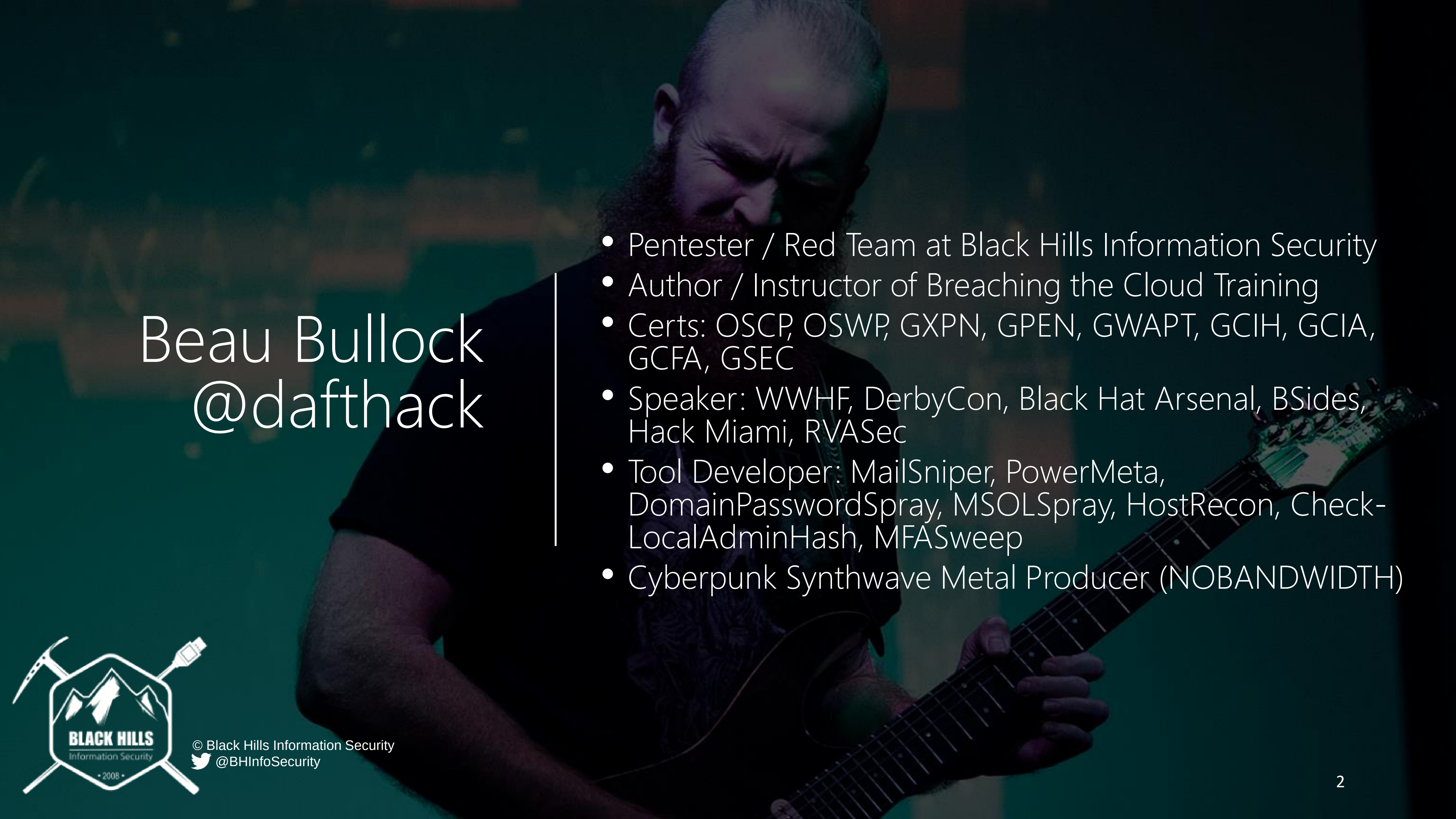
Getting Started in Pentesting the Cloud: Azure



Beau Bullock
@dafthack

- Pentester / Red Team at Black Hills Information Security
- Author / Instructor of Breaching the Cloud Training
- Certs: OSCP, OSWP, GXPN, GPEN, GWAPT, GCIH, GCIA, GCFA, GSEC
- Speaker: WWHF, DerbyCon, Black Hat Arsenal, BSides, Hack Miami, RVASec
- Tool Developer: MailSniper, PowerMeta, DomainPasswordSpray, MSOLSpray, HostRecon, Check-LocalAdminHash, MFASweep
- Cyberpunk Synthwave Metal Producer (NOBANDWIDTH)

Roadmap

- Identifying Attack Surface
- Recon & External Attacks
- Authentication
- Post-Compromise
- Azure Subscription Hierarchy
- Resource-Specific Issues
- Leveraging Scanning Tools

Why Azure?

- Extremely popular for productivity and compute resources
- Hybrid environments make cloud to on-prem pivoting possible
- Publicly accessible authentication
- More SharePoint usage facilitates accessibility to sensitive data
- Azure Pentesting techniques apply to multiple different types of engagements (Red Team, External, Assumed Compromise, WebApps, etc.)

Identifying Attack Surface

Identifying Attack Surface

- External
  - Unauthenticated
  - Attacking public resources
- Internal (Resource access)
  - Testing internal cloud resources from another resource such as a VM
- Internal (API access)
  - Authenticated
  - Identify vulnerabilities via API calls & configuration analysis

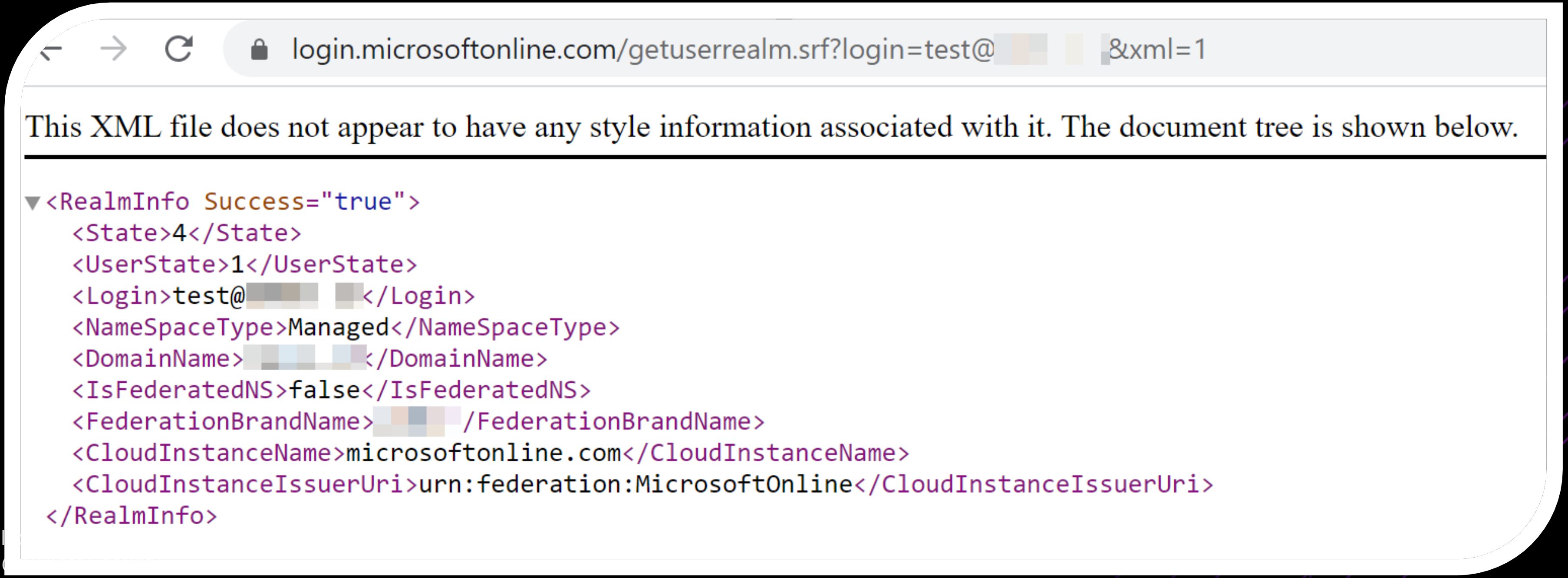
Azure RM vs Microsoft 365

- Azure Resource Manager
  - Subscriptions and Resources
    - VMs
    - Databases
    - Storage
    - Serverless
    - Many more…
- Microsoft 365
  - Productivity
    - Outlook
    - SharePoint
    - Teams

Recon & External Attacks

Recon: Cloud Asset Discovery

- Identify Microsoft 365 Usage
  - https://login.microsoftonline.com/getuserrealm.srf?login=username@acmecomputercompany.com&xml=1
  - https://login.microsoftonline.com/<target domain>/v2.0/.well-known/openid-configuration



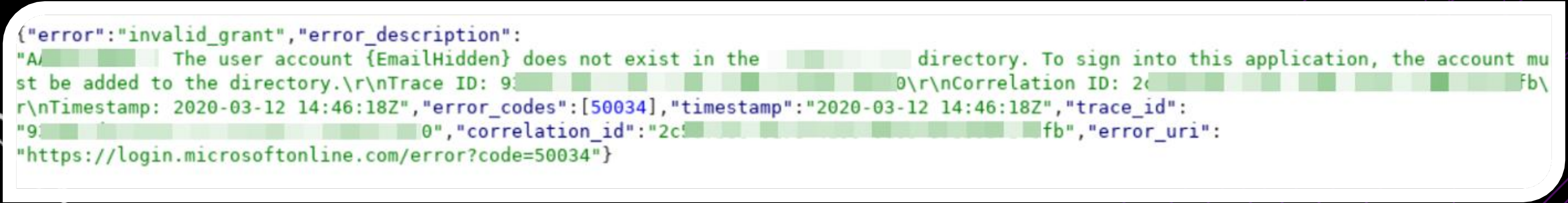
login.microsoftonline.com/getuserrealm.srf?login=test@[redacted]&xml=1

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<RealmInfo Success="true">
   <State>4</State>
   <UserState>1</UserState>
   <Login>test@[redacted]</Login>
   <NameSpaceType>Managed</NameSpaceType>
   <DomainName>[redacted]</DomainName>
   <IsFederatedNS>false</IsFederatedNS>
   <FederationBrandName>[redacted]/FederationBrandName>
   <CloudInstanceName>microsoftonline.com</CloudInstanceName>
   <CloudInstanceIssuerUri>urn:federation:MicrosoftOnline</CloudInstanceIssuerUri>
 </RealmInfo>
```

Recon: User Enumeration

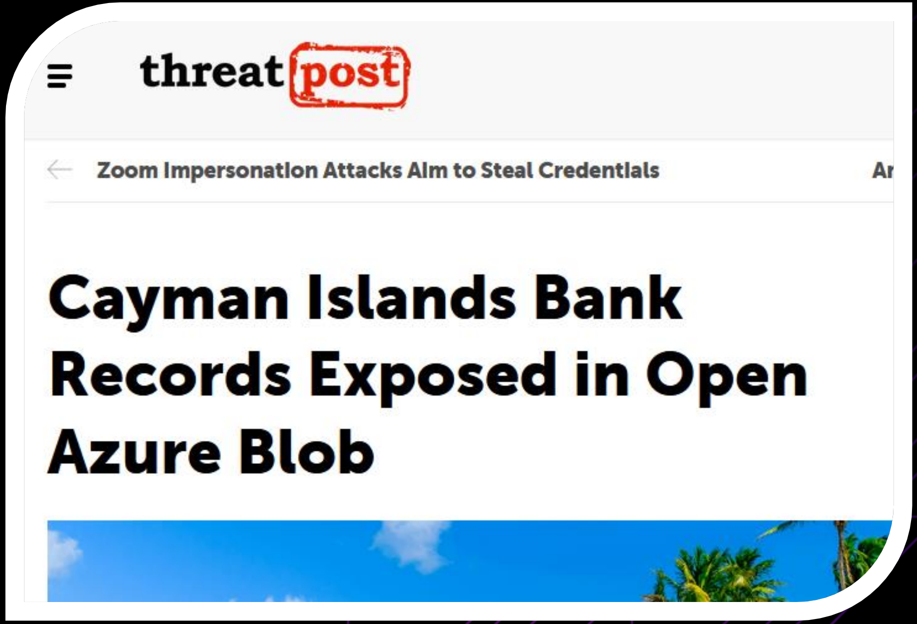
- User enumeration on Azure can be performed at https://login.Microsoft.com/common/oauth2/token
- This endpoint tells you if a user exists or not
- Detect invalid users while password spraying with:
  - https://github.com/dafthack/MSOLSpray
- May be able to enumerate users via OneDrive
  - https://github.com/nyxgeek/onedrive_user_enum



