macOS Privacy Preferences Payload in Mojave 10.14+

Description

With macOS 10.14, Apple has introduced another new payload that requires User Approved MDM.

The Privacy Preferences payload controls the settings that are displayed in the ”Privacy” tab of the ”Security & Privacy” pane in System Preferences. This forms part of Apple's security framework: Transparency Consent and Control (TCC).

Ingredients

- FileWave 13+
- macOS 10.14+
- macOS UAMDM enrolled device

FileWave Supported Version

Although FileWave 13 has initial support for Privacy Payloads, approval for the FileWave Client to access services in macOS 10.15 relies upon FileWave 13.2.2 or higher.

BundleID or Path

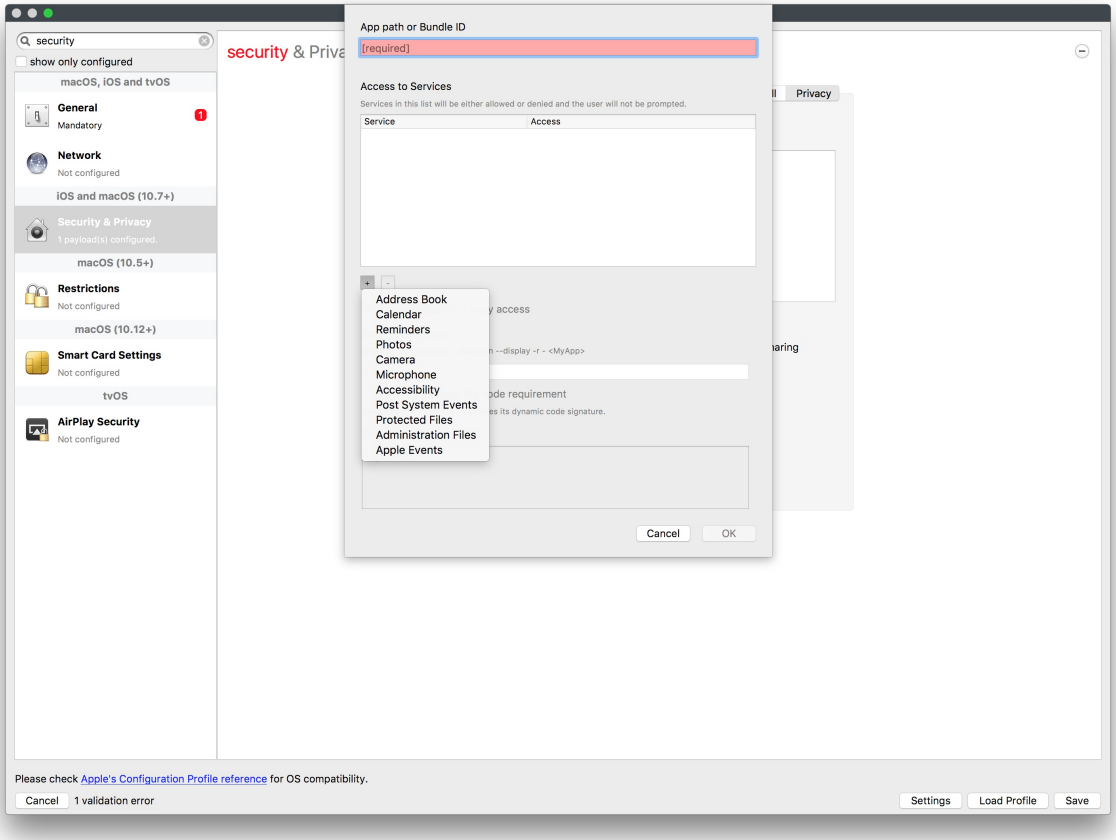
When building TCC Privacy Payloads there are two choices:

- BundleID
- Path

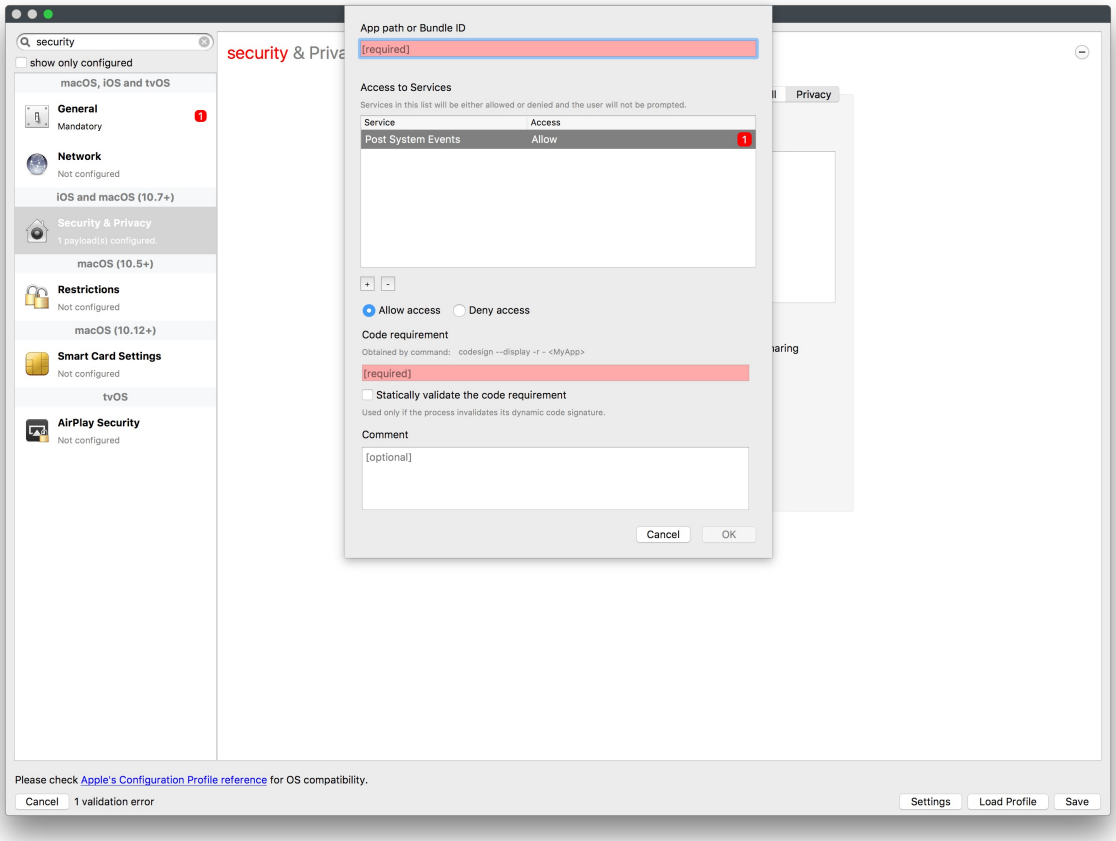
Items within bundles (.app or .bundle), must refer to the bundle and not the path. Otherwise, 'Path' should be selected.

Directions

1. Create a New Profile Fileset
2. Choose Payload
3. Select 'Privacy & Security' in the sidebar
4. Select the Privacy tab
5. Provide either the App Path or Bundle ID as required and click on the '+'



6. Choose the necessary Service type
7. Add the Code Requirement



Code Required To obtain the required code, as shown in the window the displayed 'codesign' command will need to be run, pointing to the path of either the binary or Application. See below: Requesting Application.

8. Set other settings appropriately.
9. Save, Associate, and Update Model

System Preferences

When a payload is set to configure a service on a macOS device, the System Preferences view will not reflect this setting, despite it being managed.

