Continuous Access Evasion
Evading Microsoft CAE

Nikhil Mittal

Founder, Altered Security

About me

- Twitter - @nikhil_mitt
- Founder of Altered Security - alteredsecurity.com
- GitHub - github.com/samratashok
- Creator of Nishang, RACE toolkit and more
- Interested in Active Directory, Offensive PowerShell, Azure Red Teaming and Enterprise Security.
- Previous Talks and/or Trainings
  - DEF CON, BlackHat, BruCON, AltSecCON and more.

Altered Security

- Trained more than 50000 security professionals from more than 130 countries!
- Our Red Team Labs Platform enables labs to be:
  - Affordable
  - Easy to Access
  - Stable and provide great user experience
  - Fun to Solve
  - Big enough to feel enterprise-like

ALTERED SECURITY

Red team labs	alteredsecurity.com/online-labs
Instructor-led bootcamps	alteredsecurity.com/bootcamps
GitHub	github.com/AlteredSecurity
Lab Platform	enterprisesecurity.io
Free Labs and Challenges	redlabs.enterprisesecurity.io

Agenda

- Understanding Microsoft's attempt to arrest token replay attacks using CAE.
- Analysis of CAE against well-known token stealing techniques.
- Conclude that we are not impressed :D

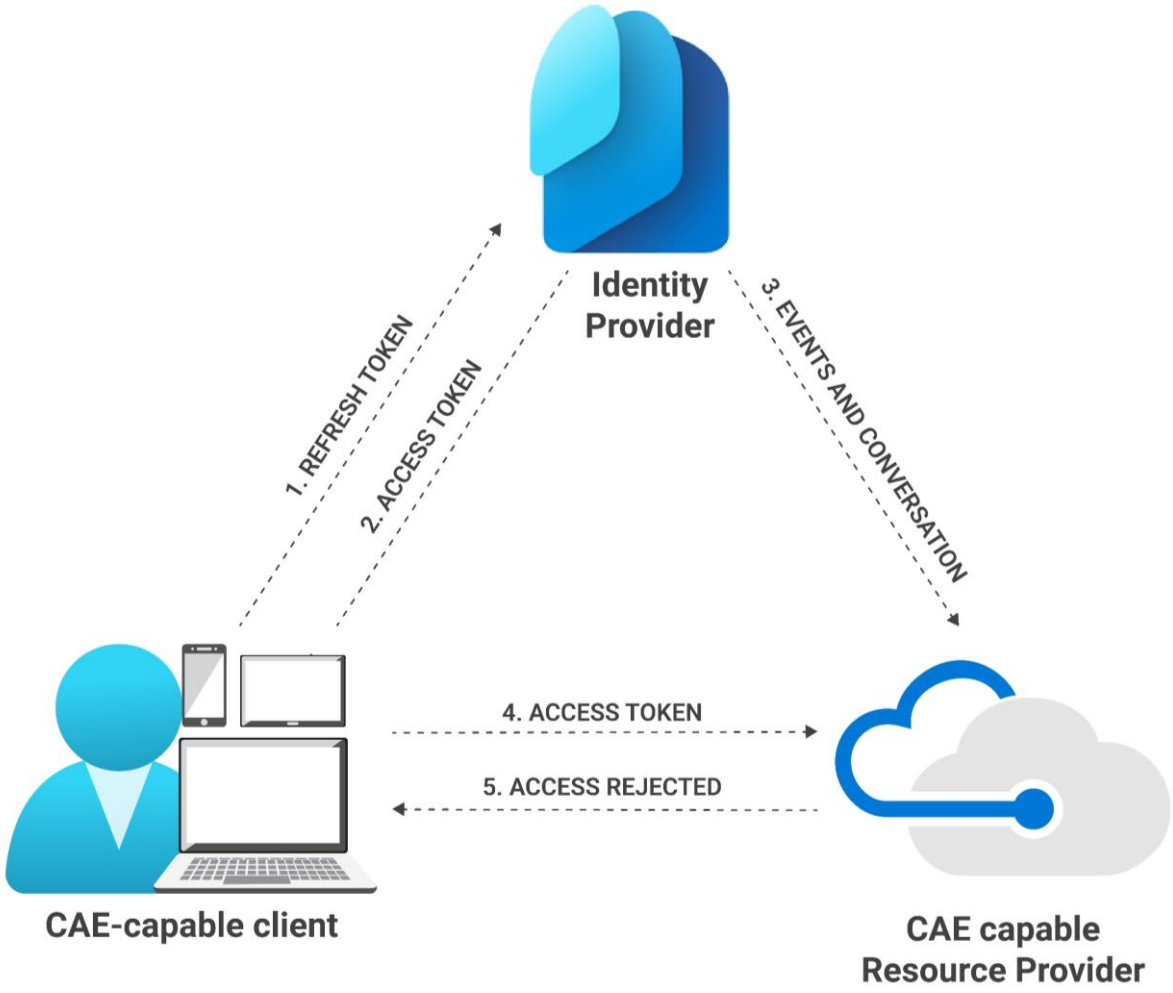
What is CAE?

