SentinelOne Deployment Using Meraki

Resource Preparation

First login to SentinelOne to get your desired

1. SentinelOne agent .pkg
2. Group or site token.

Make sure you have all the requirements before you start the installation.

Installation Script

You will need a helper script to install the SentinelOne agent on the remote endpoint. You can write one to suit your needs or use this:

```
#!/usr/bin/env bash
set -euxo pipefail

# Define the log file path
LOG_FILE="/tmp/sentinelone_install.log"

# Get the script's directory path
echo "Getting script directory..."
script_dir="$( cd "$( dirname "${BASH_SOURCE[0]}" )" >/dev/null 2>&1 && pwd )"
echo "Script directory: $script_dir"

# Set the file names
echo "Setting package and profile file names..."
sentinel_pkg="$(ls "$script_dir" | grep -i '\.pkg$' || true)"
sentinel_profile="com.sentinelone.registration-token"
echo "Package file: $sentinel_pkg"
echo "Profile file: $sentinel_profile"

# Find the SentinelOne package and profile in the script's directory
echo "Searching for package and profile files..."
SOURCE="$script_dir/$sentinel_pkg"
PROFILE="$script_dir/$sentinel_profile"
echo "Package file path: $SOURCE"
echo "Profile file path: $PROFILE"

# If the files are not found in the script's directory, use the original 'find' command
if [[ ! -f "$SOURCE" ]]; then
  echo "Package file not found in script directory. Searching recursively..."
  SOURCE=$(find "$script_dir" -iname "*Sentinel*.pkg" -print -quit)
fi

if [[ ! -f "$PROFILE" ]]; then
  echo "Profile file not found in script directory. Searching recursively..."
  PROFILE=$(find "$script_dir" -iname "*$sentinel_profile*" -print -quit)
fi

# Check if the files exist
if [[ ! -f "$SOURCE" || ! -f "$PROFILE" ]]; then
  echo "Error: SentinelOne package or profile not found."
  exit 1
fi

# Copy the files to the temporary directory
```

```bash
echo "Copying files to /tmp..."
cp "$SOURCE" "$PROFILE" /tmp

# Install SentinelOne and redirect stderr to the log file
echo "Installing SentinelOne..."
sudo /usr/sbin/installer -verboseR -pkg "/tmp/$(basename "$SOURCE")" -target /Library/ 2>> "$LOG_FILE"

# Remove the package and profile
echo "Cleaning up..."
sudo rm "/tmp/$(basename "$SOURCE")" "/tmp/$(basename "$PROFILE")"

# Exit successfully
echo "SentinelOne installation completed successfully."
exit 0
```

Get the Site Token or Group Token

From version S-SP2 you can use a Group Token for a Manual Group (previously a *Static Group*) or a Pinned Group.

To get the Site Token

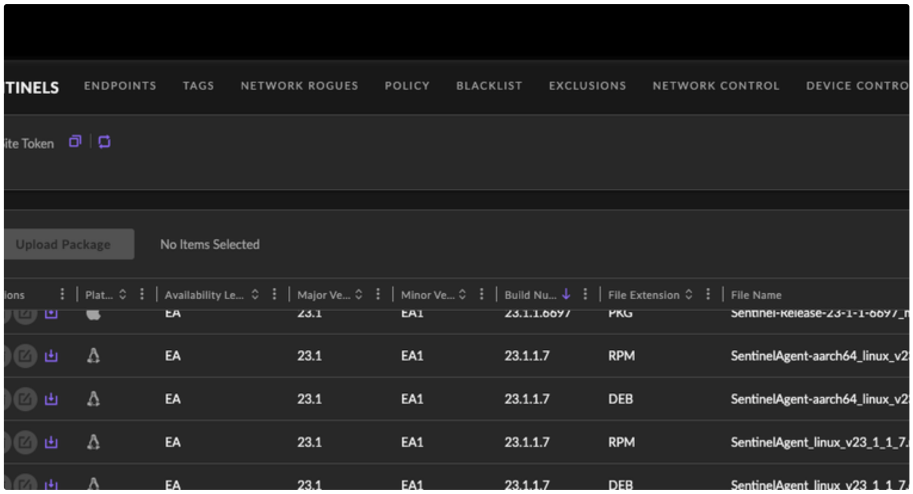
1. At the top left of the Console, click and select a scope. Select one Site. If you are in any other scope, the Site Token does not show.
2. In the **Sentinels** toolbar, click **Packages**.
3. In the **Site Token** section, click **Copy**.

To get a Group Token

1. At the top left of the Console, click the tray arrow and select a scope. You must select one Group.
2. In the sidebar, click **Sentinels**. **Endpoints** opens.
3. In the **Sentinels** toolbar, click **Group Info**.
4. In the **Group Token** section, click **Copy**.

