JUMPSEC

OHHH365

How to (somewhat) reliably hack in to M365 & what to do afterwards

BSides Cheltenham 2024

Sunny Chau (sunnyc@jumpsec.com)

$ whoami

Consultant in adv simulation at JUMPSEC
(with them since 2021)

Past career as a paediatrician
(story of how I got into cyber, over drinks?)

Passion in all things red teaming, especially in the cloud
(that's what we do)

acid jazz & math rock are my jam.

JUMPSEC

$ whoami (ctrl-d)

Socials

https://www.linkedin.com/in/gladstomych/

https://twitter.com/gladstomych

Email

sunnyc@jumpsec.com

JUMPSEC

Caveat - lofty claims much?

> *“Somewhat reliably hack into M365” ?*

- Hopefully, your interest was piqued

You be the judge after the talk if I had overstated my claim

JUMPSEC

Why M365?

Historically:

- Outlook accounts were ***just*** mail inboxes
- BEC meant further internal phishing
- Leakage of sensitive email

Crown jewels (arguably) were internal AD servers

JUMPSEC

Why M365?

2024? Most orgs are Hybrid & Startups are cloud native!

M365 accounts - key to business apps

- SharePoint - means Data access
- Teams - sensitive chat log in addition to emails
- Key to Azure Resources
  - Admin accounts - resources & users
  - Dev & DevOps - pipeline compromise
- OAuth Integration - Lateral movement to SaaS apps

JUMPSEC

Why M365?

Imagine

- A user's M365 identity was compromised
- SharePoint(s) they could read+write, host critical business data
- There is no backup or disaster recovery plan

Smells like *Ransomware* … with lower tech than Lockbit

Deployable … in the web portal? In the form of the delete button

JUMPSEC

Why M365?

One of our cloud tenets:

Think of cloud identities like endpoint devices.

The bad guys see identities as entry points to cloud envs.

JUMPSEC

How to hack into it, though?

So, we've established M365 accounts are important

"Isn't that the reason why people entrust their data to MS?
Don't they have some of the best engineers?"