- Continuous Access Evaluation (CAE) is a security control by Microsoft to revoke access token in near real time.
- CAE enables revocation of access token when certain scenarios and events are triggered.
- The goal of CAE is to restrict access token replay.

Why CAE?

- Access token replay is a potent attack and a headache to check.
- As use of access token is not an authentication activity, there are many benefits:
  - No sign-in logs
  - No Conditional Access enforcement
  - No Risk based checks or scoring by Entra ID Protection
  - With exceptions, MFA claims in the token (allowing MFA evasion)
- CAE helps in addressing this.

How CAE works?



How CAE works?

- A token claim "xms_cc" with the value "CP1" is used by both the client and the resource server to indicate support for CAE.
- The access token lifetime is supposed to increase up to 28 hours.
- Both Client and Resource Server must be CAE capable.

```
"xms_cc": [
  "CP1"
],
```

Triggers

- Critical Event Evaluation
  - User Account is deleted or disabled
  - Password for a user is changed or reset
  - Multifactor authentication is enabled for the user
  - Administrator explicitly revokes all refresh tokens for a user
  - High user risk detected by Microsoft Entra ID Protection
- Conditional Access Policy Evaluation
  - Triggered on change of location

CAE in action

- Attacker steals an access token for Graph API.
- Uses it from a new location using Az PowerShell.
- CAE steps in.

```
Az.MSGraph.internal\Get-AzADUser : Continuous access evaluation resulted in challenge with result:
InteractionRequired and code: LocationConditionEvaluationSatisfied
At C:\Program Files\WindowsPowerShell\Modules\Az.Resources\6.1.0\MSGraph.Autorest\custom\Get-AzADUser.ps1:199
char:9
+         Az.MSGraph.internal\Get-AzADUser @PSBoundParameters
+         ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : InvalidOperation: ({ ConsistencyLe...= , Expand =  }:<>f__AnonymousType5`7) [Get-Az
   ADUser_List], Exception
    + FullyQualifiedErrorId : InvalidAuthenticationToken,Microsoft.Azure.PowerShell.Cmdlets.Resources.MSGraph.Cmd
   lets.GetAzADUser_List
```

Stealing Access Tokens - Tradecraft

- Let's check different techniques of stealing access tokens and see if we get a CAE capable token.
- Each technique will be tested against Identity (Graph API), Management Plane (ARM API) and Data Plane (Key Vault and Storage).

