



Offensive TokenCraft

Practical Conditional Access Bypass On Red Team Operations

Sunny Chau 2025

\$ whoami

Head of Adverary Simulation @JUMPSEC



@gladstomych

sunnyc@jumpsec.com



'>JUMPSEC

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

Access Tokens

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

Resource - 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 'family-of-client-ids-research'. The page includes a navigation bar with links to 'README', 'MIT license', 'launch', and 'binder'. The main content features a large title 'Abusing Family Refresh Tokens for Unauthorized Access and Persistence in Azure Active Directory' and a list of authors: 'Ryan Marcotte Cobb, CTU Special Operations' and 'Tony Gore, CTU Special Operations'. Below the title is a detailed description of the research findings.

• 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>

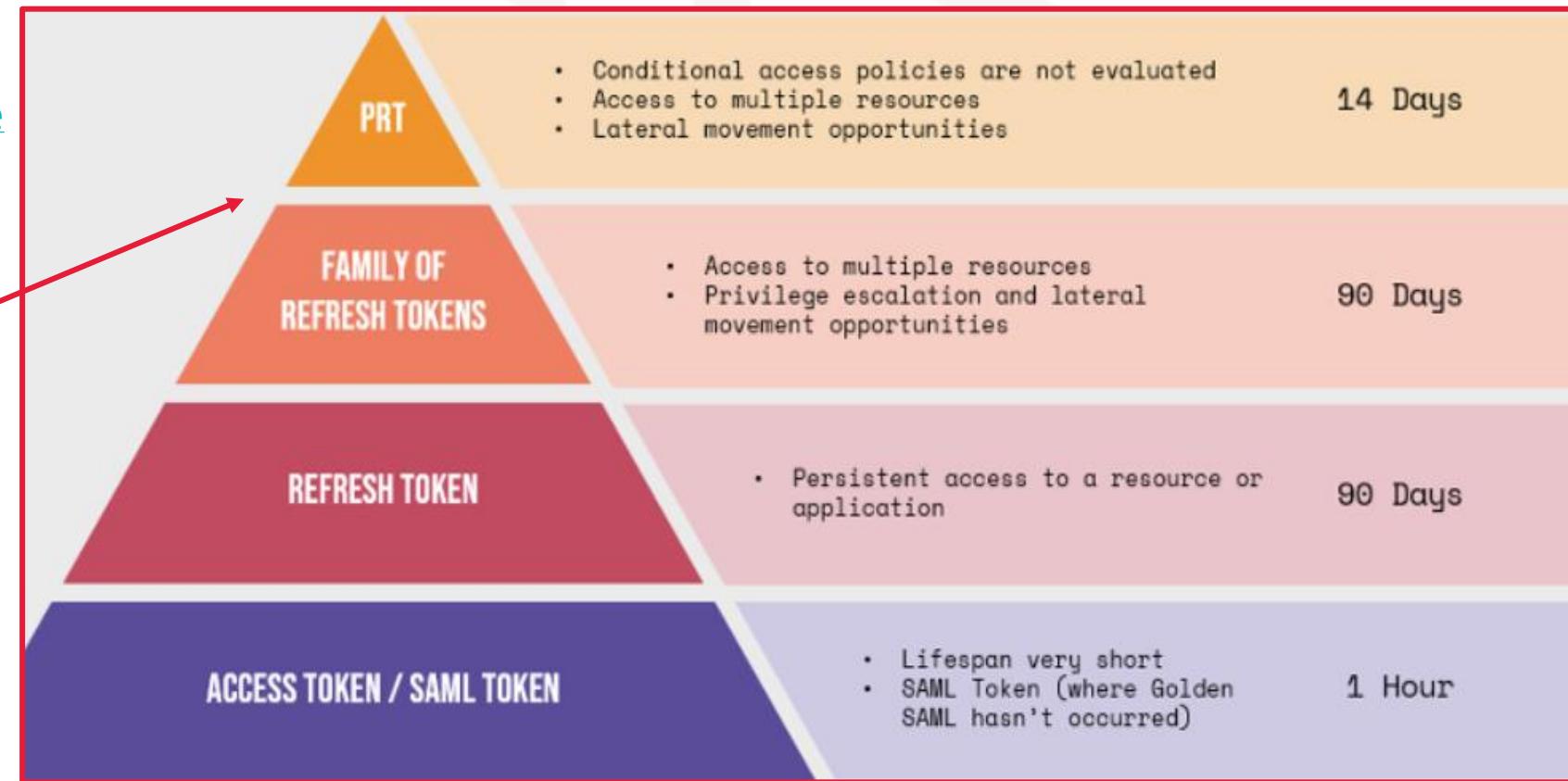
I argue -

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):
    ...
    Selenium login with ESTSAUTH or ESTSAUTHPERSISTENT cookie injection
    ...
def interceptor(request):
```

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