JUMPSEC

First, there is social engineering

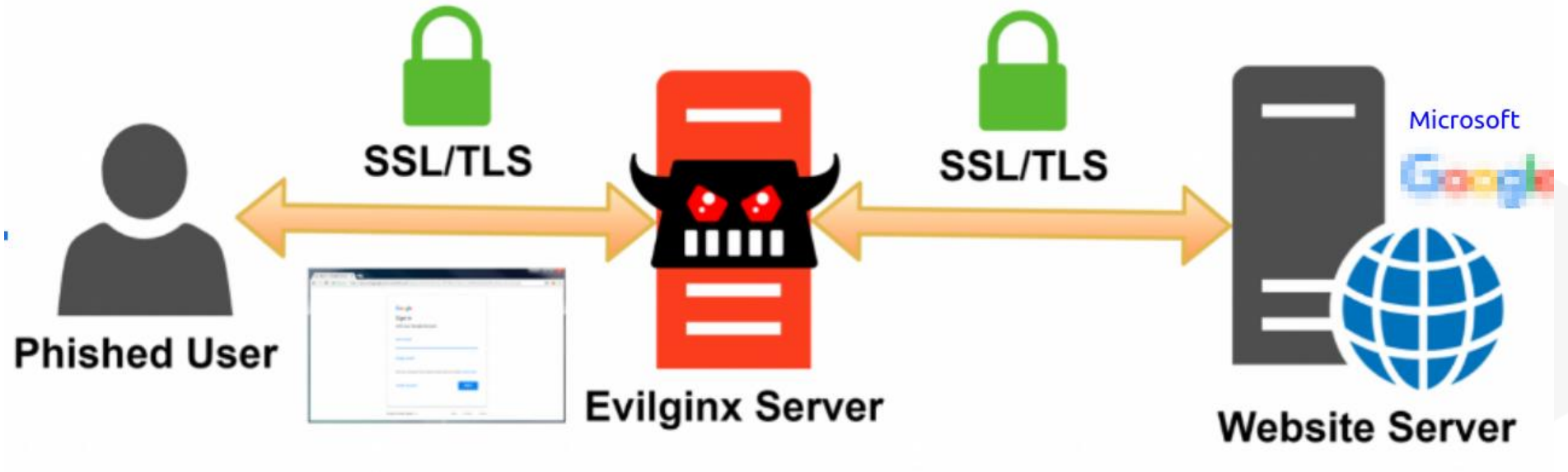
MiTM phishing - Evilginx

- Steals post-MFA token

- Traditional email-based delivery - Maturity of email filters
- MS Teams-based - Increasingly used by TAs

JUMPSEC

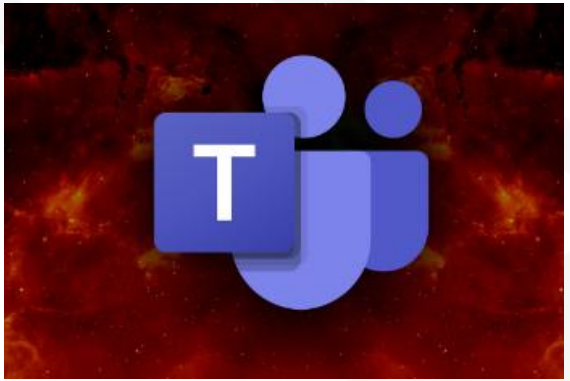
MiTM phishing visualised



JUMPSEC

Delivery of MiTM phishing link

Productivity app-based - MS Teams



(Default) External tenant allowed

- Messaging > user needs to unblock
- Calling victims via Teams
- No native way to filter phishing link

JUMPSEC

First, there is social engineering

Microsoft Teams vulnerability allows attackers to deliver malware to employees

Security researchers have uncovered a bug that could allow attackers to deliver malware directly into employees' Microsoft Teams inbox.

"Organisations that use Microsoft Teams inherit Microsoft's default configuration which allows users from outside of their organisation to reach out to their staff members," Jumpsec researcher Max Corbridge explained.

With a social engineering pretext to prime the target, a malware delivery attack exploiting this vulnerability has a considerable chance of success.

Bypassing security controls

Many organizations have permissive security controls that allow external tenants (M365 users outside the organization) to message their employees. There's a reason for that: they may want and need to allow communications via Teams

Shoutout to our research!

JUMPSEC

What if there's an alternative to Soc Eng?

JUMPSEC

Enter Jan 2024

Midnight Blizzard: Guidance for responders on nation-state attack

By Microsoft Threat Intelligence

The Microsoft security team detected a nation-state attack on our corporate systems on January 12, 2024, and immediately activated our response process to investigate, disrupt malicious activity, mitigate the attack, and deny the threat actor further access. The Microsoft Threat Intelligence investigation identified the threat actor as Midnight Blizzard, the Russian state-sponsored actor also known as NOBELIUM. The latest information from the Microsoft Security and Response

JUMPSEC

Midnight Blizzard observed activity and techniques

Initial access through password spray

Midnight Blizzard utilized password spray attacks that successfully compromised a legacy, non-production test tenant account that did not have multifactor authentication (MFA) enabled. In a password-spray attack, the adversary attempts to sign into a large volume of accounts using a small subset of the most popular or most likely passwords. In this observed Midnight Blizzard activity, the actor tailored their password spray attacks to a limited number of accounts, using a low number of attempts to evade detection and avoid account blocks based on the volume of failures. In addition, as we explain in more detail below, the threat actor further reduced the likelihood of discovery by launching these attacks from a distributed residential proxy infrastructure. These evasion techniques helped ensure the actor obfuscated their activity and could persist the attack over time until successful.