```
{"error":"invalid_grant","error_description":
"A[redacted] The user account {EmailHidden} does not exist in the [redacted] directory. To sign into this application, the account mu
st be added to the directory.\r\nTrace ID: 9[redacted]0\r\nCorrelation ID: 2[redacted]fb\
r\nTimestamp: 2020-03-12 14:46:18Z","error_codes":[50034],"timestamp":"2020-03-12 14:46:18Z","trace_id":
"9[redacted]0","correlation_id":"2c[redacted]fb","error_uri":
"https://login.microsoftonline.com/error?code=50034"}
```

Data in Public Azure Blobs

- Microsoft Azure Storage is like Amazon S3
- Blob storage is for unstructured data
- Containers and blobs can be publicly accessible via access policies
- Predictable URL’s at core.windows.net
  - storage-acct-name.blob.core.windows.net
  - storage-acct-name.file.core.windows.net
  - storage-acct-name.table.core.windows.net
  - storage-acct-name.queue.core.windows.net



threatpost

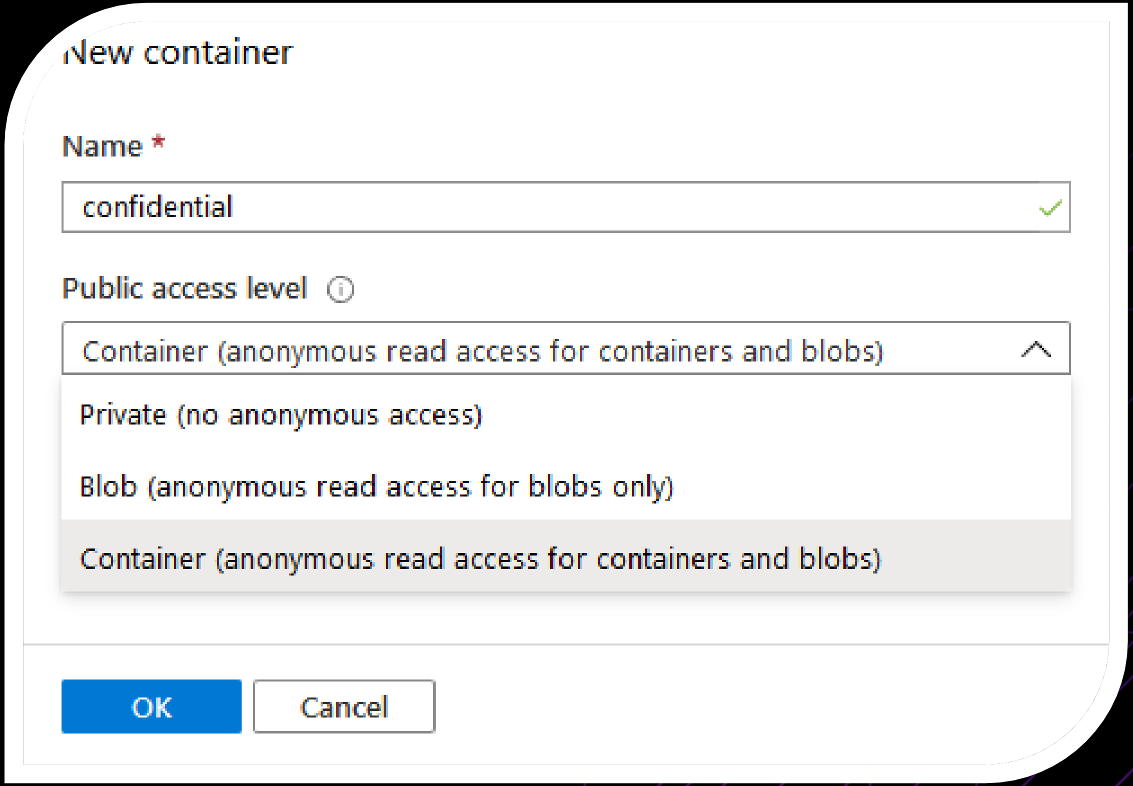
Zoom Impersonation Attacks Aim to Steal Credentials

Ar

Cayman Islands Bank Records Exposed in Open Azure Blob

Data in Public Azure Blobs

- The "Blob" access policy means anyone can anonymously read blobs, but can't list the blobs in the container
- The "Container" access policy allows for listing containers and blobs



New container

Name *

confidential

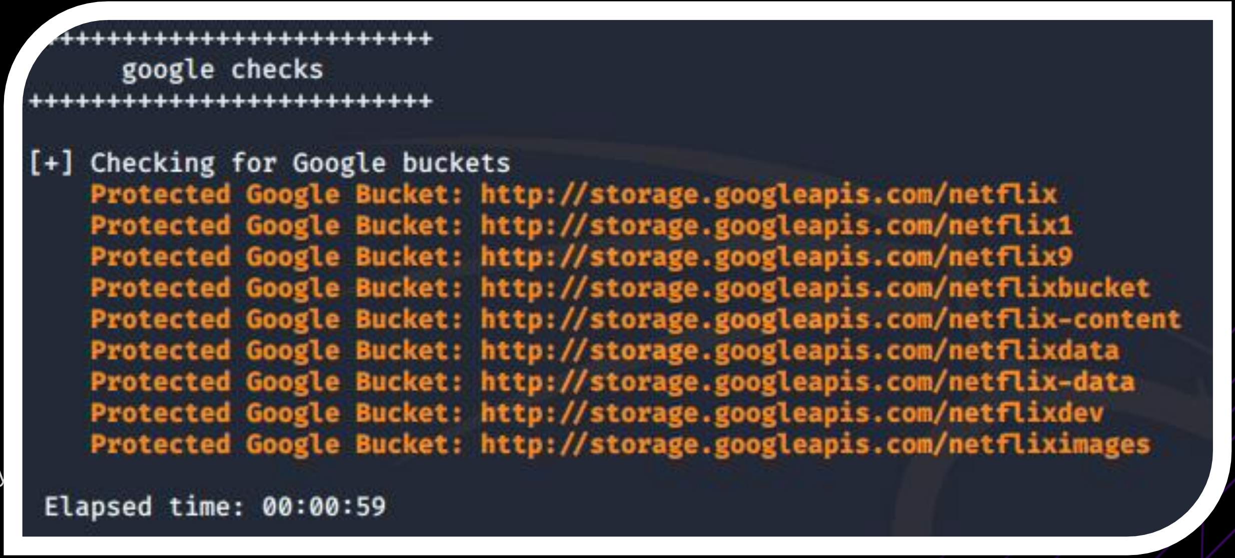
Public access level

Container (anonymous read access for containers and blobs)

Private (no anonymous access)
Blob (anonymous read access for blobs only)
Container (anonymous read access for containers and blobs)
OK
Cancel

Cloud_enum

- Cloud_enum from Chris Moberly (@initstring)
  - https://github.com/initstring/cloud_enum
  - Awesome tool for scanning Azure, AWS, & GCP for buckets and more
  - Enumerates:
    - GCP open and protected buckets as well as Google App Engine sites
    - Azure storage accounts, blob containers, hosted DBs, VMs, and WebApps
    - AWS open and protected buckets



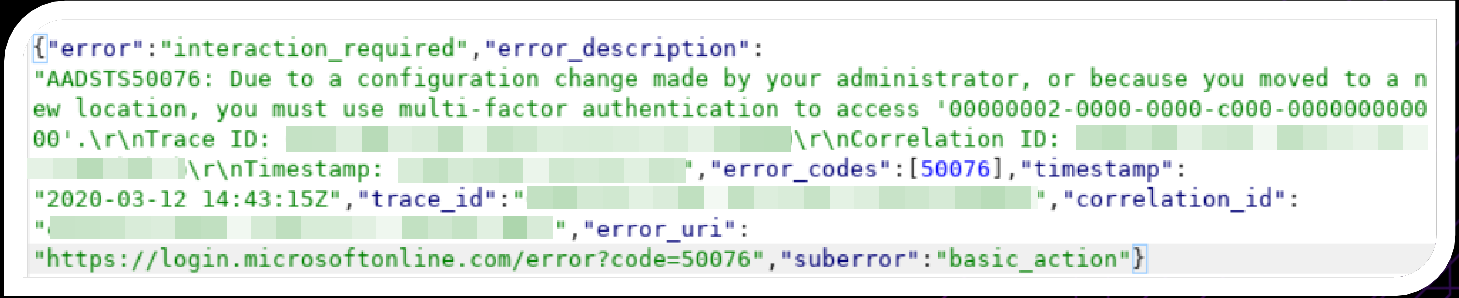
```
+++++++++++++++++++++++++++
      google checks
+++++++++++++++++++++++++++

[+] Checking for Google buckets
    Protected Google Bucket: http://storage.googleapis.com/netflix
    Protected Google Bucket: http://storage.googleapis.com/netflix1
    Protected Google Bucket: http://storage.googleapis.com/netflix9
    Protected Google Bucket: http://storage.googleapis.com/netflixbucket
    Protected Google Bucket: http://storage.googleapis.com/netflix-content
    Protected Google Bucket: http://storage.googleapis.com/netflixdata
    Protected Google Bucket: http://storage.googleapis.com/netflix-data
    Protected Google Bucket: http://storage.googleapis.com/netflixdev
    Protected Google Bucket: http://storage.googleapis.com/netfliximages

 Elapsed time: 00:00:59
```

Password Attacks

- Password Spraying
  - Trying one password for every user at an org to avoid account lockouts
  - Most systems have some sort of lockout policy
    - Example: 5 attempts in 30 mins = lockout
  - If we attempt to auth as each individual username one time every 30 mins we lockout nobody

Password Attacks

- Can use MSOLSpray to spray Azure users
- The script logs:
    - If a user cred is valid
    - If MFA is enabled on the account
    - If a tenant doesn't exist
    - If a user doesn't exist
    - If the account is locked
    - If the account is disabled
    - If the password is expired



