Nested APP Authentication

Undocumented Risk and Conditional Access Bypass

[#] Speaker: Jun Sheng Shi, Shang-De Jiang

$ whoami

- Jun Sheng Shi @Sigolon
- Security Researcher at CYCRAFT
- Focus On : Cloud Security, Windows Authentication, AI Security Technique … …
- Connect with me :
- www.linkedin.com/in/jun-sheng-shi

$ whoami

- Shang-De ‘John’ Jiang (@SecurityThunder)
- Deputy Director of Research at CYCRAFT
- UCCU Hacker Co-Founder
- HackerPeanutJohn owner
- Speaker at the following technical conferences
  - TROOPERS
  - BlackHat USA
  - CodeBlue
  - HITCON
  - …

Agenda

- Intro Conditional Access & Background
- Nested APP Authentication
- Undocumented Risk
- Conditional Access Bypass
- Discussion & Impact

Last Year in TROOPERS

Finding Entra ID CA Bypasses - the structured way

Download Slides

Watch Video

Entra ID Conditional Access is the guard dog of your tenant. But to set it up in a secure way is quite complicated, even if you know all the processing details, which are not always documented.

In this talk we go from a bypass found by accident to a structured approach of mapping Entra ID applications and the different behavior of Conditional Access policies. We found corner cases in which some policies are not or cannot be applied, which can function as bypasses if attackers want to target your tenant.

Join us for a wild ride into authentication protocols, OAuth scopes and pre-consented permissions.

And of course, we don't want to keep you in the dark about how to protect against some of those bypasses and what indicators to look out for.

Conditional Access policies are complicated in configuration, but also in evaluation on the Microsoft side. Many apps interact with other apps and may have dependencies on backend APIs that could be covered by different policies than the "front-end" app. This leads to several observable behaviour differences in the enforcement of for example compliant devices and MFA for specific resources.

We wanted to analyze all the different combinations of client apps and resources (APIs) that exist in Entra ID and Microsoft 365, to see how they behave and also what permissions each application has. That sounds simple, you just sign in to every app and then you see what kind of permissions they have. However, reality is not that simple. First you need to find all the possible applications that exist, then figure out which authentication methods you can use to authenticate to them, and finally you have to make this scalable for hundreds of applications.

In this talk we walk through our process of analyzing applications, automating token requests and defining different test scenarios for testing undocumented Conditional Access policies exceptions in a structured way, at scale.

We show several examples of behaviour differences when policies are configured in a certain way, and some hardcoded exceptions that are not documented by Microsoft. We will also show what this means for attackers and defenders, how to protect against these corner cases and how to monitor for abuse.

Dirk-jan Mollema

Dirk-jan Mollema is a hacker and researcher of Active Directory and Microsoft Entra (Azure AD) security. In 2022 he started his own company, Outsider Security, where he performs penetration tests and reviews of enterprise networks and cloud environments. He blogs at dirkjanm.io, where he publishes his research, and shares updates on the many open source security tools he has written over the years. He presented previously at TROOPERS, DEF CON, Black Hat and BlueHat and has been awarded as one of Microsoft's Most Valuable Researchers multiple times.

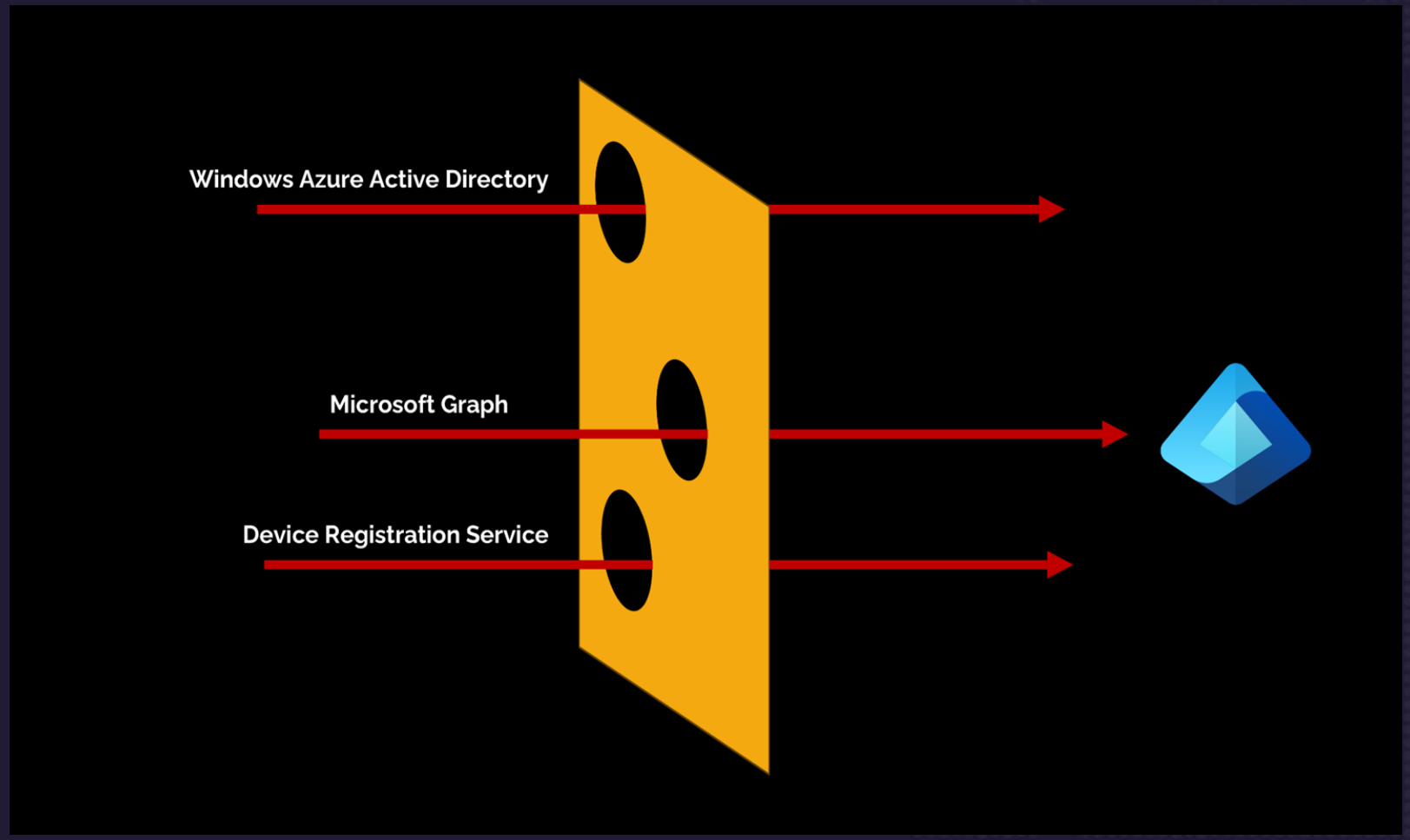
Fabian Bader

Fabian Bader is a Cyber Security Architect and Microsoft MVP from Germany. He focuses on security and cloud solutions and works mainly with Microsoft technologies. From Azure cloud to on-premises Active Directory, he likes to automate stuff with PowerShell.

Besides being a speaker at community events, he blogs at "cloudbrothers.info", hosts the "Elbe Security User Group", "Hamburg PowerShell User Group" and is part of the organizing team of "PowerShell Saturday Hamburg".

❤️ PowerShell and Security 🛡️

Azure Authentication mechanism may lead to Conditional Access Bypass



Ref: Conditional Access bypasses - Cloudbrothers

Office 365 Management

App ID: 00b41c95-dab0-4487-9791-b9d2c32c80f2

Family of Client IDs | Public client app | CA Bypasses (1)

Known Conditional Access Bypasses

Active | All

Conditional Access policies are not applied to the listed resources. This is consider
data and serves as a preliminary step before reaching the final resource.[...]' - MSF

Read more: http://cloudbrothers.info/en/conditional-access-bypasses/ →

Discovered by: @fabian_bader

Affected Resources & Scopes:

Microsoft Intune Checkin
26a4ae64-5862-427f-a9b0-044e62572a4f

- user_impersonation

Office 365 Management

App ID: 00b41c95-dab0-4487-9791-b9d2c32c80f2

← Back to Grid

Family of Client IDs | Public client app | CA Bypasses (1)

Known Conditional Access Bypasses

But these are only for non-sensitivity can bypass CA policy.

Affected Resources & Scopes:

Microsoft Intune Checkin
26a4ae64-5862-427f-a9b0-044e62572a4f

- user_impersonation

Ref: Conditional Access bypasses - Cloudbrothers

Token Exchange 101

Credentials → Refresh Token → Resource Token. One trade at a time.

A Basic approach — sign in to each client, one by one

B PRT-based approach — one PRT, repeat for all clients

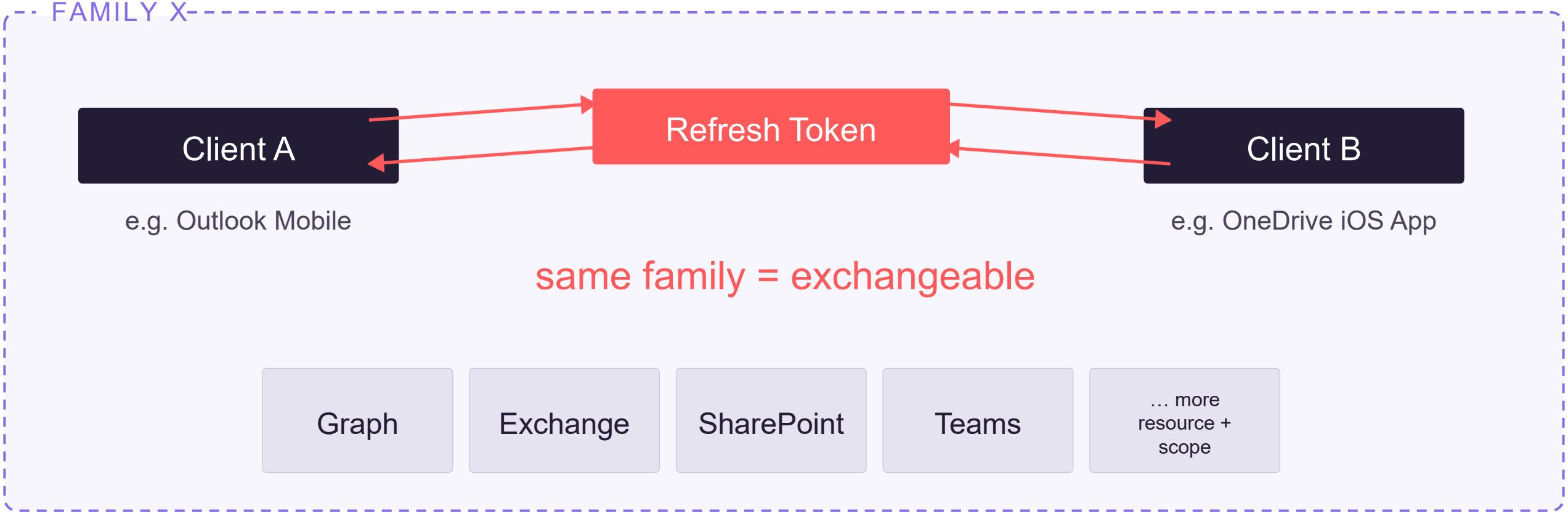
using PRTv3 broker flow

[Ref Source: Finding Entra ID CA Bypasses - the structured way]

CONCEPT 1 OF 3

FOCI — Family Client Token Exchange

Same-family clients share refresh tokens — swap one and get any other resource & scope.



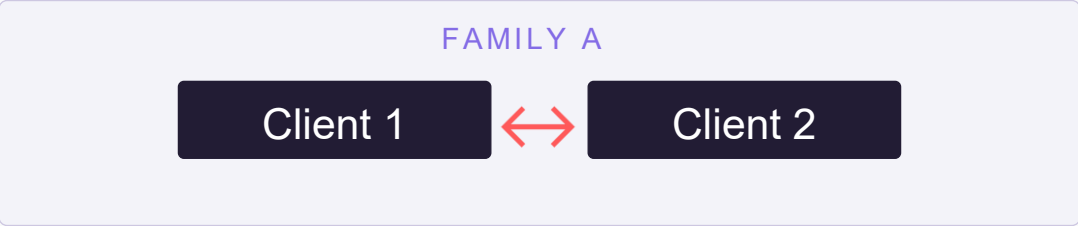
CONCEPT 2 OF 3

BroCI / NAA — Broker Client Token Exchange

Same recipe, new bottle. Clients under one broker can swap tokens too — only target & resource differ.

FOCI

Family of Client IDs: clients tagged as one family by Microsoft on the backend.

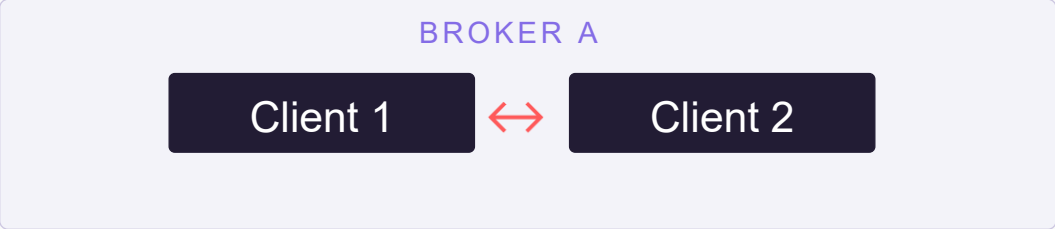


CONTAINER

Family

BroCI / NAA

Broker Client:nested clients hosted by a broker app.



