# Improving macOS Security by Reducing Authentication Prompts

# Intro

Michael Epping
Product Manager, Microsoft
michael.epping@microsoft.com
@_michaelepping

Mark Morowczynski
Product Manager, Microsoft
markmoro@microsoft.com
@markmorow

# Agenda

What is Azure AD and Conditional Access?

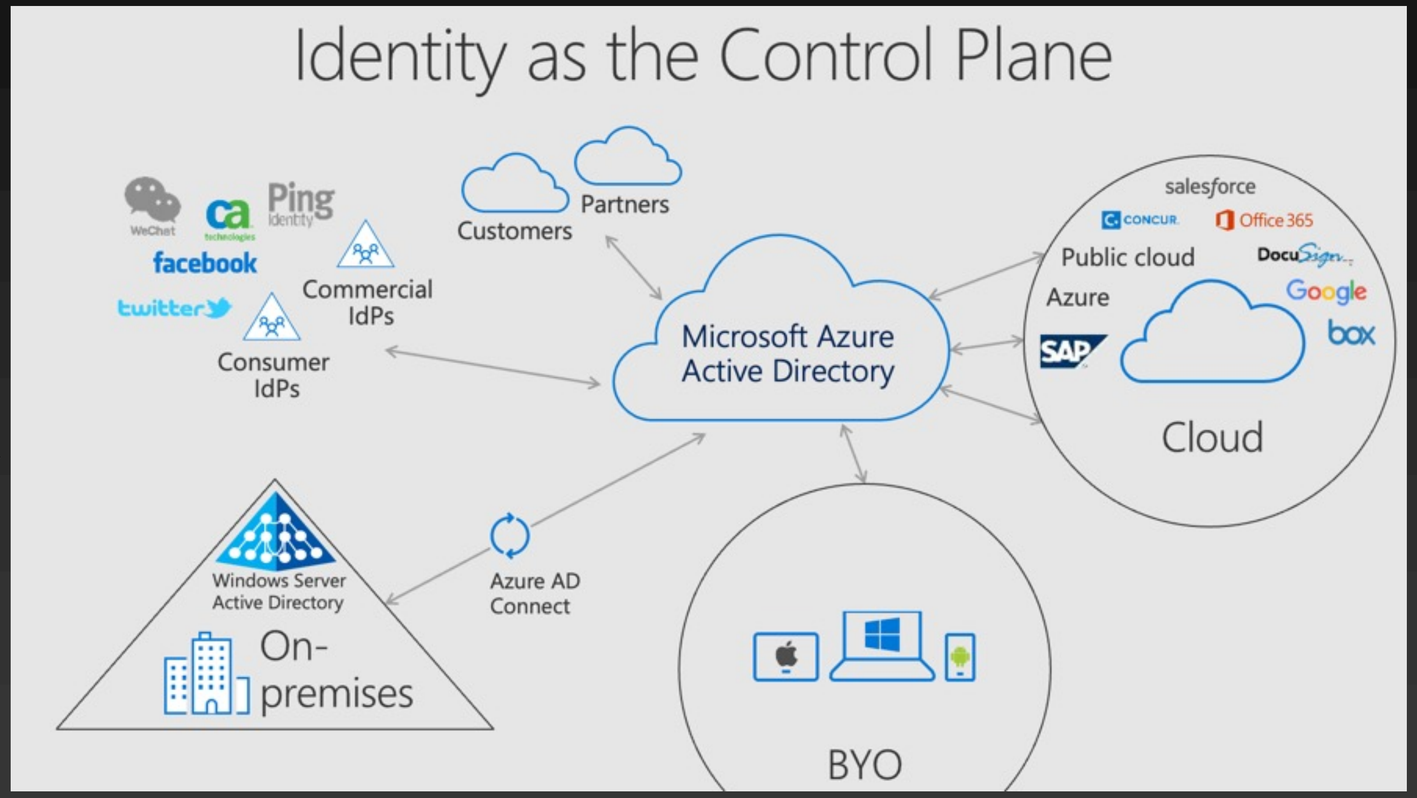
Prompting…why is it bad?

Top 5 Recommendations for the Enterprise

Go-Dos

# 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

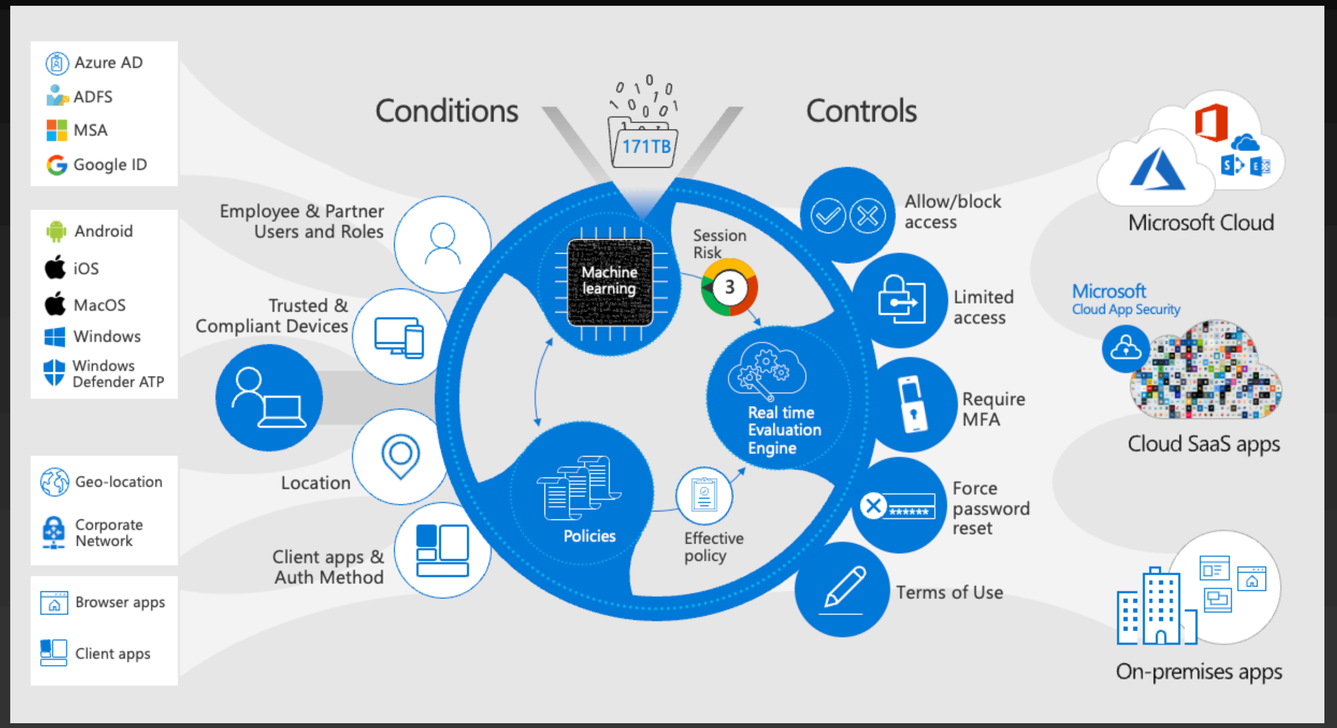


# Azure AD Protocols

- Committed to open standards, especially OpenID Connect and other modern protocols
- Microsoft cloud services are built on OpenID Connect
- Investing in new standards, like FIDO and DIF
  - See joint Passkeys announcement from FIDO foundation, Microsoft, Apple, and Google: https://aka.ms/PasskeyAnnouncement

# Conditional Access

- Zero-trust AuthN and AuthZ engine
  - Evaluate trust every time a user or device requests access to a resource
- Conditional access understands the user's activity
  - User location
  - User Risk
  - State of device
  - App requirements