```
{"error":"interaction_required","error_description":
"AADSTS50076: Due to a configuration change made by your administrator, or because you moved to a n
ew location, you must use multi-factor authentication to access '00000002-0000-0000-c000-0000000000
00'.\r\nTrace ID: [redacted]\r\nCorrelation ID: [redacted]
\r\nTimestamp: [redacted]","error_codes":[50076],"timestamp":
"2020-03-12 14:43:15Z","trace_id":"[redacted]","correlation_id":
"[redacted]","error_uri":
"https://login.microsoftonline.com/error?code=50076","suberror":"basic_action"}
```

Password Protection & Smart Lockout

- Azure Password Protection
  - Prevents users from picking passwords with certain words like seasons, company name, etc.
- Azure Smart Lockout
  - Locks out auth attempts whenever brute force or spray attempts are detected
  - Can be bypassed with FireProx + MSOLSpray
    - https://github.com/ustayready/fireprox

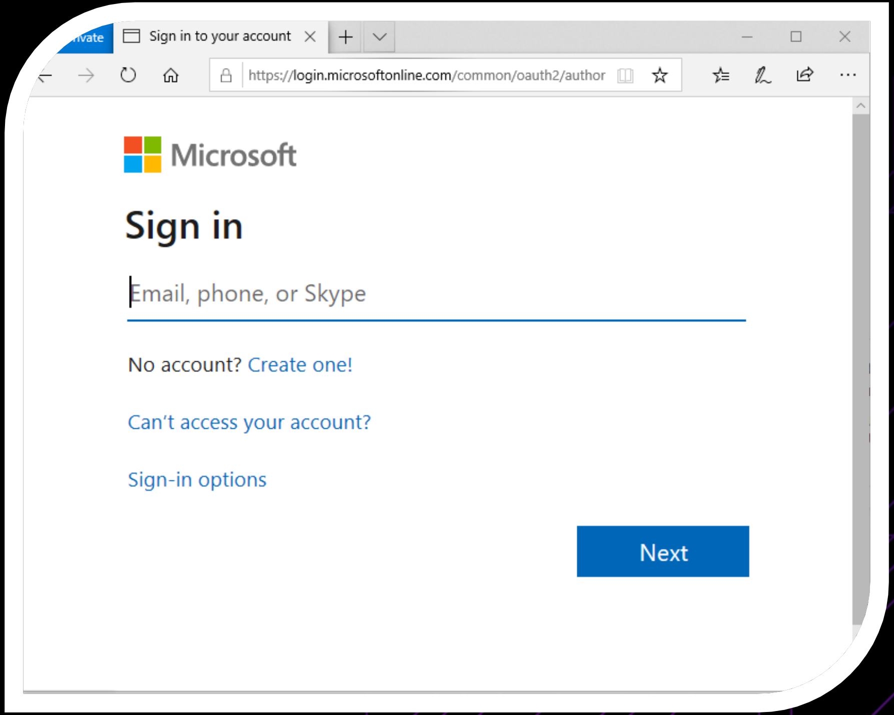
Authentication

Azure Authentication

- More ways to authenticate to cloud providers than just username and password
- API's, certificates, and more
- Multi-Factor settings might differ for things like service accounts or those that authenticate with certs
- Sometimes keys get posted publicly with code to repos
- Finding authentication points is a key first step

Cloud Authentication Methods

- Forms of authentication to consider…
  - Password Hash Synchronization
  - Pass Through Authentication
  - Active Directory Federation Services (ADFS)
  - Certificate-based auth
  - Conditional access policies
  - Long-term access tokens
  - Legacy authentication portals



Sign in

Email, phone, or Skype

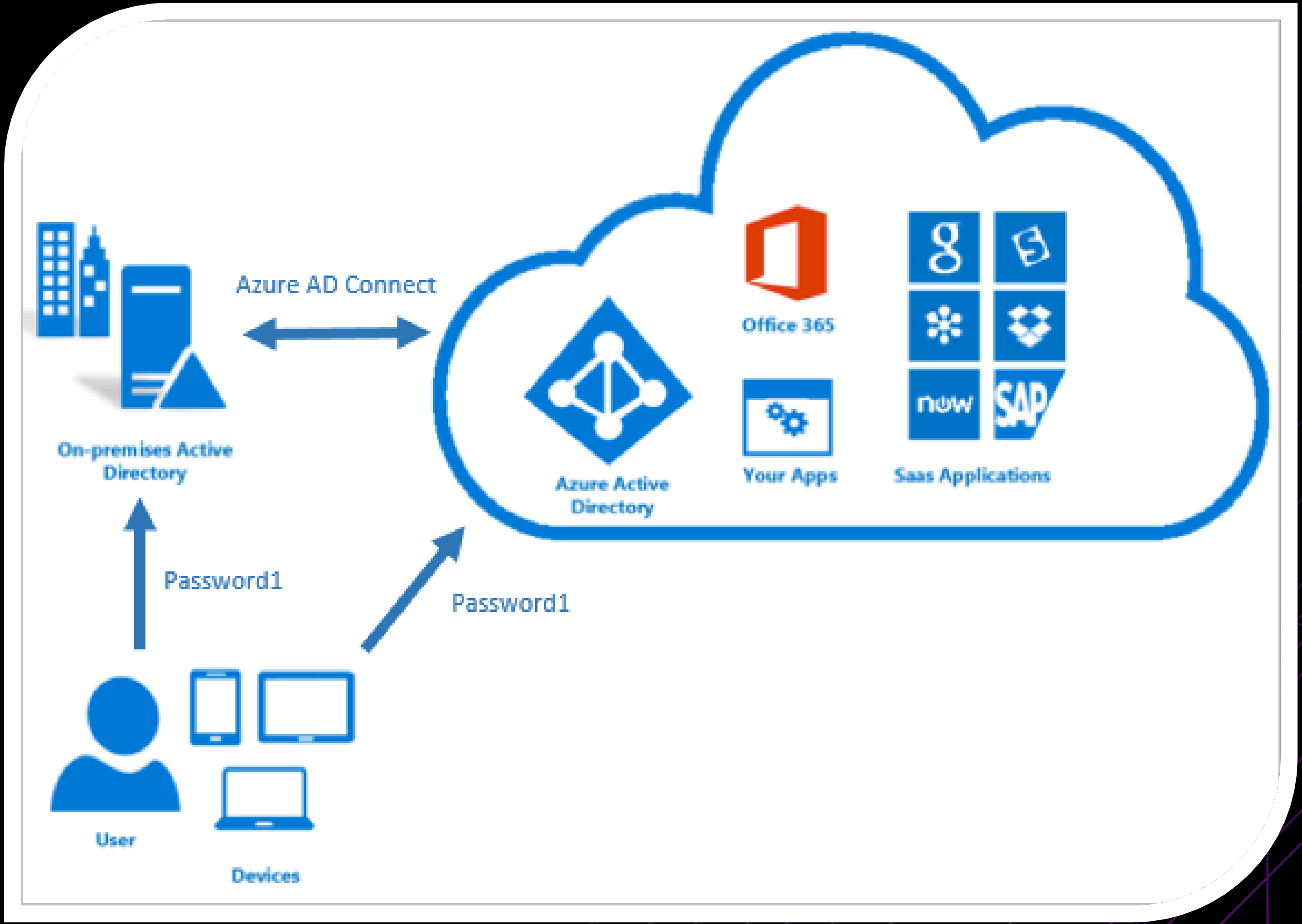
No account? Create one!

Can't access your account?

Sign-in options

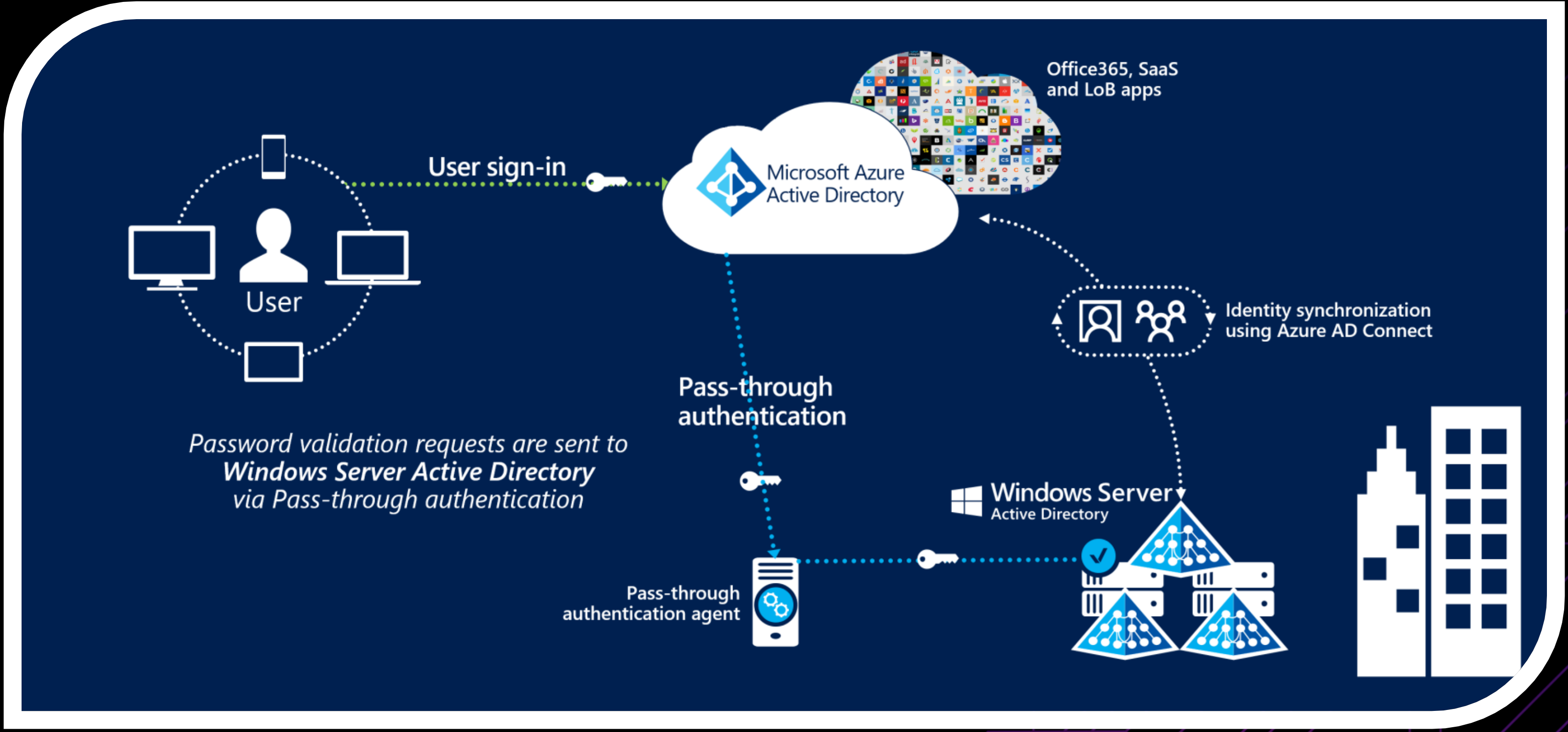
Password Hash Synchronization

- Azure AD Connect
- On-prem service synchronizes hashed user credentials to Azure
- User can authenticate directly to Azure services like O365 with their internal domain credential



Pass-Through Authentication

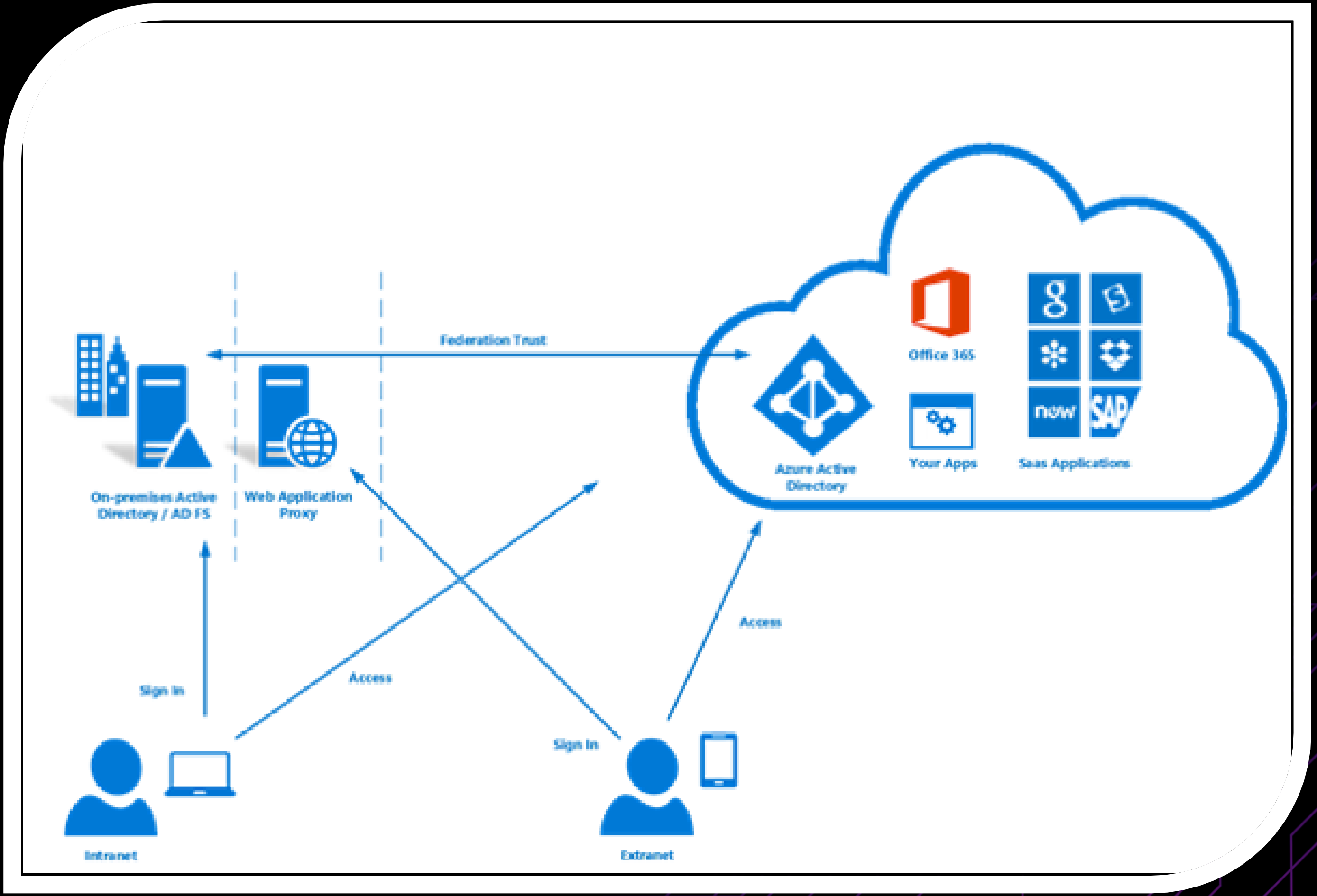
- Credentials stored only on-prem
- On-prem agent validates authentication requests to Azure AD
- Allows SSO to other Azure apps without creds stored in cloud


Office365, SaaS and LoB apps
User sign-in
Microsoft Azure Active Directory
User
Identity synchronization using Azure AD Connect
Pass-through authentication
Password validation requests are sent to Windows Server Active Directory via Pass-through authentication
Windows Server Active Directory
Pass-through authentication agent

© Black Hills Information Security
@BHInfoSecurity

Active Directory Federation Services

- Credentials stored only on-prem
- Federated trust is setup between Azure and on-prem AD to validate auth requests to the cloud
- For password attacks you would have to auth to the on-prem ADFS portal instead of Azure endpoints



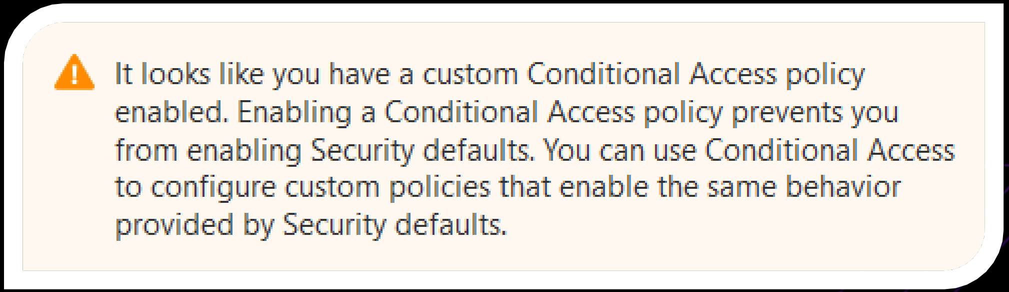
Conditional Access Policies & MFA

Microsoft MFA

- Microsoft 365 and Azure have built-in MFA options
- Free Microsoft accounts can use the MFA features
