Supercharge your user's Microsoft 365 experience with Enterprise Single Sign On

About me

Merill Fernando
Product Manager, Microsoft

merill.f@microsoft.com

@merill
@merill@infosec.exchange

Agenda

What is Azure AD?

Prompting…why is it bad?

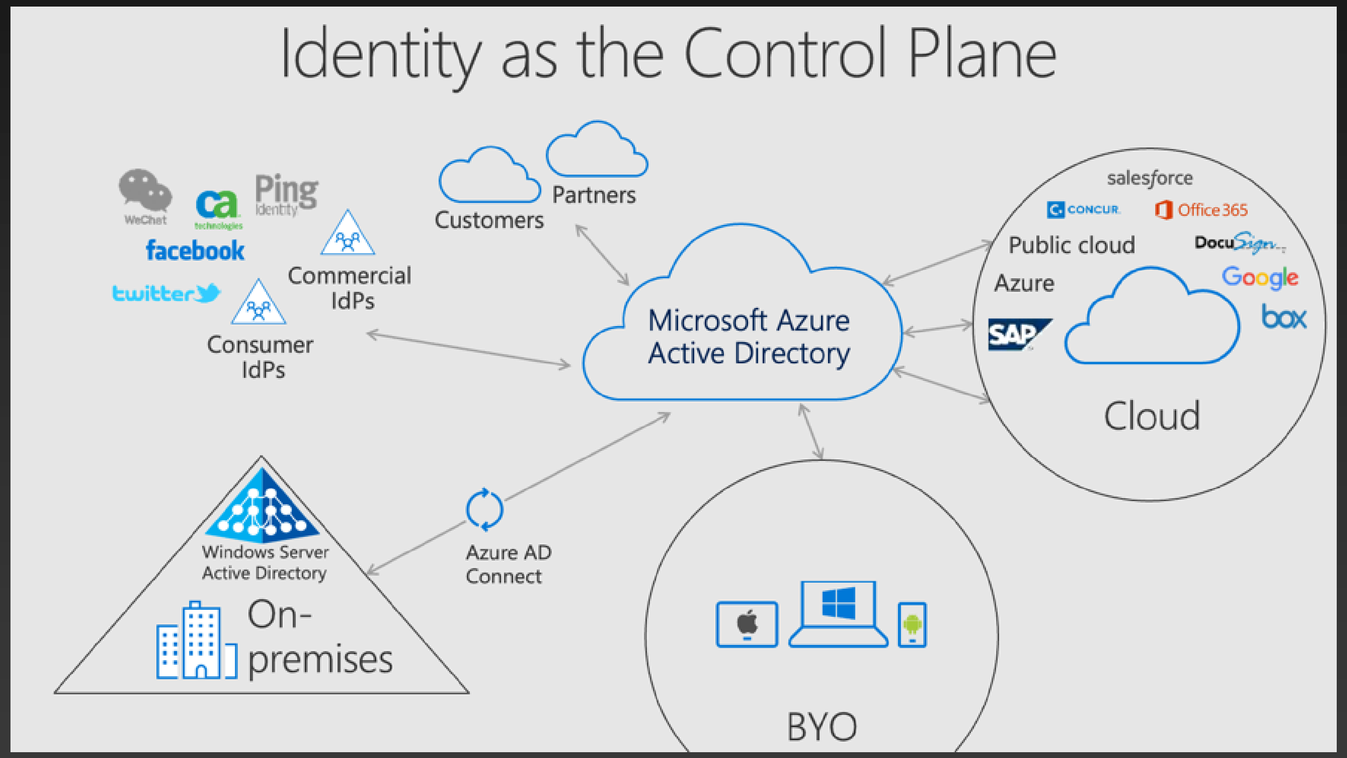
Enterprise Single Sign On (SSO) - How does it work?

Deploying Enterprise SSO

Troubleshooting Enterprise SSO

Azure AD

- Azure AD is a full blown IDaaS solution, not an IDP for just Office 365/Azure
- Resources are moving to the cloud, devices are proliferating, users are outside the office
- Identity needs to be the new control plane, rather than the network perimeter



Agenda

What is Azure AD?

Prompting…why is it bad?

Enterprise Single Sign On (SSO) - How does it work?

Deploying Enterprise SSO

Troubleshooting Enterprise SSO


Amy
@amysw_sec
PSA... don't blindly accept MFA requests if you're nc
trying to log in to something. That is all.
1:26 AM · Apr 13, 2021 · Twitter Web App
21 Retweets
4 Quote Tweets
199 Likes


Reg
@RegGBlinker
Replying to @SchizoDuckie and @amysw_sec
Unfortunately, I found a company today who refreshes
their users credentials every morning, so each morning
their entire workforce gets a push notification to login,
itiated access at that time. So,
(disclaimer: not my org!)
Phone


K. Reid Wightman
@ReverseICS
I kind of want to write an app that tracks how many hours
per week I spend 2FA'ing into different collaboration
systems.
7:15 AM · Apr 27, 2021 · TweetDeck
4 Retweets
65 Likes

Customer Case Study

European financial company simulated cyber attack.

- Attackers used password spray to find users with weak passwords.
- Users with compromised passwords were “hammered” with MFA prompts.

Findings:

- No reports of unexpected prompts to the help desk.
- Many users blindly approved MFA requests.
- One user had uninstalled the Authenticator app.

Recent phishing attacks that made the news

CyberSecurity Connect

services.

It is believed that the entry point that caused the cyber attack on the private health insurer was when a person with high-level access within Medibank's systems had their credentials stolen by a hacker. The information was then sold on a Russian-language cyber crime forum, according to a report from *The Guardian* that attributed the information to a source who was not authorised to speak publicly.

UpGuard

8. Australian Parliament House Data Breach

AUSTRALIAN PARLIAMENT HOUSE

Date: February 2019

Impact: Multiple political party networks - Liberal, Labor, and the Nationals.

The cybercriminals used phishing methods to steal employee credentials and gain entry into the government's network. This precursor attack took place on an infected external website that a small number of parliament staff visited.

6. Service NSW Data Breach

Service NSW

Date: April 2020

Impact: 104,000 people

47 Service NSW staff email accounts were hacked through a series of phishing attacks. This led to 5 million documents being accessed, 10 percent of which contains sensitive data impacting 104,000 people.