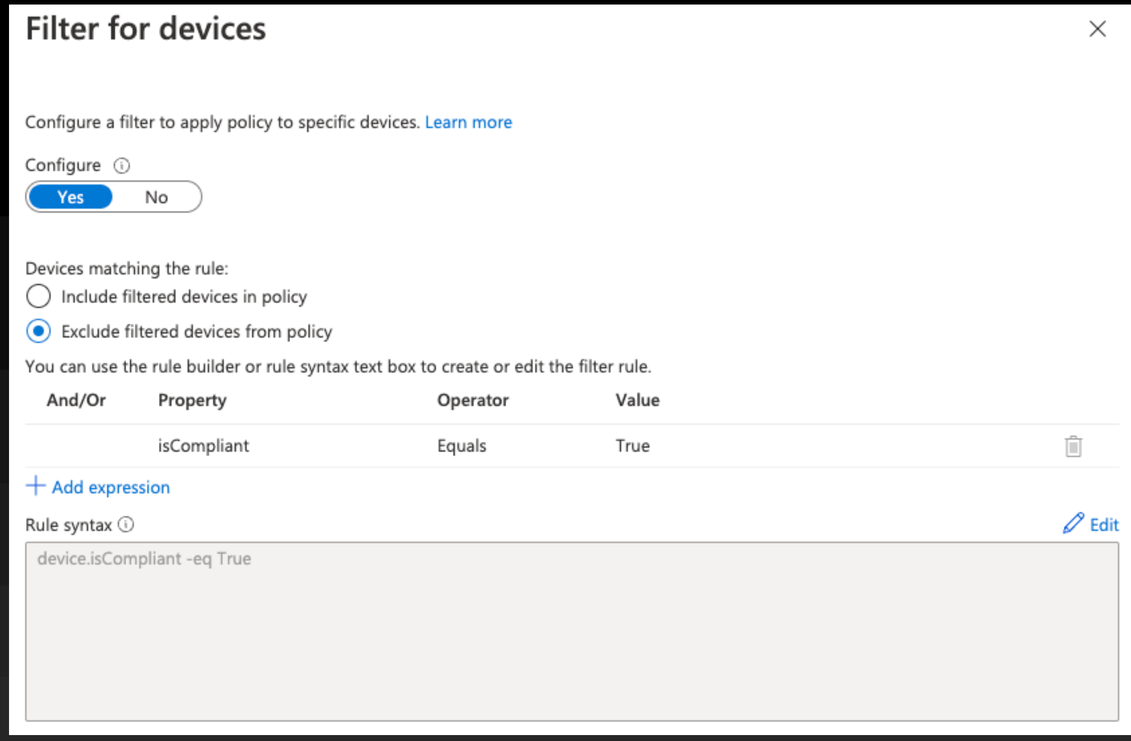
# Conditional Access Evaluation Phase

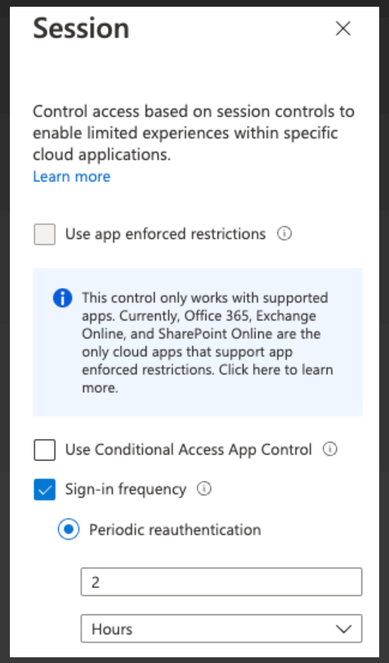
- All Conditional Access polices are ANDed together. (Not like GPO LSDO precedence)
  - Is Policy in scope of the request
  - BLOCK controls satisfied first
  - GRANT controls applied in order
    - Risk
    - MFA
    - Device
    - Approved client app/app protection
  - Tries to satisfy policy without user interaction
    - Example: Control MFA or Device compliant. If device is NOT compliant, will THEN prompt for MFA.


```
{
  "userDisplayName": "Michael Epping",
  "appDisplayName": "Azure Portal",
  "ipAddress": "97.113.39.216",
  "clientAppUsed": "Browser",
  "conditionalAccessStatus": "success",
  "riskDetail": "none",
  "riskLevelAggregated": "none",
  "riskLevelDuringSignIn": "none",
  "riskState": "none",
  "resourceDisplayName": "Windows Azure Service Management API",
  "deviceDetail": {
    "deviceId": "",
    "displayName": "",
    "operatingSystem": "MacOs",
    "browser": "Edge 102.0.1245",
    "isCompliant": false,
    "isManaged": false,
    "trustType": ""
  },
  "location": {
    "city": "Seattle",
    "state": "Washington",
    "countryOrRegion": "US",
    "geoCoordinates": {
      "altitude": null,
      "latitude": 47.61837,
      "longitude": -122.3142
    }
  }
}
```


# Common Policies

- Talk to your IAM team to understand your Conditional Access policies
- Requiring MFA for all users
- Blocking legacy auth
- Blocking access by country location
- Require compliant or hybrid join device
- Stricter Controls for non-corp managed devices (is this macOS in your environment?)
  - Sign-In Frequency to 2 hours for everything not filtered out
  - "Good" for security, but...



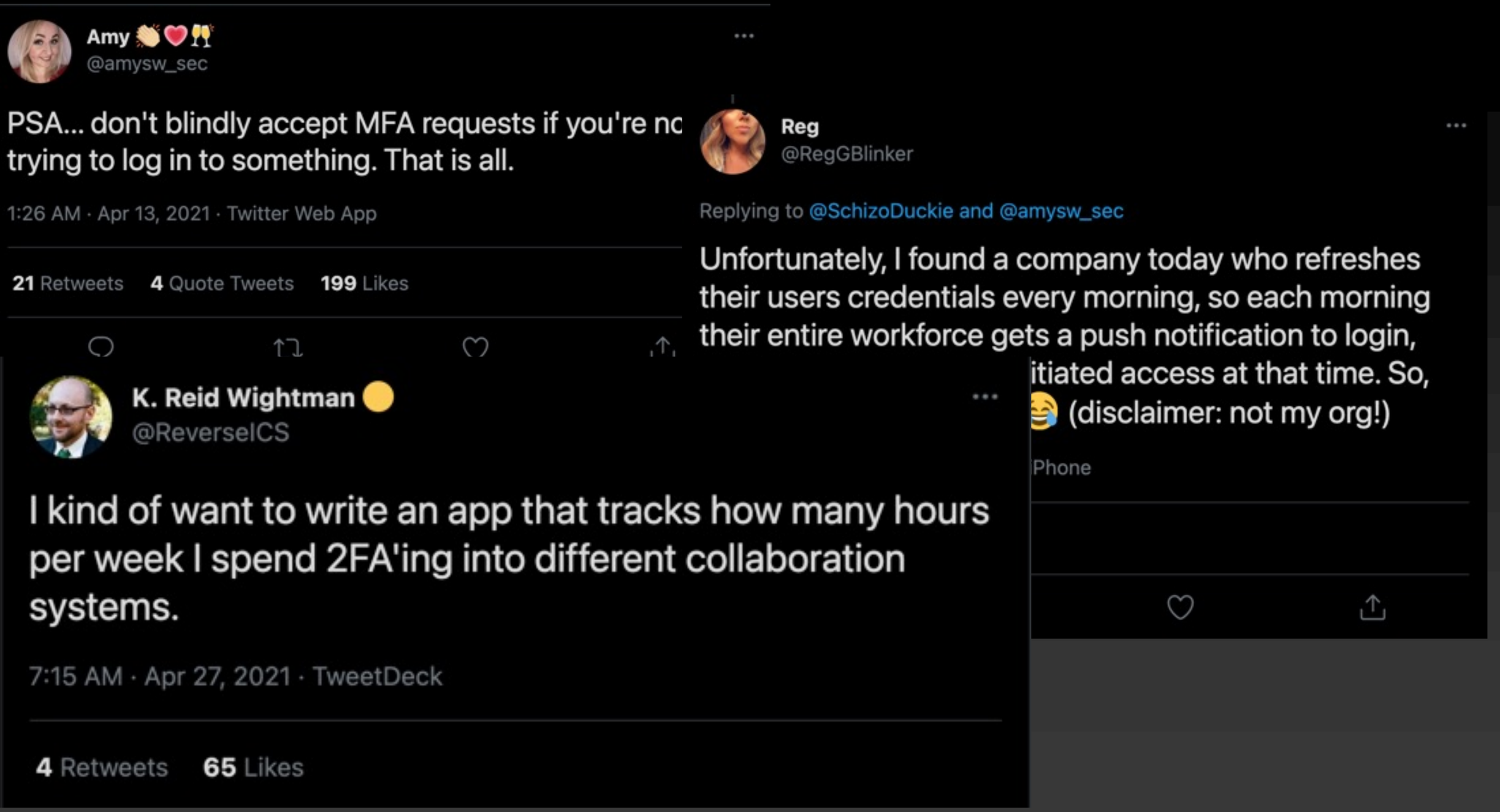
# Agenda

What is Azure AD and Conditional Access?

Prompting…why is it bad?

Top 5 Recommendations for the Enterprise

Go-Dos



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.

# 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

# Agenda

What is Azure AD and Conditional Access?

Prompting…why is it bad?

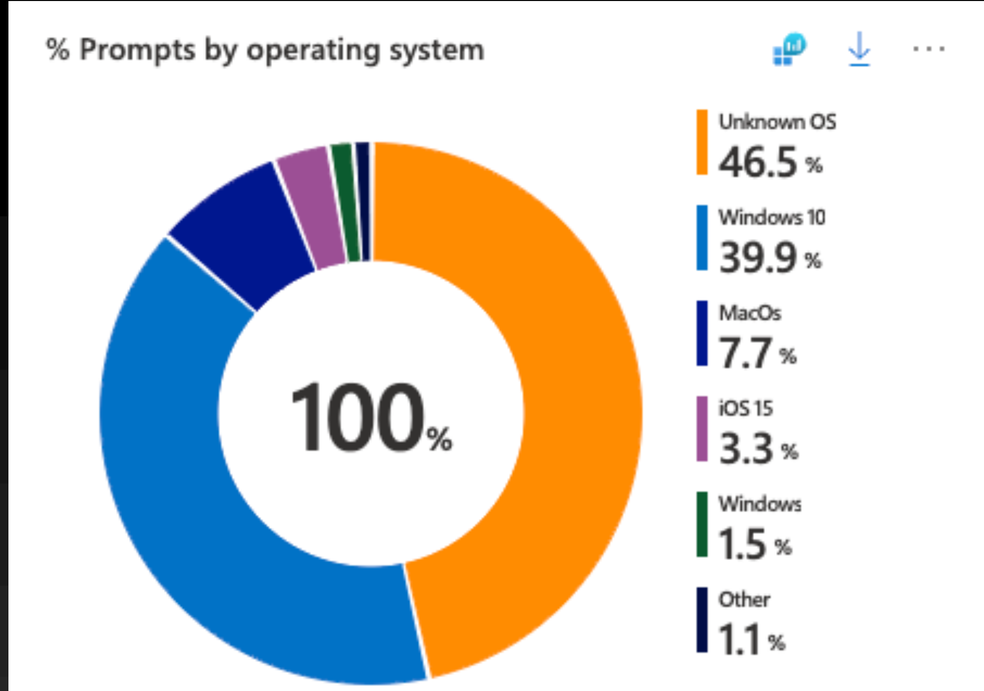
Top 5 Recommendations for the Enterprise