Get SentinelOne Agent Package

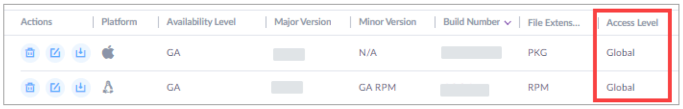
1. In the **Sentinels** toolbar, click **Packages**.



ions	Plat...	Availability Le...	Major Ve...	Minor Ve...	Build Nu...	File Extension	File Name
		EA	23.1	EA1	23.1.1.6697	PKG	Sentinel-Release-23-1-1-6697_n
		EA	23.1	EA1	23.1.1.7	RPM	SentinelAgent-aarch64_linux_v2
		EA	23.1	EA1	23.1.1.7	DEB	SentinelAgent-aarch64_linux_v2
		EA	23.1	EA1	23.1.1.7	RPM	SentinelAgent_linux_v23_1_1_7.
		EA	23.1	EA1	23.1.1.7	DEB	SentinelAgent_linux_v23_1_1_7.

2. Download the latest macOS installer package.

   Make sure the scope of the package includes the Site that the Agent will go to.



Actions	Platform	Availability Level	Major Version	Minor Version	Build Number	File Extens...	Access Level
		GA	[redacted]	N/A	[redacted]	PKG	Global
		GA	[redacted]	GA RPM	[redacted]	RPM	Global

> *Best Practice: Download the file to the local endpoint.*

Best Practice: Download the file to the local endpoint.

3. Save the **Site Token** or **Group Token** in a plain text file in a folder with the Installer package. Name the Token file: `com.sentinelone.registration-token`. Change the ownership of the file to root with `sudo chown root`.

   Important: Agent versions 3.4.x+ require the **com.sentinelone.registration-token** file to be defined for Terminal installation to be successful.

TO-DO: Document where to get mobileconfig files from the jamf portion of mac install documentation

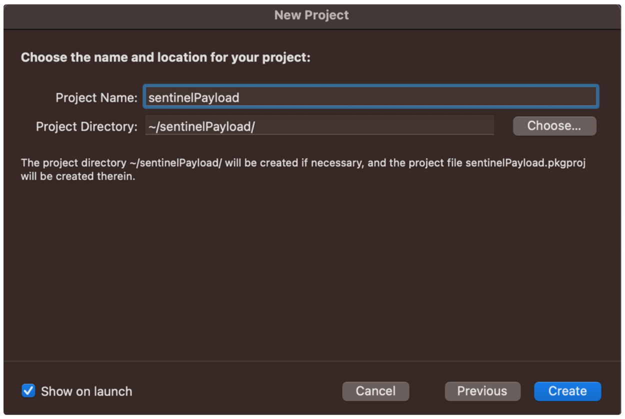
Packaging for Meraki

Create Deployment Package for Use With Meraki

Using Packages

Place the helper script, your site/group token file ***com.sentinelone.registration-token,*** and your SentinelOne agent .pkg I a new directory. This directory will be used to bundle these resources together in flat package file. This can be done using an open source tool called packages. You can also use a packing script, see more info here.

Open Packages, and create a new package, Packages>Files>New Project. Create a Raw Package.



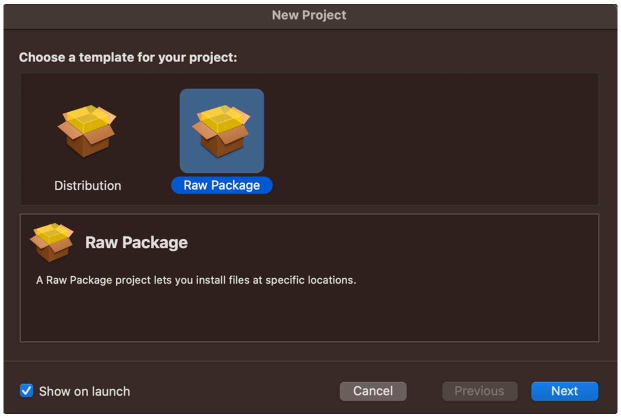
Choose the name and location for your project:

Project Name: sentinelPayload

Project Directory: ~/sentinelPayload/

The project directory ~/sentinelPayload/ will be created if necessary, and the project file sentinelPayload.pkgproj will be created therein.

☑ Show on launch



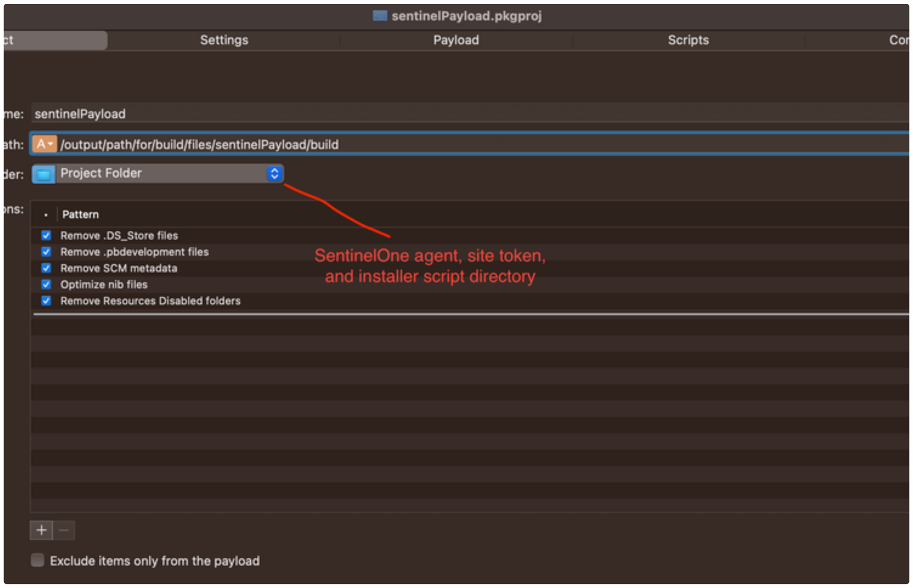
Choose a template for your project:

Raw Package

A Raw Package project lets you install files at specific locations.