High volume AiTM phishing kits

Microsoft

Microsoft Security

March 13, 2023 • 7 min read

DEV-1101 enables high-volume AiTM campaigns with open-source phishing kit

Microsoft Threat Intelligence

Figure 1. DEV-1101 announcement on their AiTM tool license

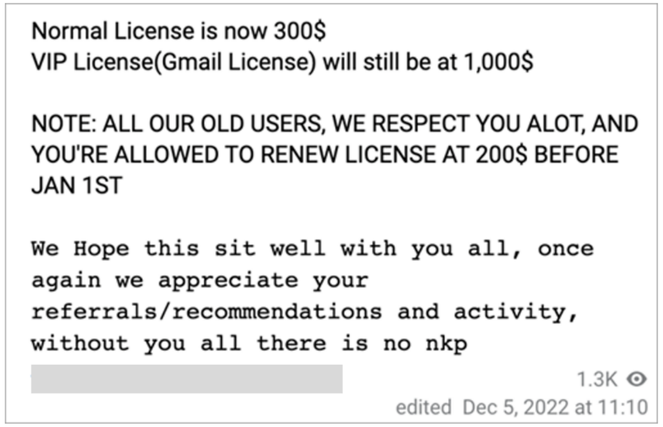
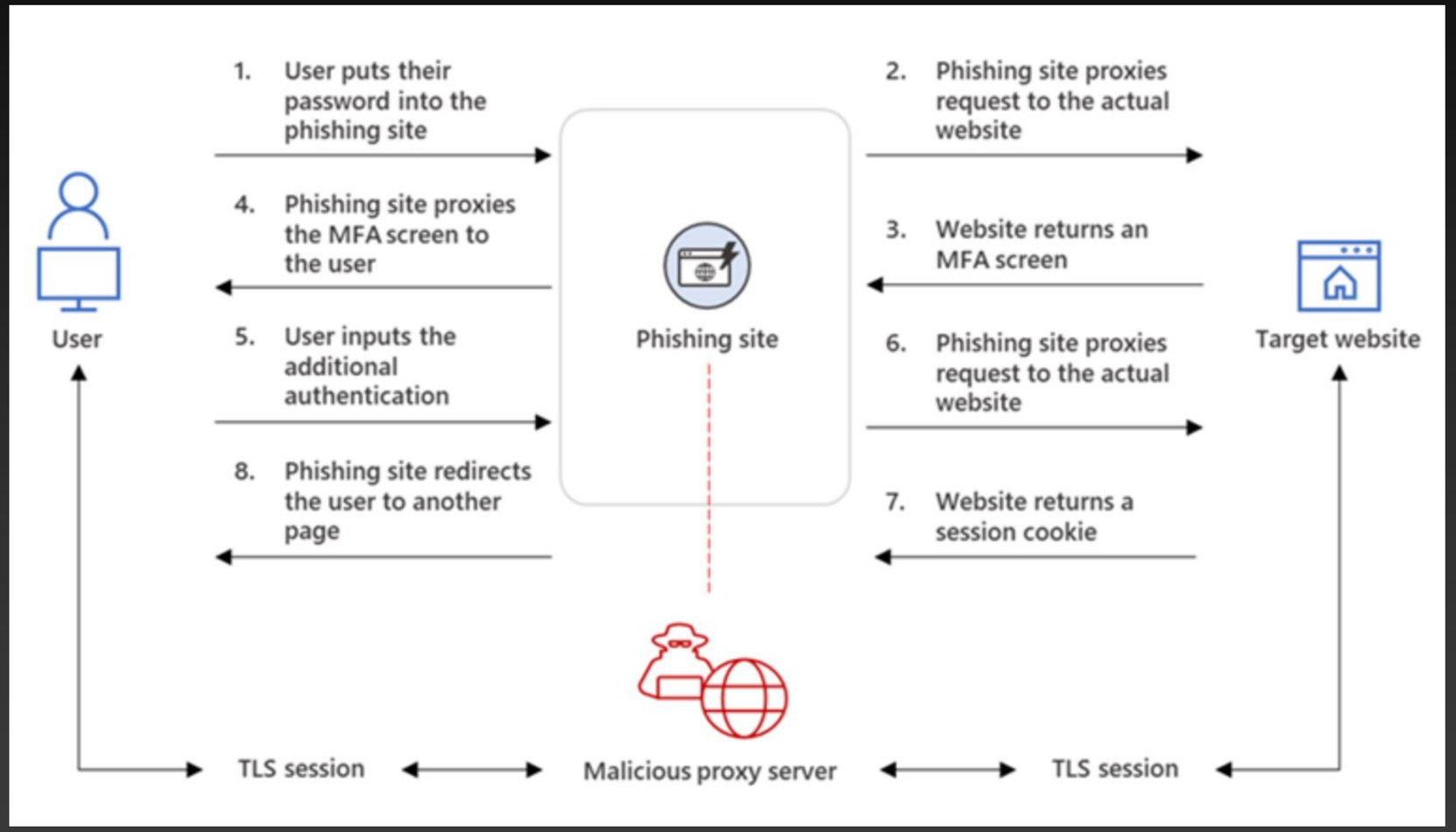


Figure 2. DEV-1101 increased their prices repeatedly due to rapid growth.

How AiTM phishing works



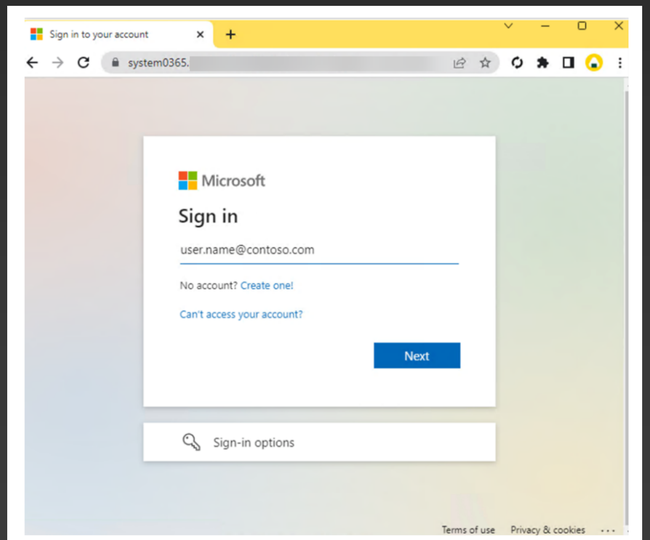


Figure 7. Credential harvester mimicking a Microsoft sign-in portal.

Why Prompting is Bad

- Over-prompting leads to compromise
  - Users learn bad behaviors, like blindly approving MFA requests
- Prompts impact productivity, especially on platforms without SSO
- Prompting is especially common on macOS, which does not do SSO with Azure AD out of the box
- Should strive to improve user experience AND security
  - Prompt when *needed*, such as new device, new location, change in risk, etc.
  - Passwordless makes prompting less impactful when it IS needed

Prompts are bad

Agenda

What is Azure AD?

Prompting…why is it bad?

Enterprise Single Sign On (SSO) - How does it work?

Deploying Enterprise SSO

Troubleshooting Enterprise SSO

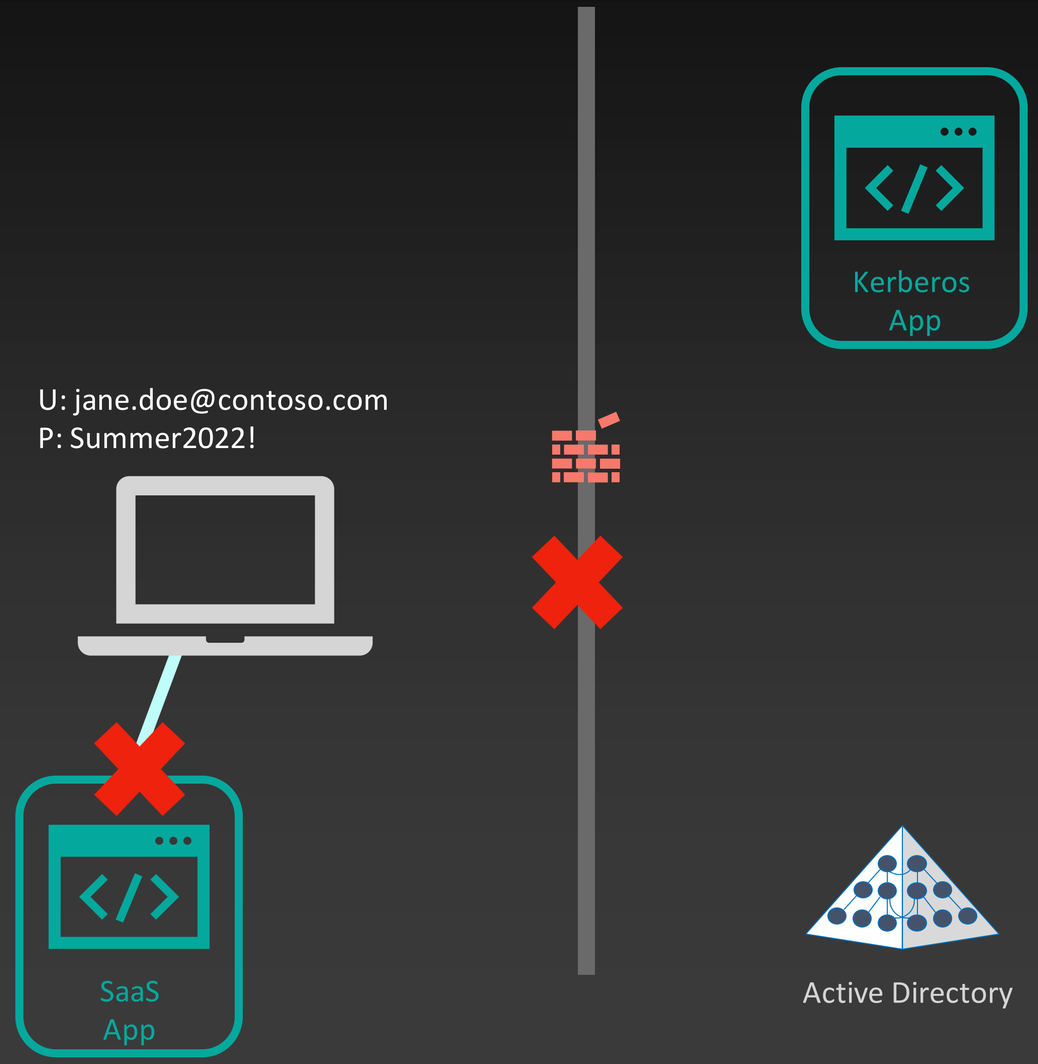
Set up SSO Infrastructure

- macOS can provide SSO in a few different ways:
  - Kerberos, via BIND to an LDAP directory, commonly on-premises Active Directory
    - Apple is actively telling customers to move away from this
  - Kerberos, via Apple's Kerberos SSO Extension
    - Must be deployed through MDM
    - Still designed for on-premises directory services, not really designed for the cloud
  - Modern Auth (tokens), via IDP vendor-provided plug-ins for Apple's Extensible Enterprise SSO Framework
    - IDP vendor…that's me!
    - Must be deployed through MDM
    - Two types:
      - Credential
      - Redirect – Azure AD's option is this type