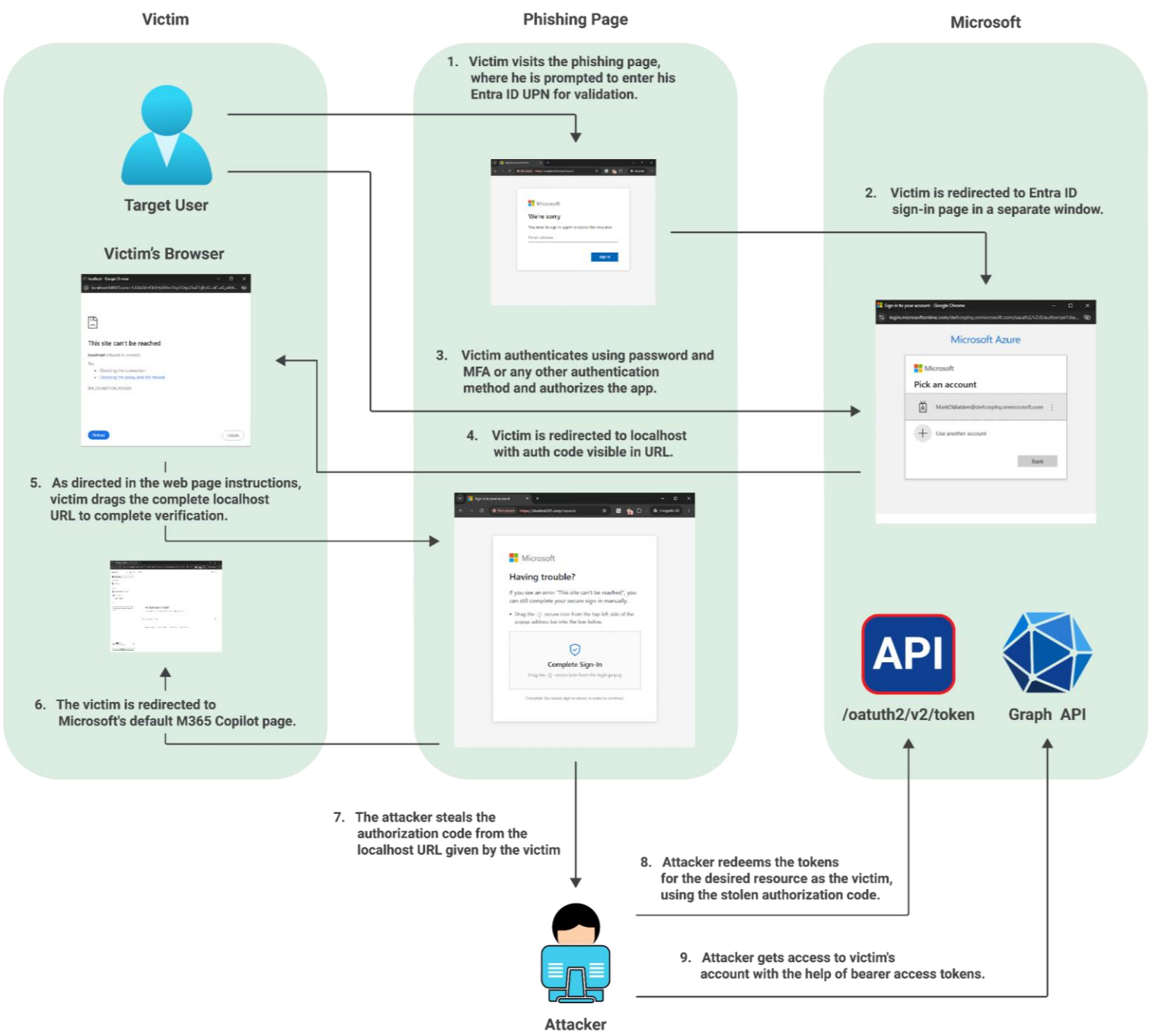
Research Methodology

- The focus of the research is on access tokens. Refresh tokens are assumed to be OPSEC unsafe.
- I wanted to find out if tokens are CAE-capable when we steal them and not when we request them with refresh tokens.

