



# Offensive TokenCraft

Practical Conditional Access Bypass On Red Team Operations

Sunny Chau 2025

# \$ whoami

Head of Adverary Simulation @JUMPSEC



@gladstomych

sunnyc@jumpsec.com



# Agenda

- Why play with Entra tokens, what are they
- A Browser-first Workflow
- Make-your-own-AiTM
- 3 Scenarios

MFA Gap, 'Typical Cookie theft', Intune-bypass Cookie Theft

- Promise it's packed with TTPs
- All from real engagements

# Why Entra Tokens and What are they

```
GET /v1.0/me HTTP/2
Host: graph.microsoft.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJub25jZSI6IktmLWdzejNncHJrb2F4bDNNX0VMQmxwVUZ6QTZyZnAyLUUzLTBkMTEtMTYxOTg0MDA0LnRlb2kiOjE2ODc1NDUzMTUzNTUzMDE2ZW5kaGUiLCJhdWQiOiJ1cm9wdCIsImF1dG8iOiJjb25jaWRpdGEudGVhbSIsImFjdGlzeSI6InVzZXJuZXQifQ==

Response
```

Pretty Raw Hex Render

```
Odata-Version: 4.0
Date: Thu, 17 Oct 2024 14:50:45 GMT
{
  "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#users/'$ser'",
  "businessPhones": [
  ],
  "displayName": "Sunny Chau",
  "givenName": "Sunny",
  "jobTitle": null,
  "mail": "sunnyc@lumpsec.com"
```

# Access Tokens

This is what you prob  
think of 'Entra tokens'  
as

Resource -  
[graph.microsoft.com](https://graph.microsoft.com)

API call –  
GET /v1.0/me

# Baseline Refresh Tokens

- Lives for 90 days
- 1.A...
- Same or Less scope, Same Resource
- Used to get new Access Tokens

# Family of Refresh Tokens – it's super cool!



The screenshot shows a GitHub repository page for 'secureworks/family-of-client-ids-research'. The page has a dark theme. At the top, there's a navigation bar with 'README' and 'MIT license' links. Below that, there are buttons for 'launch' and 'binder'. The main heading is 'Abusing Family Refresh Tokens for Unauthorized Access and Persistence in Azure Active Directory'. Underneath the heading, there are two authors listed: Ryan Marcotte Cobb, CTU Special Operations and Tony Gore, CTU Special Operations. A paragraph of text describes the research: 'Undocumented functionality in Azure Active Directory allows a group of Microsoft OAuth client applications to obtain special "family refresh tokens," which can be redeemed for bearer tokens as any other client in the family. We will discuss how this functionality was uncovered, the mechanism behind it, and various attack paths to obtain family refresh tokens. We will demonstrate how this functionality can be abused to access sensitive data. Lastly, we will share relevant information to mitigate the theft of family refresh tokens.' At the bottom, there's a section titled 'Updates'.

github.com/secureworks/family-of-client-ids-research

README MIT license

launch binder

## Abusing Family Refresh Tokens for Unauthorized Access and Persistence in Azure Active Directory

- Ryan Marcotte Cobb, CTU Special Operations
- Tony Gore, CTU Special Operations

Undocumented functionality in Azure Active Directory allows a group of Microsoft OAuth client applications to obtain special "family refresh tokens," which can be redeemed for bearer tokens as any other client in the family. We will discuss how this functionality was uncovered, the mechanism behind it, and various attack paths to obtain family refresh tokens. We will demonstrate how this functionality can be abused to access sensitive data. Lastly, we will share relevant information to mitigate the theft of family refresh tokens.

### Updates

There are boring  
Ref tokens and  
there are **Foci  
Ref Tokens**

Cross-client  
Cross-resource  
Cross-scope

(with caveat)

# What about Browser Cookies?

<https://www.xintra.org/blog/tokens-in-entra-id-guide>

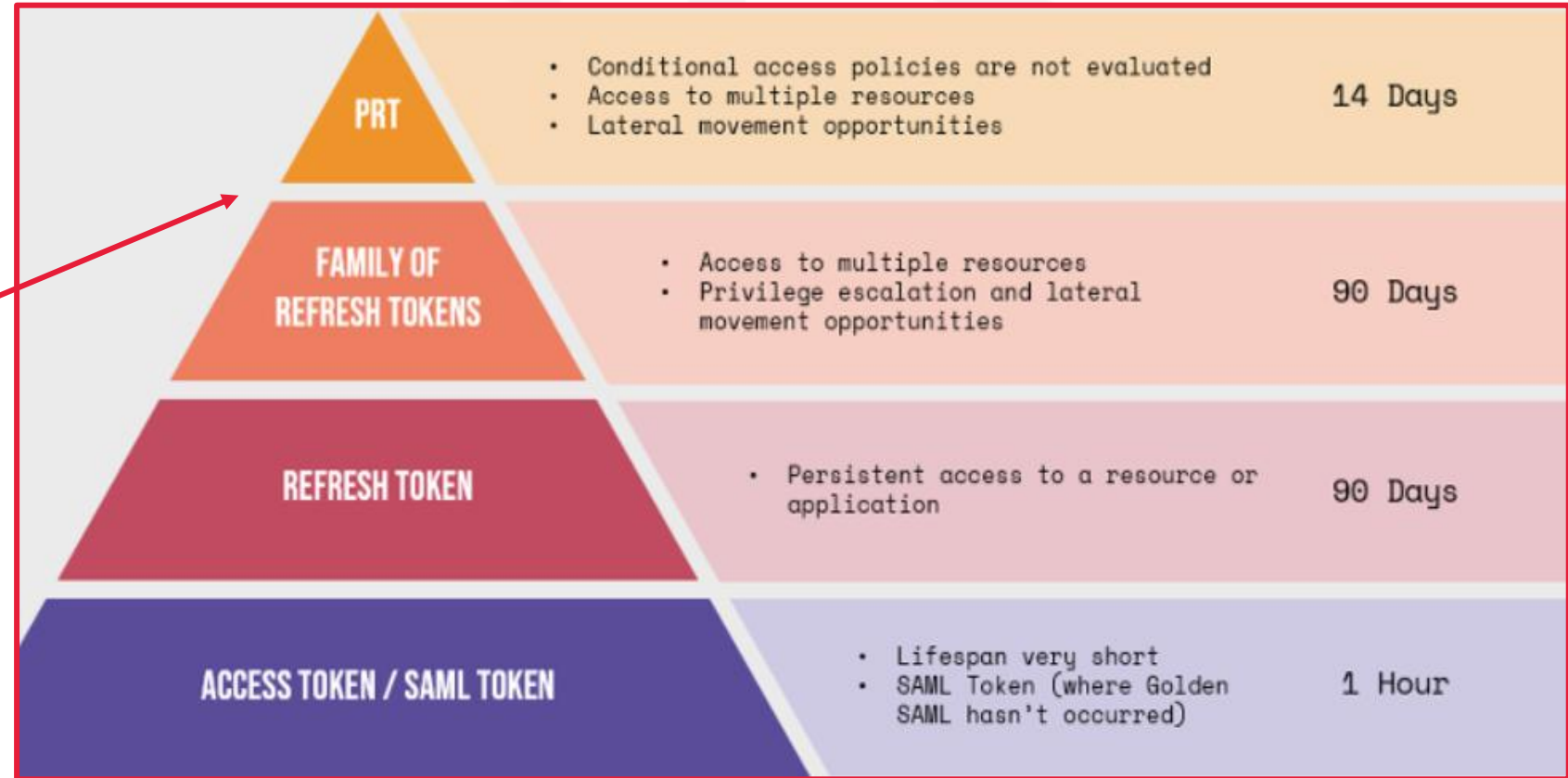
I argue -

Browser cookies:

**ESTSAUTH &  
ESTSAUTHPERSISTENT**

Above "Family tokens"  
Below PRT

New ones called BroCi  
tokens





Browser as the new frontier

# Browser is the new frontier (vs endpoint)

- Our Experience – Entire RT's without touching user endpoint
- Both in Hybrid & Cloud-native land
- TI / Our own IR – TA's go payload-less

why? Initial access via SSO in VPN, Tooling in Linux

- Or, one of our favorites


(TTP 0 – ask for VPN provision)

(TTP 1 – look for a VPN installer in SharePoint)

(TTP 2 – search for a VPN installer with OSINT tech)

# Bridging the Gap between Browser and Tokens

- Your Ref tokens are cool but... how do you get your hands on one?
- **ESTS Cookies (Auth & Persistent)** are what you get from phishing\
- Ref & Access Tokens were what you use to run tools, but ...?



```
12abb4f ROADtools / roadtx / roadtools / roadtx / selenium.py
Blame 665 lines (622 loc) · 33.9 KB Raw
def selenium_login_with_estscookie(self, url, identity=None, password=None, otpseed=None, keep=False, capture=False, estscookie=None)
...
def selenium_login_with_estscookie(self, url, identity=None, password=None, otpseed=None, keep=False, capture=False, estscookie=None)
...
Selenium login with ESTSAUTH or ESTSAUTHPERSISTENT cookie injection
...
def interceptor(request):
```

Hey mom, I inspired people ('s tooling)!  
(EntraTokenAid, TokenTacticsV2, even roadtx?)

github.com/zh54321/EntraTokenAid

README MIT license

2025-04-11

Added

- It is now possible now generate the authentication, copy the URL containing the token. `$tokens = Invoke-Auth -Manual`

Note: Inspired by:

- [TokenTacticsV2](#)
- [TokenSmith](#)

## roadtx codeauth

This command exchanges an authorization code for an access / refresh token. This is essentially a helper method for the [code grant flow](#), the most common flow in OAuth2 authentication in Azure AD.