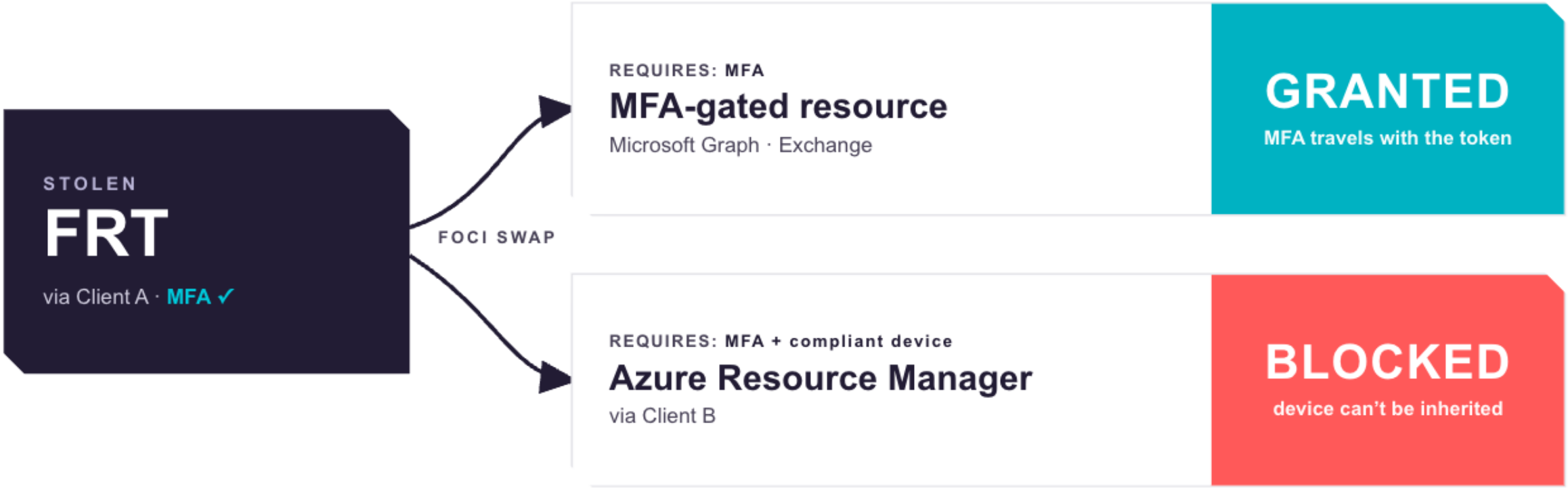
CONTAINER

Broker

CONCEPT 3 OF 3

Conditional Access re-evaluates every redemption

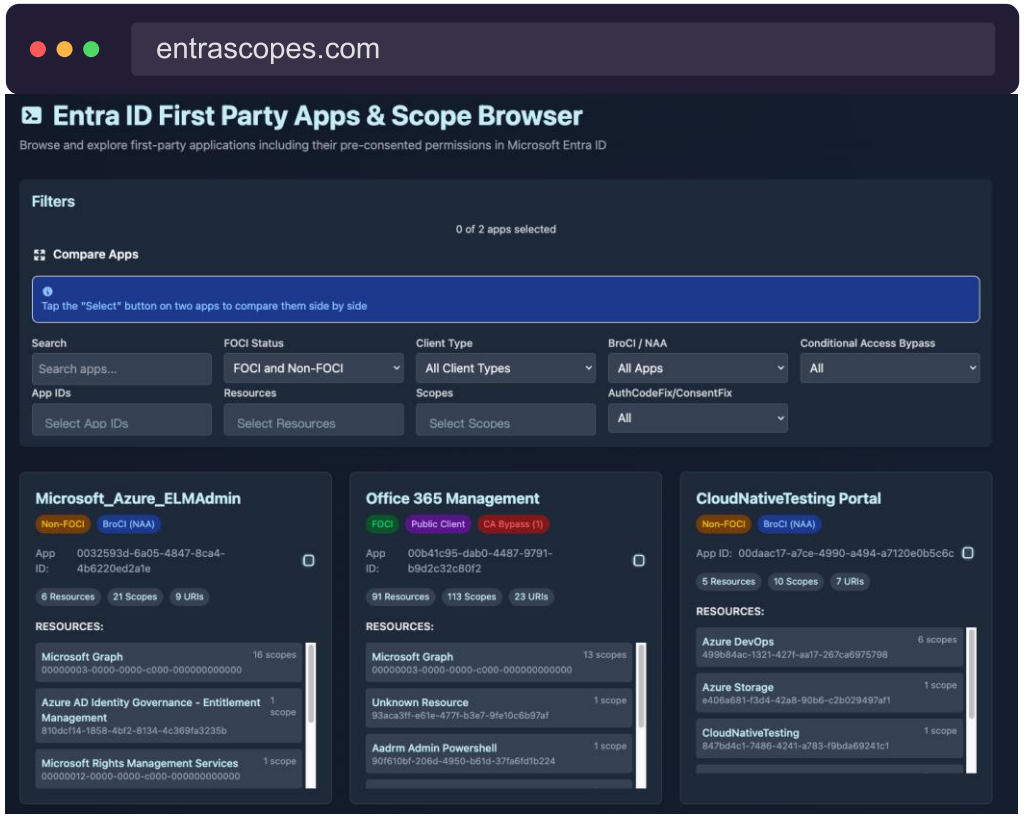
Even they under same family/broker, Conditional Access still re-evaluates.



What we already know.

- Inside the same FOCI / NAA, identical conditions make tokens exchangeable.
- Some scenarios ship with bypass-able resource + scope combos out of the box.

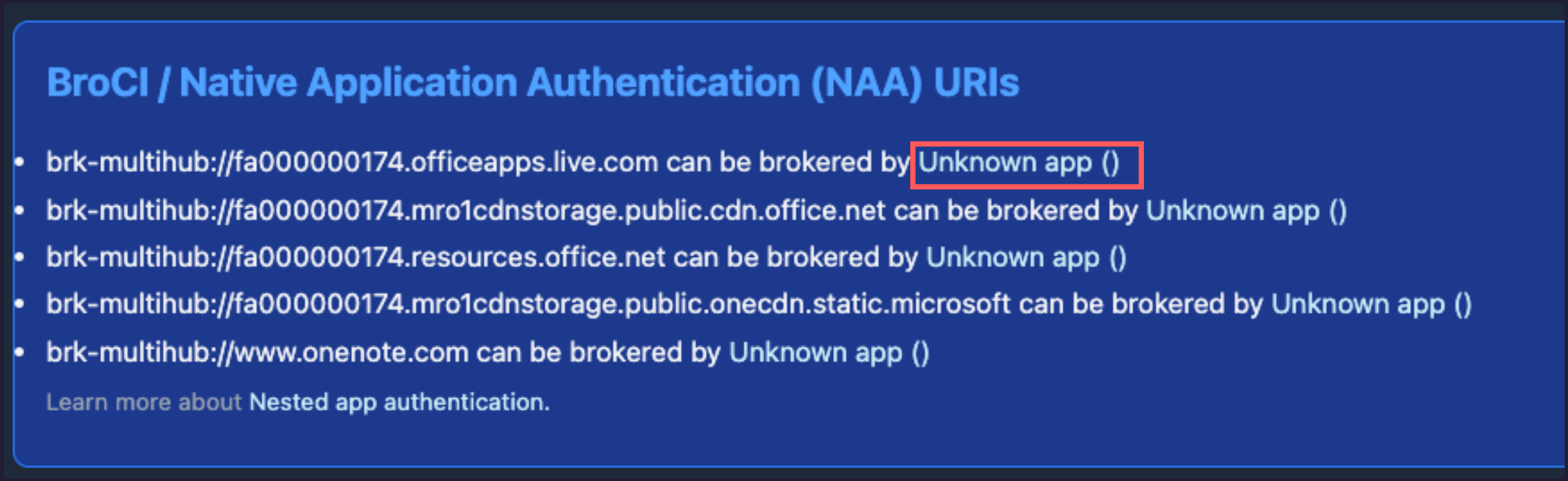
> entrascopes.com



Known to be BroCI/NAA - but who's the broker?

EntraScopes.com list a batch of known BroCI URIs — some of them labelled as Unknown app.

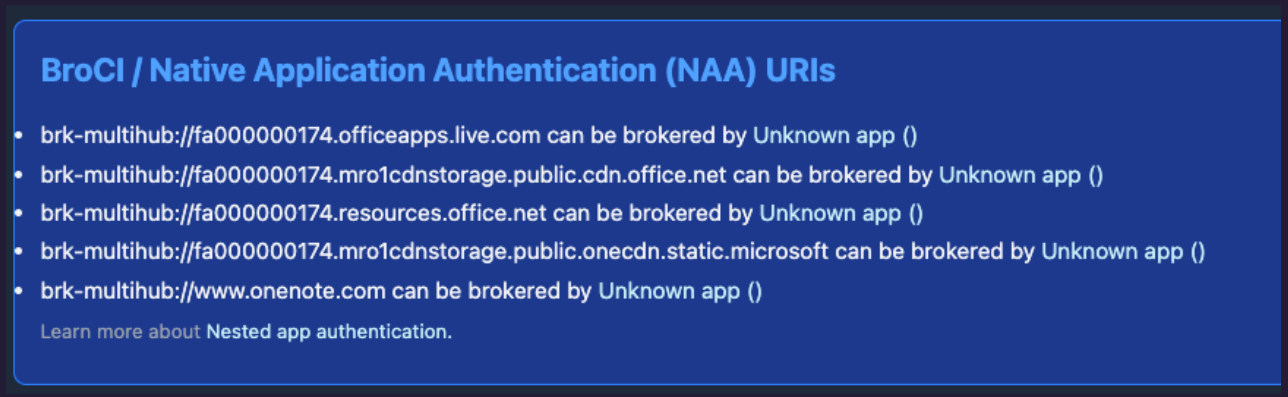
OneNote · ENTRASCOPES.COM



Known to be BroCI/NAA - but who's the broker?

EntraScopes.com list a batch of known BroCI URIs — some of them labelled as Unknown app.

OneNote · ENTRASCOPES.COM



BroCI / Native Application Authentication (NAA) URIs

- brk-multihub://fa000000174.officeapps.live.com can be brokered by Unknown app ()
- brk-multihub://fa000000174.mro1cdnstorage.public.cdn.office.net can be brokered by Unknown app ()
- brk-multihub://fa000000174.resources.office.net can be brokered by Unknown app ()
- brk-multihub://fa000000174.mro1cdnstorage.public.onecdn.static.microsoft can be brokered by Unknown app ()
- brk-multihub://www.onenote.com can be brokered by Unknown app ()

Learn more about Nested app authentication.

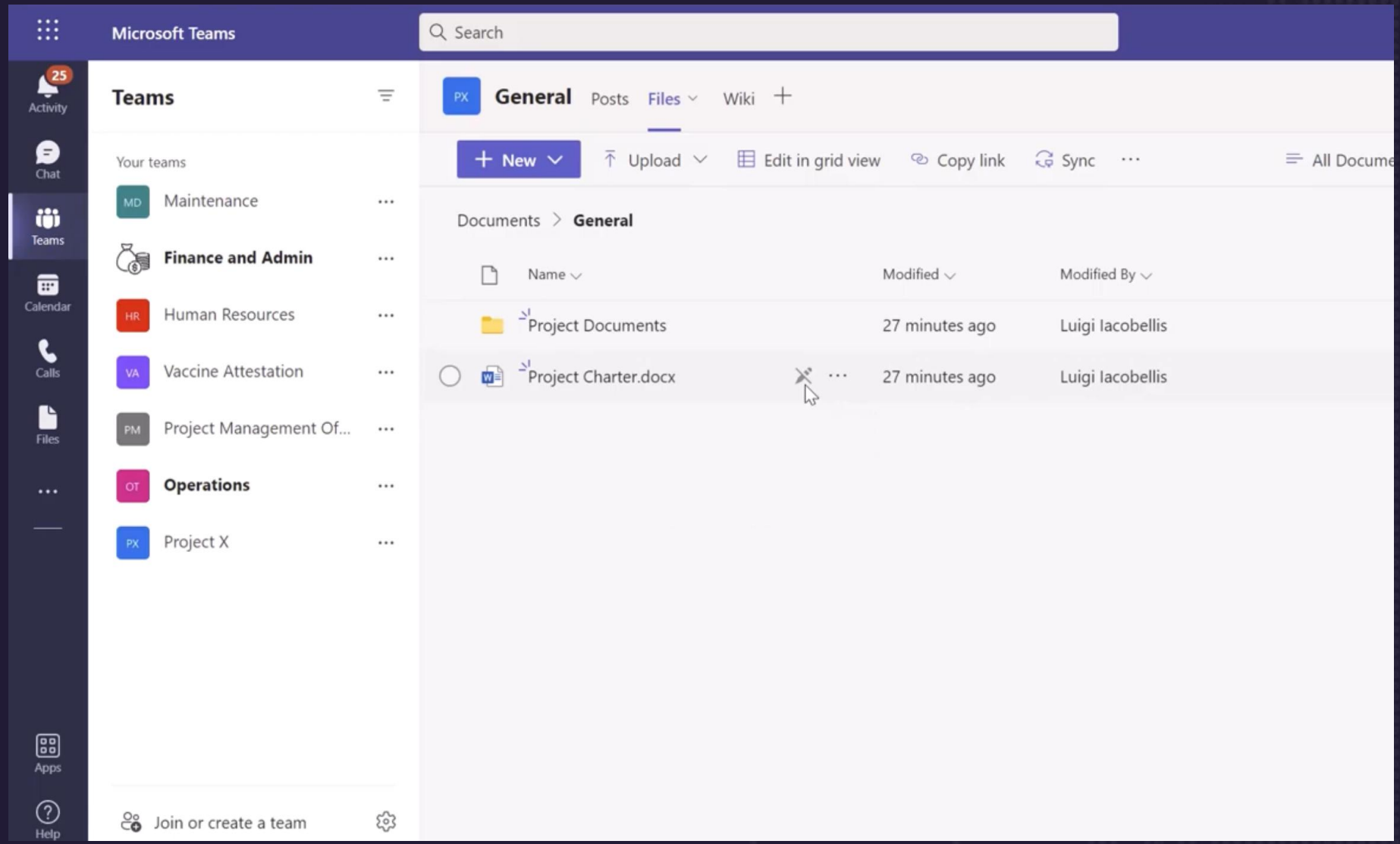
OUR QUESTIONS

Q1 How bad does it actually look?

Q2 Could it be worse than FOCI ?

Nested App Authentication (NAA)

What's a Nested App?