SSO with Kerberos just doesn’t cut it anymore

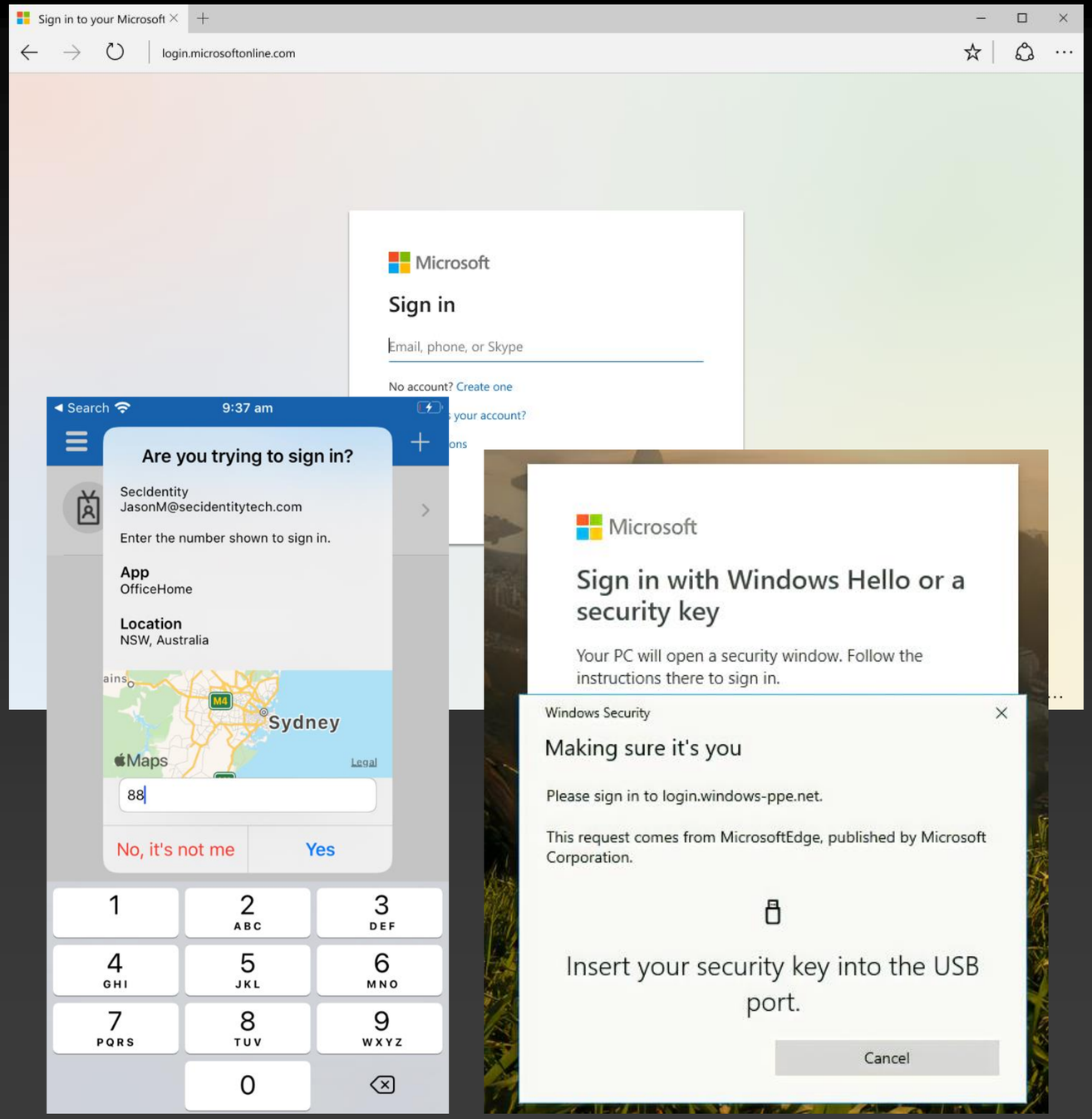
- What’s the issue with Kerberos SSO?
- It doesn’t work over the internet, so it isn’t very modern
- Imagine we have a SaaS app instead of an internal Kerberos app
- Kerberos doesn’t make sense for the SaaS app, because devices on the internet shouldn’t be able to find a DC

1) User provides device with their enterprise username and password
2) Should the device still want to send the creds to AD and ask for a Kerberos Ticket-Granting Ticket (TGT)?
3) No, this won’t work without a VPN

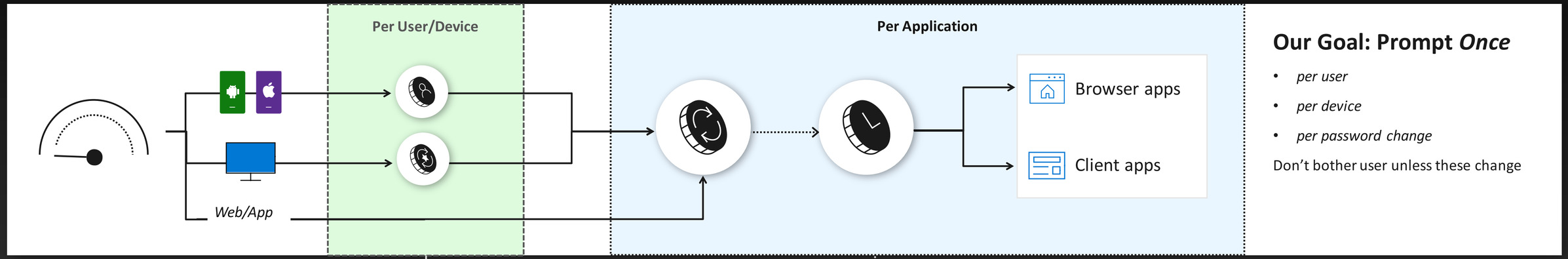


SSO – Modernize w/ Modern Auth

- The solution is Modern Auth!
  - SAML – good
  - OpenID Connect and OAuth 2 - better!
- The key advantage of Modern Auth is that it is web-based
  - The flexibility of web technology gives us many security options:
    - Challenge for certificates
    - Many forms of MFA (FIDO, Auth apps, Smartcards, SMS codes, etc.)
    - Direct traffic through proxied sessions to block downloads
    - And much more!



