Identity based attack paths in Azure and Azure AD

Cody Burkard
Principal Security Architect
cody@o3c.no | +47 482 77 939

Identity Day Norway
16.03.2023

O3 CYBER

About me

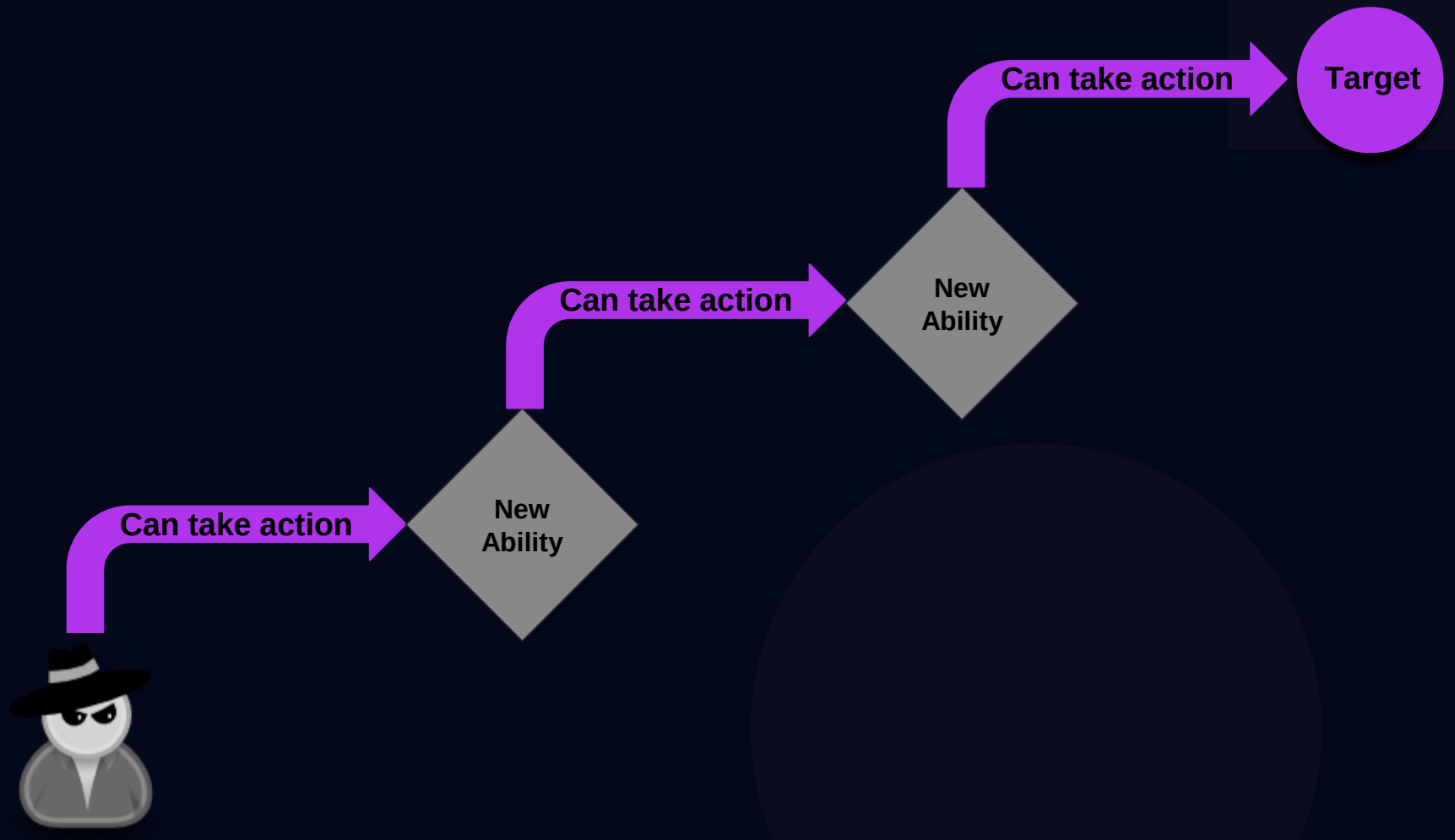
- Cody Burkard
- Partner and Principal Consultant O3 Cyber
- Security testing and cloud security architecture
- Security researcher – Offensive primitives in Azure