☑ Show on launch

Set path to absolute, and point it to where you want the final .pkg file to be output to.



me: sentinelPayload

ath: A /output/path/for/build/files/sentinelPayload/build

der: Project Folder

ons:

Pattern

☑ Remove .DS_Store files

☑ Remove .pbdevelopment files

☑ Remove SCM metadata

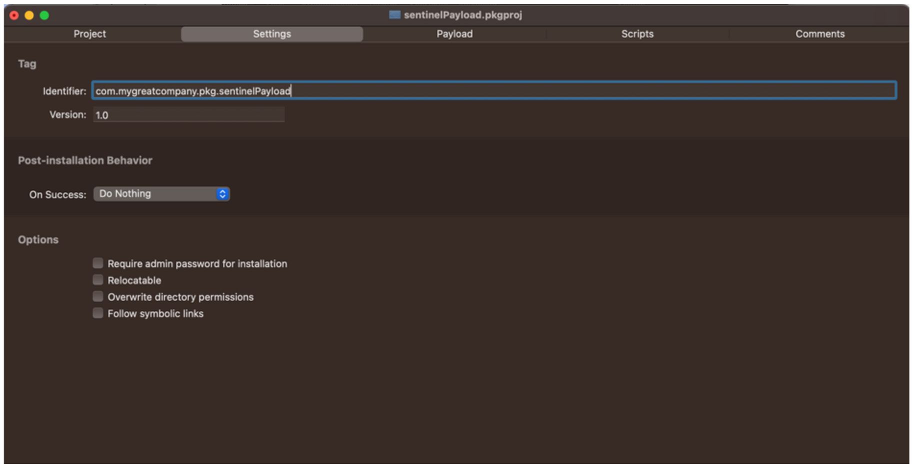
☑ Optimize nib files

☑ Remove Resources Disabled folders

☐ Exclude items only from the payload

SentinelOne agent, site token, and installer script directory

Set your identifier, and version. It is recommended version match the SentinelOne agent version. If you are wanting a silent install, set On Success to Do Nothing, and uncheck all additional options.



Tag

Identifier: com.mygreatcompany.pkg.sentinelPayload

Version: 1.0

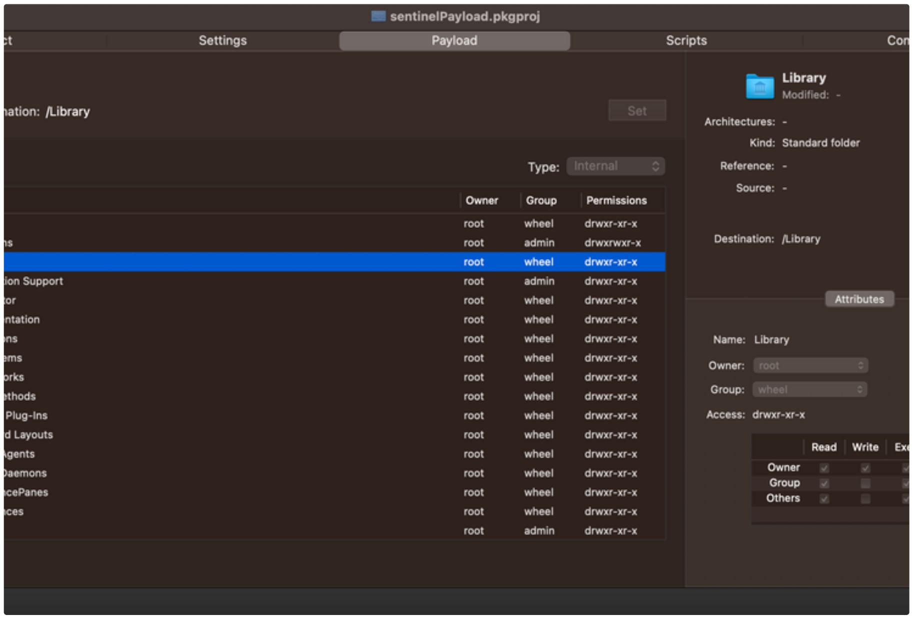
Post-installation Behavior

On Success: Do Nothing

Options

□ Require admin password for installation
□ Relocatable
□ Overwrite directory permissions
□ Follow symbolic links

Set the target location as /Library



ation: /Library

Type: Internal

	Owner	Group	Permissions
	root	wheel	drwxr-xr-x
ns	root	admin	drwxrwxr-x
	root	wheel	drwxr-xr-x
ion Support	root	admin	drwxr-xr-x
tor	root	wheel	drwxr-xr-x
ntation	root	wheel	drwxr-xr-x
ns	root	wheel	drwxr-xr-x
ems	root	wheel	drwxr-xr-x
orks	root	wheel	drwxr-xr-x
ethods	root	wheel	drwxr-xr-x
Plug-Ins	root	wheel	drwxr-xr-x
d Layouts	root	wheel	drwxr-xr-x
Agents	root	wheel	drwxr-xr-x
Daemons	root	wheel	drwxr-xr-x
cePanes	root	wheel	drwxr-xr-x
ces	root	wheel	drwxr-xr-x
	root	admin	drwxr-xr-x