- Microsoft MFA verification options:
  - Microsoft Authenticator app
  - OAUTH Hardware token
  - SMS
  - Voice call

Security Defaults

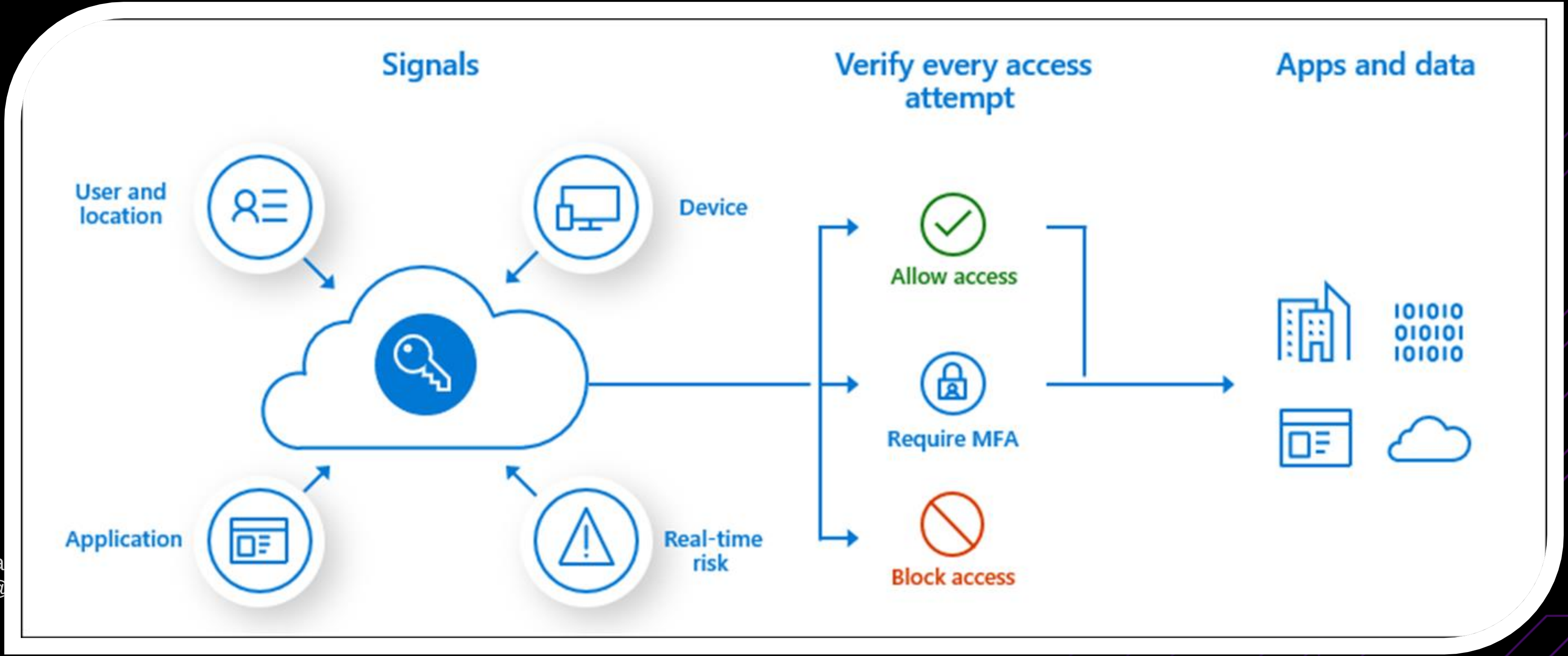
- Security Defaults is an Azure AD setting that helps protect accounts by:
  - Requires all users register for MFA
  - Blocks legacy auth protocols (EWS, IMAP, etc.)
  - Requires MFA during auth when necessary
  - Protects privileged activities like access to Azure portal
- These are great settings to have but sometimes more granular options are necessary.
- Conditional Access Policies are more advanced, but Security Defaults must be disabled to use them.



⚠ It looks like you have a custom Conditional Access policy enabled. Enabling a Conditional Access policy prevents you from enabling Security defaults. You can use Conditional Access to configure custom policies that enable the same behavior provided by Security defaults.

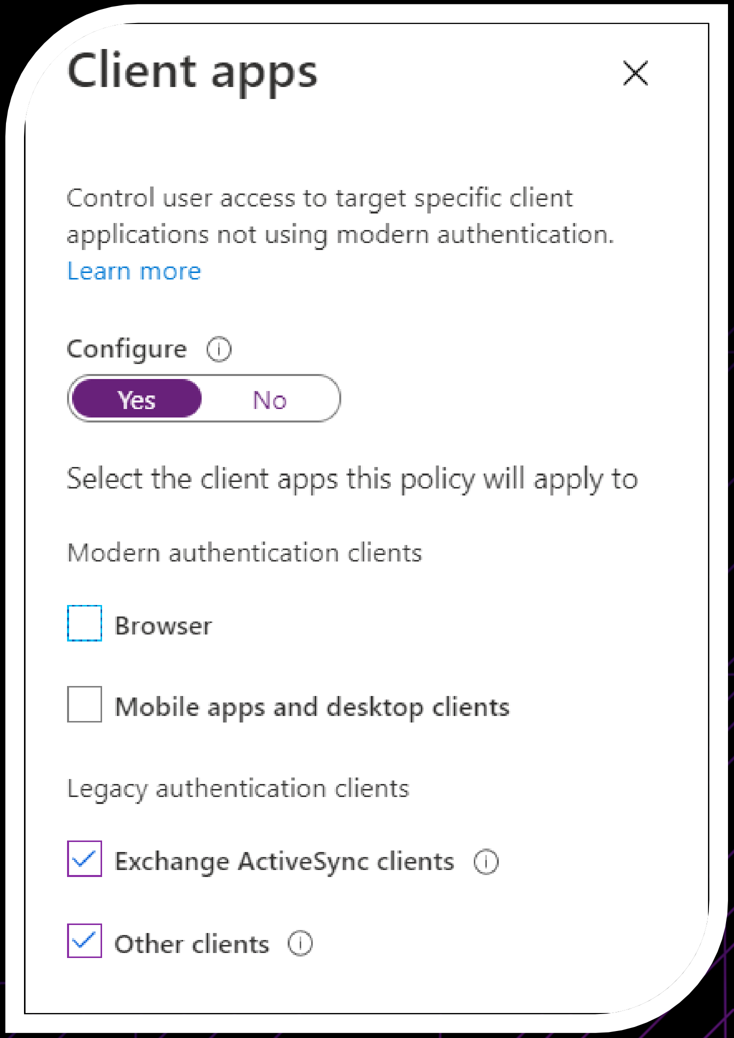
Conditional Access Policies

- Fine-grained controls for access to resources and when/where MFA is applied
- Can be built around different scenarios such as:
  - The user, location they are coming from, device they are using, their "real-time risk" level, and more



Legacy Auth

- Legacy Authentication – SMTP, IMAP, EAS, EWS, POP3, etc.
- Sometimes employees need access to legacy portals (ex. Outlook for Mac)
- These can be completely blocked with conditional access policies
- Note that Exchange ActiveSync has its own checkbox
- Legacy auth End of Life pushed back to 2nd half of 2021



Client apps

Control user access to target specific client applications not using modern authentication. Learn more

Configure: [✓ Yes] No

Select the client apps this policy will apply to

Modern authentication clients

□ Browser

□ Mobile apps and desktop clients

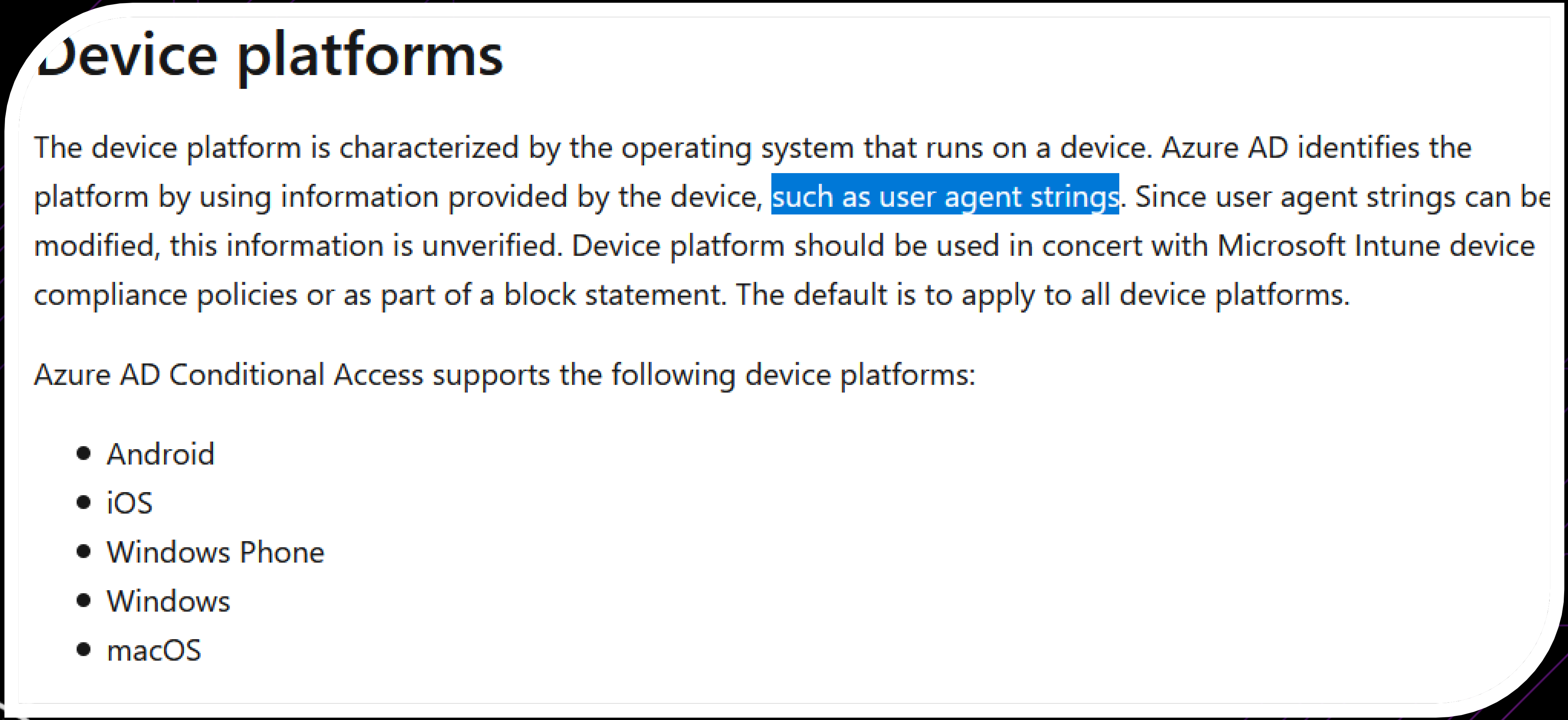
Legacy authentication clients

☑ Exchange ActiveSync clients

☑ Other clients

© Black Hills Information Security
@BHInfoSecurity

Device Platforms



Device platforms

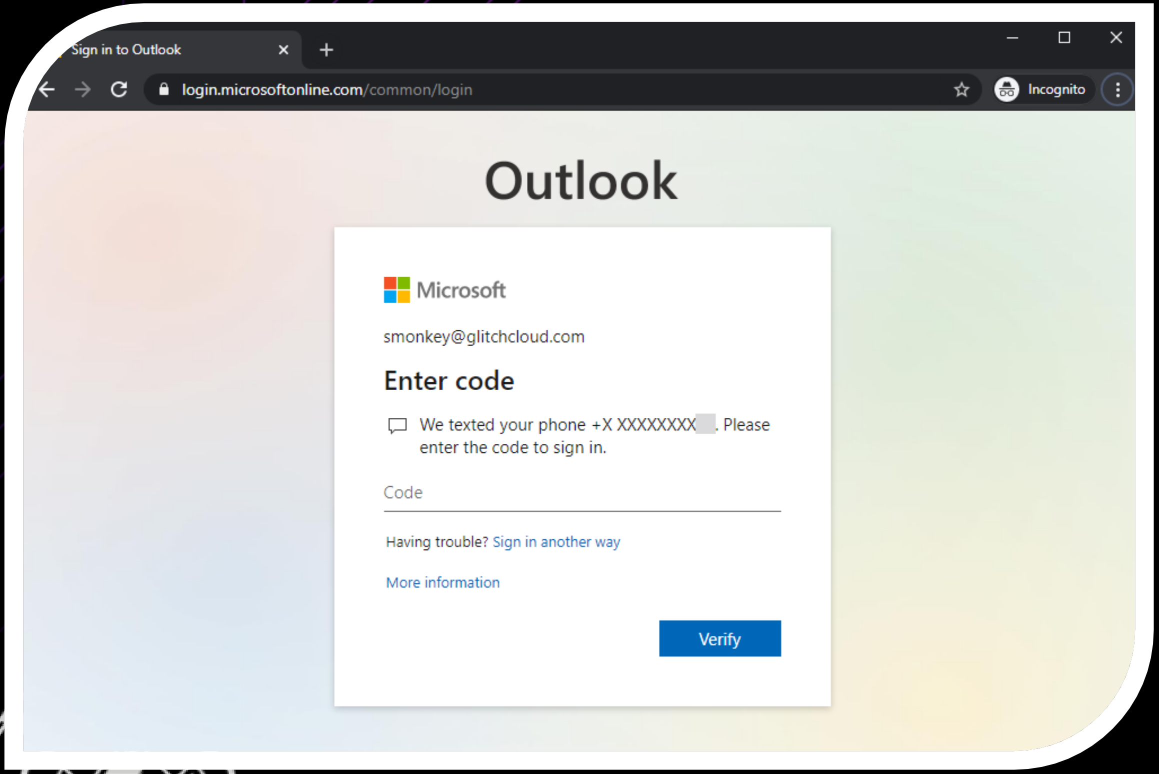
The device platform is characterized by the operating system that runs on a device. Azure AD identifies the platform by using information provided by the device, such as user agent strings. Since user agent strings can be modified, this information is unverified. Device platform should be used in concert with Microsoft Intune device compliance policies or as part of a block statement. The default is to apply to all device platforms.

Azure AD Conditional Access supports the following device platforms:

- Android
- iOS
- Windows Phone
- Windows
- macOS

Device Platforms

- Authentication without a mobile user agent and with



Outlook

Microsoft

smonkey@glitchcloud.com

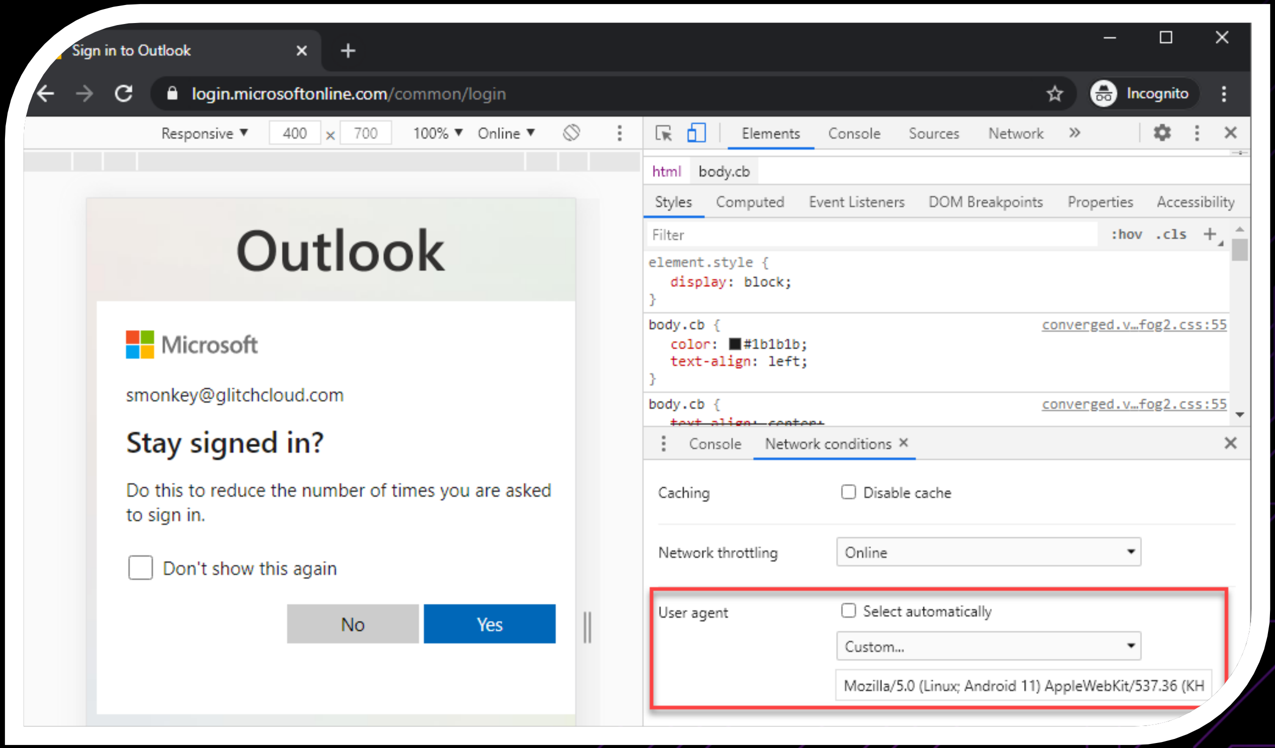
Enter code

We texted your phone +X XXXXXXX[redacted]. Please enter the code to sign in.

Code

Having trouble? Sign in another way

More information



Outlook

Microsoft

smonkey@glitchcloud.com

Stay signed in?

Do this to reduce the number of times you are asked to sign in.

☐ Don't show this again