github.com/f-bader/TokenTacticsV2

README BSD-3-Clause license

0.2.6 (2025-01-04)

- Fix bug custom scopes in `Get-AzureAuthorizationCode` and `Get-AzureTokenFromAuthorizationCode`
- Change default redirect Uri for `Get-AzureAuthorizationCode`

0.2.5 (2025-01-04)

- Added new cmdlets `Get-AzureAuthorizationCode` and `Get-AzureTokenFromAuthorizationCode`. Those cmdlets are heavily inspired by [TokenSmith](#) maintained by [@gladstomych](#)
- Added new cmdlet `Invoke-RefreshToDeviceRegistrationToken` which is a TokenTactics version of the [AADInternals](#) cmdlet `Get-AccessTokenForAADJoin`
- Added v1 endpoint support for `Invoke-RefreshToToken` with the `UseV1Endpoint`. This was required to add `Invoke-RefreshToDeviceRegistrationToken`

# What is the Authorization code flow?

/oauth2/v2.0/**authorize** **response\_type=code**



Go through Password/MFA, or **Existing Cookies**



Entra (feeling satisfied): Redirected code=<auth\_code>



POST to **/token** to redeem Access & Refresh Tokens

# What is the Authorization code flow?

/oauth2/v2.0/**authorize** **response\_type=code**



Go through Password/MFA, or **Existing Cookies**



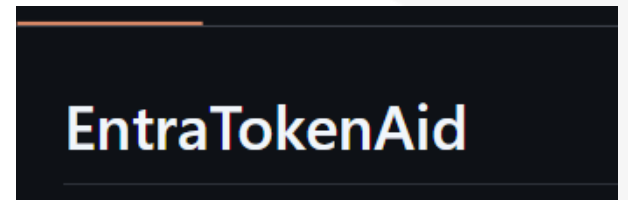
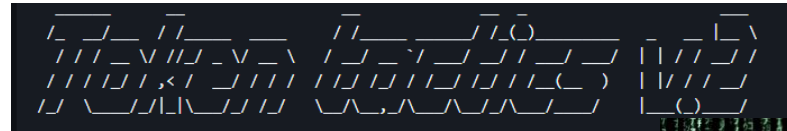
Entra: Redirected code=<auth\_code>



*(THIS PART CAN BE BROKEN UP)*



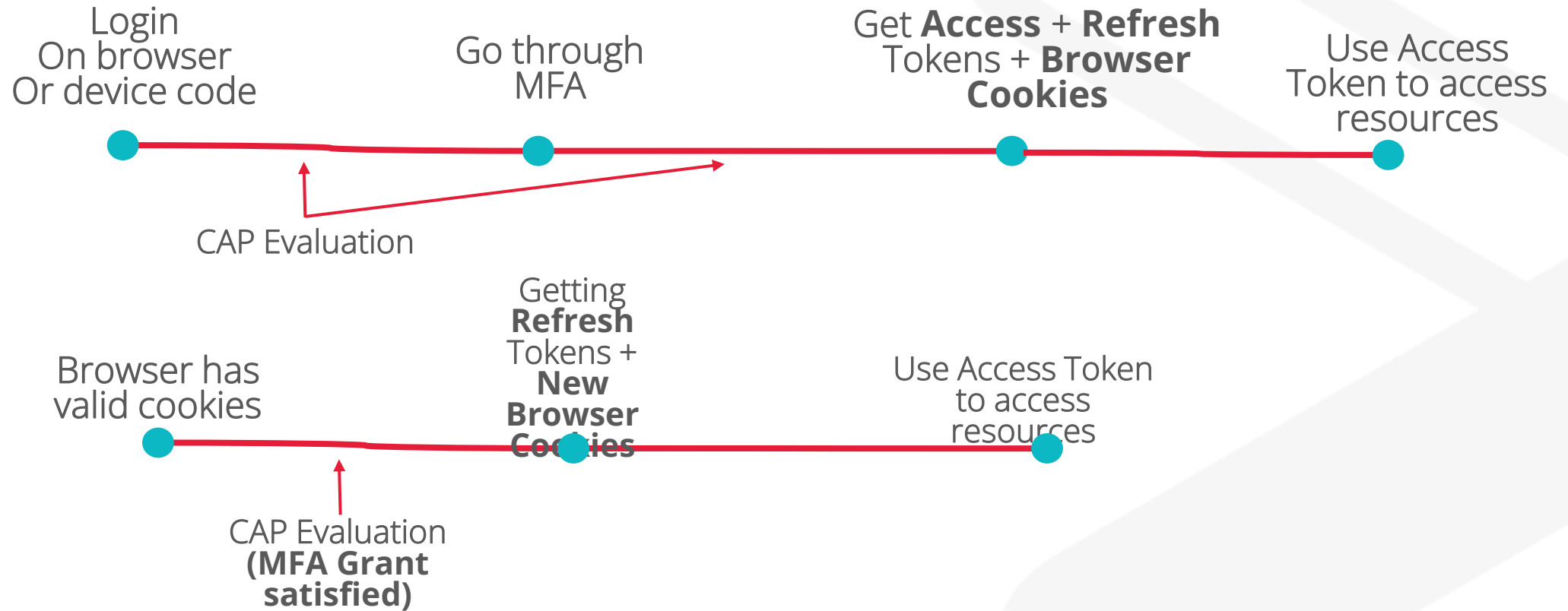
POST to **/token** to redeem **Access & Refresh Tokens**



Let's lay some groundwork first -  
What is Grant, When is CAP eval'd

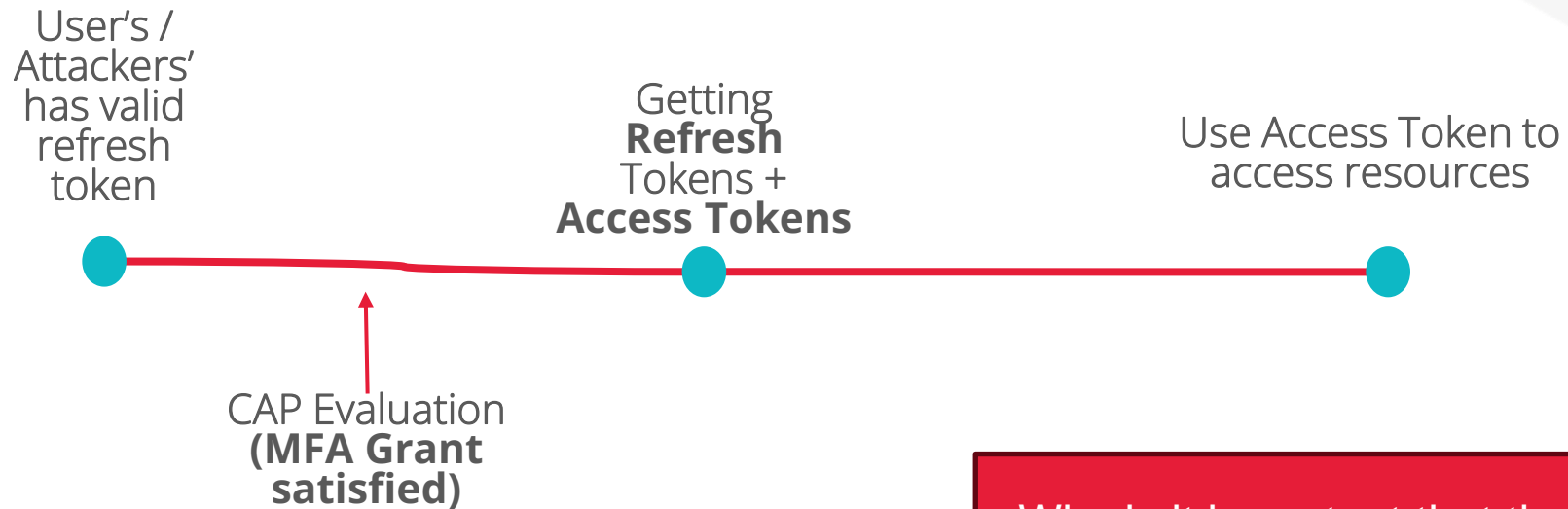
One must imagine Entra ID happy?