A Password Spray is …

Light on passwords

10-20

Heavy on users

Hundreds

*in our engagements

JUMPSEC

A Password Spray is …

A statistical attack that one of your hundreds of users' password is:

Welcome@2024

Spring@2024

Welcome2024!

Contoso@2024

*note the capitalisation

JUMPSEC

Theory (?) behind

A “complex” password isn’t necessarily good, an unguessable one is.

(if you’re a sysadmin, you know!)

Users’ gonna user.

Statistics are on the attackers’ side when you have hundreds of users

JUMPSEC

A Password Spray is … wait

What about MFA?

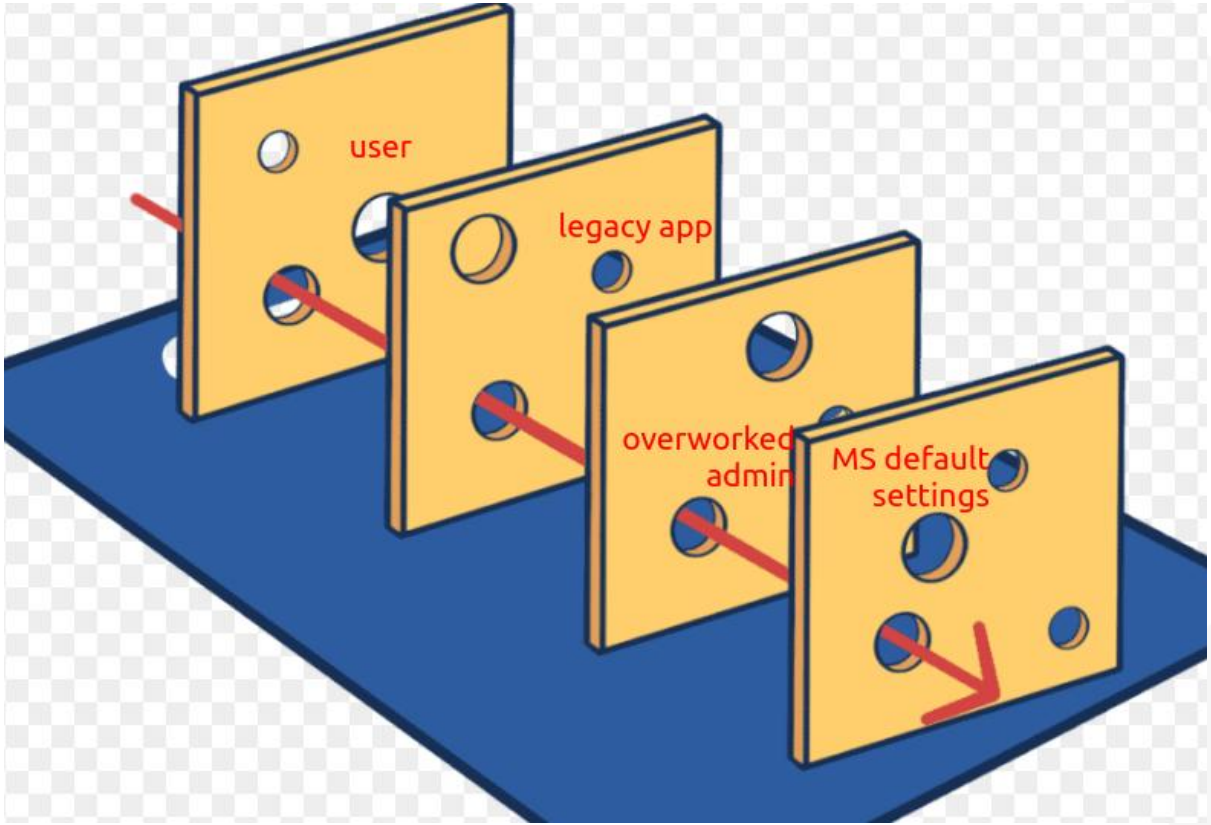
Reasons accounts lack MFA

- Service accounts that does not support MFA usage.
- New starter, no company device yet
- Leaver accounts from a time before MFA implemented.
- Legacy applications the business needs, does not support MFA.
- …

