

MDM Hygiene for MacAdmins



How safe is your Mac fleet?

MacDevOpsYVR 2025

Mykola Grymalyuk - June 13th, 2025

Bill of Materials



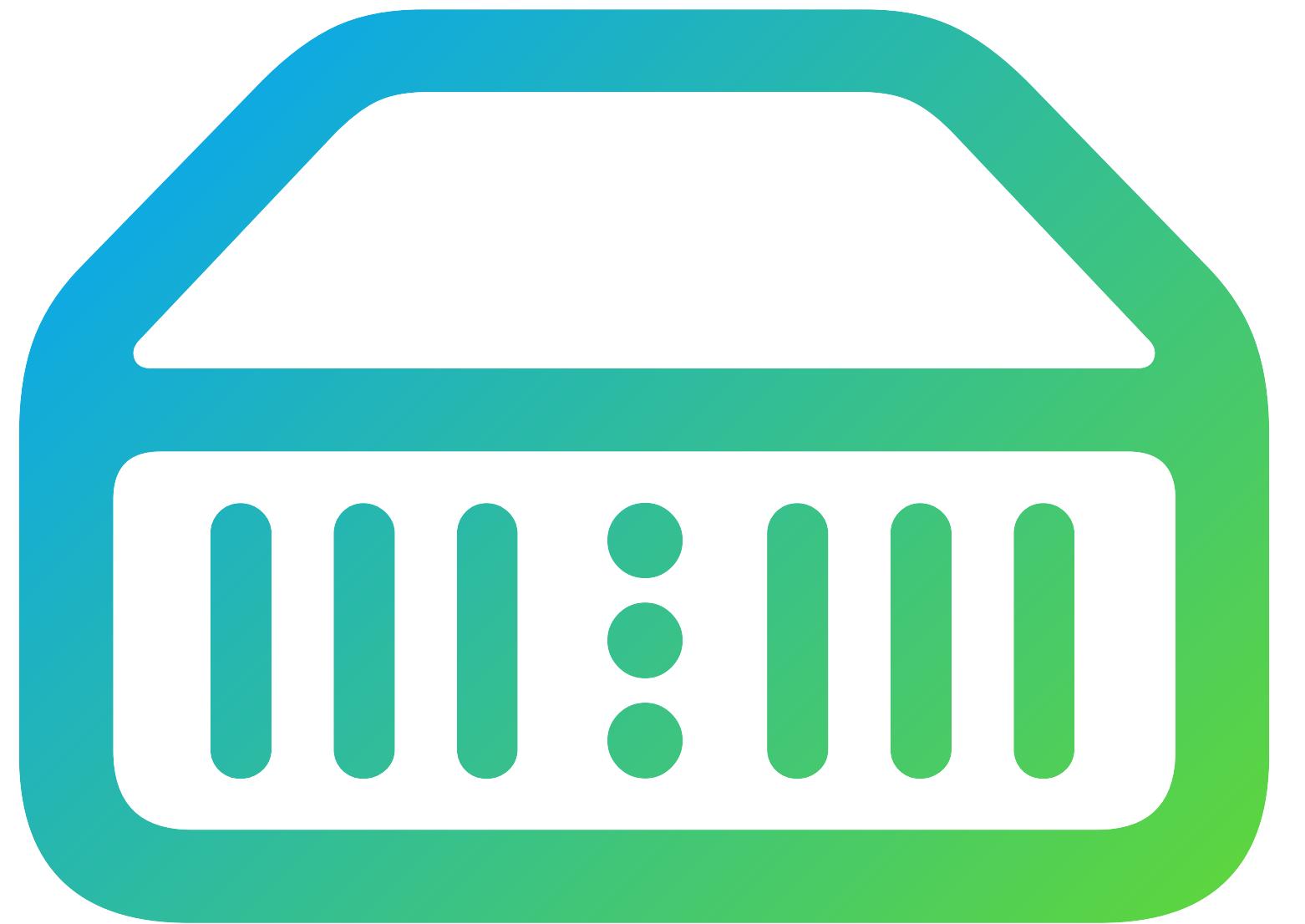
Bill of Materials

- Intro to MDM.



Bill of Materials

- Intro to MDM.
- Importance of security.



Bill of Materials

- Intro to MDM.
- Importance of security.
- Common Security Pitfalls.



Bill of Materials

- Intro to MDM.
- Importance of security.
- Common Security Pitfalls.
- Poke MDM vendors.



Bill of Materials

- Intro to MDM.
- Importance of security.
- Common Security Pitfalls.
- Poke MDM vendors.
- Future research for the audience.



\$ '/usr/bin/whoami'

> "Mykola Grymalyuk"

- Lead Security and Software Engineer at RIPEDA Consulting.
- Project lead of OpenCore Legacy Patcher.
- Breaking macOS internals on khronokernel.com.



**With great power comes great
responsibility**

**With great power comes great
responsibility**

**And with an MDM server, a ton of ways
to break a fleet...**

What is an MDM?

What is an MDM?

Mobile Device Management

What is an MDM?

Mobile Device Management

What is an MDM?

Mobile Device Management



**Centralized Device
Management**

What is an MDM?

Mobile Device Management



**Centralized Device
Management**



**Managed by 3rd
party software ***

What is an MDM?

Mobile Device Management



**Centralized Device
Management**



**Managed by 3rd
party software ***



**Privileged
access**

What is an MDM?

Mobile Device Management



**Centralized Device
Management**



**Managed by 3rd
party software ***



**Privileged
access**

([InstallProfileCommand](#))

Why is security important?

Why is security important?

Why is security important?

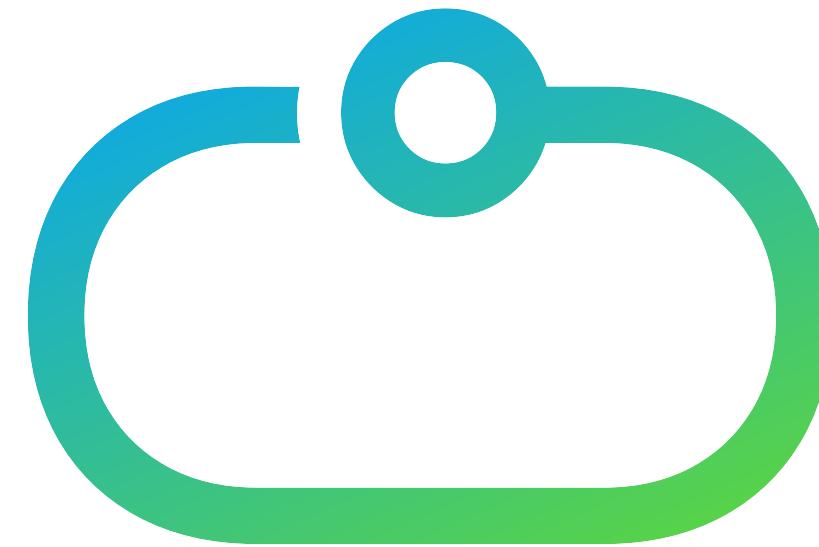


Targeted Attacks

Why is security important?



Targeted Attacks



Automated Attacks

Why is security important?

“We detected a 101% increase of macOS info stealers between the last two quarters of 2024.”

- Unit 42 (Palo Alto Networks)