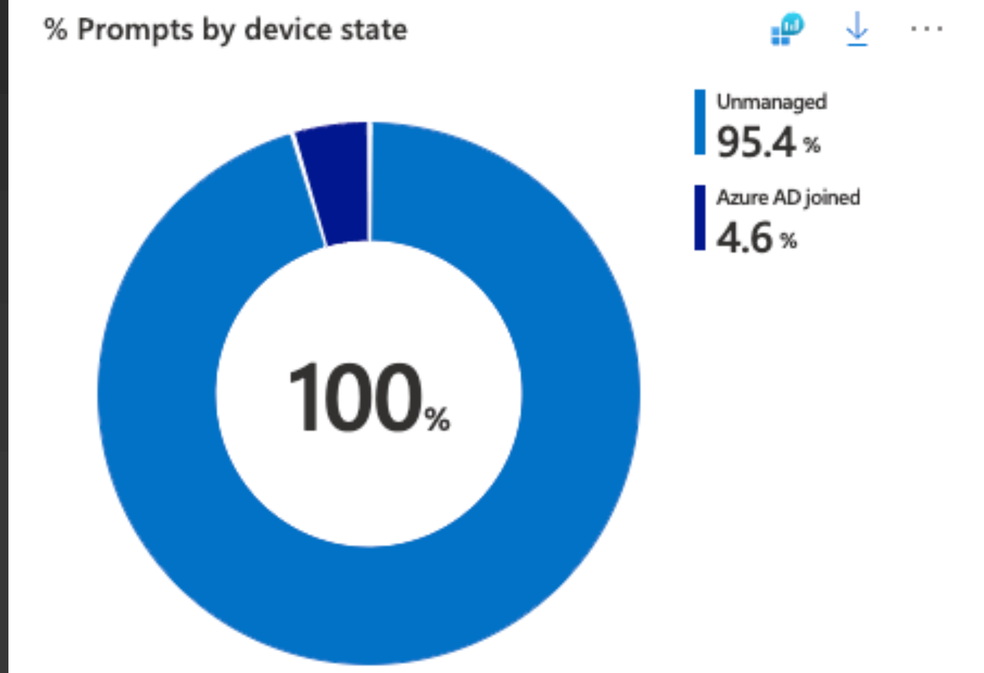
Go-Dos

# Recommendation 1: Determine if you have a prompting problem

## Show it with data!

- All the data you need is in the Azure AD sign-in logs
- Use the pre-built Azure AD Workbook http://aka.ms/MFAPromptsWorkbook
- Comes with data visualizations as well as recommendations:
  - Which users are being prompted the most?
  - Which applications have a high prompt count?
  - What is the device state?



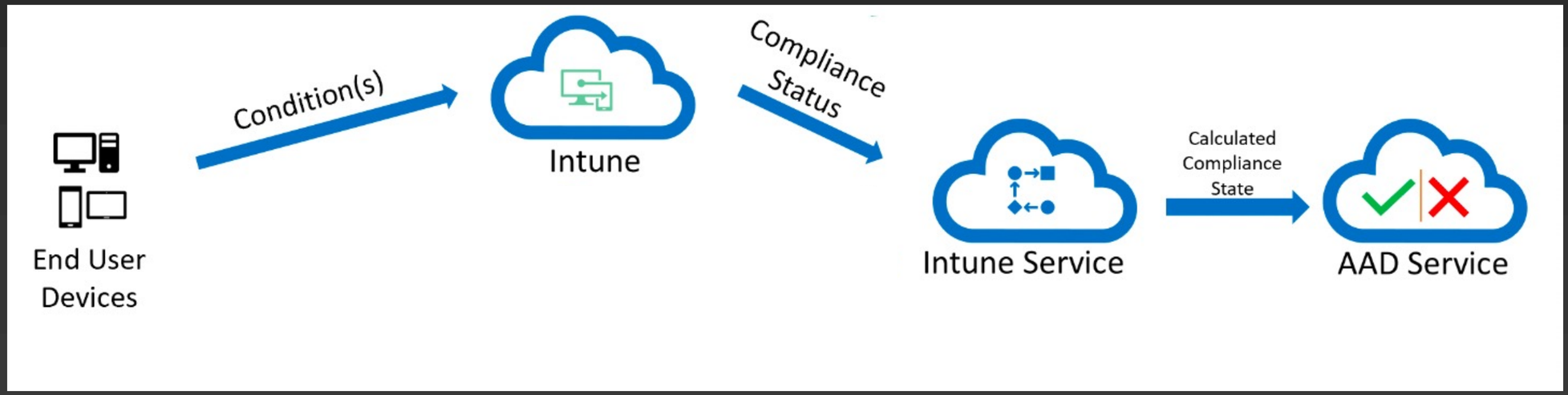



# Recommendation 2: Enroll in MDM, Use Device Compliance

- MDM is the only *modern* way to deploy enterprise features to macOS
  - MDM helps us improve device and identity security (Conditional Access)
  - SSO helps us improve end-user experience (fewer prompts) and security (over-prompting trains users to make poor decisions)
  - These are *related*, but *different* features
- Microsoft Endpoint Manager (MEM) or MEM-Integrated MDMs can send compliance information to Azure AD
  - This information is critical for those device-based Conditional Access policies
- Without MEM or an MEM-integrated MDM, Azure AD sees all Macs as unmanaged

# Recommendation 2: Enroll in MDM, Use Device Compliance

- Device health and compliance integration with Azure AD is easy to deploy if MEM is the MDM
- Jamf Pro and other 3rd Party MDMs can integrate with Intune to support device compliance
  - Extra work, but *worth it*



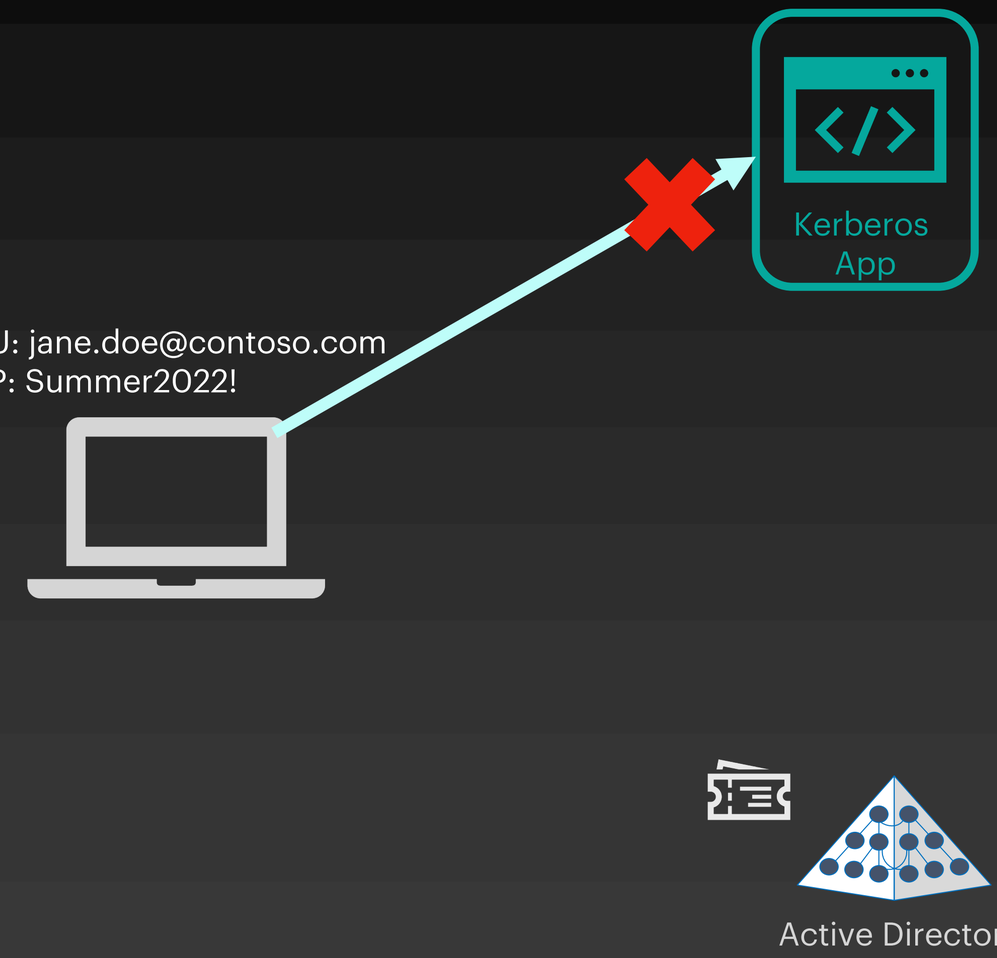
# Recommendation 3: Set up SSO Infrastructure

- 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

# Recommendation 3: SSO Infrastructure - Let's Start with Kerberos

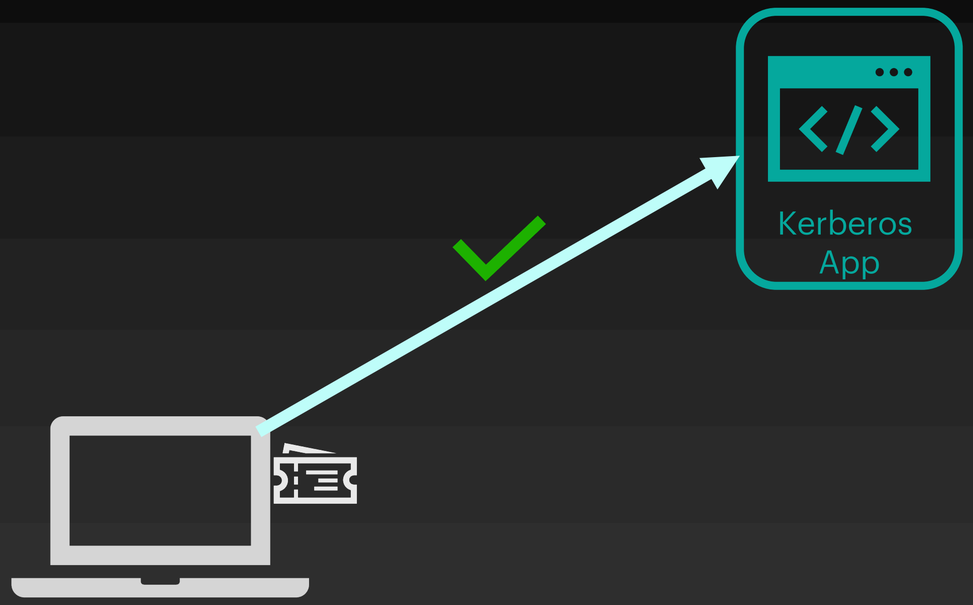
*If* you need Kerberos, use the modern, MDM-provisioned Kerberos SSO Extension from Apple:

1) User provides device with their enterprise username and password
2) The device sends the creds to AD and asks for a Kerberos Ticket-Granting Ticket (TGT)
3) AD validates the creds and returns the TGT
4) The user tries to access an app, probably in their browser, but needs a Kerberos ticket