```
element.style {
  display: block;
}
body.cb {                                   converged.v…fog2.css:55
  color: ■#1b1b1b;
  text-align: left;
}
body.cb {                                   converged.v…fog2.css:55
  ~~text-align: center;~~
```

Caching ☐ Disable cache

Network throttling: Online

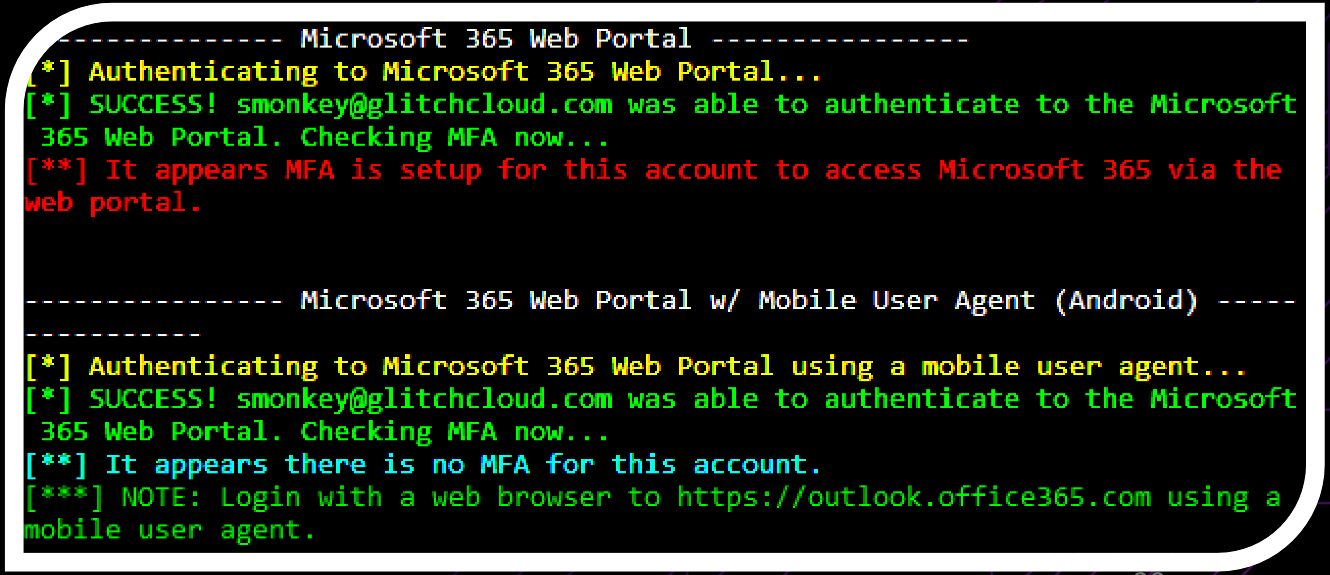
User agent ☐ Select automatically

Custom...

Mozilla/5.0 (Linux; Android 11) AppleWebKit/537.36 (KH

MFASweep

- Tool to help find inconsistencies in Microsoft MFA deployments
  - Microsoft Graph API
  - Azure Service Management API
  - Microsoft 365 Exchange Web Services
  - Microsoft 365 Web Portal
  - Microsoft 365 Web Portal Using a Mobile User Agent
  - Microsoft 365 Active Sync
  - ADFS
- https://github.com/dafthack/MFASweep