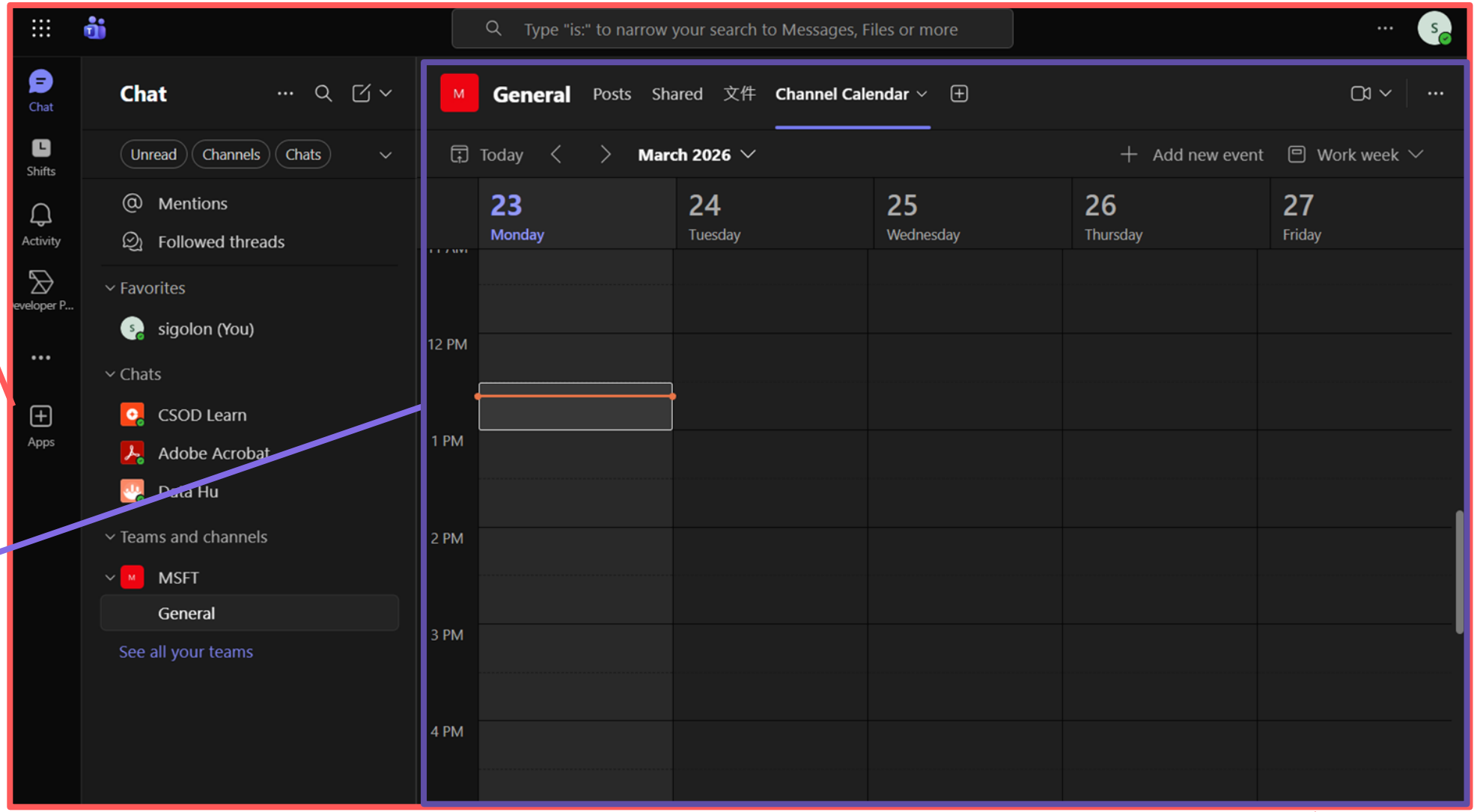


CYCRAFT

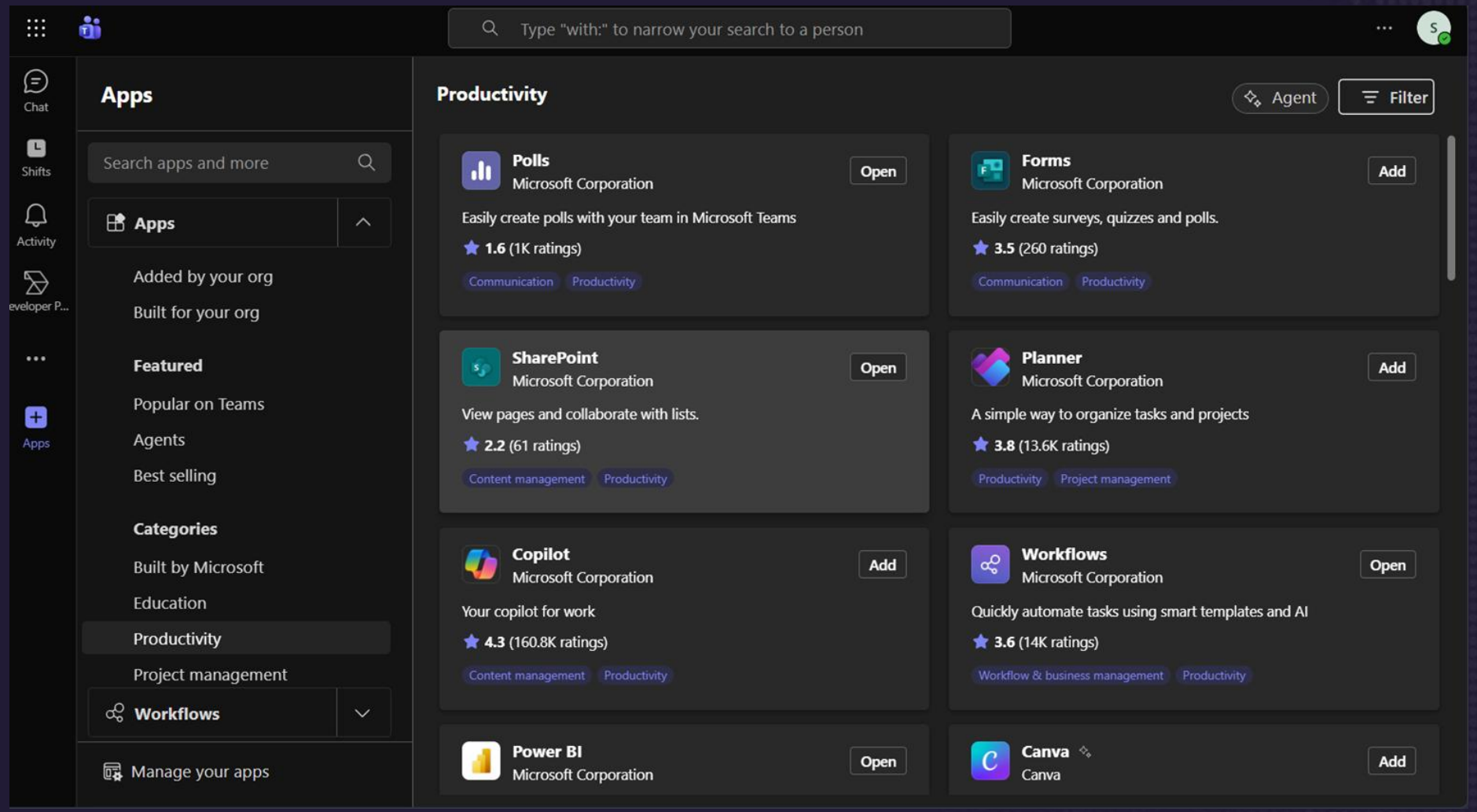
Nested App Example

Broker Client: Teams

Nested App: Channel Calendar

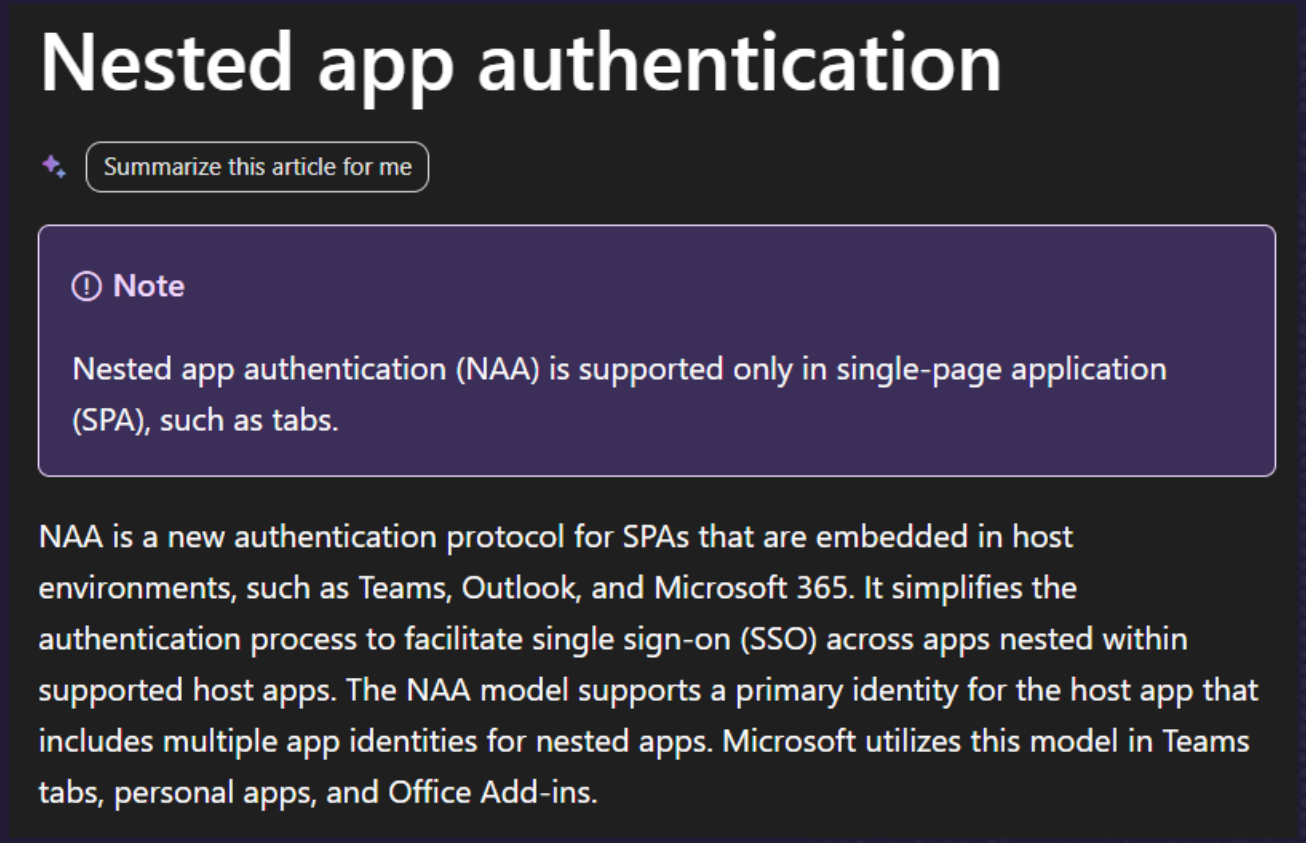


Teams Marketplace Has Many Nested Apps



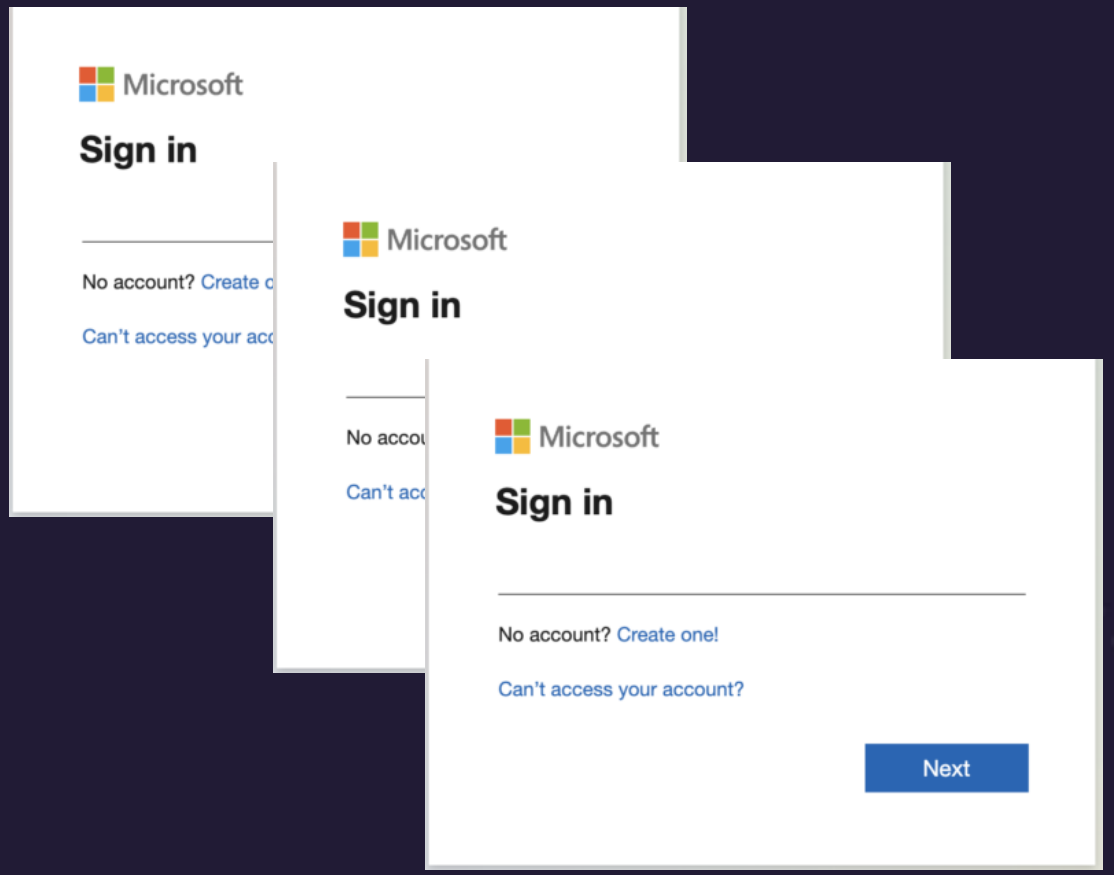
CYCRAFT

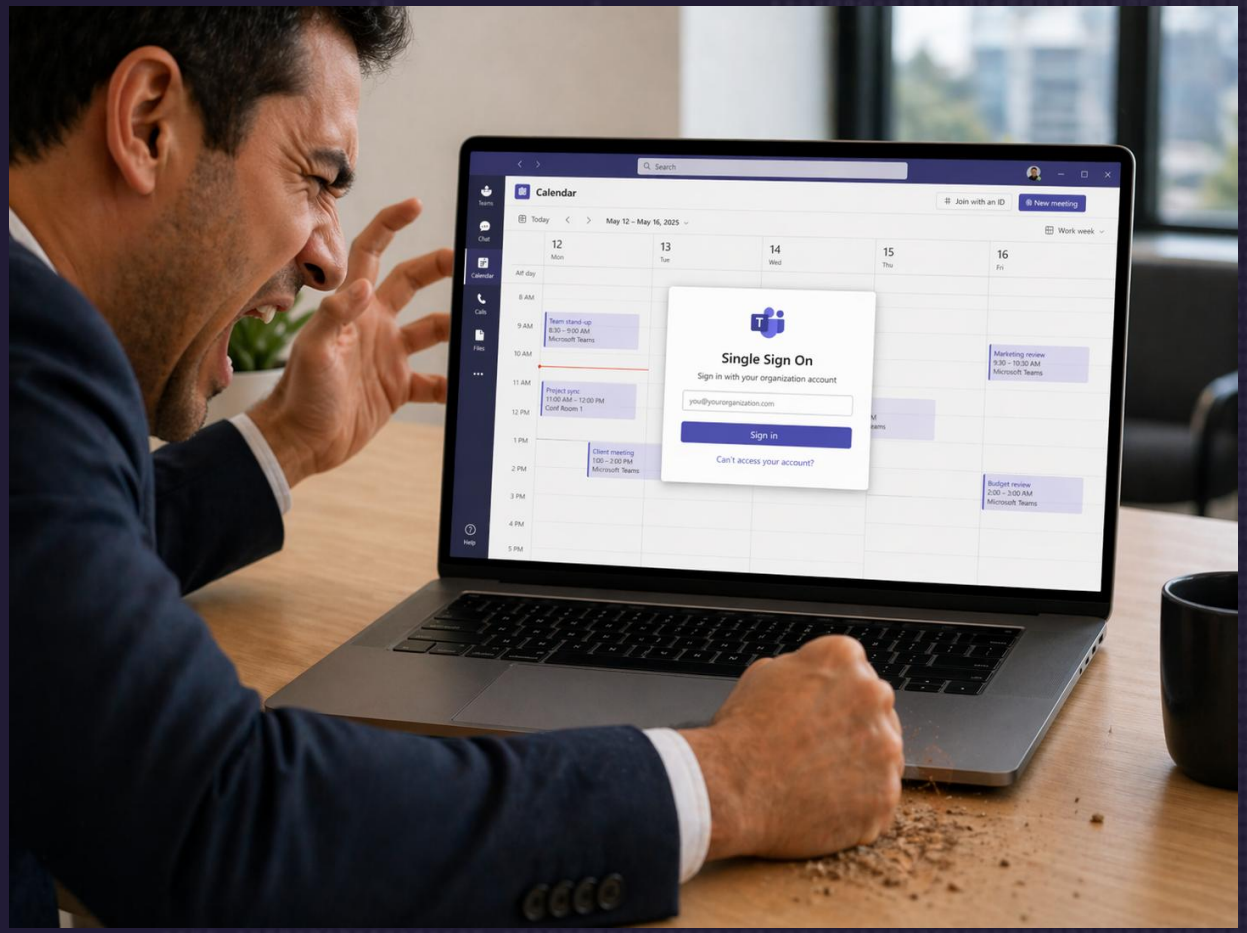
What Is Nested App Authentication?