Targeted Attacks

Automated Attacks

Common MacAdmin Security Pitfalls

Common MacAdmin Security Pitfalls

Common MacAdmin Security Pitfalls



Profiles

Common MacAdmin Security Pitfalls



Profiles



Scripts

Common MacAdmin Security Pitfalls



Profiles



Scripts



Software catalogs

Common MacAdmin Security Pitfalls



Profiles



Scripts



Software catalogs



Enrollments





Profiles



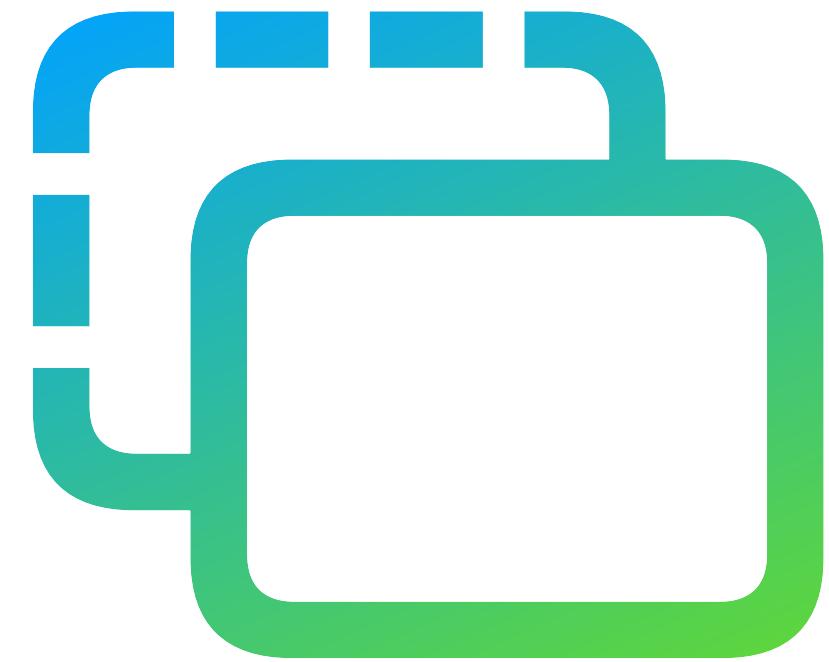
Full Disk Access



Profiles



Full Disk Access



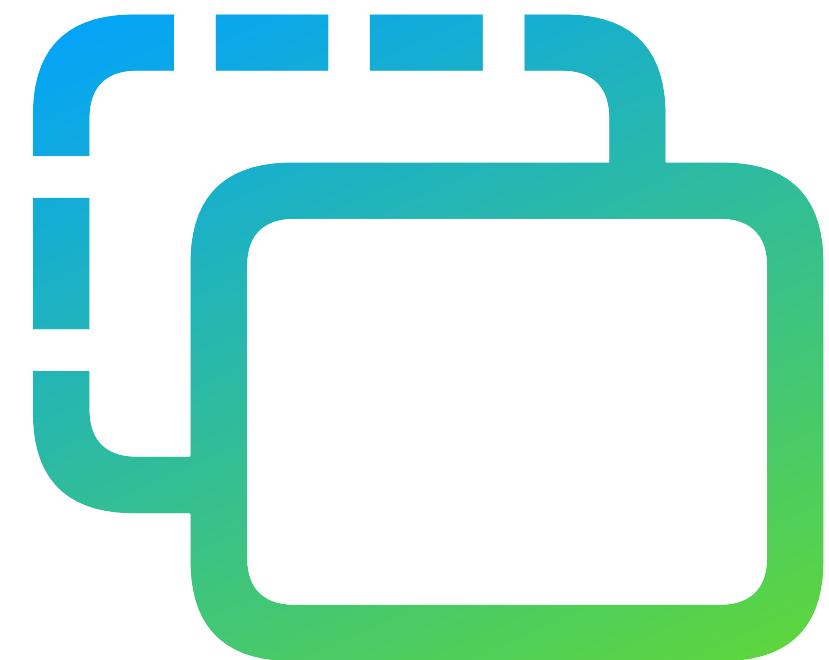
Background Tasks



Profiles



Full Disk Access



Background Tasks



Gatekeeper



Full Disk Access

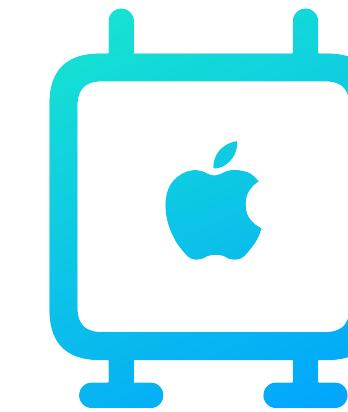


Full Disk Access





Full Disk Access



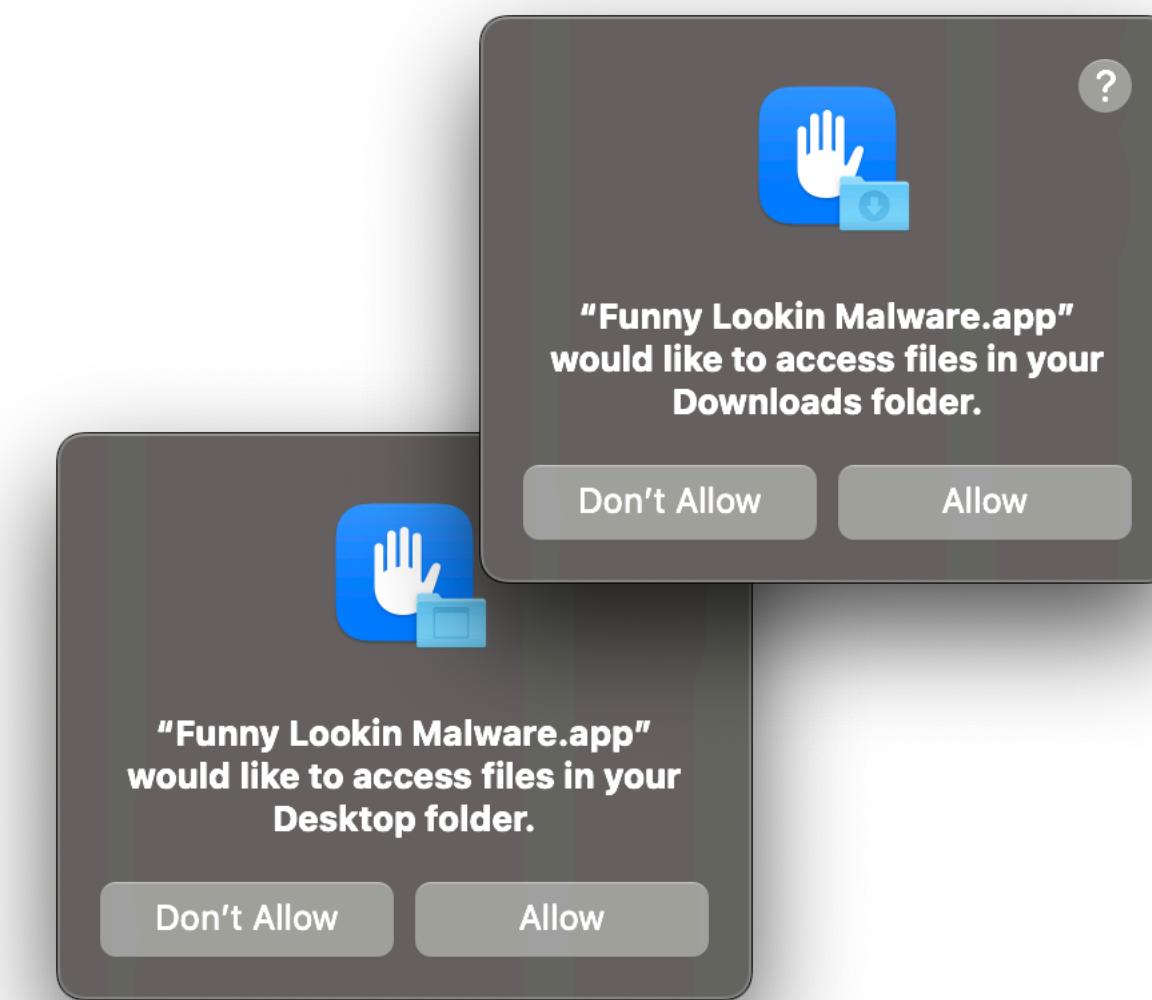


Full Disk Access



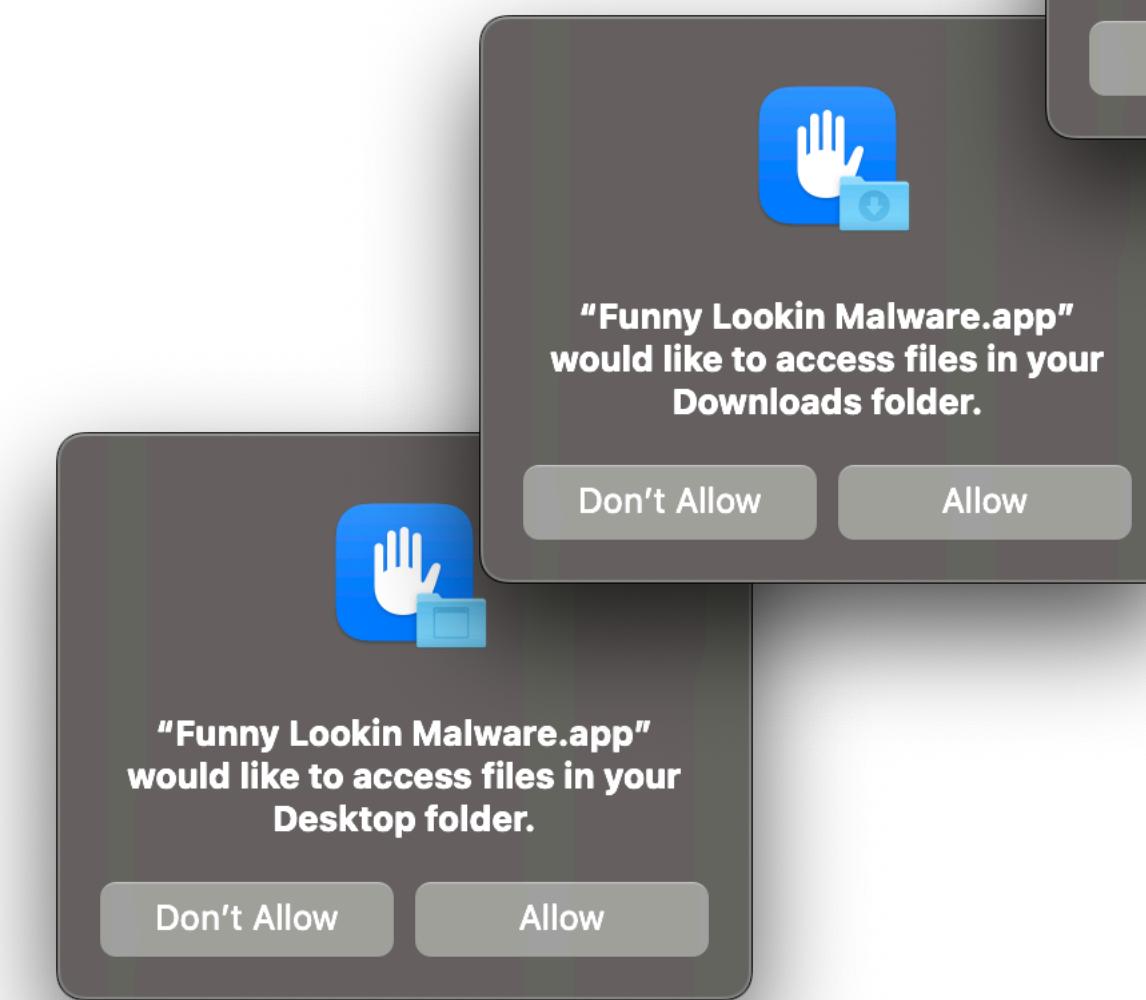


Full Disk Access



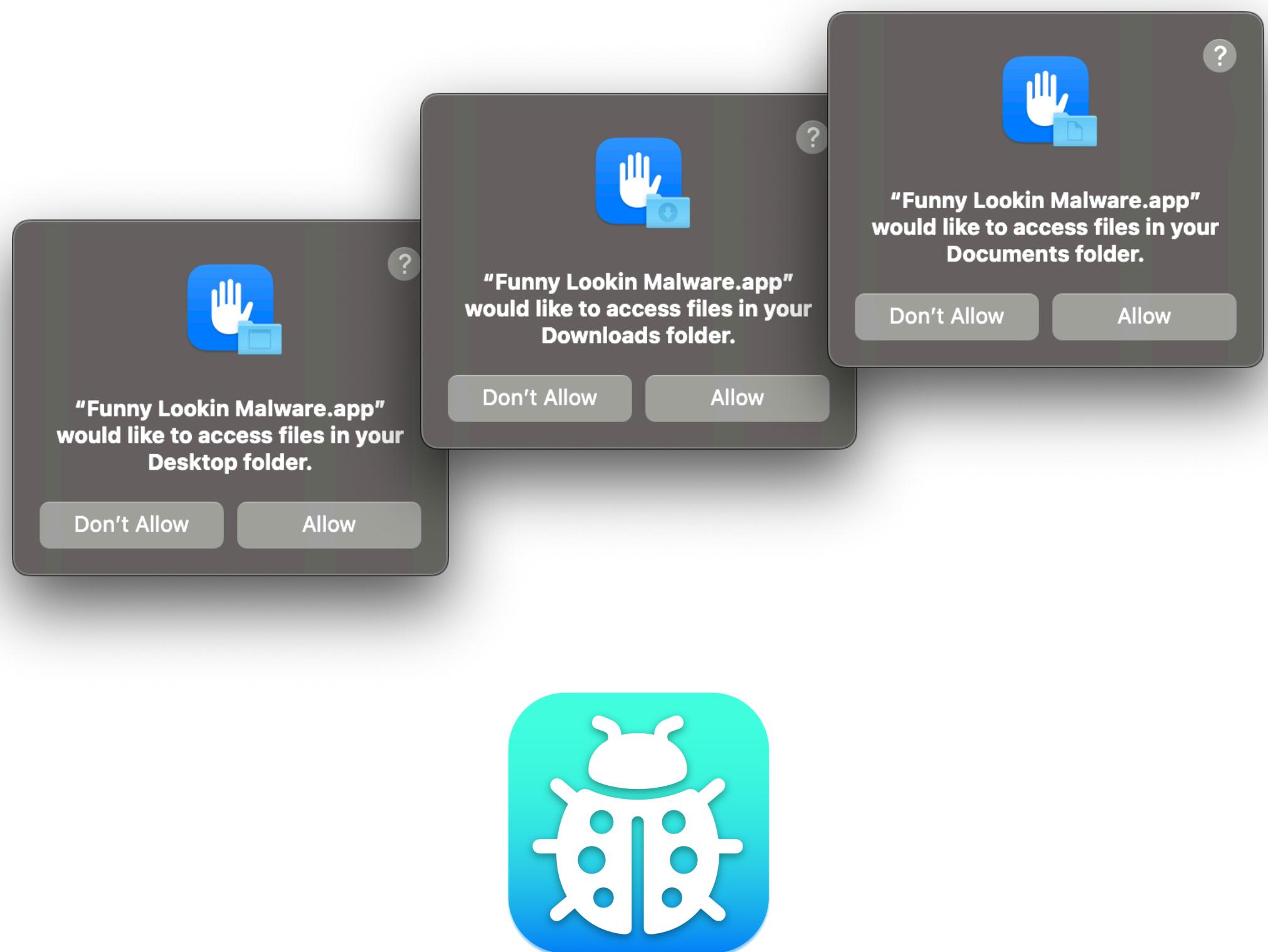


Full Disk Access





Full Disk Access

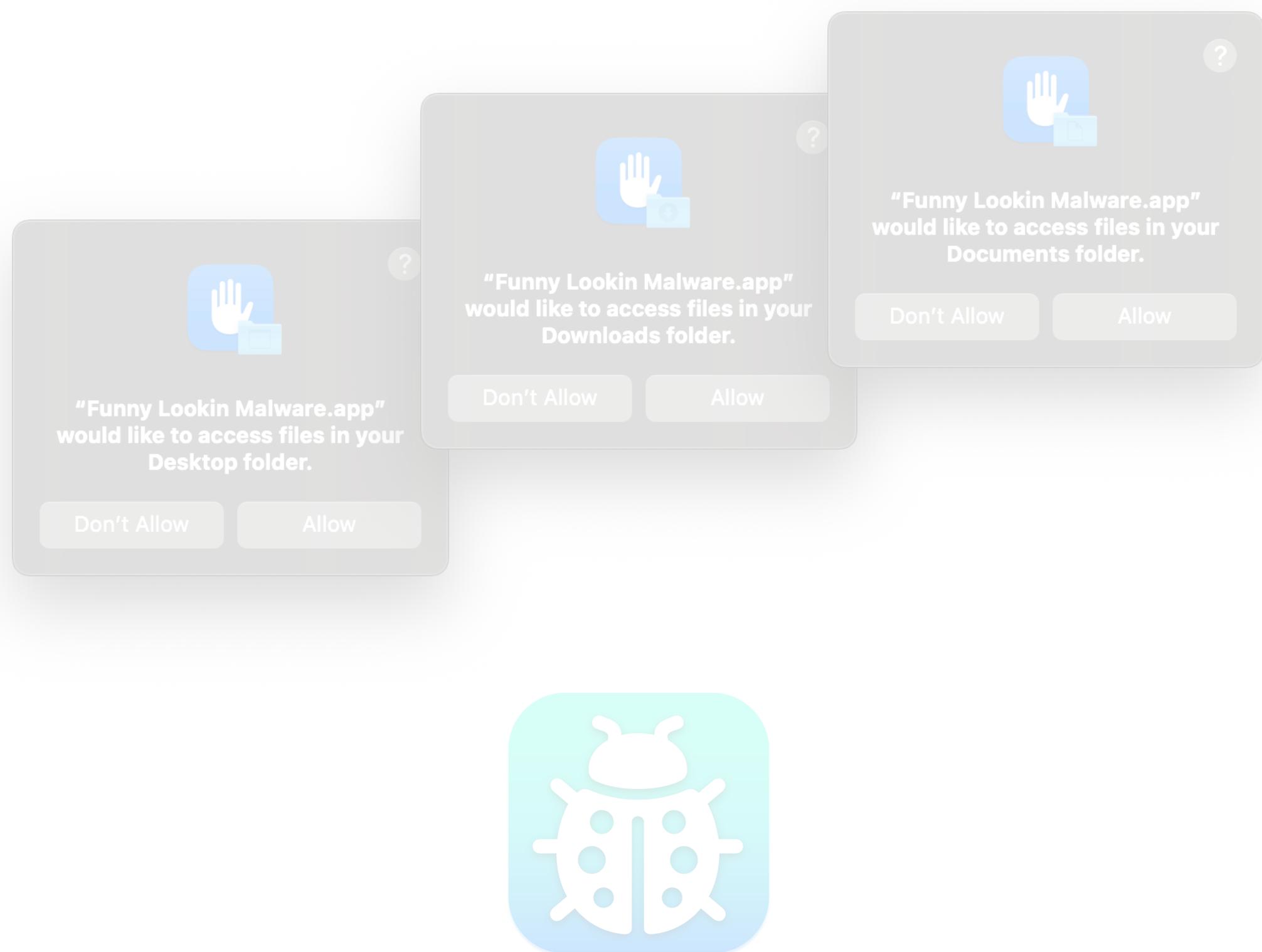


```
com.apple.TCC.configuration-profile-policy
```

```
"Services": {  
    "SystemPolicyAllFiles": [  
        {  
            "Allowed": true,  
            "CodeRequirement": "...",  
            "Identifier": "...",  
            "IdentifierType": "...",  
            "StaticCode": false  
        }  
    ]  
}
```



Full Disk Access



com.apple.TCC.configuration-profile-policy

```
"Services": {  
    "SystemPolicyAllFiles": [  
        {  
            "Allowed": true,  
            "CodeRequirement": "...",  
            "Identifier": "...",  
            "IdentifierType": "...",  
            "StaticCode": false  
        }  
    ]  
}
```



Full Disk Access



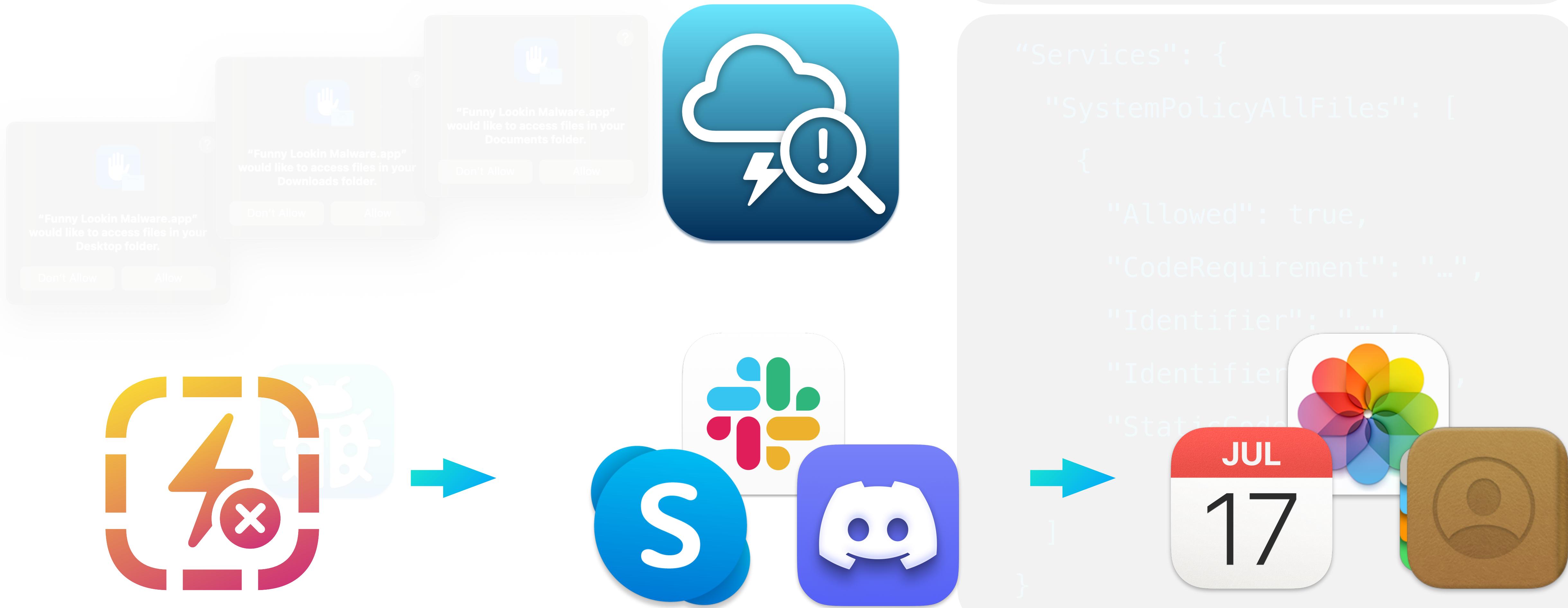
**Vulnerable apps with
Arbitrary Code Execution**



```
com.apple.TCC.configuration-profile-policy

{
    "Services": {
        "SystemPolicyAllFiles": [
            {
                "Allowed": true,
                "CodeReviewComment": "...",
                "Identifier": "...",
                "IdentifierType": "...",
                "StaticCode": false
            }
        ]
    }
}
```

Vulnerable apps with Arbitrary Code Execution



Vulnerable apps with Arbitrary Code Execution



com.apple.TCC.configuration-policy



```
"Services": {  
    "SystemPolicyAllFiles": [  
        {  
            "Allowed": true,  
            "Path": "/bin/zsh & /bin/bash",  
            "Identifier": "...",  
            "IdentifierType": "...",  
            "StaticCode": false  
        }  
    ]  
}
```

Vulnerable apps with Arbitrary Code Execution



com.apple.TCC.configuration-profile-policy



"Services": {

"SystemPolicyAllFiles": [

/bin/zsh & /bin/bash

"CodeRequirement": "",

"IdentifierType": "",

"StaticCode": false

}

]

