Public Session notes (raw presentation) - FAUG goes Full Day Learning (Azure)

Joosua Santasalo - Senior Principal Security Researcher at Secureworks

Whoami?

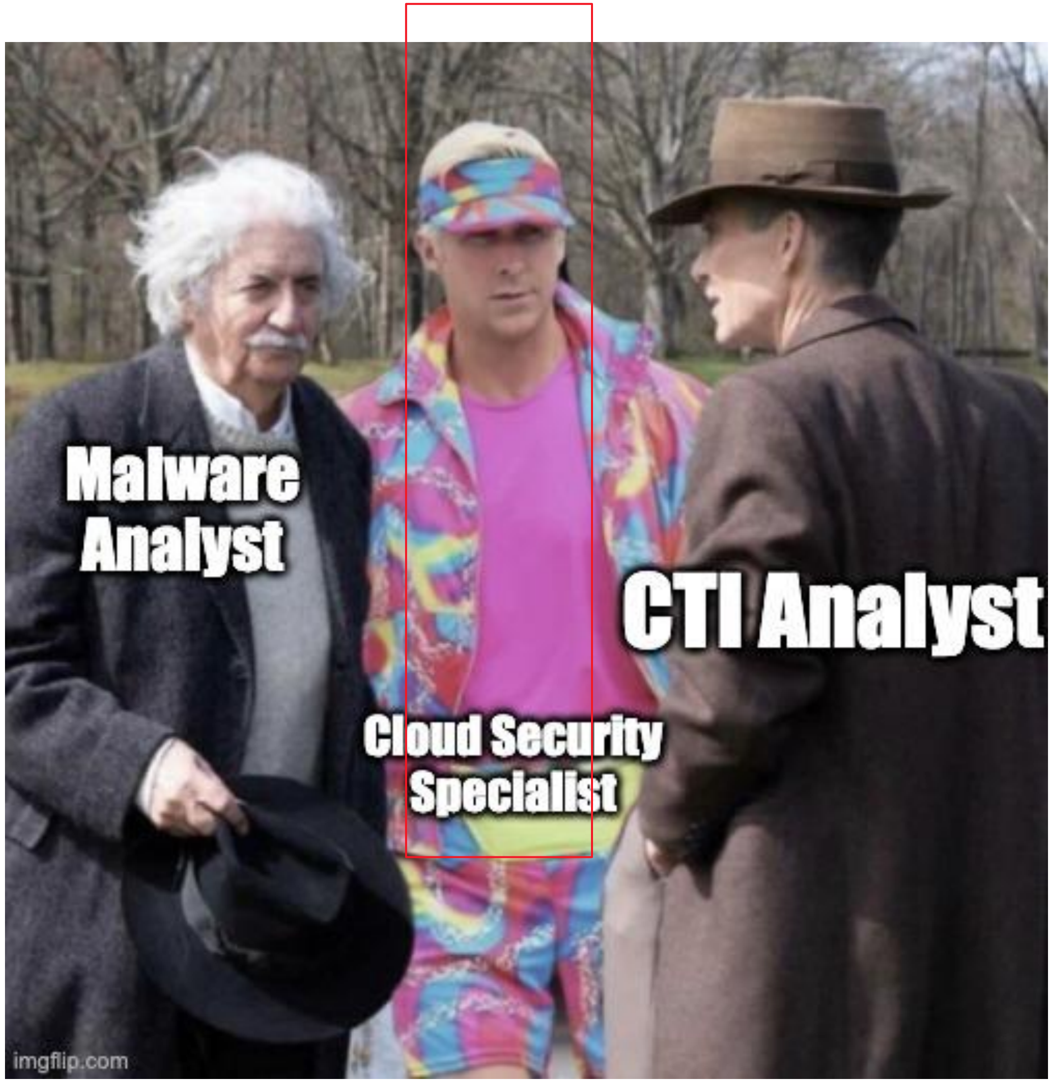
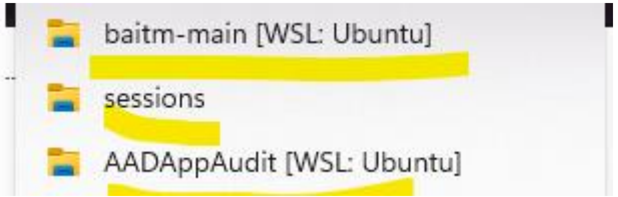


Image Credits: Florian Roth on X: ""DFIR Team Support" (2024, colorized)
https://t.co/on05AfnVfB" / X (twitter.com)

Agenda (Just demos and random folk talking at stage)

Demos:
Do fresh login,
add IP’s
Reboot WSL wsl –shutdown


baitm-main [WSL: Ubuntu]
sessions
AADAppAudit [WSL: Ubuntu]

Secureworks®

Base facts: two kids, wife and dog. Lives in Helsinki

Work: 2017-2022 Nixu -> **2022 -> Secureworks**

Links: LI, X, Securecloud.blog, https://github.com/jsa2

Creds: MS MVP (Azure) 2020 -> MSRC Most Valued Researcher Top 100 (80#) 2023

Doing: Security research - Security tool development offensive and blue-team tooling. Tooling Mostly written in Node.JS

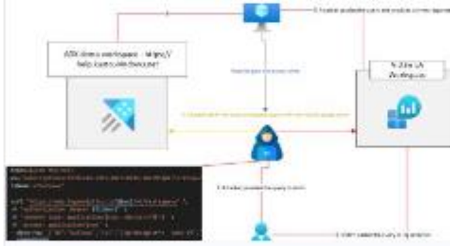
Published research 2022-2023 (stay tuned for 2024)

Power Platform Privilege Escalation

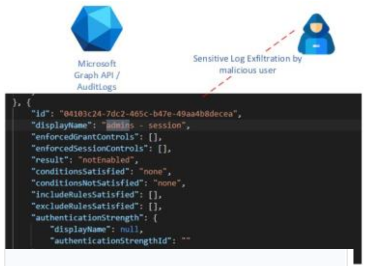
Azure Active Directory Flaw Allowed SAML Persistence

Spoofing Microsoft Entra ID Verified Publisher Status

Admin Isolation on Shared Clusters



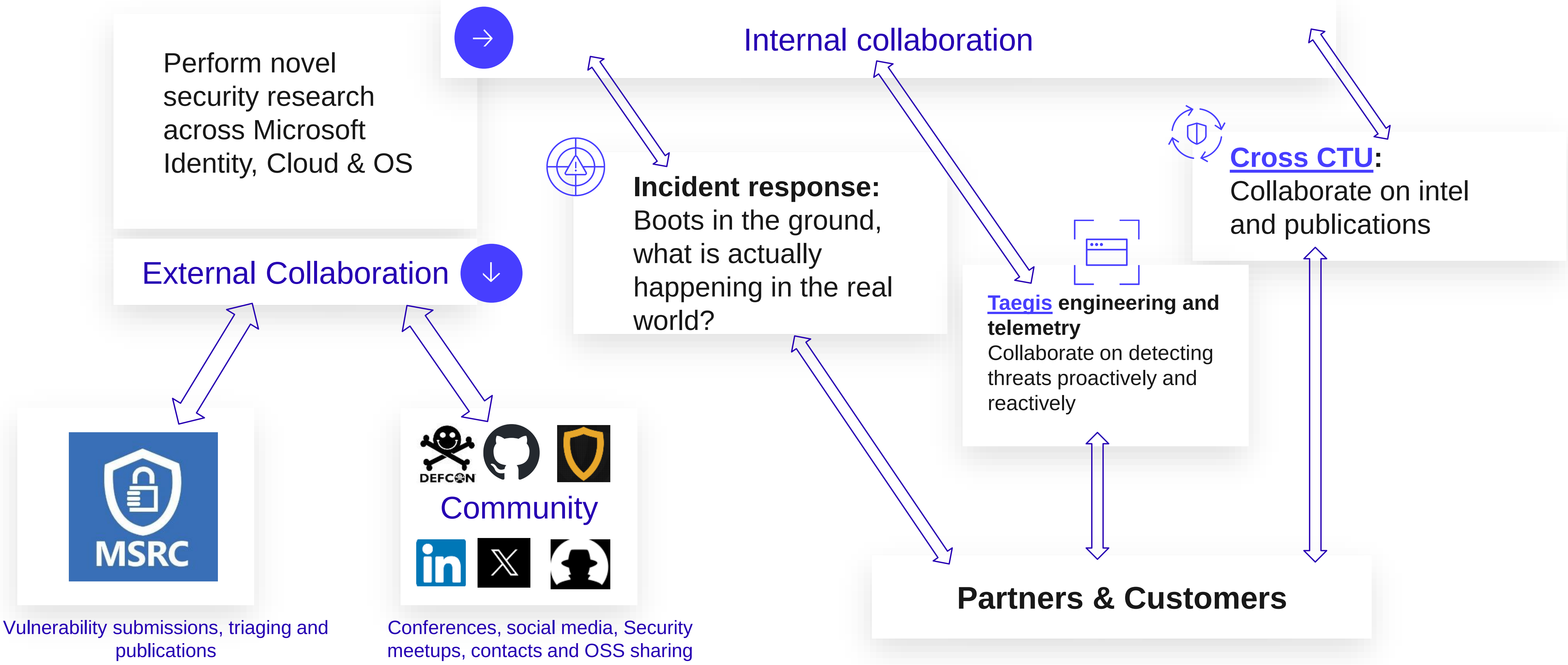
Azure Monitor – Malicious KQL Query



```
}, {
    "id": "04103c24-7dc2-465c-b47e-49aa4b8decea",
    "displayName": "admins - session",
    "enforcedGrantControls": [],
    "enforcedSessionControls": [],
    "result": "notEnabled",
    "conditionsSatisfied": "none",
    "conditionsNotSatisfied": "none",
    "includeRulesSatisfied": [],
    "excludeRulesSatisfied": [],
    "authenticationStrength": {
        "displayName": null,
        "authenticationStrengthId": ""
```

Microsoft Cloud Security Research – Public Disclosure –...

Stuff we do at Secureworks in researcher role



Perform novel security research across Microsoft Identity, Cloud & OS

Internal collaboration

External Collaboration

Incident response: Boots in the ground, what is actually happening in the real world?