Tradecraft - ConsentFix

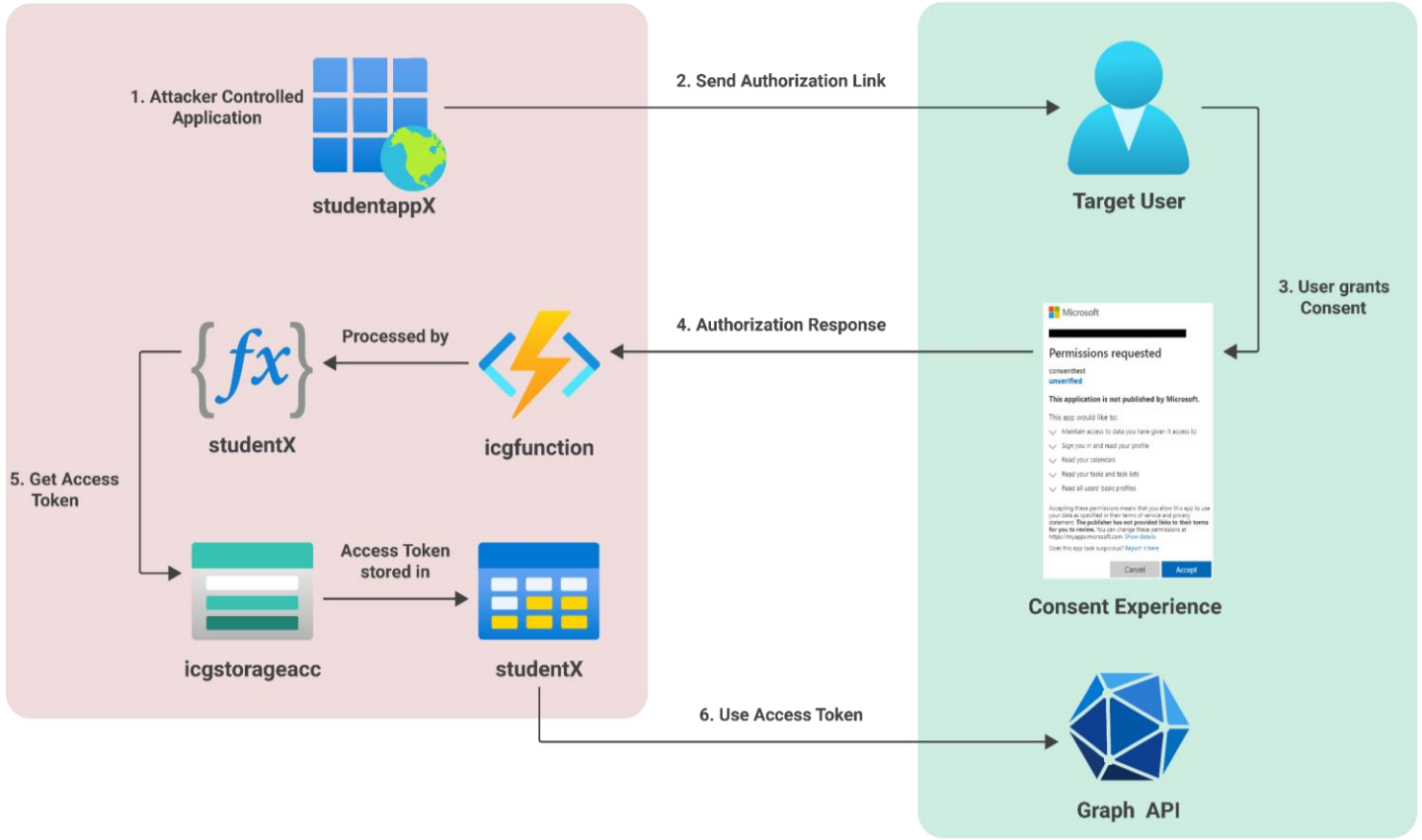
- A variant of the infamous ClickFix attack.
- Trick a victim in sharing Authorization URL resulting in refresh and access token compromise.
- Evades Conditional Access and MFA.

https://pushsecurity.com/blog/consentfix


Victim
Phishing Page
Microsoft
1. Victim visits the phishing page, where he is prompted to enter his Entra ID UPN for validation.
Target User
Victim's Browser
2. Victim is redirected to Entra ID sign-in page in a separate window.
Microsoft Azure
Pick an account
3. Victim authenticates using password and MFA or any other authentication method and authorizes the app.
4. Victim is redirected to localhost with auth code visible in URL.
5. As directed in the web page instructions, victim drags the complete localhost URL to complete verification.
Having trouble?
Complete Sign-In
6. The victim is redirected to Microsoft's default M365 Copilot page.
API
/oatuth2/v2/token
Graph API
7. The attacker steals the authorization code from the localhost URL given by the victim
8. Attacker redeems the tokens for the desired resource as the victim, using the stolen authorization code.
9. Attacker gets access to victim's account with the help of bearer access tokens.
Attacker