# Where are CAPs evaluated? (interactive sign in)



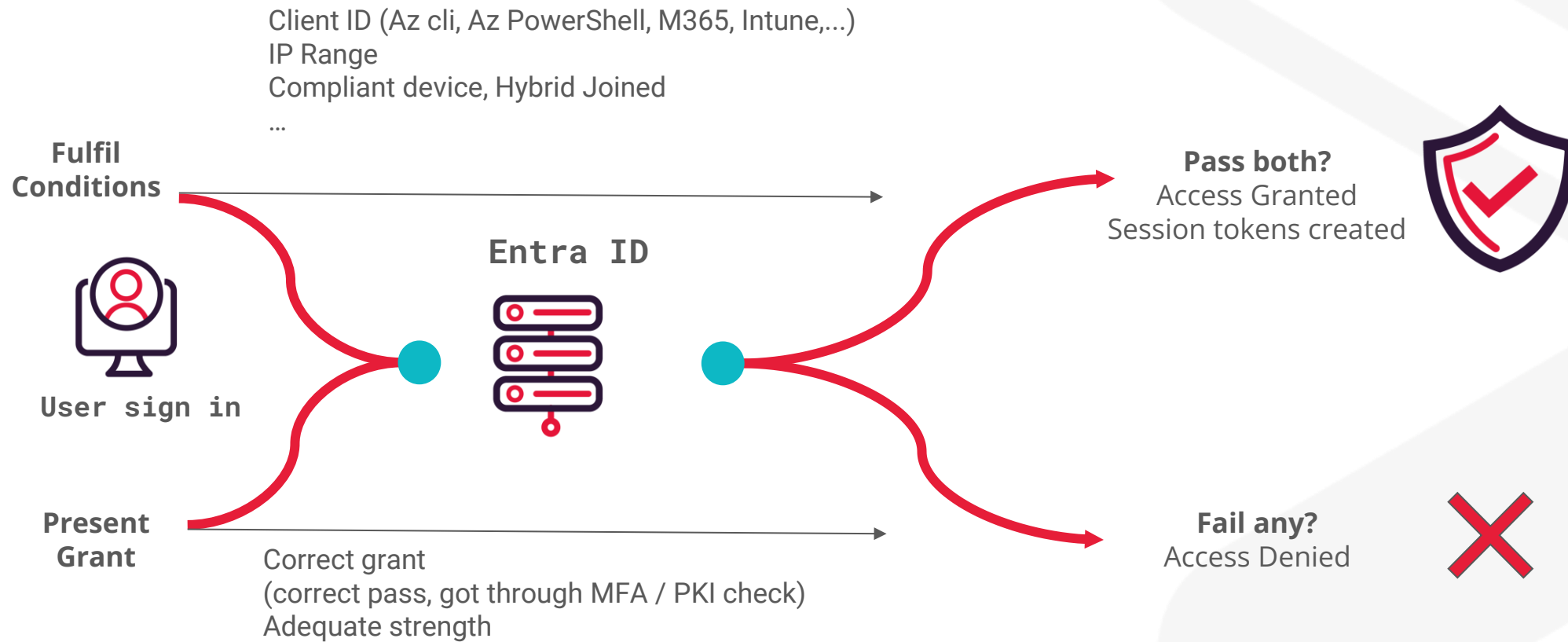


# What's with ref tokens? (non-interactive logins)



Why is it important that the Ref token Contains the original MFA grant?

# Pictorially



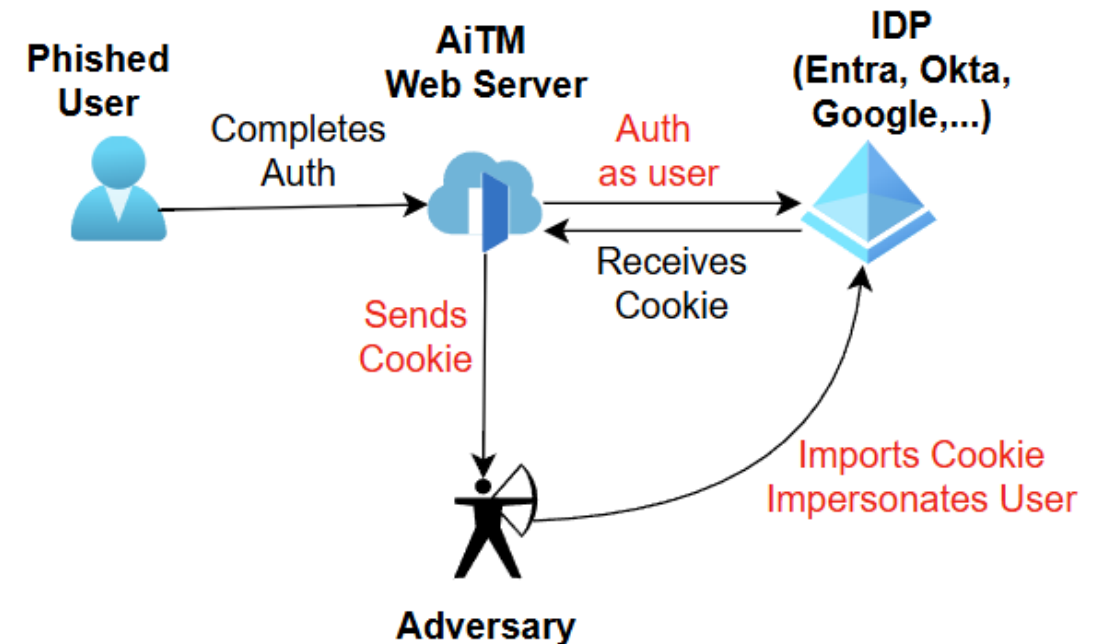
Let's lay some groundwork first -  
AiTM Phishing

# AiTM Flow for Microsoft Entra ID - 1

User lured to AiTM site (acting as a reverse proxy)

User enters credentials and MFA.

Malicious Server intercepts the returning ESTS\* cookies for authentication

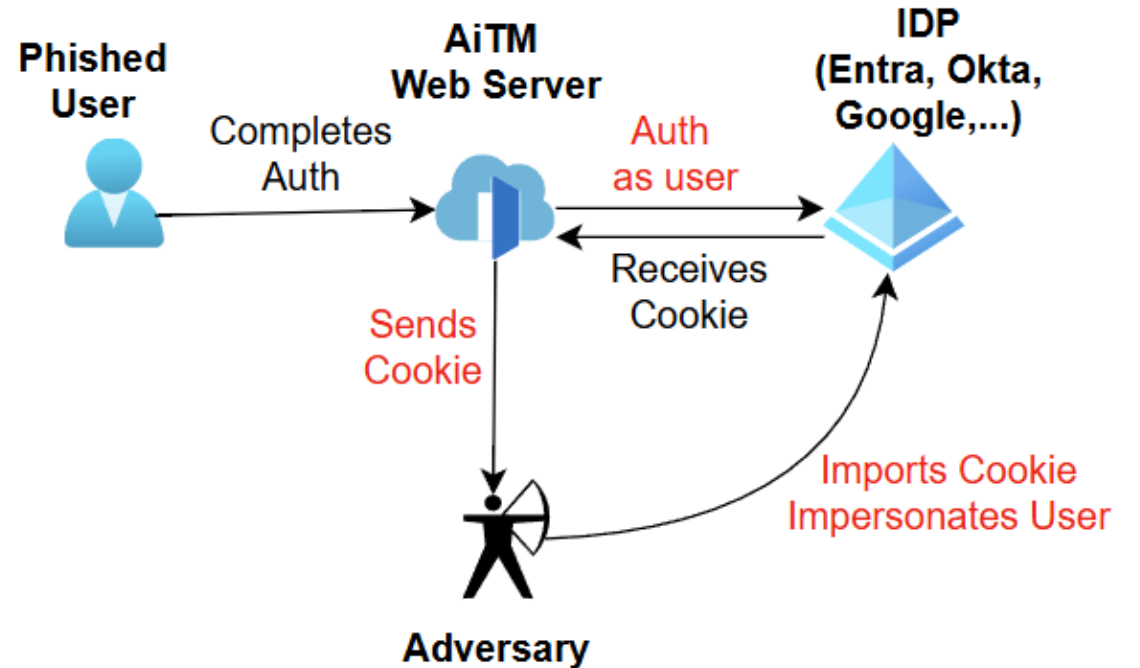


# AiTM Flow for Microsoft Entra ID - 2

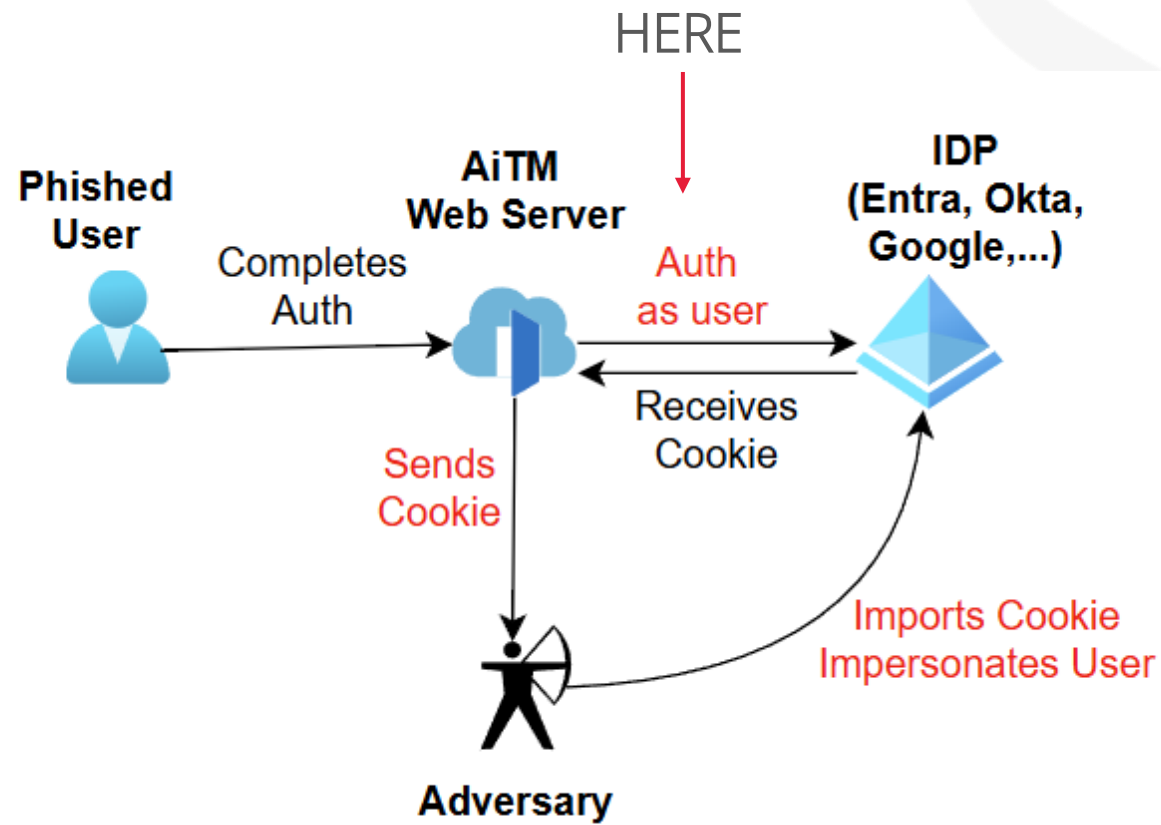
User returned to attacker-controlled redirection site

Attacker Imports ESTS\* Tokens into the browser for session theft

OAUTH Flow for swapping the ESTS\* cookie for Graph and Refresh tokens



# Where's CAP eval in this?



# ESTSAUTH Tokens

Wednesday, July 2nd



PhishingData APP 1:42 PM

🎯 Pwned - Password received!