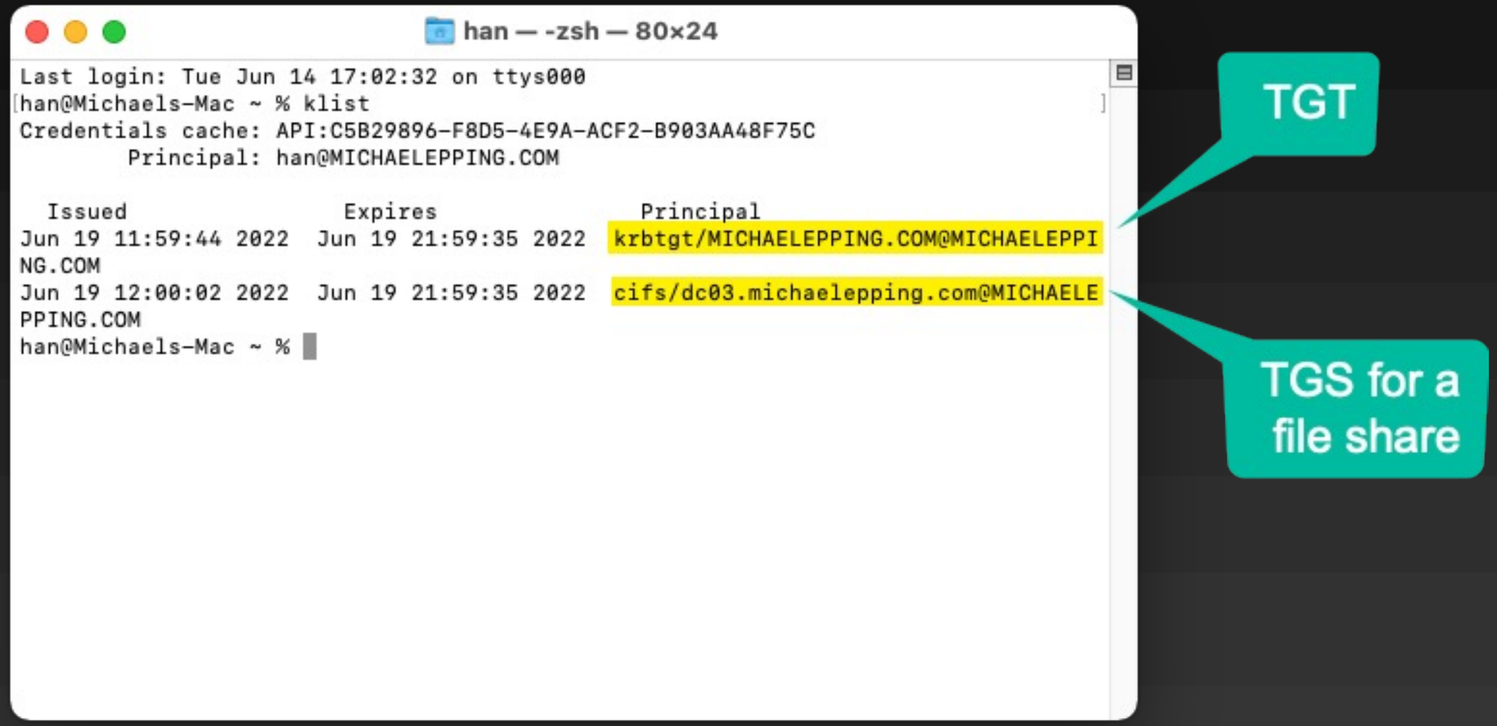
# Recommendation 3: SSO Infrastructure - Let's Start with Kerberos

5) macOS sends the TGT to AD, asking for a ticket specific to the app (TGS)
6) AD validates the TGT and returns the TGS
7) The user's browser or other client sends the TGS to the app
8) The user successfully accesses the app




Active Directory

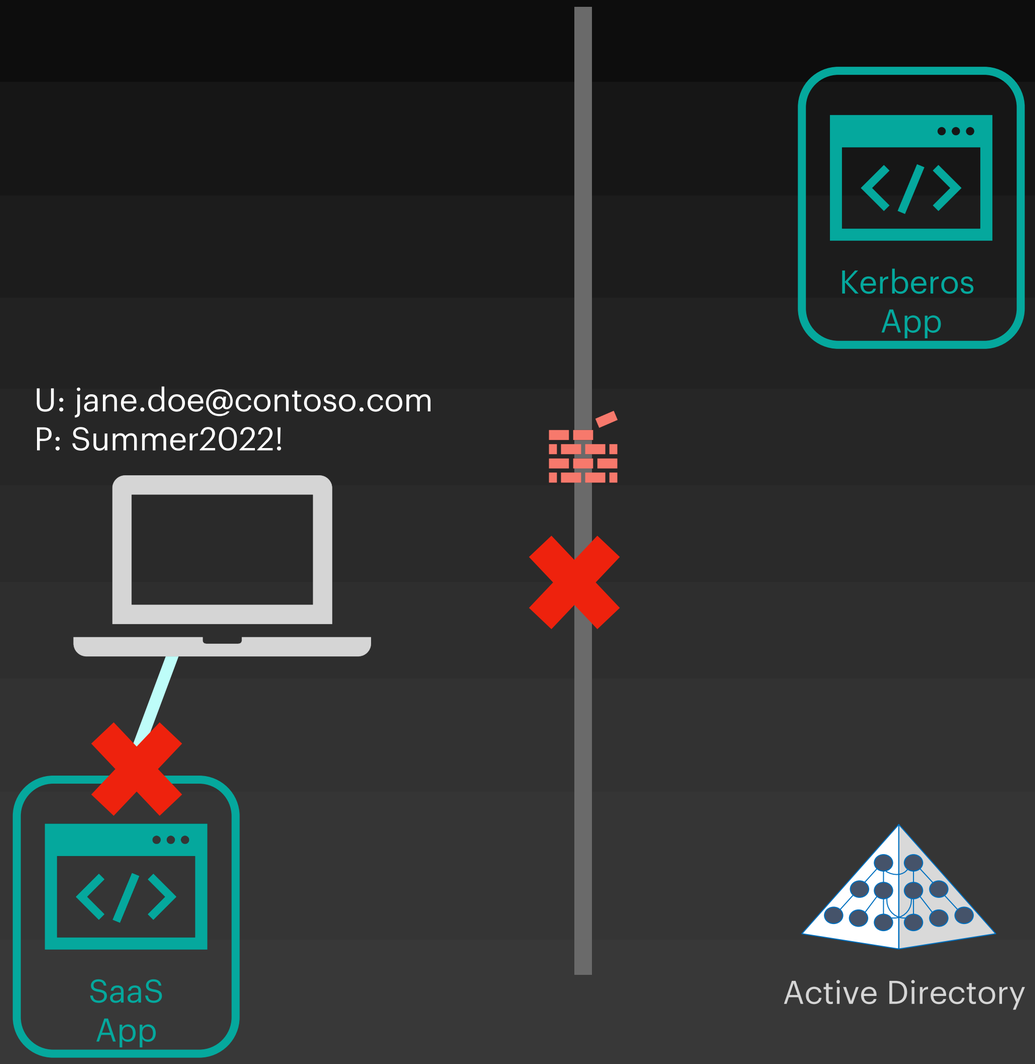
# Recommendation 3: SSO Infrastructure - Let's Start with Kerberos



# Recommendation 3: SSO - Let's Start with Kerberos

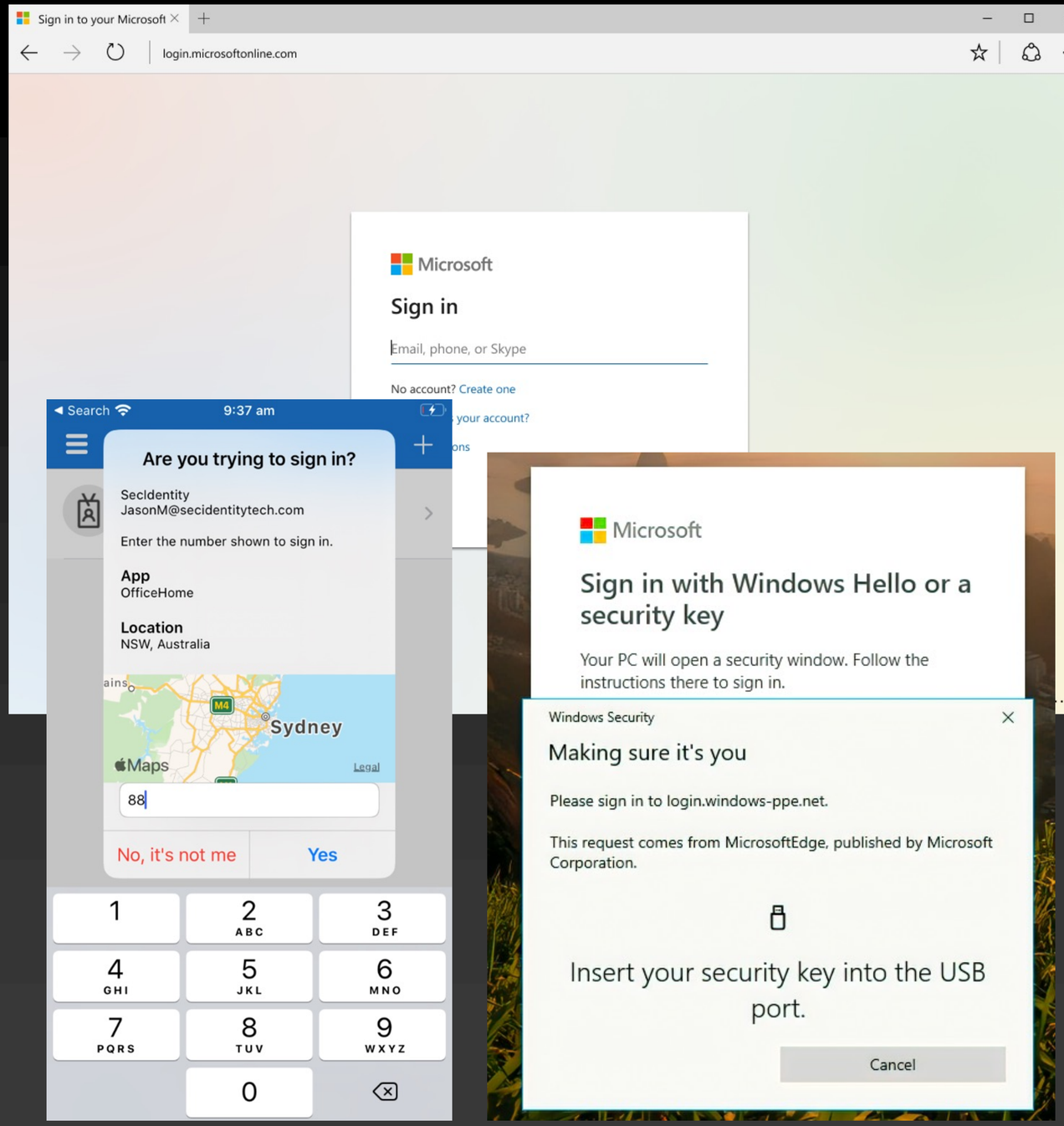
- What's the issues with this story?
- It doesn't work over the internet, so it isn't very modern
- Imagine we have a SaaS app instead of an internal Kerberos app
- Kerberos doesn't make sense for the SaaS app, because devices on the internet shouldn't be able to find a DC