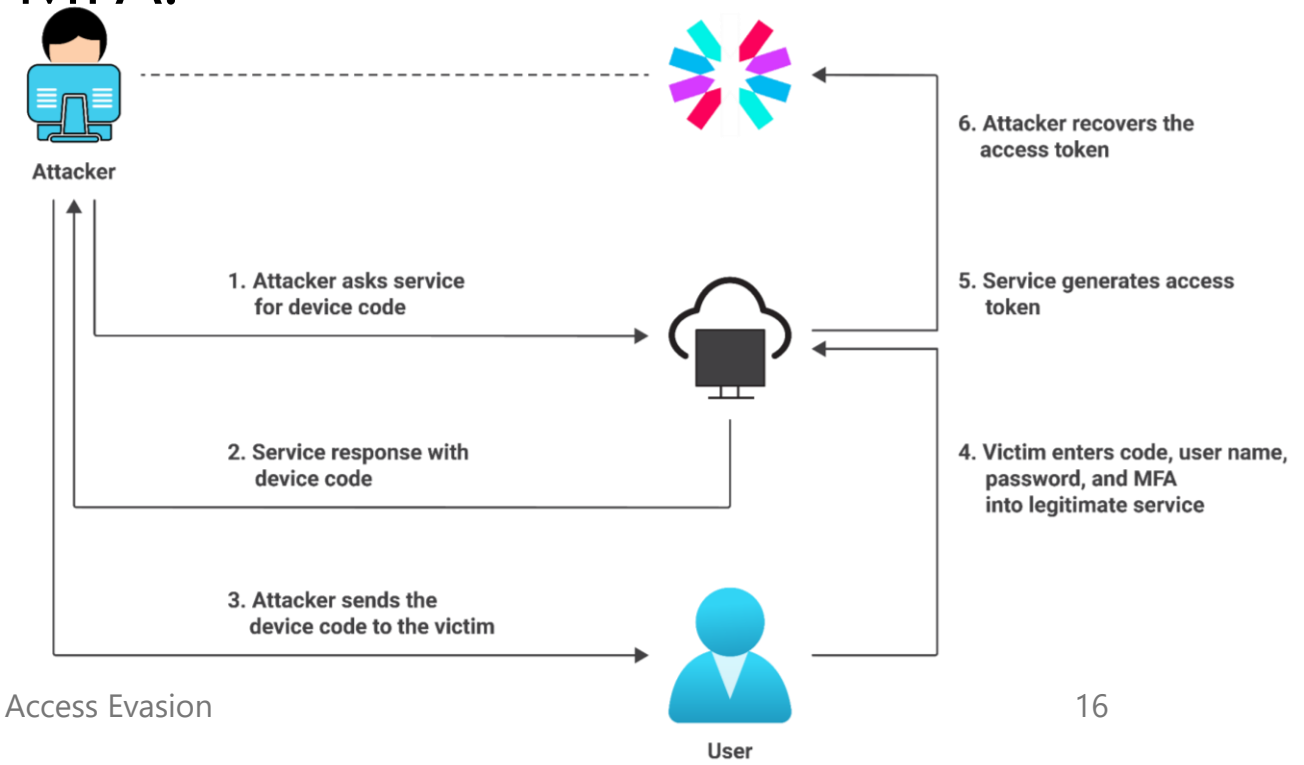
Tradecraft - Illicit Consent Grant

- Trick a user in consenting to malicious Entra ID application. More effective with admin consent.
- Evades Conditional Access and MFA.