User: [derpy.](#)

Password: **A password here**

Note: The 1st cookie below is unlikely to provide you access unless they have no 2FA.

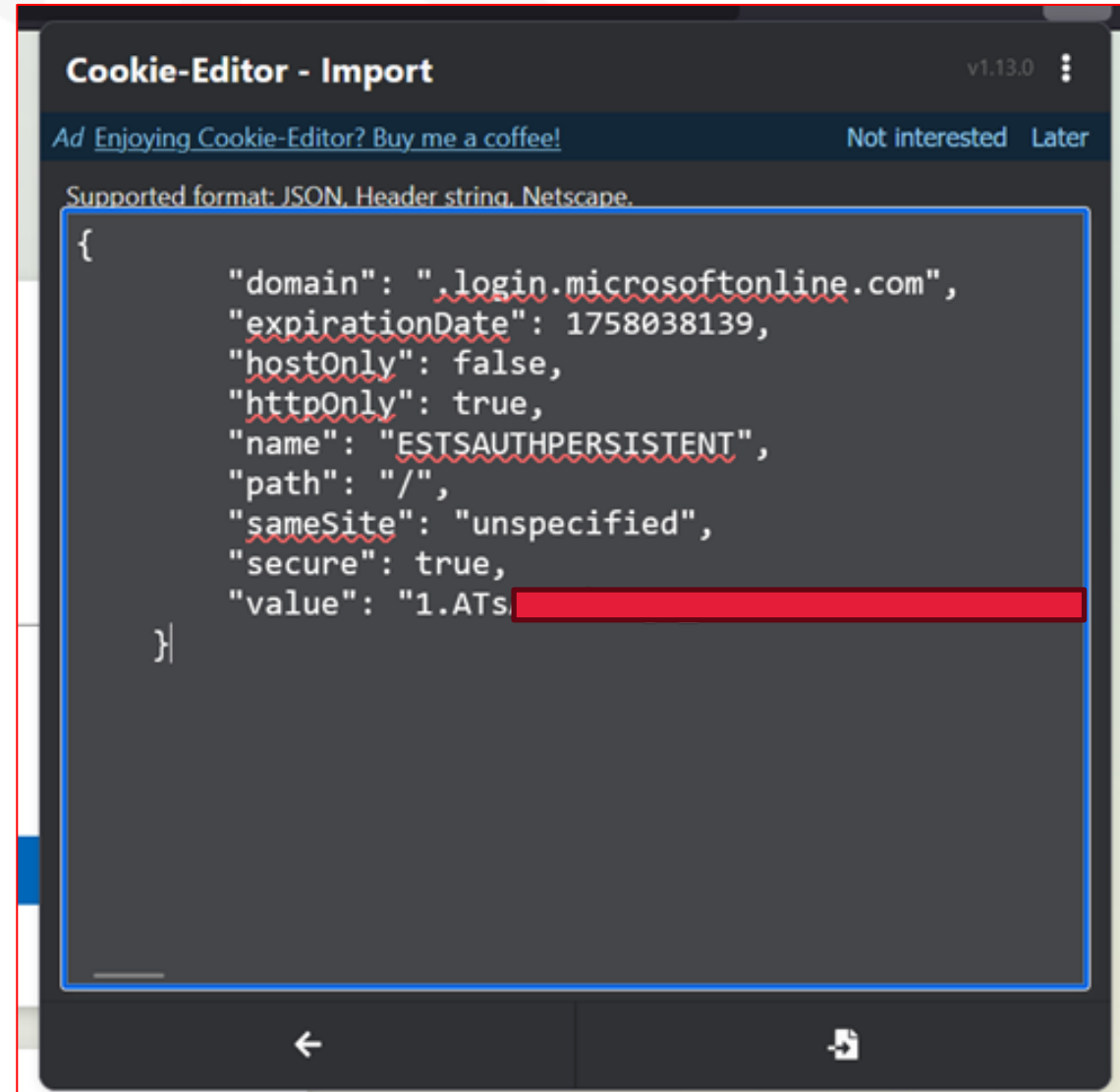
The 2nd cookie will contain the 2FA data and the 3rd one the 2FA + 'Stay signed in' data.

🍪 Cookies found!

esctx-jQZDvhRHY4=; domain=.login.microsoftonline.com; expires=Tue, 01-Jul-2025 12:42:33 GMT; path=/; SameSite=None;  
ESTSAUTHPERSISTENT=1.AUEBe6mZqbzPUkCgIYFUh6CA8VtEZUfGMrBJg-

# Session Theft

```
te@tdejmp:/mnt/c/Users/te/Downloads$ cat oi.cok  
[REDACTED]; domain=.login.microsoftonline.com; expires=Tue, 17-  
ESTSAUTHPERSISTENT=1.ATs, [REDACTED]
```





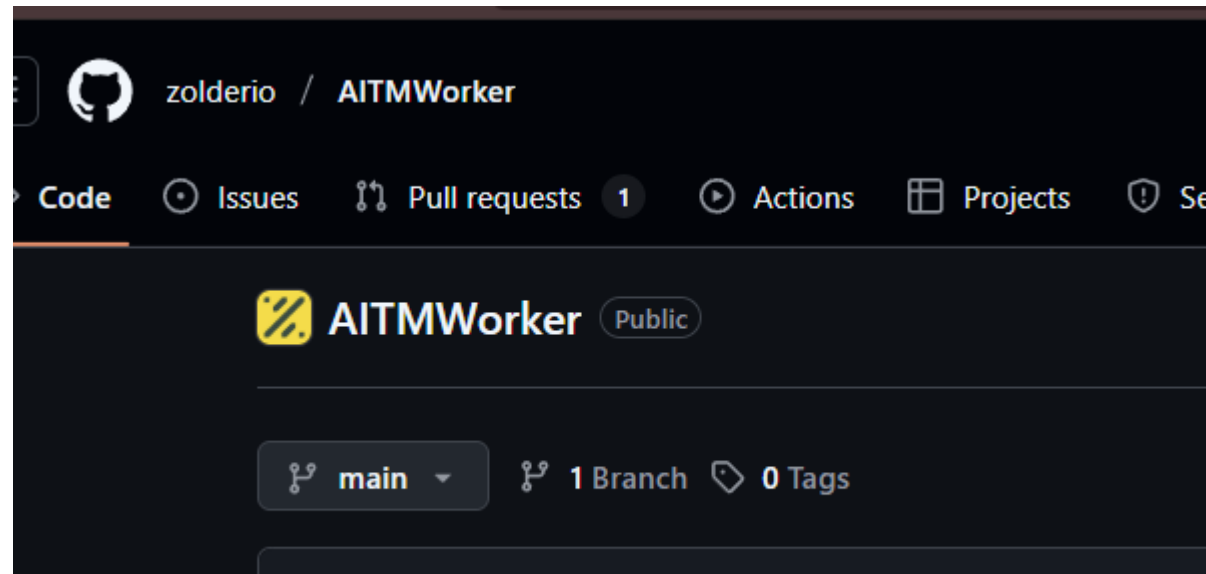
# Let's build our AiTM logic

- Reverse Proxy over – login.microsoftonline.com
- Starts user on an appropriate /oauth2/v2.0/**authorize** URL when they hit your Lure path
- THERE WILL ALWAYS BE a final redirect, today it's still by Location header
  - Check it in your proxy, and redirect your victim accordingly
- Set-Cookies: ESTSAUTH, ESTSAUTHPERSIST
  - Send it to you
- User punch in username & password? –
  - Sent it to you

# Let's build our AiTM logic

- Reverse Proxy over – login.microsoftonline.com
- Starts user on an appropriate /oauth2/v2.0/**authorize**
- Check redirect by Location header
- Set-Cookies: ESTSAUTH, ESTSAUTHPERSIST > to attacker
- User punch in username & password? > to attacker

Is it really that simple???



