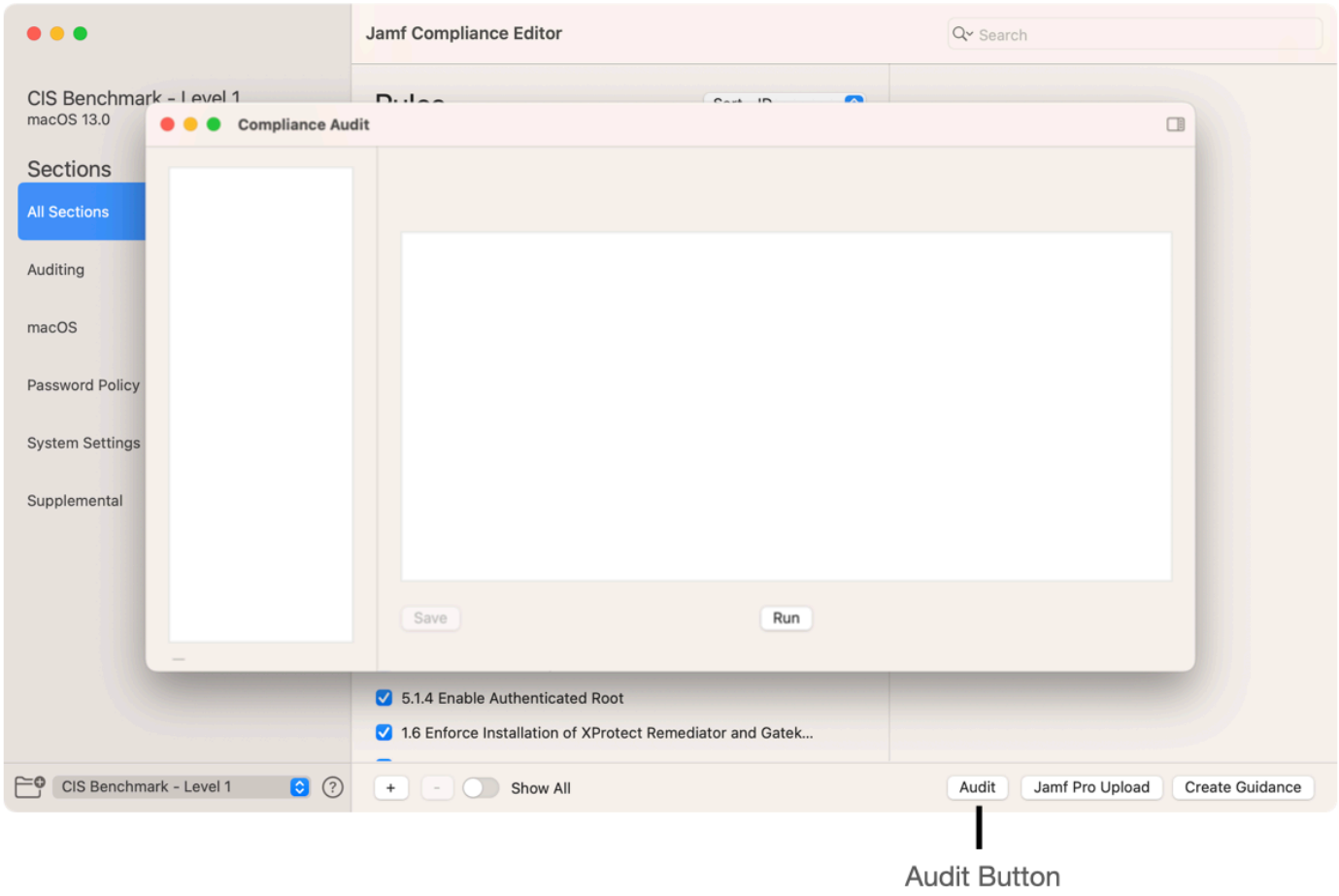
Jamf Compliance Editor 1.2 and later allows an administrator to run an on demand audit of the device running the application. This can inform the admin of the state of that macOS device against the benchmark generated.

After customization of your security benchmark and completion of the Create Guidance action, Jamf Compliance Editor will allow you to execute a full scan or audit of the local Mac against the customized benchmark.