}



Full Disk Access

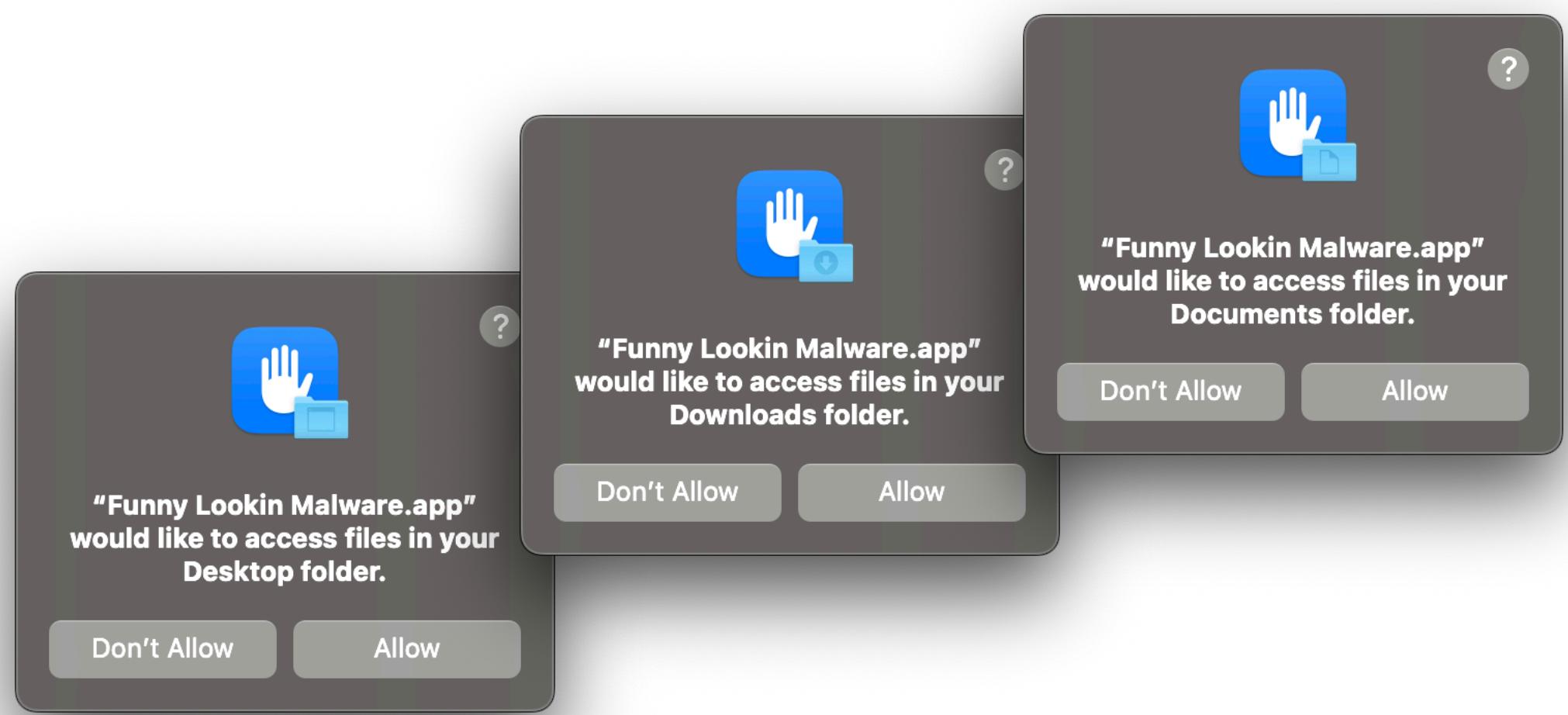


```
com.apple.TCC.configuration-profile-policy
```

```
"Services": {  
    "SystemPolicyAllFiles": [  
        {  
            "Allowed": true,  
            "CodeRequirement": "...",  
            "Identifier": "...",  
            "IdentifierType": "...",  
            "StaticCode": false  
        }  
    ]  
}
```



Full Disk Access



```
com.apple.TCC.configuration-profile-policy
```

```
"Services": {  
    "SystemPolicyDesktopFolder": [  
        ...  
    ],  
    "SystemPolicyDownloadsFolder": [  
        ...  
    ],  
    "SystemPolicyDocumentsFolder": [  
        ...  
    ],  
}
```



Full Disk Access

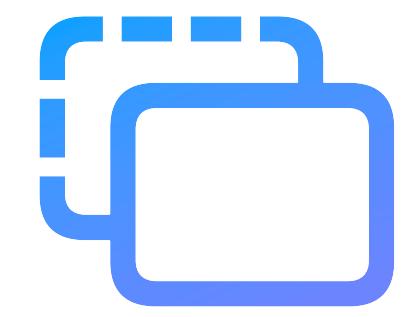


com.apple.TCC.configuration-profile-policy

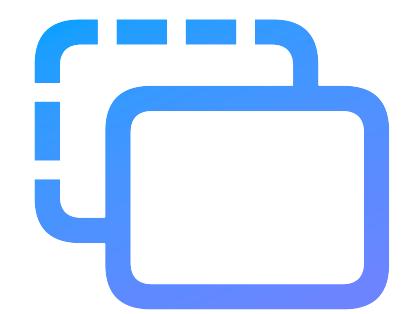
```
"Services": {  
    "SystemPolicyDesktopFolder": [  
        ...  
    ],  
    "SystemPolicyDownloadsFolder": [  
        ...  
    ],  
    "SystemPolicyDocumentsFolder": [  
        ...  
    ]  
}
```

**Do you need this?
(general apps)**

**And if so, could you
scope accordingly?
(shells & tooling)**

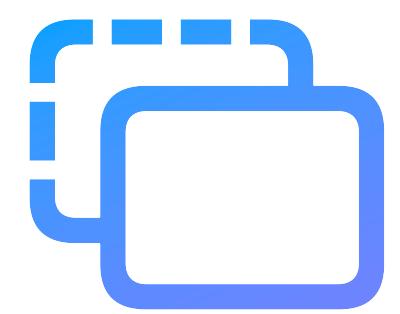


Background Tasks

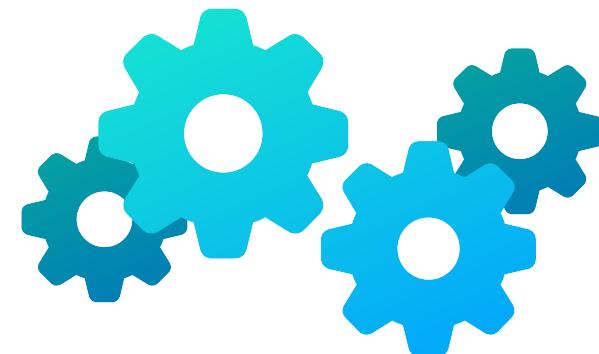


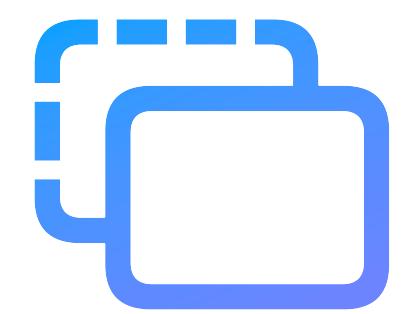
Background Tasks



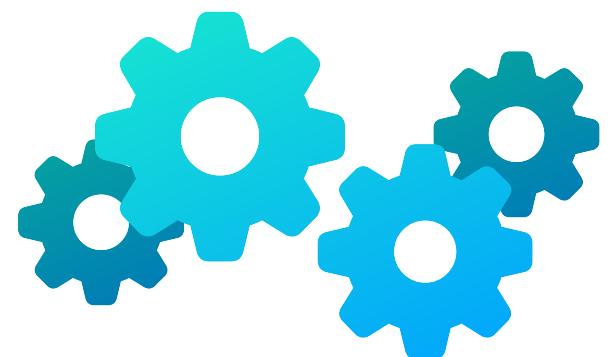


Background Tasks

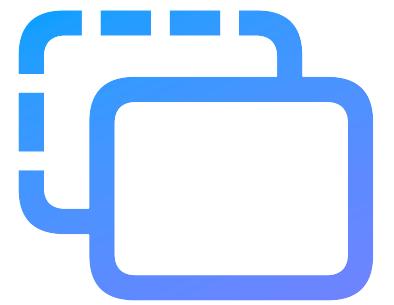




Background Tasks



-  **Background Items Added**
"Funny Lookin Malware" is an item that can run in the background. You can manage this in Login Items & Extensions.
-  **Background Items Added**
"Funny Lookin Malware" is an item that can run in the background. You can manage this in Login Items & Extensions.
-  **Background Items Added**
"Funny Lookin Malware" is an item that can run in the background. You can manage this in Login Items & Extensions.



Background Tasks

com.apple.servicemanagement

"Rules": [

{

 "Comment": "Funny Malware",

 "RuleType": "LabelPrefix",

 "RuleValue": "com.malware",

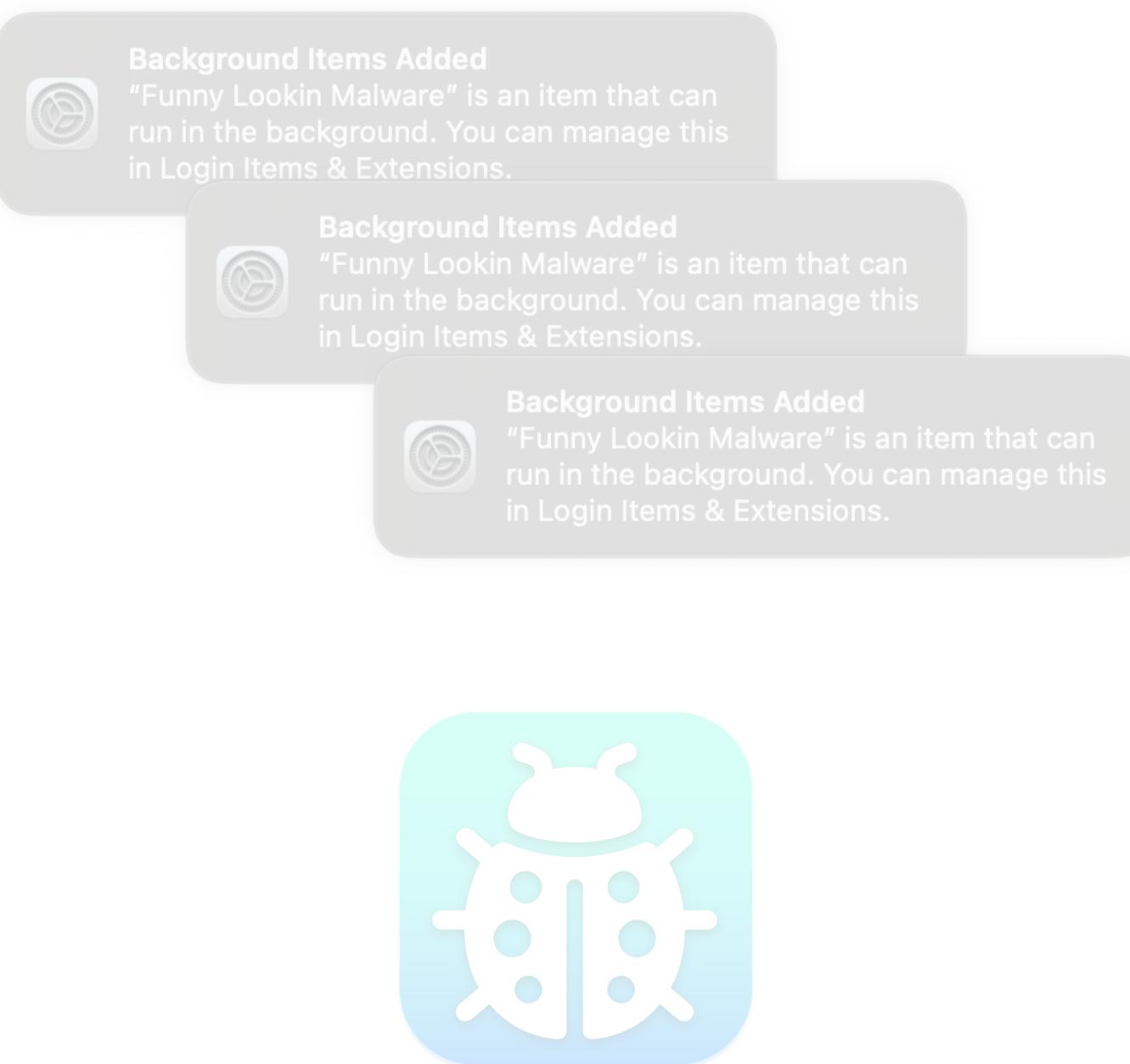
}

]



Managed Login Items Added
Your organization added items that can run in the background. You can view these in Login Items & Extensions.

Background Tasks



com.apple.servicemanagement

```
"Rules": [  
  {  
    "Comment": "Funny Malware",  
    "RuleType": "LabelPrefix",  
    "RuleValue": "com.malware",  
  }  
]
```



Managed Login Items Added
Your organization added items that can run in the background. You can view these in Login Items & Extensions.

Background Tasks

com.apple.servicemanagement

"Rules": [

{

 "Comment": "Funny Malware",

 "RuleType": "LabelPrefix",

 "RuleValue": "com.malware",

}



Spoofing identifiers

Managed Login Items Added



Your organization added items that can run in the background. You can view these in Login Items & Extensions.

Background Tasks

com.apple.servicemanagement

"Rules": [

{

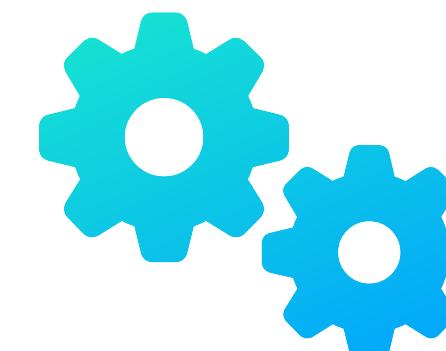
 "Comment": "Funny Malware",

 "RuleType": "LabelPrefix",

 "RuleValue": "com.malware",

}

Spoofing identifiers



Managed Login Items Added

Your organization added items that can run in the background. You can view these in Login Items & Extensions.

Background Tasks

com.apple.servicemanagement

"Rules": [

{

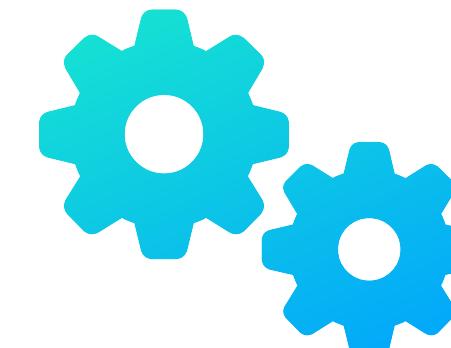
 "Comment": "Funny Malware",

 "RuleType": "LabelPrefix",

 "RuleValue": "com.malware",

}

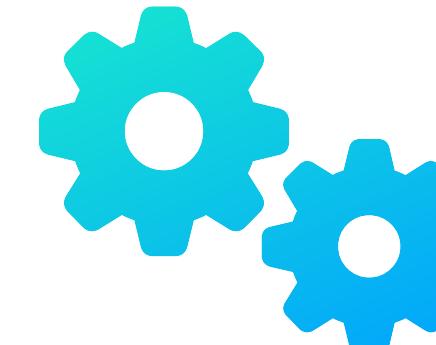
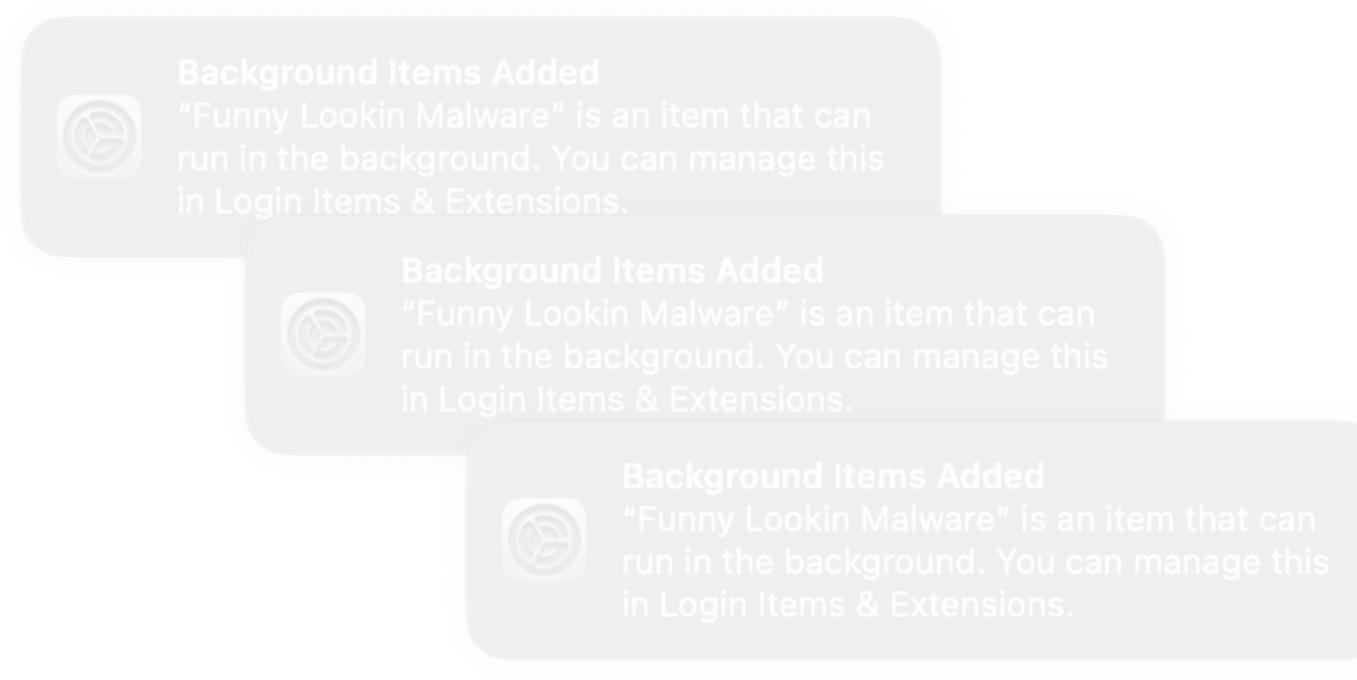
Spoofing identifiers



Managed Login Items Added

Your organization added items that can run in the background. You can view these in Login Items & Extensions.