SSO authentication for nested apps - Teams | Microsoft Learn

However, if a user needs to authenticate every nested app again…

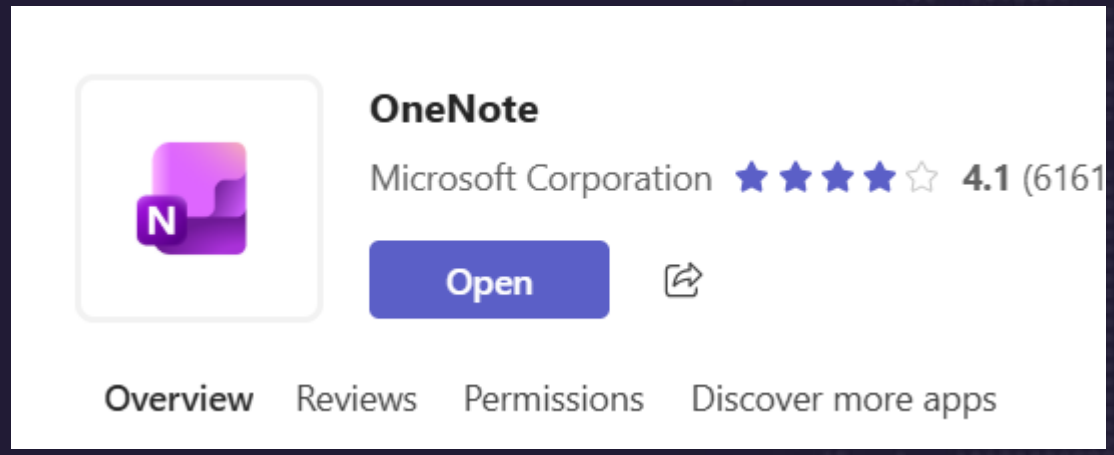




CYCRAFT

Microsoft improve the user experience for NAA

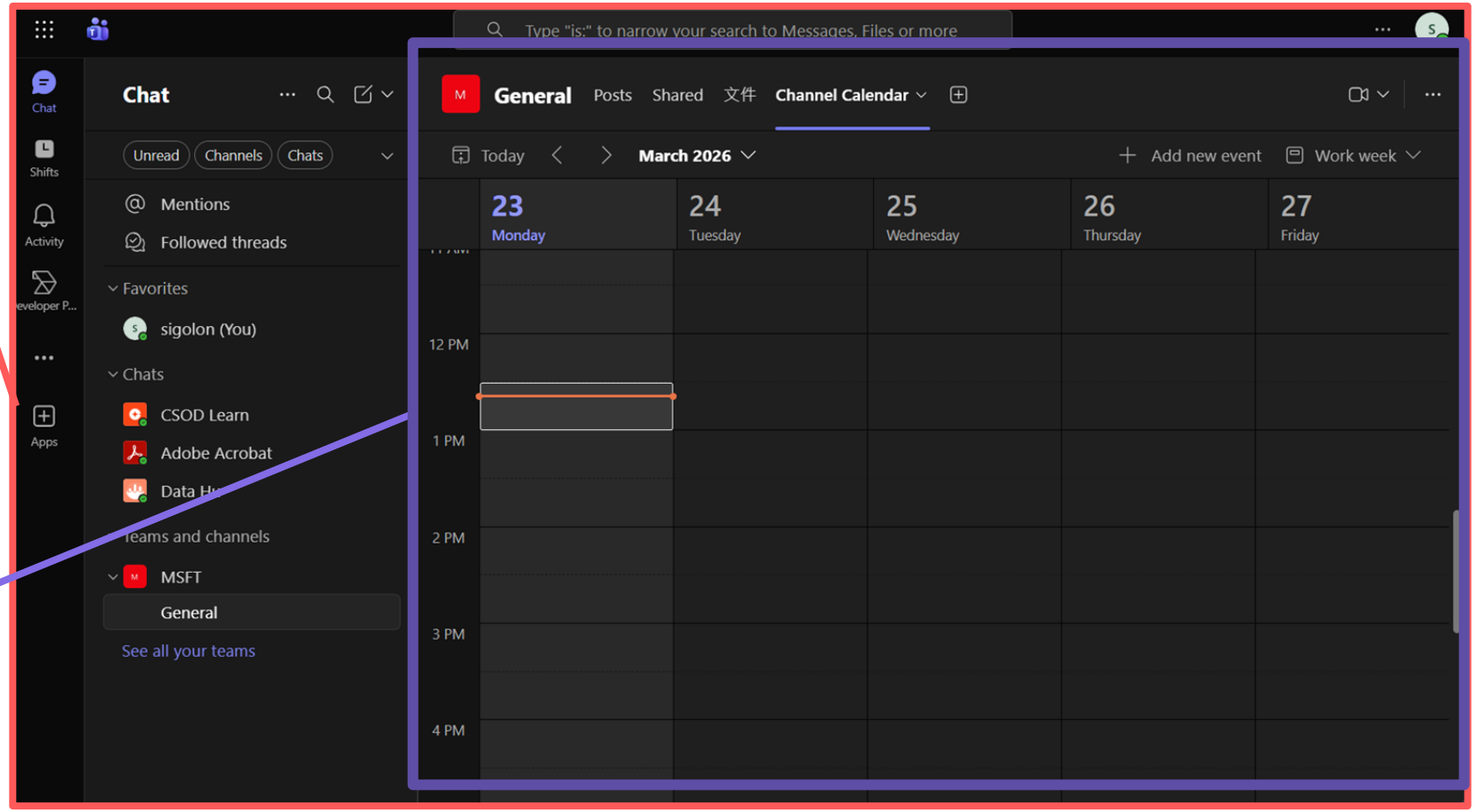
- Microsoft created a pre-auth setting for first-party nested apps, such as OneNote.