Cross CTU: Collaborate on intel and publications

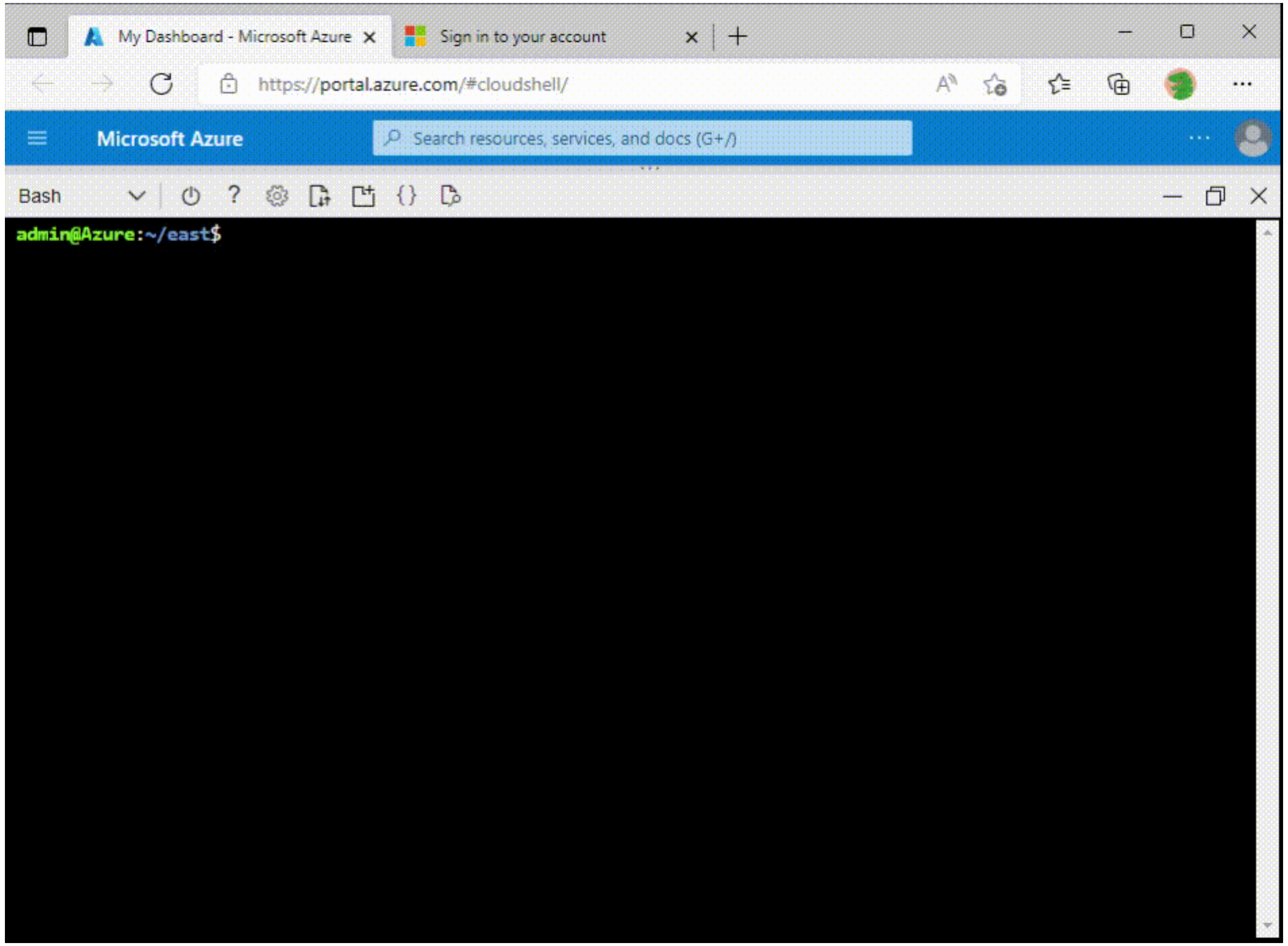
Taegis engineering and telemetry Collaborate on detecting threats proactively and reactively

Partners & Customers

Vulnerability submissions, triaging and publications

Conferences, social media, Security meetups, contacts and OSS sharing

Hope you are not tired of waiting stuff…

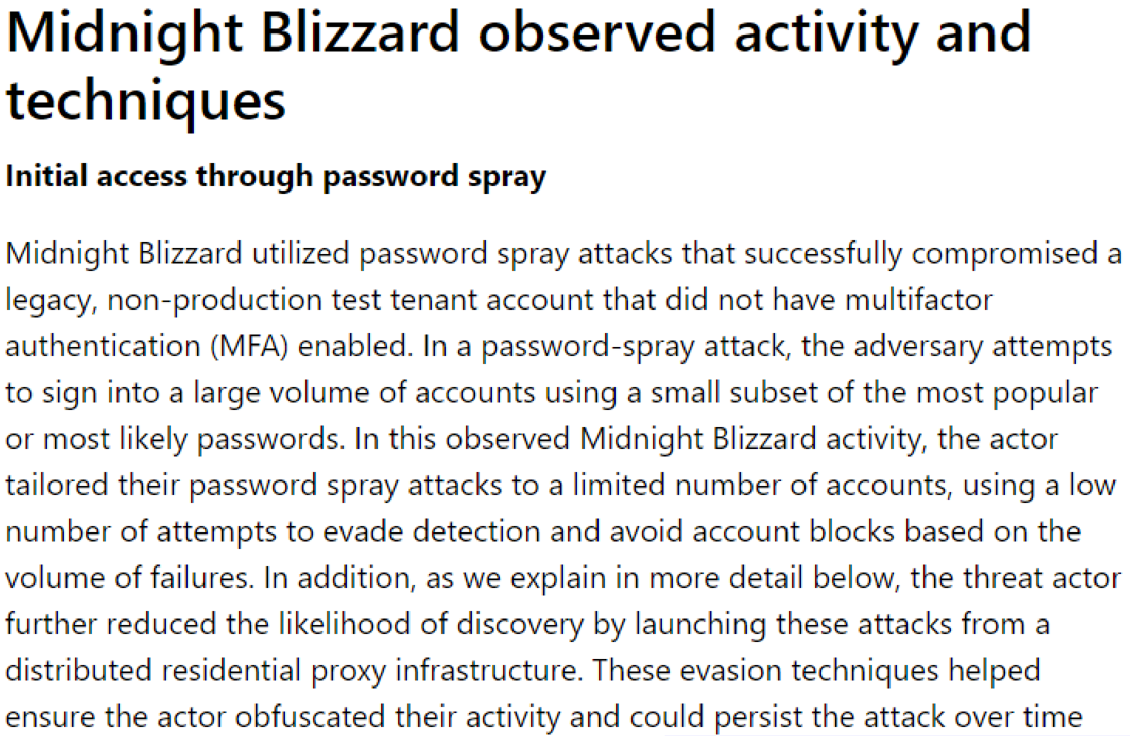

My Dashboard - Microsoft Azure
Sign in to your account
https://portal.azure.com/#cloudshell/
Microsoft Azure
Search resources, services, and docs (G+/)
Bash

```
admin@Azure:~/east$
```

Areas highlighted in Midnight Blizzard

Audit gaps in Conditional Access

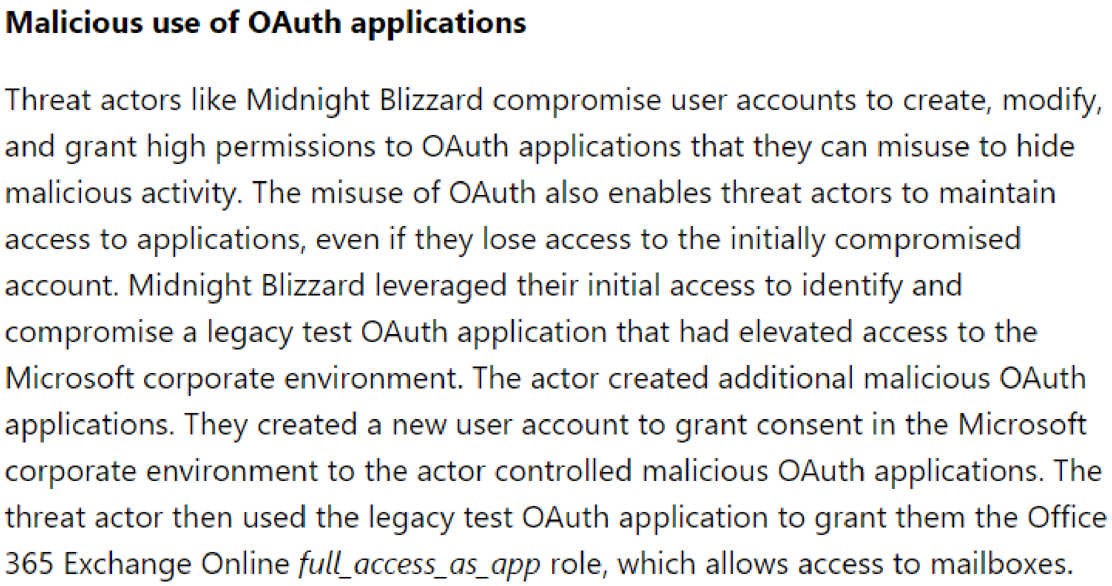
Audit Entra ID applications



Midnight Blizzard observed activity and techniques

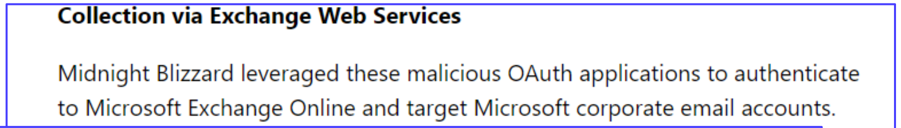
Initial access through password spray

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.



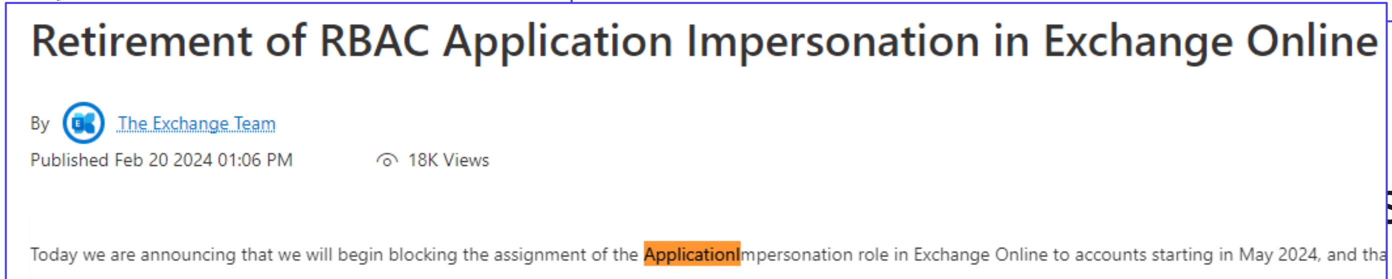
Malicious use of OAuth applications

Threat actors like Midnight Blizzard compromise user accounts to create, modify, and grant high permissions to OAuth applications that they can misuse to hide malicious activity. The misuse of OAuth also enables threat actors to maintain access to applications, even if they lose access to the initially compromised account. Midnight Blizzard leveraged their initial access to identify and compromise a legacy test OAuth application that had elevated access to the Microsoft corporate environment. The actor created additional malicious OAuth applications. They created a new user account to grant consent in the Microsoft corporate environment to the actor controlled malicious OAuth applications. The threat actor then used the legacy test OAuth application to grant them the Office 365 Exchange Online *full_access_as_app* role, which allows access to mailboxes.



Collection via Exchange Web Services

Midnight Blizzard leveraged these malicious OAuth applications to authenticate to Microsoft Exchange Online and target Microsoft corporate email accounts.