Topics

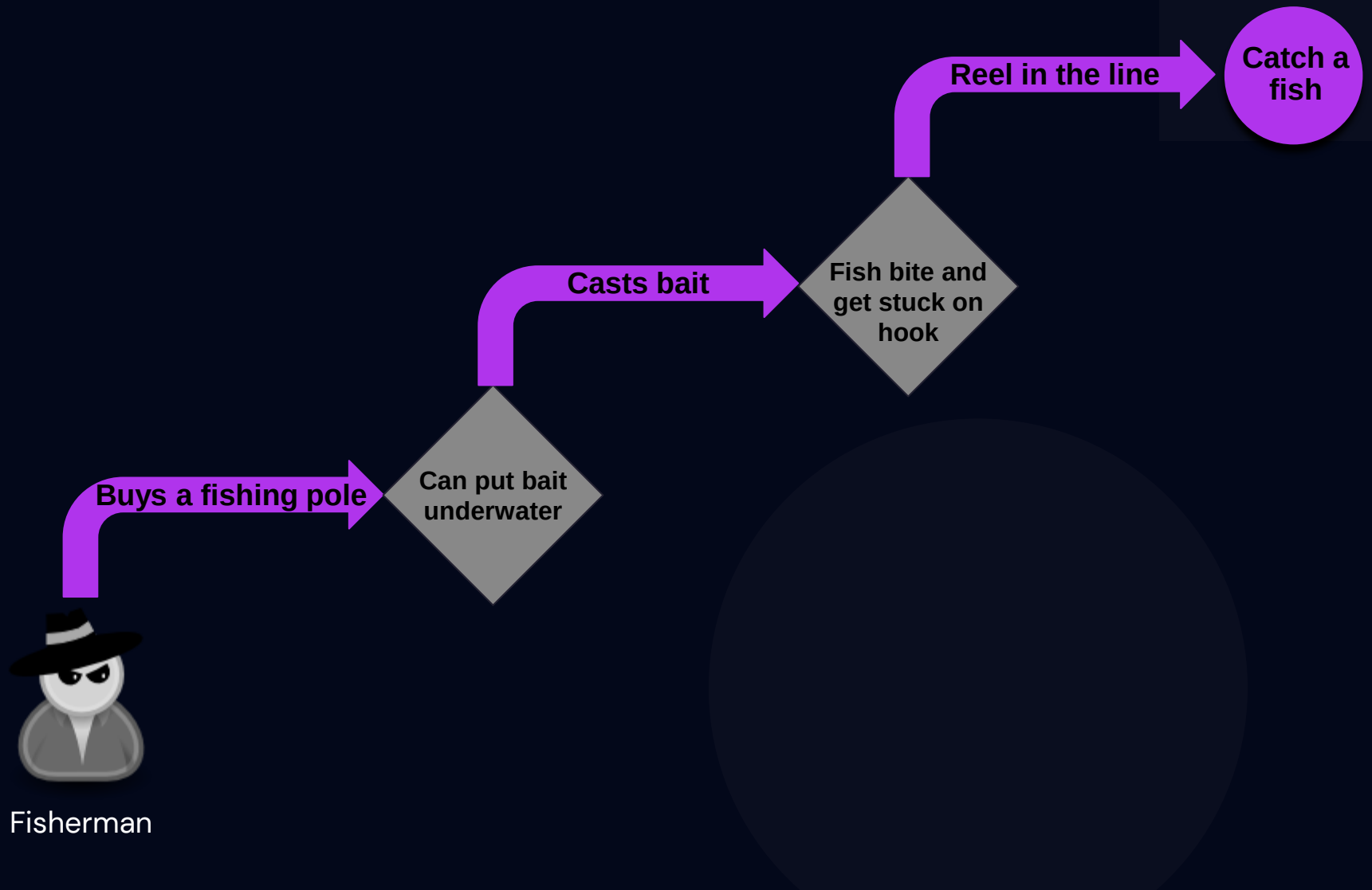
- Introduction to attack paths
- Identity-centric attack primitives
- Azure AD Background
- Putting it together

A brief introduction to attack paths

What is an “attack path”?