Library
Modified: -

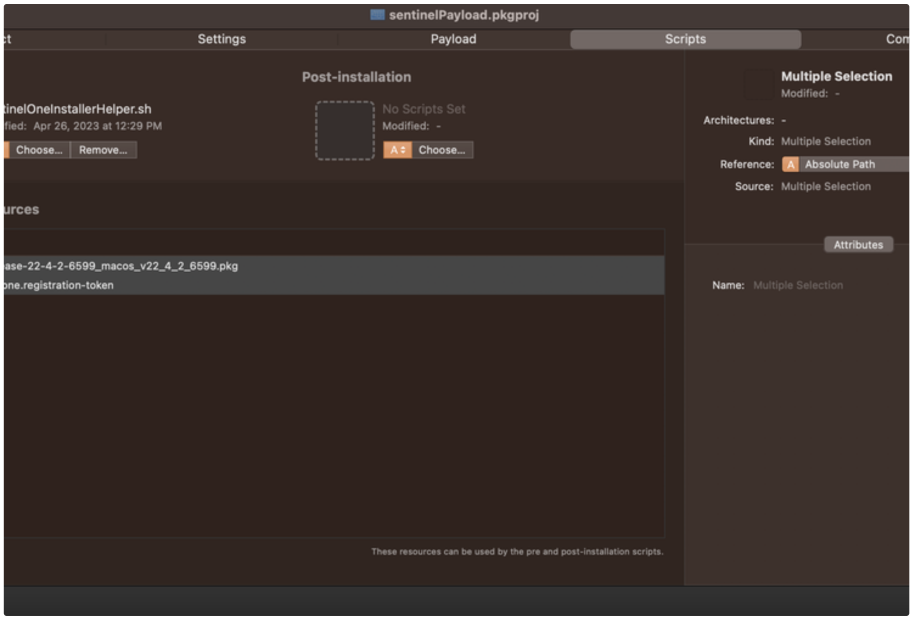
Architectures: -
Kind: Standard folder
Reference: -
Source: -

Destination: /Library

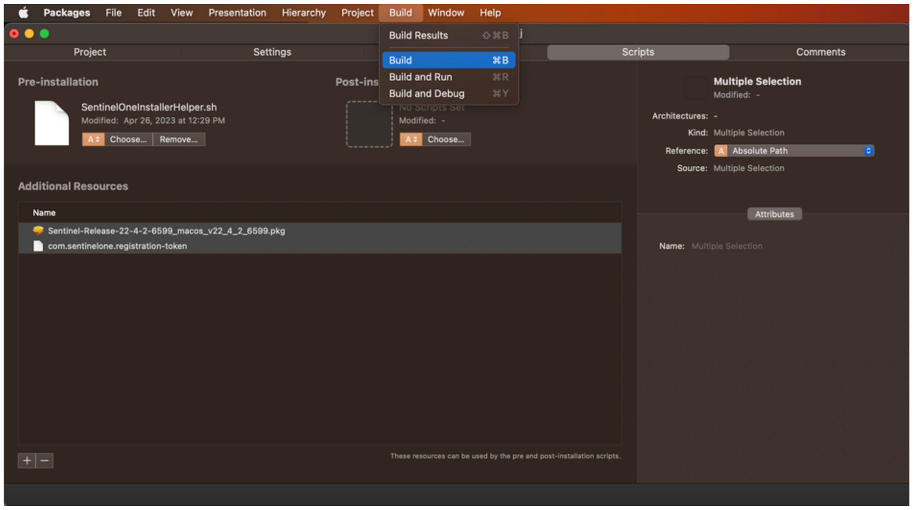
Name: Library
Owner: root
Group: wheel
Access: drwxr-xr-x

	Read	Write	Ex
Owner	☑	☑	☑
Group	☑	□	☑
Others	☑	□	☑

Set the preinstall script to the helper script you downloaded earlier, and add the remaining resources as shown below.



Build the package.

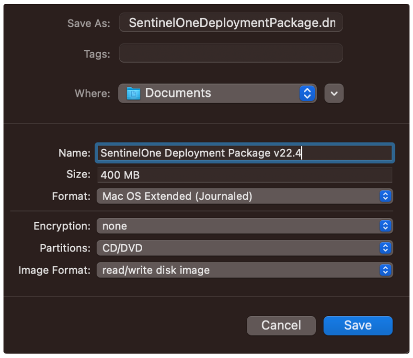


Place Installer in Deployable Image

The final step is to place these files in a .dmg image.

Using the Disc Utility app to package your application as a disk image (`.dmg` file) for distribution on OS X.