Retirement of RBAC Application Impersonation in Exchange Online

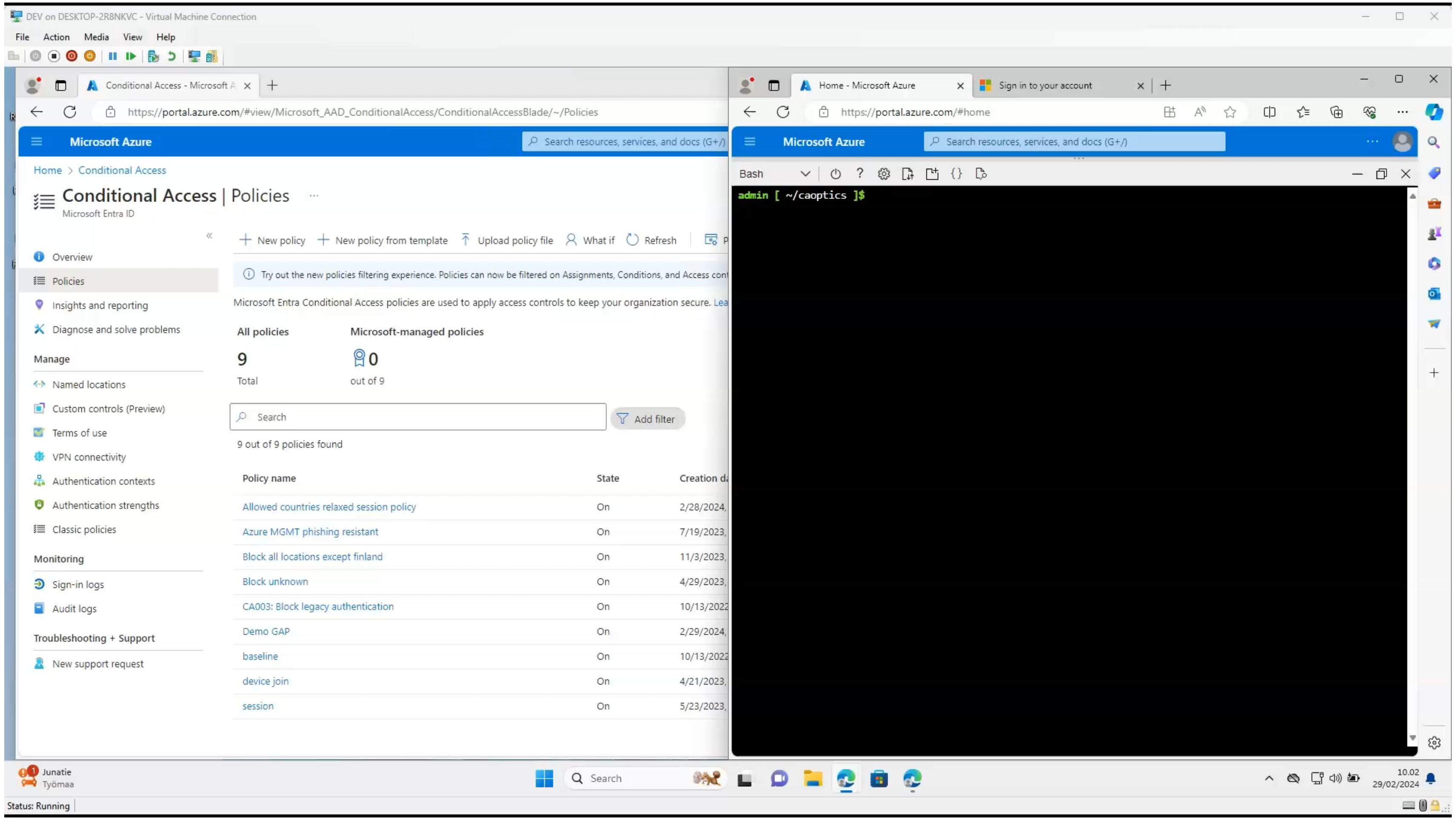
By The Exchange Team

Published Feb 20 2024 01:06 PM 18K Views

Today we are announcing that we will begin blocking the assignment of the ApplicationImpersonation role in Exchange Online to accounts starting in May 2024, and tha

Areas highlighted in Midnight Blizzard

Audit gaps in Conditional Access



Conditional Access | Policies

Try out the new policies filtering experience. Policies can now be filtered on Assignments, Conditions, and Access cont

Microsoft Entra Conditional Access policies are used to apply access controls to keep your organization secure. Lea

All policies	Microsoft-managed policies
9	0
Total	out of 9

9 out of 9 policies found

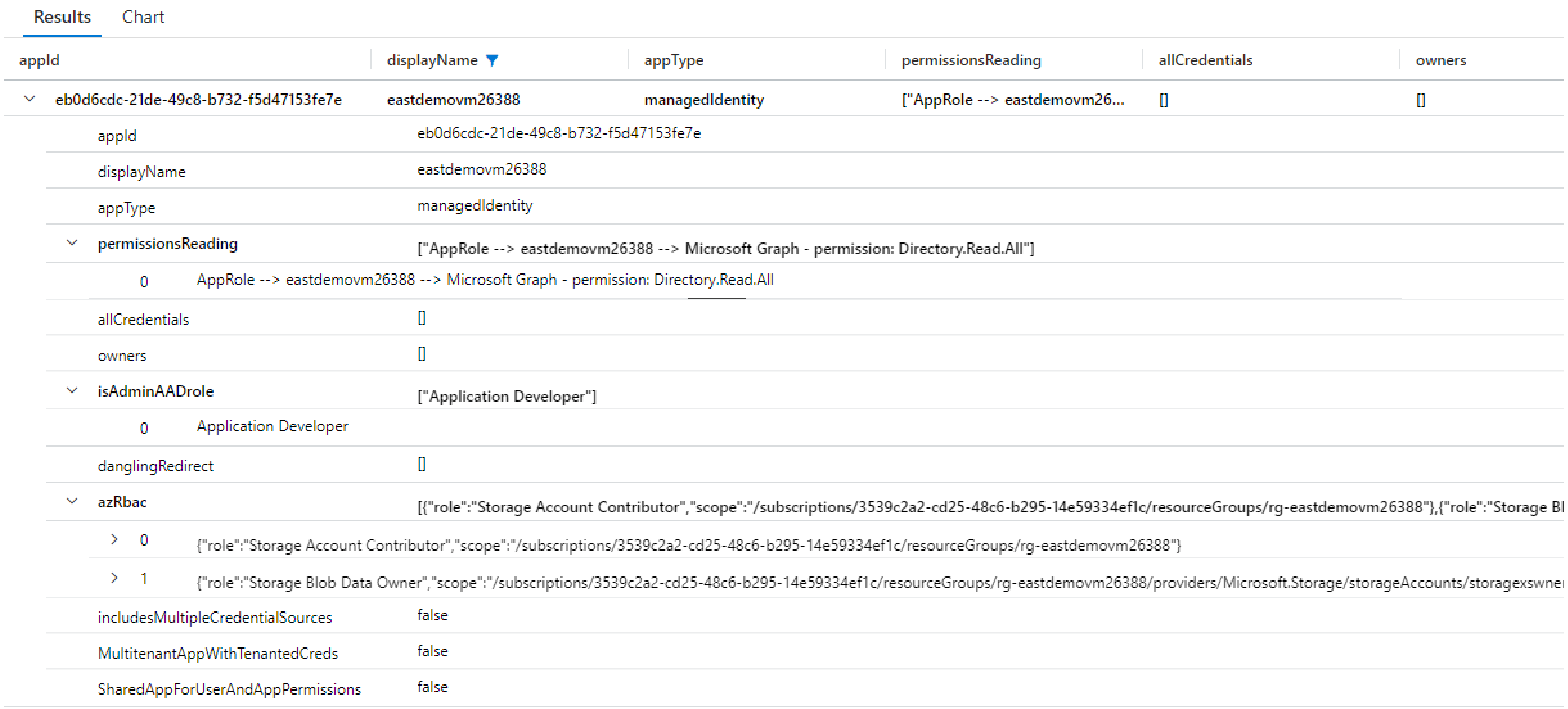
Policy name	State	Creation da
Allowed countries relaxed session policy	On	2/28/2024,
Azure MGMT phishing resistant	On	7/19/2023,
Block all locations except finland	On	11/3/2023,
Block unknown	On	4/29/2023,
CA003: Block legacy authentication	On	10/13/2022
Demo GAP	On	2/29/2024,
baseline	On	10/13/2022
device join	On	4/21/2023,
session	On	5/23/2023,

```
admin [ ~/caoptics ]$
```

Secureworks®

Areas highlighted in Midnight Blizzard

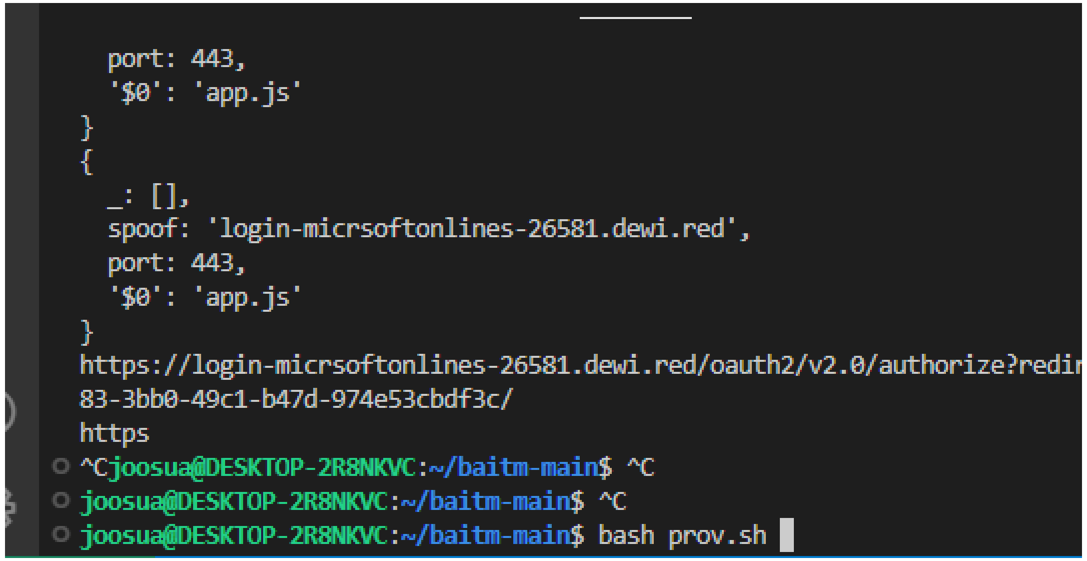
Audit Entra ID Oauth2 apps



appId	displayName	appType	permissionsReading	allCredentials	owners
eb0d6cdc-21de-49c8-b732-f5d47153fe7e	eastdemovm26388	managedIdentity	["AppRole --> eastdemovm26...	[]	[]

appId		eb0d6cdc-21de-49c8-b732-f5d47153fe7e
displayName		eastdemovm26388
appType		managedIdentity
permissionsReading		["AppRole --> eastdemovm26388 --> Microsoft Graph - permission: Directory.Read.All"]
	0	AppRole --> eastdemovm26388 --> Microsoft Graph - permission: Directory.Read.All
allCredentials		[]
owners		[]
isAdminAADrole		["Application Developer"]
	0	Application Developer
danglingRedirect		[]
azRbac		[{"role":"Storage Account Contributor","scope":"/subscriptions/3539c2a2-cd25-48c6-b295-14e59334ef1c/resourceGroups/rg-eastdemovm26388"},{"role":"Storage Bl
	0	{"role":"Storage Account Contributor","scope":"/subscriptions/3539c2a2-cd25-48c6-b295-14e59334ef1c/resourceGroups/rg-eastdemovm26388"}
	1	{"role":"Storage Blob Data Owner","scope":"/subscriptions/3539c2a2-cd25-48c6-b295-14e59334ef1c/resourceGroups/rg-eastdemovm26388/providers/Microsoft.Storage/storageAccounts/storagexswne
includesMultipleCredentialSources		false
MultitenantAppWithTenantedCreds		false
SharedAppForUserAndAppPermissions		false