The screenshot shows a browser window with two tabs open. The left tab is for the repository `github.com/zh54321/EntraTokenAid`, which has a dark theme. It displays a **2025-04-11** release note under the **Added** section, which includes a PowerShell command snippet. The right tab is for the repository `github.com/f-bader/TokenTacticsV2`, which also has a dark theme. It displays release notes for versions 0.2.5 and 0.2.6, both dated 2025-01-04. These notes mention fixes for custom scopes and changes to redirect URIs, and add new cmdlets like `Get-AzureAuthorizationCode` and `Get-AzureTokenFromAuthorizationCode`. Both repos link to their respective README files and license information.

2025-04-11

Added

- It is now possible now generate the auth token for Azure AD authentication, copy the URL containing the token.
```powershell  
\$tokens = Invoke-Auth -Manually -Scopes "User.Read" -RedirectUri "https://www.bader.it"  
\$tokens | ConvertTo-Json | Out-File tokens.json  
```

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.

EC

What is the Authorization code flow?

/oauth2/v2.0/**authorize** **response_type=code**



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



Enter (feeling satisfied): Redirected code=<auth_code>



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

What is the Authorization code flow?

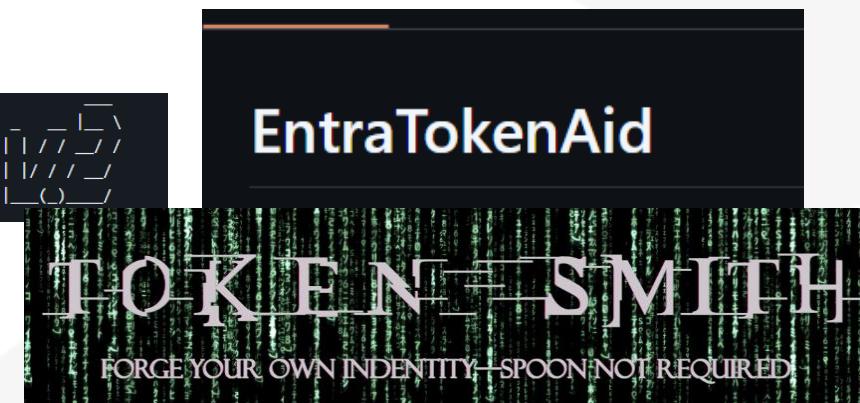
/oauth2/v2.0/**authorize** **response_type=code**

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

↓
Entra: Redirected code=<auth_code>

↓
(THIS PART CAN BE BROKEN UP)

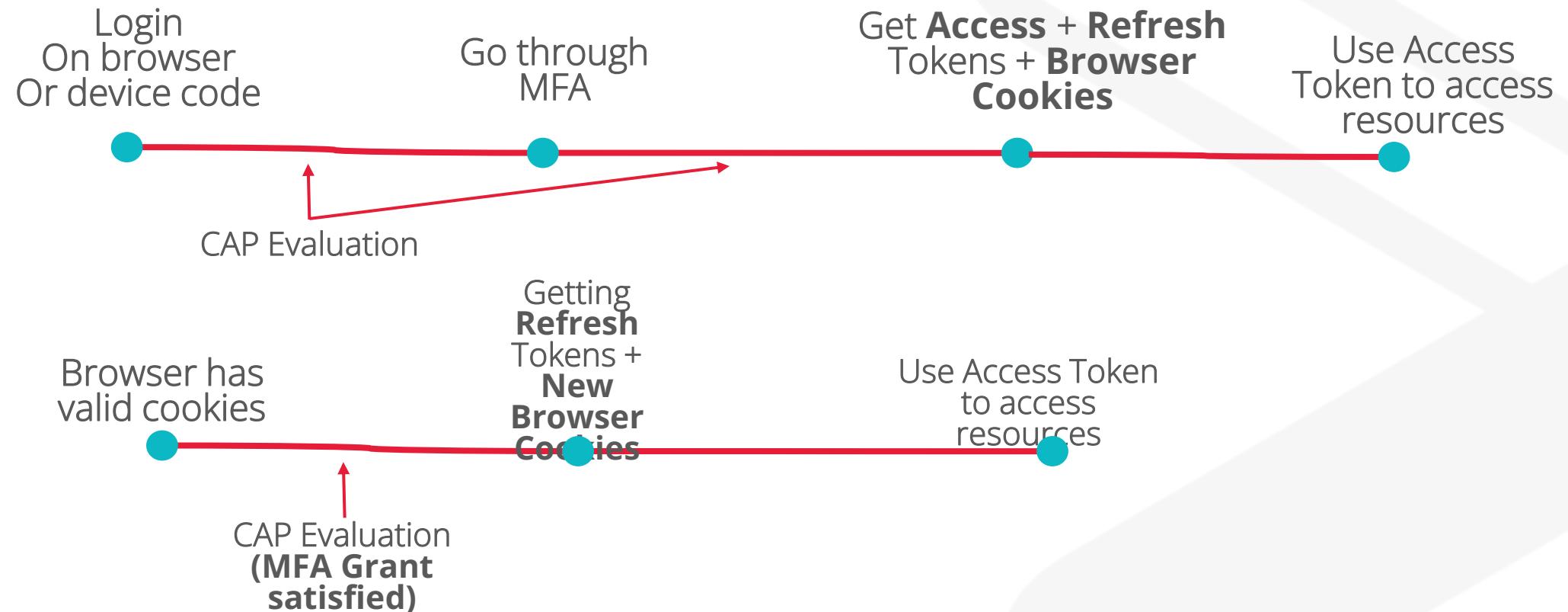
↓
POST to /**token** to redeem **Access & Refresh Tokens**



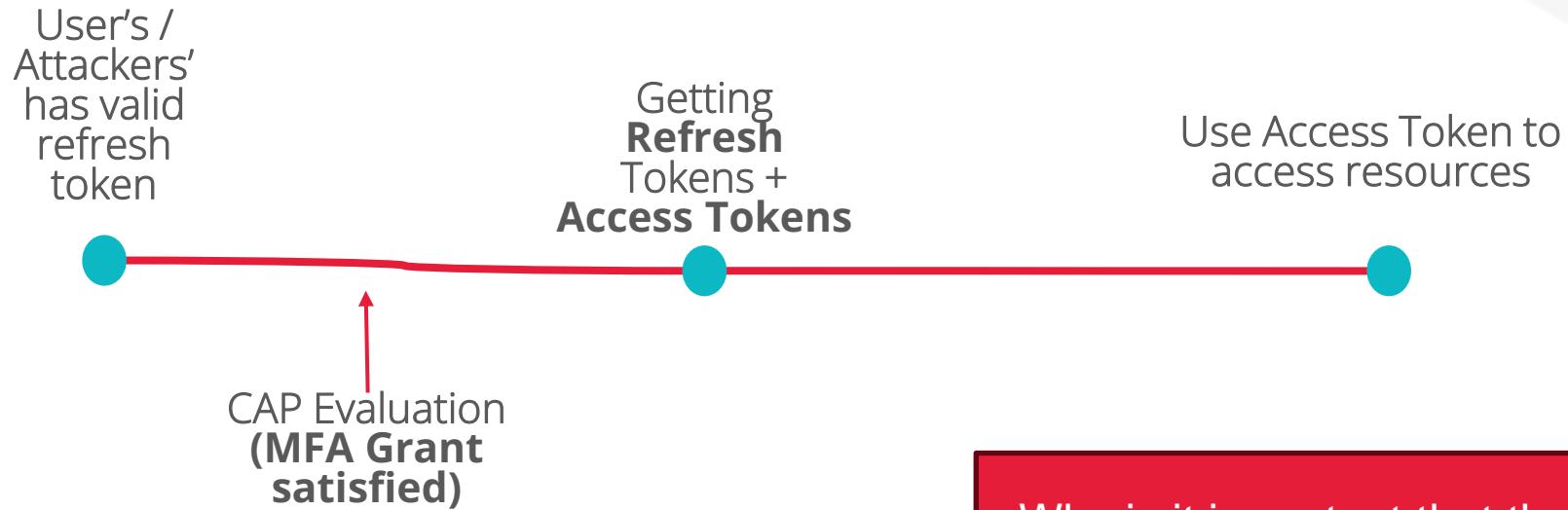
Let's lay some groundwork first -
What is Grant, When is CAP eval'd

One must imagine Entra ID happy?

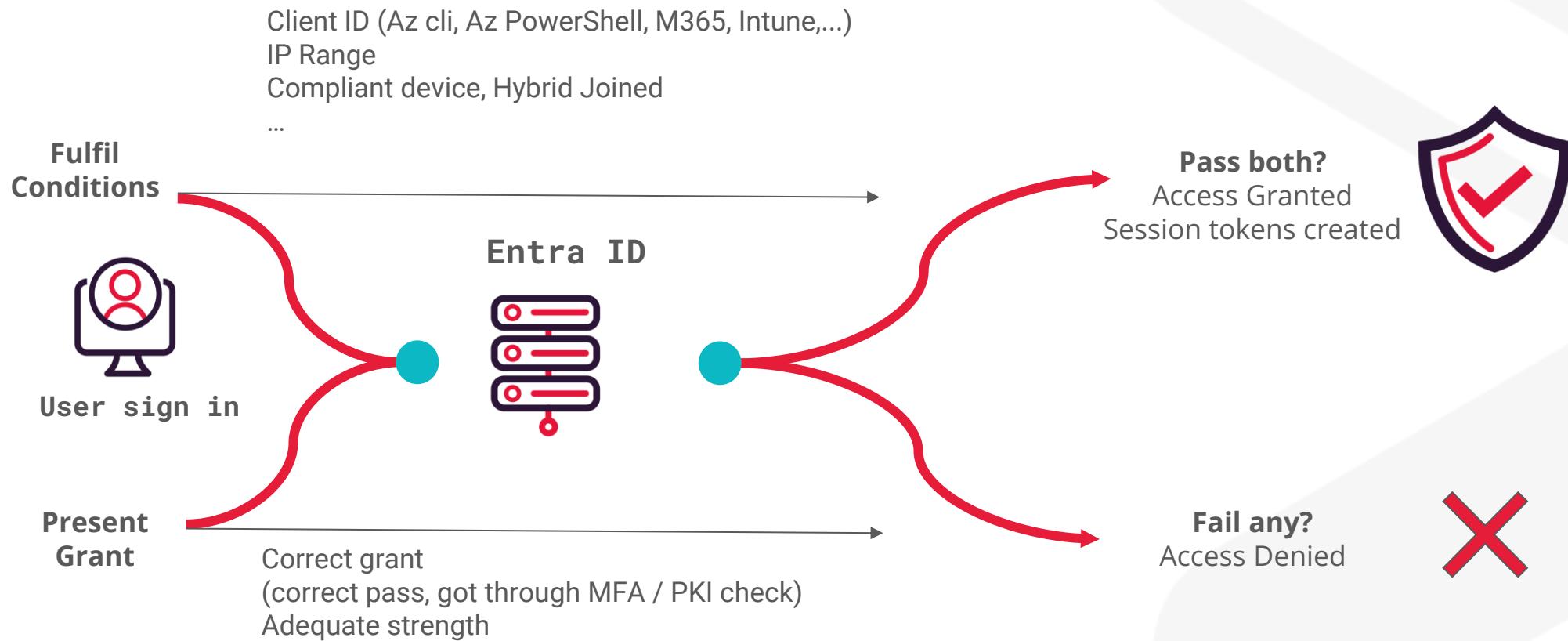
Where are CAPs evaluated? (interactive sign in)



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



Pictorially





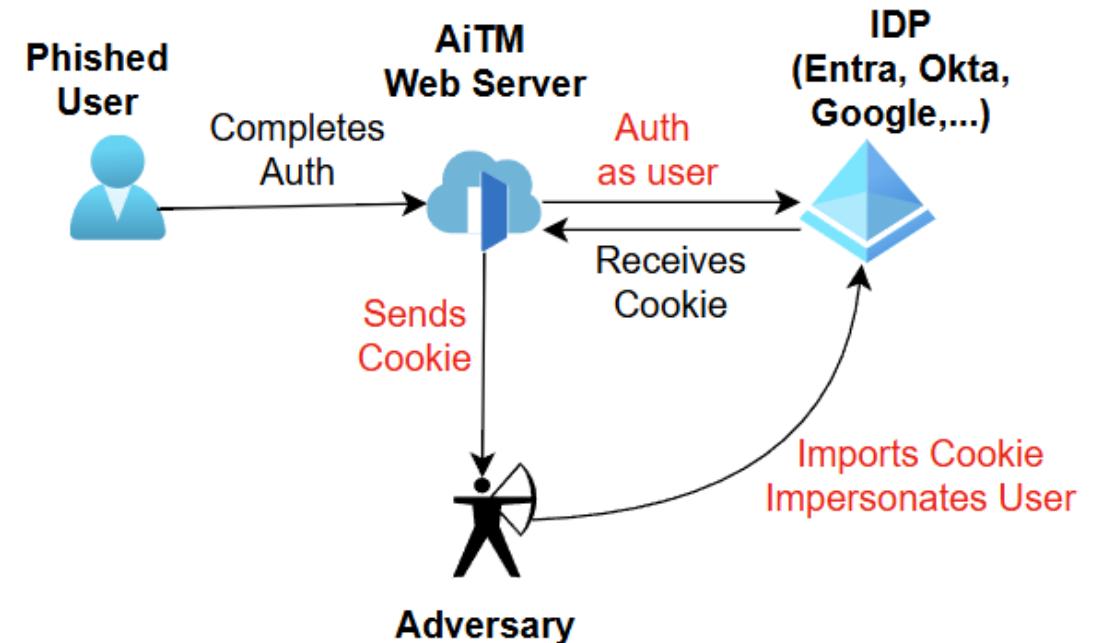
Let's lay some groundwork first -
AiTM Phishing

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

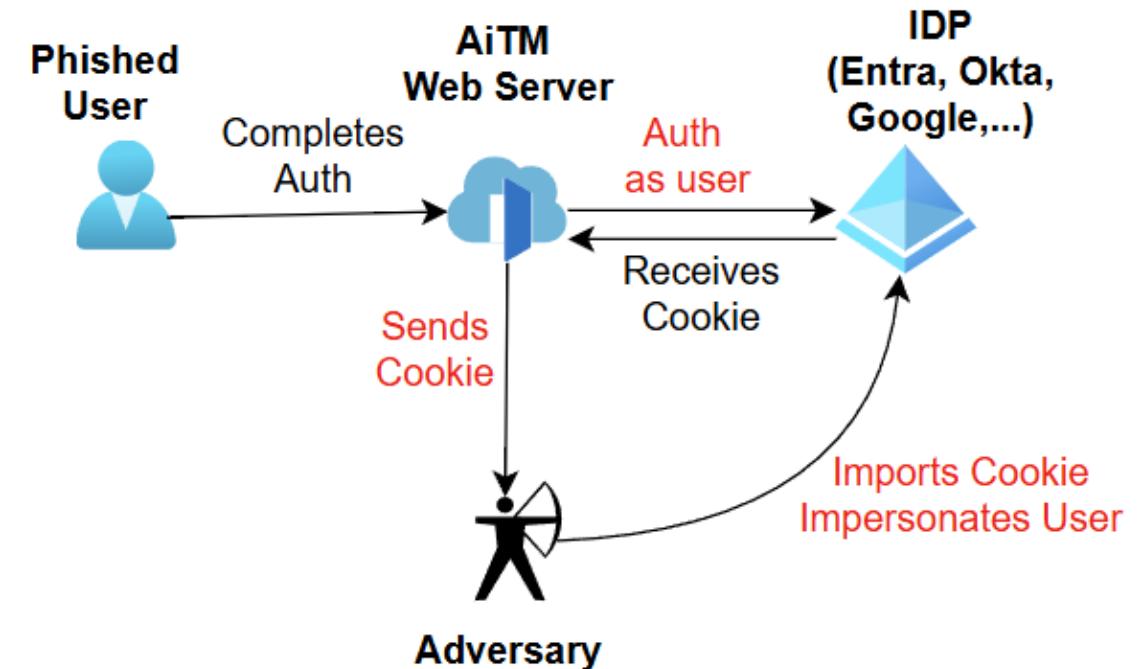


AiTM Flow for Microsoft Entra ID - 2

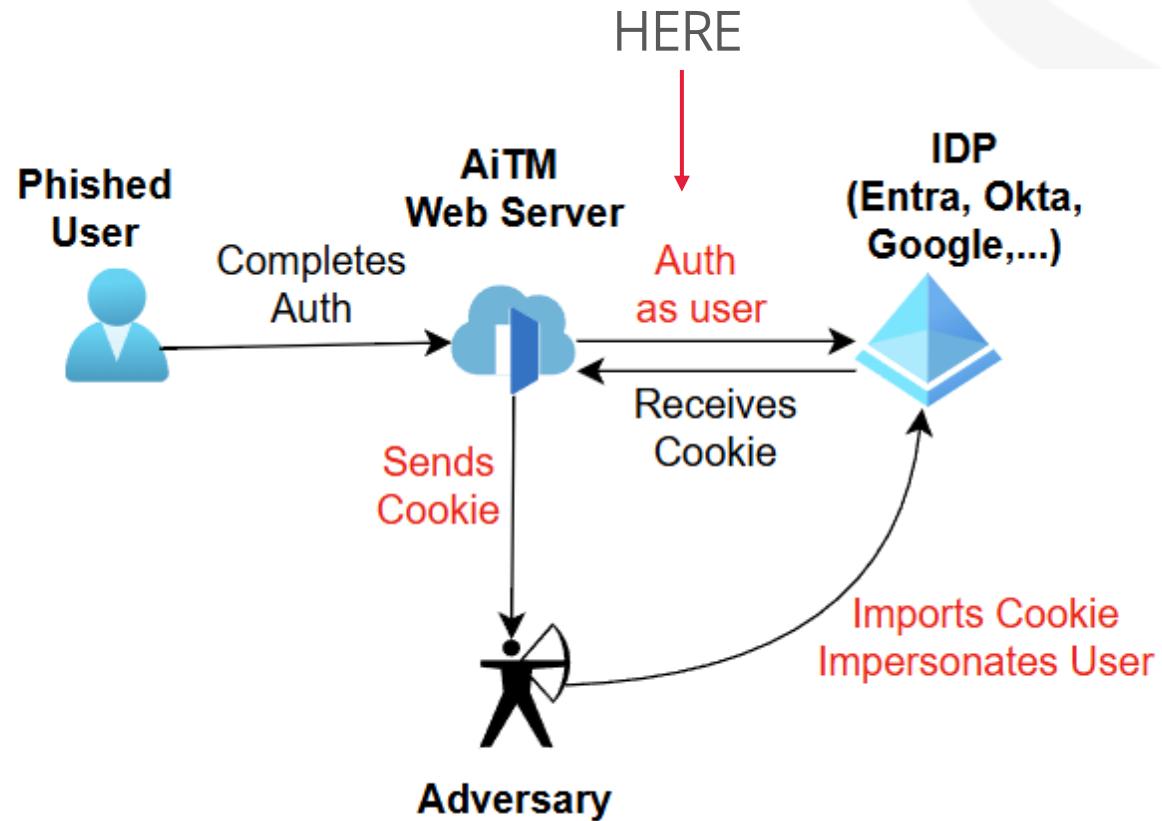
User returned to attacker-controlled redirection site

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 26th