Spoofing identifiers



com.apple.servicemanagement

```
"Rules": [  
  {  
    "Comment": "Funny Malware",  
    "RuleType": "LabelPrefix",  
    "RuleValue": "com.malware",  
  }  
]
```

Managed Login Items Added
Your organization added items that can run in the background. You can view these in Login Items & Extensions.

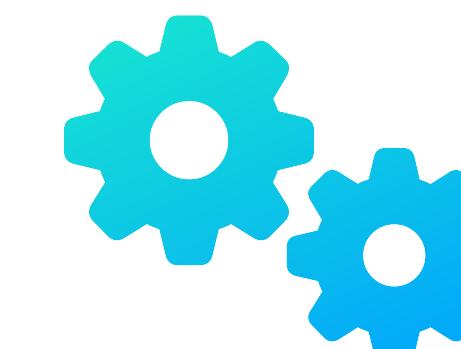
Spoofing identifiers

com.apple.servicemanagement

RuleType Possible Values:

- BundleIdentifier
- BundleIdentifierPrefix
- Label
- LabelPrefix
- TeamIdentifier

```
"Rules": [  
  {  
    "Comment": "Funny Malware",  
    "RuleType": "LabelPrefix",  
    "RuleValue": "com.malware",  
  }  
]
```



Managed Login Items Added
Your organization added items that can run in the background. You can view these in Login Items & Extensions.

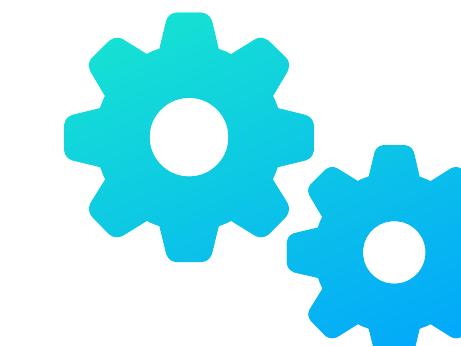
Spoofing identifiers

RuleType Possible Values:

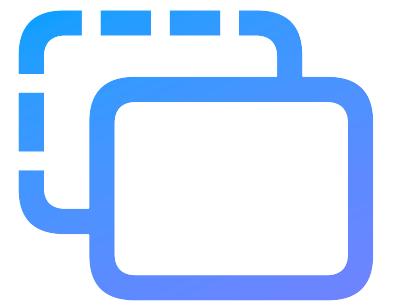
- BundleIdentifier
- BundleIdentifierPrefix
- Label
- LabelPrefix
- TeamIdentifier

com.apple.servicemanagement

```
"Rules": [  
  {  
    "Comment": "Funny Malware",  
    "RuleType": "LabelPrefix",  
    "RuleValue": "com.malware",  
  }  
]
```



Managed Login Items Added
Your organization added items that can run in the background. You can view these in Login Items & Extensions.



Background Tasks

com.apple.servicemanagement

"Rules": [

{

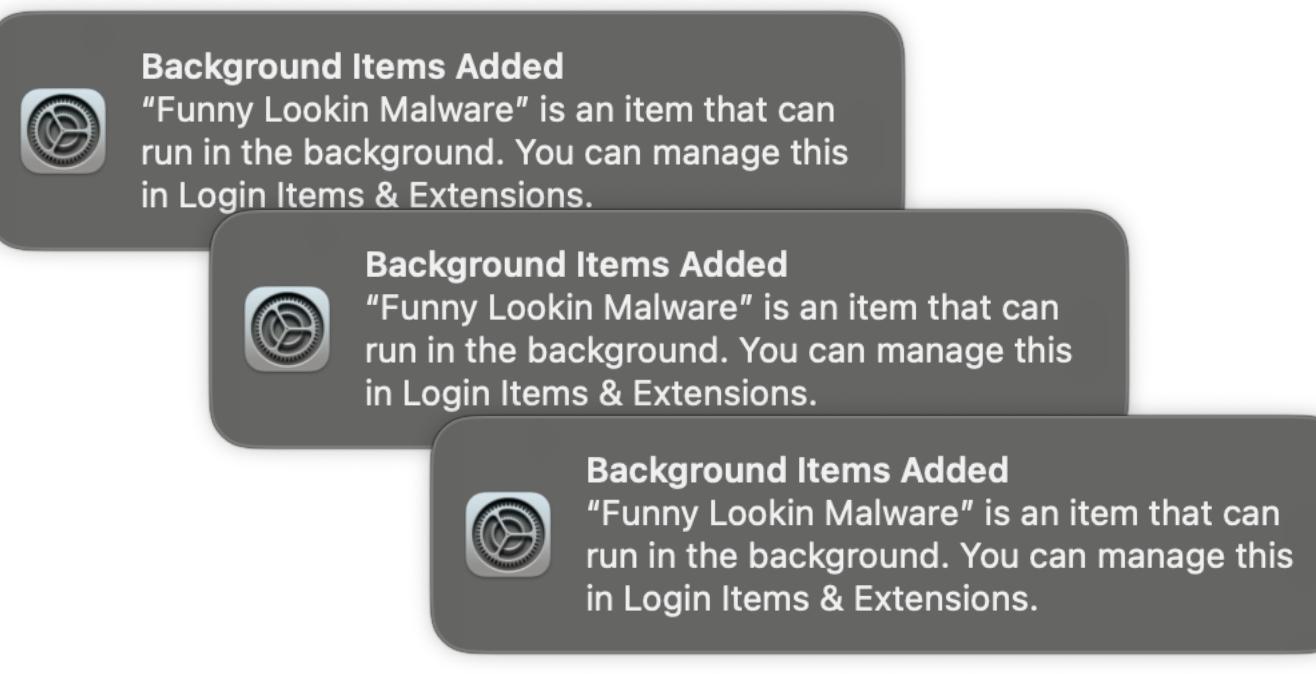
 "Comment": "Funny Malware",

 "RuleType": "LabelPrefix",

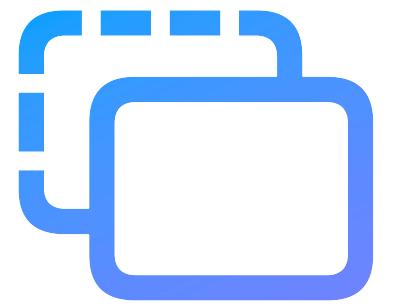
 "RuleValue": "com.malware",

}

]



Managed Login Items Added
Your organization added items that can run in the background. You can view these in Login Items & Extensions.



Background Tasks

com.apple.servicemanagement

"Rules": [

{

 "Comment": "Funny Malware",

 "RuleType": "TeamIdentifier",

 "RuleValue": "S74BDJXQMD",

}

]



Managed Login Items Added

Your organization added items that can run in the background. You can view these in Login Items & Extensions.



Background Tasks

Arbitrary Code Execution

com.apple.servicemanagement



Rules : [

{

 "Comment": "Funny Malware",

 "RuleType": "TeamIdentifier",

 "RuleValue": "S74BDJXQMD",

}

]



- Background Items Added
"Funny Lookin Malware" is an item that can run in the background. You can manage this in Login Items & Extensions.
- Background Items Added
"Funny Lookin Malware" is an item that can run in the background. You can manage this in Login Items & Extensions.
- Background Items Added
"Funny Lookin Malware" is an item that can run in the background. You can manage this in Login Items & Extensions.

Managed Login Items Added

Your organization added items that can run in the background. You can view these in Login Items & Extensions.