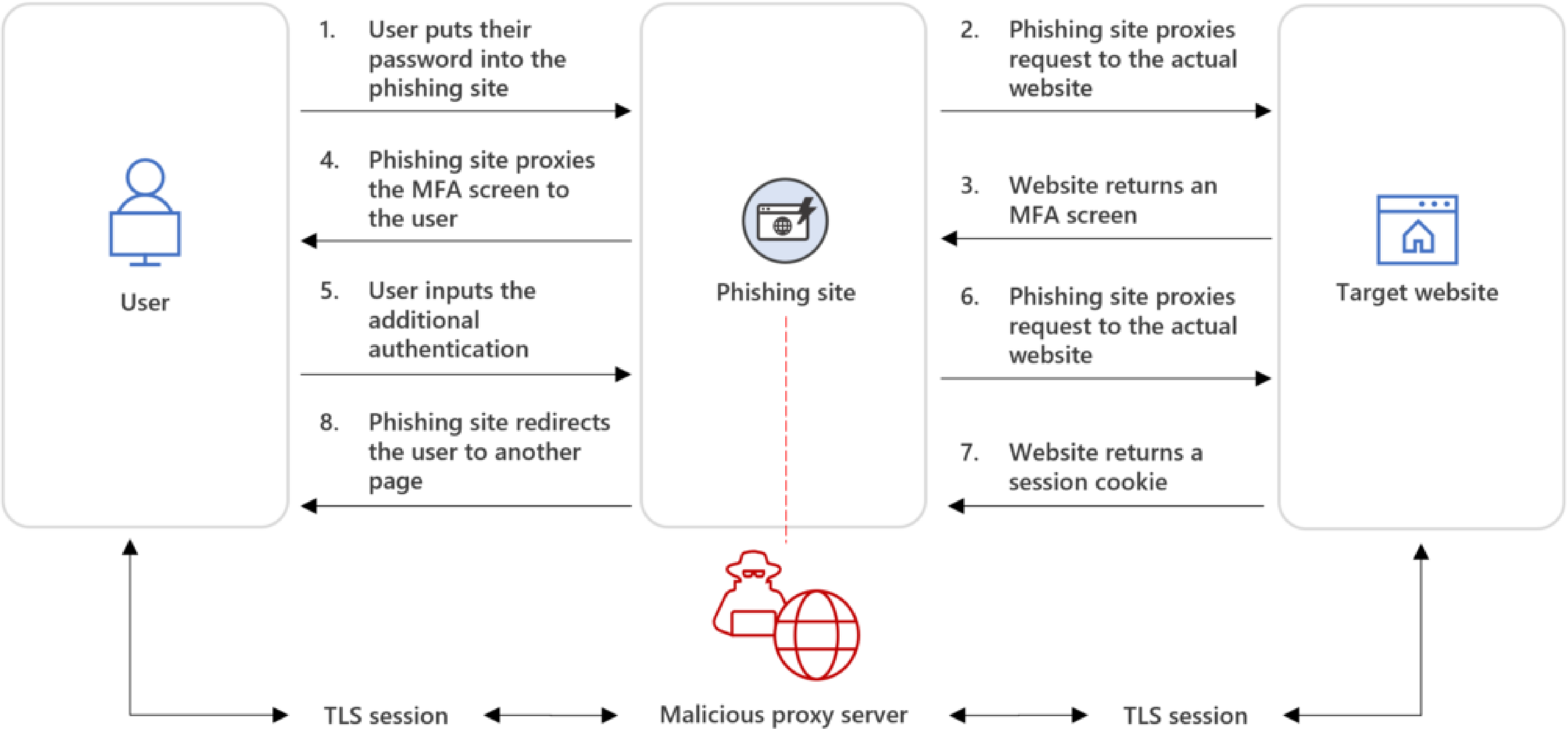
AITm- again?



```
    port: 443,
    '$0': 'app.js'
  }
  {
    _: [],
    spoof: 'login-micrsoftonlines-26581.dewi.red',
    port: 443,
    '$0': 'app.js'
  }
  https://login-micrsoftonlines-26581.dewi.red/oauth2/v2.0/authorize?redir
  83-3bb0-49c1-b47d-974e53cbdf3c/
  https
○ ^Cjoosua@DESKTOP-2R8NKVC:~/baitm-main$ ^C
○ joosua@DESKTOP-2R8NKVC:~/baitm-main$ ^C
○ joosua@DESKTOP-2R8NKVC:~/baitm-main$ bash prov.sh
```

DEV-1101 enables high-volume AiTM campaigns with open-source phishing kit | Microsoft Security Blog

AITm- again?


User
1. User puts their password into the phishing site
4. Phishing site proxies the MFA screen to the user
5. User inputs the additional authentication
8. Phishing site redirects the user to another page
Phishing site
2. Phishing site proxies request to the actual website
3. Website returns an MFA screen
6. Phishing site proxies request to the actual website
7. Website returns a session cookie
Target website
TLS session
Malicious proxy server
TLS session

[DEV-1101 enables high-volume AiTM campaigns with open-source phishing kit | Microsoft Security Blog]

AITM Video

https://youtu.be/6ey-sMBBtyI?si=FWTilW8TOoiOq02R

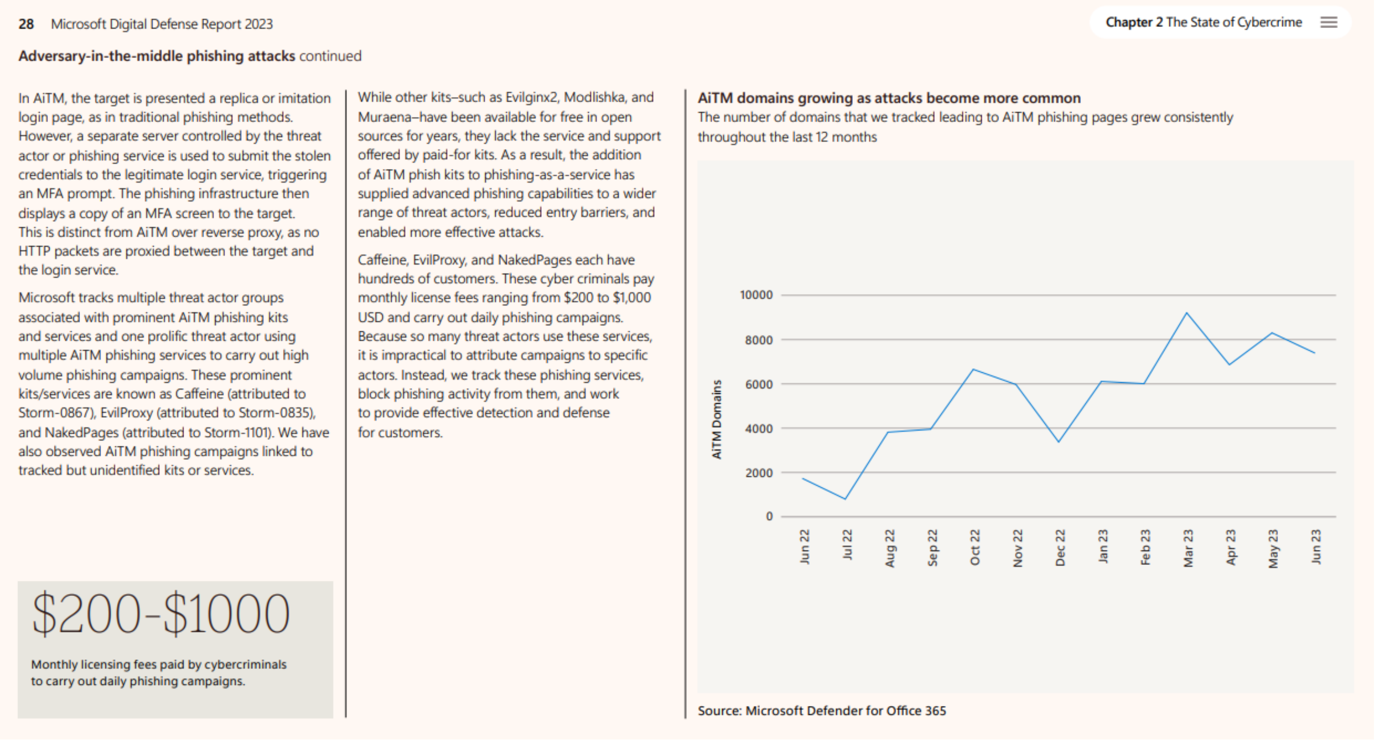
AITM statistics

EvilProxy Phishing Kit Targets Microsoft Users via Indeed.com Vulnerability

hackread.com • 4 min read

Compared to June 2022 baseline the registrations of AITM related domains has seen approx. 400% growth, this correlates or can be stipulated to at least similar growth in attacks

A significant surge in AiTM phishing campaigns was observed in mid-July 2022, indicating an effort to bypass MFA on a massive scale.



28 Microsoft Digital Defense Report 2023

Chapter 2 The State of Cybercrime

Adversary-in-the-middle phishing attacks continued

In AiTM, the target is presented a replica or imitation login page, as in traditional phishing methods. However, a separate server controlled by the threat actor or phishing service is used to submit the stolen credentials to the legitimate login service, triggering an MFA prompt. The phishing infrastructure then displays a copy of an MFA screen to the target. This is distinct from AiTM over reverse proxy, as no HTTP packets are proxied between the target and the login service.

Microsoft tracks multiple threat actor groups associated with prominent AiTM phishing kits and services and one prolific threat actor using multiple AiTM phishing services to carry out high volume phishing campaigns. These prominent kits/services are known as Caffeine (attributed to Storm-0867), EvilProxy (attributed to Storm-0835), and NakedPages (attributed to Storm-1101). We have also observed AiTM phishing campaigns linked to tracked but unidentified kits or services.

$200-$1000

Monthly licensing fees paid by cybercriminals to carry out daily phishing campaigns.

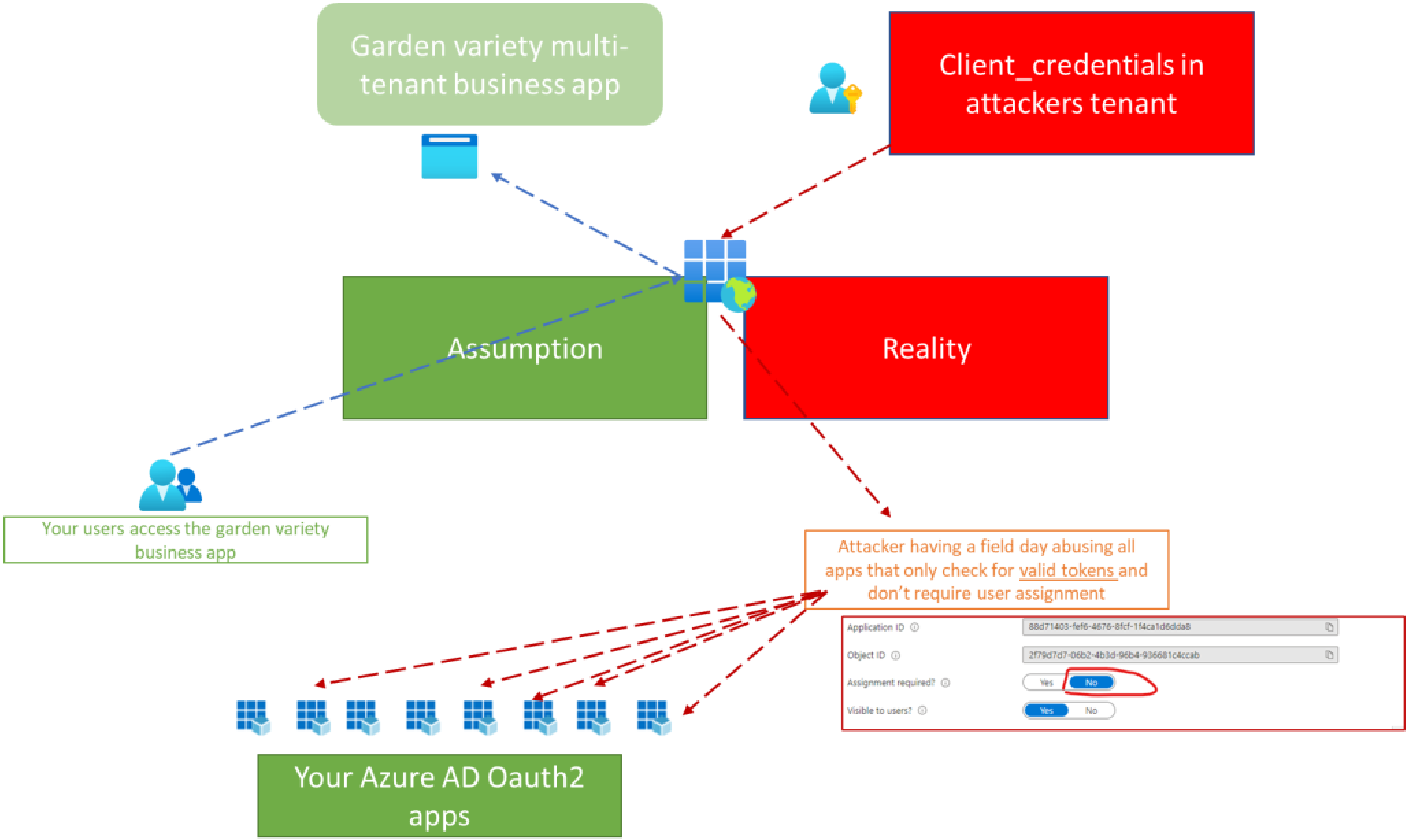
While other kits–such as Evilginx2, Modlishka, and Muraena–have been available for free in open sources for years, they lack the service and support offered by paid-for kits. As a result, the addition of AiTM phish kits to phishing-as-a-service has supplied advanced phishing capabilities to a wider range of threat actors, reduced entry barriers, and enabled more effective attacks.