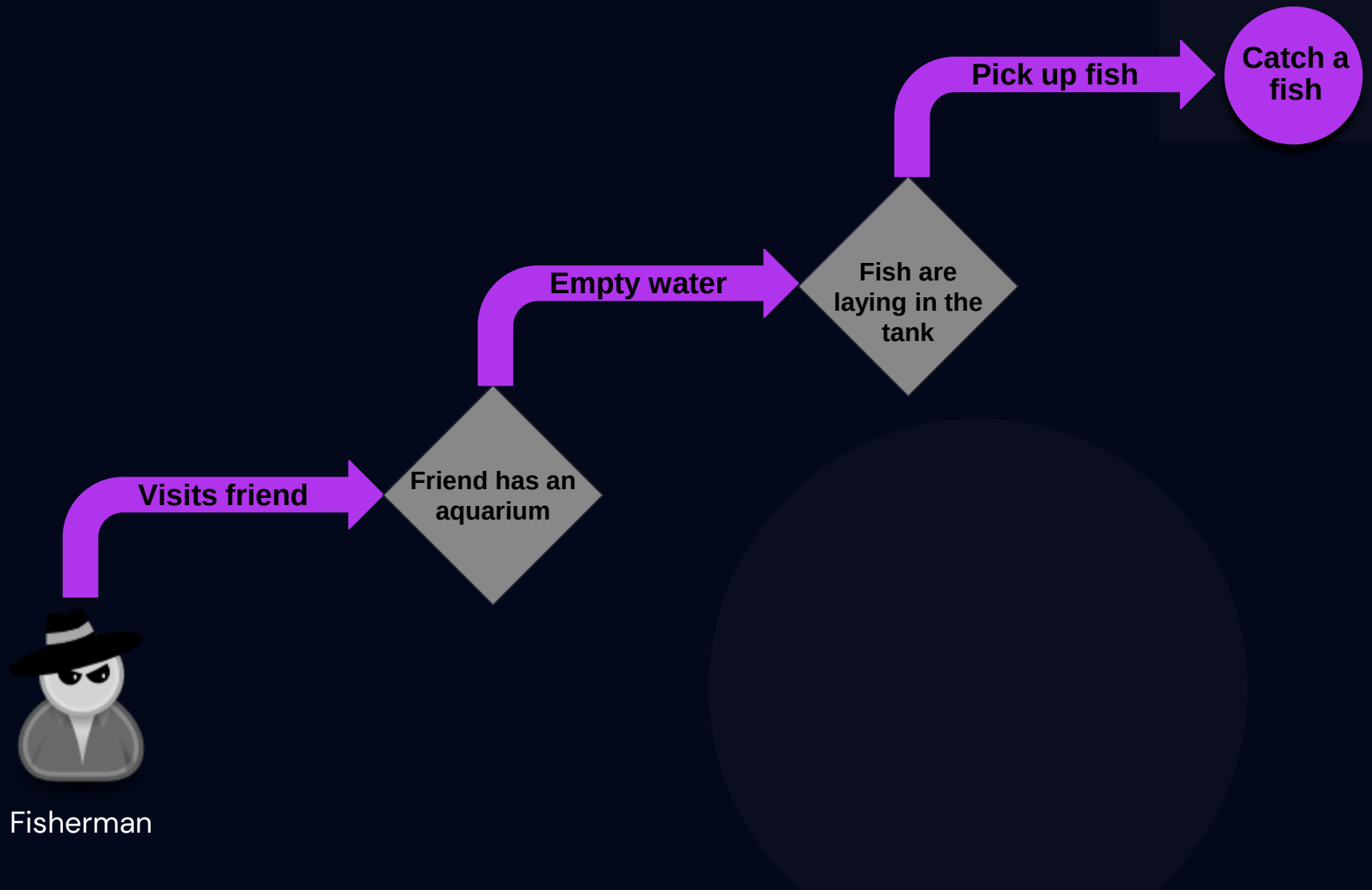


What is an “attack path”?