Nested App – Teams Example

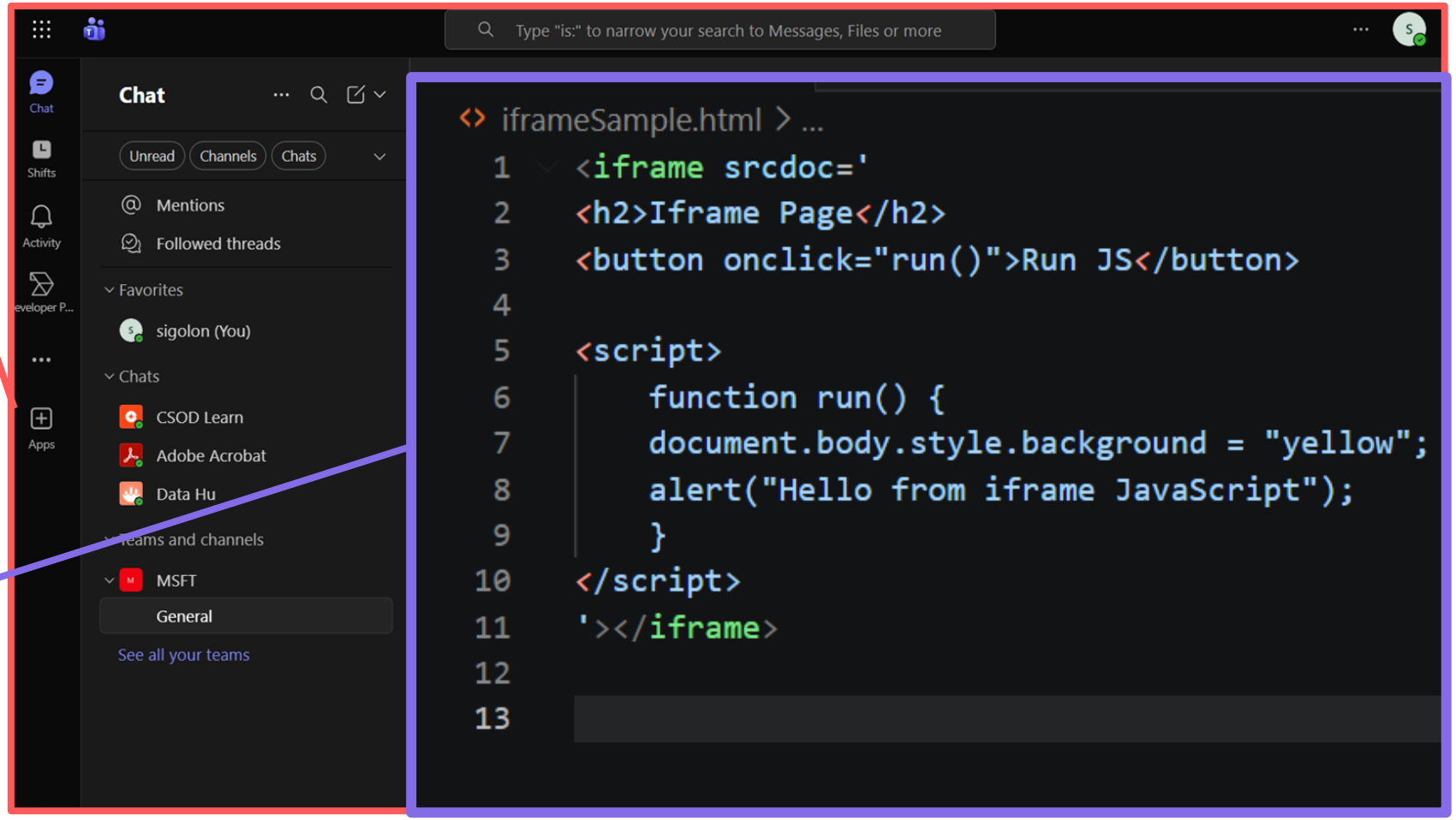
Broker Client: Teams

Nested App



Broker Client: Teams

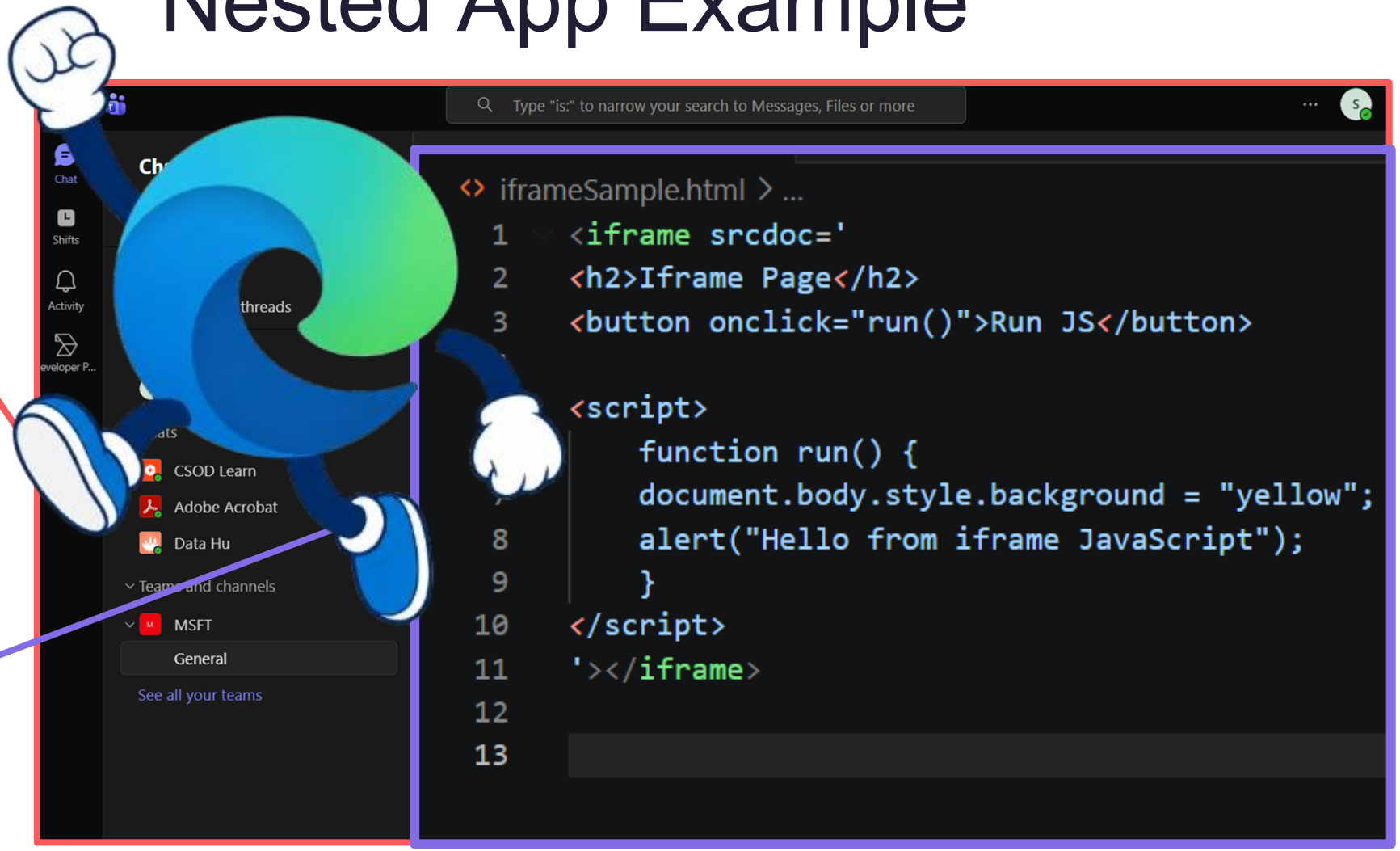
iframe



Nested App Example

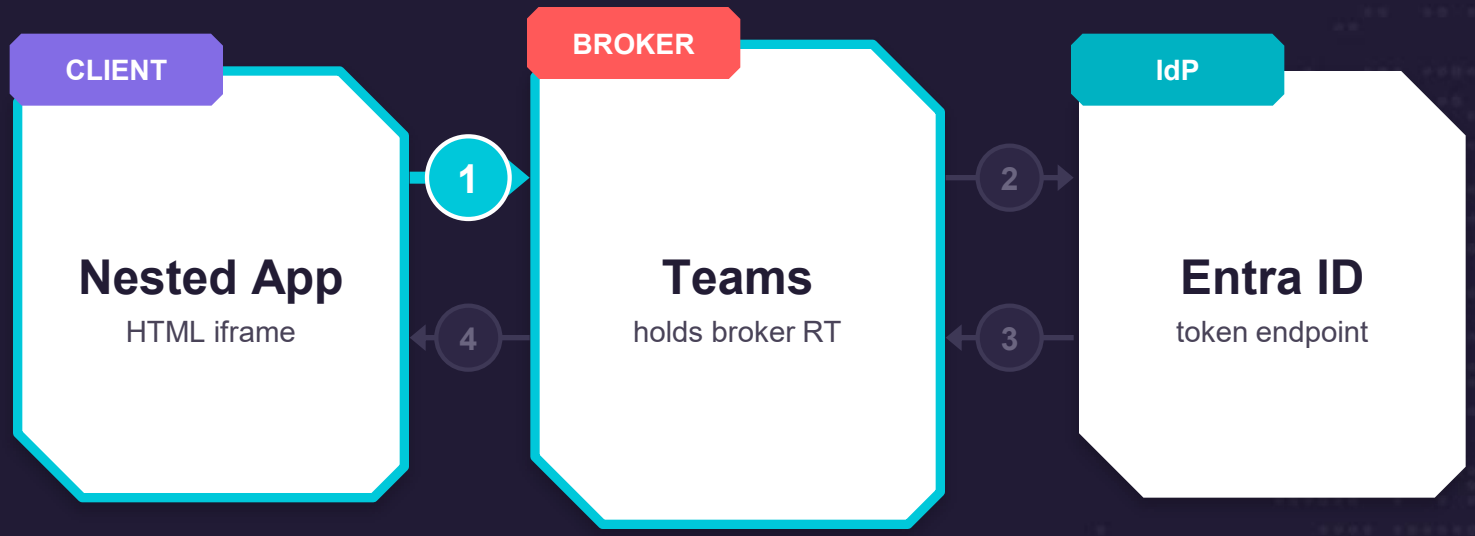
WebView2

Nested App



CYCRAFT

NAA Auth Workflow

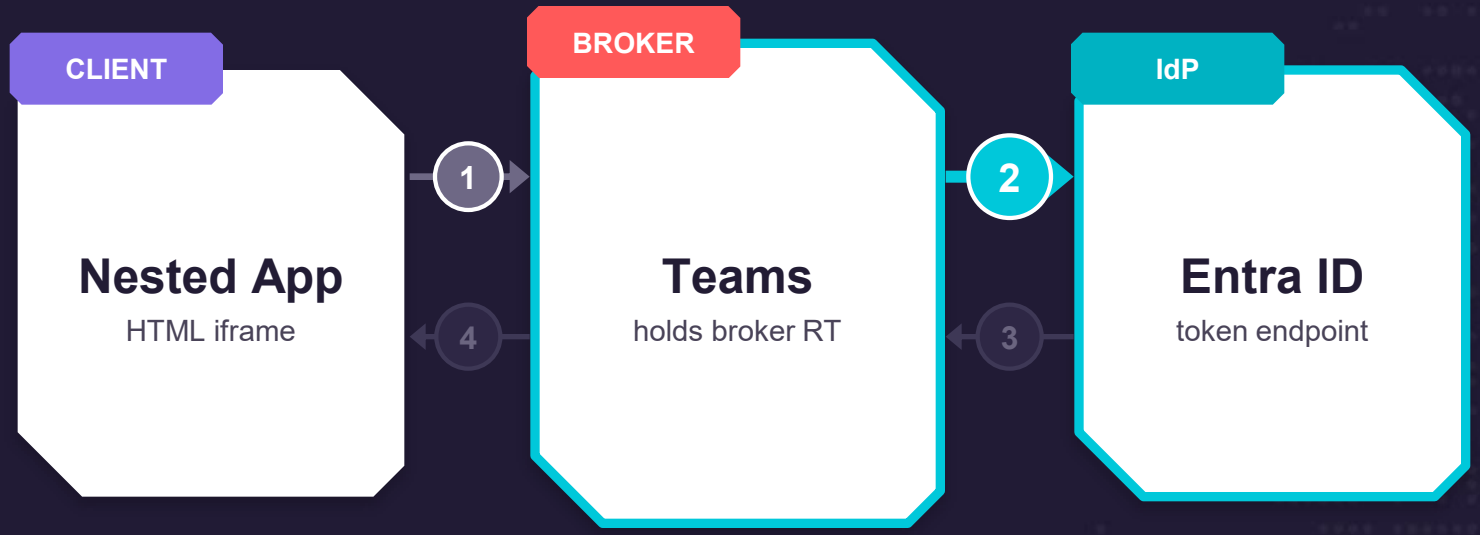


WORKFLOW · 4 STEPS

1. **Send token-exchange task to broker**
2. Use broker refresh token for access token
3. Entra ID returns the access token
4. Forward access token to nested app

CYCRAFT

NAA Auth Workflow



WORKFLOW · 4 STEPS

✓ Send token-exchange task to broker

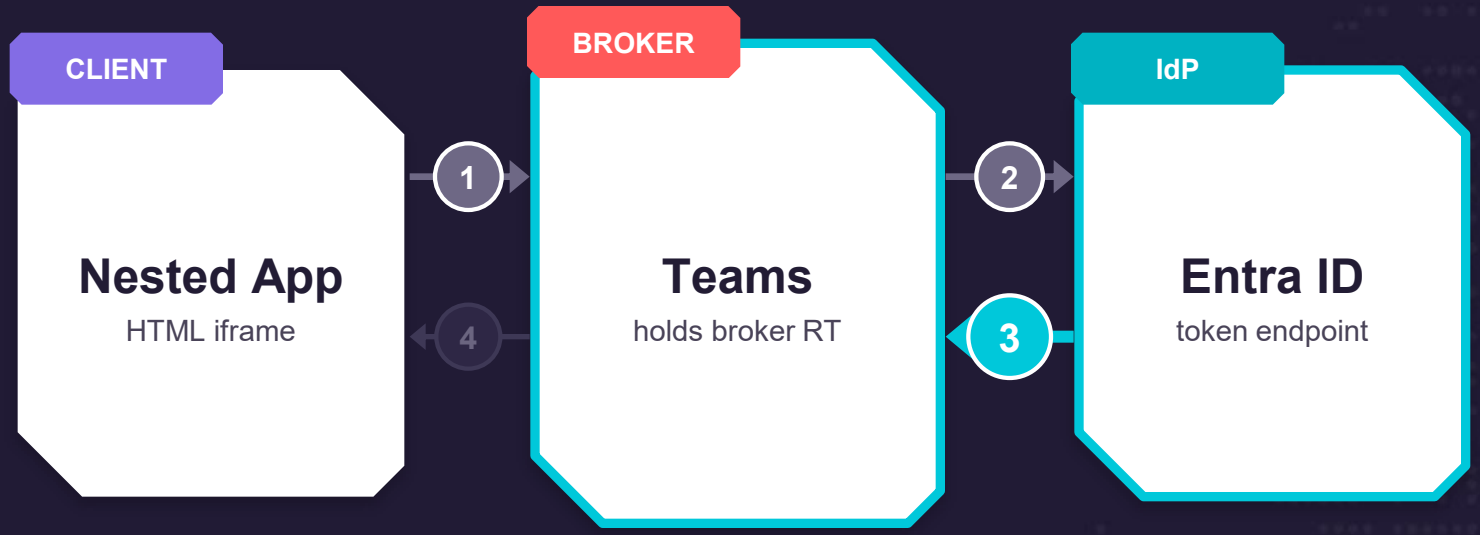
2 Use broker refresh token for access token

3 Entra ID returns the access token

4 Forward access token to nested app

CYCRAFT

NAA Auth Workflow

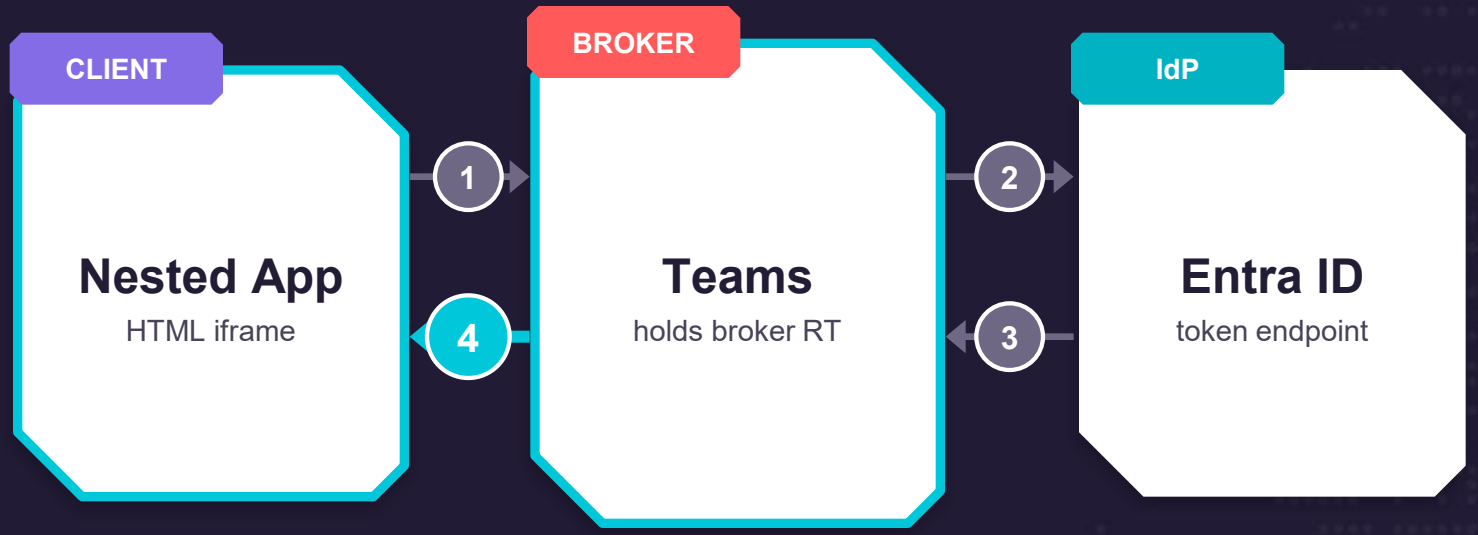


WORKFLOW · 4 STEPS

- ✓ Send token-exchange task to broker
- ✓ Use broker refresh token for access token
- **3 Entra ID returns the access token**
- 4 Forward access token to nested app

CYCRAFT

NAA Auth Workflow



WORKFLOW · 4 STEPS

- ✓ Send token-exchange task to broker
- ✓ Use broker refresh token for access token
- ✓ Entra ID returns the access token
- 4 **Forward access token to nested app**

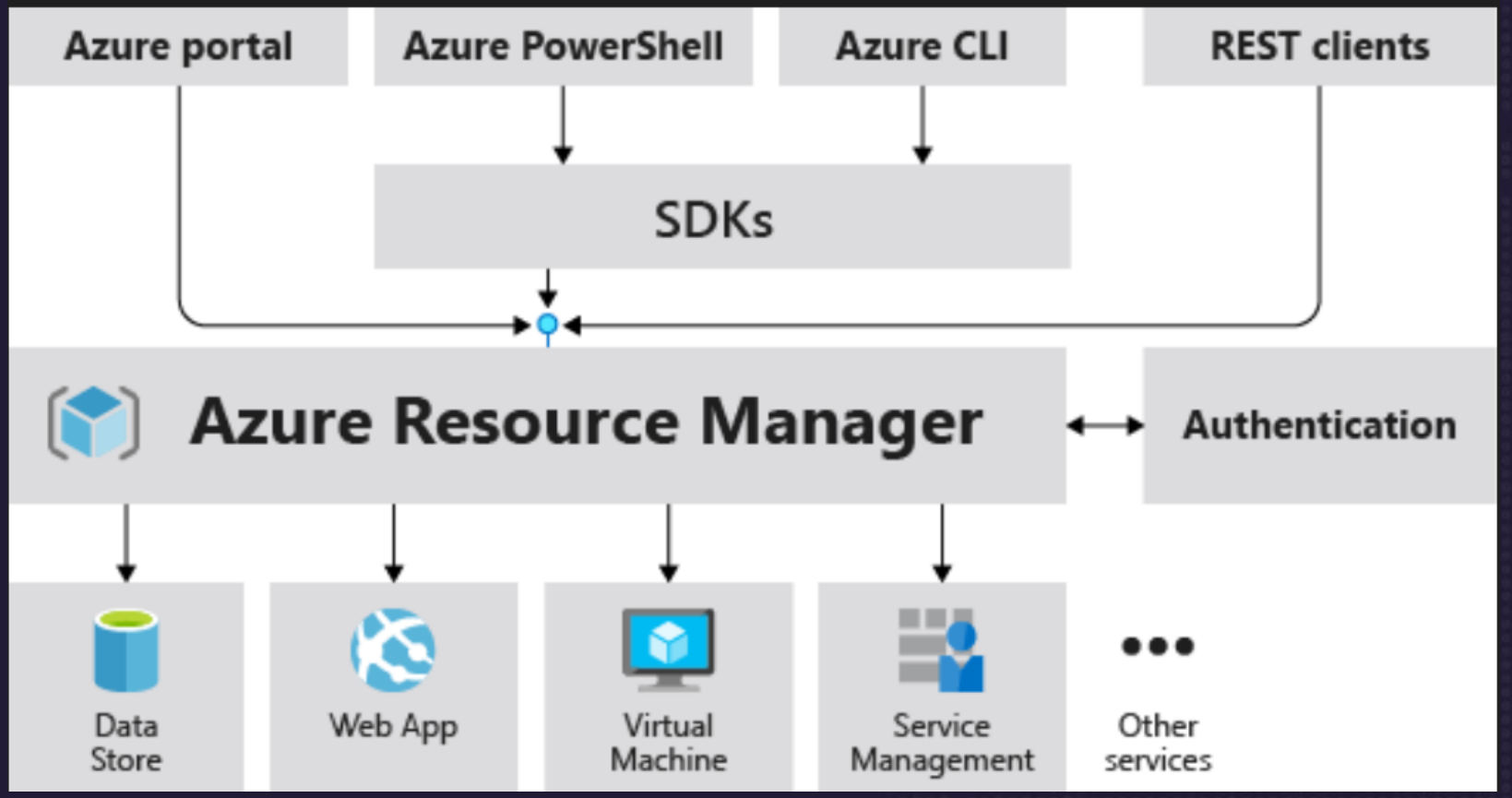
CYCRAFT

But Nested APP Just Enable Access Microsoft 365, Right?