Caffeine, EvilProxy, and NakedPages each have hundreds of customers. These cyber criminals pay monthly license fees ranging from $200 to $1,000 USD and carry out daily phishing campaigns. Because so many threat actors use these services, it is impractical to attribute campaigns to specific actors. Instead, we track these phishing services, block phishing activity from them, and work to provide effective detection and defense for customers.

AiTM domains growing as attacks become more common

The number of domains that we tracked leading to AiTM phishing pages grew consistently throughout the last 12 months

Source: Microsoft Defender for Office 365



Garden variety multi-tenant business app

Client_credentials in attackers tenant

Assumption

Reality

Your users access the garden variety business app

Attacker having a field day abusing all apps that only check for valid tokens and don't require user assignment

Application ID ⓘ 88d71403-fef6-4676-8fcf-1f4ca1d6dda8

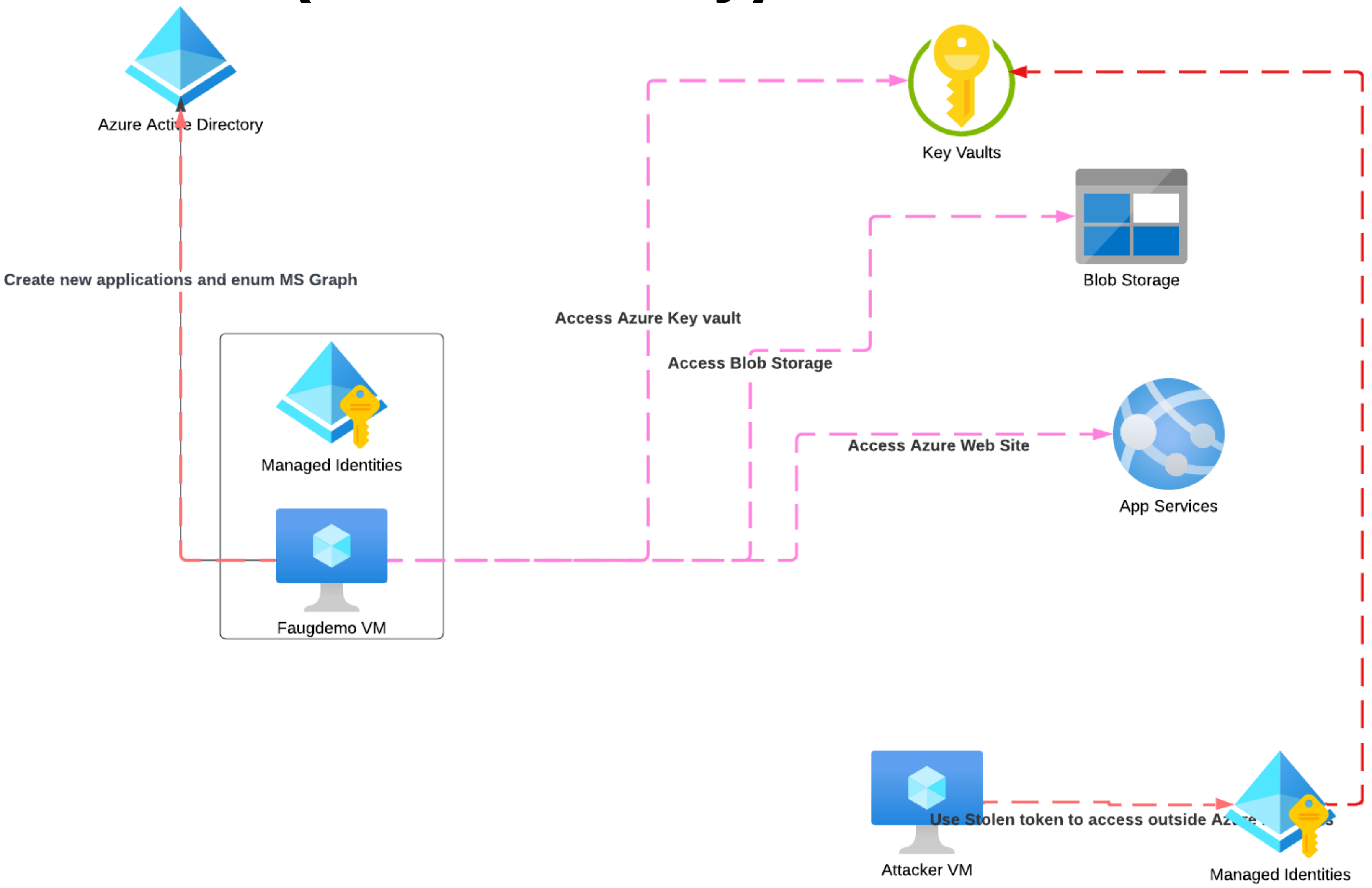
Object ID ⓘ 2f79d7d7-06b2-4b3d-96b4-936681c4ccab

Assignment required? ⓘ Yes [✓ No]

Visible to users? ⓘ [✓ Yes] No

Your Azure AD Oauth2 apps

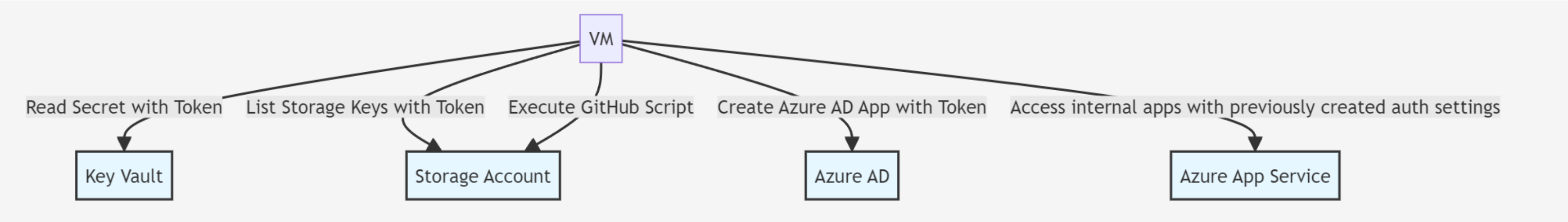
Why EntraID* auth is better, even if you are compromised (observability)



* Formerly known as Azure Active Directory

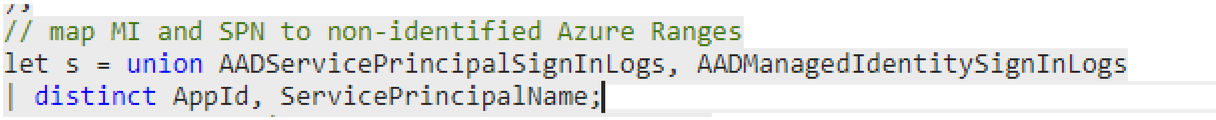
Secureworks®

Logs produced

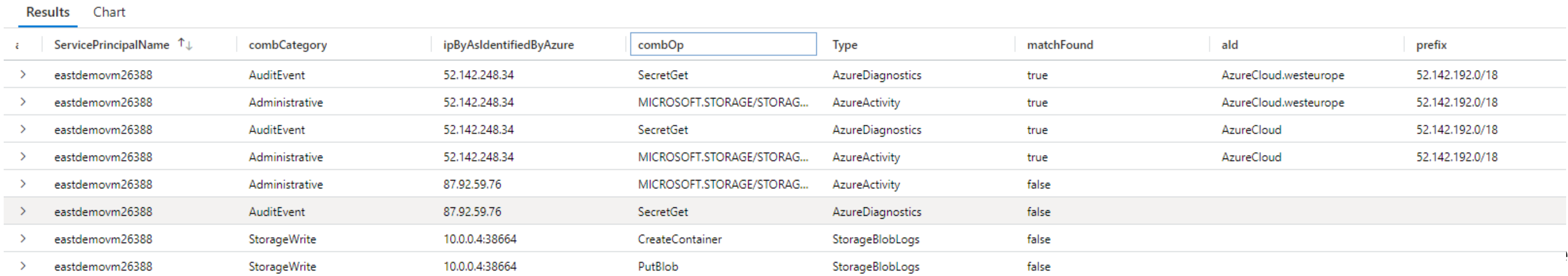


VM
Read Secret with Token
List Storage Keys with Token
Execute GitHub Script
Create Azure AD App with Token
Access internal apps with previously created auth settings
Key Vault
Storage Account
Azure AD
Azure App Service

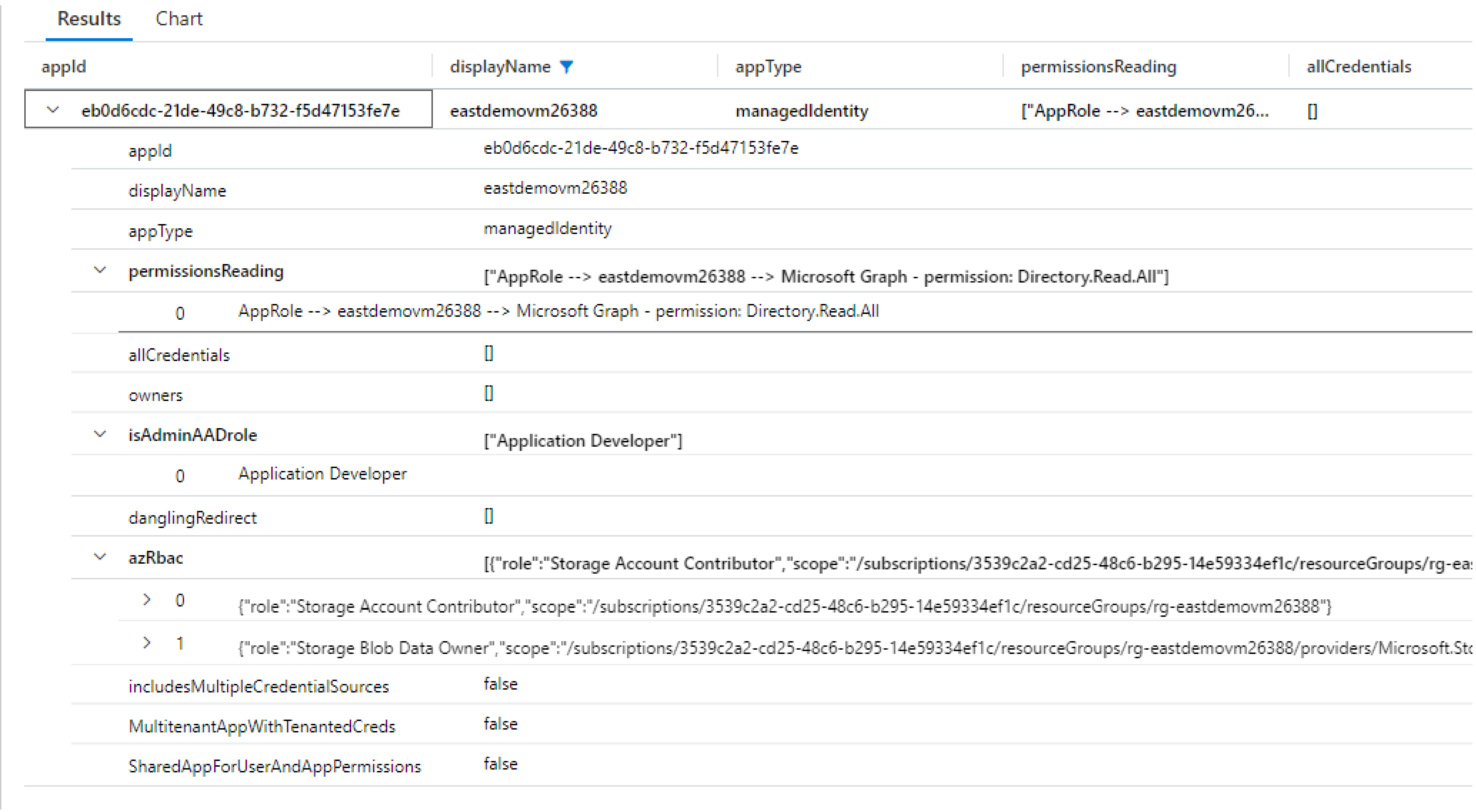
- Who owns the app?
- To which services the app has requested tokens for?
- Which Azure DataPlane actions have the app been up to?
- Which read or write operations has the app been up to in MS Graph?
- Is the managed identity being used outside of Azure IP ranges?
- Does the app has creds outside the UI?
- Does the app have Oauth2 roles?
- Does the app have AAD roles?
- Does the app have Azure RBAC roles?



```
// map MI and SPN to non-identified Azure Ranges
let s = union AADServicePrincipalSignInLogs, AADManagedIdentitySignInLogs
| distinct AppId, ServicePrincipalName;
```