 **PhishingData** APP 1:42 PM

 Pwned - Password received!

User: [derpy.t](#) [REDACTED]

Password: **A password here**

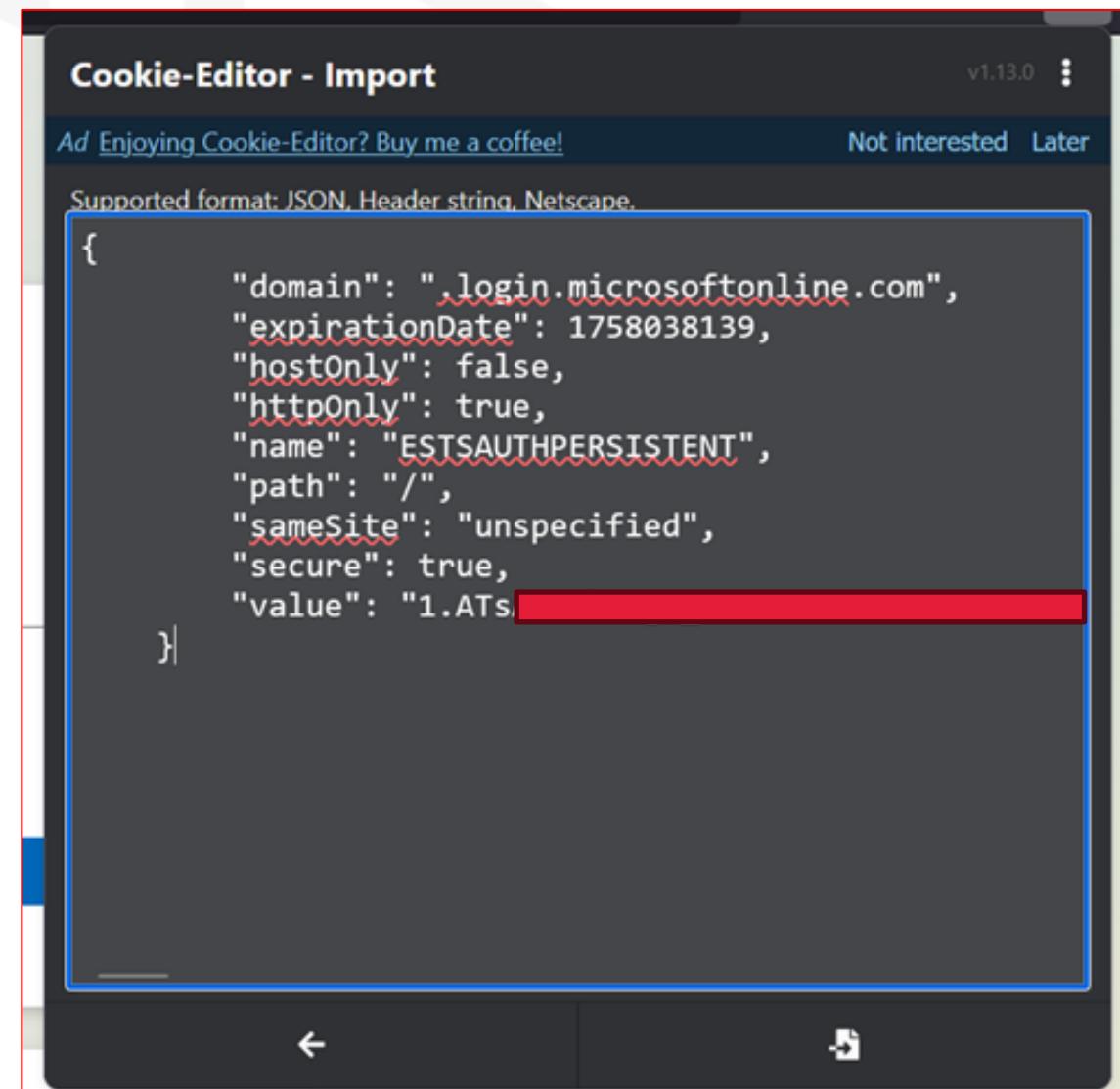
Note: The 1st cookie below is unlikely to provide you access unless they have no 2FA.
The 2nd cookie will contain the 2FA data and the 3rd one the 2FA + 'Stay signed in' data.

 Cookies found!

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

Session Theft

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



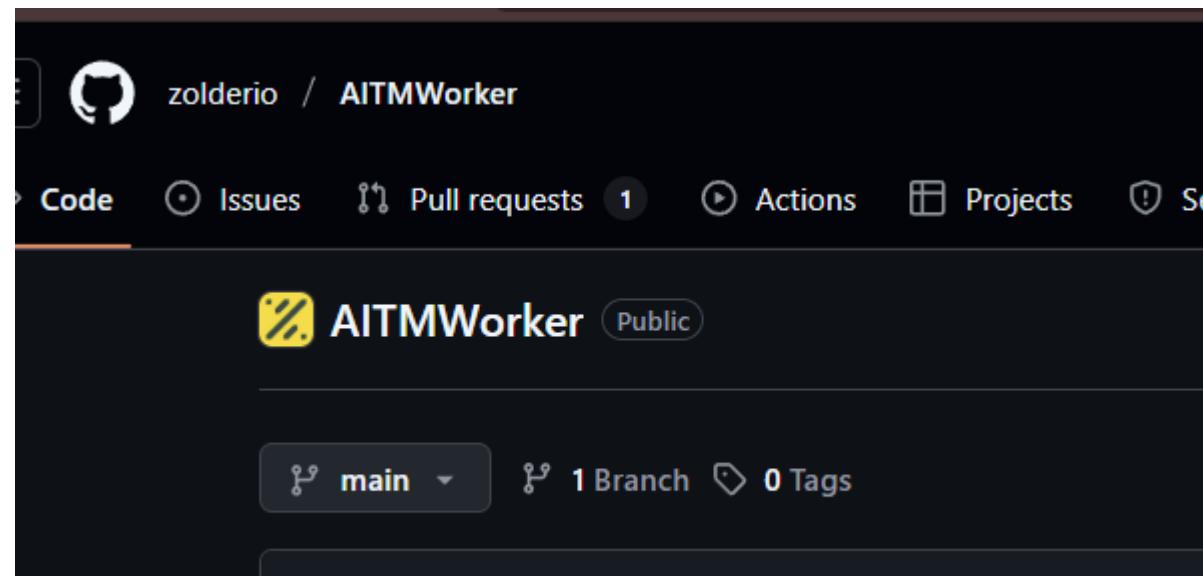
Let's build our AiTM logic

- 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

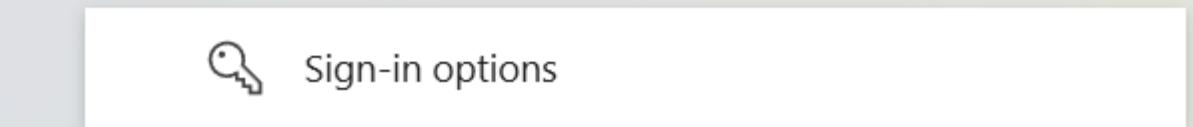
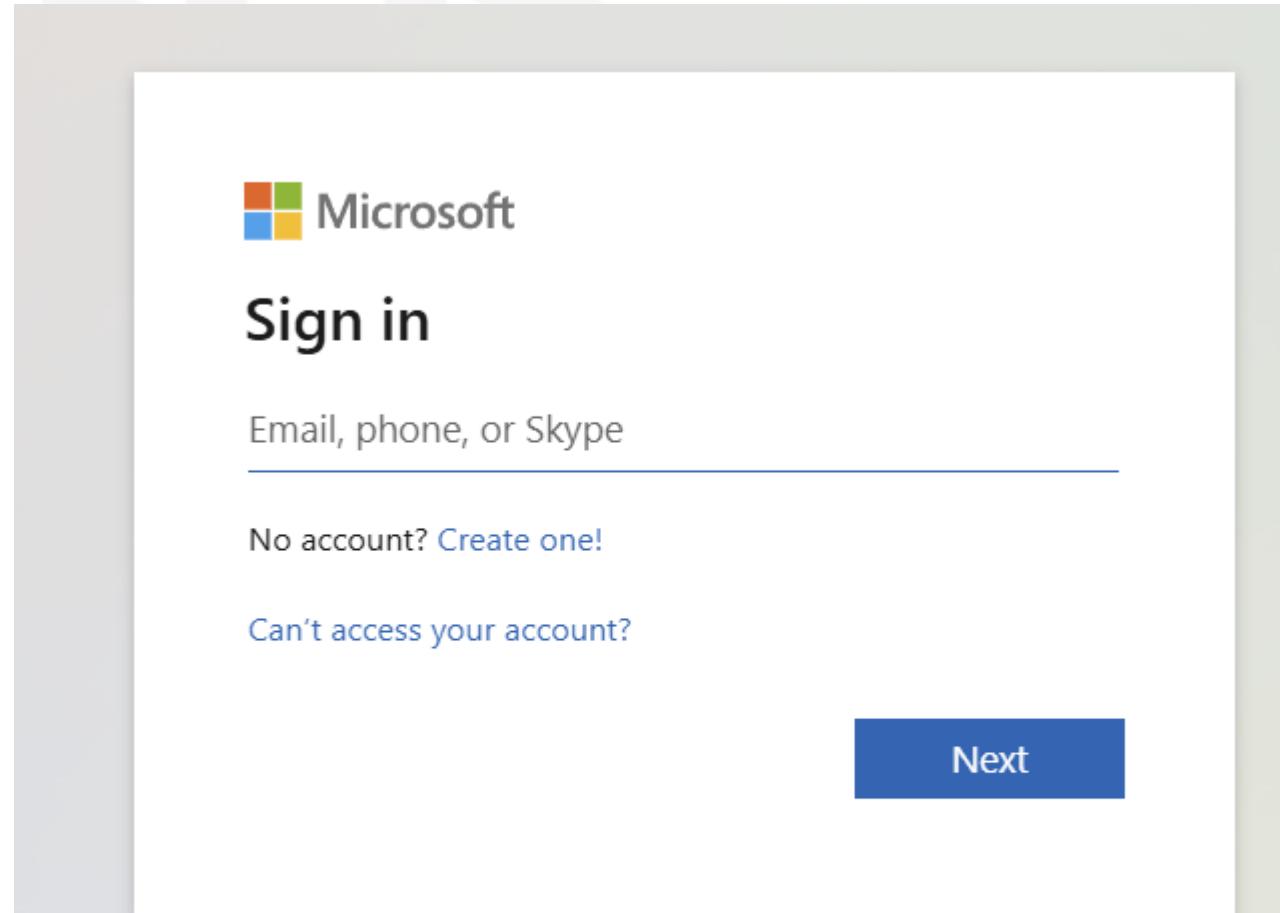
Client Branding..

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

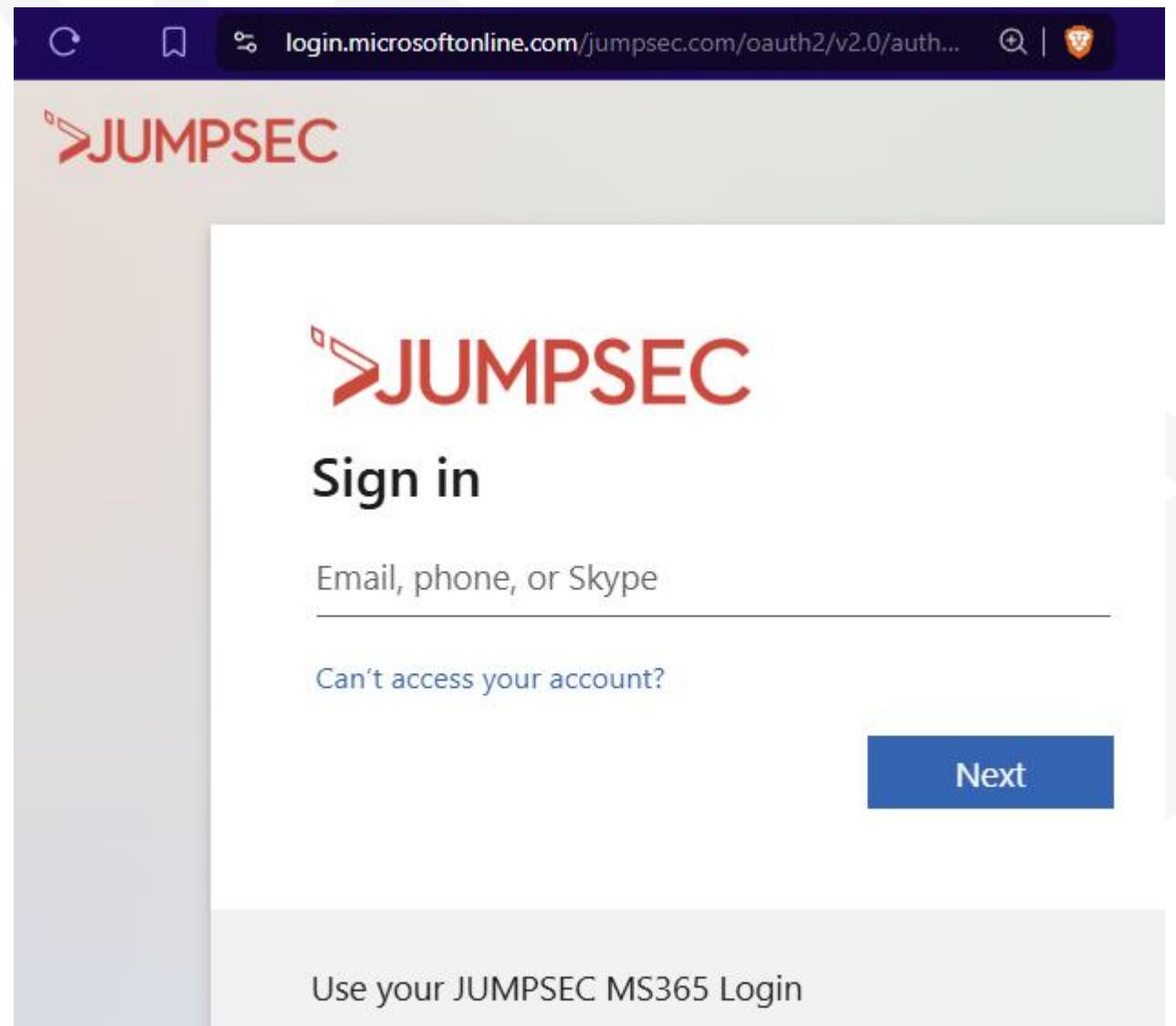
login.microsoftonline.com/common/oauth2/v2.0/authorize...



Trick 1: 'common' trick

Easiest way to do client branding

[login.microsoftonline.com/**client.domain**/oauth2/v2.0/authorize...](https://login.microsoftonline.com/client.domain/oauth2/v2.0/authorize...)



'>JUMPSEC

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 .\TeamFiltration\config\config.json  
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  