Execute Audit

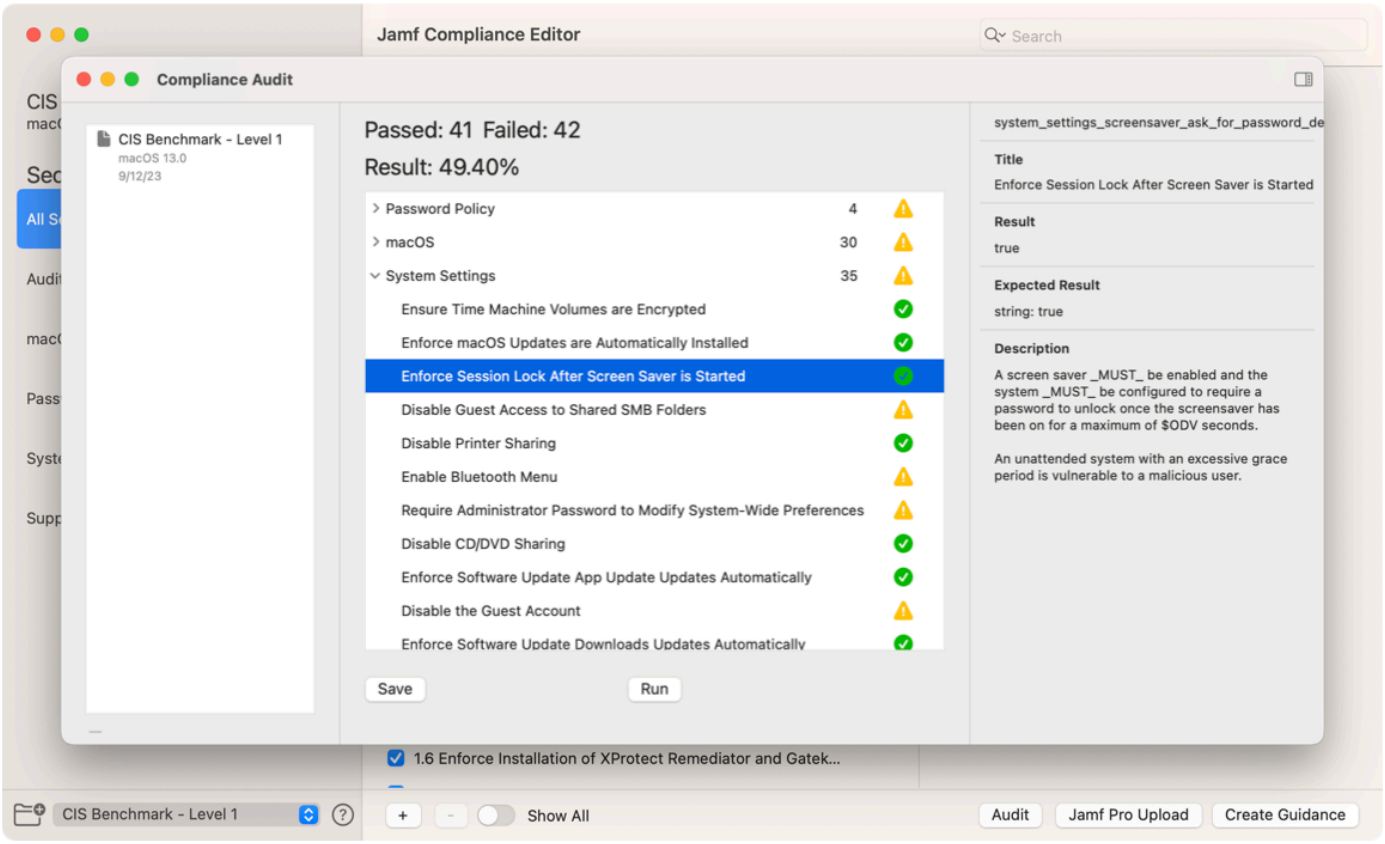
Within the Jamf Compliance Editor window, the Audit button resides to the left of the Jamf Pro Upload button in the lower right. If it is greyed out it indicates that the Create Guidance has not been completed on this benchmark.

Click the Audit button, then when ready to audit the local Mac click Run:



Review Audit Results

The Jamf Compliance Editor window will present the results within the same Audit pop-up window. Historical audits will also appear as a list. The results of a scan can be saved to a .csv for further inspection by selecting the previous scan and clicking **Save**.