# Let's build our AiTM logic

*Is it really that simple??? **Actually no***

## **Opsec requirement**

- Blocking bots
- Allowlisting your IP before goes live
- Maybe reCAPTCHA
- Visits without proper lure URL would be redir to harmless 302

## **Sensible SSL**

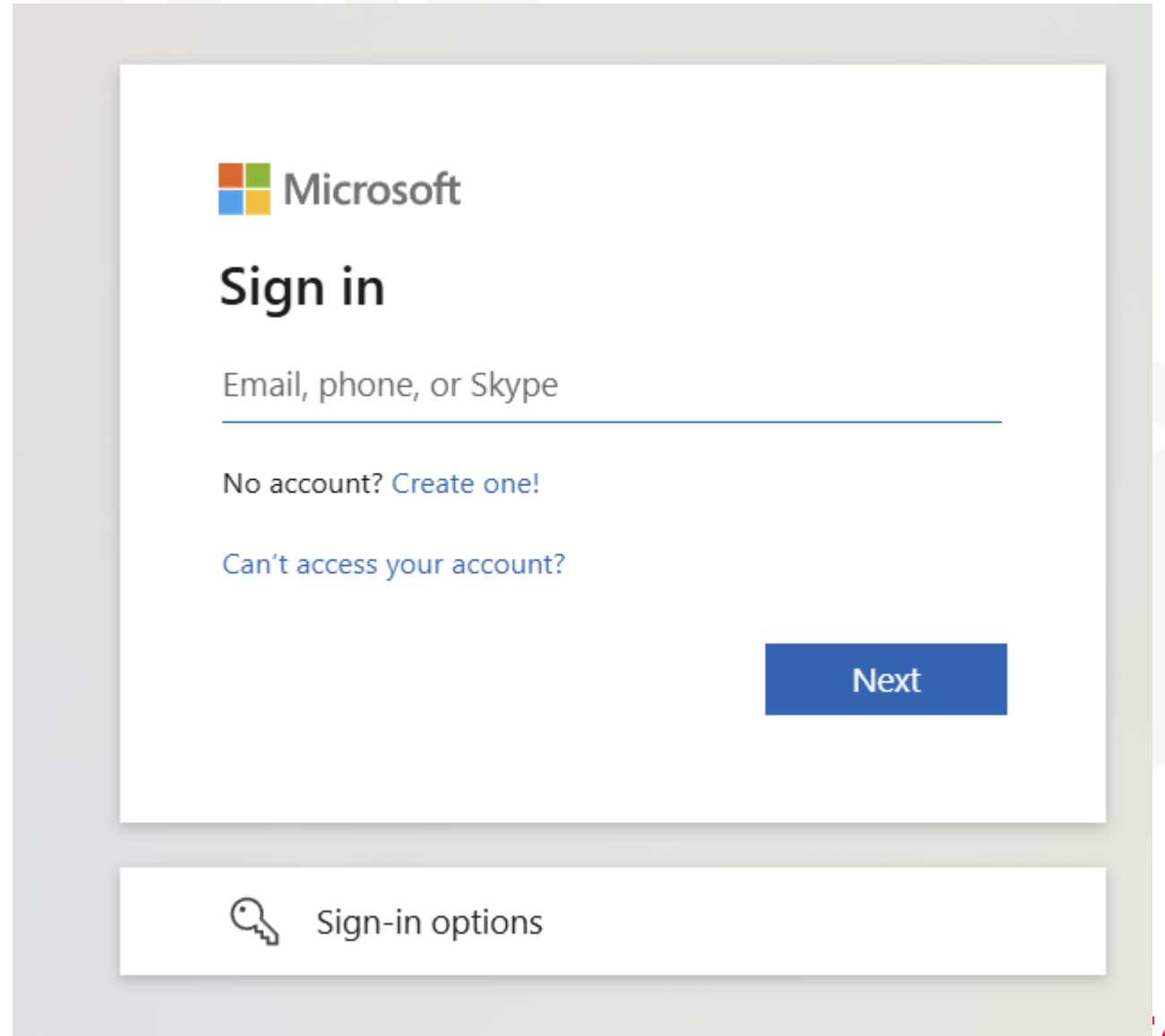
## **Client Branding..**

*If you don't like xxxxGinx (or any available tooling), it is something that can be sensibly build in a week (or 4) in fact*

# TTP3: 'common' trick

Easiest way to do client branding

login.microsoftonline.com/**common**/oauth2/v2.0/authorize...

A screenshot of the Microsoft Sign in page. At the top left is the Microsoft logo. Below it is the text "Sign in". Underneath is a text input field with the placeholder "Email, phone, or Skype". Below the input field are two links: "No account? Create one!" and "Can't access your account?". At the bottom right is a blue button labeled "Next". At the bottom left is a link with a key icon labeled "Sign-in options".

Microsoft


Sign in

Email, phone, or Skype

No account? [Create one!](#)

[Can't access your account?](#)

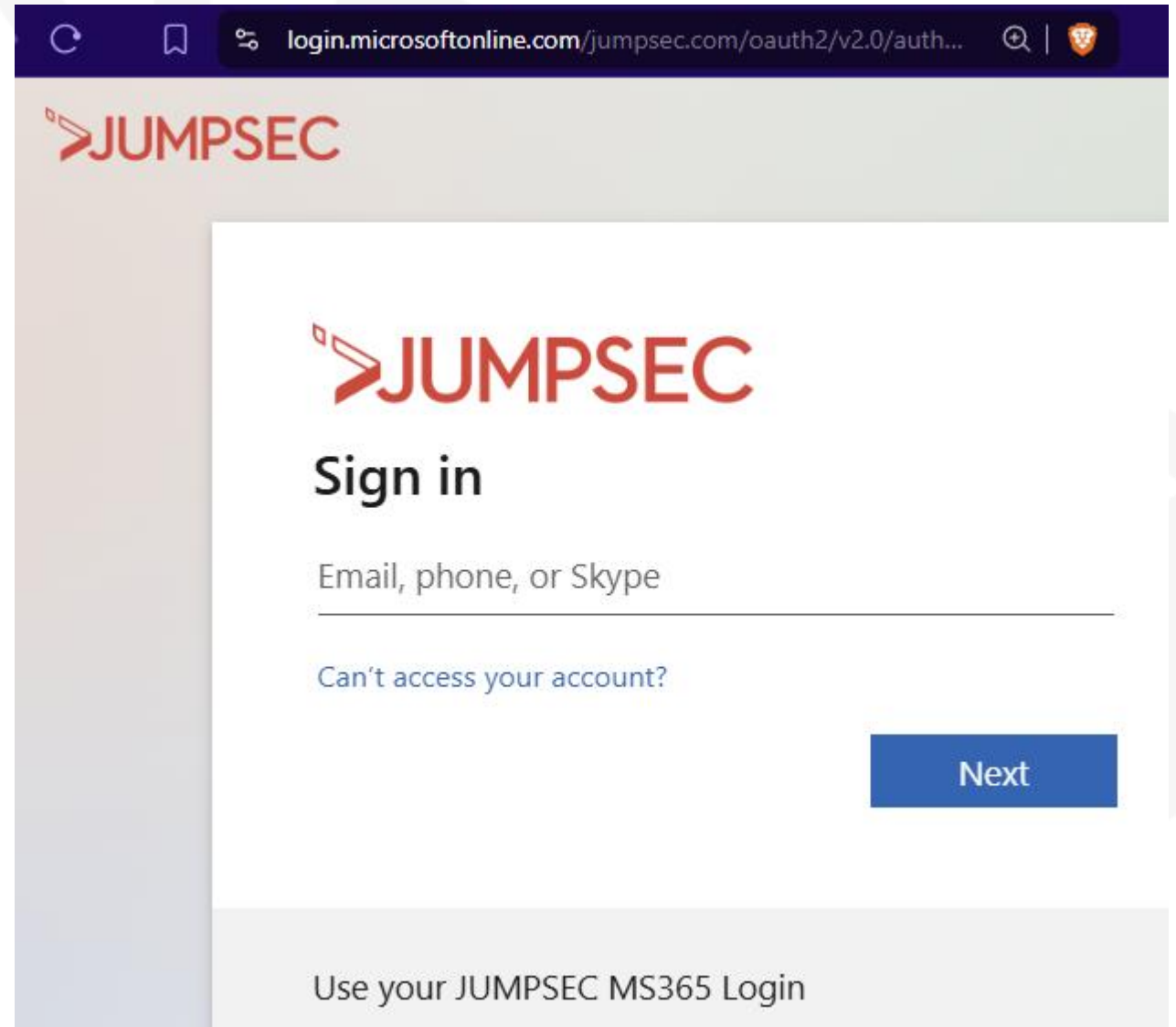
Next

 [Sign-in options](#)

# Trick 1: 'common' trick

Easiest way to do client branding

`login.microsoftonline.com/client.d  
omain/oauth2/v2.0/authorize...`



# Practical RT Scenarios

# Scenario 1 – RT has valid creds but no MFA

My first time ever hacking into an M365 environment – there was a gap with Teams on iPhone

TTP4 – Teamfiltration & MFA gap bruteforcing



```
.\\beac0n\\ --exfil --username derpy.fonder@defi[REDACTED] --password [REDACTED] --config .\\TeamF
es does not use FireProx, ORIGIN IP WILL BE LOGGED, are you an adult? (Y/N)
/2025 5:46:20 PM EST Sprayed derpy.fonder@defi[REDACTED] => VALID BUT MFA (76)
PM EST Attempting to enumerate potential conditional access policy
/2025 5:46:20 PM EST URI: https://api.spaces.skype.com/ APP: Microsoft Teams PLATFORM: Android => VALID BUT MFA (76)
/2025 5:46:21 PM EST URI: https://api.spaces.skype.com/ APP: Microsoft Teams PLATFORM: iPhone => CAN ACCESS
unny\\Tools\\TeamFiltration]- (10/09 22:46:21)
```

# What do you mean, Teams on iPhone gap

- Device Platforms
- Include: ANY
- Client Apps: ALL
- This is the correct setting

**Device platforms** ✕

Apply policy to selected device platforms.  
[Learn more](#)

Configure ⓘ

☒ Yes ☐ No

**Include** **Exclude**

☒ Any device

☐ Select device platform:

- ☐ Android
- ☐ iOS
- ☐ Windows Phone
- ☐ Windows
- ☐ macOS
- ☐ Linux

**Client apps** ✕

Control user access to target specific client applications not using modern authentication. [Learn more](#)