MFA requirement != MFA enrollment

JUMPSEC

Is what it is

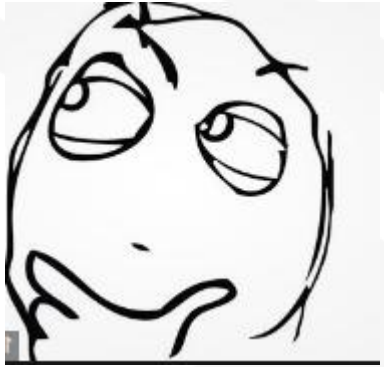


JUMPSEC

Interim Summary

To hack into M365 -

- Steal session token with MiTM phishing
- Weak password + no MFA...?



JUMPSEC

Last line of defence

MS Entra ID Smart Lockout

Default settings - lock after

- 10 fail attempts
- Locks again with each subsequent failure
- Undisclosed incremental timer
- Differentiates familiar vs unfamiliar location
    - *User & attacker have different timers!*

JUMPSEC

Last line of defence

MS Entra ID Smart Lockout

Default settings

- User can self-serve password reset
- Enable by default
- Available to **ALL** M365 & Azure tenants
- And also …

Default protections

In addition to Smart lockout, Microsoft Entra ID also protects against attacks by analyzing signals including IP traffic and identifying anomalous behavior. Microsoft Entra ID blocks these malicious sign-ins by default and returns AADSTS50053 - IdsLocked error code, regardless of the password validity.

JUMPSEC

Distributed Password Spraying?

Default protections

In addition to Smart lockout, Microsoft Entra ID also protects against attacks by analyzing signals including IP traffic and identifying anomalous behavior. Microsoft Entra ID blocks these malicious sign-ins by default and returns AADSTS50053 - IdsLocked error code, regardless of the password validity.

For our emulation of adversaries, rather than through residential proxy:

Fireprox i.e. AWS API gateway proxying

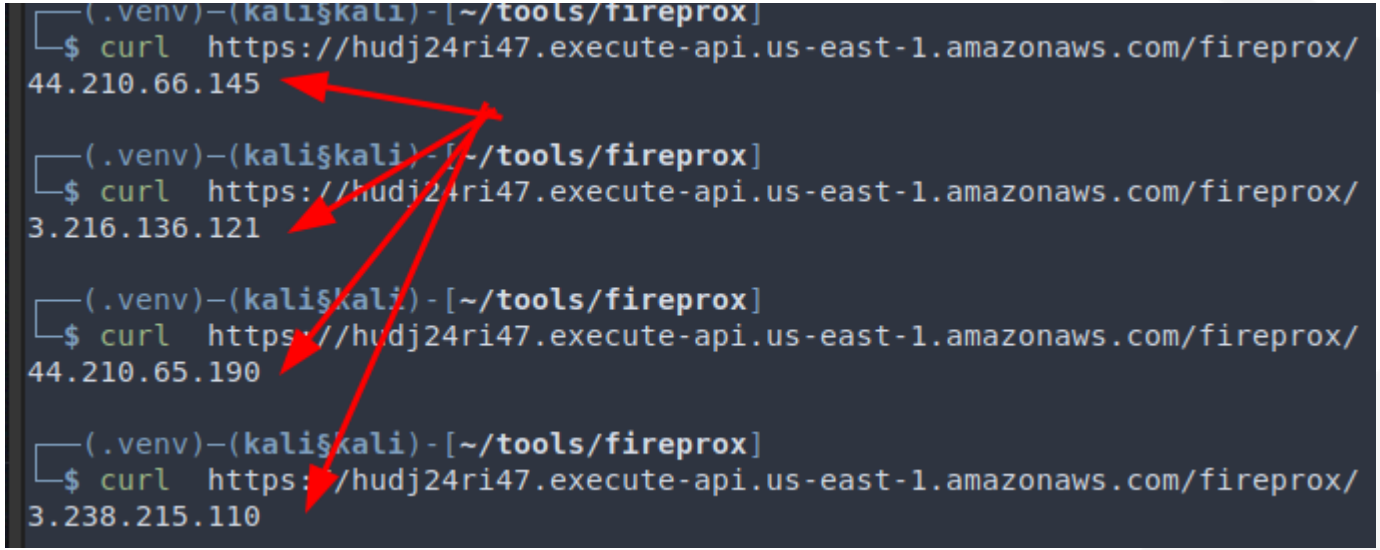
Proxy your request through AWS API gateways