Identifying Accessing Service

Run a 'log stream' command on the destination device and then attempt to push the Fileset. Using the following command, the requesting App may be identified:

```
$ /usr/bin/log stream --debug --predicate 'subsystem == "com.apple.TCC" AND eventMessage BEGINSWITH
"AttributionChain"'
Filtering the log data using "subsystem == "com.apple.TCC" AND composedMessage BEGINSWITH "AttributionChain""
Timestamp                       Thread     Type        Activity             PID    TTL
2019-04-04 09:17:22.668813+0100 0x14c8f    Info        0x2a2c2              209    0    tccd:
[com.apple.TCC:access] AttributionChain: RESP:{ID: com.filewave.fwcld, PID[8318], auid: 0, euid: 0, responsible
path: '/usr/local/sbin/FileWave.app/Contents/MacOS/fwcld', binary path:
'/usr/local/sbin/FileWave.app/Contents/MacOS/fwcld'}, ACC:{ID: com.apple.ls, PID[8691], auid: 0, euid: 0, binary
path: '/bin/ls'}, REQ:{ID: com.apple.sandboxd, PID[7499], auid: 0, euid: 0, binary path: '/usr/libexec/sandboxd'}
```

This has shown both the bundle ID and the path to the requesting agent:

```
# log show --info -last 15m --predicate 'subsystem == "com.apple.TCC" AND eventMessage contains "service="'
Filtering the log data using "subsystem == "com.apple.TCC" AND composedMessage CONTAINS "service=""
Skipping debug messages, pass --debug to include.
Timestamp                       Thread     Type        Activity             PID    TTL
2019-04-04 09:17:22.665903+0100 0x14c8f    Info        0x2a2c2              209    0    tccd:
[com.apple.TCC:access] tccd[209](0): handling request from PID[7499](-1): {
        service="kTCCServiceSystemPolicyAllFiles"
        function="TCCAccessRequest"
        preflight=false
        target_token={pid:8691, auid:-1, euid:0}
```

```
        background_session=false
}
```

Profile Services

Apple's list of manageable services:

- Apple MDM Support Privacy Preferences Policy Control List
- Apple Developer Privacy Preferences Policy Control List

In FileWave 13 (up to 13.1.5), Protected Files refers to System Policy All Files.

Denied Only

Some items only have the option to be denied; allowance is only possible by the user. Examples include:

- Camera (macOS 10.14+)
- Microphone (macOS 10.14+)
- Screen Capture (macOS 10.15+)

macOS 10.5+

Default behaviour of some services is different between macOS 10.14 and 10.15+. The below example is a demonstration of this experience. Testing of each major version of macOS, from 10.14 up, is advised.

Example

Configure FileWave to use Apple's Screen Sharing Application and allow the Apple Screen Sharing Agent.

macOS 10.15+

As of macOS 10.15+ Screen Sharing has been included as a separate service, which may only be denied. Only Users can allow Screen Sharing. However, devices that are UAMDM enrolled, may have Apple's Screen Sharing service enabled and FileWave may be configured to use this service as suggested in the next section along with the following Privacy Payload.

It should be noted, that there are two options when logging in with 10.15+ if the user authenticating Screen Sharing differs from the user currently logged in:

- Log in as yourself (the user name that was used to authenticate the screen sharing)
- Ask for permissions to view the display (user must accept the prompt that will appear on their screen)

It is not possible to configure macOS 10.15+ with permission for Screen Sharing control of a users environment, without the users accepting the request, where a user is logged in that differs from the authenticated Screen Sharing user. However this method offers the option to prompt the user to accept the request; prompting will occur on every new Screen Sharing attempt.

This method with 10.15+ provides both Observation and Control.

Reconfigure FileWave for Apple's Screen Sharing Application

By default, FileWave has its own built-in Screen Sharing Agent. To adapt this, follow the details laid out in Apple's Screen Sharing Application

Prior to macOS 10.14+ this would allow full control of the macOS device. Due to the new TCC Privacy Framework, macOS 10.14+ will only allow observation when using this method and a privacy payload is required.

Privacy Payload to Allow Apple's Screen Sharing Agent

Apple have provided details to allow the Apple Screen Sharing Agent, which provides the necessary information to build the profile. If these details were not provided though, they may be obtained (see 'Requesting Application' below).