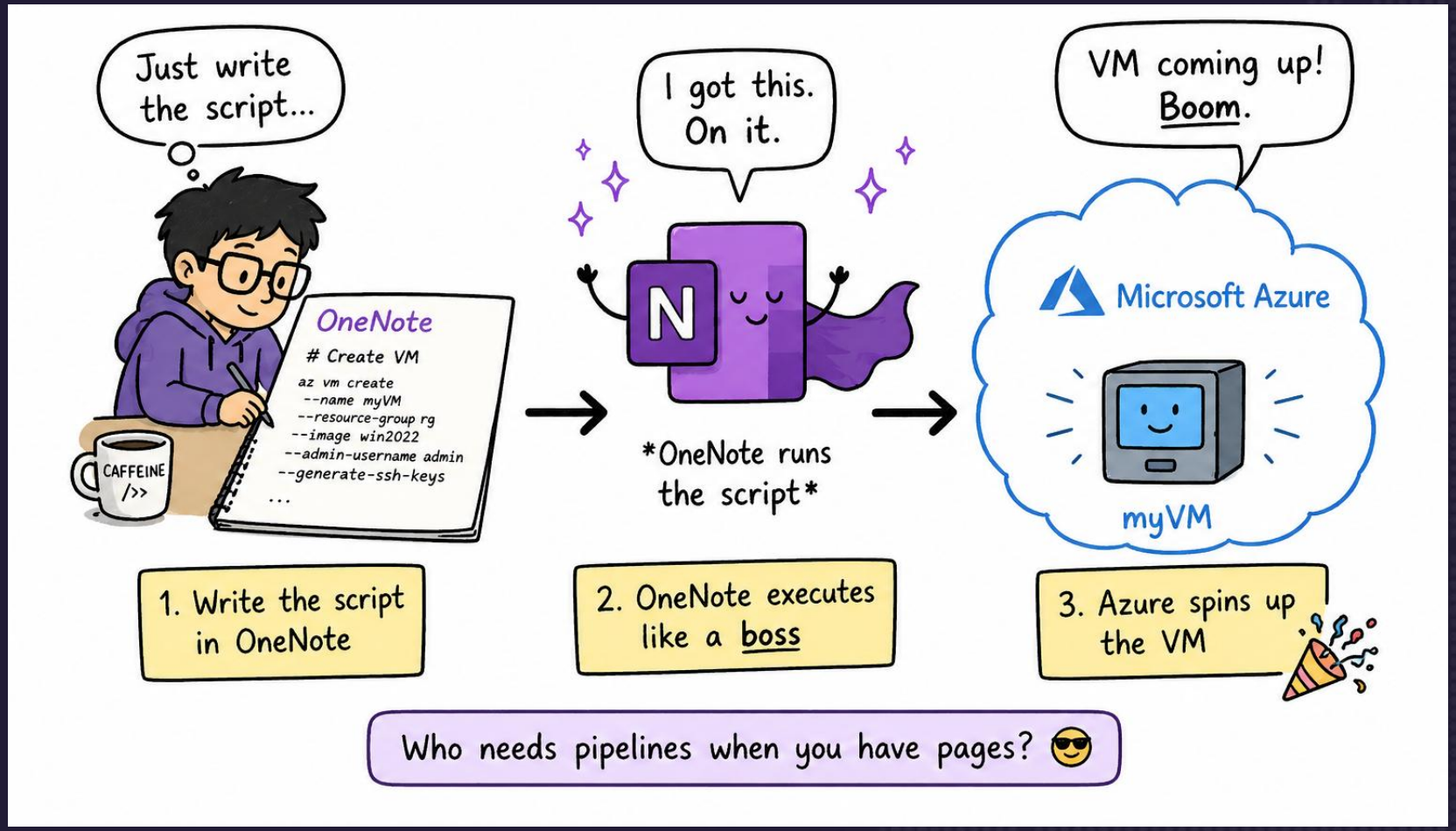
We notice OneNote enable Access ARM

CYCRAFT

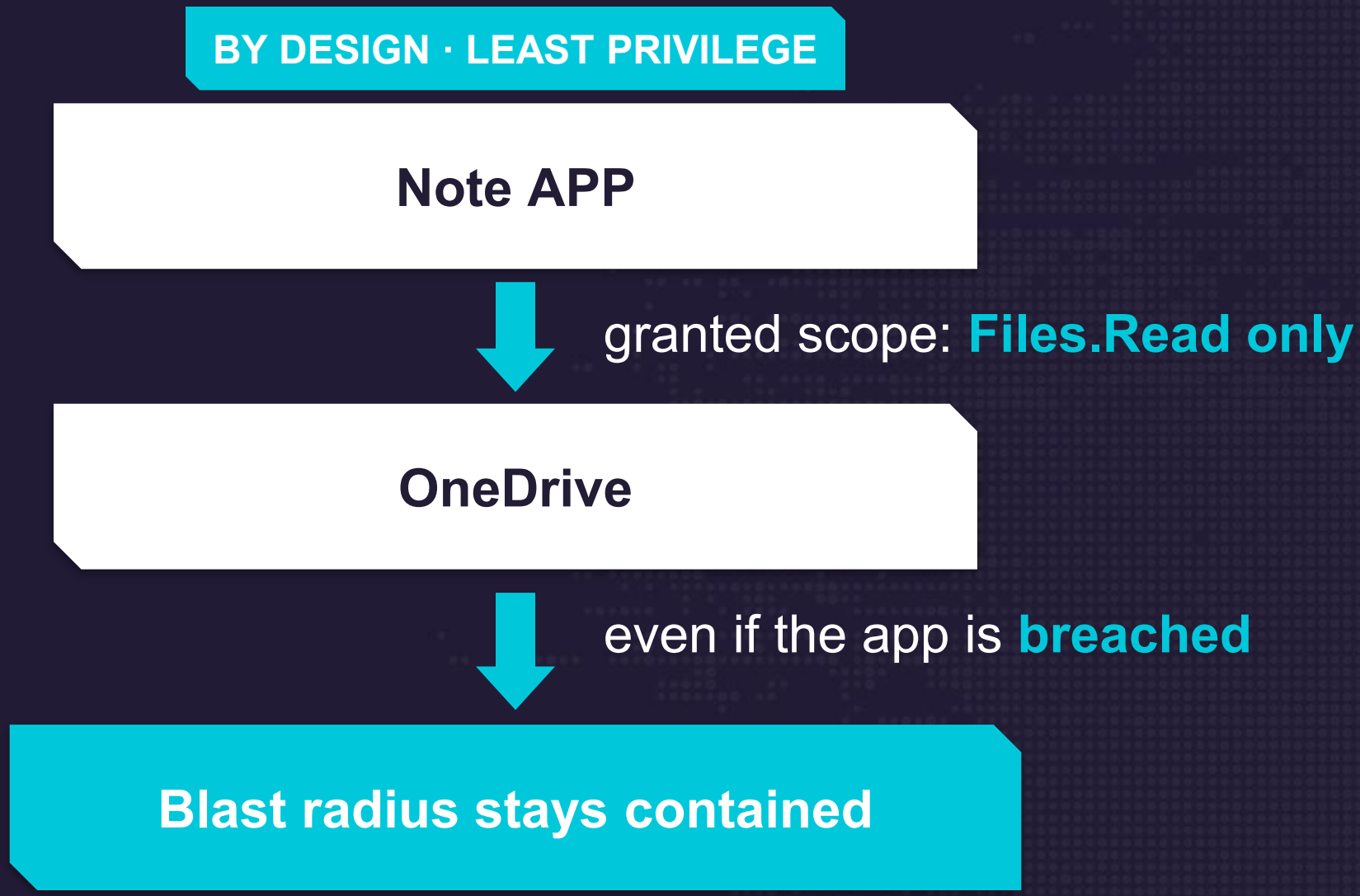
What's Azure Resource Manager?



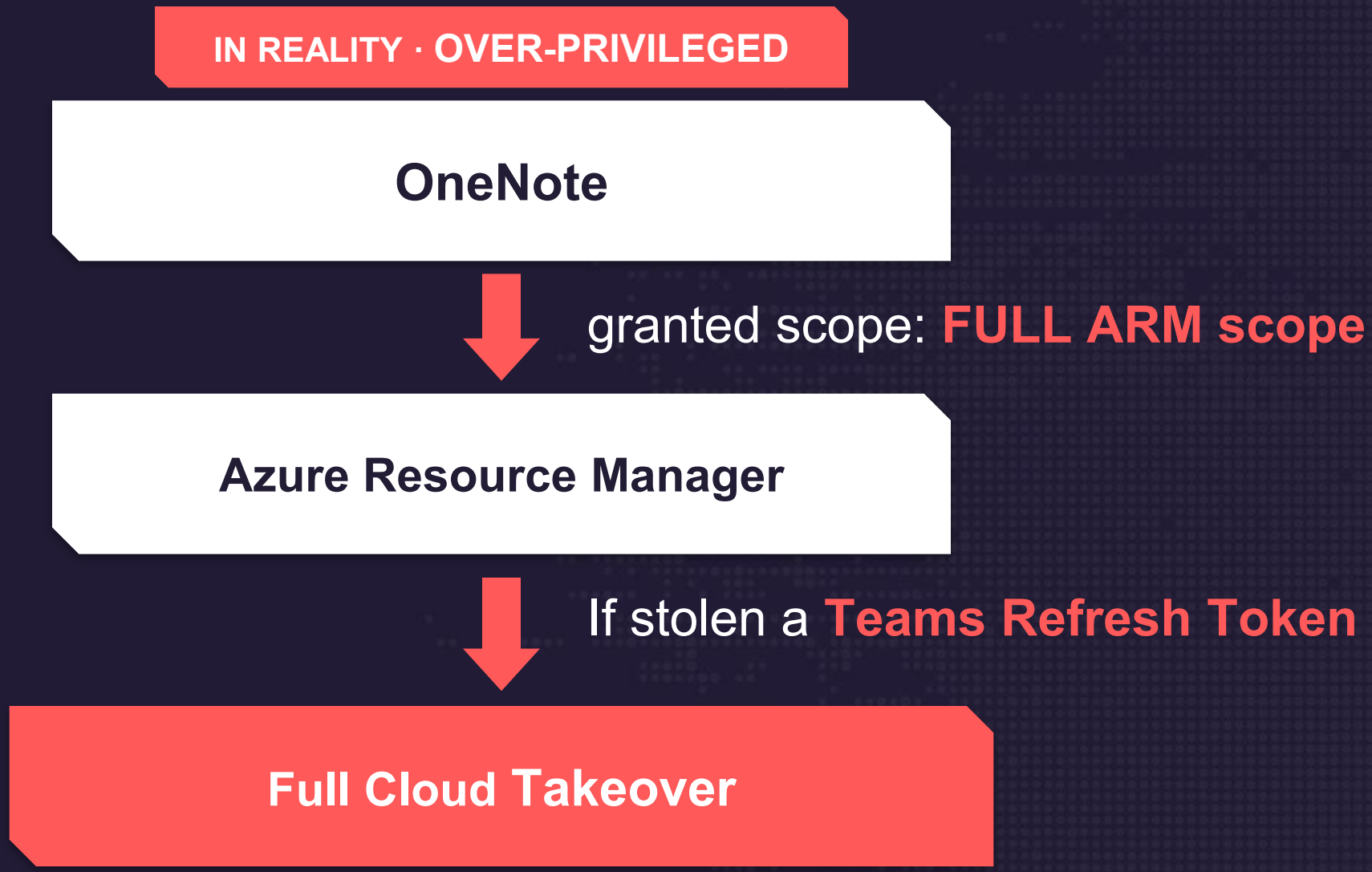
DevOps… the OneNote way



OAuth Scope Defines the Blast Radius

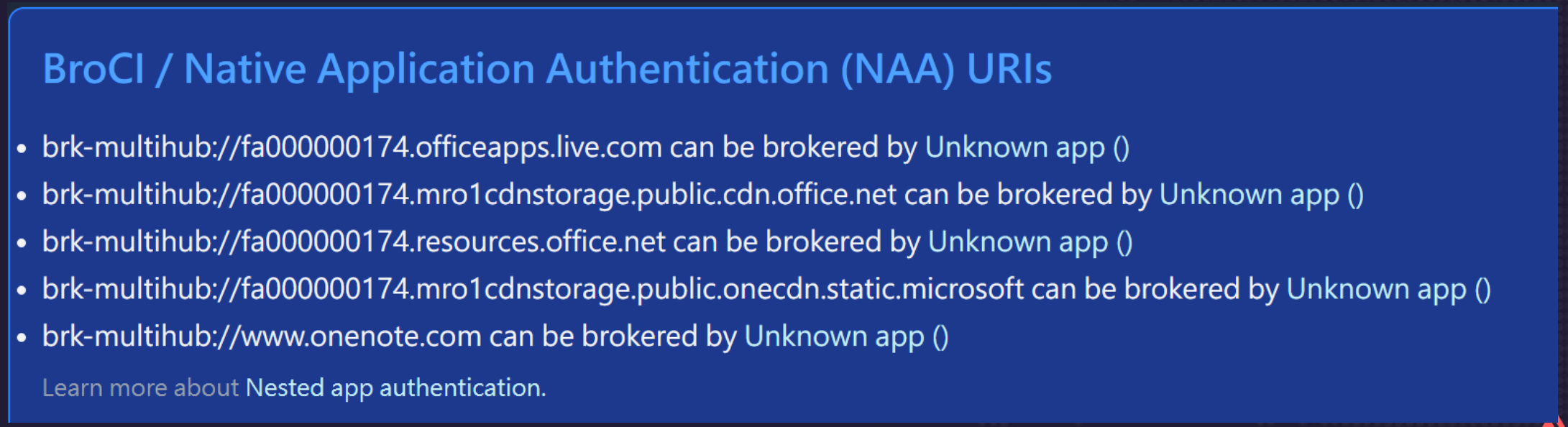


OAuth Scope Defines the Blast Radius



How to use brk-multihub?