Our mission



Our Goal: Prompt *Once*

- *per user*
- *per device*
- *per password change*

Don't bother user unless these change

AUTHENTICATION

Until Revoked or Password Change (If actively used within 14 days)

Primary Refresh Token

Long term authentication w/ SSO broker on Windows, macOS, or iOS

ID Token

Long term authentication on Mobile Device
(used by authenticator app and/or company portal)
Note: Authenticator App has two functions: brokering authentication locally + MFA validation

(COARSE) AUTHORIZATION

Refresh Token – (Per App)

Long term access to an application

Note: Includes whether MFA was used for authentication

Until revoked or Password Changed

Access Token – (Per App)

1 hours

Provides user access to use application (short term)
Note: Policy is re-evaluated every time you get a new access token (using the refresh token)

SSO – Modernize w/ Modern Auth

- Here's what you need for Modern Auth and SSO on Apple Platforms:
  - IDP that supports SAML and/or OpenID Connect
    - Azure AD is Microsoft's cloud IDP, but there are plenty of others on the market
  - Apps integrated with the IDP
  - IDP Vendor must create an SSO Extension plugin
  - Macs under MDM management

SSO – Modernize w/ IDP Vendor SSO Extensions

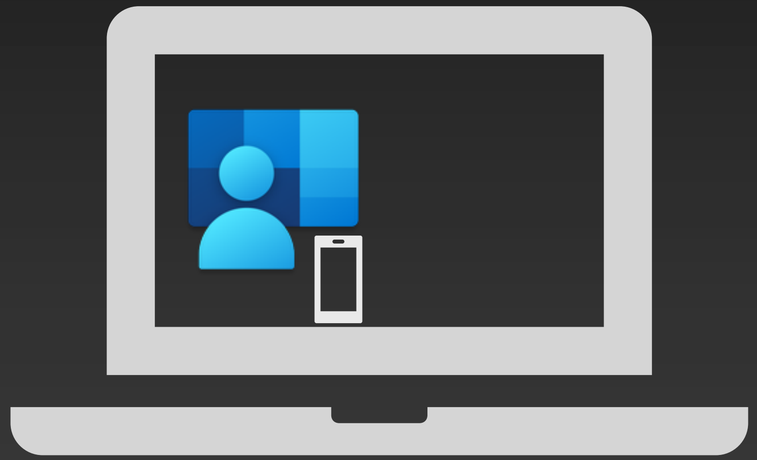
- The *modern* approach is to use an IDP, modern auth, and tokens
- SSO Extension is bundled in the Microsoft Company Portal

1) User authenticates to Azure AD in the SSO Extension window – this can be in Company Portal or another app, such as Safari
    - Azure AD supports many more credential types than AD does
2) Azure AD SSO Extension acquires a Primary Refresh Token (PRT) from Azure AD after the user signs in, stores it in the keychain
    - PRTs are good for a rolling 14 day window, constantly refreshed when the user uses the Mac

MS Authenticator Passwordless Phone Sign-In

Username+Pwd+MFA (App, OTP, SMS, Phone)

App



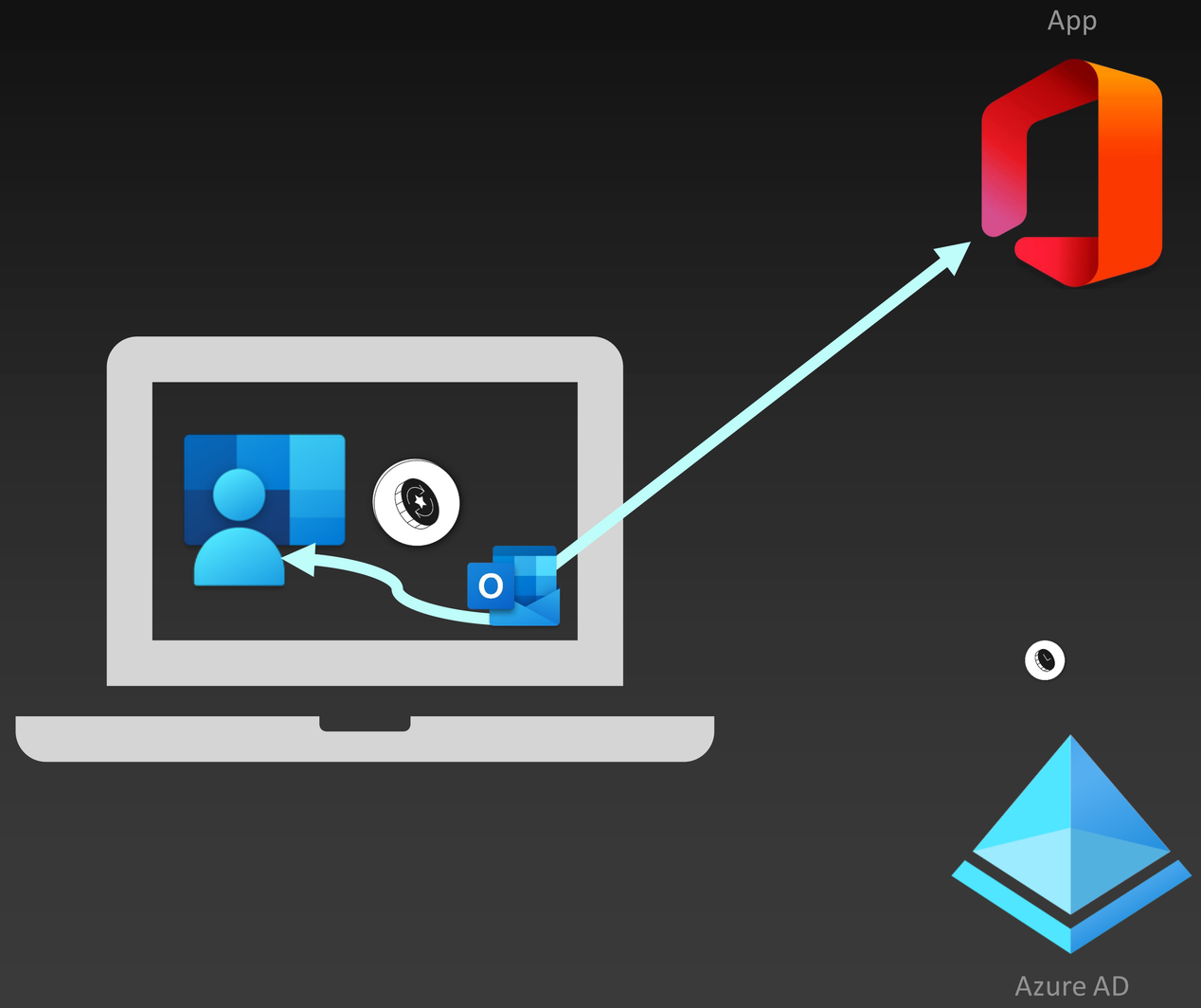
Azure AD

SSO Extension with MSAL

One more wrinkle…there's two different flows for apps to get tokens

We'll start with the MSAL flow (MSAL is Microsoft Authentication Library, our auth library provided to make app integration with Azure AD easy):

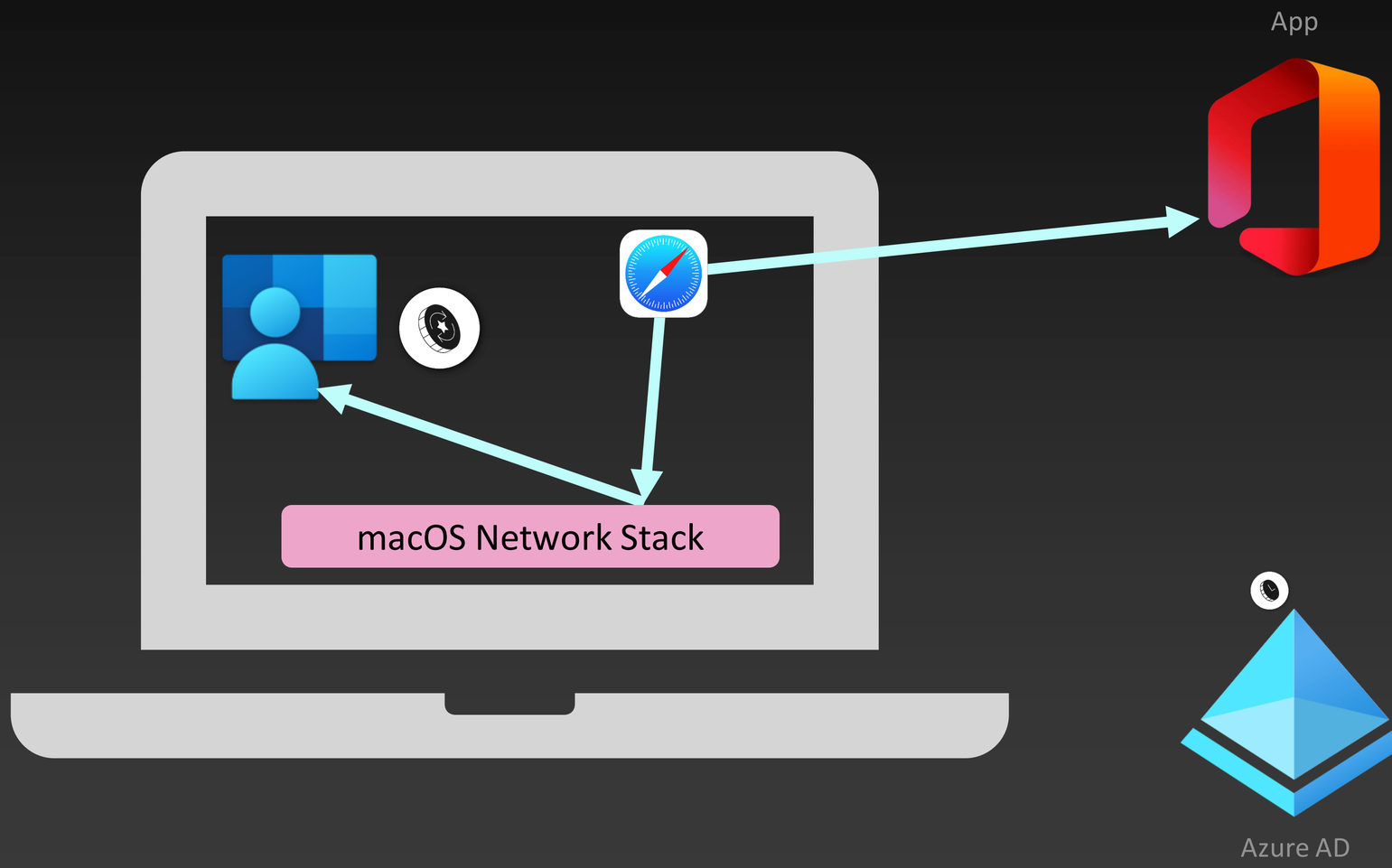
3. App that uses MSAL talks to the SSO Extension directly, asks it to get a token
4. AAD validates the PRT and returns the app-specific token
5. The token is given to the client and the client sends the token to the app
6. The user successfully accesses the app