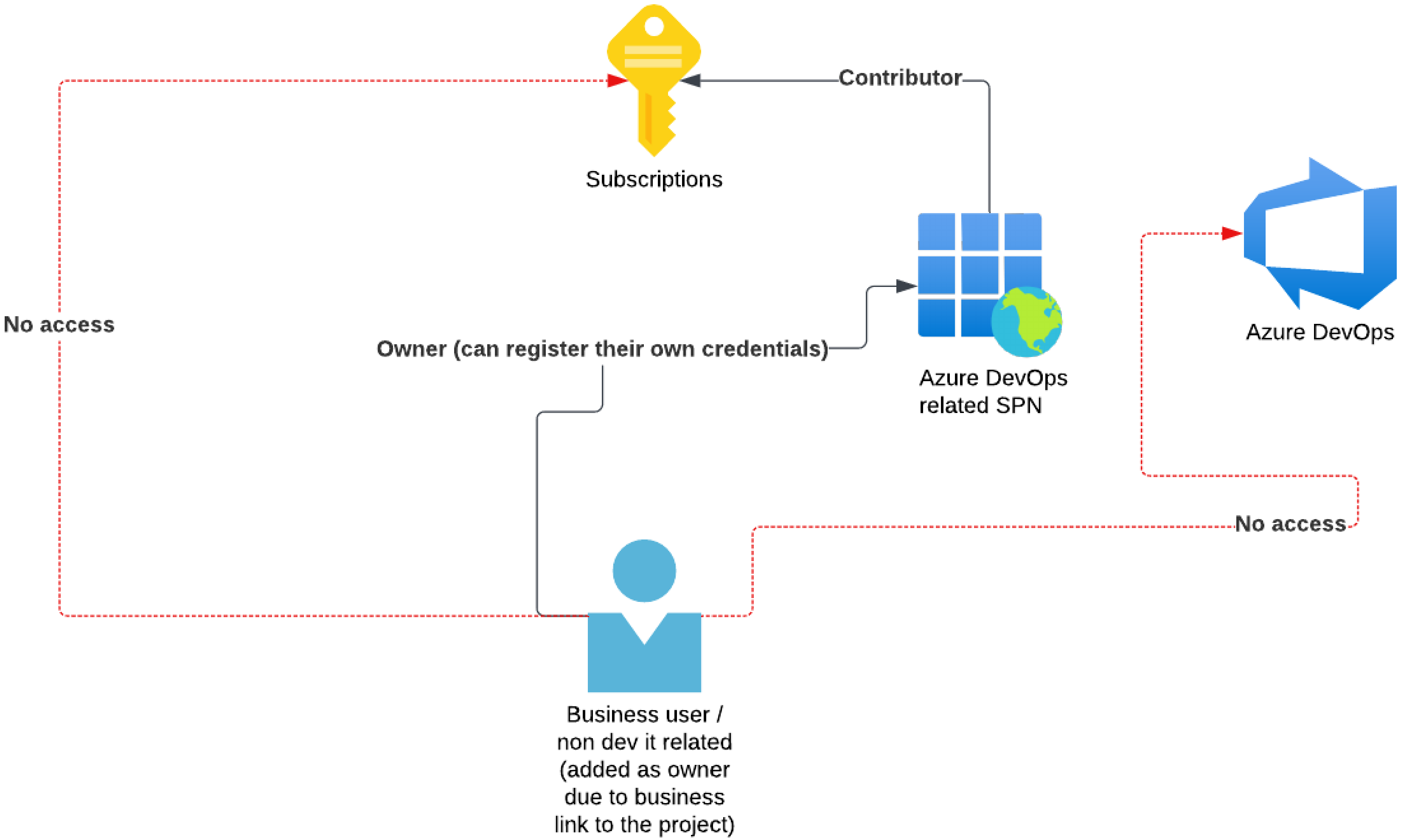


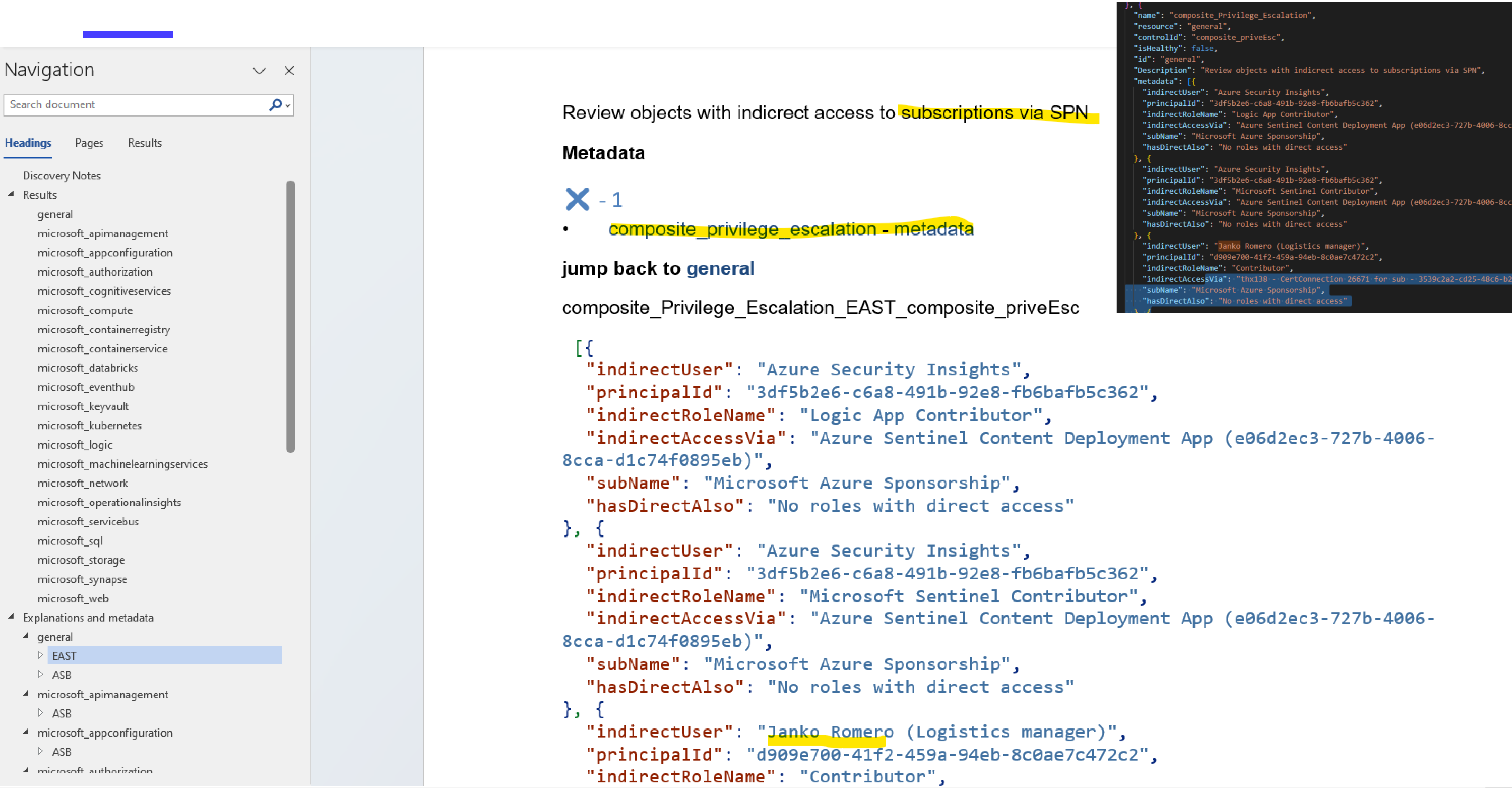
	ServicePrincipalName ↑↓	combCategory	ipByAsIdentifiedByAzure	combOp	Type	matchFound	aId	prefix
>	eastdemovm26388	AuditEvent	52.142.248.34	SecretGet	AzureDiagnostics	true	AzureCloud.westeurope	52.142.192.0/18
>	eastdemovm26388	Administrative	52.142.248.34	MICROSOFT.STORAGE/STORAG...	AzureActivity	true	AzureCloud.westeurope	52.142.192.0/18
>	eastdemovm26388	AuditEvent	52.142.248.34	SecretGet	AzureDiagnostics	true	AzureCloud	52.142.192.0/18
>	eastdemovm26388	Administrative	52.142.248.34	MICROSOFT.STORAGE/STORAG...	AzureActivity	true	AzureCloud	52.142.192.0/18
>	eastdemovm26388	Administrative	87.92.59.76	MICROSOFT.STORAGE/STORAG...	AzureActivity	false		
>	eastdemovm26388	AuditEvent	87.92.59.76	SecretGet	AzureDiagnostics	false		
>	eastdemovm26388	StorageWrite	10.0.0.4:38664	CreateContainer	StorageBlobLogs	false		
>	eastdemovm26388	StorageWrite	10.0.0.4:38664	PutBlob	StorageBlobLogs	false		



appId	displayName	appType	permissionsReading	allCredentials
eb0d6cdc-21de-49c8-b732-f5d47153fe7e	eastdemovm26388	managedIdentity	["AppRole --> eastdemovm26...	[]

appId		eb0d6cdc-21de-49c8-b732-f5d47153fe7e
displayName		eastdemovm26388
appType		managedIdentity
permissionsReading		["AppRole --> eastdemovm26388 --> Microsoft Graph - permission: Directory.Read.All"]
	0	AppRole --> eastdemovm26388 --> Microsoft Graph - permission: Directory.Read.All
allCredentials		[]
owners		[]
isAdminAADrole		["Application Developer"]
	0	Application Developer
danglingRedirect		[]
azRbac		[{"role":"Storage Account Contributor","scope":"/subscriptions/3539c2a2-cd25-48c6-b295-14e59334ef1c/resourceGroups/rg-ea
	0	{"role":"Storage Account Contributor","scope":"/subscriptions/3539c2a2-cd25-48c6-b295-14e59334ef1c/resourceGroups/rg-eastdemovm26388"}
	1	{"role":"Storage Blob Data Owner","scope":"/subscriptions/3539c2a2-cd25-48c6-b295-14e59334ef1c/resourceGroups/rg-eastdemovm26388/providers/Microsoft.St
includesMultipleCredentialSources		false
MultitenantAppWithTenantedCreds		false
SharedAppForUserAndAppPermissions		false


Subscriptions
Contributor
No access
Owner (can register their own credentials)
Azure DevOps related SPN
Azure DevOps
No access
Business user / non dev it related (added as owner due to business link to the project)