uny\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

The screenshot shows two policy configuration pages from Microsoft Intune.

Device platforms page:

- Apply policy to selected device platforms.
- Configure: Yes (selected)
- Include tab selected.
- Any device (radio button selected).
- Select device platform: (radio button unselected)
- Options: Android, iOS, Windows Phone, Windows, macOS, Linux (checkboxes unselected).

Client apps page:

- Control user access to target specific client applications not using modern authentication.
- Configure: Yes (selected)
- Select the client apps this policy will apply to:
 - Modern authentication clients:
 - Browser (checkbox selected)
 - Mobile apps and desktop clients (checkbox selected)
 - Legacy authentication clients:
 - Exchange ActiveSync clients (checkbox selected)
 - Other clients (checkbox selected)

Background text (partially visible):

- utlook PLATFORM: Windows => VALID BUT MFA (76)
- utlook PLATFORM: Windows Phone => VALID BUT MFA (76)
- neNote PLATFORM: Android => VALID BUT MFA (76)
- neNote PLATFORM: iPhone => VALID BUT MFA (76)
- neNote PLATFORM: Mac OS => VALID BUT MFA (76)
- neNote PLATFORM: Linux => VALID BUT MFA (76)
- neNote PLATFORM: Windows => VALID BUT MFA (76)
- Windows Phone => VALID BUT MFA (76)
- FORM: Android => VALID BUT MFA (76)
- FORM: iPhone => VALID BUT MFA (76)
- FORM: Mac OS => VALID BUT MFA (76)
- FORM: Linux => VALID BUT MFA (76)
- FORM: Windows => VALID BUT MFA (76)
- FORM: Windows Phone => VALID BUT MFA (76)
- PLATFORM: Android => VALID BUT MFA (76)
- PLATFORM: iPhone => VALID BUT MFA (76)
- PLATFORM: Mac OS => VALID BUT MFA (76)
- PLATFORM: Linux => VALID BUT MFA (76)
- PLATFORM: Windows => VALID BUT MFA (76)
- PLATFORM: Windows Phone => VALID BUT MFA (76)
- Shell PLATFORM: Android => VALID BUT MFA (76)
- Shell PLATFORM: iPhone => VALID BUT MFA (76)
- Shell PLATFORM: Mac OS => VALID BUT MFA (76)
- Shell PLATFORM: Linux => VALID BUT MFA (76)
- Shell PLATFORM: Windows => VALID BUT MFA (76)
- Shell PLATFORM: Windows Phone => VALID BUT MFA (76)
- PLATFORM: Android => VALID BUT MFA (76)
- PLATFORM: iPhone => VALID BUT MFA (76)
- PLATFORM: Mac OS => VALID BUT MFA (76)

JUMPSEC

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?

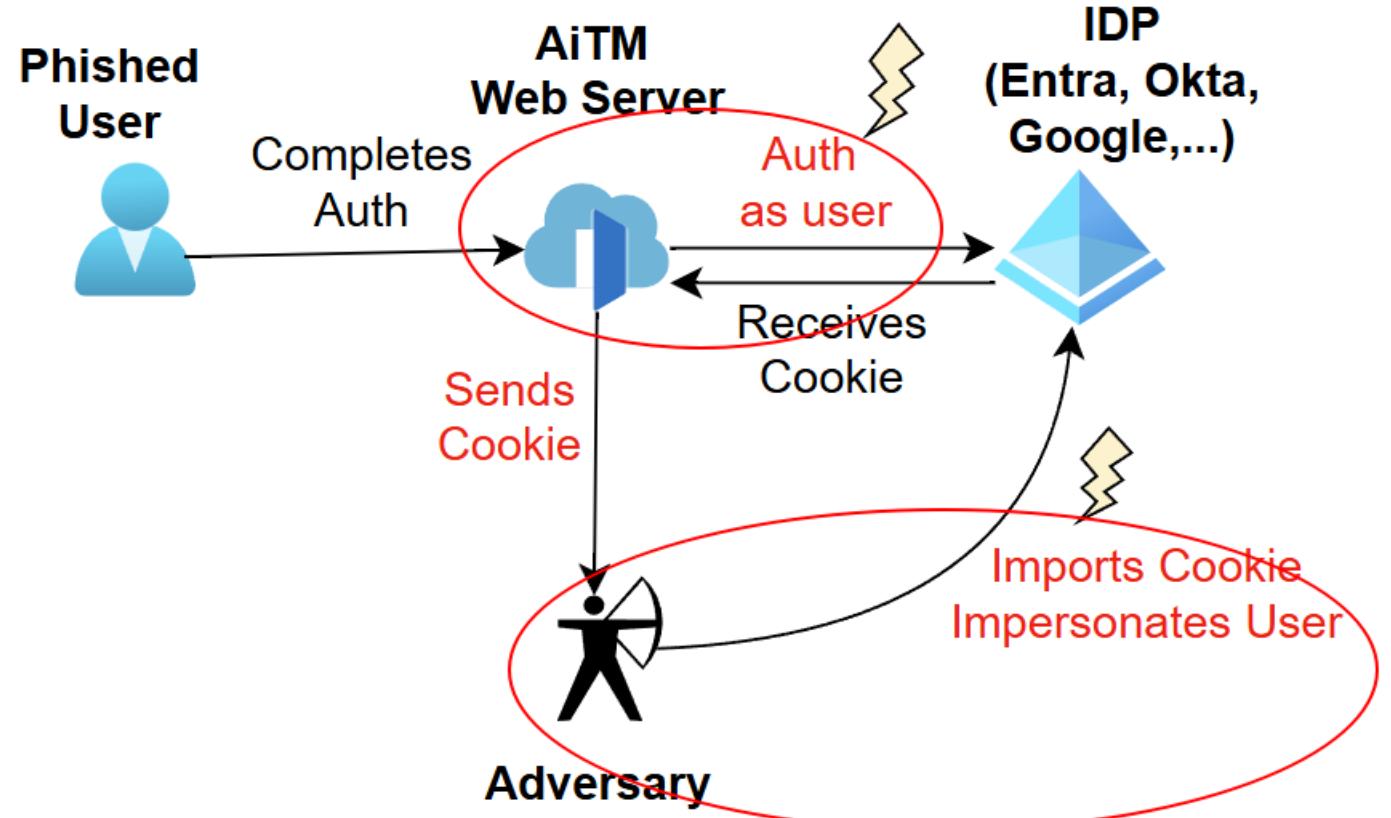
What happens on a RT?

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