Appendix 1 — Application Change Log

Version 1.4.0

- Support for macOS 15, iOS/iPadOS 18, and visionOS 2
- Added network check
- Prompt when moving project folder
- Revert button for rules
- Bug Fixes

Version 1.3.1

- Bug Fixes
- Addressed local privilege escalation vulnerability - CVE-2024-4395
  - Thanks to Mykola Grymalyuk of RIPEDA Consulting

Version 1.3

- Added support for iOS/iPadOS branches
- Bug Fixes

Version 1.2.1

- Added support for macOS Sonoma

Version 1.2

- Set minimum macOS version to 13
- Added audit feature
- Added showAllBranches flag
- Added ability to import custom versions of existing rules
- Added ability to search by references
- Prompt to save .jce file after generate guidance is complete
- Jamf Pro Upload Enhancements
  - Added support for API Roles/Clients
  - Add ability to upload to multiple Jamf Pro servers
  - Select what files to upload
- Resolved an issue with Configuration Profile uploads
- Bug Fixes

Version 1.1.5 - Bug Fixes

- Resolved an issue where generate guidance would not create any files
- Resolved an issue where prompt to rename baseline did not appear

- Mobileconfig folder is refreshed after clicking “Create Guidance” button

Version 1.1.4 - Bug Fixes

- Resolves an issue with organization defined values and profiles

Version 1.1.3 - Bug Fixes

- Resolves an application crash during Jamf Pro uploads
- Added unified logging events for Jamf Pro uploads

Version 1.1.2 - Bug Fixes

- Resolves an application crash when Show All is toggled

Version 1.1.1

- Unchecking a rule will trigger the option to rename the baseline
- Jamf Pro upload button now works with renamed baselines
- Previous `.jce` files display the correct selected rules when opened

Version 1.1

- Added application update mechanism
- Added baseline update mechanism
- Added tailoring support
  - Prompt to rename baseline when changes are made
  - Added author support
  - Added custom banner support
- Added Show All function for Rules
- Extension attributes now added to EA field in Jamf Pro
- Suppress warning when editing CIS or DISA STIG
- Search term to display rules containing “*mobileconfig*”

Version 1.0.3 - Bug Fixes

- DISA STIG artifacts now upload properly

Version 1.0.2 - Bug fixes

- `.GlobalPreferences` payload now applied properly
- `com.apple.mobiledevice.passwordpolicy` now uses the Jamf Pro built-in Configuration profile payload
- Fixed organizational defined values issue in baseline creation
- Fixed refresh issue
- Added `icloud_` to custom rules at creation

Version 1.0.1 - Internal Code Review

Version 1.0.0 - Initial Release

Appendix 2 — Troubleshooting

Accessing the logs

Compliance Editor writes logs to the macOS unified logging system. These logs can be used when troubleshooting Jamf Compliance Editor.

Enter the following in Terminal.app to review the log messages related to mscp and content creation.

```
log stream --info --debug --predicate 'subsystem ==
"com.jamf.complianceeditor" AND category == "mscp"'
```

Enter the following in Terminal.app to review the log messages related to uploading the content to Jamf Pro.

```
log stream --info --debug --predicate 'subsystem ==
"com.jamf.complianceeditor" AND category == "jamfpro"'
```

Enter the following in Terminal.app to review the log messages related to running an audit.

```
log stream --info --debug --predicate 'subsystem ==
"com.jamf.complianceeditor" AND category == “audit"'
```

Appendix 2 — Known Issues

PI120483

When creating an iOS 17 baseline using indigo Base/High, the configuration profile for restrictions (com.apple.applicationaccess) will not upload properly unless rules are de-selected from the baseline. This is due to a product issue in Jamf Pro related to sub restrictions in this payload.

Workaround: Sign the configuration profile that Jamf Compliance Editor outputs before uploading to Jamf Pro. (This option is not available when using the Jamf Pro Upload button within the app)

More detailed workaround:

indigo Base

To workaround this issue in indigo Base ensure these settings are configured:

In Search, type **Siri**

Uncheck Disallow user generate content with Siri

Uncheck Ensure allow Siri while device is locked is set to Disabled

In Search, type **Screen**

Uncheck Prevent remote screen observation

Uncheck Prevent unprompted screen observation

Click Generate Guidance

indigo High

To workaround this issue in indigo High ensure these settings are configured:

In Search, type **Siri**

Uncheck Disallow user generate content with Siri

Uncheck Ensure allow Siri while device is locked is set to Disabled

Uncheck Disallow Siri server side logging

In Search, type **Screen**

Uncheck Prevent remote screen observation

Uncheck Prevent unprompted screen observation

In Search, type **AirPrint**

Uncheck Requires trusted certificates for TLS printing communication

Click Generate Guidance