Gatekeeper



Gatekeeper





Gatekeeper





Gatekeeper

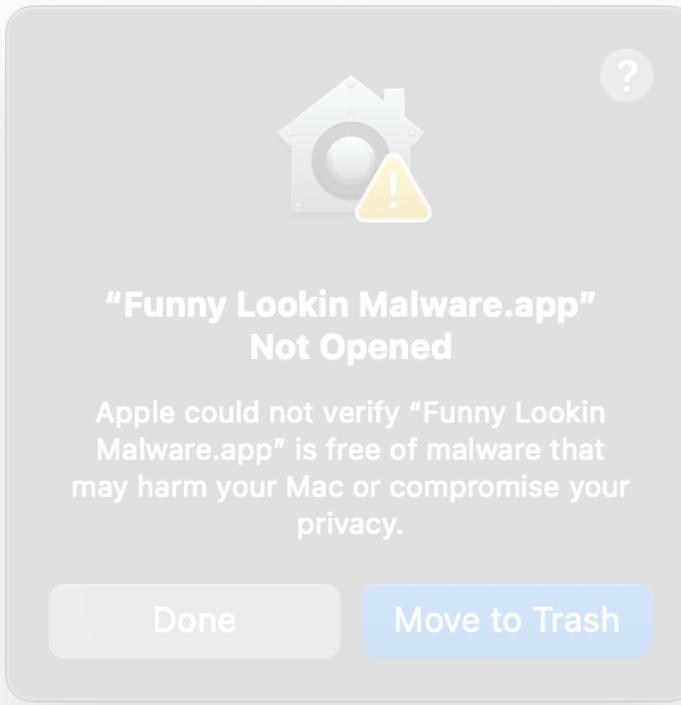


`com.apple.systempolicy.control`

`"EnableAssessment": False`



Gatekeeper

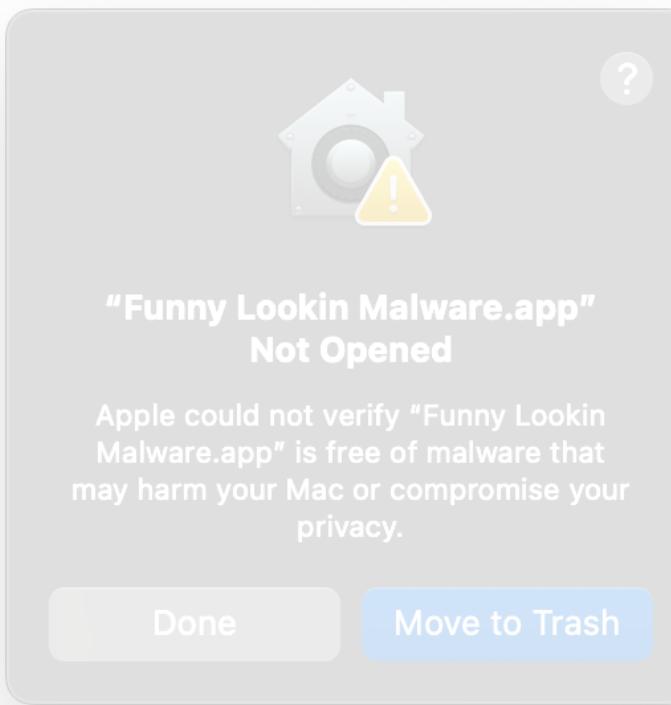


`com.apple.systempolicy.control`

`"EnableAssessment": False`



A LOT OF THINGS

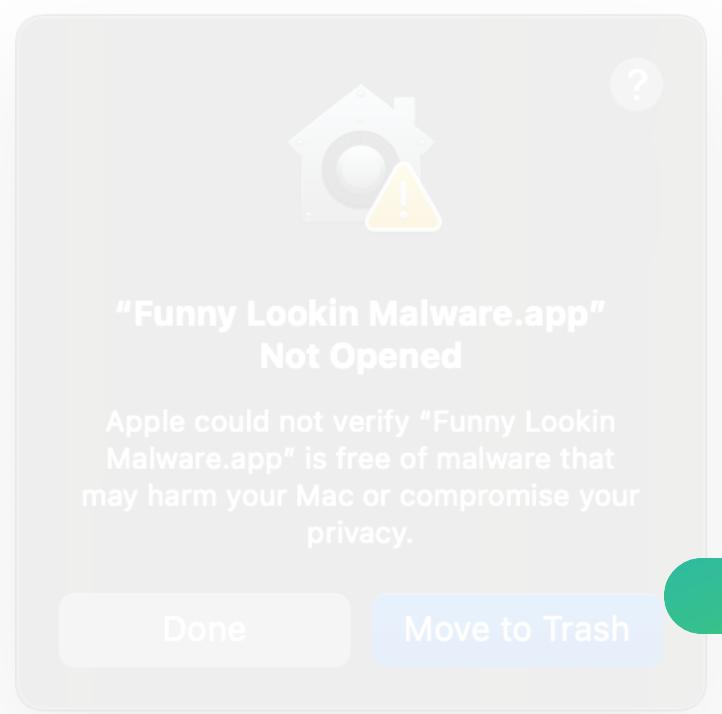


com.apple.systempolicy.control

"EnableAssessment": False



A LOT OF THINGS



com.apple.systempolicy.control

"EnableAssessment": False



Gatekeeper

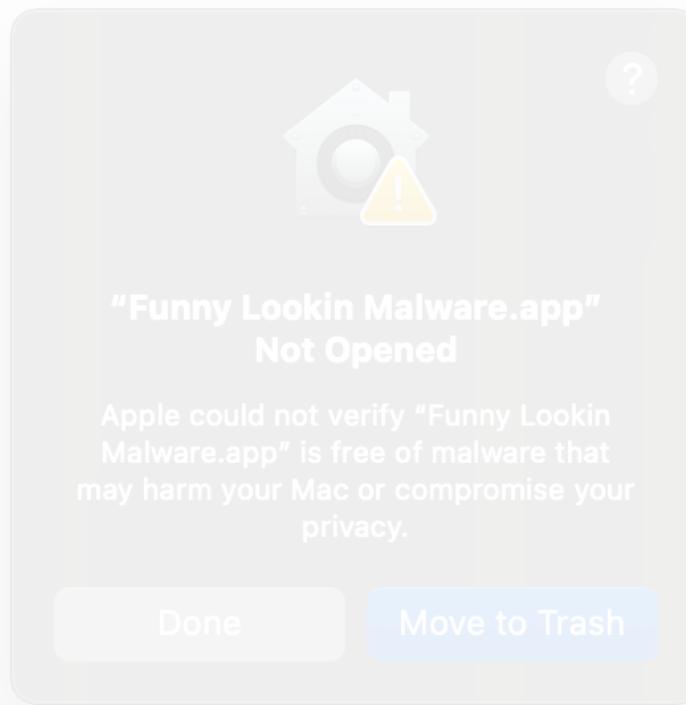


`com.apple.systempolicy.control`

`"EnableAssessment": False`



Gatekeeper

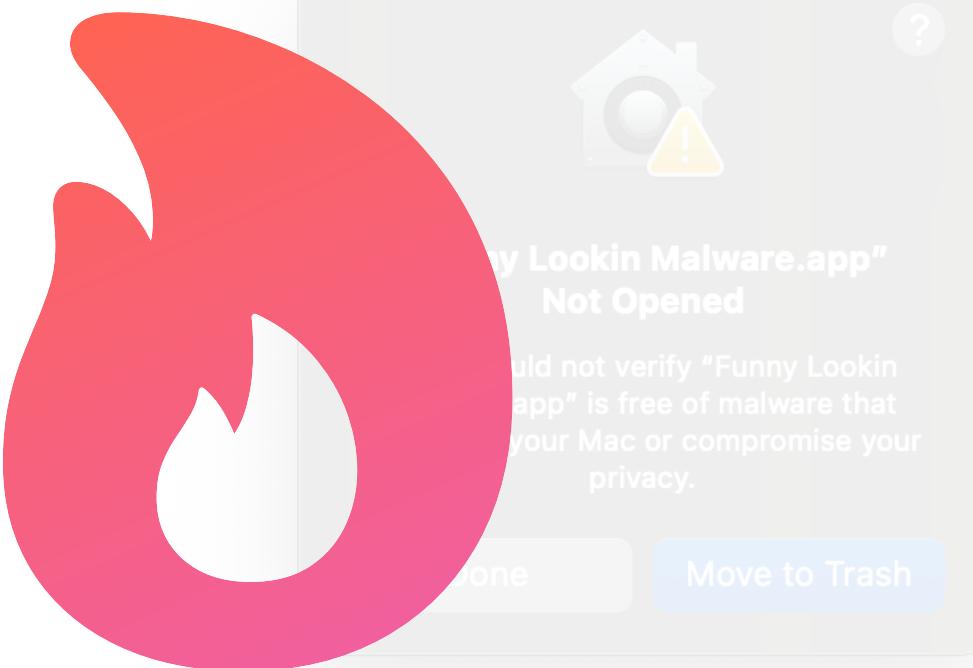


com.apple.systempolicy.control

"EnableAssessment": False



Gatekeeper

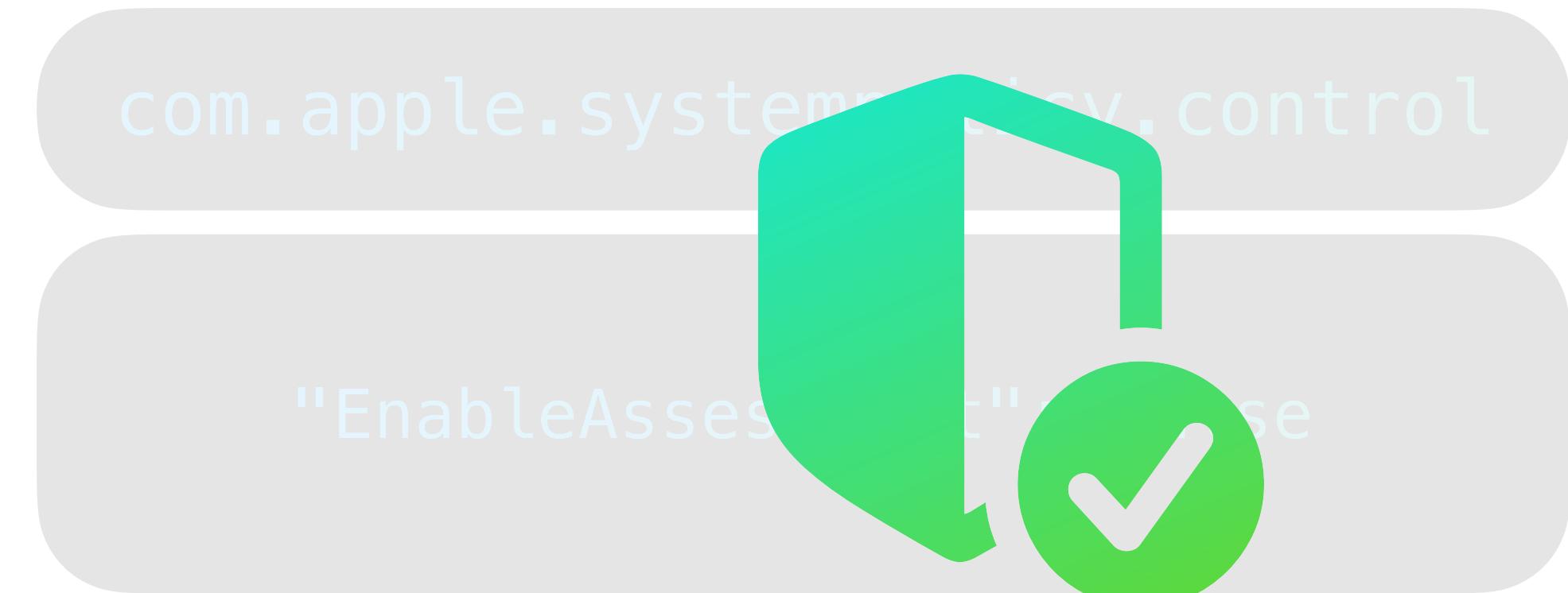
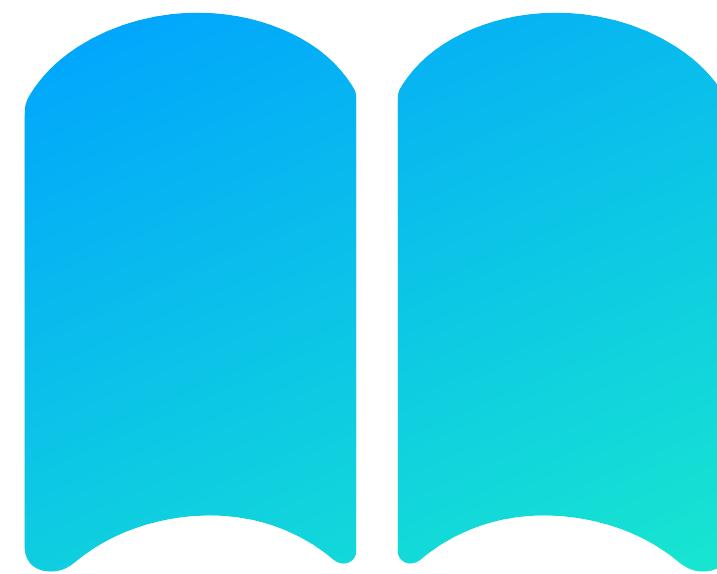
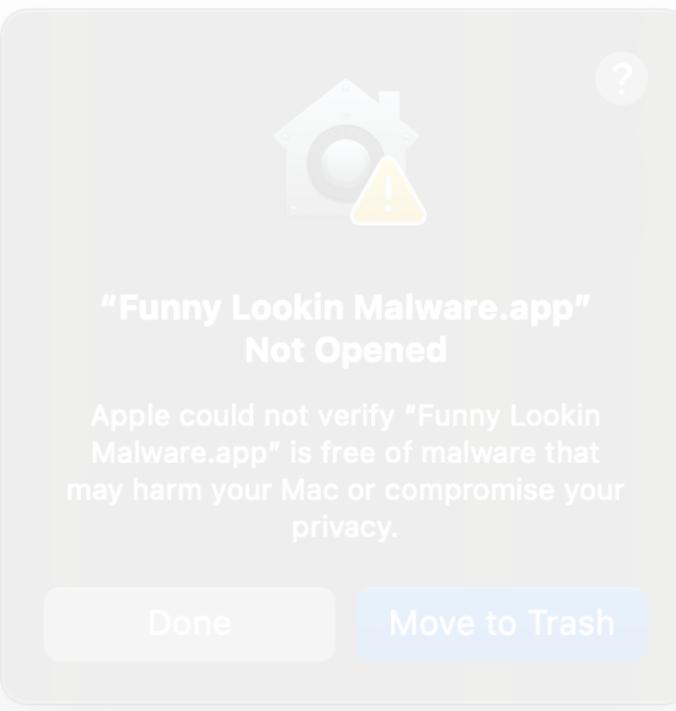


com.apple.systempolicy.control

"Enabled": false



Gatekeeper





Scripts



Scripts



Scripts



**Unsafe
Directories**



Scripts



**Unsafe
Directories**



**Dirty
environments**



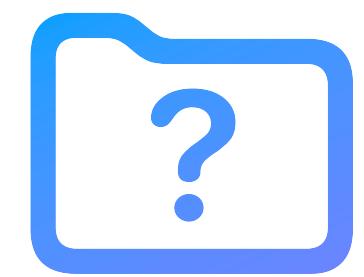
Unsafe Directories



Unsafe Directories

QNAP Updater Privilege Escalation

CVE-2024-53694



Unsafe Directories

QNAP Updater Privilege Escalation

CVE-2024-53694



`/Library/PrivilegedHelperTools/com.qnap.qsoftwareupdate`



Unsafe Directories

QNAP Updater Privilege Escalation

CVE-2024-53694



`/Library/PrivilegedHelperTools/com.qnap.qsoftwareupdate`

1. Tool is given a DMG



Unsafe Directories

QNAP Updater Privilege Escalation

CVE-2024-53694



`/Library/PrivilegedHelperTools/com.qnap.qsoftwareupdate`

1. Tool is given a DMG
2. Tool mounts DMG



Unsafe Directories

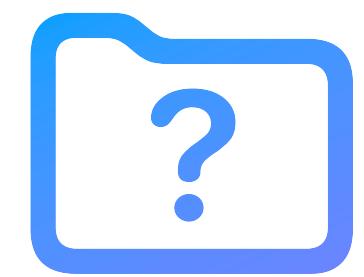
QNAP Updater Privilege Escalation

CVE-2024-53694



`/Library/PrivilegedHelperTools/com.qnap.qsoftwareupdate`

1. Tool is given a DMG
2. Tool mounts DMG
3. Tool verifies signature of PKG inside DMG



Unsafe Directories

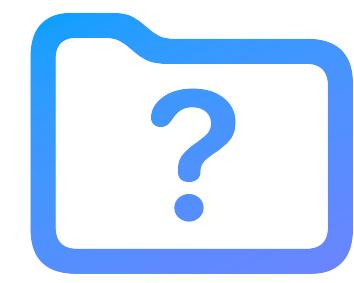
QNAP Updater Privilege Escalation

CVE-2024-53694



`/Library/PrivilegedHelperTools/com.qnap.qsoftwareupdate`

1. Tool is given a DMG
2. Tool mounts DMG
3. Tool verifies signature of PKG inside DMG
4. Tool copies PKG inside DMG to a working dir



Unsafe Directories

QNAP Updater Privilege Escalation

CVE-2024-53694



`/Library/PrivilegedHelperTools/com.qnap.qsoftwareupdate`

1. Tool is given a DMG
2. Tool mounts DMG
3. Tool verifies signature of PKG inside DMG
4. Tool copies PKG inside DMG to a working dir
5. Tool installs PKG



Unsafe Directories

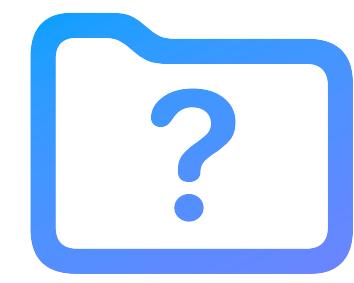
QNAP Updater Privilege Escalation
CVE-2024-53694



/Library/PrivilegedHelperTools/com.qnap.qsoftwareupdate



1. Tool is given a DMG
2. Tool mounts DMG
3. Tool verifies signature of PKG inside DMG
4. Tool copies PKG inside DMG to a working dir
5. Tool installs PKG



Unsafe Directories

QNAP Updater Privilege Escalation

CVE-2024-53694



`/Library/PrivilegedHelperTools/com.qnap.qsoftwareupdate`

1. Tool is given a DMG
2. Tool mounts DMG
3. Tool verifies signature of PKG inside DMG
4. Tool copies PKG inside DMG to a **working dir**
5. Tool installs PKG



Unsafe Directories

QNAP Updater Privilege Escalation

```
#!/bin/zsh
```

```
# Mount DMG file
```

```
/Lib
```

```
hdiutil attach Funny.dmg
```

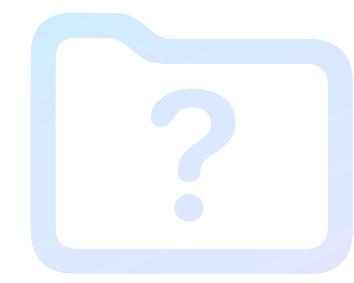
```
ater
```

```
# Copy PKG to /tmp
```

```
cp /Volumes/Funny/Funny.pkg /tmp/Funny.pkg
```

```
# Install the PKG
```

```
installer -pkg /tmp/Funny.pkg -target /
```



Unsafe Directories

QNAP Updater Privilege Escalation

```
#!/bin/zsh
```

```
# Mount DMG file
```

```
hdiutil attach Funny.dmg
```

```
# Copy PKG to /tmp
```

```
cp /Volumes/Funny/Funny.pkg /tmp/Funny.pkg
```

```
# Install the PKG
```

```
installer -pkg /tmp/Funny.pkg -target /
```

694

• qnap.qsoftwareupdate

← Anyone can write to it

f PKG inside DMG

G to a working dir

← Installed with root privileges



Unsafe Directories

QNAP Updater Privilege Escalation

```
#!/bin/zsh
```

"/bin/mv" and "/bin/cp" keep the original file owner if it exists at the destination.

```
# Mount DMG file
```

```
hdiutil attach F
```

```
# Copy PKG to /tr
```

```
cp /Volumes/Funny
```

Therefore place a dummy file, then replace the actual PKG with malware.

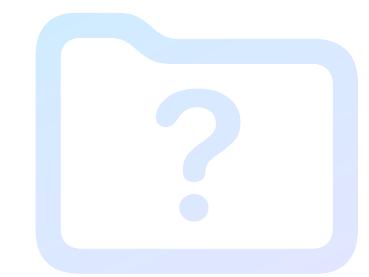
```
# Install the PKG
```

```
installer -pkg /tmp/Funny.pkg -target /
```

f PKG inside DMG

to a working dir

← Installed with root privileges



Unsafe Directories

QNAP Updater Privilege Escalation

```
#!/bin/zsh
```

```
# Mount DMG file
```

```
/Lib
```

```
hdiutil attach Funny.dmg
```

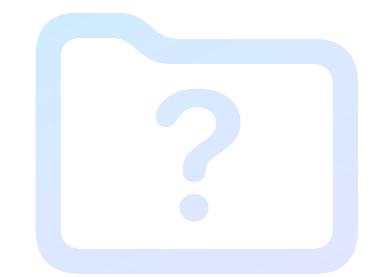
```
ter
```

```
# Copy PKG to /tmp
```

```
cp /Volumes/Funny/Funny.pkg /tmp/Funny.pkg
```

```
# Install the PKG
```

```
installer -pkg /tmp/Funny.pkg -target /
```



Unsafe Directories

QNAP Updater Privilege Escalation

```
#!/bin/zsh
```

```
# Mount DMG file
```

```
/Lib  
hdiutil attach Funny.dmg
```

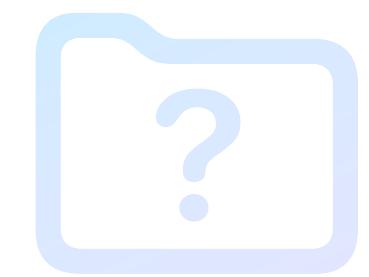
```
# Copy PKG to /tmp
```

```
cp /Volumes/Funny/Funny.pkg /tmp/Funny.pkg
```

```
# Install the PKG
```

```
installer -pkg /tmp/Funny.pkg -target /
```

← Change to a directory you control and know is empty.



Unsafe Directories

QNAP Updater Privilege Escalation

```
#!/bin/zsh
```

```
# Mount DMG file
```

```
/Lib
```

```
hdiutil attach Funny.dmg
```

```
ter
```

```
# Copy PKG to root-owned tmp dir
```

```
dir=$(mktemp -d)
```

```
cp /Volumes/Funny/Funny.pkg $dir/Funny.pkg
```

```
# Install the PKG
```

```
installer -pkg $dir/Funny.pkg -target /
```



Dirty environments



Dirty environments

macOS PackageKit Privilege Escalation
CVE-2024-27822



Dirty environments

macOS PackageKit Privilege Escalation
CVE-2024-27822

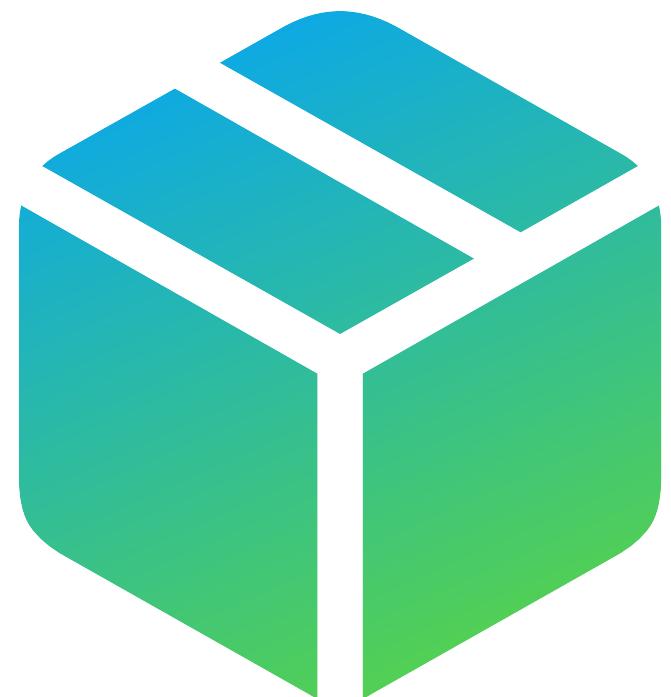


Install as root



Dirty environments

macOS PackageKit Privilege Escalation
CVE-2024-27822



Install as root

`#!/bin/zsh`



Dirty environments

macOS PackageKit Privilege Escalation
CVE-2024-27822



Install as root

`#!/bin/zsh`

`~/.zshenv`



Dirty environments

macOS PackageKit Privilege Escalation
CVE-2024-27822



Install as root

`#!/bin/zsh`

`~/.zshenv`

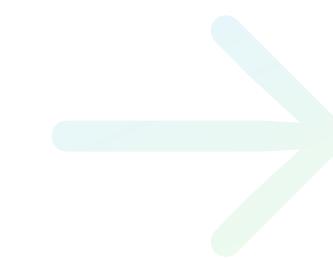
(User owned)