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

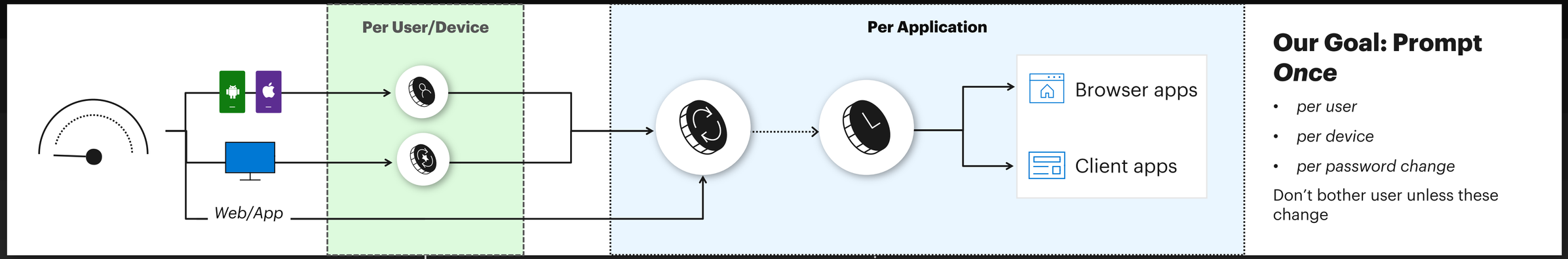


# Recommendation 3: SSO – Modernize w/ Modern Auth

- 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!



# Recommendation 3: SSO – Modernize w/ Modern Auth




## 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)

**Until revoked or Password Changed**

Long term access to an application

**Note:** Includes whether MFA was used for authentication

### 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)

# Recommendation 3: 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

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

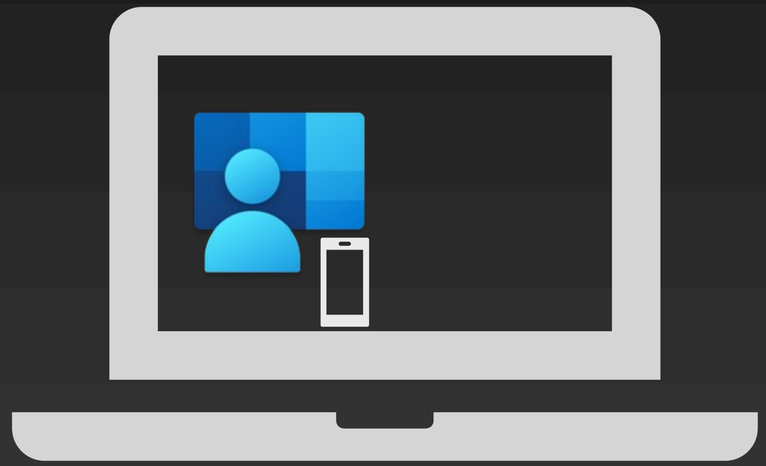
- The *modern* approach is to use an IDP, modern auth, and tokens
- SSO Extension is bundled in the Microsoft Company Portal

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



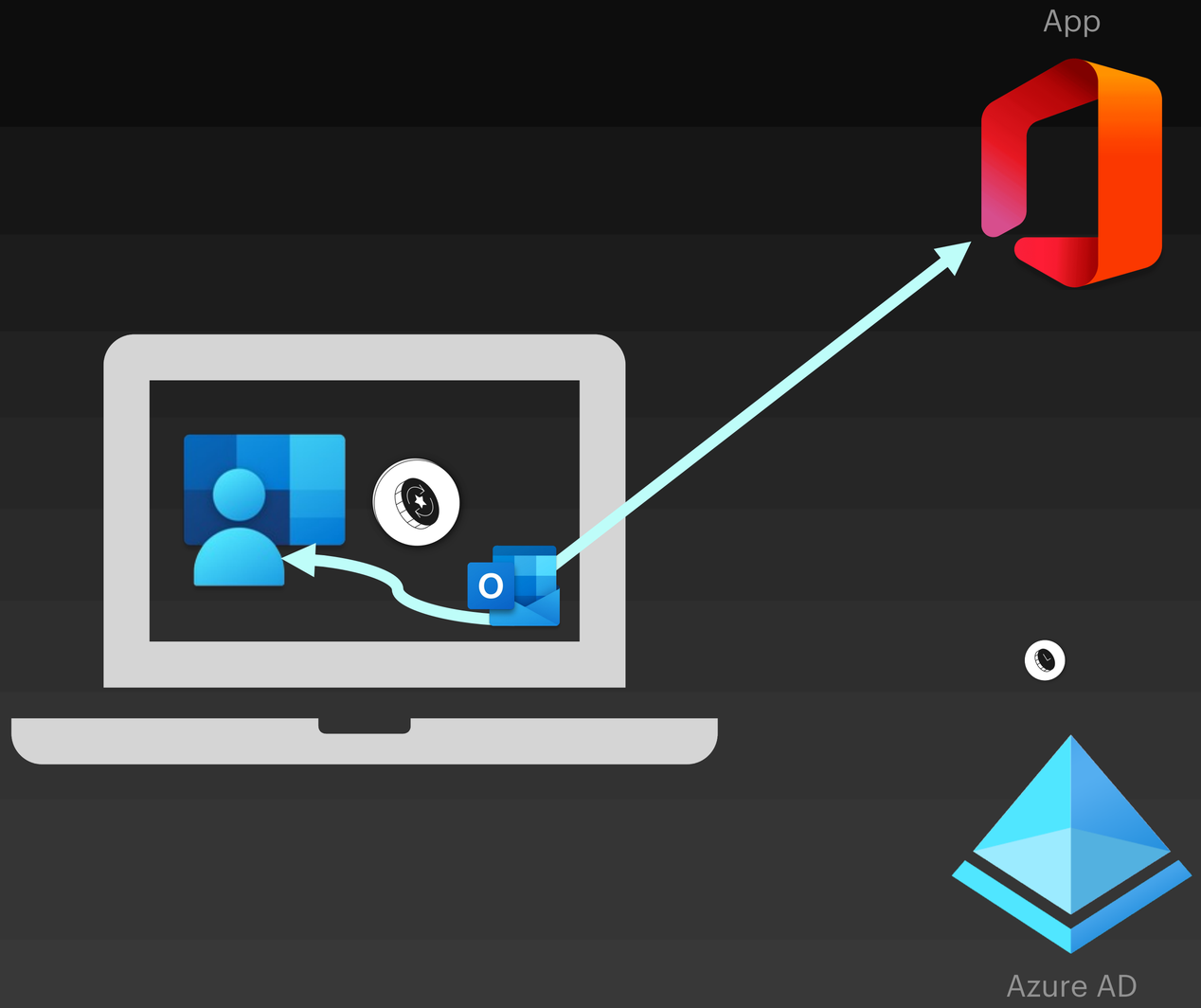
Azure AD

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

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):

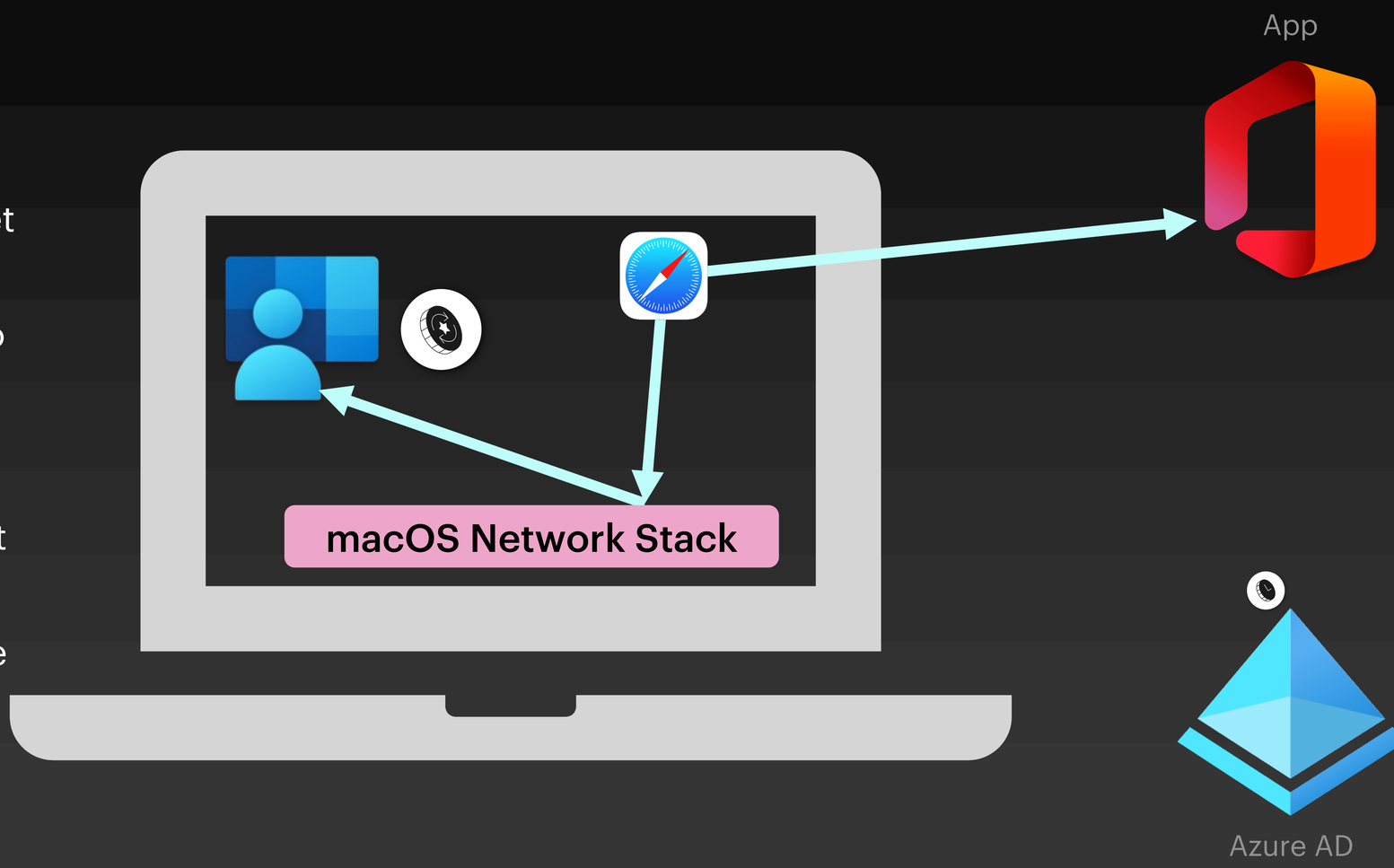
3. App that uses MSAL talks to the SSO Extension directly, asks it to get a token
4. AAD validates the PRT and returns the app-specific token
5. The token is given to the client and the client sends the token to the app
6. The user successfully accesses the app