Configure ⓘ

☒ Yes ☐ No

Select the client apps this policy will apply to

Modern authentication clients

- ☒ Browser
- ☒ Mobile apps and desktop clients

Legacy authentication clients

- ☒ Exchange ActiveSync clients
- ☒ Other clients ⓘ

```
Outlook PLATFORM: Windows => VALID BUT MFA (76)
Outlook PLATFORM: Windows Phone => VALID BUT MFA (76)
OneNote PLATFORM: Android => VALID BUT MFA (76)
OneNote PLATFORM: iPhone => VALID BUT MFA (76)
OneNote PLATFORM: Mac OS => VALID BUT MFA (76)
OneNote PLATFORM: Linux => VALID BUT MFA (76)
OneNote PLATFORM: Windows => VALID BUT MFA (76)
OneNote PLATFORM: Windows Phone => VALID BUT MFA (76)
Outlook PLATFORM: Android => VALID BUT MFA (76)
Outlook PLATFORM: iPhone => VALID BUT MFA (76)
Outlook PLATFORM: Mac OS => VALID BUT MFA (76)
Outlook PLATFORM: Linux => VALID BUT MFA (76)
Outlook PLATFORM: Windows => VALID BUT MFA (76)
Outlook PLATFORM: Windows Phone => VALID BUT MFA (76)
OneNote PLATFORM: Android => VALID BUT MFA (76)
OneNote PLATFORM: iPhone => VALID BUT MFA (76)
OneNote PLATFORM: Mac OS => VALID BUT MFA (76)
OneNote PLATFORM: Linux => VALID BUT MFA (76)
OneNote PLATFORM: Windows => VALID BUT MFA (76)
OneNote PLATFORM: Windows Phone => VALID BUT MFA (76)
Outlook PLATFORM: Android => VALID BUT MFA (76)
Outlook PLATFORM: iPhone => VALID BUT MFA (76)
Outlook PLATFORM: Mac OS => VALID BUT MFA (76)
Outlook PLATFORM: Linux => VALID BUT MFA (76)
Outlook PLATFORM: Windows => VALID BUT MFA (76)
Outlook PLATFORM: Windows Phone => VALID BUT MFA (76)
Outlook PLATFORM: Android => VALID BUT MFA (76)
Outlook PLATFORM: iPhone => VALID BUT MFA (76)
Outlook PLATFORM: Mac OS => VALID BUT MFA (76)
Outlook PLATFORM: Linux => VALID BUT MFA (76)
Outlook PLATFORM: Windows => VALID BUT MFA (76)
Outlook PLATFORM: Windows Phone => VALID BUT MFA (76)
Outlook PLATFORM: Android => VALID BUT MFA (76)
Outlook PLATFORM: iPhone => VALID BUT MFA (76)
Outlook PLATFORM: Mac OS => VALID BUT MFA (76)
Outlook PLATFORM: Linux => VALID BUT MFA (76)
Outlook PLATFORM: Windows => VALID BUT MFA (76)
Outlook PLATFORM: Windows Phone => VALID BUT MFA (76)
```



# Scenario 1 – RT has valid creds but no MFA

Okay you got Teams token with Graph access now

- You can Enum now right?
- Yes – by calling Graph API with the Token Directly
- Why does GraphRunner / RoadRecon not working

Answer: Inadequate Grant (no MFA grant in Ref Token)

How to bypass MFA?

- TTP5 - TI backed – Call helpdesk

# Scenario 2 – RT has phished post MFA cookies

Now we have post MFA grant, what should we do?

- Assuming No full coverage passwordless requirement
- Assuming No full coverage compliant device requirement

# Scenario 2 – RT has phished post MFA cookies

Now we have post MFA grant, what should we do?

- TTP6 – Get on myaccount.Microsoft.com and reg your malicious MFA device!
- And do your usual post-ex business, looting emails, files, get on VPNs and etc
- Now you have a **Hot** browser window, GraphRunner gettoken and RoadRecon Auth should work unless client blocks device code

# Scenario 3 – We know the client requires, or probably requires compliant device

We have post MFA grant but blocked by CAP, what should we do?

Why did we leave this till last?

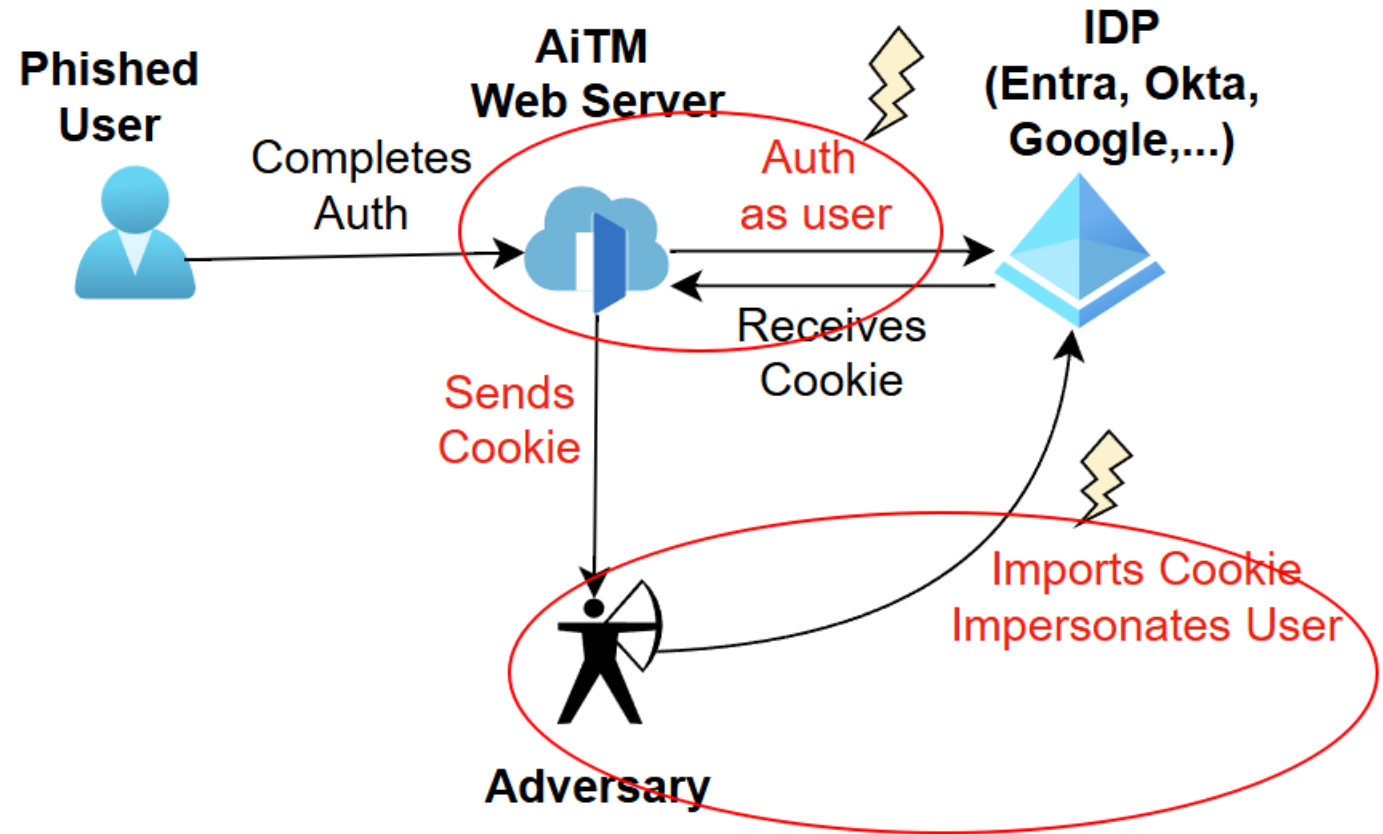
# What happens on a RT?

Something the **user** possesses

But

The AiTM server, and the Adversary does **not**

If the AiTM server does not fit the CAP reqs – then no valid session cookies would be minted



# December whispers

- **HINT 1:**
  - **Client ID: '9ba1a5c7-f17a-4de9-a1f1- 6178c8d51223'**
- **HINT 2:**
  - **[Intune Company Portal]**



User Browser

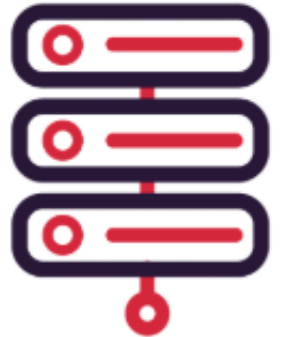


login.microsoft.com/...?**redirect\_uri=attacker.server.com**

completes Entra authentication but client ID does not match pre-registered redirect in Entra

Throws an "incorrect redirect URI error" without providing authorization code

Entra ID



does not get redirected

Attacker Web Server



Cannot redeem tokens from Entra for user

[Placeholder Sunny]

# Message after RTFM

- Redirect URI is not Arbitrary
- Microsoft doesn't publish their first party App redirectors
- So, this is probably the main thing we need to reverse engineer



# How would you approach this?

**WHAT** – being able to run offensive tools

**HOW** - Authenticate into Entra ID with compliant device CAP, without using a compliant device



The red team, using a Microsoft 0-day on the next engagement

# We're blocked on baseline login

Review of logs:

- Did not satisfy the CAP



Microsoft

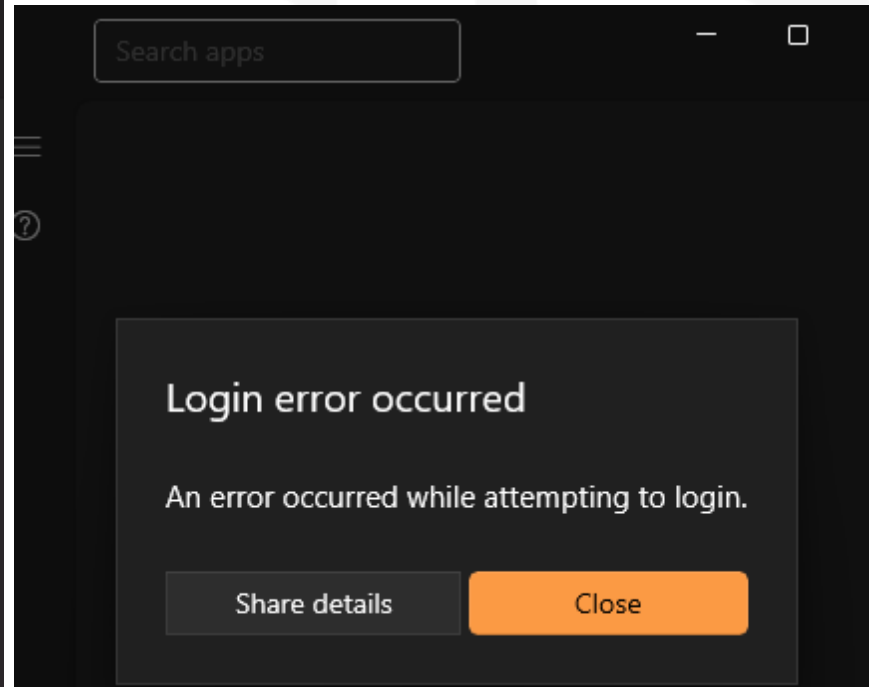
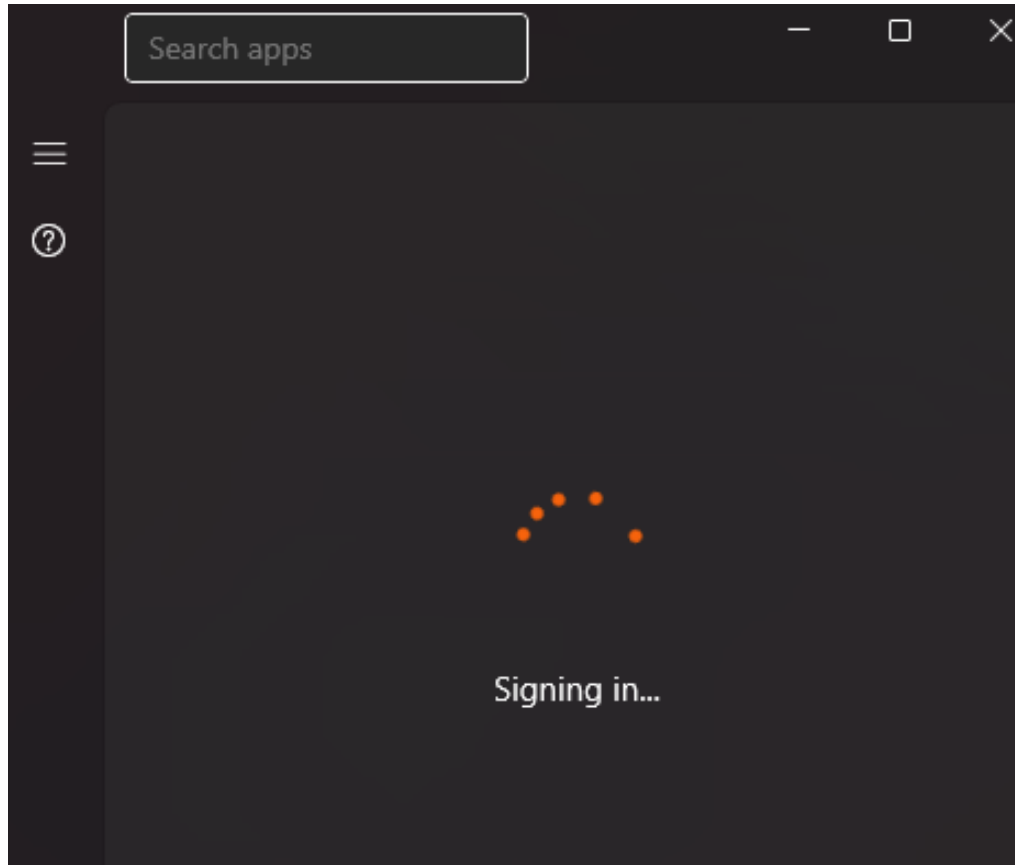
derpy.fonder

## Help us keep your device secure

Your sign-in was successful, but your admin requires the device that's requesting access to be managed by Entra Research to access this resource.

[More details](#)

But for what we needed ... it did not work out



Reason?  
mTLS check?

Proxy  
detection?

Logged at webaccountprocessor.cpp, line: 233, method: AAD::Core::WebAccountProcessor::ReportOperationError.

### Error: 0xCAA82EE2 The request has timed out.

Log Name: Microsoft-Windows-AAD/Operational

Source: Microsoft-Windows-AAD

Date: 15/07/2020 16:00:58

**Event ID: 1098**

Task Category: AadTokenBrokerPlugin Operation

Level: Error

Keywords: Operational,Error

User:

Computer:

Description:

**Error: 0xCAA82EE2 The request has timed out.**

**Exception of type 'class HttpException' at xmlhttpwebrequest.cpp, line: 163, method:**

**XMLHttpRequest::ReceiveResponse.**

Log: 0xcaa10083 Exception in WinRT wrapper.

Logged at authorizationclient.cpp, line: 233, method: ADALRT::AuthorizationClient::AcquireToken.

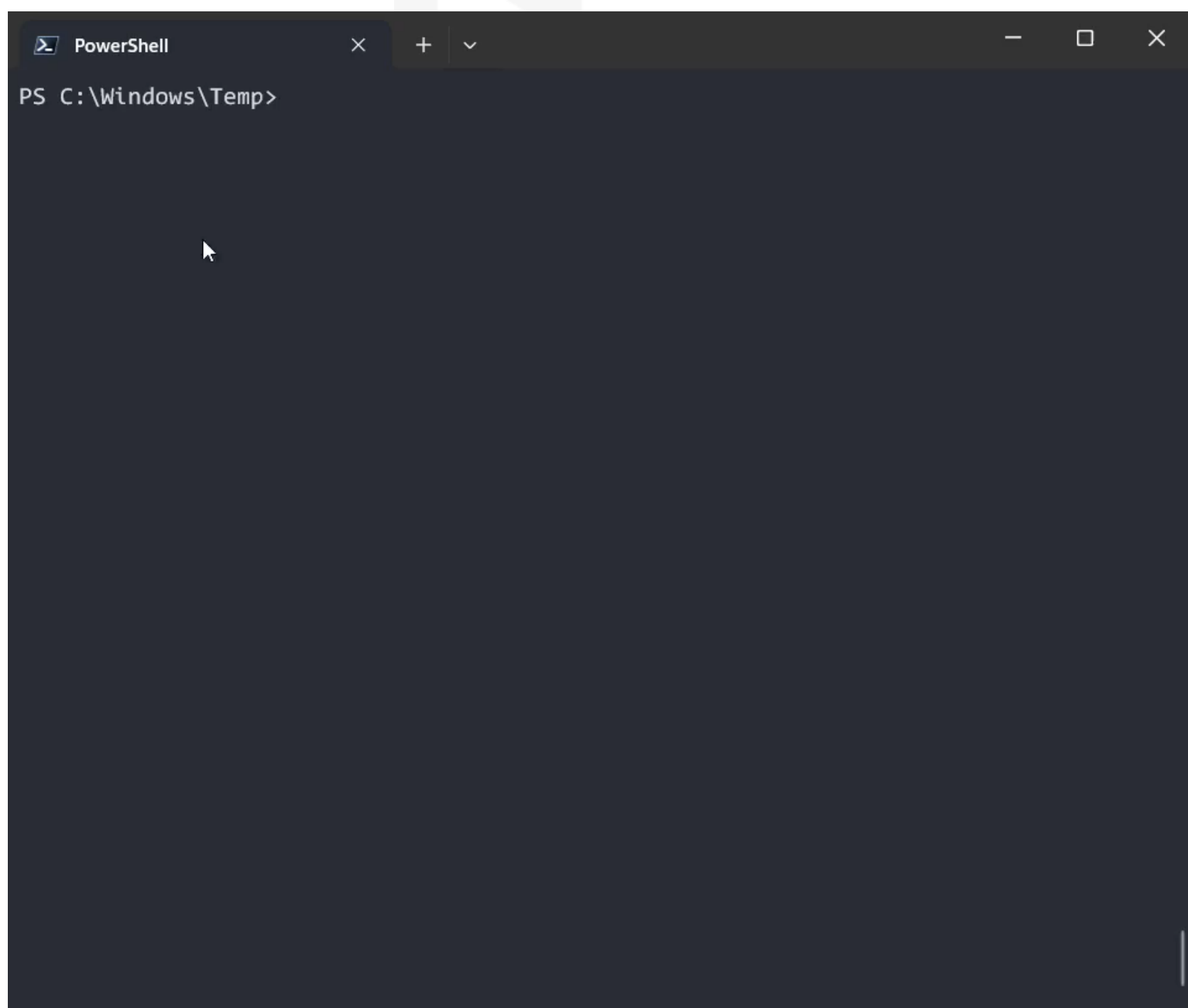
Request: authority: **https://login.microsoftonline.com/common**, client: 8ba1a5c7-f19a-5de9-a1f1-

7178c8d51343, redirect URI: **ms-appx-web://Microsoft.AAD.BrokerPlugin/S-1-15-2-2666988183-**

**1750391847-2906264630-3525785777-2857982319-3063633125-1907478113**

wait what?