Enterprise SSO Redirect flow

Now let’s look at the redirect flow:

3. User tries to log into app, is told to get a token from Azure AD
4. App that doesn’t use MSAL tries to go to an Azure AD URL…the macOS Network Stack intercepts the traffic and redirects it to the SSO Extension
5. SSO Extension uses its PRT to request a token
6. AAD validates the PRT and returns the app-specific token
7. The token is given to the client and the client sends the token to the app
8. The user successfully accesses the app



Demo

Agenda

What is Azure AD?

Prompting…why is it bad?

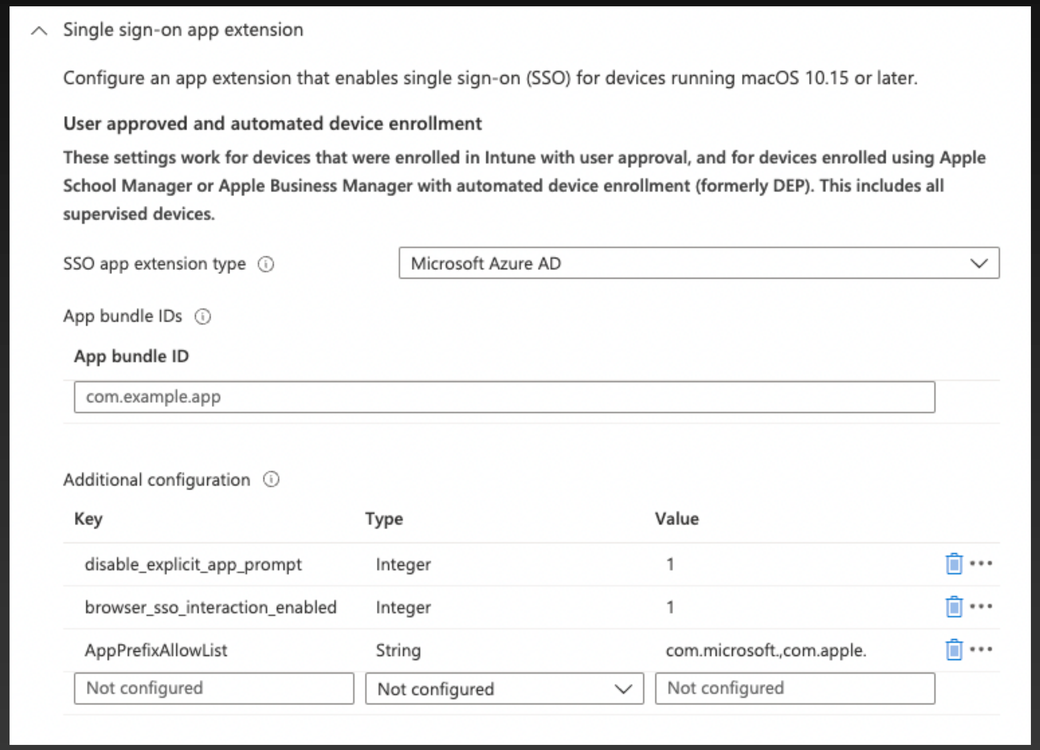
Enterprise Single Sign On (SSO) - How does it work?

Deploying Enterprise SSO

Troubleshooting Enterprise SSO

Deploying Enterprise SSO with Intune

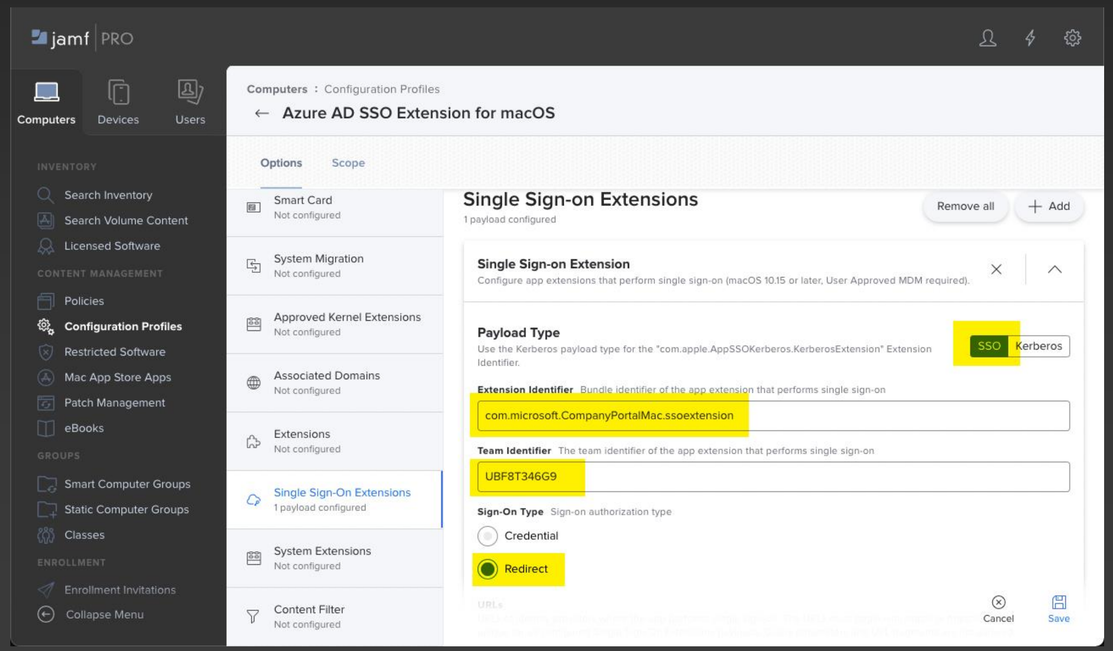
- Redirect SSO Extension Profiles *must* be deployed via MDM:
  - Very easy deployment if Intune is your MDM