Review objects with indicrect access to subscriptions via SPN

Metadata

✕ - 1

- composite_privilege_escalation - metadata

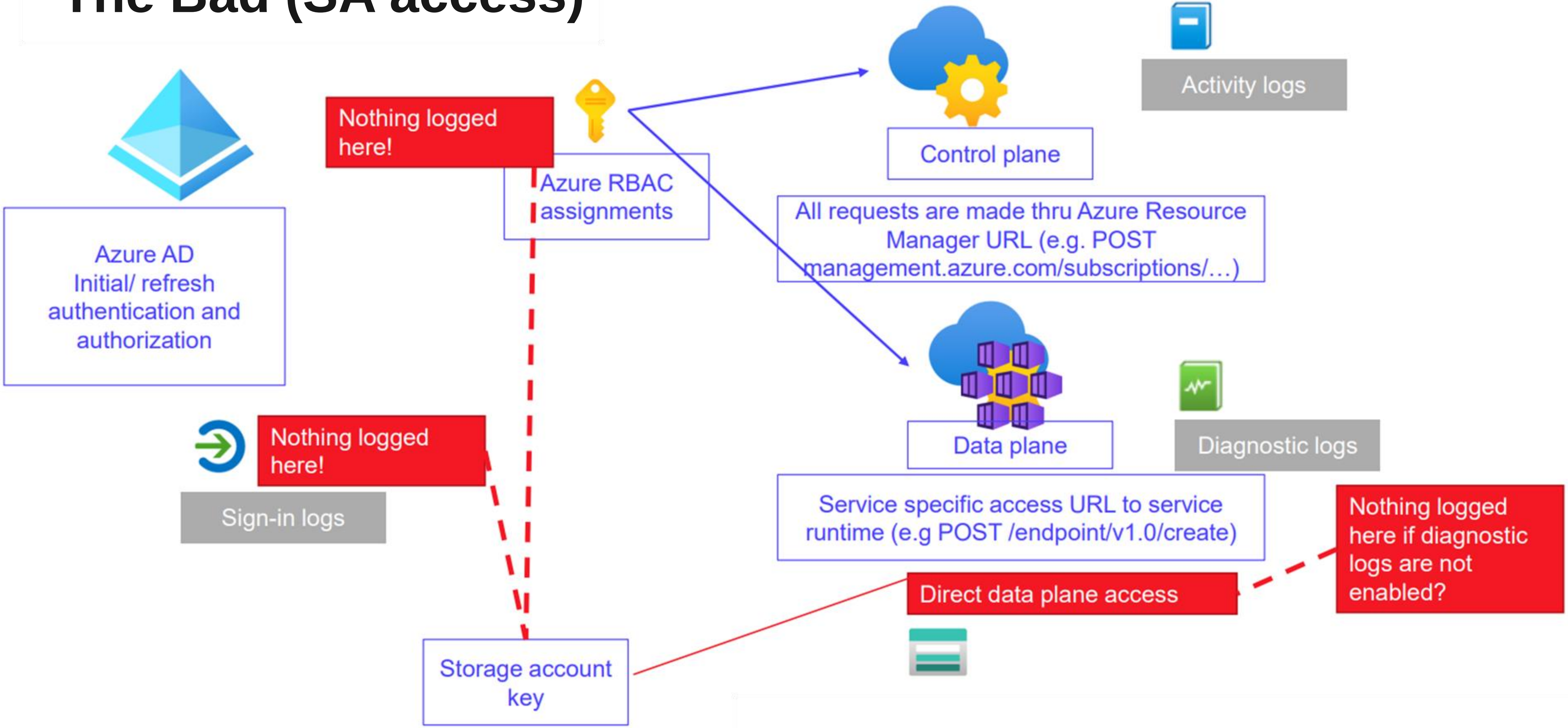
jump back to general

composite_Privilege_Escalation_EAST_composite_priveEsc

```
 [{
  "indirectUser": "Azure Security Insights",
  "principalId": "3df5b2e6-c6a8-491b-92e8-fb6bafb5c362",
  "indirectRoleName": "Logic App Contributor",
  "indirectAccessVia": "Azure Sentinel Content Deployment App (e06d2ec3-727b-4006-8cca-d1c74f0895eb)",
  "subName": "Microsoft Azure Sponsorship",
  "hasDirectAlso": "No roles with direct access"
}, {
  "indirectUser": "Azure Security Insights",
  "principalId": "3df5b2e6-c6a8-491b-92e8-fb6bafb5c362",
  "indirectRoleName": "Microsoft Sentinel Contributor",
  "indirectAccessVia": "Azure Sentinel Content Deployment App (e06d2ec3-727b-4006-8cca-d1c74f0895eb)",
  "subName": "Microsoft Azure Sponsorship",
  "hasDirectAlso": "No roles with direct access"
}, {
  "indirectUser": "Janko Romero (Logistics manager)",
  "principalId": "d909e700-41f2-459a-94eb-8c0ae7c472c2",
  "indirectRoleName": "Contributor",
```

```
}, {
  "name": "composite_Privilege_Escalation",
  "resource": "general",
  "controlId": "composite_priveEsc",
  "isHealthy": false,
  "id": "general",
  "Description": "Review objects with indicrect access to subscriptions via SPN",
  "metadata": [{
    "indirectUser": "Azure Security Insights",
    "principalId": "3df5b2e6-c6a8-491b-92e8-fb6bafb5c362",
    "indirectRoleName": "Logic App Contributor",
    "indirectAccessVia": "Azure Sentinel Content Deployment App (e06d2ec3-727b-4006-8cc
    "subName": "Microsoft Azure Sponsorship",
    "hasDirectAlso": "No roles with direct access"
  }, {
    "indirectUser": "Azure Security Insights",
    "principalId": "3df5b2e6-c6a8-491b-92e8-fb6bafb5c362",
    "indirectRoleName": "Microsoft Sentinel Contributor",
    "indirectAccessVia": "Azure Sentinel Content Deployment App (e06d2ec3-727b-4006-8cc
    "subName": "Microsoft Azure Sponsorship",
    "hasDirectAlso": "No roles with direct access"
  }, {
    "indirectUser": "Janko Romero (Logistics manager)",
    "principalId": "d909e700-41f2-459a-94eb-8c0ae7c472c2",
    "indirectRoleName": "Contributor",
    "indirectAccessVia": "thx138 - CertConnection 26671 for sub - 3539c2a2-cd25-48c6-b2
    "subName": "Microsoft Azure Sponsorship",
    "hasDirectAlso": "No roles with direct access"
  }, {
```

The Bad (SA access)



Nothing logged here!

Azure RBAC assignments

Control plane

Activity logs

All requests are made thru Azure Resource Manager URL (e.g. POST management.azure.com/subscriptions/…)

Azure AD Initial/ refresh authentication and authorization

Nothing logged here!

Sign-in logs

Data plane

Diagnostic logs

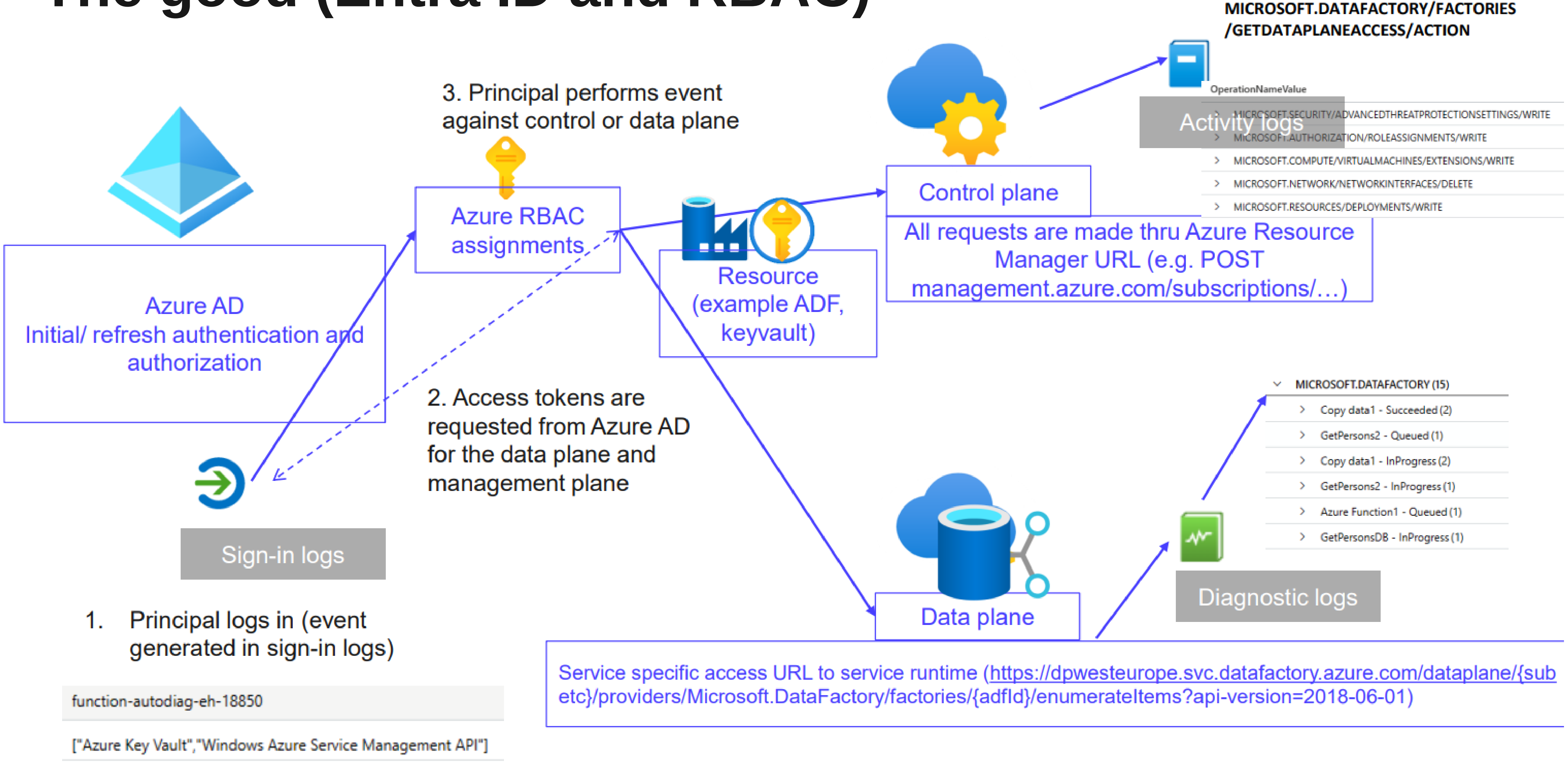
Service specific access URL to service runtime (e.g POST /endpoint/v1.0/create)

Direct data plane access

Nothing logged here if diagnostic logs are not enabled?