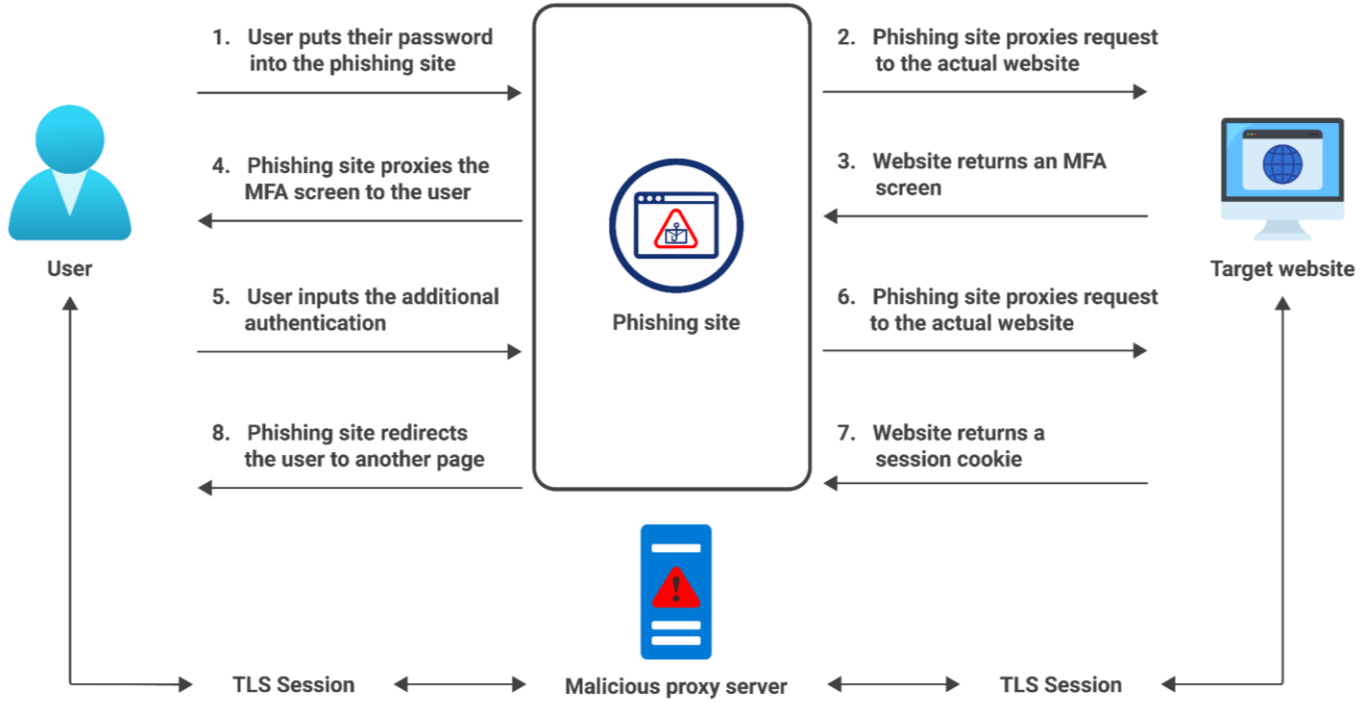
Tradecraft - Device Code Phishing

- Trick a user in using attacker generated Device Authorization code.
- Evades Conditional Access and MFA.



Tradecraft - AiTM

- Attacker sits between the victim and legit website, relays credentials, non-phishing-resistant MFA and captures credentials + authentication cookies.



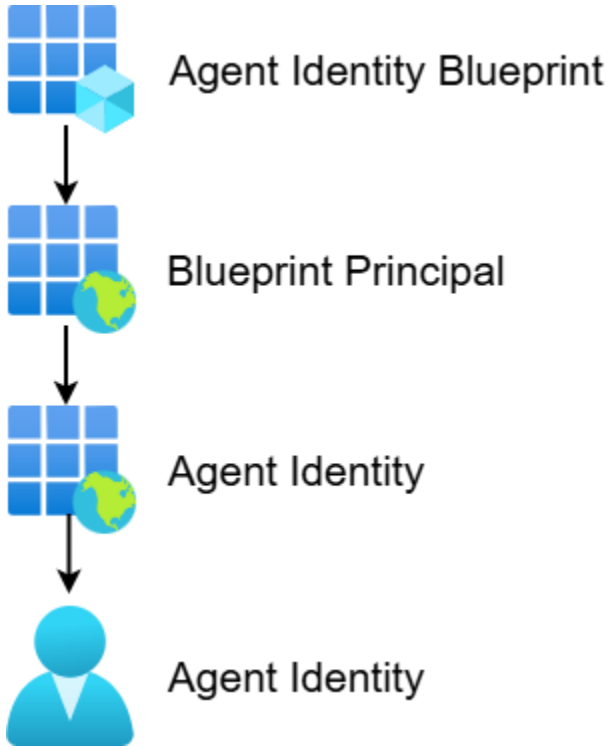
Tradecraft - FOCI and BroCI Abuse

- Family Of Client IDs (FOCI) is a set of "First-Party" Microsoft clients that are compatible with each other.
- Refresh token for a client can be exchanged for tokens for any other client applications in the family.
- For example, a token for MS Office can be used to request a token for Teams.
- Similarly, certain applications (like Azure Portal) can act as a broker to get tokens for some other applications - BroCI/NAA.