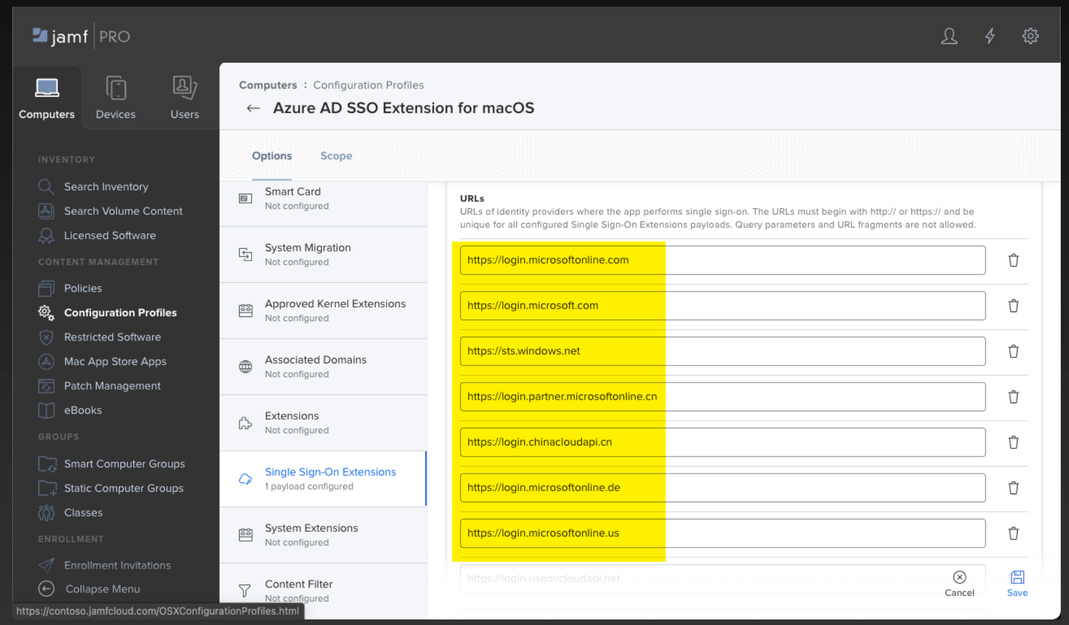


https://aka.ms/AppleSSO-Intune

Deploying Enterprise SSO with Jamf Pro

- Redirect SSO Extension Profiles *must* be deployed via MDM:
  - Jamf Pro config is quite straightforward with a PLIST file