Storage account key

The good (Entra ID and RBAC)



Azure AD
Initial/ refresh authentication and authorization

Sign-in logs

1. Principal logs in (event generated in sign-in logs)

function-autodiag-eh-18850

["Azure Key Vault","Windows Azure Service Management API"]

2. Access tokens are requested from Azure AD for the data plane and management plane

3. Principal performs event against control or data plane

Azure RBAC assignments

Resource (example ADF, keyvault)

Control plane

All requests are made thru Azure Resource Manager URL (e.g. POST management.azure.com/subscriptions/…)

MICROSOFT.DATAFACTORY/FACTORIES
/GETDATAPLANEACCESS/ACTION

Activity logs

OperationNameValue

- MICROSOFT.SECURITY/ADVANCEDTHREATPROTECTIONSETTINGS/WRITE
- MICROSOFT.AUTHORIZATION/ROLEASSIGNMENTS/WRITE
- MICROSOFT.COMPUTE/VIRTUALMACHINES/EXTENSIONS/WRITE
- MICROSOFT.NETWORK/NETWORKINTERFACES/DELETE
- MICROSOFT.RESOURCES/DEPLOYMENTS/WRITE

Data plane

Service specific access URL to service runtime (https://dpwesteurope.svc.datafactory.azure.com/dataplane/{sub etc}/providers/Microsoft.DataFactory/factories/{adfId}/enumerateItems?api-version=2018-06-01)

Diagnostic logs

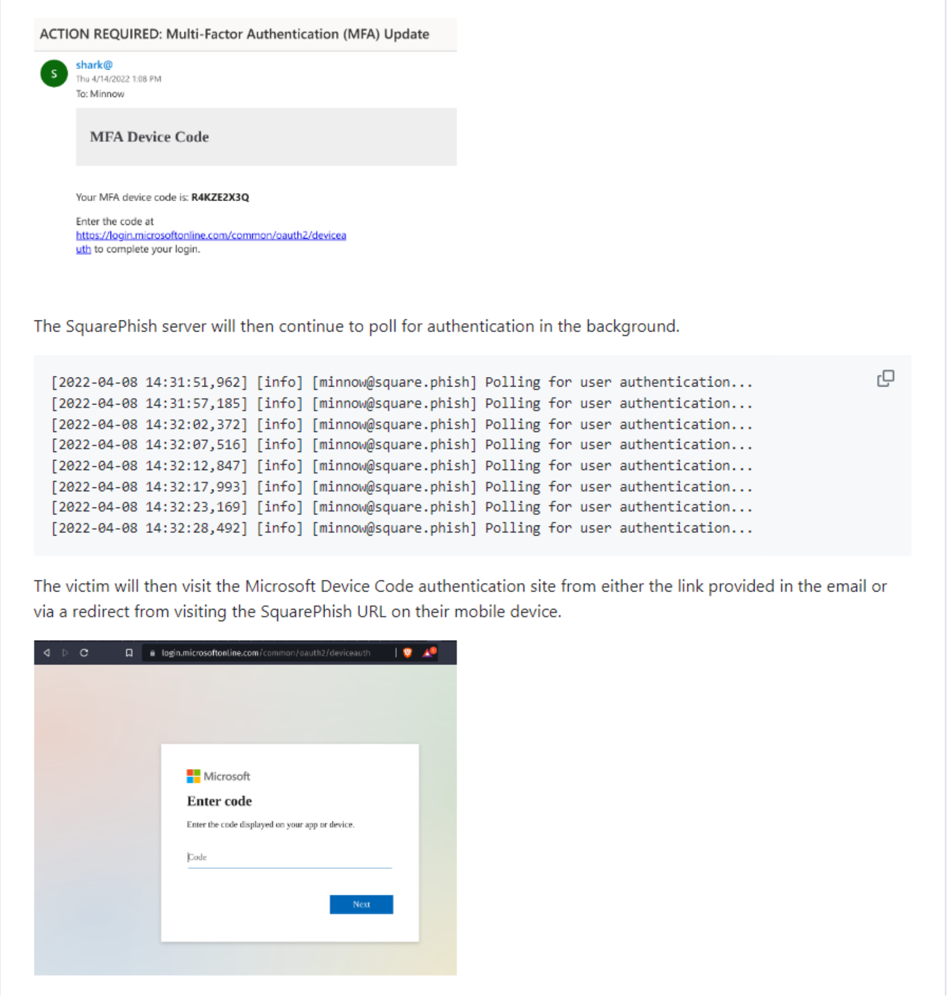
MICROSOFT.DATAFACTORY (15)

- Copy data1 - Succeeded (2)
- GetPersons2 - Queued (1)
- Copy data1 - InProgress (2)
- GetPersons2 - InProgress (1)
- Azure Function1 - Queued (1)
- GetPersonsDB - InProgress (1)

Entra

[secureworks/squarephish (github.com)](https://github.com/secureworks/squarephish)

Block Device Code Flow used in many attacks



ACTION REQUIRED: Multi-Factor Authentication (MFA) Update

shark@
Thu 4/14/2022 1:08 PM
To: Minnow

MFA Device Code

Your MFA device code is: **R4KZE2X3Q**

Enter the code at https://login.microsoftonline.com/common/oauth2/deviceauth to complete your login.

The SquarePhish server will then continue to poll for authentication in the background.

```
[2022-04-08 14:31:51,962] [info] [minnow@square.phish] Polling for user authentication...
[2022-04-08 14:31:57,185] [info] [minnow@square.phish] Polling for user authentication...
[2022-04-08 14:32:02,372] [info] [minnow@square.phish] Polling for user authentication...
[2022-04-08 14:32:07,516] [info] [minnow@square.phish] Polling for user authentication...
[2022-04-08 14:32:12,847] [info] [minnow@square.phish] Polling for user authentication...
[2022-04-08 14:32:17,993] [info] [minnow@square.phish] Polling for user authentication...
[2022-04-08 14:32:23,169] [info] [minnow@square.phish] Polling for user authentication...
[2022-04-08 14:32:28,492] [info] [minnow@square.phish] Polling for user authentication...
```

The victim will then visit the Microsoft Device Code authentication site from either the link provided in the email or via a redirect from visiting the SquarePhish URL on their mobile device.

login.microsoftonline.com/common/oauth2/deviceauth

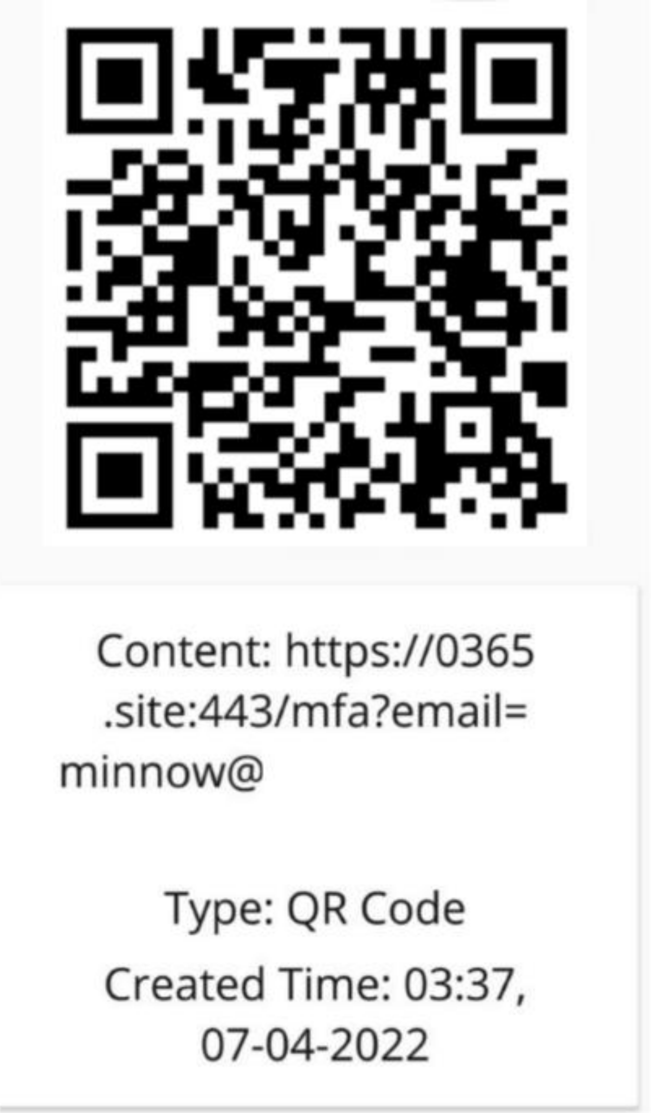
Microsoft

Enter code

Enter the code displayed on your app or device.

Code

Next



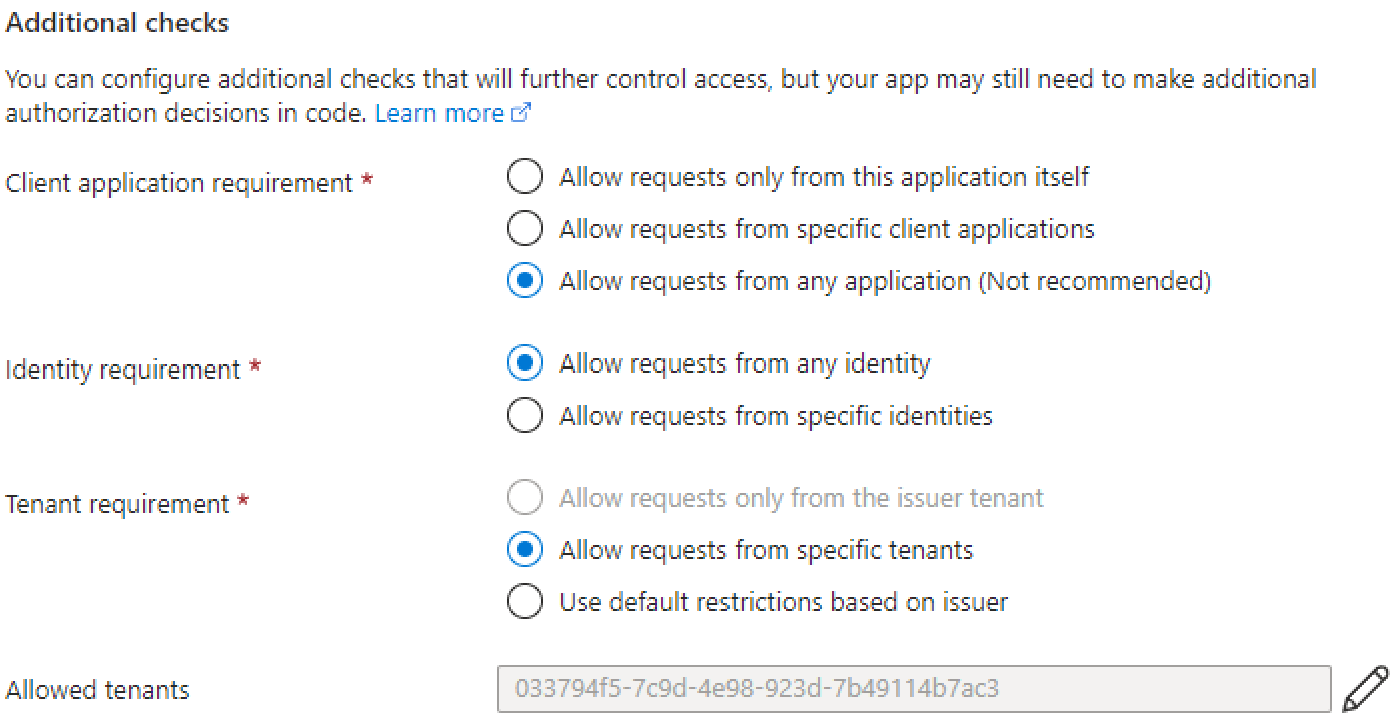
Content: https://0365.site:443/mfa?email=minnow@

Type: QR Code

Created Time: 03:37, 07-04-2022

Azure Web Apps

We had recently attack demonstrated which could implant untrusted AAD apps into victim tenant, one of the main goals was to access Azure Web Apps using the out-of-the-box config for Azure AD Auth. These attacks worked as long as the Azure Web App only checked that the Issuer and Audience values in the tokens were correct (Essentially any user or SPN in the tenant can satisfy those conditions)



Additional checks

You can configure additional checks that will further control access, but your app may still need to make additional authorization decisions in code. Learn more

Client application requirement *
- ○ Allow requests only from this application itself
- ○ Allow requests from specific client applications
- ● Allow requests from any application (Not recommended)

Identity requirement *
- ● Allow requests from any identity
- ○ Allow requests from specific identities

Tenant requirement *
- ○ Allow requests only from the issuer tenant
- ● Allow requests from specific tenants
- ○ Use default restrictions based on issuer

Allowed tenants: 033794f5-7c9d-4e98-923d-7b49114b7ac3

Secureworks

End!