1. Open `/Applications/Utilities/Disk Utility.app` by double-clicking it.
2. From the Image menu, choose New Blank Image. Disk Utility opens a new window with customization options as shown:



Save As: SentinelOneDeploymentPackage.dn

Tags:

Where: Documents

Name: SentinelOne Deployment Package v22.4

Size: 400 MB

Format: Mac OS Extended (Journaled)

Encryption: none

Partitions: CD/DVD

Image Format: read/write disk image

3. In the “Save as” text box, enter the name of the compressed file that you will distribute. By default, a `.dmg` suffix is appended to the name you enter. Although it is not required, it is a good idea to retain this suffix for clarity and simplicity.
4. In the Volume Name text field, enter the name of the volume that you want to appear in the Finder of a user’s computer. Usually this name is the same as the name of the compressed file without the `.dmg` suffix.
5. In the file browser, set the location to save the file on your computer. This has nothing to do with the installation location on the end user’s computer, only where it saves it on your computer.
6. Set the Size pop-up menu to a size that is large enough to hold your application (2x the size of resources being placed in it).
7. Leave the Format set to Mac OS Extended (Journaled).
8. Leave Encryption set to none. If you change it, the end user must enter a password before the image can be mounted, which will not work with Meraki.
9. Click Save.

Once you have a disk image, mount it by double-clicking it. You can now copy your files to that mounted image. When you have everything on the image that you want, you should make your image read-only. Again from Disk Utility, perform these steps:

1. Unmount the disk image by dragging the volume to the Trash, clicking the eject button next to the device in a Finder window, or selecting the mounted volume and choosing Eject from the Finder’s File menu.
2. Choose Convert Image from the Image menu.
3. In the file browser, select the disk image you just modified and click Convert.
4. Choose a location to save the resulting file, change the image format to read-only, and click Convert.

You now have a disk image for your application that is easy to distribute.

Scripted Packaging and Deployment Image Creation

If you created a tmp folder with the installation script, site token, and package, you can use this script to combine them into a deployable Image for Meraki. Copy the path to the tmp directory ahead of time and the run the script:

https://github.com/usmc0341/pkg-dmg-creator - Connect your Github account

Meraki Deployment Overview

This guide covers deploy SentinelOne via meraki for a silent install. This requires the predeployment of mobileconfig files to grant the SentinelOne agent needed permissions, and creation of a SentinelOne Application in Meraki System Manager.

The process looks like this:

1. Prepare SentinelOne Deployment Package.
2. Upload mobileconfig files to Merkai SM.
3. Add SentinelOne Deployment Package to Meraki SM applications catalog.
4. Create Meraki SM Target Group.
5. Tag target devices for profile installation.
6. Verify profile installation to target devices and tag them for deployment.

Upload Mobile Config Files To Meraki

Prior to pushing the agent, you can create a silent installation by pre installing the .mobileconfig profiles to the target endpoints. Template profiles can be found on the sentinelone support site in the section for installing with jamf. The SentinelOne macOS Agent, compatible with the latest macOS release, is always released after complete development and testing. Target is GA-level support for every new major macOS release, as quickly as possible. The time required is dependent on the changes introduced by Apple in their latest GA release. Typically, it takes a few days. In rare cases, the support period may exceed 45 days, due to the complexity of Apple changes between the Developer version and the GA release, and between the new major release and the previous release.

- If an unprotected device has macOS Big Sur and you try to deploy the 4.6 Agent, the Agent requires Full Disk Access permission. If the user does not grant permissions, the endpoint will not be protected.
- If an unprotected device has High Sierra and you try to deploy the 4.3 Agent, the OS will ask for permission to change the kernel extension. Whether the user approves or not, the Agent will not work.

  For more information, see the official Apple post for the User-Approved Kernel

  Extension Loading feature and High Sierra changes.
- For corporate-owned devices, you can pre-approve the SentinelOne Agent kernel

  extension in your MDM (Mobile Device Manager, such as JAMF/Casper or
  MobileIron). This is supported from macOS High Sierra 10.13.2. Use these parameters:

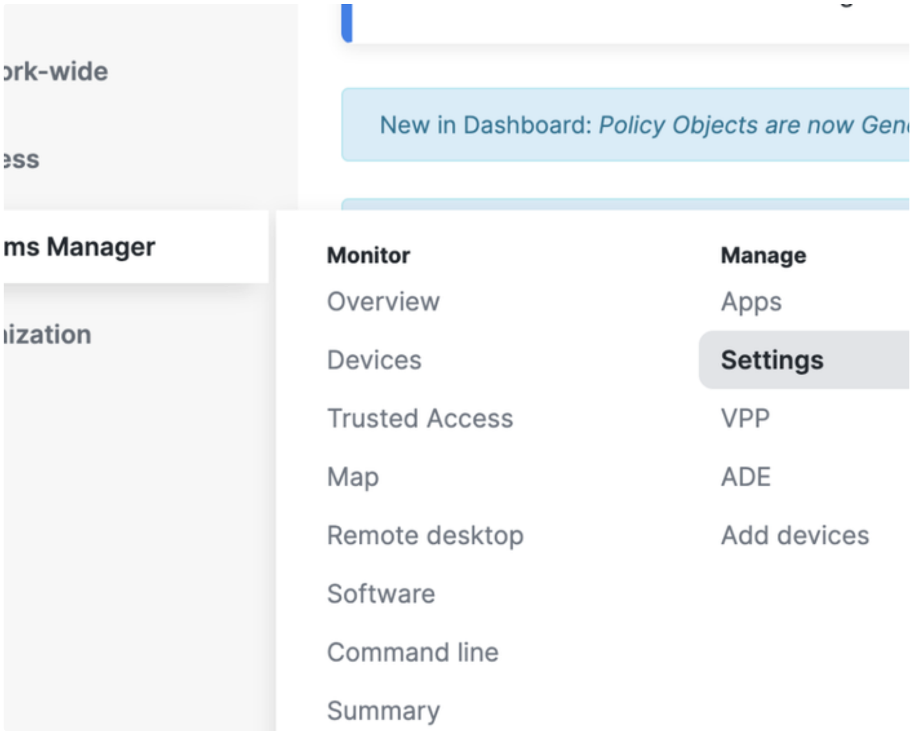
o KextBundle ID - com.sentinelone.sentinel-kext DeveloperID - 4AYE5J54KN