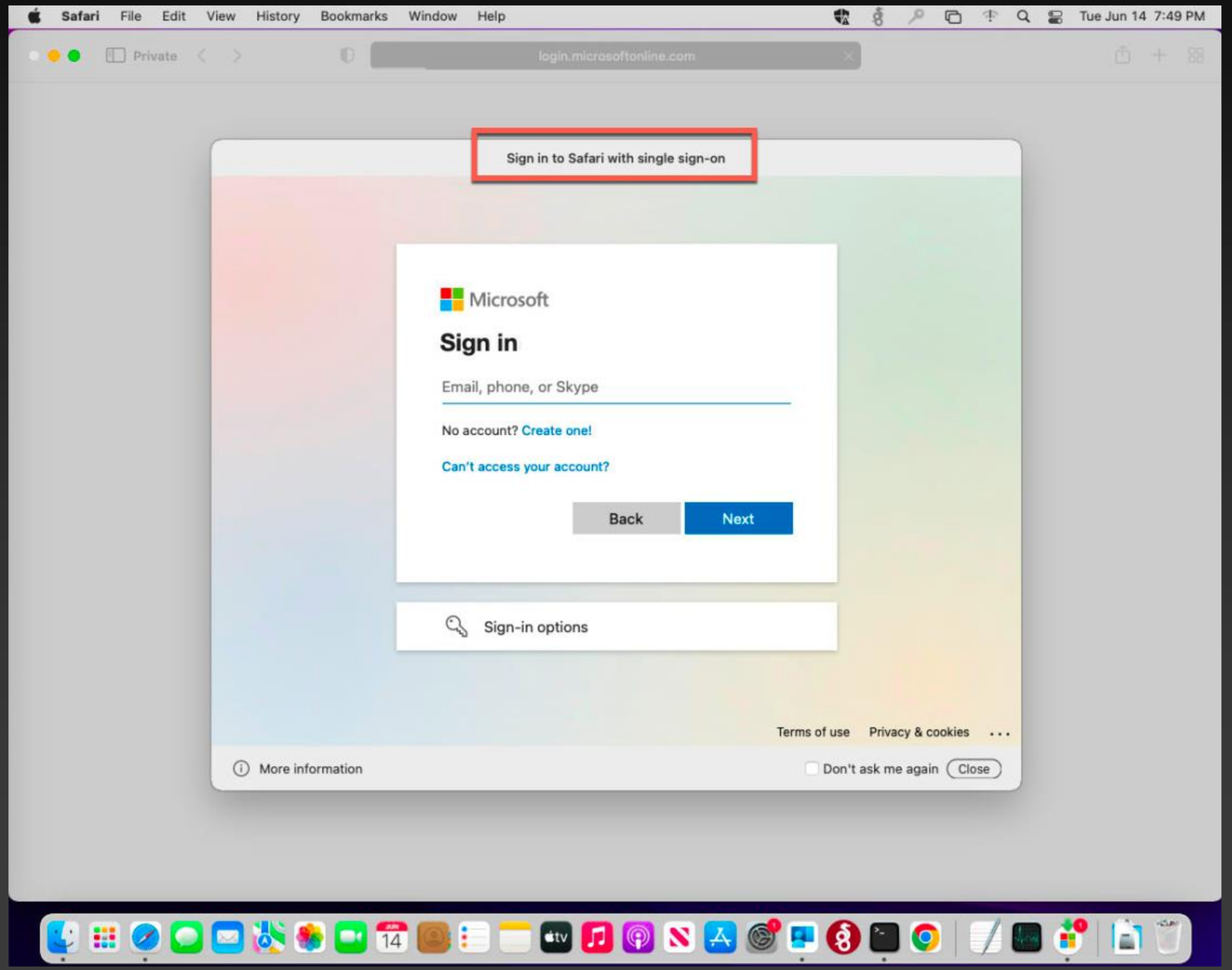
XML Copy

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/P
<plist version="1.0">
<dict>
    <key>AppPrefixAllowList</key>
    <string>com.microsoft.,com.apple.</string>
    <key>browser_sso_interaction_enabled</key>
    <integer>1</integer>
    <key>disable_explicit_app_prompt</key>
    <integer>1</integer>
</dict>
</plist>
```

https://aka.ms/AppleSSO-JamfPro

Recommendation 3: SSO – Modernize w/ IDP Vendor SSO Extensions

- Redirect SSO Extension Profiles *must* be deployed via MDM:
  - Out of the box support with Intune if it is your MDM
  - Jamf Pro config is quite straightforward with a PLIST file
  - Guidance provided for other MDM vendors
- *Can configure settings so users never need to open Company Portal*
  - Company Portal must always be installed, but users don't need to open it if you follow recommended config



Some things to keep in mind

There’s a few limitations/caveats/warnings:

- SSO Extension component from Microsoft is still Public Preview (supported)
- Apps must use MSAL or Apple’s system frameworks for network requests
  - This means that some apps don’t work…the SSO Extension is unaware of them and they don’t use Apple’s network stack
  - Chrome and Firefox are the primary examples