Dirty environments

macOS PackageKit Privilege Escalation
CVE-2024-27822

Free privilege escalation



Install as root

`#!/bin/zsh`

`~/.zshenv`

(User owned)

Preventing dirty envs

```
#!/bin/zsh
```

```
#!/bin/bash
```

```
#!/bin/sh
```

Preventing dirty envs

```
#!/bin/zsh      →  !#/bin/zsh --no-rcs  
#!/bin/bash    →  !#/bin/bash --noprofile --norc  
#!/bin/sh      →  !#/bin/sh (no change needed)
```

Preventing dirty envs

```
$ where mkdir  
/bin/mkdir
```

mkdir →

/bin/mkdir

codesign →

/usr/bin/codesign

mdmclient →

/usr/libexec/mdmclient

Preventing dirty envs

```
#!/bin/zsh

# Mount DMG file
hdiutil attach Funny.dmg

# Copy PKG to root-owned tmp dir
dir=$(mktemp -d)
cp /Volumes/Funny/Funny.pkg $dir/Funny.pkg

# Install the PKG
installer -pkg $dir/Funny.pkg -target /
```

m
c
m

Preventing dirty envs

```
#!/bin/zsh --no-rcs

# Mount DMG file
/usr/bin/hdiutil attach Funny.dmg

# Copy PKG to root-owned tmp dir
dir=$(/usr/bin/mktemp -d)
/bin/cp /Volumes/Funny/Funny.pkg $dir/Funny.pkg

# Install the PKG
/usr/sbin/installer -pkg $dir/Funny.pkg -target /
```



Software Catalogs



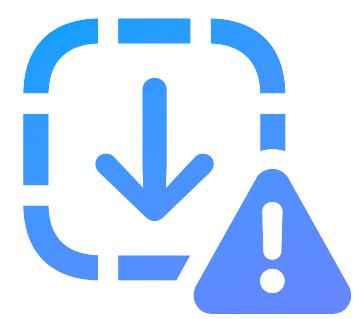
Software Catalogs



Software Catalogs



Outdated applications



Outdated applications



Outdated applications

FortiClient Privilege Escalation
CVE-2025-46774



Outdated applications

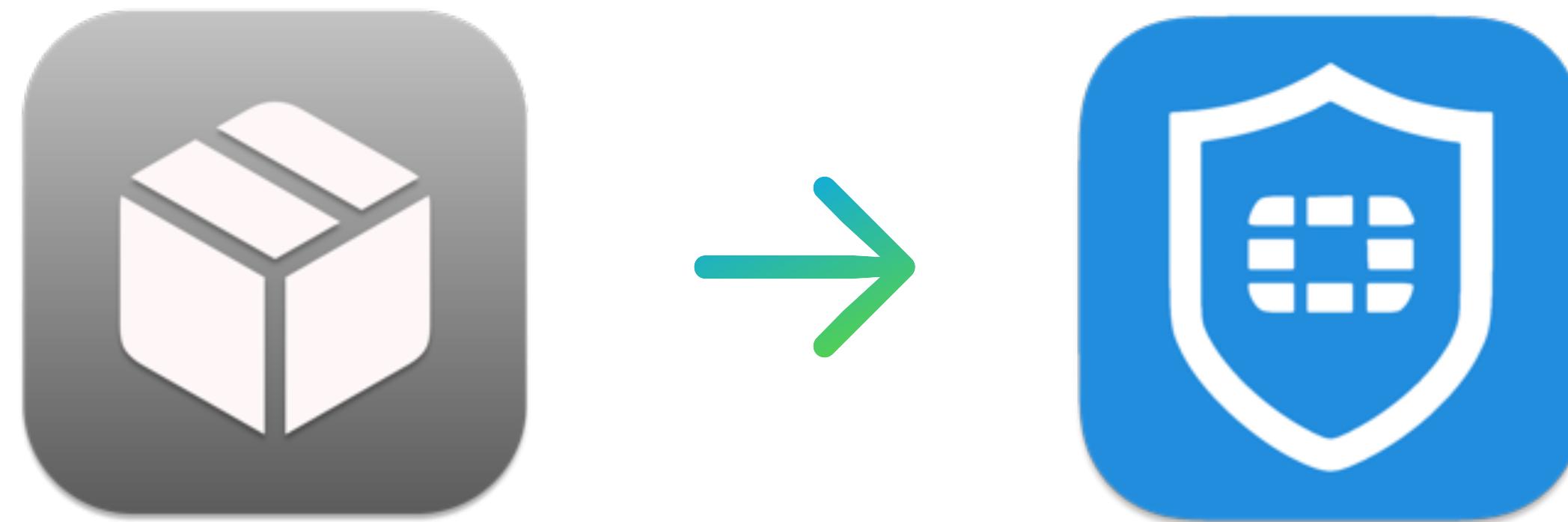
FortiClient Privilege Escalation
CVE-2025-46774





Outdated applications

FortiClient Privilege Escalation
CVE-2025-46774





Outdated applications

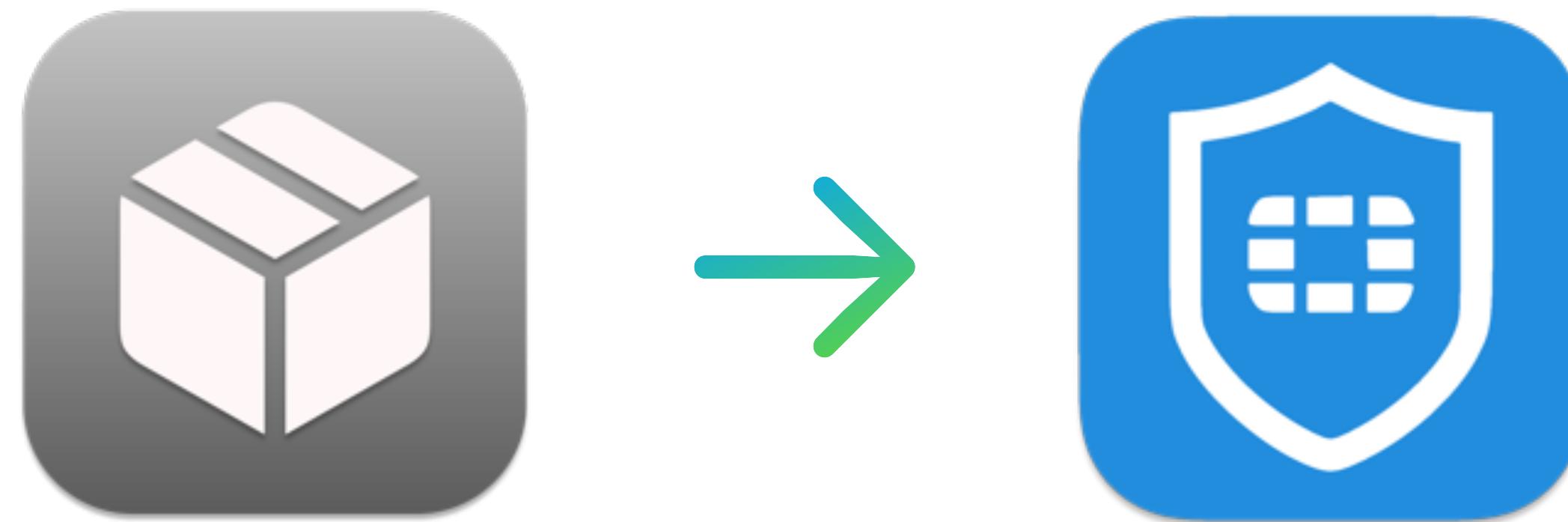
FortiClient Privilege Escalation
CVE-2025-46774





Outdated applications

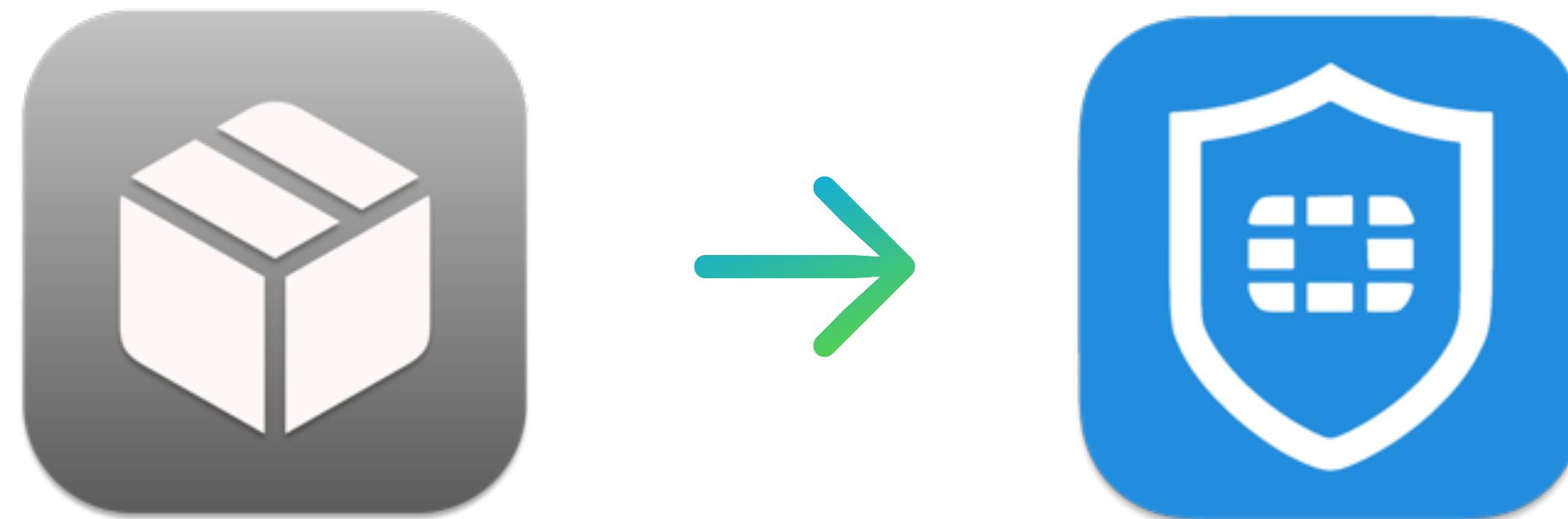
FortiClient Privilege Escalation
CVE-2025-46774





Outdated applications

FortiClient Privilege Escalation
CVE-2025-46774



(Outdated)

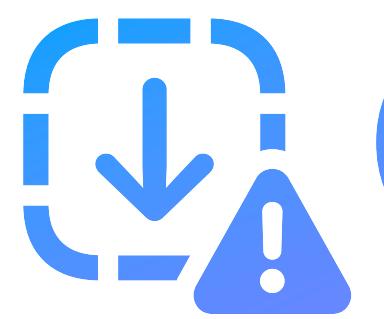


Outdated applications

FortiClient Privilege Escalation
CVE-2025-46774



Privilege escalation through installation



Outdated applications

FortiClient Privilege Escalation
CVE-2025-46774





Outdated applications

FortiClient Privilege Escalation
CVE-2025-46774

Best ways to handle?



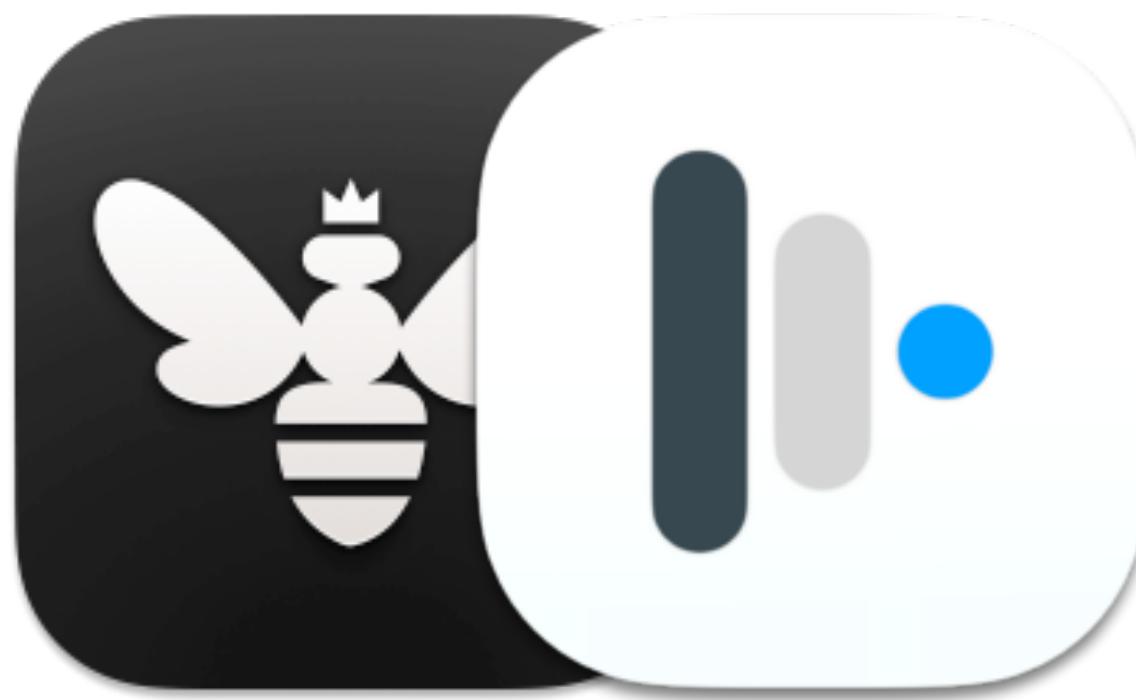
Best ways to handle?

Best ways to handle?



**MDM provided
apps**

Best ways to handle?