You should end up pushing 4 profiles:

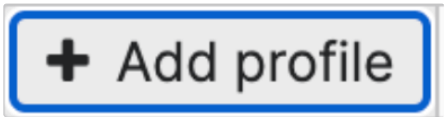
1. Network Filter Profile
2. Network Monitoring Profile
3. Notification Profile
4. Privacy Control Profile

Ventura and above also require a 5th profile, system manager.

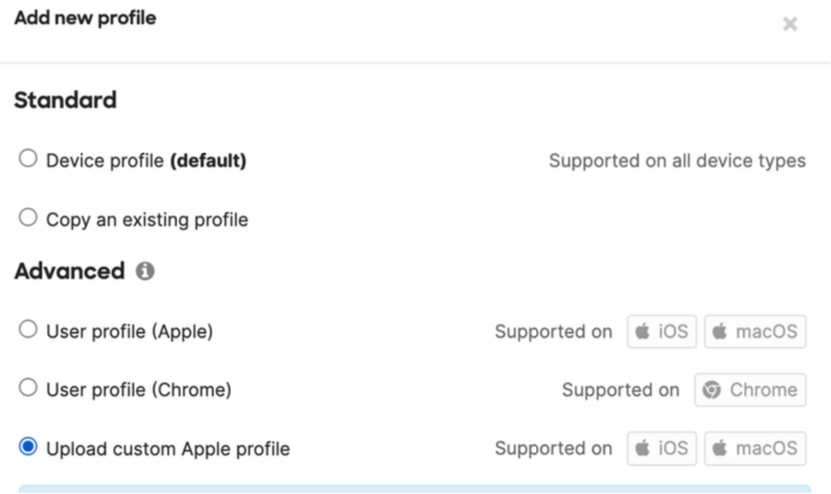
Create the profiles in meraki.
a. In meraki go to Systems Manager > Manage > Settings



c. Select Add Profile



e. Select “Upload custom Apple profile”


Add new profile ×

Standard

○ Device profile **(default)** — Supported on all device types

○ Copy an existing profile

Advanced

○ User profile (Apple) — Supported on iOS macOS

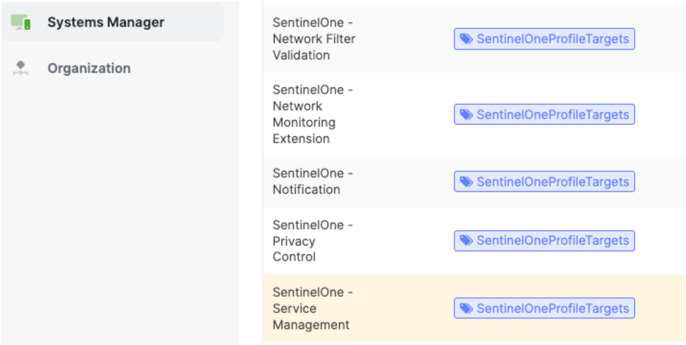
○ User profile (Chrome) — Supported on Chrome

● Upload custom Apple profile — Supported on iOS macOS

Upload each of the .mobileconfig files.

1. SentinelOne_-_Network_Filter_Validation__2_.mobileconfig
2. SentinelOne_-_Network_Monitoring_Extension__1_.mobileconfig
3. SentinelOne_-_Notification__1_.mobileconfig
4. SentinelOne_-_Privacy_Control_Agent_version_21.7_and_later__2_.mobileconfig

You should end up with something like this.