```
---------------- Microsoft 365 Web Portal ----------------
[*] Authenticating to Microsoft 365 Web Portal...
[*] SUCCESS! smonkey@glitchcloud.com was able to authenticate to the Microsoft
 365 Web Portal. Checking MFA now...
[**] It appears MFA is setup for this account to access Microsoft 365 via the
web portal.


---------------- Microsoft 365 Web Portal w/ Mobile User Agent (Android) -----
-----------
[*] Authenticating to Microsoft 365 Web Portal using a mobile user agent...
[*] SUCCESS! smonkey@glitchcloud.com was able to authenticate to the Microsoft
 365 Web Portal. Checking MFA now...
[**] It appears there is no MFA for this account.
[***] NOTE: Login with a web browser to https://outlook.office365.com using a
mobile user agent.
```

MFASweep

- To run MFASweep all you need is a set of credentials you want to test
- **WARNING: This script attempts to log in to the provided account SIX (6) different times (7 if you include ADFS). If you enter an incorrect password, this may lock the account out.**
- Import MFASweep into a PowerShell session

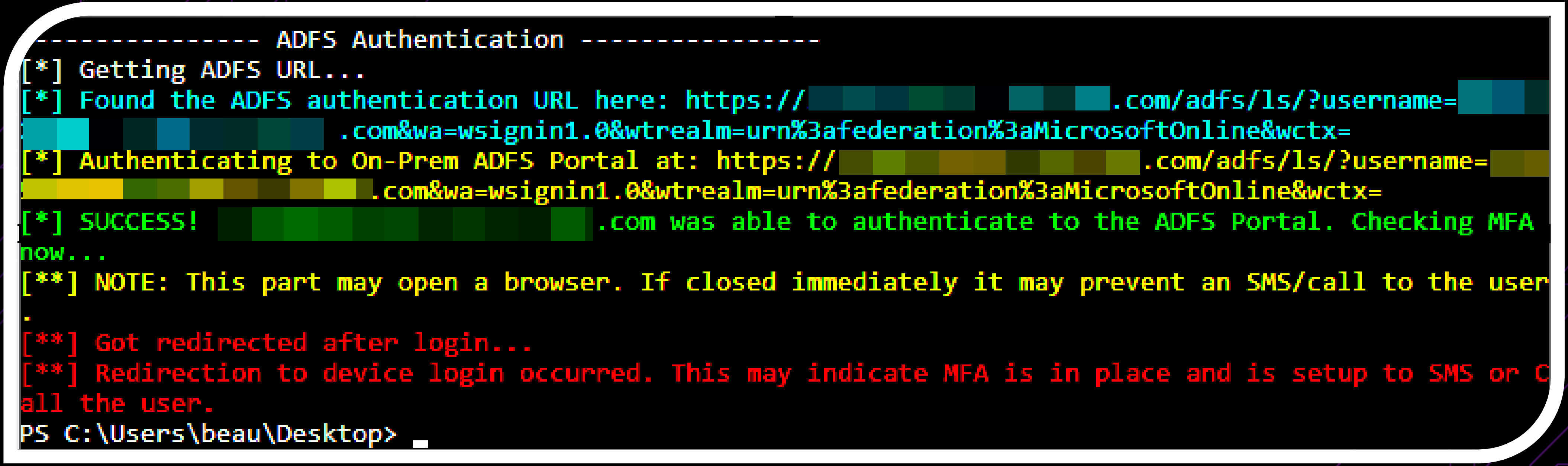
```
Import-Module MFASweep.ps1
```

- Run the Invoke-MFASweep module with the credentials

```
Invoke-MFASweep -Username targetuser@targetdomain.com -Password
Winter2020
```

MFASweep

- Can also check ADFS