Different IP, different region in the word, in every req

JUMPSEC

Built-in IP rotation

curl https://ifconfig.me behind the gateway



JUMPSEC

Other heuristics & blocks?

- Long duration in-between each user is hit again (> 1 hour)
  - For SmartLockout timer
- Scrambling user list
- Jittering
  - user A -[10 sec]- user B -[7.41 sec]- user C -[8.13 sec]- **...**
- **Rotating IP address & Region**

JUMPSEC

What tools spray like this

- TeamFiltration
- Credmaster
- o365spray
- go365
- spraycharles

Our red team uses TeamFiltration for the features we need;
Tho all of them use Fireprox / AWS API gateway for IP rotation

JUMPSEC

TeamFiltration Intro

--enum : validate users through Teams or MSOL API

--spray : low and slow password spray

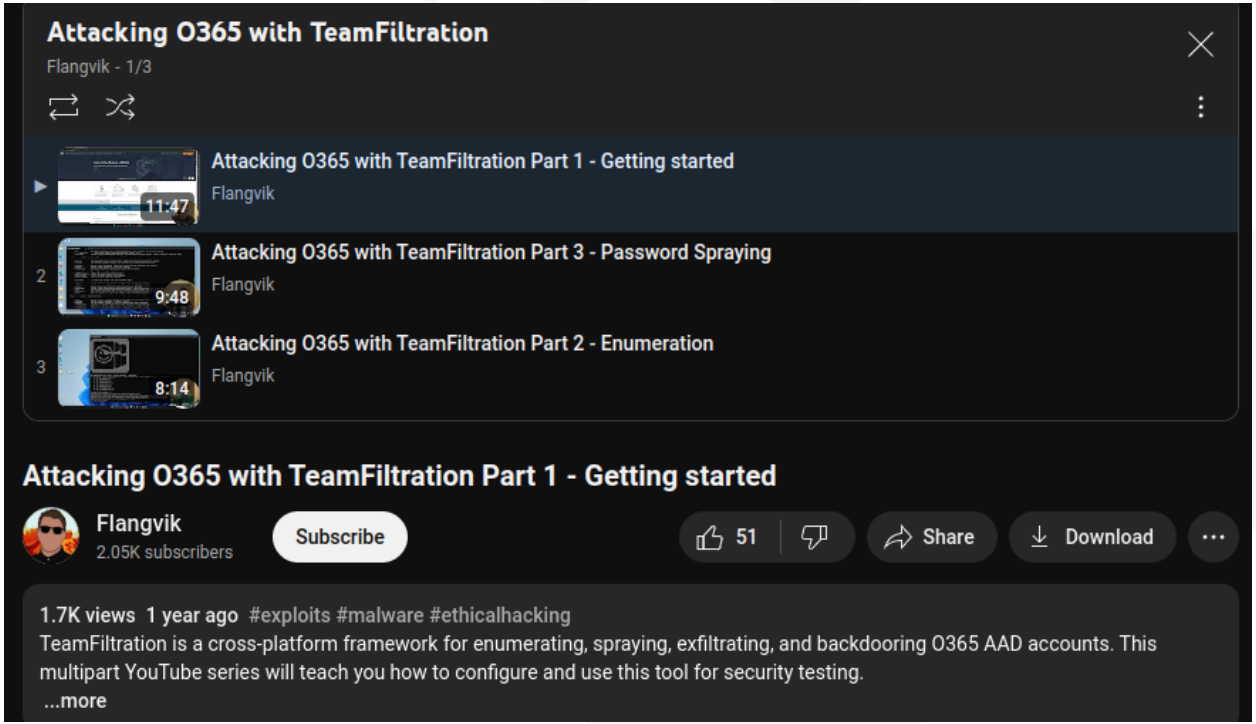
--exfil : look for an MFA gap, login and loot!