Demo time



# The shadow patch

- **Roughly 20<sup>th</sup> Feb 2025** Microsoft quietly reduced the scope for the token you could get from company portal
- Noticeably narrower than the original, notably only on top of the Intune related ones:
  - **ServicePrincipalEndpoint.Read & User.Read**
- Also, Tokensmith's executable has become 'malware'

# So ... is it now useless?

## Remember again

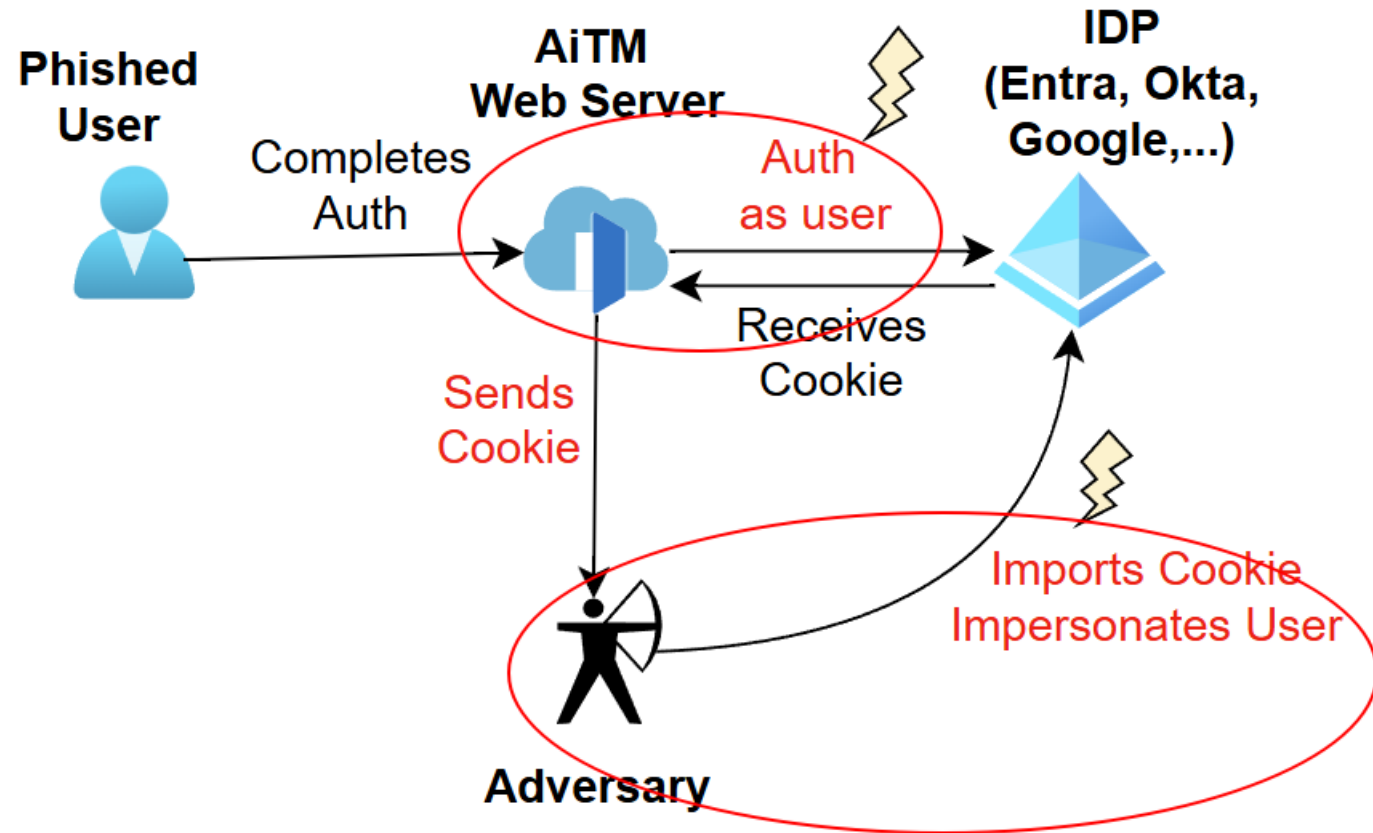
That the

Blank device > Compliant device

Enrollment process

Can never require a Compliant device

**What if the AiTM Web Server tries to sign in to device enrollment?**



Yes it can be done

**TTP7 – use Intune device enrollment endpoint on AiTM web server**



The image shows a YouTube video player interface. The video title is "Combining TokenSmith with Evilginx". The video has 1.6K views and was uploaded 7 months ago. The channel is "SYNACK Time". An update note from 2-20-25 states: "From Sunny (developer of TokenSmith) Microsoft silently patched the scopes accessible by abusin...". The video has 10 chapters: Introduction | Conditional Access Policy | Demo | Evilginx | Modifying Evilginx | ...

**Combining TokenSmith with Evilginx**

TokenSmith Meets Evilginx: Token Theft Combined with Entra Conditional Access Bypass

1.6K views • 7 months ago

SYNACK Time

UPDATE (2-20-25): From Sunny (developer of TokenSmith) Microsoft silently patched the scopes accessible by abusin...

10 chapters Introduction | Conditional Access Policy | Demo | Evilginx | Modifying Evilginx | ...

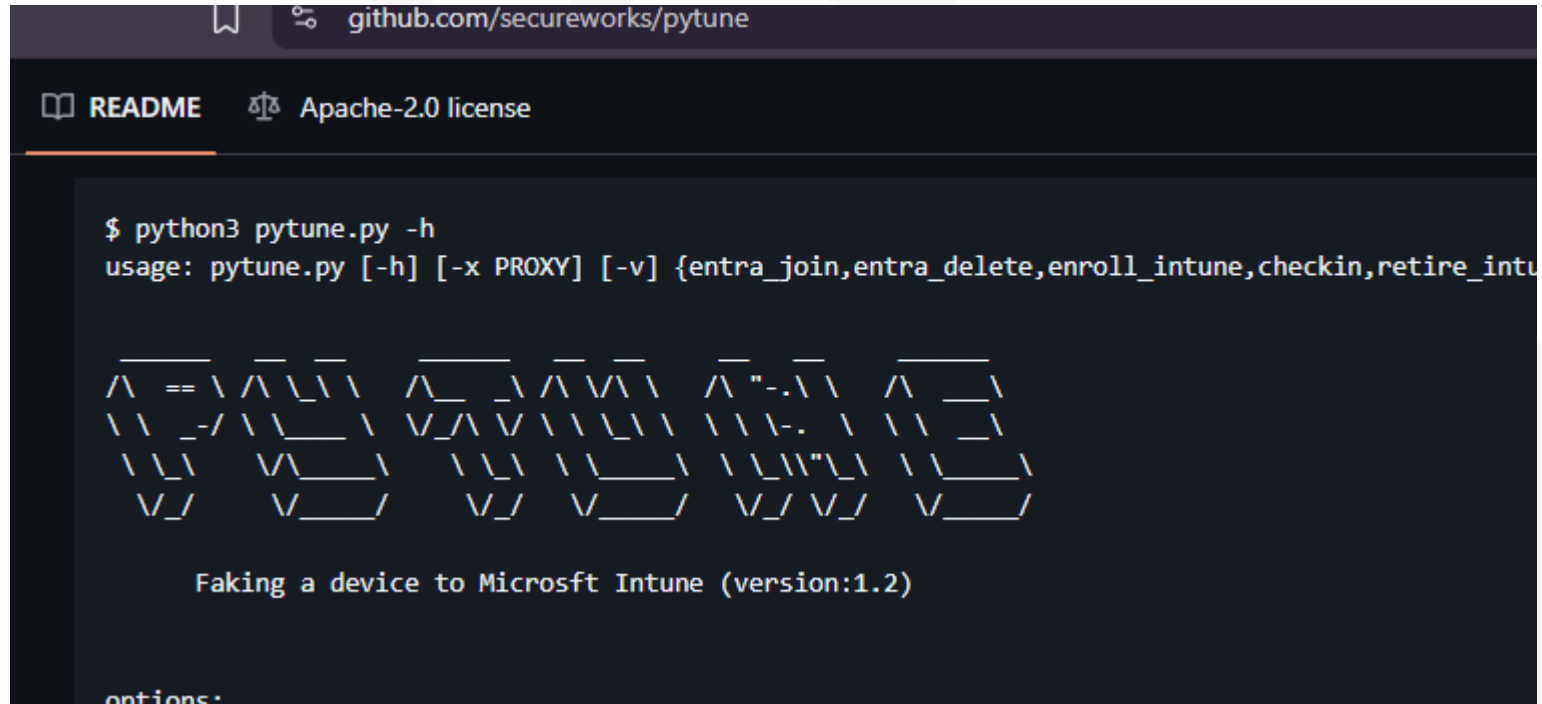


# What do you do with the ESTS Cookies then?

TTP8 – Register a malicious device in client's Entra, Enroll into Intune

And *potentially* fake compliance

*RoadTune (if you're on Outflank OST?)*



```
$ python3 pytune.py -h
usage: pytune.py [-h] [-x PROXY] [-v] {entra_join,entra_delete,enroll_intune,checkin,retire_intune}

Faking a device to Microsoft Intune (version:1.2)

options:
```

# What do you do with the ESTS Cookies then?

You can use the same cookies on any assumed breach device, yes

Also can try other User-Agent, yes

What's even better?

**TTP9 – roadtx auth – device-code bypass**



## Azure Active Directory PowerShell

You have signed in to the Azure Active Directory PowerShell application on your device. You may now close this window.

```
$ roadtx auth --device-code
Requesting token for resource https://graph.windows.net
To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the
Tokens were written to .roadtools_auth
```

# RECAP

Why play with Entra tokens, what are they

A Browser-first Workflow

Make-your-own-AiTM

3 Scenarios

MFA Gap, 'Typical Cookie theft', Intune-bypass Cookie Theft

- Promise it's packed with TTPs
- All from real engagements



# Latest Work in the area

- EntraScopes.com
- Dirk-jan's work on bruteforcing CAP bypasses