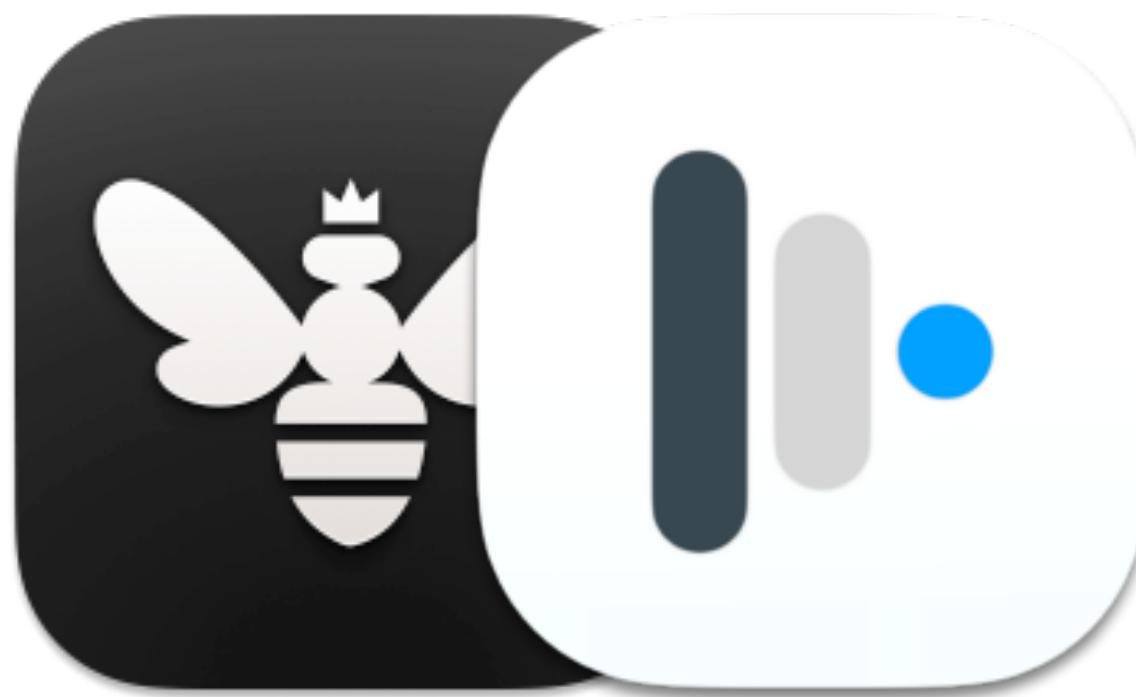


**MDM provided
apps**



Munki + AutoPkg

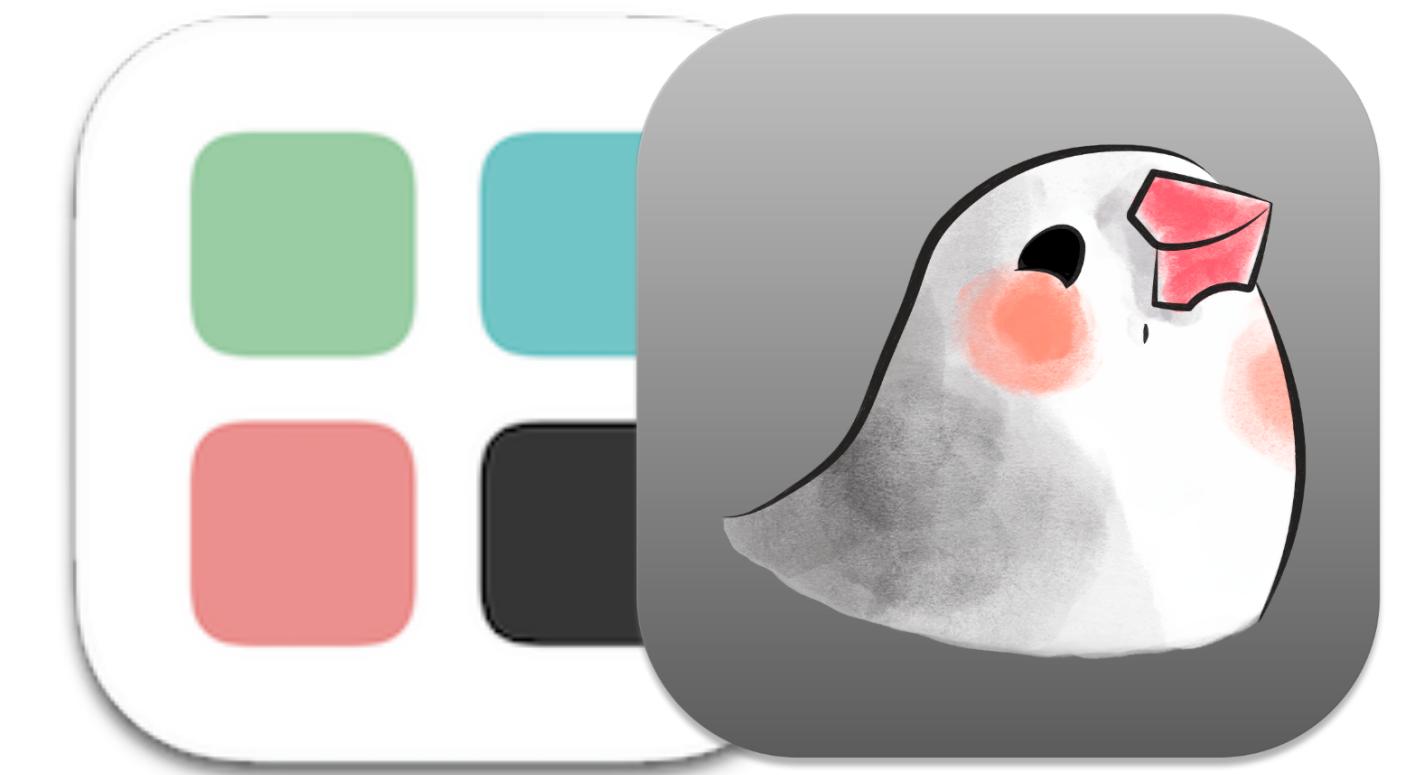
Best ways to handle?



**MDM provided
apps**



Munki + AutoPkg



**Root3 App Catalog /
Electrona Patch**



Enrollments



Enrollments



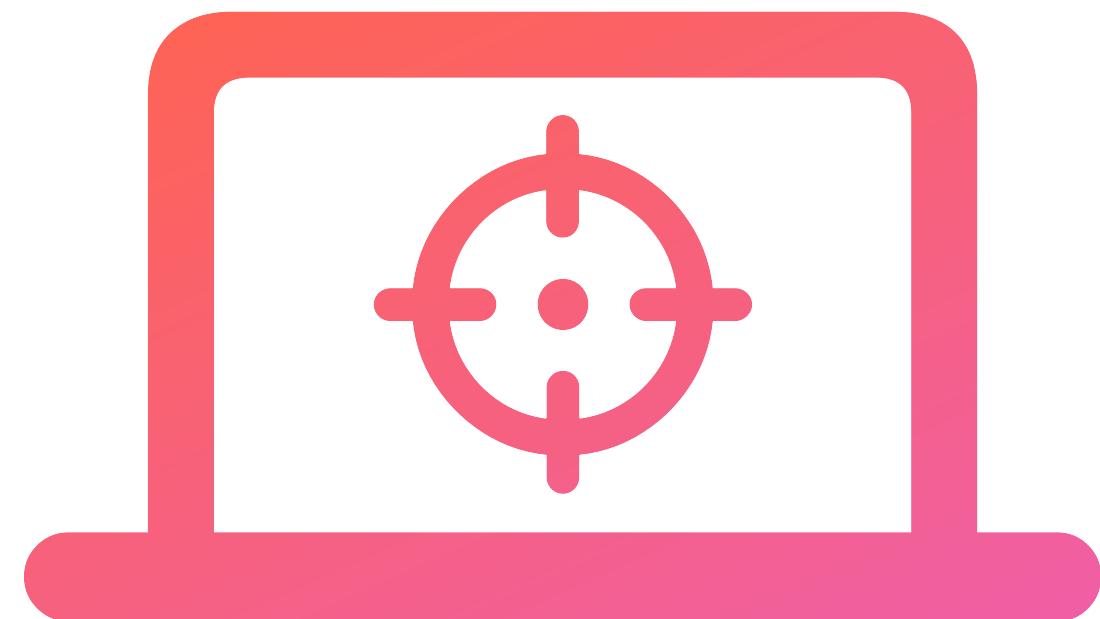
Enrollments



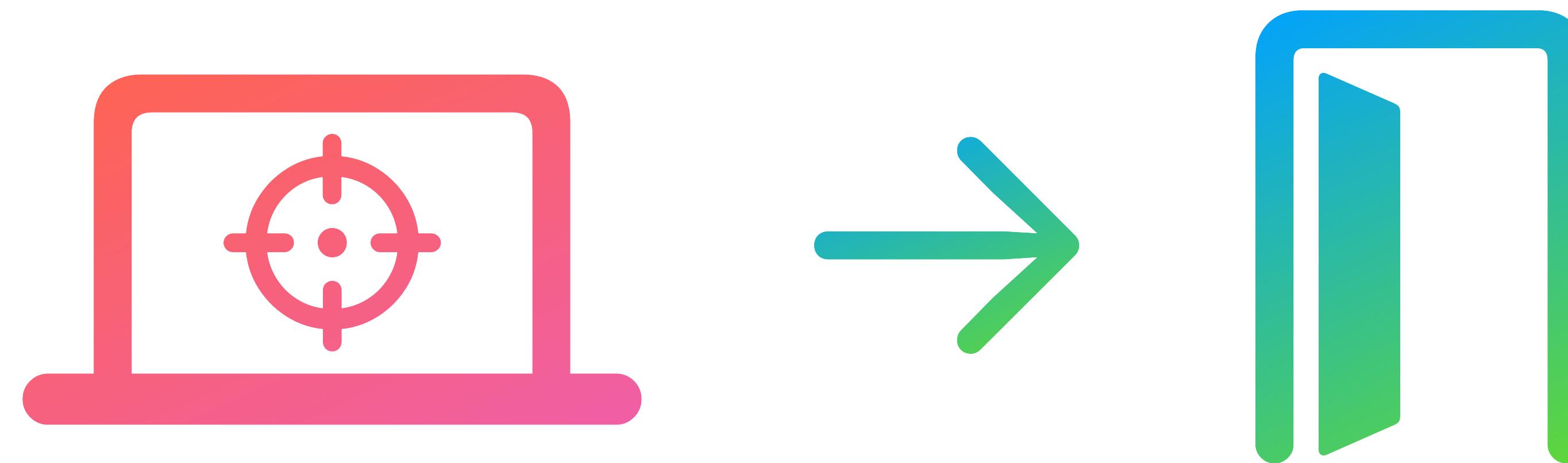
Weak / missing authentication



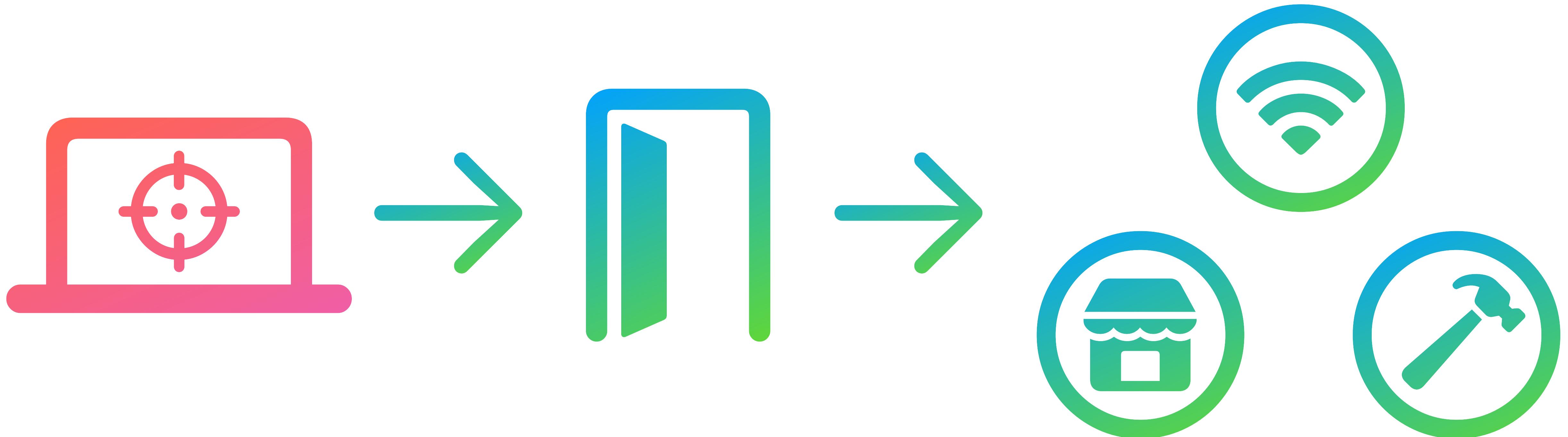
Weak / missing authentication



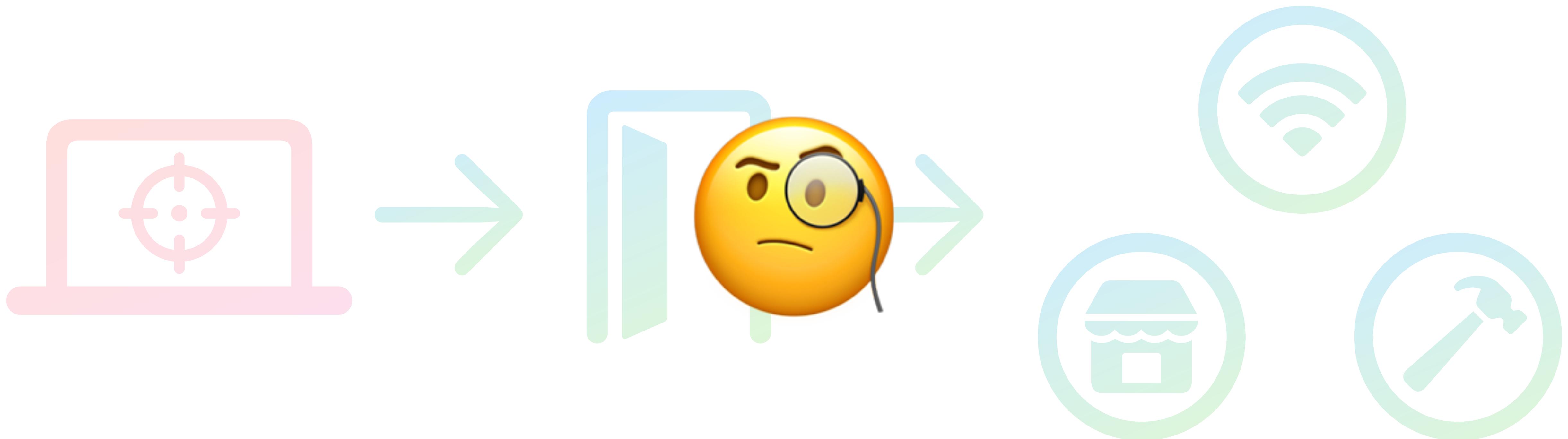
Weak / missing authentication



Weak / missing authentication



⚠ Weak / missing authentication



 Weak / missing authentication

Lock down enrollments 

Authentication

Quarantine

Querying MDM enrollments

Querying MDM enrollments



Project Indago

Latin - Search

Querying MDM enrollments

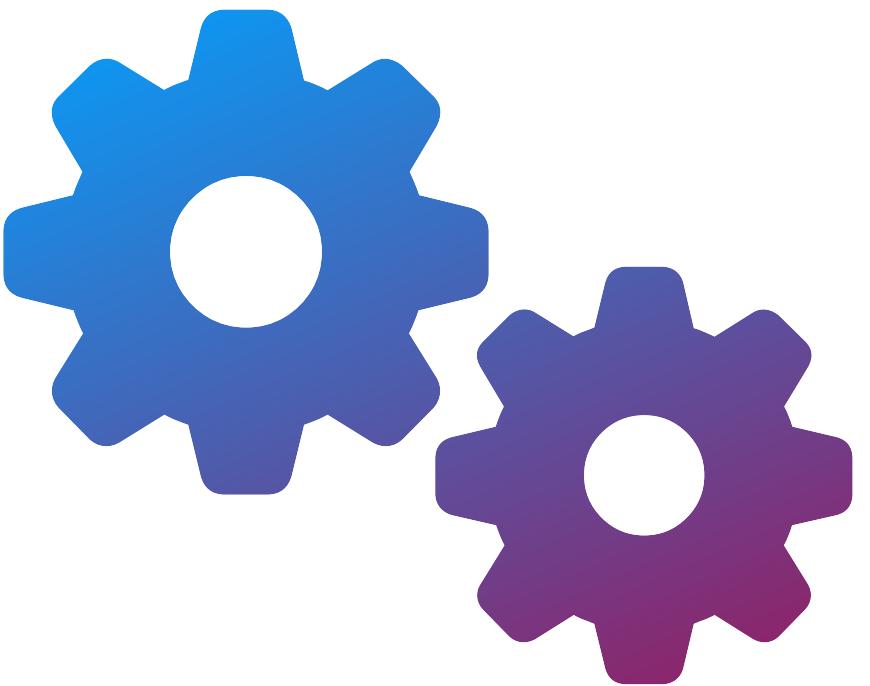


Project Indago

Latin - Search



Querying MDM enrollments



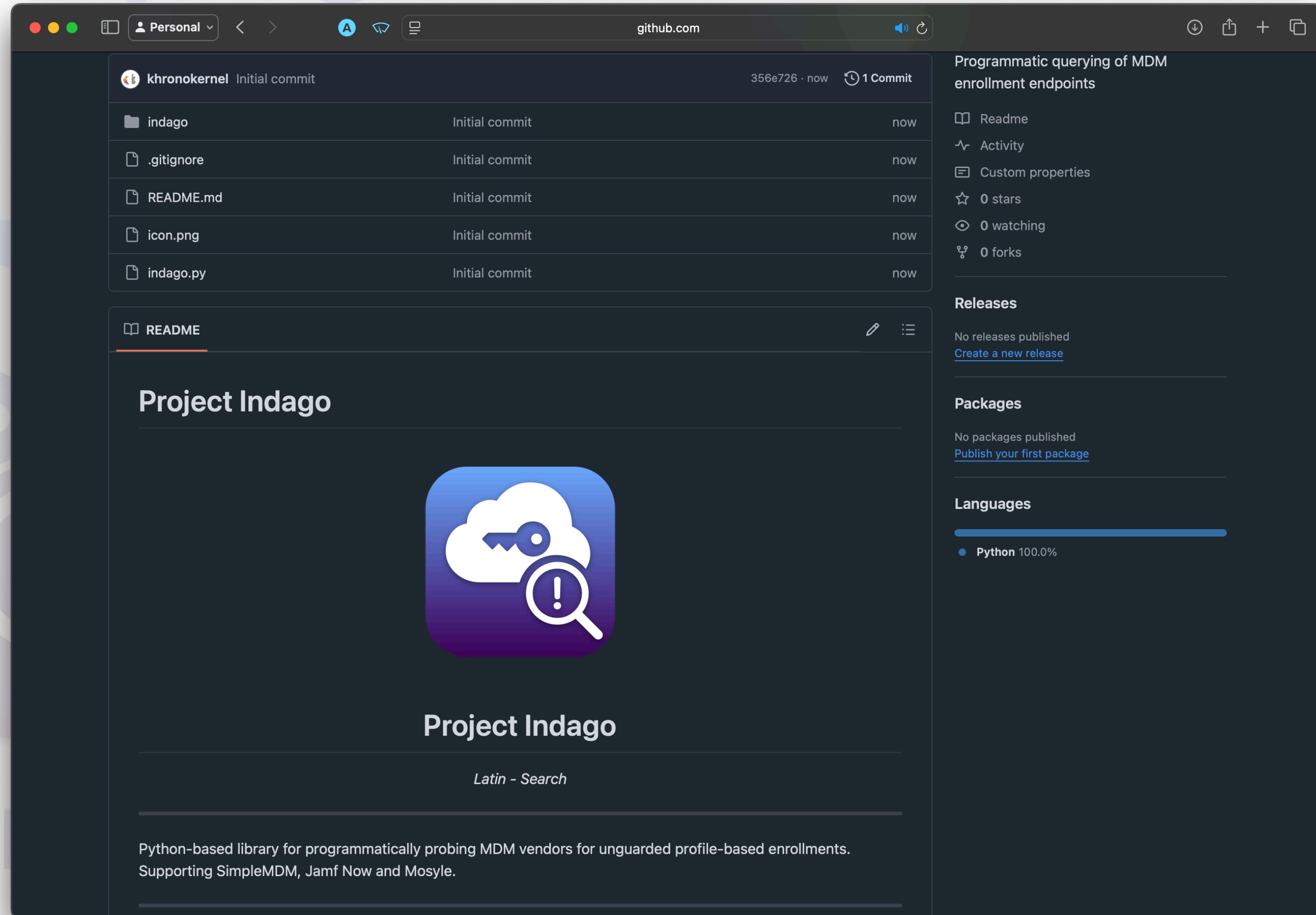
01101001 01101110 01100100
01100001 01100111 01101111