Tradecraft - Agent Identity Abuse

- Three usable agent identities in Entra ID derived from Agent Blueprint.

- By adding a client secret (Blueprint credential) to the Blueprint, we can request tokens for **all** the identities associated with it.



Tradecraft - PRT Abuse

- Primary Refresh Token (PRT) is a special type of refresh token to enable SSO.
- If we compromise PRT for a user, it can be used to request refresh and access tokens (Pass-the-PRT).
- PRT has the capability to request refresh and access token for any client and any resource.

Tradecraft - Token from Browser, Disk or Desktop Apps

- A token can be stolen from Browser cookies (ESTSAUTH*).
- From Disk '%USERPROFILE%\.Azure' (by Az PowerShell and az cli) and '%LOCALAPPDATA%\Microsoft\TokenBroker\Cache' (by Office Apps and Teams) - DPAPI protected but extractable from user session.
- Memory of Desktop Applications (like Office Apps and Teams).
- Traffic Interception using tools like mitmdump.

Tradecraft - Token from Active Session

- If we compromise a machine where a user has active logon using CLI tools like Az PowerShell or az cli, it is possible simply request access tokens using that login.
- Additionally, if someone uses an access token with Az PowerShell to connect to a tenant (and doesn't logout), it is cached in clear-text in '%USERPROFILE%\.Azure\AzureRmContext.json'. This is specially useful for access tokens for Workload Identities as they have longer lifetime.

Stealing Access Tokens - Tradecraft Summary

Technique Name	Graph token CAE-cabaple?	ARM and Data Plane Tokens CAE-capable?
ConsentFix / AuthCodeFix	No	No
Illicit Consent Grant	No	No
Device Code Phishing	No	No
AiTM	No	No
Agent Identity Abuse	No	No
Primary Refresh Token Abuse	No	No
FOCI and BroCI Abuse	No	No
Token Stolen from Browser or Disk	No	No
Token Stolen from Desktop Apps (Office, Teams)	**Yes* (in some cases)**	No
Token Stolen from Active Session (az cli, Az PowerShell and Mg Module)	**Yes**	No