  - Talk to your app vendors about the need to support SSO extensions! They should want their apps to work, Apple is only making SSO extensions more important as time goes on
- No support for FIDO keys as a passwordless auth method in the SSO Extension window
  - Authenticator App Phone Sign-In passwordless mode works well

Agenda

What is Azure AD?

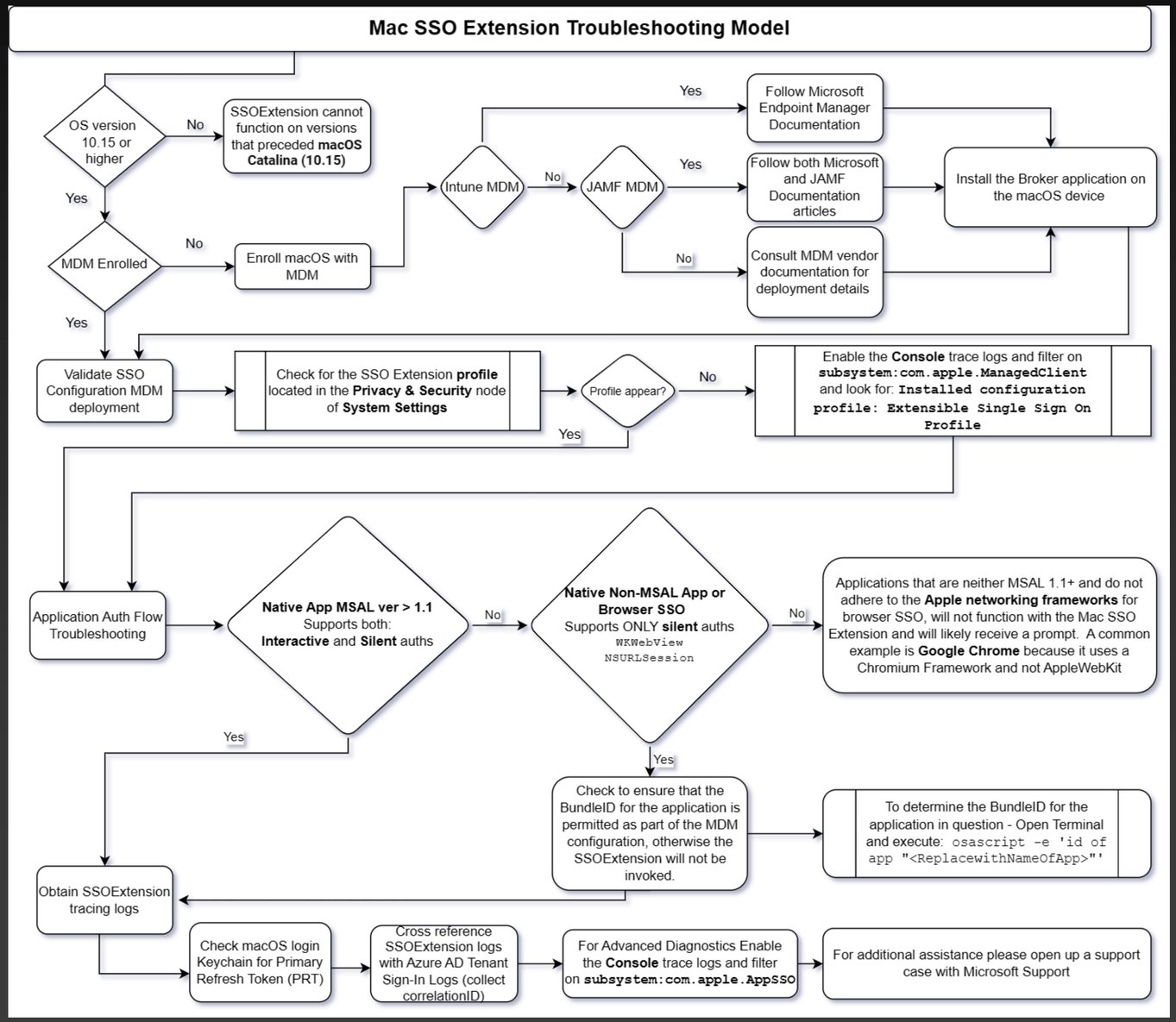
Prompting…why is it bad?

Enterprise Single Sign On (SSO) - How does it work?

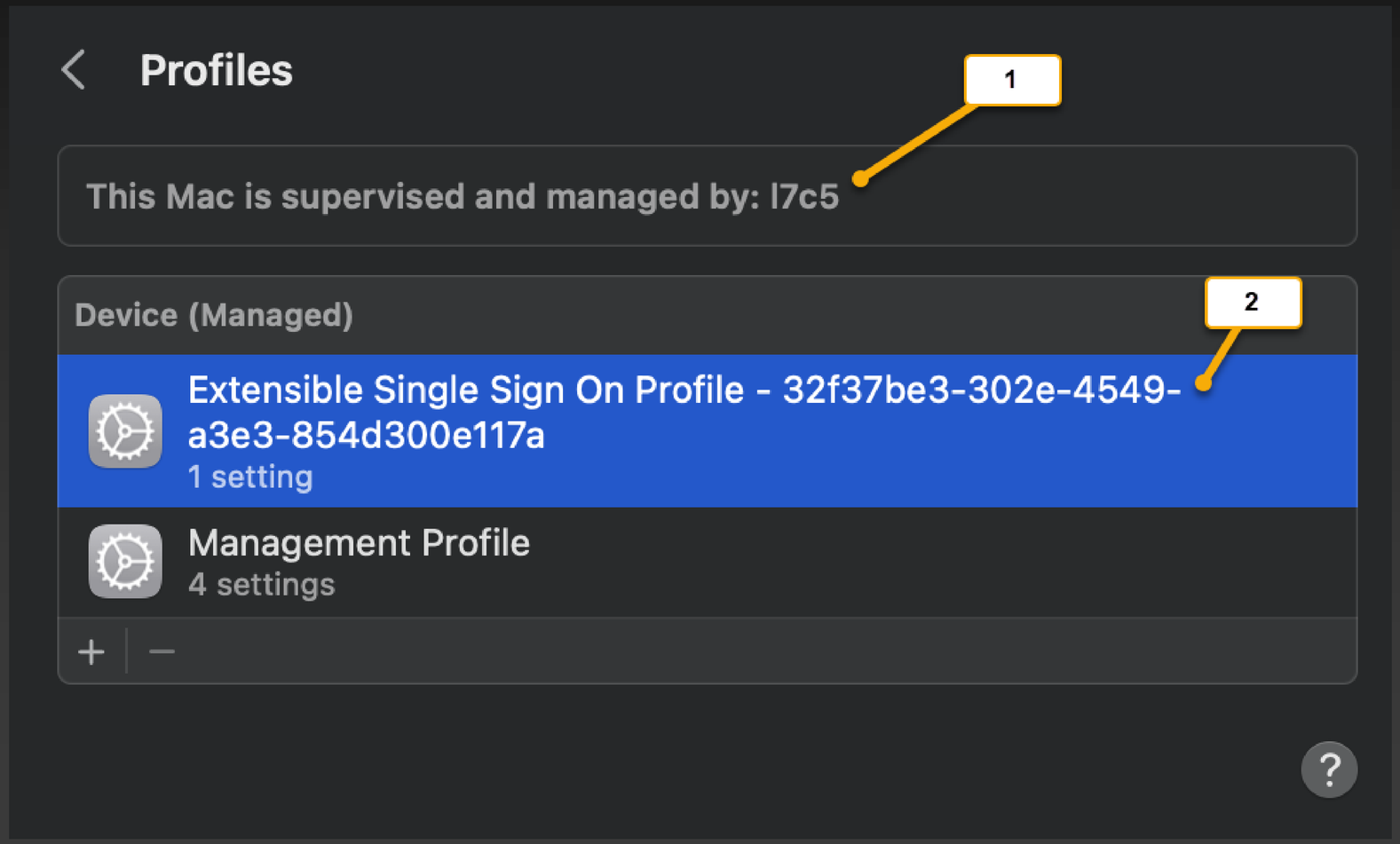
Deploying Enterprise SSO

Troubleshooting Enterprise SSO

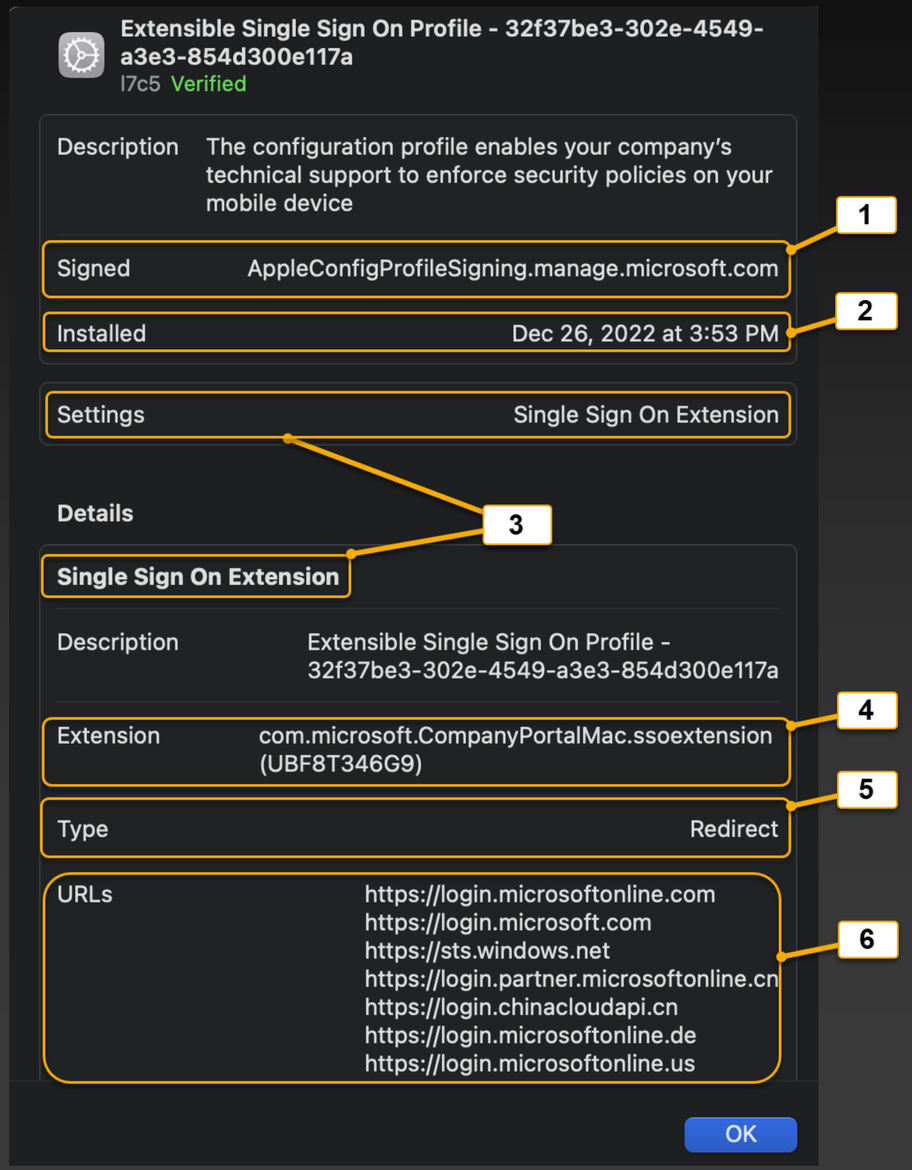
Troubleshooting Enterprise SSO



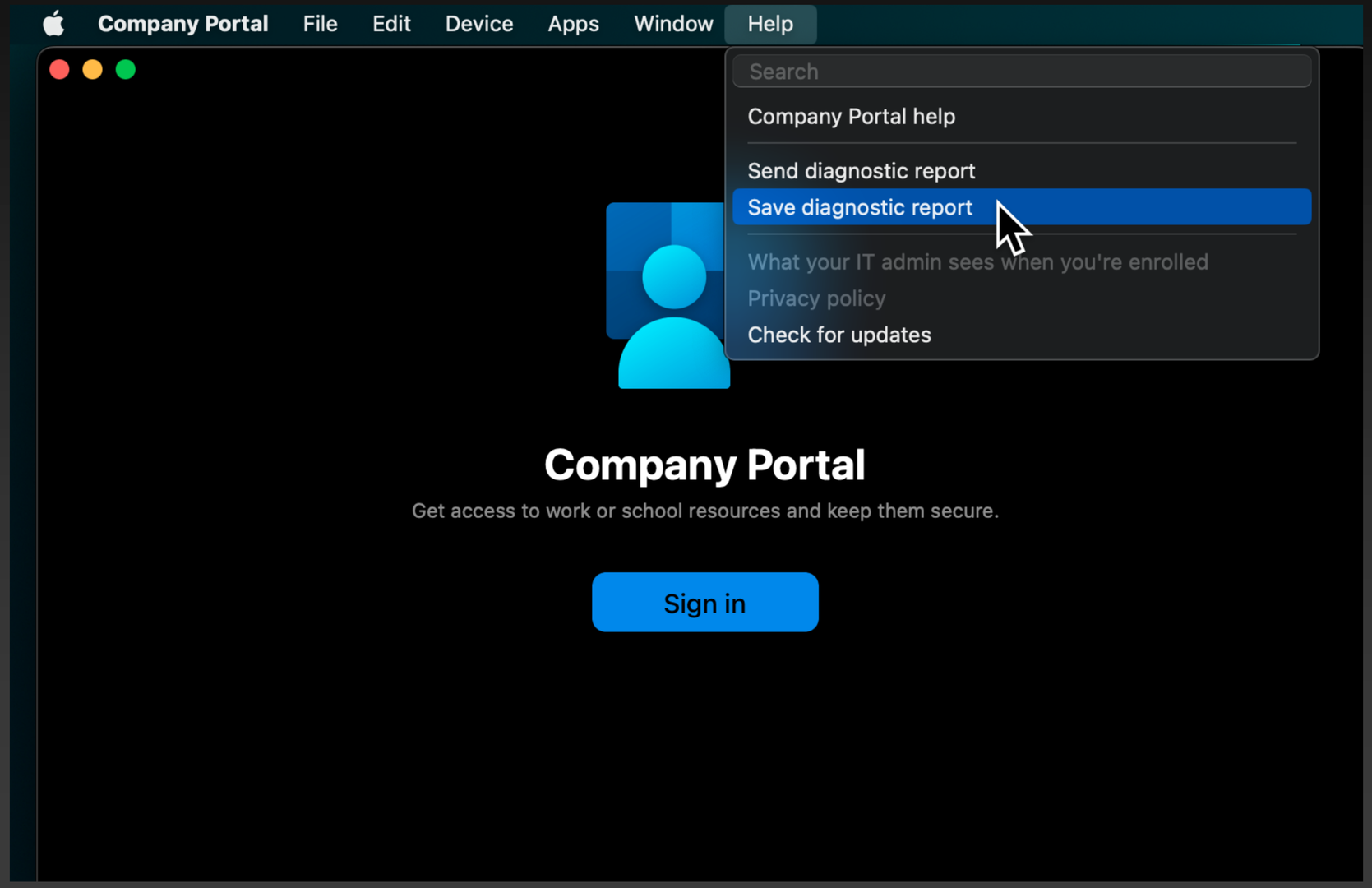
Locate SSO extension MDM profile



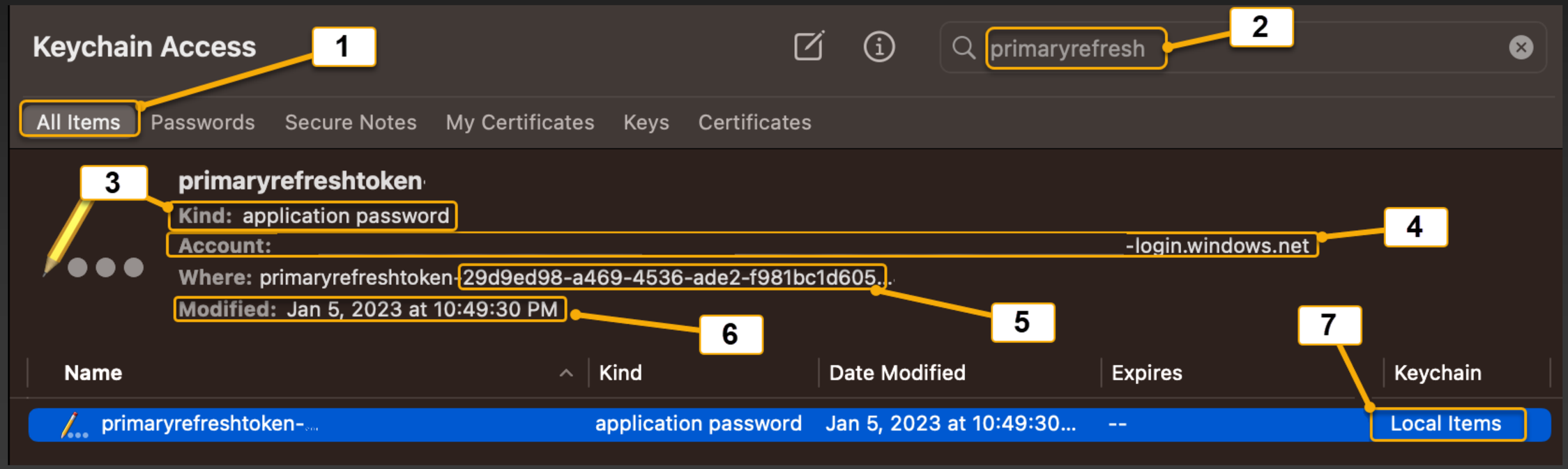
Verify the extension configuration



Collect Enterprise SSO logs



Check keychain access for PRT



Recap & Go Dos!

1. Work with your IAM/Security team on the end user experience
   - Use data in the Azure AD Authentication Prompt analysis (http://aka.ms/MFAPromptsWorkbook)
2. Set device compliance via an MDM
3. Deploy the Azure AD Enterprise SSO plugin to macOS *and iOS*
4. Nudge users to use the Microsoft Authenticator app on iOS/Android and start moving to passwordless
5. More SSO! Bring your modern auth apps to your IAM team. Move away from apps that require line of sight to a DC

Thank You

Slides: aka.ms/AADOBTS22