Project Indago

Latin - Search



Querying MDM enrollments



Project Indago

Latin - Search



SimpleMDM

Querying MDM enrollments

The screenshot shows a GitHub repository page for 'Project Indago'. The repository was created by 'khronokernel' and has one commit. The repository contains files: .gitignore, README.md, icon.png, and indago.py. The README file is currently selected. It features a large icon of a cloud with a key and a magnifying glass. The description of the project is as follows:

Project Indago

Python-based library for programmatically probing MDM vendors for unguarded profile-based enrollments.
Supporting SimpleMDM, Jamf Now and Mosyle.

Programmatic querying of MDM enrollment endpoints

- Readme
- Activity
- Custom properties

0 stars
0 watching
0 forks

No releases published
[Create a new release](#)

No packages published
[Publish your first package](#)

Languages

Python 100.0%

Latin - Search



Querying SimpleMDM



Simple MDM

Querying SimpleMDM

<https://a.simplemdm.com/enroll/?c=xxxxxxxx>



Simple MDM

Querying SimpleMDM

<https://a.simplemdm.com/enroll/?c=xxxxxxxx>

Identifier format:

- **8 characters**
- **Integers only**



Simple MDM

Querying SimpleMDM

<https://a.simplemdm.com/enroll/?c=xxxxxxxx>

Identifier format:

- 8 characters
- Integers only

Samples:

- 94756678
- 43409620
- 13012603
- 41895752



Simple MDM

Querying SimpleMDM

<https://a.simplemdm.com/enroll/?c=xxxxxxxx>

Identifier format:
- 8 characters
- Integers only

Samples:
- 94756678
- 43409620
- 13012603
- 41895752

Authentication methods:
- None
- SAML



Simple MDM

Querying SimpleMDM

<https://a.simplemdm.com/enroll/?c=xxxxxxxx>

Identifier format:
- 8 characters
- Integers only

Samples:
- 94756678
- 43409620
- 13012603
- 41895752

Authentication methods:
- None
- SAML

Recommendations:
- URL complexity
- Challenges
- Passcode



Simple MDM

Querying SimpleMDM

<https://a.simplemdm.com/enroll/?c=xxxxxxxx>

Identifier format:
- 8 characters
- Integers only

Samples:
- 94756678
- 43409620
- 13012603
- 41895752

Authentication methods:
- None
- SAML

Recommendations:
- URL complexity
- Challenges
- Passcode

Notified
September 2024

URL complexity!

<https://a.simplemdm.com/enroll/?>

C=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX



SimpleMDM

Querying SimpleMDM

<https://a.simplemdm.com/enroll/?c=xxxxxxxx>

Existing enrollments

Identifier format:

- 8 characters
- Integers only

Notified:

still vulnerable



Solution:

Increase URL length
and complexity

Querying Jamf Now



Querying Jamf Now

<https://go.jamfnow.com/xxxxxx>



Querying Jamf Now

<https://go.jamfnow.com/xxxxxx>

Identifier format:

- 6 characters
- Integers, upper and lower case letters



Querying Jamf Now

<https://go.jamfnow.com/xxxxxx>

Identifier format:

- 6 characters
- Integers, upper and lower case letters

Samples:

- p0PgnX
- ZddV5k
- 2BE0Ca
- u009jt



Querying Jamf Now

<https://go.jamfnow.com/xxxxxx>

Identifier format:

- 6 characters
- Integers, upper and lower case letters

Samples:

- p0PgnX
- ZddV5k
- 2BE0Ca
- u009jt

Authentication method:

- Passcode (required 



Querying Jamf Now

<https://go.jamfnow.com/xxxxxx>

Identifier format:

- 6 characters
- Integers, upper and lower case letters

Samples:

- p0PgnX
- ZddV5k
- 2BE0Ca
- u009jt

Authentication method:

- Passcode (required 

Recommendations:

- URL complexity
- Challenges



Querying Jamf Now

<https://go.jamfnow.com/xxxxxx>

Identifier format:

- 6 characters
- Integers, upper and lower case letters

Samples:

- p0PgnX
- ZddV5k
- 2BE0Ca
- u009jt

Authentication method:

- Passcode (required 

Recommendations:

- URL complexity
- Challenges

**Notified
January 2025**



NOW

Querying Jamf Now

<https://go.jamfnow.com/xxxxxx>

Identifier format:

- 6 characters
- Integers, upper and lower case letters

Samples:

- p0PgnX
- 2B20G
- u009jt

-\(\wedge\)(\u009jt)-

Authentication method:

- Passcode (required)

?

Recommendations:

- URL complexity
- Challenges

Notified
January 2025

?

Querying Mosyle



Querying Mosyle

<https://join.mosyle.com/xxxxxx>



Querying Mosyle

<https://join.mosyle.com/xxxxxx>

Identifier format:

- **6 characters**
- **Integers and uppercase letters**



Querying Mosyle

<https://join.mosyle.com/xxxxxx>

Identifier format:

- **6 characters**
- **Integers and uppercase letters**

Samples:

- 6WK4K9
- TU7U33
- KQ9PMT
- VPT71Z



Querying Mosyle

<https://join.mosyle.com/xxxxxx>

Identifier format:

- **6 characters**
- **Integers and uppercase letters**

Samples:

- 6WK4K9
- TU7U33
- KQ9PMT
- VPT71Z

Authentication methods:

- **None**
- **SAML**
- **Pre-approved devices**



Querying Mosyle

<https://join.mosyle.com/xxxxxx>

Identifier format:

- **6 characters**
- **Integers and uppercase letters**

Samples:

- 6WK4K9
- TU7U33
- KQ9PMT
- VPT71Z

Authentication methods:

- **None**
- **SAML**
- **Pre-approved devices**

Recommendations:

- **URL complexity**
- **Challenges**
- **Disable codes for new groups**



Querying Mosyle

<https://join.mosyle.com/xxxxxx>

Identifier format:

- **6 characters**
- **Integers and uppercase letters**

Samples:

- 6WK4K9
- TU7U33
- KQ9PMT
- VPT71Z

Authentication methods:

- **None**
- **SAML**
- **Pre-approved devices**

Recommendations:

- **URL complexity**
- **Challenges**
- **Disable codes for new groups**

Notified
September 2024



Querying Mosyle

<https://join.mosyle.com/xxxxxx>

Identifier format:

- 6 characters
- Integers and uppercase letters

Samples:

6WK4K9
107833
K09PMT
07-27-2024

Authentication methods:

- SAML
- Pre-approved devices

**Disable unused enrollments
after 90 days**

Recommendations:

- URL complexity
- Challenges
- Disable codes for new groups

Notified
September 2024



Querying Mosyle

<https://join.mosyle.com/xxxxxx>

Identifier format:

- 6 characters
- Integer string of uppercase letters

Still dead easy to abuse

Samples:

- 6WK4K9
- TH7U33
- VPT71Z

Authentication methods:

- None
- Pre-applied devices



Recommendations:

- URL complexity
- Challenges
- Disable codes for new groups

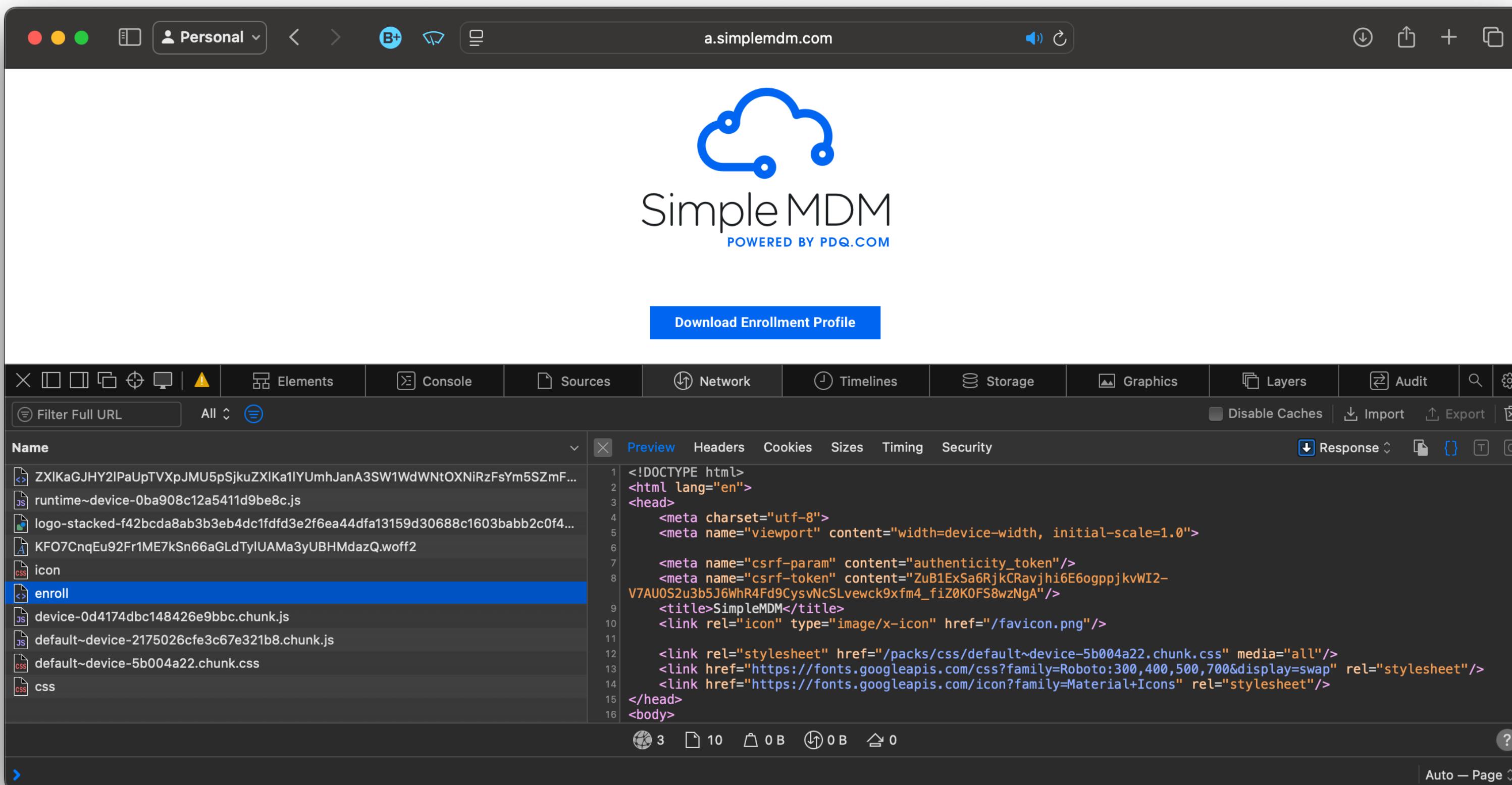
Notified
September 2024

Future research - Other MDMs?

Future research - Other MDMs?

Safari -> Settings -> Advanced -> Show features for web developers

Develop -> Show web inspector

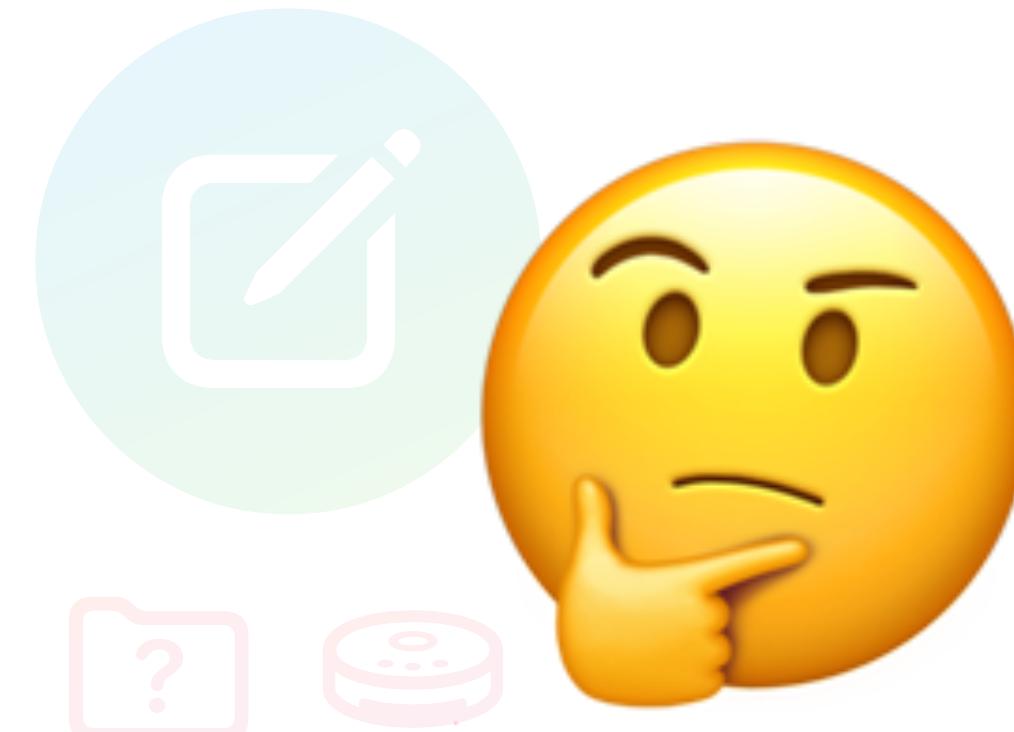


Wrapping up

Wrapping up



Wrapping up



Thanks for listening to my rambles!

Mirrored on khronokernel.com



Socials I guess?

- Twitter: <https://twitter.com/khronokernel>
- GitHub: <https://github.com/khronokernel>
- LinkedIn: <https://www.linkedin.com/in/mykola-grymalyuk>

References

- Project Indago: <https://github.com/ripeda/indago>
- Project Lectricus: <https://github.com/ripeda/lectricus>
- Unit 42 Blog: <https://unit42.paloaltonetworks.com/macos-stealers-growing/>
- Apple Configuration Profile Reference: <https://developer.apple.com/business/documentation/Configuration-Profile-Reference.pdf>
- Root3's App Catalog: <https://appcatalog.cloud>
- Electrona Patch: <https://www.electrona.com/patch>
- Munki: <https://github.com/munki/munki>
- AutoPkg: <https://github.com/autopkg/autopkg>
- SimpleMDM: <https://simplemdm.com/>
- Mosyle: <https://mosyle.com/>
- Jamf Now: <https://www.jamf.com/products/jamf-now/>