For simplicity, use this provided Fileset: Profile - TCC - Screen Sharing.fileset.zip

Deployment

Associate both the Fileset to configure FileWave to leverage Apple's Screen Sharing and the Fileset to allow the Apple's Screen Sharing Agent, Update Model and test client observation.

Requesting Application

Details of the requesting Application can be viewed if need be. On a test client device, run the following command in Terminal:

```
$ /usr/bin/log stream --debug --predicate 'subsystem == "com.apple.TCC" AND eventMessage BEGINSWITH
"AttributionChain"'
```

This will give a live output of events requesting access. Configure a device to use Apple's Screen Sharing App from above and attempt to run the required Application (in this case use 'Observe Client' from the Admin console) to the test client. The following should be observed on the client:

```
$ /usr/bin/log stream --debug --predicate 'subsystem == "com.apple.TCC" AND eventMessage BEGINSWITH
"AttributionChain"'

Filtering the log data using "subsystem == "com.apple.TCC" AND composedMessage BEGINSWITH "AttributionChain""

Timestamp                       Thread     Type        Activity             PID    TTL
2018-10-12 07:54:39.013089+0100 0xa1b1     Info        0x21eb3              209    0    tccd:
[com.apple.TCC:access] AttributionChain: ACC:{ID: com.apple.screensharing.agent, PID[1125], auid: 501, euid: 501,
binary path:
'/System/Library/CoreServices/RemoteManagement/ScreensharingAgent.bundle/Contents/MacOS/ScreensharingAgent'}, REQ:
{ID: com.apple.WindowServer, PID[176], auid: 88, euid: 88, binary path:
'/System/Library/PrivateFrameworks/SkyLight.framework/Versions/A/Resources/WindowServer'}
```

This has shown both the bundle ID and the path to the requesting agent:

- `com.apple.screensharing.agent`
- `/System/Library/CoreServices/RemoteManagement/ScreensharingAgent.bundle/Contents/MacOS/ScreensharingAgent`

Using the agent path, the Code Requirement can be retrieved.

```
$ codesign -dr -
/System/Library/CoreServices/RemoteManagement/ScreensharingAgent.bundle/Contents/MacOS/ScreensharingAgent

Executable=/System/Library/CoreServices/RemoteManagement/ScreensharingAgent.bundle/Contents/MacOS/ScreensharingAge
nt

designated => identifier "com.apple.screensharing.agent" and anchor apple
```

In this case, the Code Requirement is:

- `identifier "com.apple.screensharing.agent" and anchor apple`

These details may now be used to create a Privacy payload

Identify Accessing Service

The 'log show' command may be used to observe actions that have previously occurred. In this example

```
# log show --info -last 15m --predicate 'subsystem == "com.apple.TCC" AND eventMessage contains "service="'
Filtering the log data using "subsystem == "com.apple.TCC" AND composedMessage CONTAINS "service=""
Skipping debug messages, pass --debug to include.
Timestamp                       Thread     Type        Activity             PID    TTL
2019-04-04 09:17:22.665903+0100 0x14c8f    Info        0x2a2c2              209    0    tccd:
[com.apple.TCC:access] tccd[209](0): handling request from PID[7499](-1): {
        service="kTCCServiceSystemPolicyAllFiles"
        function="TCCAccessRequest"
        preflight=false
        target_token={pid:8691, auid:-1, euid:0}
        background_session=false
}
```

FileWave Client

Providing access to services to the FileWave Client requires:

- App path or Bundle ID: com.filewave.client
- Code Requirement: identifier "com.filewave.fwcld.pkg" and anchor apple generic and certificate 1[field.1.2.840.113635.100.6.2.6] /* exists */ and certificate leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and certificate leaf[subject.OU] = "83S2TRZ3CS"

Revision #2
★Created 15 July 2023 00:34:36 by Josh Levitsky
✎Updated 15 July 2023 00:56:51 by Josh Levitsky