- To test the impact scope of this Pre-Auth, we first need to know which Broker Clients can use OneNote
- However, it is not yet known which Broker Clients can access the brk-multihub://



Teams Can Use brk-multihub

NAA is a new authentication protocol for SPAs that are embedded in host environments, such as Teams, Outlook, and Microsoft 365. It simplifies the

Microsoft Teams

App ID: 1fec8e78-bce4-4aaf-ab1b-5451cc387264

```
(teams2Azure) PS C:\Users\USER\Desktop\cycraft\MSBrokerAppScan> .\script\nested_token_exchange.ps1 -broker_client "1fec8e78-bce4-4aaf-ab1b-5451cc387264" -nested_client "2d4d3d8e-2be3-4bef-9f87-7875a61c29de" -broker_uri "brk-multihub://www.onenote.com" -refreshtoken_file ".\teams_refreshtoken" -resource "https://management.azure.com/user_impersonation"
roadtx : start
Requesting token with scope https://management.azure.com/user_impersonation
Tokens were written to test
roadtx : end
access token information :
aud : https://management.azure.com
amr : pwd rsa
scp : user_impersonation
```

Nested App Pre-Auth Scope Risk

38
nested apps available

38
distinct resources reachable

High-sensitivity Azure targets

- **Azure Resource Manager** — user_impersonation · **5 apps**
- **Azure Storage** — user_impersonation · **2 apps**
- **Azure DevOps** — vso.profile vso.work_write · **1 app**

Pre-authorized nested apps reach Azure with **no extra consent**

Nested App Pre-Auth Scope Risk

KEY INSIGHT

Attackers can use these nested apps to obtain **ARM access tokens** — all via user_impersonation.

Teams Pre-auth nested apps

- **OneNote**
- **Microsoft Flow Portal**
- **Copilot Studio – Dogfood**
- **make.powerapps.com**
- **Microsoft Teams Platform Monetization**

Who Is a Broker Client?

NAA is a new authentication protocol for SPAs that are embedded in host environments, such as Teams, Outlook, and Microsoft 365. It simplifies the authentication process to facilitate single sign-on (SSO) across apps nested within supported host apps. The NAA model supports a primary identity for the host app that includes multiple app identities for nested apps. Microsoft utilizes this model in Teams tabs, personal apps, and Office Add-ins.

SSO authentication for nested apps - Teams | Microsoft Learn

Broker Clients, We Discovered

7 broker clients can request tokens on behalf of the nested apps

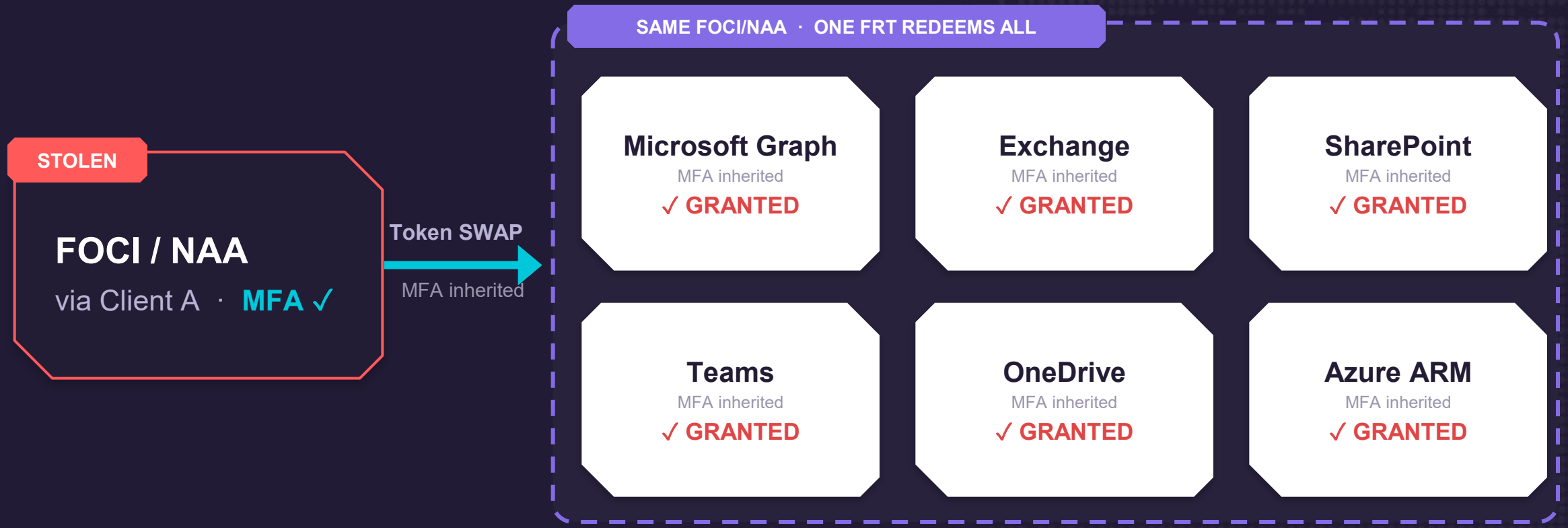
- Microsoft 365 Copilot
- Microsoft Teams
- Outlook Mobile
- Microsoft Outlook
- Microsoft Teams-T4L
- Microsoft Teams Web Client
- Microsoft Office

If we enable CA can Block NAA Expansion Risk. Is it true?

GENERAL SETTING: All resource need MFA

General Setting

Conditional Access · ALL resources require **MFA** · for ALL users

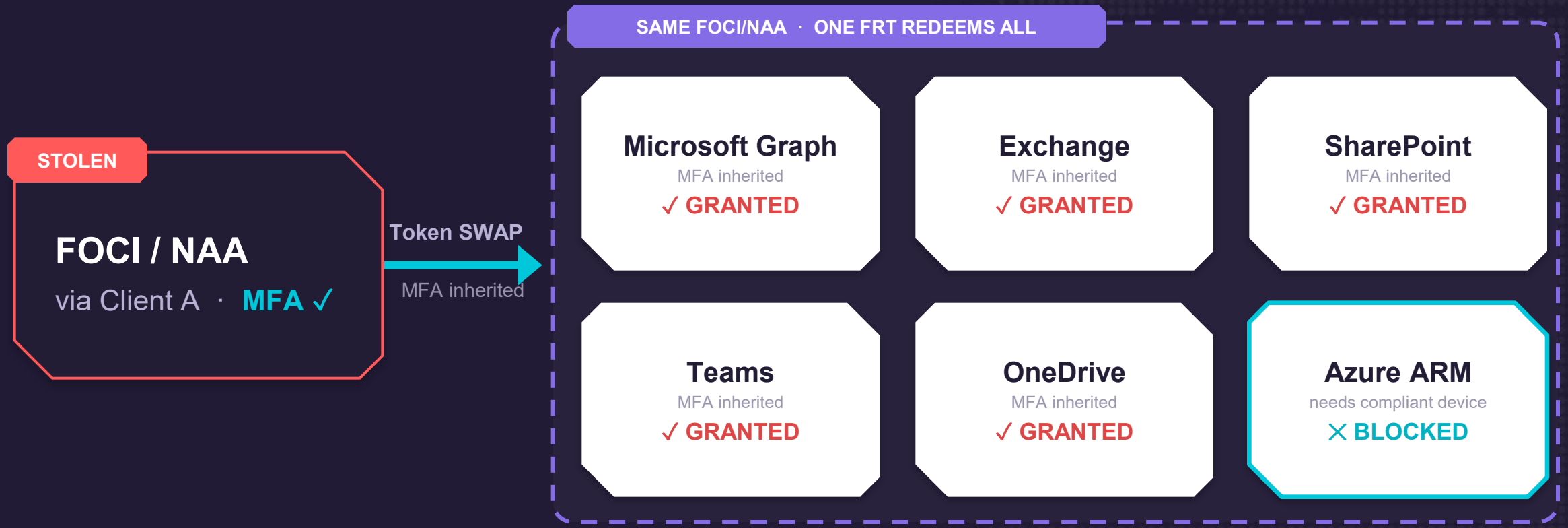


CYCRAFT

More Restrict CA Setting: Critical resource more restrict

More Restrict Setting

ALL resources require **MFA** · Azure ARM also needs **compliant device**



CYCRAFT


All resource require MFA
Case by Case set requirement
IT Team
JAKE-CLARK.TUMBLR

Is It Easy To Track The Attack? No.

Nested Application

Does not need to be installed on the device.

1/26/2026, 8:00:00	Aggregate	sigolon@8flw88.on...	OneNote	Success	[redacted]	Azure Resource Manager
1/26/2026, 8:00:00	8b1625a2-2392-405d-a	sigolon@8flw88.on...	Microsoft Teams	Success		App Studio for Microsoft Teams

Broker Application

Conditional Access Bypass

CA Bypass in Prior Research

Prior research focused on the **Scope-Based Bypass** — just change the Scope array.

NO BYPASS

scope = email
CA still enforced

BYPASS

scope = openid
CA bypassed

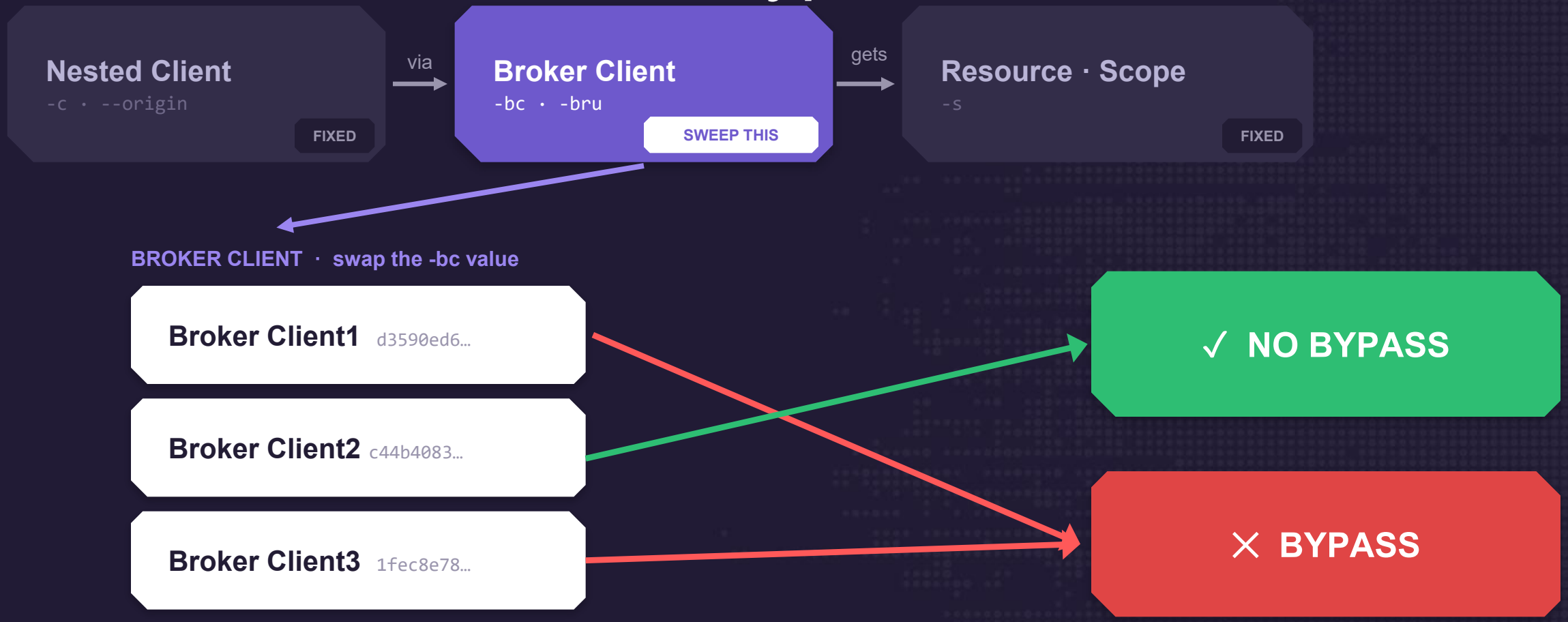
The Conditional Access Bypass within NAA

3 bypass methodologies	3 CA bypasses in NAA
Broker Client-Based (New)	MFA Bypass
Nested Client-Based (New)	Require Compliant Device Bypass
Scope-Based	Token Protection Bypass

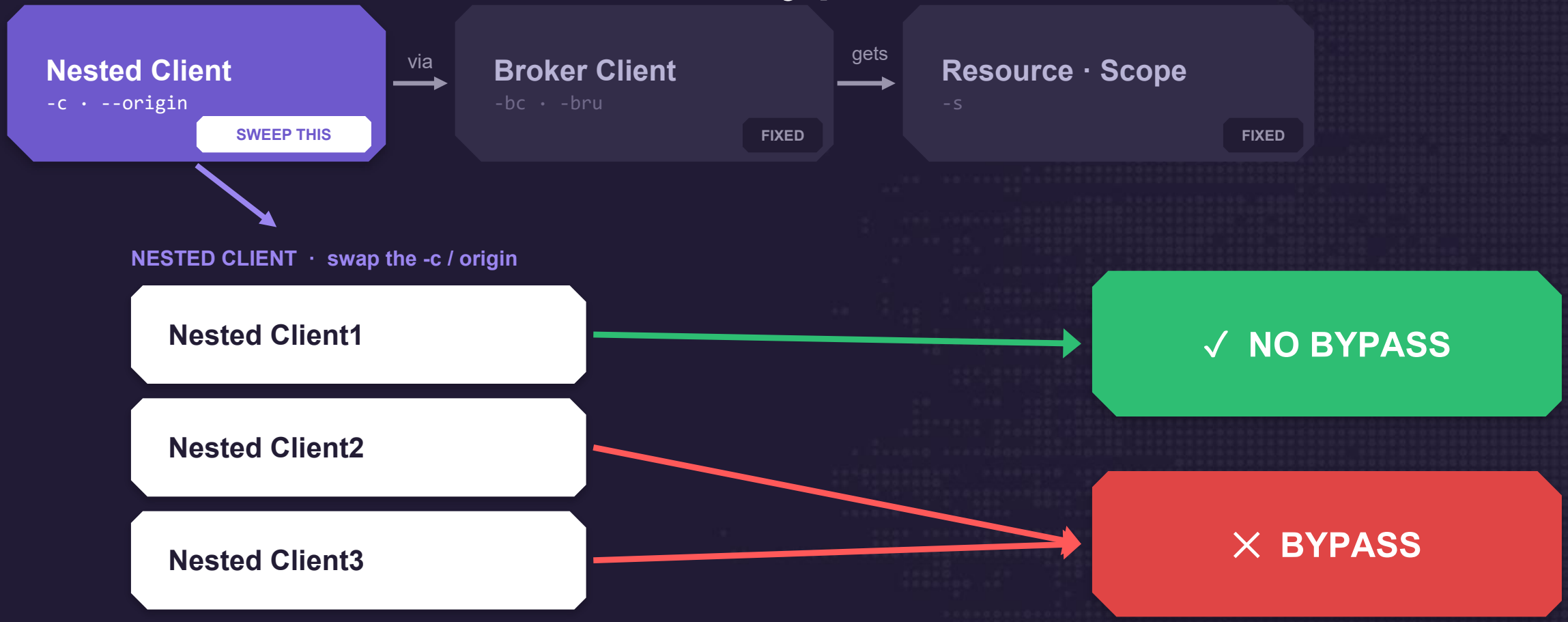
The Conditional Access Bypass within NAA