```
---------------- ADFS Authentication ----------------
[*] Getting ADFS URL...
[*] Found the ADFS authentication URL here: https://[redacted].com/adfs/ls/?username=[redacted]
[redacted] .com&wa=wsignin1.0&wtrealm=urn%3afederation%3aMicrosoftOnline&wctx=
[*] Authenticating to On-Prem ADFS Portal at: https://[redacted].com/adfs/ls/?username=[redacted]
[redacted].com&wa=wsignin1.0&wtrealm=urn%3afederation%3aMicrosoftOnline&wctx=
[*] SUCCESS! [redacted].com was able to authenticate to the ADFS Portal. Checking MFA
now...
[**] NOTE: This part may open a browser. If closed immediately it may prevent an SMS/call to the user
.
[**] Got redirected after login...
[**] Redirection to device login occurred. This may indicate MFA is in place and is setup to SMS or C
all the user.
PS C:\Users\beau\Desktop> _
```

- For more information check out the blog post here: https://www.blackhillsinfosec.com/exploiting-mfa-inconsistencies-on-microsoft-services/

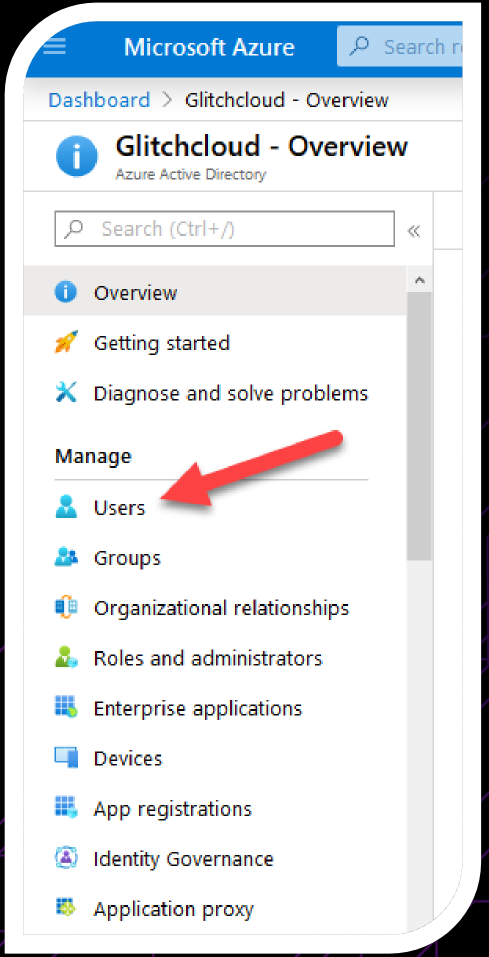
Post Compromise

Post-Compromise Recon

- Who do we have access as?
- What roles do we have?
- Is MFA enabled?
- What can we access (webapps, storage, etc.?)
- Who are the admins?
- How are we going to escalate to admin?
- Any security protections in place (ATP, GuardDuty, etc.)?

Azure Portal

- Standard users can access Azure domain information and isn’t usually locked down
- Authenticated users can go to portal.azure.com and click Azure Active Directory
- O365 Global Address List has this info as well
- Even if portal is locked down PowerShell cmdlets will still likely work
- There is a company-wide setting that locks down the entire org from viewing Azure info via cmd line:
  - Set-MsolCompanySettings –UsersPermissionToReadOtherUsersEnabled $false



Command Line Access

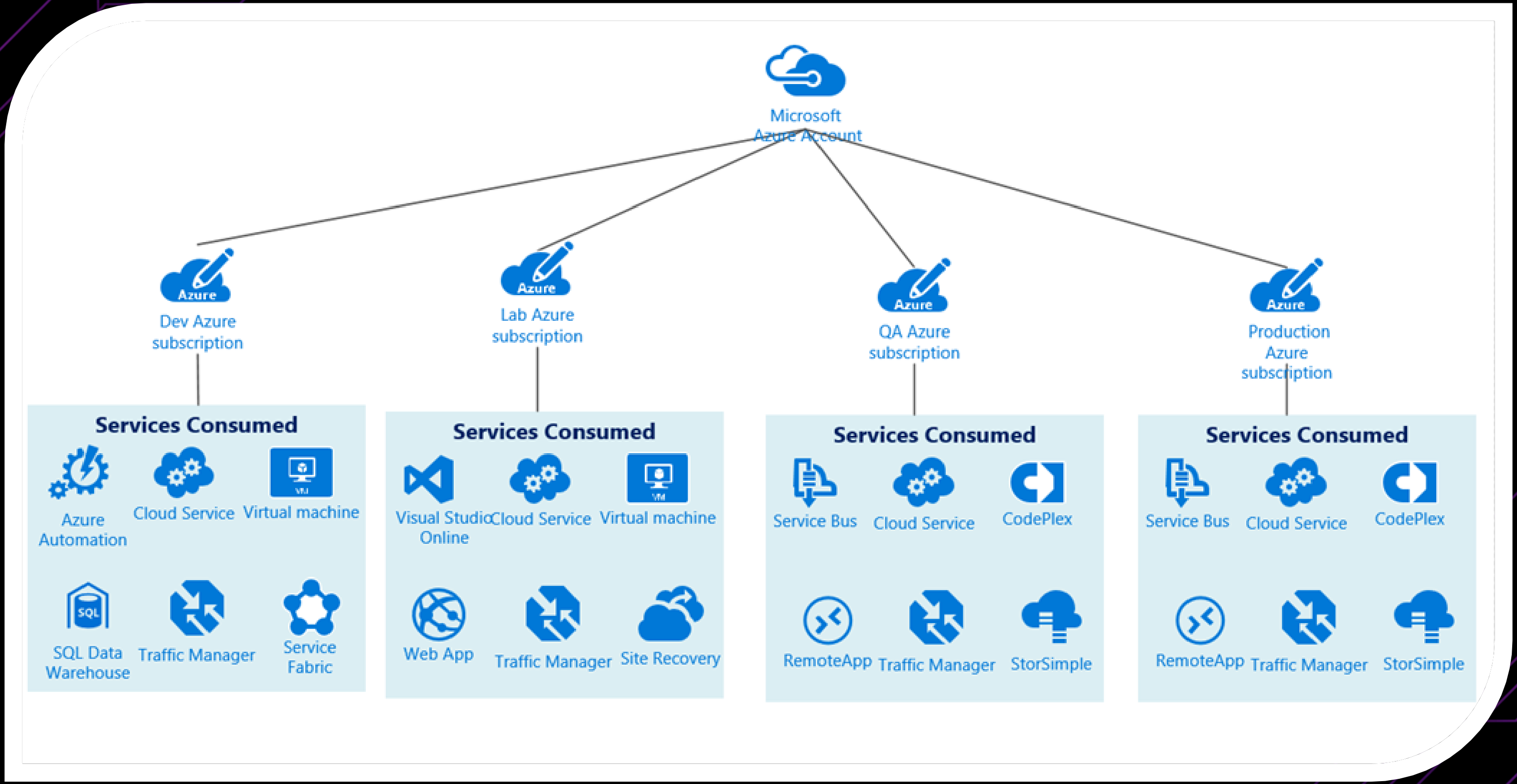
- PowerShell Modules
  - Az
  - AzureAD & MSOnline
- Azure Cross-platform CLI Tools (az cli)
  - Linux and Windows clients
- CloudPentestCheatsheets
  - https://github.com/dafthack/CloudPentestCheatsheets

Azure Subscription Hierachy

© Black Hills Information Security
@BHInfoSecurity

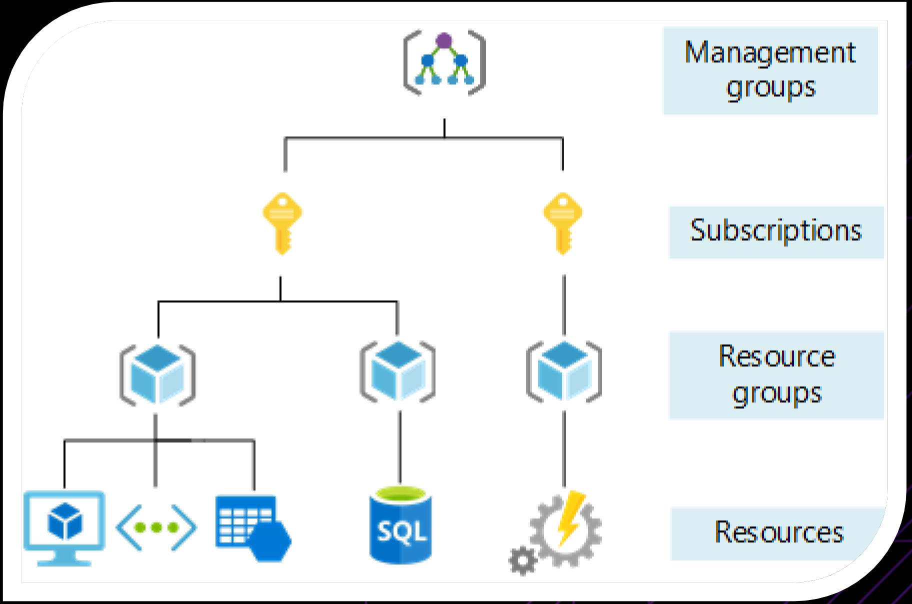
Subscriptions

- Organizations can have multiple subscriptions



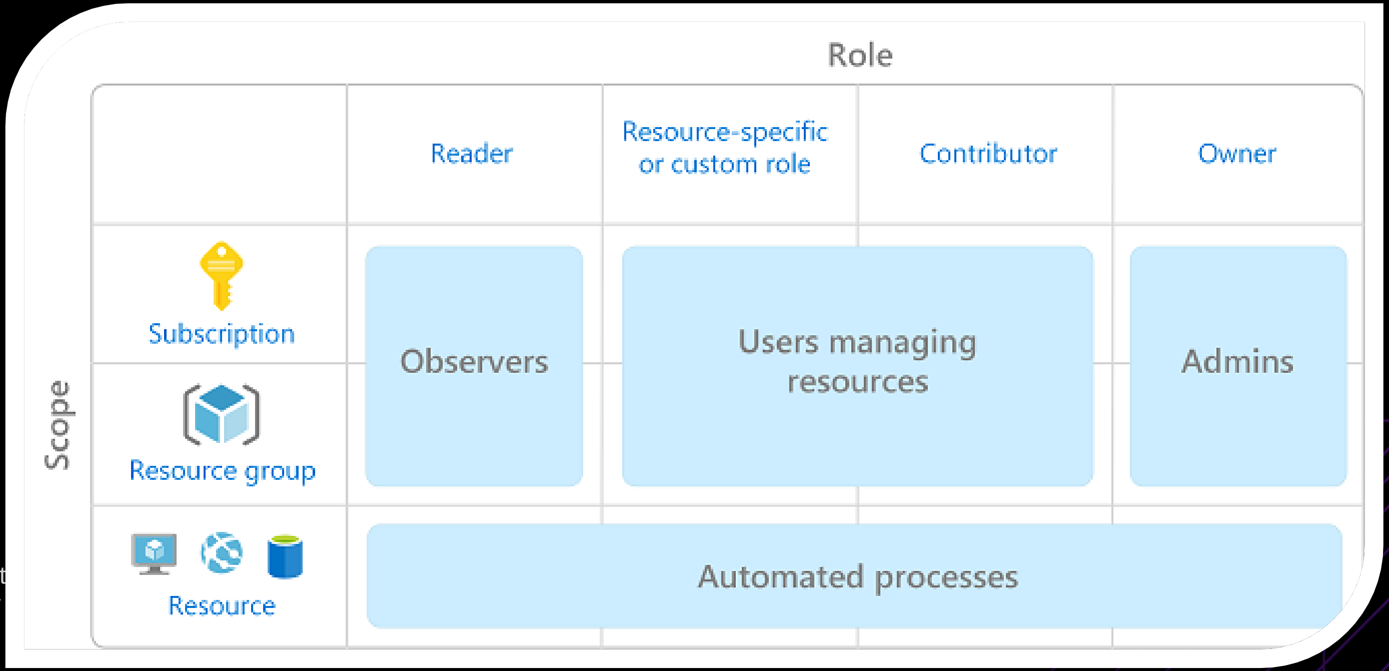
Subscriptions

- A good first step is to determine what subscription you are in
- The subscription name is usually informative
- It might have “Prod”, or “Dev” in the title
- Multiple subscriptions can be under the same Azure AD directory (tenant)
- Each subscription can have multiple resource groups



Roles

- Built-In Azure Subscription Roles
  - Owner (full control over resource)
  - Contributor (All rights except the ability to change permissions)
  - Reader (can only read attributes)
  - User Access Administrator (manage user access to Azure resources)



Role

Scope

	Reader	Resource-specific or custom role	Contributor	Owner
Subscription	Observers	Users managing resources		Admins
Resource group				
Resource	Automated processes			

Resource-Specific Issues

Serverless Environment Variables

- Azure Functions – Serverless apps in Azure
  - Secrets should be called from Key Vaults
  - Sometimes plaintext values get added as environment vars or within source code as connection strings
  - Reader level access to Functions allows viewing

Instance Metadata Service

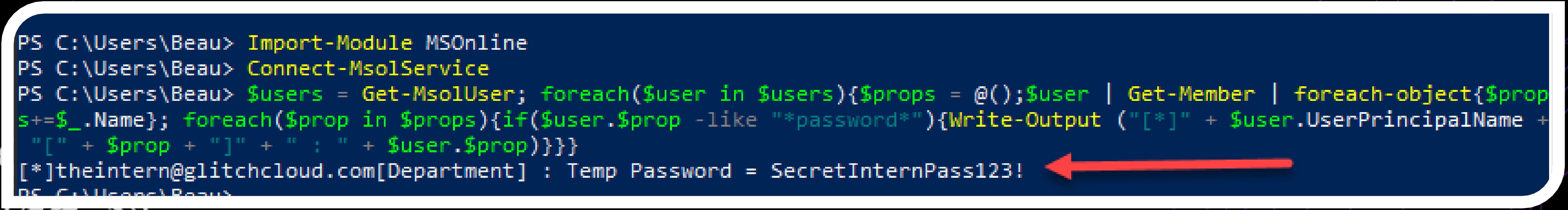
- Cloud servers need a way to orient themselves because of how dynamic they are
- A “Metadata” endpoint was created and hosted on a non-routable IP address at 169.254.169.254
- Can contain access/secret keys to AWS and IAM credentials
- This *should* only be reachable from the localhost
- Server compromise or SSRF vulnerabilities might allow remote attackers to reach it

http://169.254.169.254/metadata/identity/oauth2/token?api-version=2018-02-01&resource=https://management.azure.com/

Azure AD User Attributes

- User attributes and sensitive information
- Very often find credentials in description or comment fields
- Use this one-liner to search every Azure AD user field for passwords

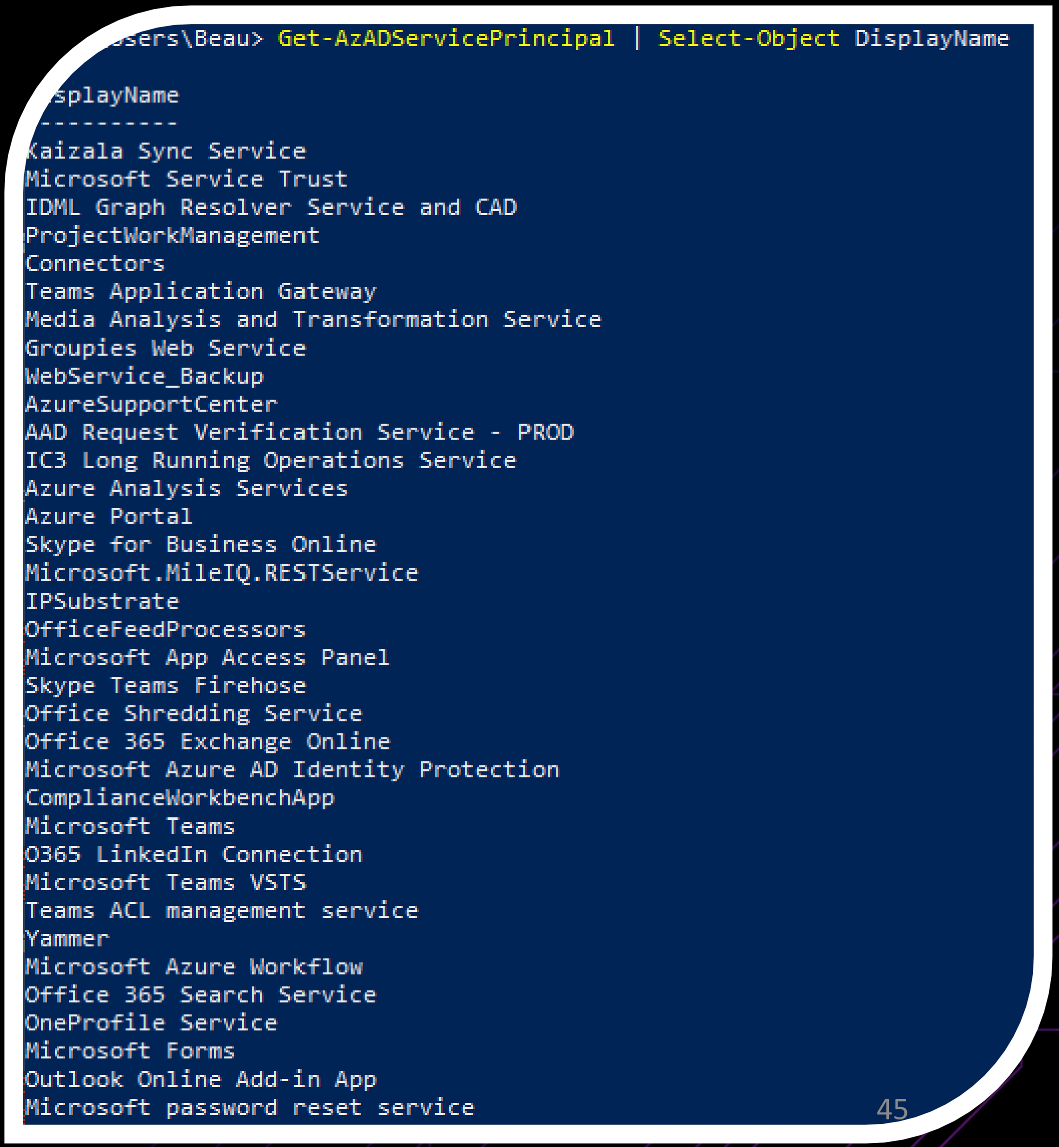
```
PS> $users = Get-MsolUser; foreach($user in $users){$props =
@();$user | Get-Member | foreach-object{$props+=$_.Name};
foreach($prop in $props){if($user.$prop -like "*password*"){Write-
Output ("[*]" + $user.UserPrincipalName + "[" + $prop + "]" + " : " +
$user.$prop)}}}
```