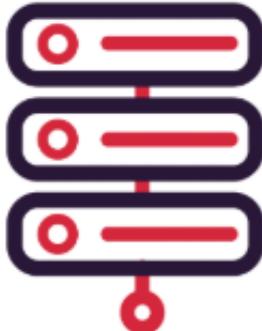
login.microsoft.com/...?redirect_uri=attacker.server.com

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



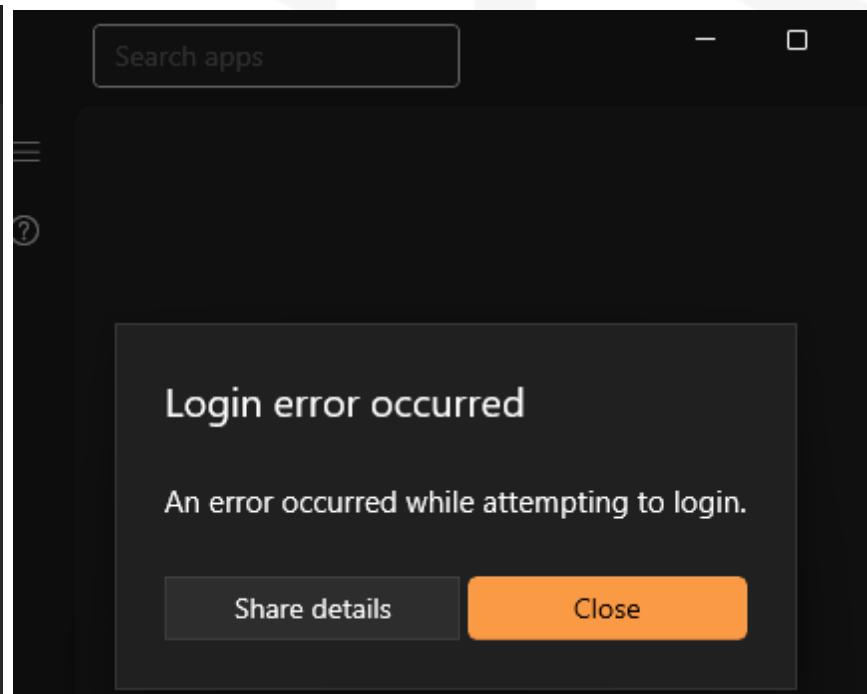
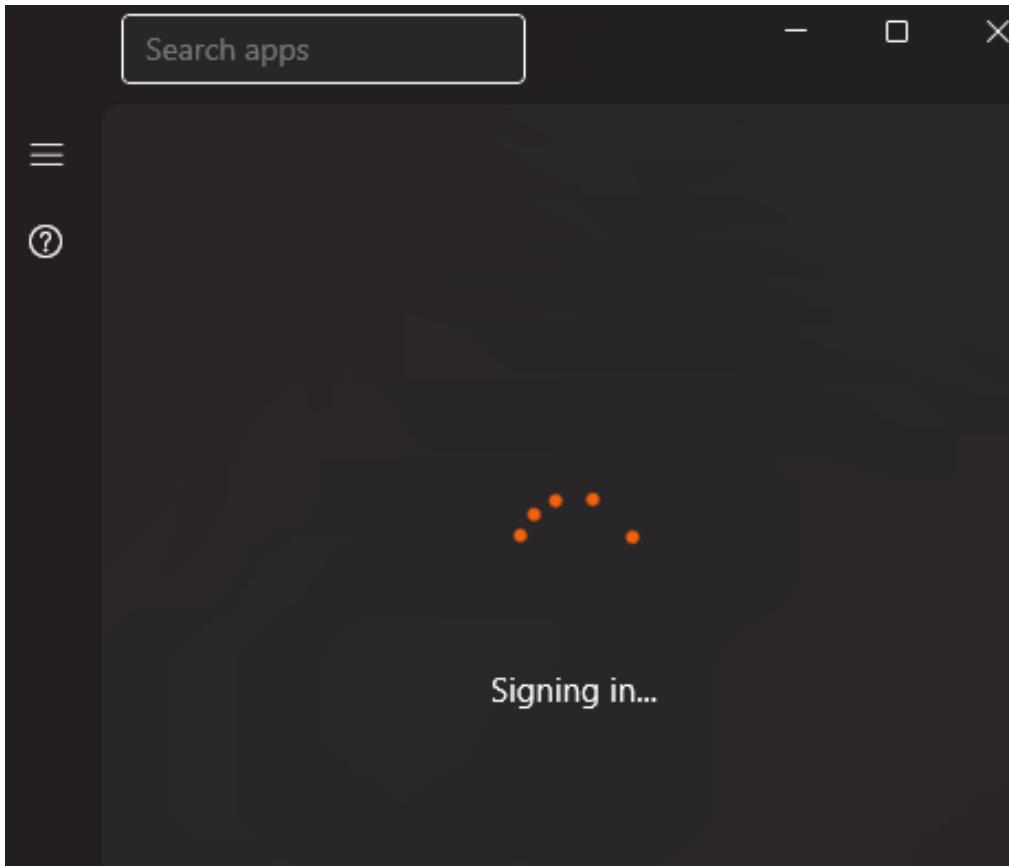
derpy.fonde

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?

anoopcnair.com/fix-intune-company-portal-app-login-issues/

Logged at authorizationclient.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:

XMLHTTPWebRequest::ReceiveResponse.

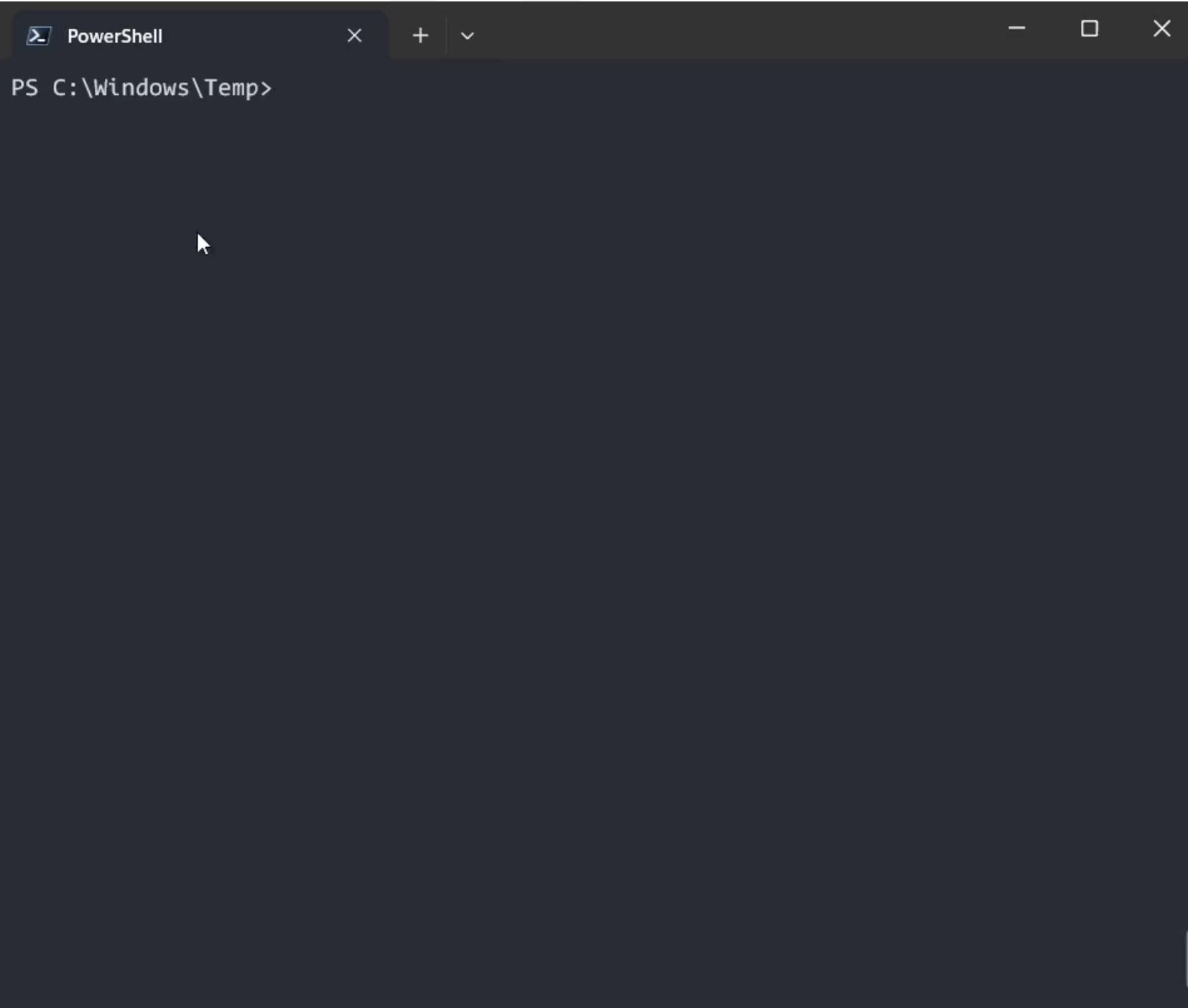
Log: 0xcaa10083 Exception in WinRT wrapper.

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



A screenshot of a PowerShell window titled "PowerShell". The window has a dark theme. At the top, it shows the title bar and a toolbar with standard window controls (minimize, maximize, close). Below the title bar, the command prompt is displayed as "PS C:\Windows\Temp>". The main body of the window is mostly empty, with a small white cursor icon visible near the top center. The background of the slide features a large, semi-transparent watermark of the MPSEC logo.

The shadow patch

- **Roughly 20th 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'