JUMPSEC

How to set up & use TeamFiltration

Flangvik (Author)'s Youtube playlist

"**Attacking O365 with TeamFiltration**"



JUMPSEC

Spray

```
$ ./TeamFiltration --config TFconfig_clientA.json \
    --outpath . --spray --passwords passwords.txt \
    --sleep-min 60 --sleep-max 70 --jitter 10 \
    --shuffle-users --shuffle-regions
```

JUMPSEC

What happens during a spray

`--sleep-min 60 --sleep-max 70` `--jitter 10`

In our engagements, usually we do 1000 - 5000 login requests

The math goes:

- 8 business hour day, 200 users / rounds, ~1 hour between each round > 1.6k reqs per day
- A dedicated password spray can take 1-3 days to perform
- Viable to fit into many types of engagement windows
- Stop and restart at any time
- QoL - bingo webhook, run in background while you hack other stuff, etc.

JUMPSEC

Tips and Tricks - quick fire

1. `--common-only` flag for passwords is surprisingly good
2. Spin up a testing M365 tenant (1 month free!)
    - spray it, observe the errors, what works, what does not
    - Azure error messages & error codes
    - be very sure about what the tool does (read the src, it's C#)
3. **Always** ask for permission before starting a spray because lockouts are unlikely but real
    - Good idea to spray during office hours only
    - Start with a small subset of users first; confirm list with client
    - If we make risk / value proposition clear, clients are usually okay with this technique.

JUMPSEC

Note on alert & detection

How loud are thousands of failed login reqs?

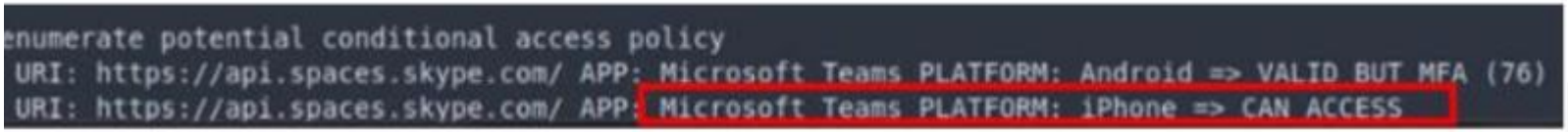
By default

- Alerting starts at a **successful** login (username: password pair)
  - Unsuccessful attempts are logged but no default alerting
  - Some SOCs write custom detection rules on top of Smart Lockout

- We recommend using creds as soon as you got them, before pwd reset etc.