Evasion for Location Event Evaluation

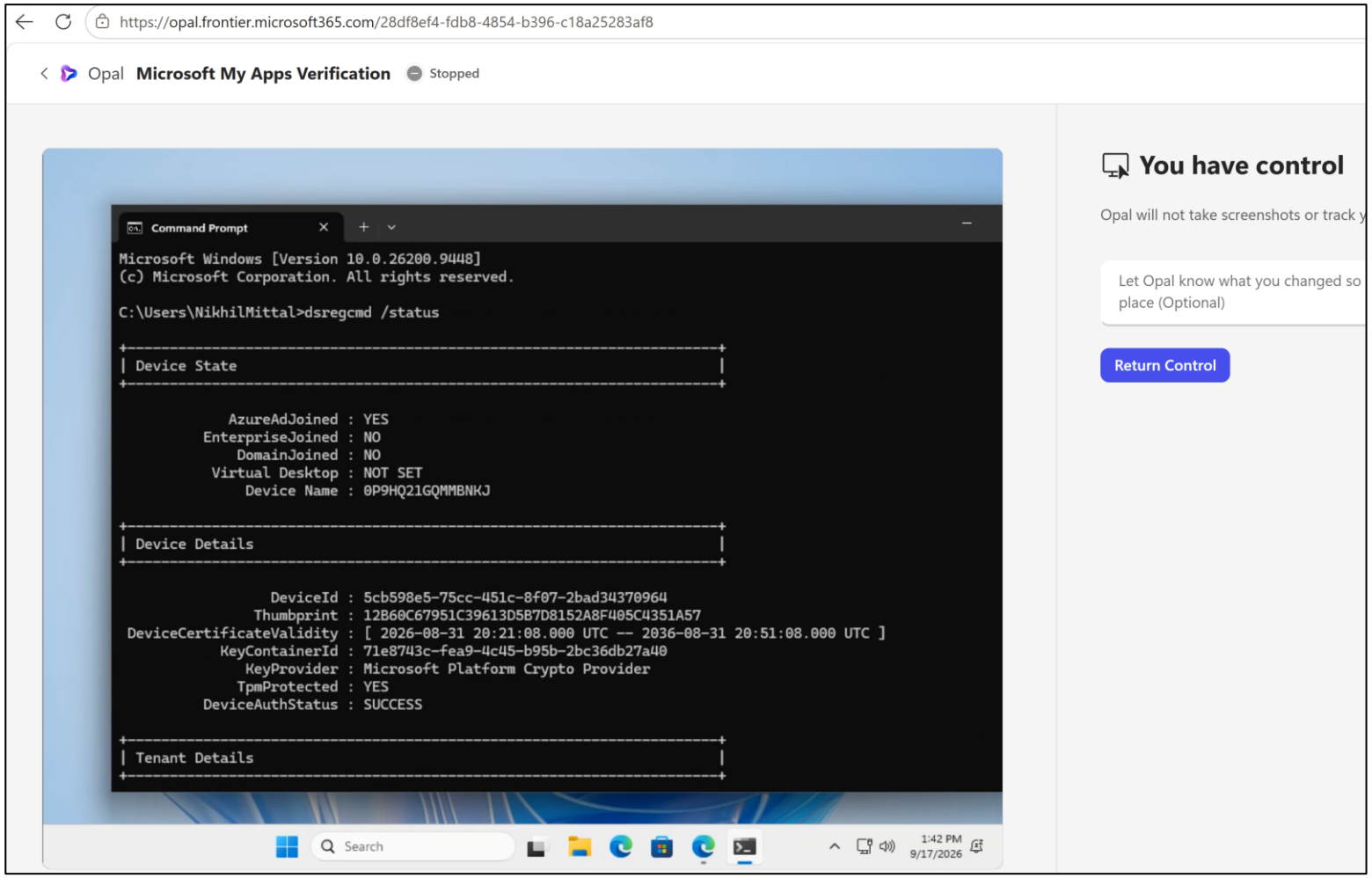
- Use compromised compute resources that may have their outbound IPs trusted.
- This may not only defeat CAE but also Location based Conditional Access, Entra ID Protection and NSGs.

- Examples: App Services, Functions, VMs, Azure Arc machines, On-Prem foothold etc.

Bonus Evasion - Project Opal

- Very interesting project - ~~Microsoft~~ Microslop Project Opal.
- As of September 2026, with credentials of a user who has access to Opal, you can access a fresh Windows 365 Cloud PC that is Entra Joined, Compliant and Company Owned.
- The web portal to access Opal has no mandatory MFA enforcement - https://opal.frontier.microsoft365.com

Bonus Evasion - Project Opal



Useful CAE "Features" for Attackers

- A client can request non-CAE token even if the Resource Server is capable!
- Managed Identities are not supported for CAE (and the access token for a MI has a lifetime of 24+ hours).

- **ARM**, Storage, Vault and more are not CAE capable.

ARM -> Access a resource -> Profit

ARM -> Access a resource -> Managed Identity -> Profit

- CAE doesn't support Guest accounts.

Research Methodology

- The focus of the research is on access tokens. Refresh tokens were assumed to be OPSEC unsafe.

- I wanted to find out if tokens are CAE-capable when we steal them and not when we request them with refresh tokens.

Prior Work

- In-depth research by researchers from ERNW on Token Theft, CAE and more - "Token Theft in Microsoft Entra ID - An Analysis of Controls" - https://ernw.de/en/whitepapers/issue-80.html

Defense

- To defend tokens, Microsoft suggests a Defense-in-depth strategy.
  - Minimize risk - Reducing the attack surface
  - Detect + Mitigate - Risk based CAP and CAE.
  - Protect against replay - Token Protection, IP Allowlisting etc.

  https://learn.microsoft.com/en-us/entra/identity/devices/protecting-tokens-microsoft-entra-id#defense-in-depth-strategy-against-token-theft

- For CAE evasion, there is no specific defense as we rely on implementation gaps.

Conclusion

- Server-side enforcement, unlike the current client-driven issuance.
- Microsoft needs to start using it more for own applications and token acquisitions.
- Trigger on change in both Azure and Entra ID Roles, Group Membership, ABAC etc.
- Start supporting Managed Identites. Report-only mode if revocation can't be enforced.

Thank you

Questions?