```
PS C:\Users\Beau> Import-Module MSOnline
PS C:\Users\Beau> Connect-MsolService
PS C:\Users\Beau> $users = Get-MsolUser; foreach($user in $users){$props = @();$user | Get-Member | foreach-object{$prop
s+=$_.Name}; foreach($prop in $props){if($user.$prop -like "*password*"){Write-Output ("[*]" + $user.UserPrincipalName +
 "[" + $prop + "]" + " : " + $user.$prop)}}}
[*]theintern@glitchcloud.com[Department] : Temp Password = SecretInternPass123!
PS C:\Users\Beau>
```

Service Principal Hijacking

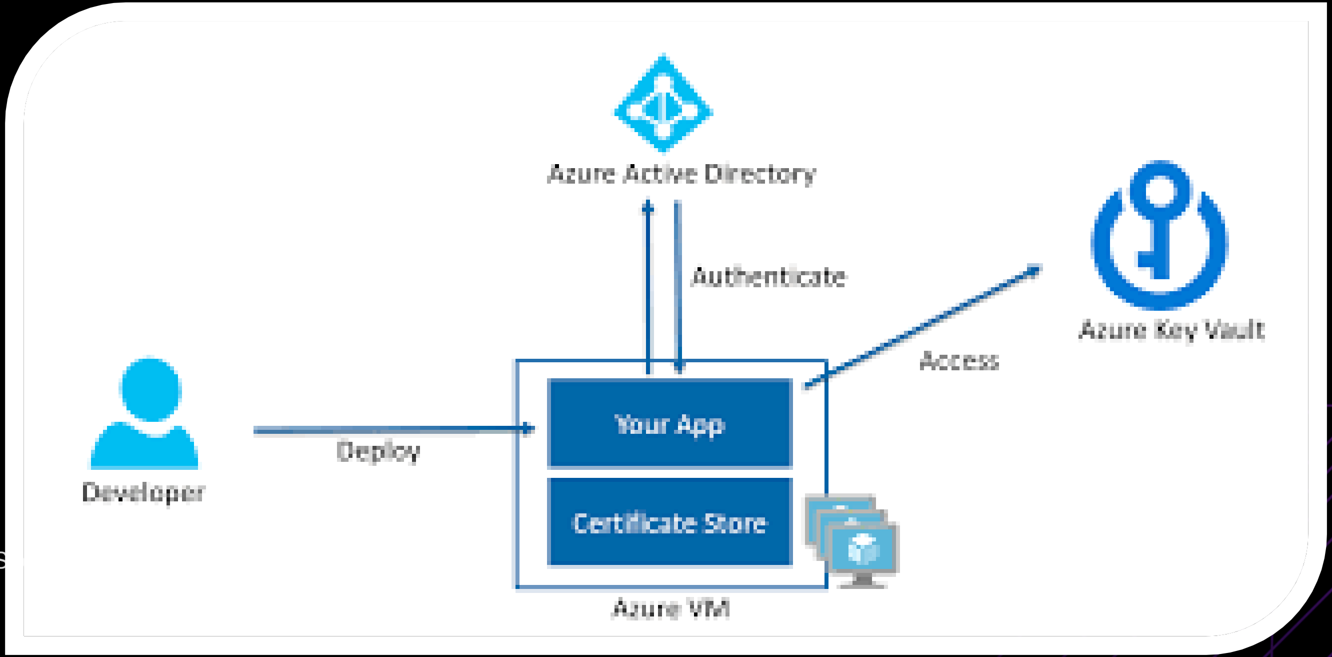
- There are over 200 default service principals in an O365 tenant
- None of them are listed in the Azure GUI portal
- They all have varying levels of permissions through Microsoft Graph
- An "Application Administrator" can change passwords or certificates for service principals… even the default ones



```
sers\Beau> Get-AzADServicePrincipal | Select-Object DisplayName

splayName
----------
Kaizala Sync Service
Microsoft Service Trust
IDML Graph Resolver Service and CAD
ProjectWorkManagement
Connectors
Teams Application Gateway
Media Analysis and Transformation Service
Groupies Web Service
WebService_Backup
AzureSupportCenter
AAD Request Verification Service - PROD
IC3 Long Running Operations Service
Azure Analysis Services
Azure Portal
Skype for Business Online
Microsoft.MileIQ.RESTService
IPSubstrate
OfficeFeedProcessors
Microsoft App Access Panel
Skype Teams Firehose
Office Shredding Service
Office 365 Exchange Online
Microsoft Azure AD Identity Protection
ComplianceWorkbenchApp
Microsoft Teams
O365 LinkedIn Connection
Microsoft Teams VSTS
Teams ACL management service
Yammer
Microsoft Azure Workflow
Office 365 Search Service
OneProfile Service
Microsoft Forms
Outlook Online Add-in App
Microsoft password reset service
```

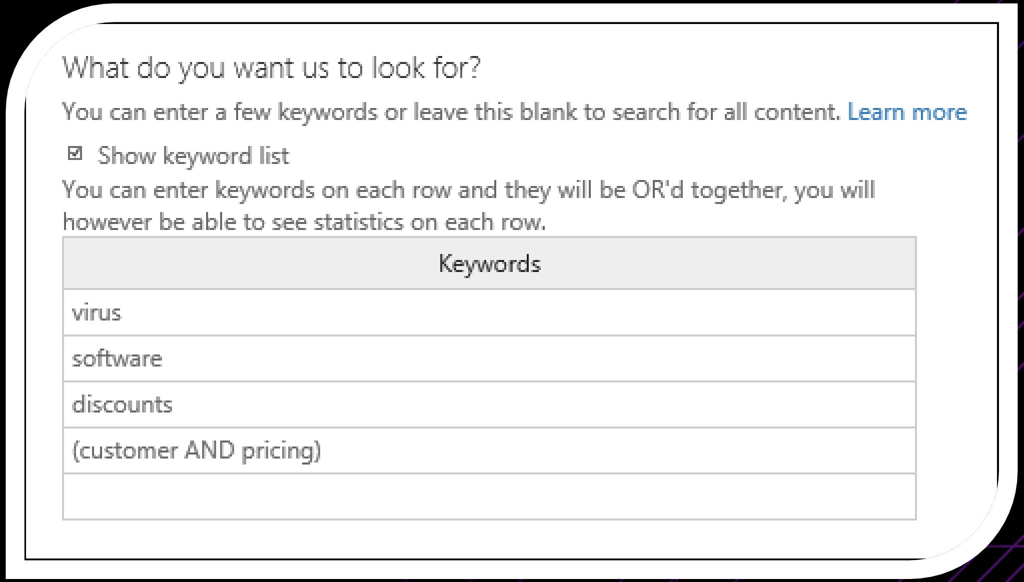
Key Vaults

- Azure Key Vault
  - Vault for storing passwords and other secrets
  - Other cloud apps and services can use these
  - Easily store and manage SSL/TLS certs
  - By default only the owner of the key vault can access the keys
  - Contributors over key vault resources can give themselves access



Microsoft 365 Compliance Search

- Microsoft 365 Compliance Search
- https://protection.office.com
- Must be a member of "eDiscovery Manager" role group in Security & Compliance Center (Administrator, compliance officer, or eDiscover manager)
- Search and report across all Microsoft 365 services
  - Exchange Email
  - Skype for Business
  - Teams messages
  - SharePoint Sites
  - OneDrive Accounts
  - And more…



What do you want us to look for?

You can enter a few keywords or leave this blank to search for all content. Learn more

☑ Show keyword list

You can enter keywords on each row and they will be OR'd together, you will however be able to see statistics on each row.

Keywords
virus
software
discounts
(customer AND pricing)

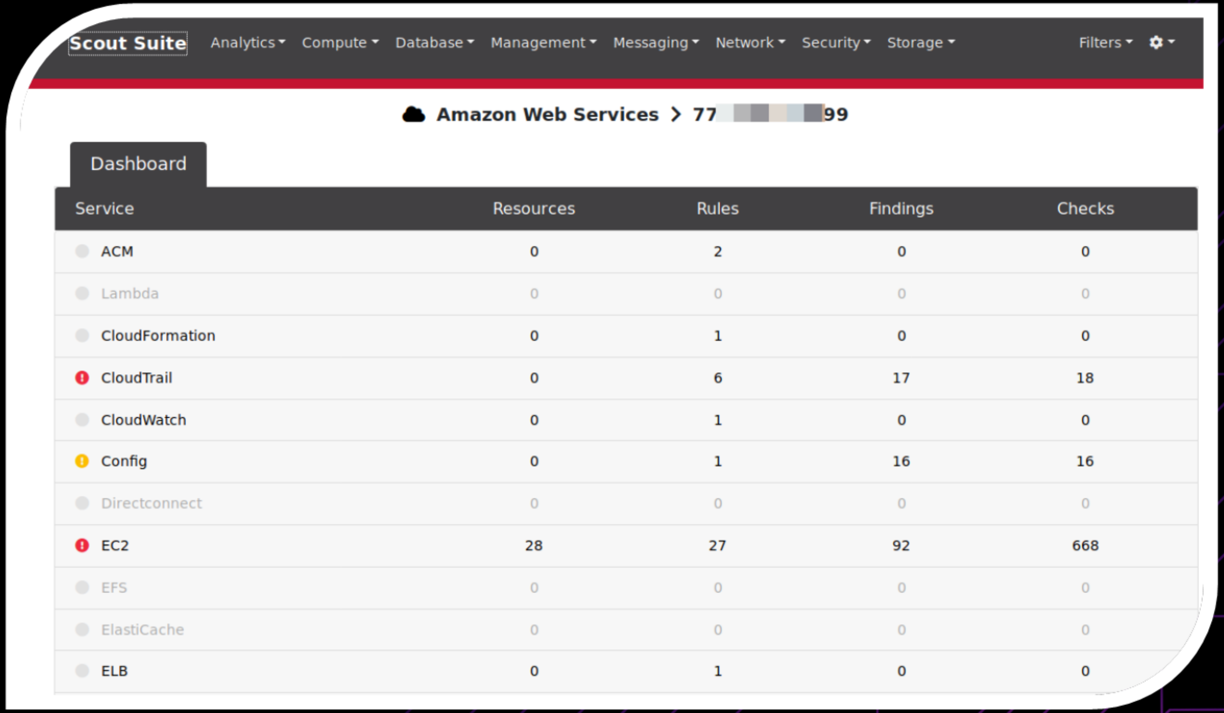
Leveraging Scanning Tools

Leveraging Scanning Tools

- How can automation help?
- Manual inspection of cloud resources is likely a good starting point to be less noisy but scanning can help expedite vulnerability discovery
- Quickly assess cloud environments for common security issues
  - IAM permissions
  - Public accessibility of resources
  - VM/Instance storage encryption
  - Network ingress/egress rules
  - Serverless
  - VM metadata
  - …and more

Scanning with ScoutSuite

- ScoutSuite by NCC Group - Multi-Cloud Auditing Tool
  - https://github.com/nccgroup/ScoutSuite
- Support for the following cloud providers:
  - Amazon Web Services
  - Microsoft Azure
  - Google Cloud Platform
  - Alibaba Cloud (alpha)
  - Oracle Cloud Infrastructure (alpha)



Amazon Web Services > 77[redacted]99

Service	Resources	Rules	Findings	Checks
● ACM	0	2	0	0
● Lambda	0	0	0	0
● CloudFormation	0	1	0	0
❗ CloudTrail	0	6	17	18
● CloudWatch	0	1	0	0
❗ Config	0	1	16	16
● Directconnect	0	0	0	0
❗ EC2	28	27	92	668
● EFS	0	0	0	0
● ElastiCache	0	0	0	0
● ELB	0	1	0	0

Tools

- Additional tools to help automate post-compromise
- ROADTools
  - https://github.com/dirkjanm/ROADtools
- PowerZure
  - https://github.com/hausec/PowerZure
- MicroBurst
  - https://github.com/NetSPI/MicroBurst
- Stormspotter
  - https://github.com/Azure/Stormspotter
- AzureHound
  - https://github.com/BloodHoundAD/AzureHound

Key Takeaways

1. Reconnaissance is key to understanding cloud asset usage
2. Cloud attack surface enables multiple ways to gain access
3. Configuration of cloud resources is a wild west and changes daily
4. Key methods for gaining a foothold include:
    1. Key disclosure in repos
    2. Password attacks
    3. Phishing
    4. Remote code execution
5. Situational awareness will help drive decisions post-compromise

The End

- Follow me on Twitter
  - Beau Bullock - @dafthack
- Breaching the Cloud Training
  - https://wildwesthackinfest.com/training/breaching-the-cloud-beau-bullock/
- Black Hills Information Security
  - https://www.blackhillsinfosec.com
  - @BHInfoSecurity