JUMPSEC

After a hit - MFA gap enum



Enumerate potential conditional access policy

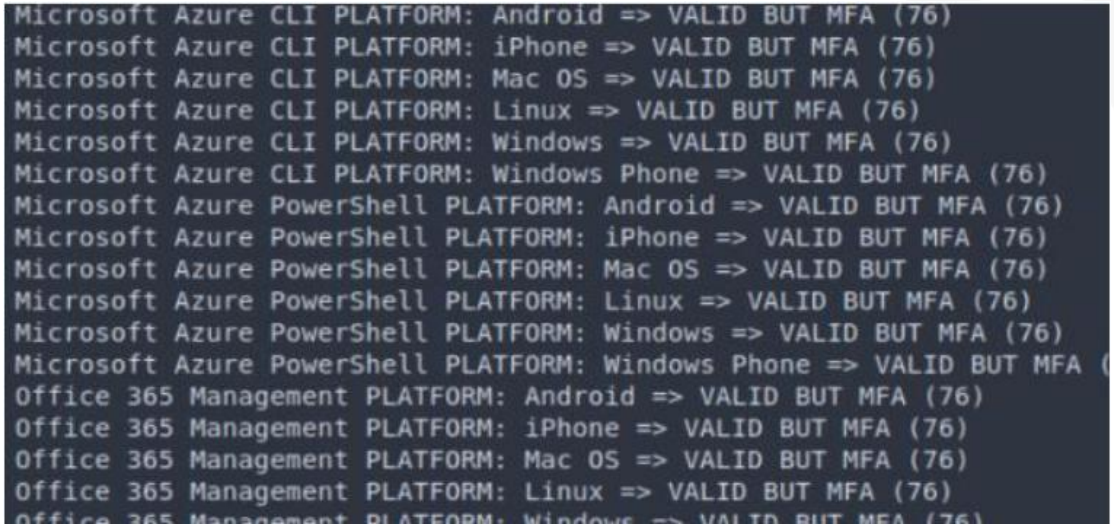
URI: https://api.spaces.skype.com/ APP: Microsoft Teams: Android => VALID BUT MFA (76)

URI: https://api.spaces.skype.com/ APP: Microsoft Teams: iPhone => CAN ACCESS

Example - MFA Gap present

JUMPSEC

After a hit - MFA gap enum



MFA is air tight for this user

JUMPSEC

Post Ex - Exfiltration

```
EST Refreshed a token for => https://outlook.office365.com
EST Exfiltrating emails from Outlook!
EST Fetched 2000 email ID's , exfiltrating content!
EST Found valid access token in database for => https://api.spaces.skype.com
EST Refreshed a token for => https                    sharepoint.com
EST Refreshed a token for => https                    .sharepoint.com
EST Exfiltrating recently used contacts
EST Exfiltrating Teams chat logs and attachments
EST Parsing conversations
EST Found valid access token in database for => https:/                    sharepoint.c
EST Found valid access token in database for => https:/                    sharepoint.c
EST Found valid access token in database for => https://outlook.office365.com
EST Refreshed a token for => https://management.core.windows.net
EST Found valid access token in database for => https://graph.microsoft.com
EST Found valid access token in database for => https://graph.windows.net
EST Exfiltrating shared files from OneDrive
EST Exfiltrating the entire personal OneDrive
EST                    docx
EST                    .xlsx                    200 MiB of goodies ...
EST                    .docx
```

JUMPSEC

Post Ex - Persistence and Lateral Mvt

19 OCT 2023

BEAU BULLOCK, HOW-TO, RED TEAM, RED TEAM TOOLS, STEVE BOROSH AZURE, CLOUD, MICROSOFT365

Introducing GraphRunner: A Post-Exploitation Toolset for Microsoft 365

By Beau Bullock & Steve Borosh

JUMPSEC

Limitations

- No user found on OSINT / Small company (< 50 users)
  - Soc Eng might be a better approach
- Airtight setup through and through
  - No MFA Gap
  - Everyone has strong passwords
  - SOC wrote custom detections & jump on you within 5 minutes
  - It's good to tell a client they have done well!

JUMPSEC

Tooling Shill

Single-command install

Your own vulnerable Azure Tenant to hack

5 flags (so far), Hints on repo

Walkthroughs planned - will be on **https://labs.jumpsec.com**

https://github.com/gladstomych/AHHHZURE

Check it out, give a star if you like what you see!

JUMPSEC

Credits

Teamfiltration - Credits to Flangvik @ TrustedSec

Fireprox - ustrayready @ Black Hills InfoSec

Graphrunner - Credits to dafthack @ Black Hills InfoSec

JUMPSEC

Q&A

JUMPSEC