Systems Manager

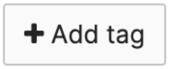
Organization

SentinelOne - Network Filter Validation	SentinelOneProfileTargets
SentinelOne - Network Monitoring Extension	SentinelOneProfileTargets
SentinelOne - Notification	SentinelOneProfileTargets
SentinelOne - Privacy Control	SentinelOneProfileTargets
SentinelOne - Service Management	SentinelOneProfileTargets

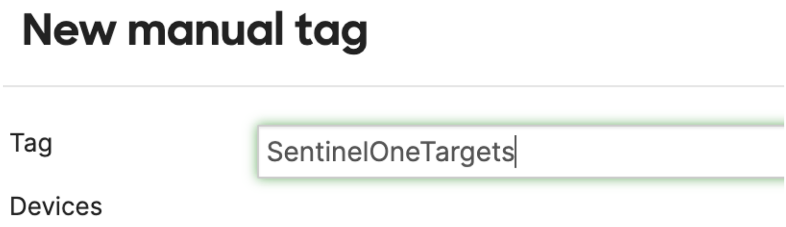
Create Devices Tags and Deployment Target Group In Meraki

We first need to create a new tag for target devices to associate them with the target group we are creating:

1. Login to the Meraki Dashboard and go to target groups
   a. System Manager > Tags > Add Tag



2. Set the name of the tag and select the desired devices.



New manual tag

Tag: SentinelOneTargets

Devices

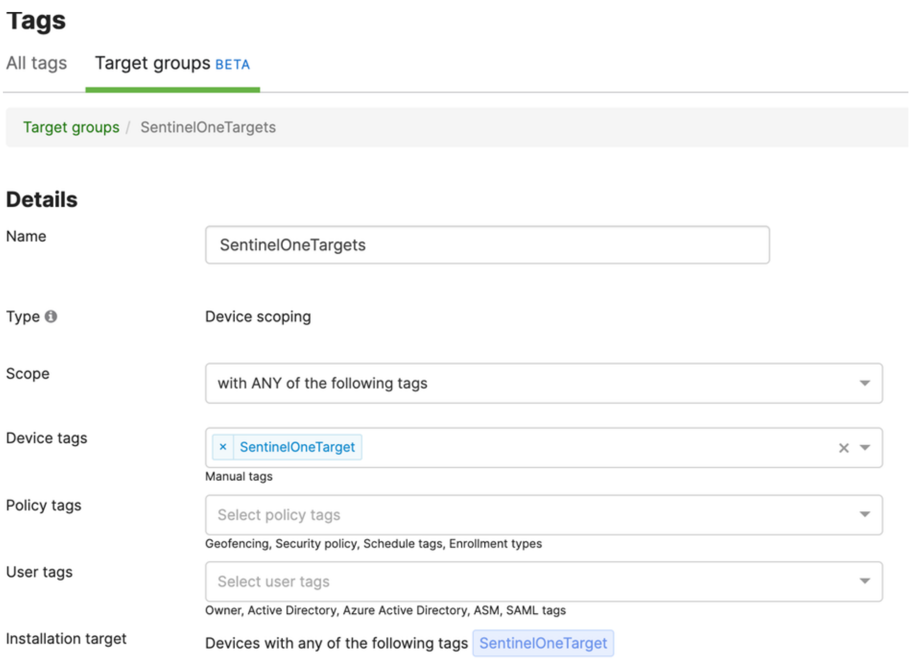
As of this writing, the target group feature in Meraki System Manager is still in Beta. This is also not the only way you can manage target device selection and deployment for profiles and the Sentinel agent. This process does however ensure the mobileconfig files are correctly installed prior to deploying the agent via SM. This allows for a mostly silent install; users will still be notified of new org managed installs on new versions of macOS.

For more information on Meraki Target Groups, see Merakis documentation.

Creating Target Groups:

1. Login to the Meraki Dashboard and go to target groups
   a. System Manager > Tags > TargetGroups
2. Add a new target group

3. Configure the new group to look like this:



Tags

All tags | Target groups BETA

Target groups / SentinelOneTargets

Details

Name: SentinelOneTargets

Type: Device scoping

Scope: with ANY of the following tags

Device tags: × SentinelOneTarget
Manual tags

Policy tags: Select policy tags
Geofencing, Security policy, Schedule tags, Enrollment types

User tags: Select user tags
Owner, Active Directory, Azure Active Directory, ASM, SAML tags

Installation target: Devices with any of the following tags SentinelOneTarget

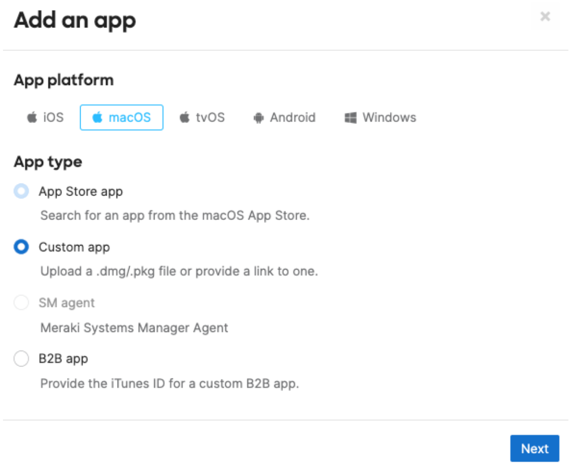
You can also automate this process with the script found here:

https://github.com/usmc0341/meraki_configurator/blob/main/README.md - Connect your Github account

Create SentinelOne Application In Meraki

Using the .dmg file created in previous steps:

1. From System Manager > Manage Apps > Add App.



Add an app

App platform

iOS | macOS | tvOS | Android | Windows

App type

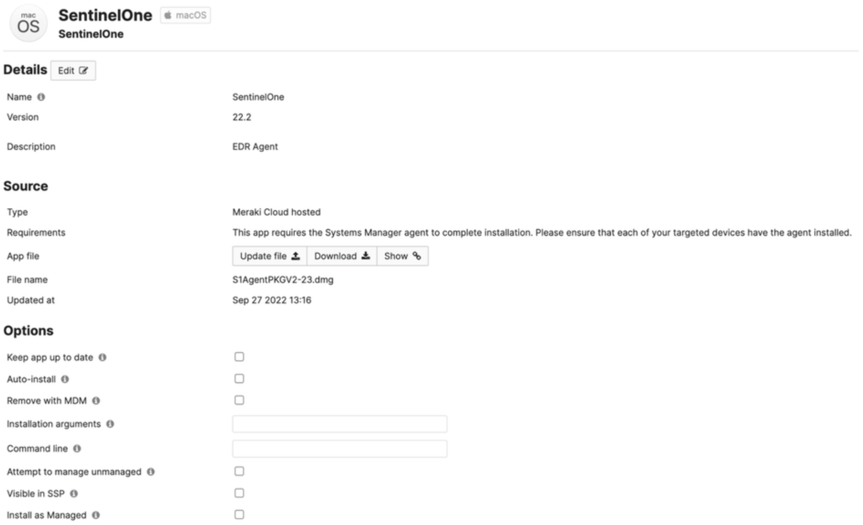
○ App Store app
Search for an app from the macOS App Store.

● Custom app
Upload a .dmg/.pkg file or provide a link to one.

○ SM agent
Meraki Systems Manager Agent

○ B2B app
Provide the iTunes ID for a custom B2B app.

2. Below is the recommended config



SentinelOne
SentinelOne

Details

Name	SentinelOne
Version	22.2
Description	EDR Agent

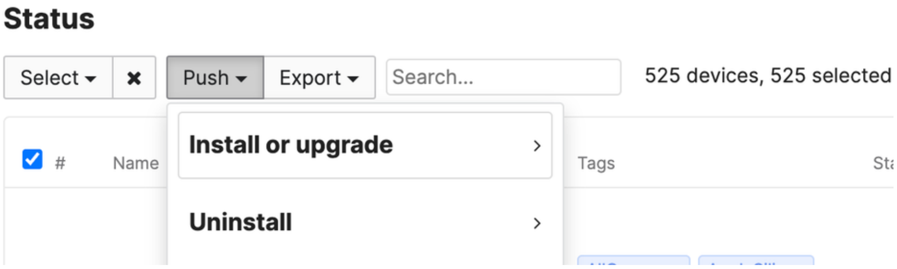
Source

Type	Meraki Cloud hosted
Requirements	This app requires the Systems Manager agent to complete installation. Please ensure that each of your targeted devices have the agent installed.
App file	Update file · Download · Show
File name	S1AgentPKGV2-23.dmg
Updated at	Sep 27 2022 13:16

Options

Keep app up to date	□
Auto-install	□
Remove with MDM	□
Installation arguments	
Command line	
Attempt to manage unmanaged	□
Visible in SSP	□
Install as Managed	□

The deploy group target should be set to the the target group created in the previous step. With the recommended configuration above Meraki will not auto install the app to target devices. This allows the adminstrator the ability to verify the mobileconfig profiles have deployed successfully to target devices. After verifying the install, select the devices in the target group and push the agent installer bundle.

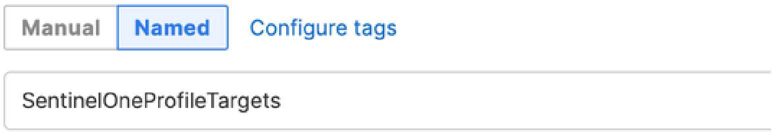


525 devices, 525 selected

It is recommended this process is combined with the verification process below. This allows deployment in batches by adding the profile verification tag as a search filter in the dashboard.

Push .mobileconfig To Target Devices

To start deployment of the mobileconfig files to target devices, return to the mobileconfig profiles created previously, and set the target to the newly created Target Group from the previous step. Meraki will being pushing the profiles to target devices.

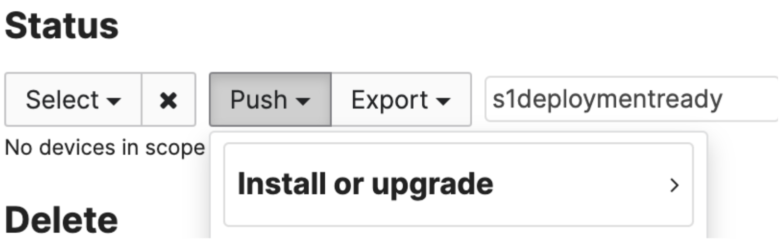


SentinelOneProfileTargets

Verifying Target Readiness and Agent Deployment Tagging

As a result of some limiting factors in the way Meraki deploys profiles, and the way they are associated in the management console, it can be difficult to verify all needed profiles have installed. This project has an additional module that checks for the profiles by querying the meraki api, and adding an “s1deploymentready” tag to target devices. You can then filter by this tag in the SentinelOne application in SystemManager for more manageable target selection.

https://github.com/usmc0341/meraki_configurator/blob/main/README.md - Connect your Github account



Status

s1deploymentready

No devices in scope

Delete