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

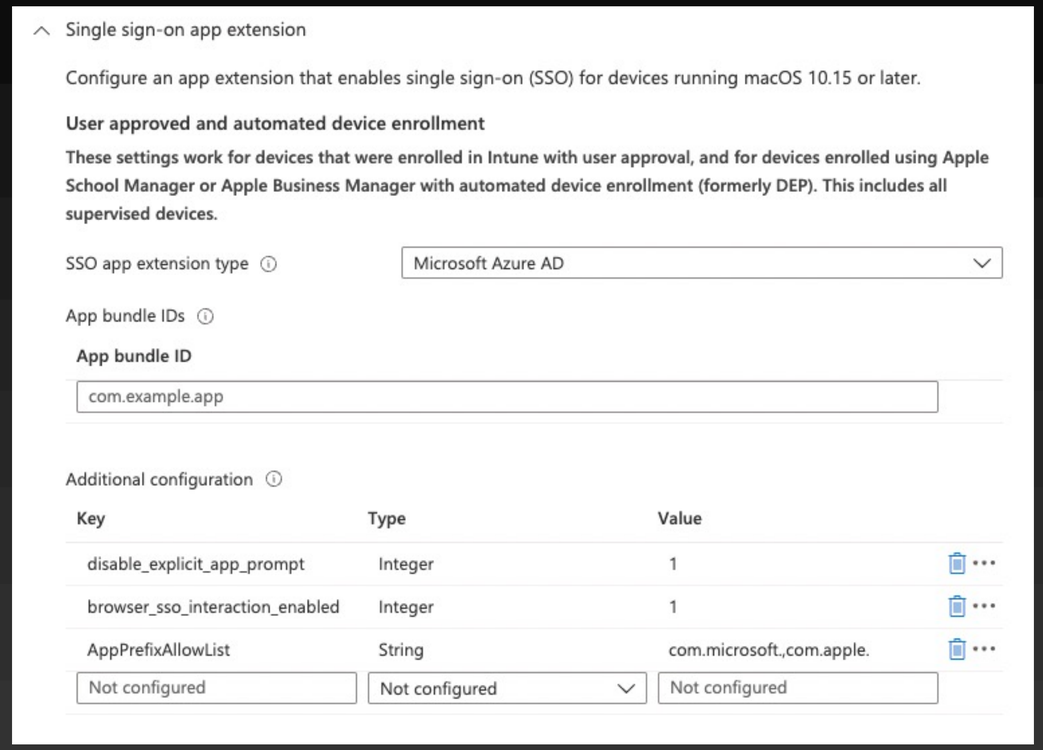
Now let's look at the redirect flow:

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



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

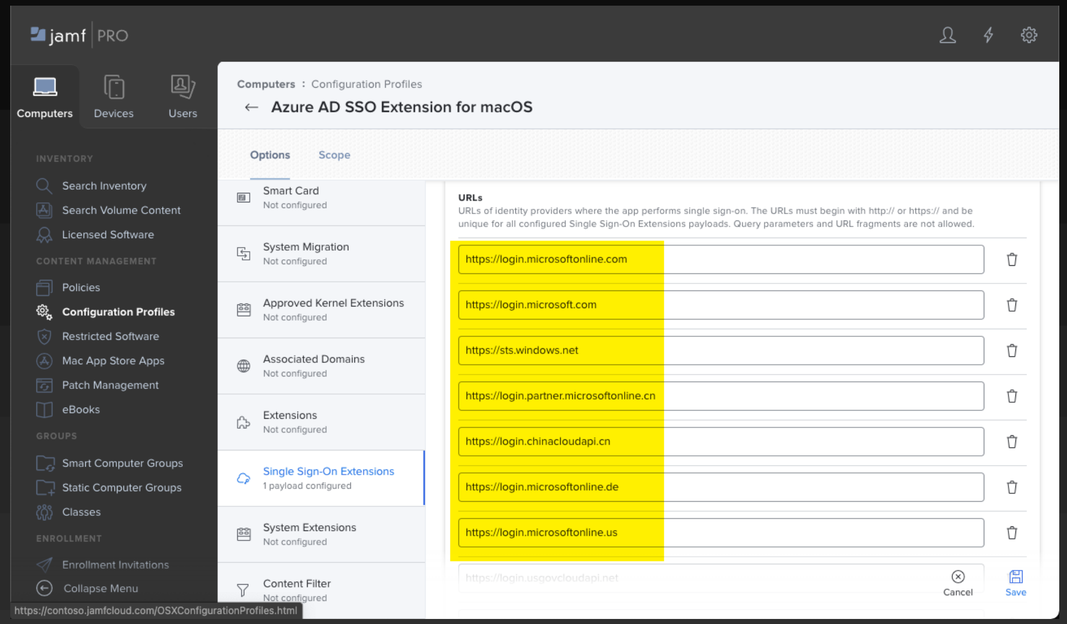
- Redirect SSO Extension Profiles *must* be deployed via MDM:
  - Very easy deployment with MEM as your MDM

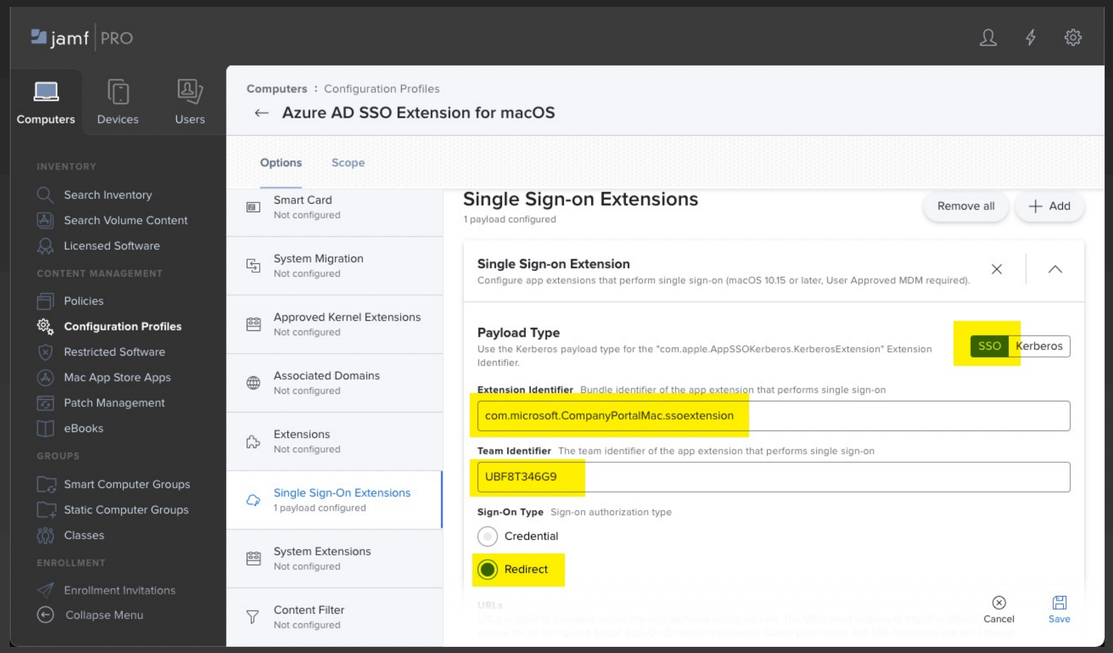



https://aka.ms/AppleSSO-Intune

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

- Redirect SSO Extension Profiles *must* be deployed via MDM:
  - Very easy deployment with MEM as your MDM
  - Jamf Pro requires a little more work and a PLIST file