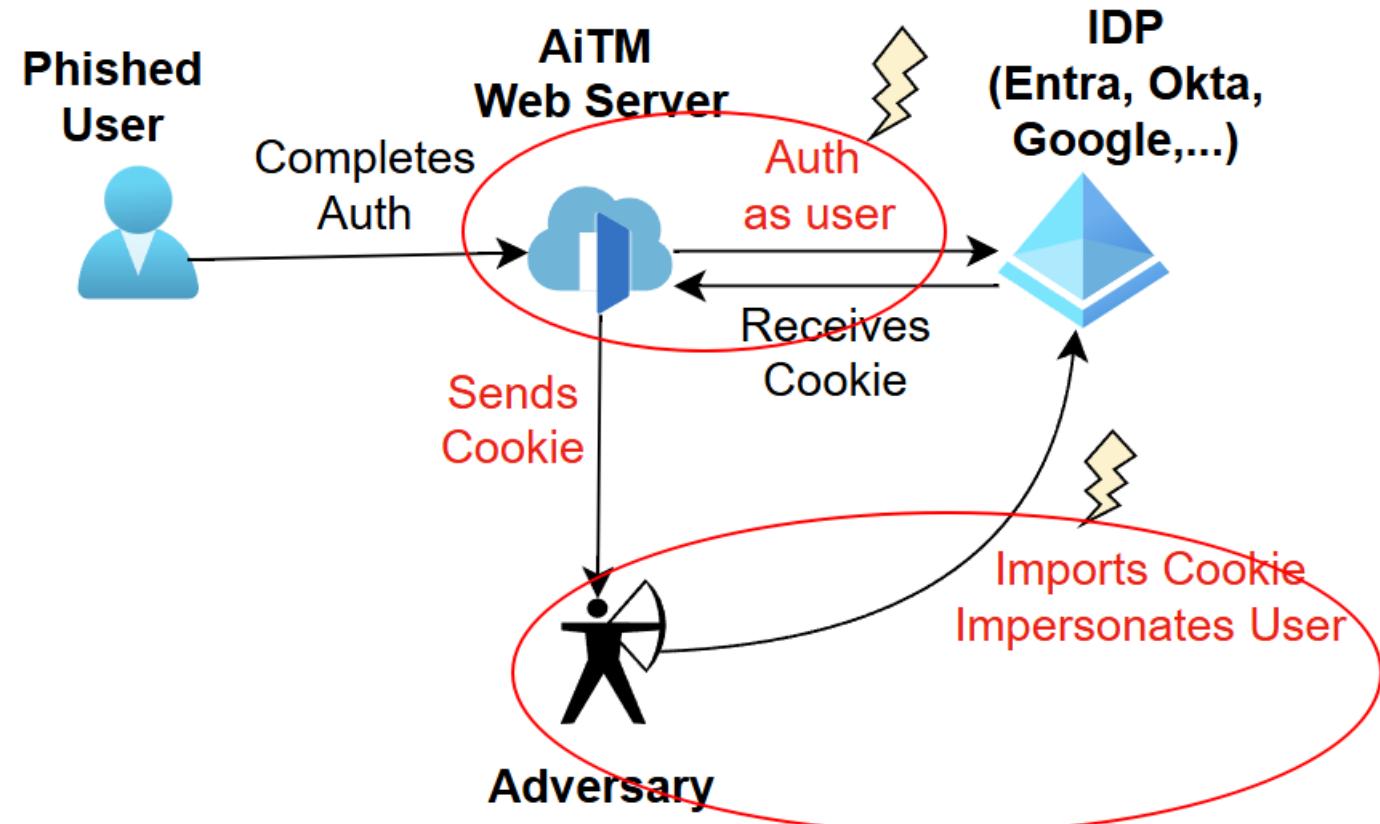
So ... is it now useless?

Remember again

That the
Blank device > Compliant device

Enrollment process

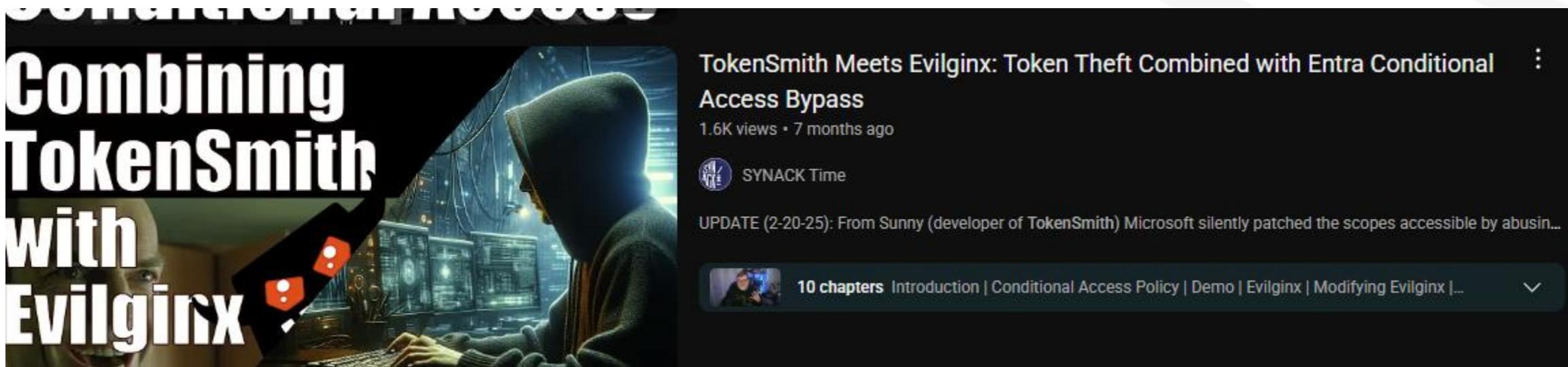
Can never require a Compliant device



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



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 abusing...



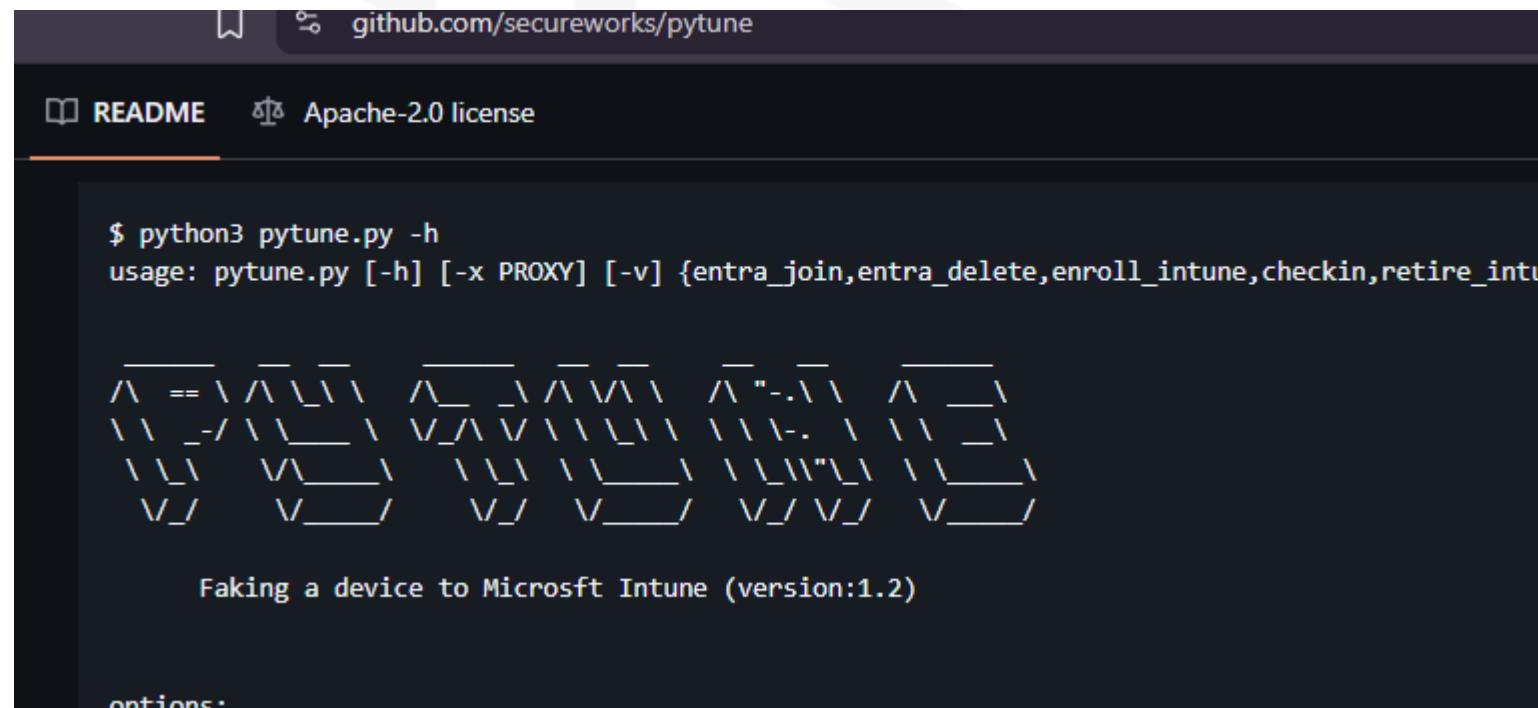
10 chapters Introduction | Conditional Access Policy | Demo | Evilginx | Modifying Evilginx |...

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?)



The screenshot shows a GitHub repository page for 'pytune' by 'secureworks'. The 'README' tab is selected. It contains a command-line usage example: '\$ python3 pytune.py -h' followed by 'usage: pytune.py [-h] [-x PROXY] [-v] {entra_join,entra_delete,enroll_intune,checkin,retire_intu...'. Below this is a large ASCII art logo consisting of various symbols like '^', '=', ']', 'V', '(', ')', and '|'. At the bottom of the logo, the text 'Faking a device to Microsoft Intune (version:1.2)' is displayed.

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.

```
L$ 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