3 bypass methodologies	3 CA bypasses in NAA
Broker Client-Based (New)	MFA Bypass
Nested Client-Based (New)	Require Compliant Device Bypass
Scope-Based	Token Protection Bypass

Broker Client-Based Bypass



Nested Client-Based Bypass



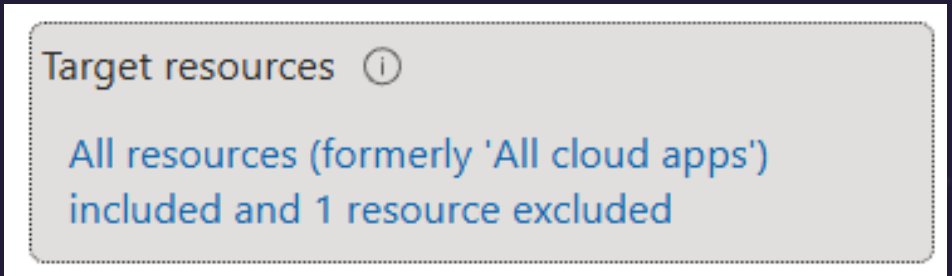
Bypass Pattern Analysis

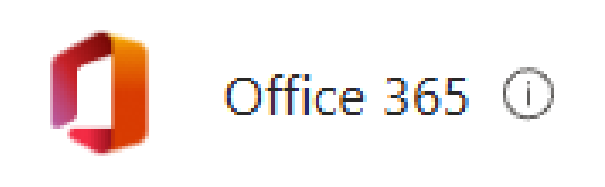
Exclude Logic Pattern of CA Bypass

- In our testing, the Conditional Access Policy was configured as :

  MFA Bypass and Device Compliance Bypass

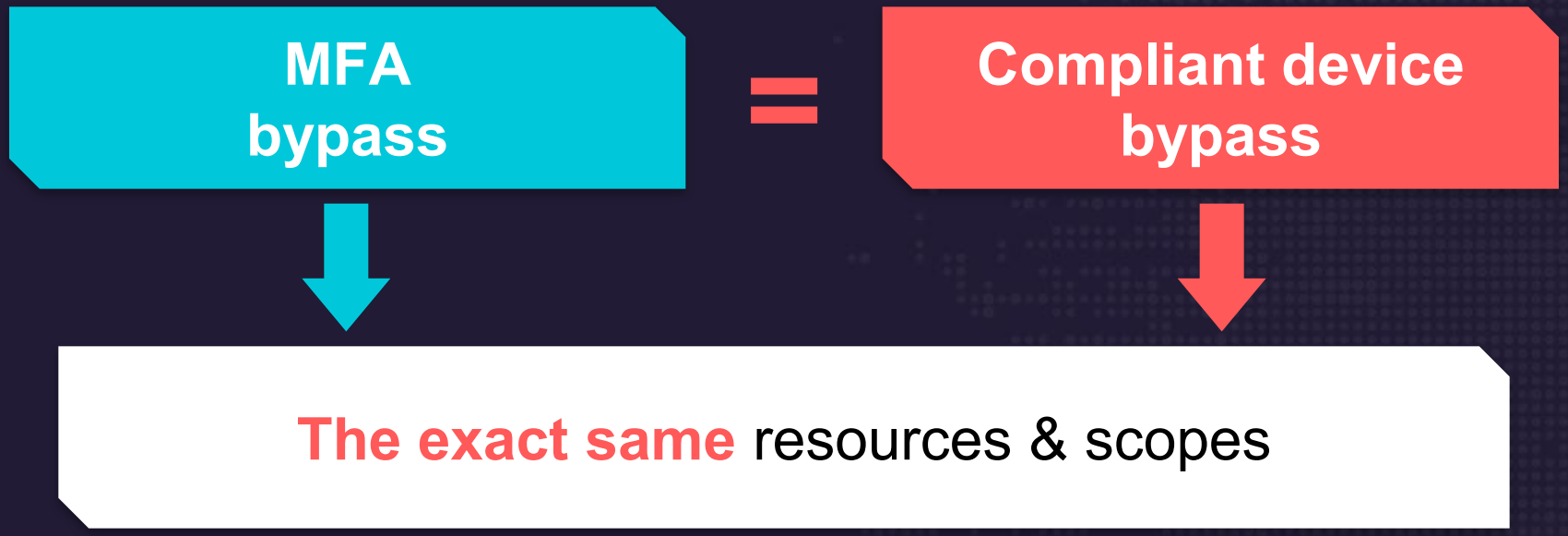
- These were set to apply to all resources, excluding Office 365.





Exclude Logic Pattern of CA Bypass

Surprisingly, the two bypasses are **completely identical** in what they access



HYPOTHESIS

CA bypass impact likely depends more on the **exclude configuration** than on the claim type the token requires.

Exclude Logic Bypass Impact

112
resource × scope combos

5
sensitive resources

Microsoft Graph

- user_impersonation
- Application.ReadWrite.All
- GroupMember.ReadWrite.All
- Directory.Read.All
- Files.ReadWrite.All

Bypass expands the range of **sensitive resources & scopes**

Exclude Logic Bypass — Expand Teams Access Scope

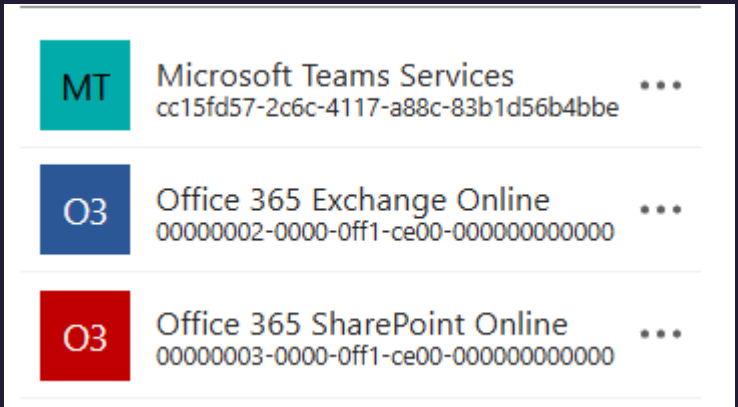
Teams not Enable Access These

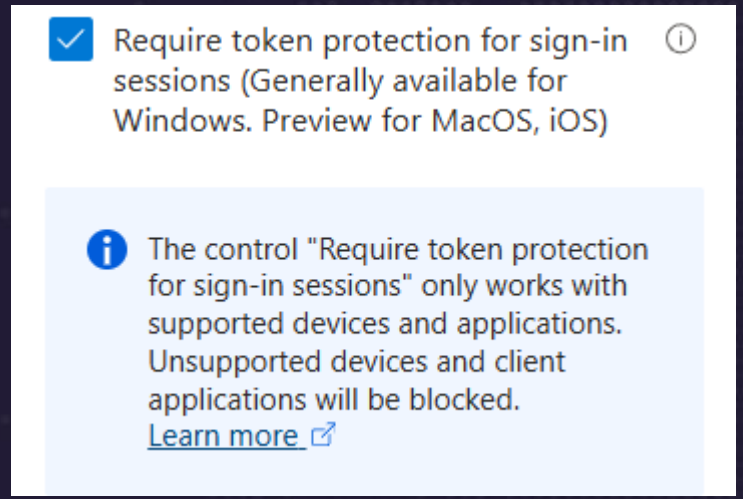
Microsoft Graph — now reachable

- user_impersonation
- Application.ReadWrite.All
- GroupMember.ReadWrite.All
- Directory.Read.All
- Files.ReadWrite.All

Include Logic Pattern of CA Bypass

- Based on the Conditional Access policy (three resources requiring token protection)
- There bypasses the attacker enabled allow access to 21 resources and 283 resource/scope combinations.





Include Logic Bypass Impact

221
resource × scope combos

6
sensitive resources

Impacted resources

- **Microsoft Graph** — 5 sensitive scopes
- **Windows Azure AD** — user_impersonation
- **Office 365 Exchange Online** — AdminApi.AccessAsUser.All
- **Office 365 SharePoint Online** — user_impersonation
- **Microsoft Teams Services** — user_impersonation
- **MS Teams Graph Service** — UserAuthSettings.Read

Include Logic Bypass Breaks ALL Protection

The resources we included to protect are exactly the ones **fully exposed**

Office 365 Exchange Online — **AdminApi.AccessAsUser.All**

Office 365 SharePoint Online — **user_impersonation**

Microsoft Teams Services — **user_impersonation**

Include-logic bypass breaks protection on **every included resource**

Future Research Path:
Include / Exclude Combination

Summary

Bypass severity is driven by the server-side **include / exclude logic** — not client-side parameters.

01 · FOCUS

Focus on include & exclude logic

02 · TEST

Test more combinations

03 · DISCOVER

Find new bypass patterns

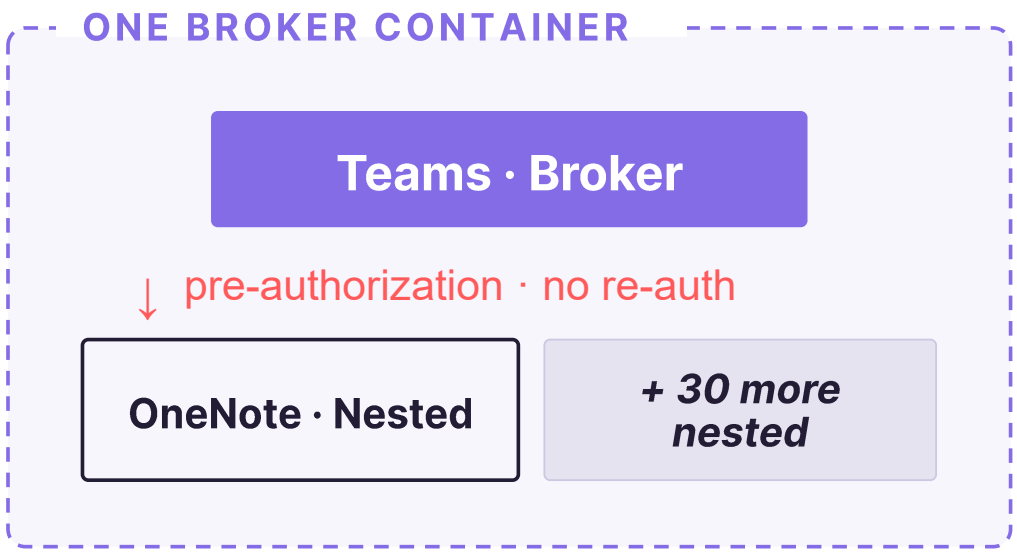
WHERE WE'VE LANDED

Recap & Takeaways.

RECAP 1 OF 2

What NAA really is — convenience, by design.

A broker pre-authorizes its nested apps to get tokens with no second sign-in.
FOCI's broker cousin — same recipe, new bottle. **More NAA broker has been discovered.**



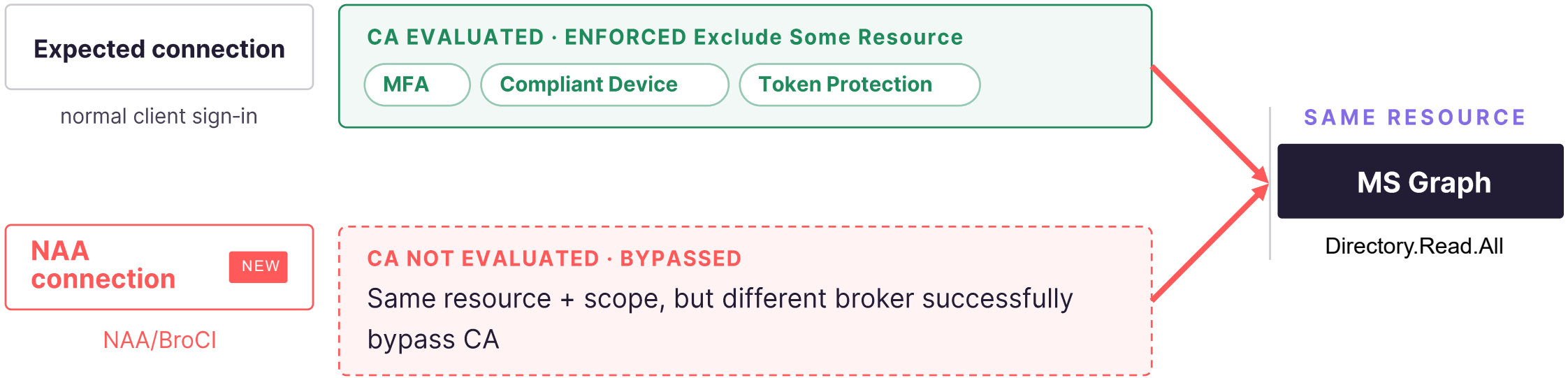
> Built for first-party UX — but that silently-granted trust **is** the attack surface.

TAKEAWAY | Tokens are interchangeable inside one broker container.

RECAP 2 OF 2

You think it's CA-protected.

Under specific conditional access policy , when you reach the same resource through a different NAA connection and those Conditional Access controls simply don't apply.



TAKEAWAY | Same resource, different connection method — Conditional Access silently stops applying.

Takeaways

1. NAA is a pre-auth scopes silently exchange different scope ARM / File / DevOps...
   Now more NAA broker has been discovered.

2. The token exchange inherits **MFA & device claims** and impact scope might same as FOCI — always assume **a stolen token = the nested-app is already open.**

3. Conditional Access is complex - **don't casually exclude app grouping(e.g. Office365).**
   The same applies to include - **scoping only include to specific apps is just as error-prone;**
   Always prefer "include all" to avoid conditional access gaps.

Thank You

Reference

Past Research

- 2024 DEFCON〈 Abusing Windows Hello Without a Severed Hand 〉
- 2025 TROOPERS〈 Finding Entra ID CA bypasses the structured way 〉
- 2025 Fabian Bader : Conditional Access bypasses - Cloudbrothers

Documents & Tools & Social Media

- https://learn.microsoft.com/en-us/microsoftteams/platform/concepts/authentication/nested-authentication
- https://learn.microsoft.com/zh-tw/azure/azure-resource-manager/management/overview
- https://learn.microsoft.com/en-us/entra/identity/conditional-access/policy-all-users-mfa-strength
- https://entrascopes.com/
- https://github.com/dirkjanm/ROADtools/wiki/ROADtools-Token-eXchange-(roadtx)
- https://github.com/R3alM0m1X82/SpecterBroker
- https://x.com/_dirkjan/status/2015812004326347169
- https://www.cleanpng.com/png-teams-logo-microsoft-teams-logo-with-group-of-peop-8021157/