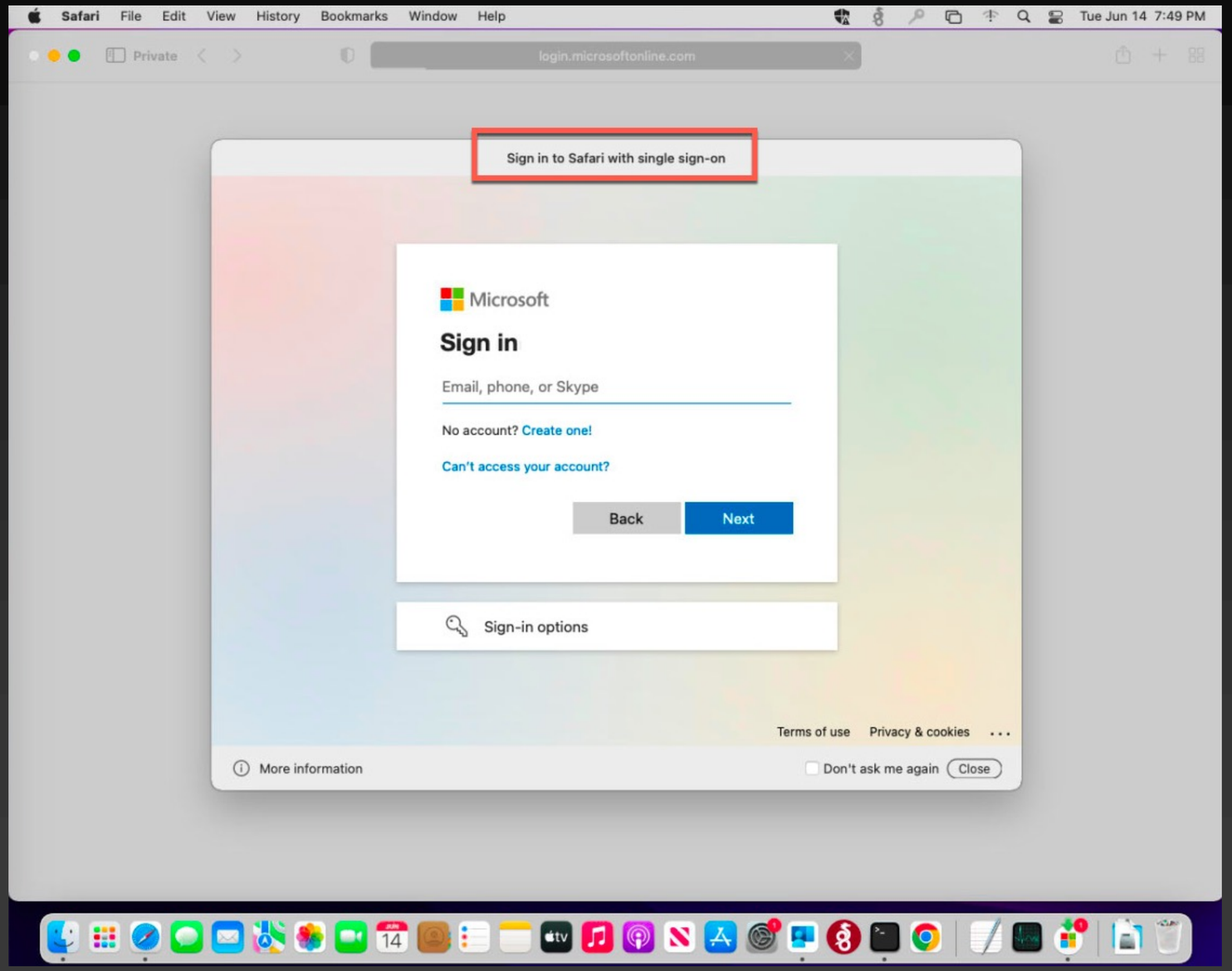

XML Copy

```
<?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:
  - Very easy deployment with Intune as your MDM
  - Jamf Pro requires a little more work and a PLIST file
- *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



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

There’s a few limitations/caveats/warnings:

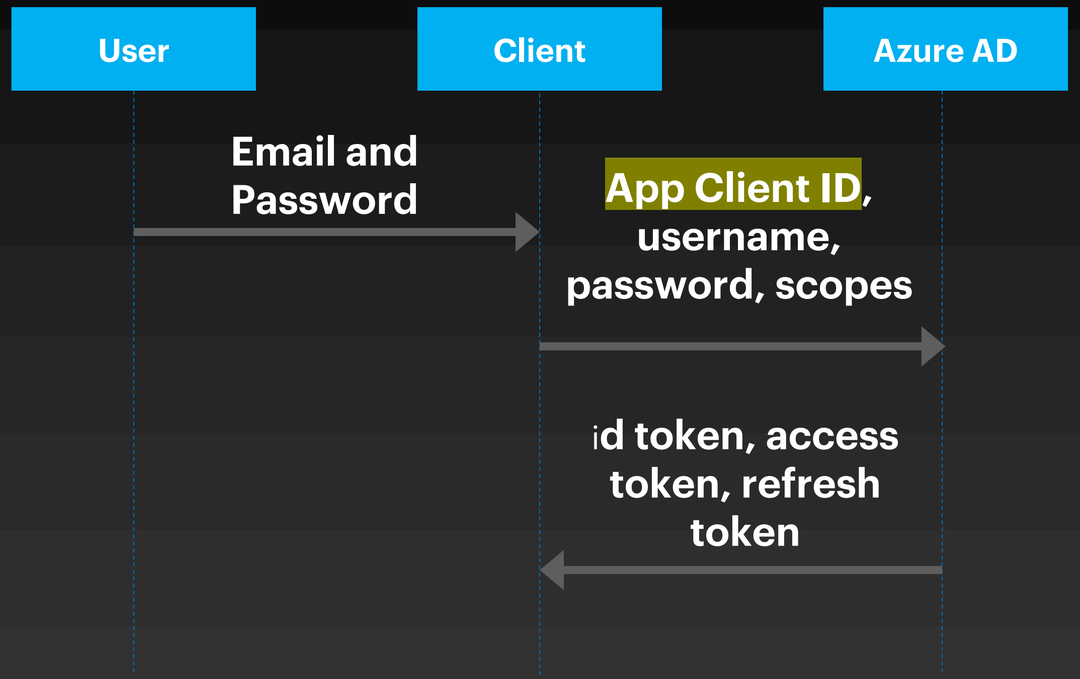
- SSO Extension component from Microsoft is still Public Preview (supported)
- Apps must use MSAL or Apple’s system frameworks for network requests
  - This means that some apps don’t work…the SSO Extension is unaware of them and they don’t use Apple’s network stack
  - Chrome and Firefox are the primary 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
  - More on Passwordless next…

# Quick Aside on Jamf Connect and Similar Tools

- We hear from a lot of customers using Jamf Connect and similar tools, instead of BIND or Apple's Kerberos extension
  - These offer lots of features that Apple's Kerberos features do not
  - IDP sign-in from the lock screen, including MFA, is a big one
- Few common issues to be aware of:
  - Can't check for device compliance from the lock screen
  - Going to see sign-in failures in the Azure AD logs if you have Conditional Access policies that target "All Cloud Apps" – this is due to how the OAuth 2.0 Resource Owner Password Credentials (ROPC) flow works
    - Jamf docs refer to this as ROPG

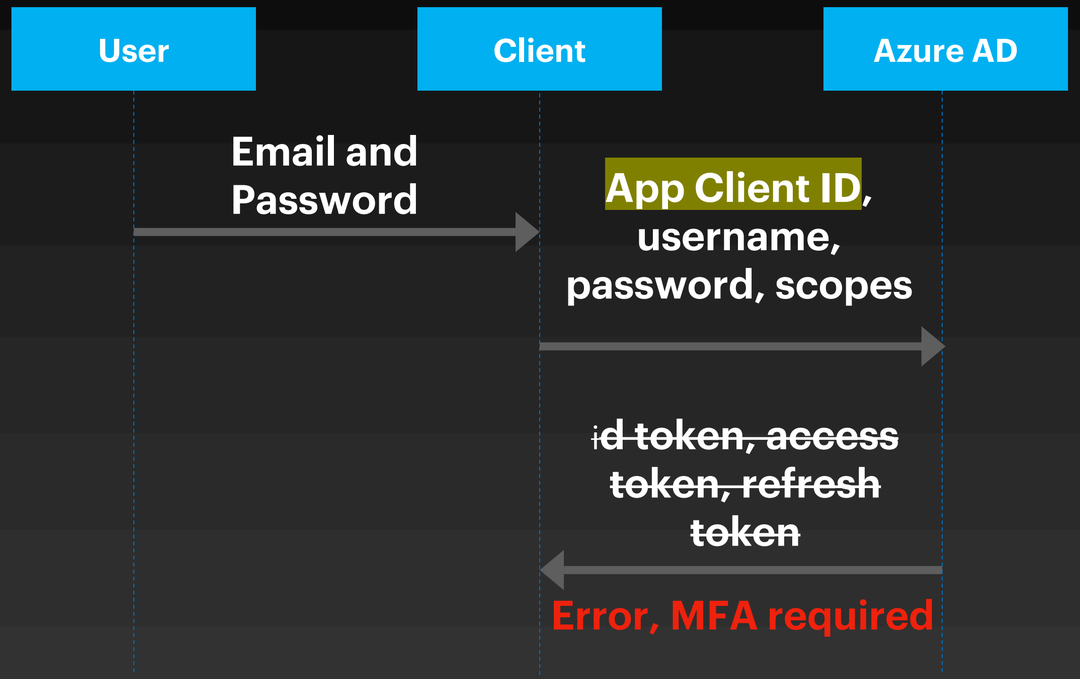
# Quick Aside on Jamf Connect and Similar Tools

- ROPC can be useful, but is the least preferred OAuth flow
  - Not interactive auth, the client application simply sends an http POST
  - Therefore, Azure AD can't interactively challenge for additional factors, such as MFA
  - Username and password are exposed to the client app – usually frowned upon!
- ROPC requires you register an Application with Azure AD
  - "All Cloud Apps" Conditional Access policies will apply to requests against this app – end result will be no tokens, as MFA, device compliance check, etc. do not work for ROPC



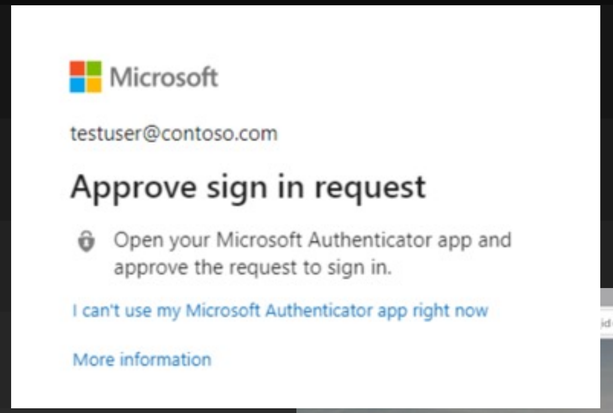
# Quick Aside on Jamf Connect and Similar Tools

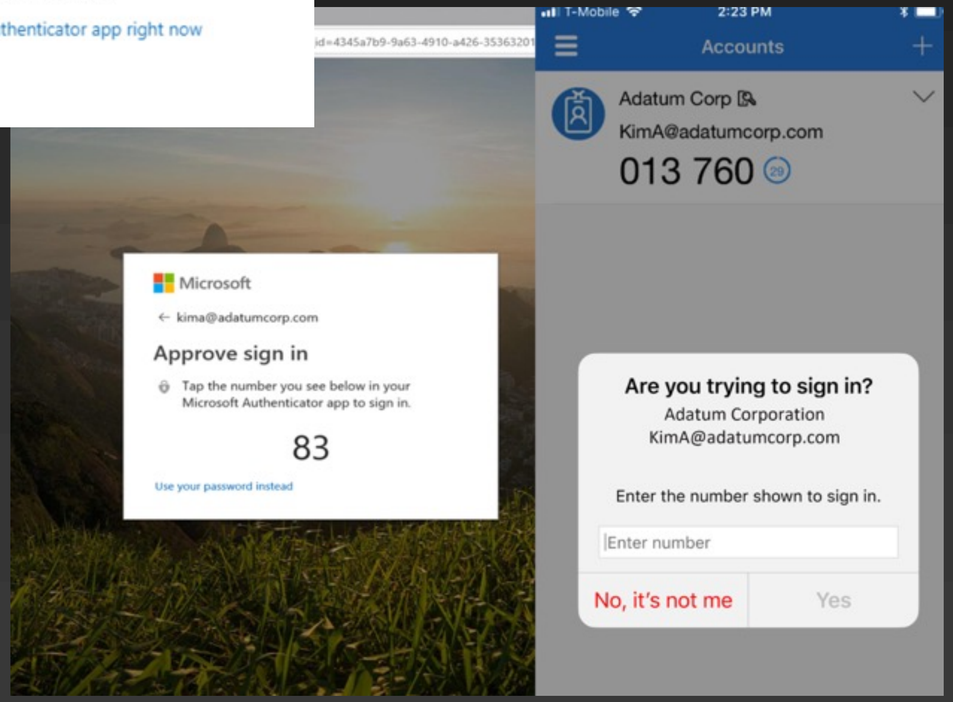
- Jamf Connect can still figure out that the username and password were correct, based on the error from Azure AD
- This can have some negative impacts in Azure AD
  - Azure AD sign-in logs will show failed sign-ins
  - Azure AD Identity Protection may flag the user as compromised, potentially locking the user out of the tenant
- Can avoid these issues by exempting Jamf Connect logins from Conditional Access policies – work with your IAM team



# Recommendation 4: Authenticator App and Passwordless

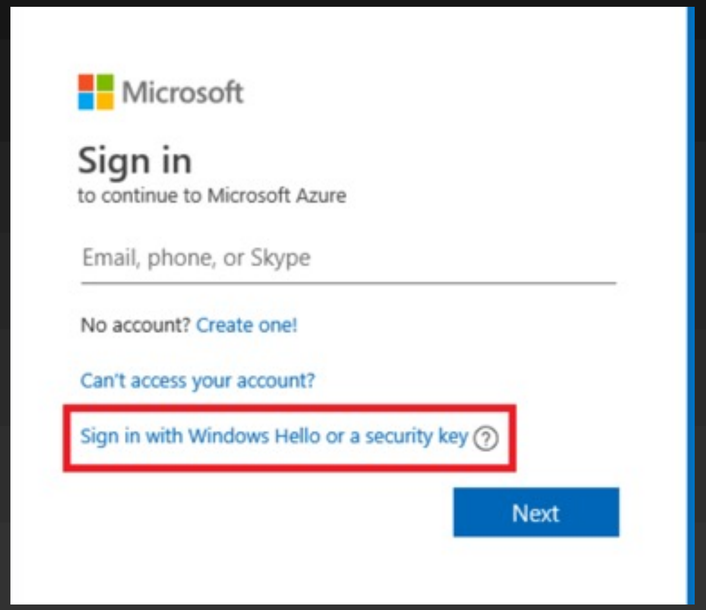
- Authenticator App used as a token broker for iOS devices (similar to Company Portal on MacOS)
  - Provides that PRT experience
- https://aka.ms/nudge will interrupt on sign-in to register for Authenticator App
- Move from push notification to number match if possible (MFA hammering)
- Also used as a passwordless method

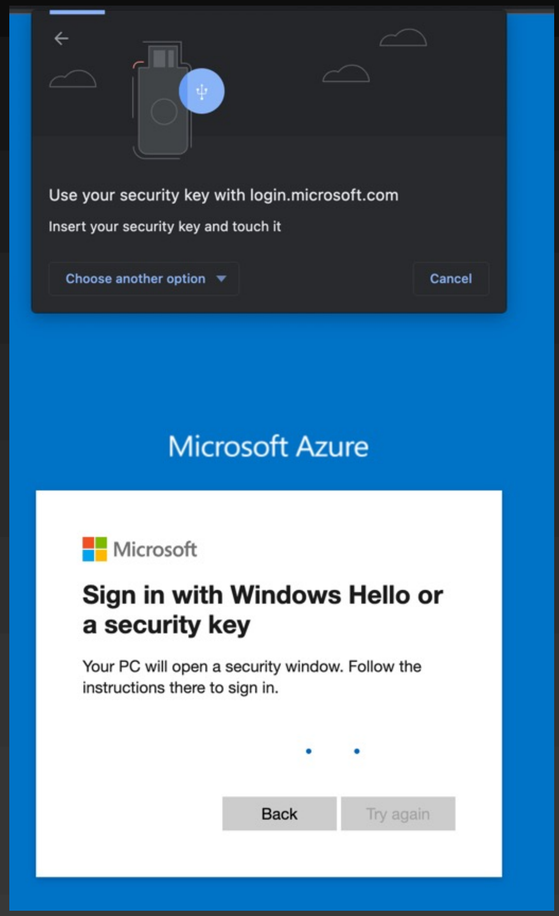


# Recommendation 4: Authenticator App and Passwordless

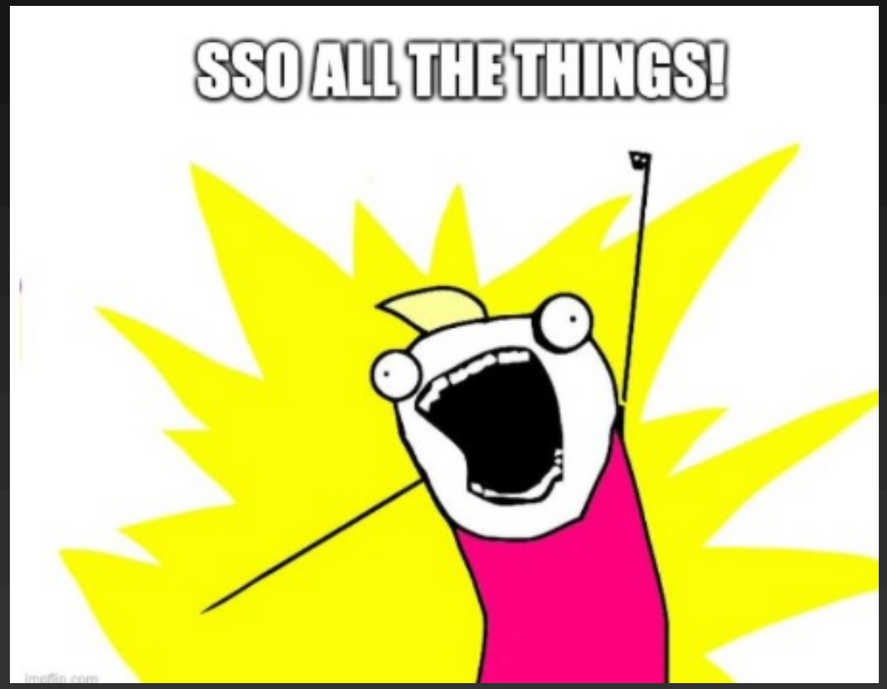
- Best user experience + Best security
  - We've been passwordless since Nov 2020 on macOS!
  - Can be used with any app integrated in your Azure AD
- Passwordless methods
  - Authenticator app number match
  - FIDO2 Key
    - Private key never leaves the physical key
    - Edge and Chrome today
    - Safari in the future
- Passkeys
  - Emerging standard supported by Apple, Microsoft and Google!
  - Passkey synced across devices on same device platform



# Recommendation 5: SSO All the things!

- All the work you do for steps 1-4 won't matter much if your apps aren't integrated with your IDP
- Azure AD can publish many kinds of apps
  - Modern Auth (SAML, OAuth 2.0, OIDC)
  - On-premises legacy Kerberos
  - Password-based
  - Almost anything else via 3rd party integrations (F5, Akamai, etc.)
- We try to make it easy for you…



# Recommendation 5: SSO All the things!

## 3000+ pre-integrated apps in the gallery

### Federated Connectors

| | | |
|---|---|---|
| Sauce Labs – Mobile and Web testing | SkyHigh Networks | Jamf Pro |
| Skillport | Palo Alto Networks | Fidelity NetBenefits |
| OneTrust Privacy Management Software | Adobe Creative Cloud | Experience Manager |
| Apptio | Carlson Wagonlit Travel | DigiCert |
| SAP Cloud Platform Identity Authentication | Form.com | OrgChart Now |

### Provisioning Connectors

| | |
|---|---|
| Cisco WebEx | Samanage |
| GitHub | LucidChart |
| BlueJeans | Zendesk |
| Tableau Online | ThousandEyes |
| Pingboard | Slack |

### 3rd party native Azure AD apps

| | | | |
|---|---|---|---|
| Myday | Canvas | Calendly | Templafy |
| Doodle AG | Smartsheet | Nine for Office365 | K2 for Office365 |
| Exclaimer Cloud | Firefly | Insights | Cronofy |
| Flipgrid | Edmodo | Boomerang | Bluemail |

...and more added each month

**Request a gallery app**: https://aka.ms/AADAppGalleryRequest

# Agenda

What is Azure AD and Conditional Access?

Prompting…why is it bad?

Top 5 Recommendations for the Enterprise

Go-Dos

# 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