Also an attack path...

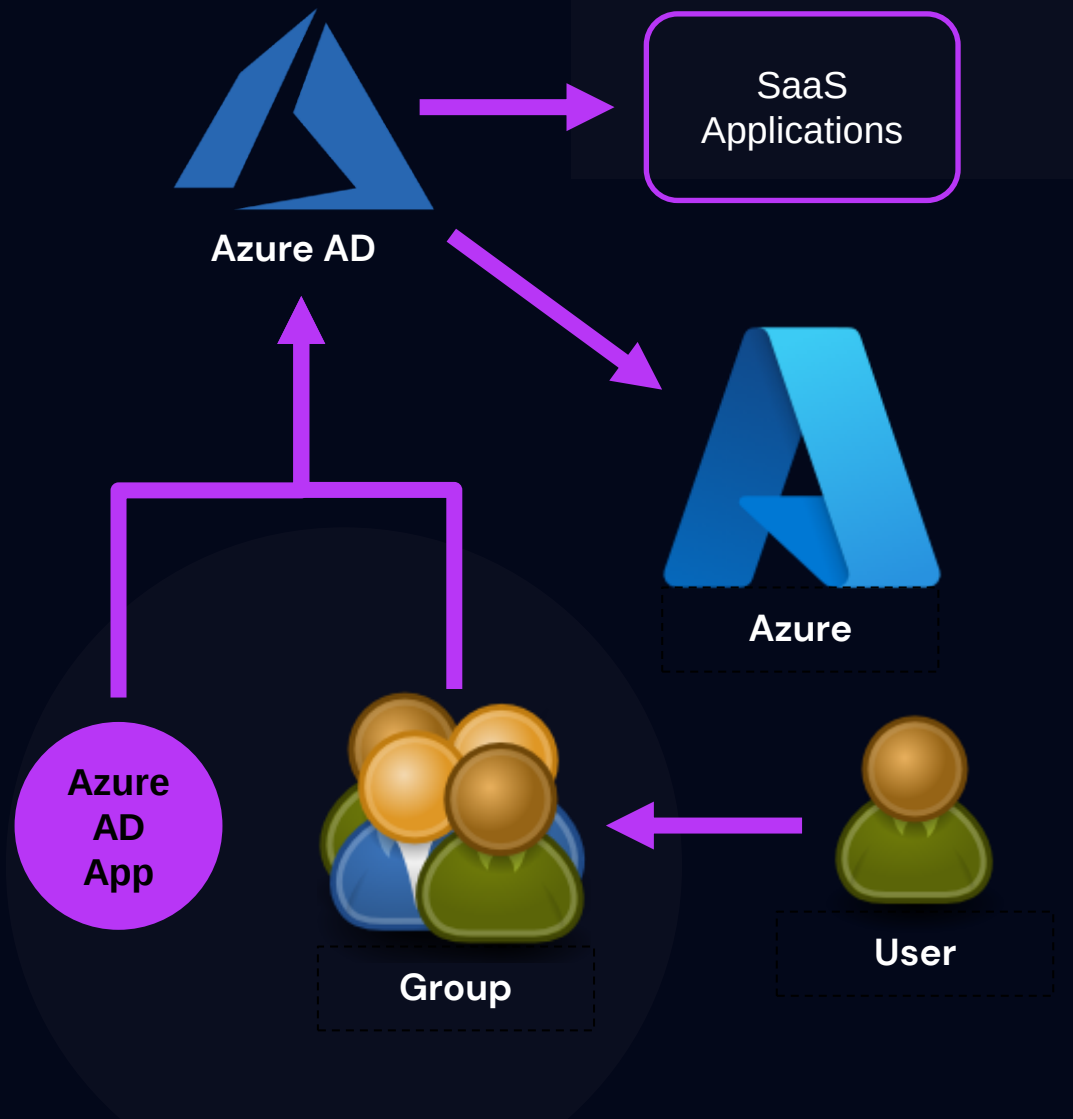
Alternative attack path



Azure AD technical background

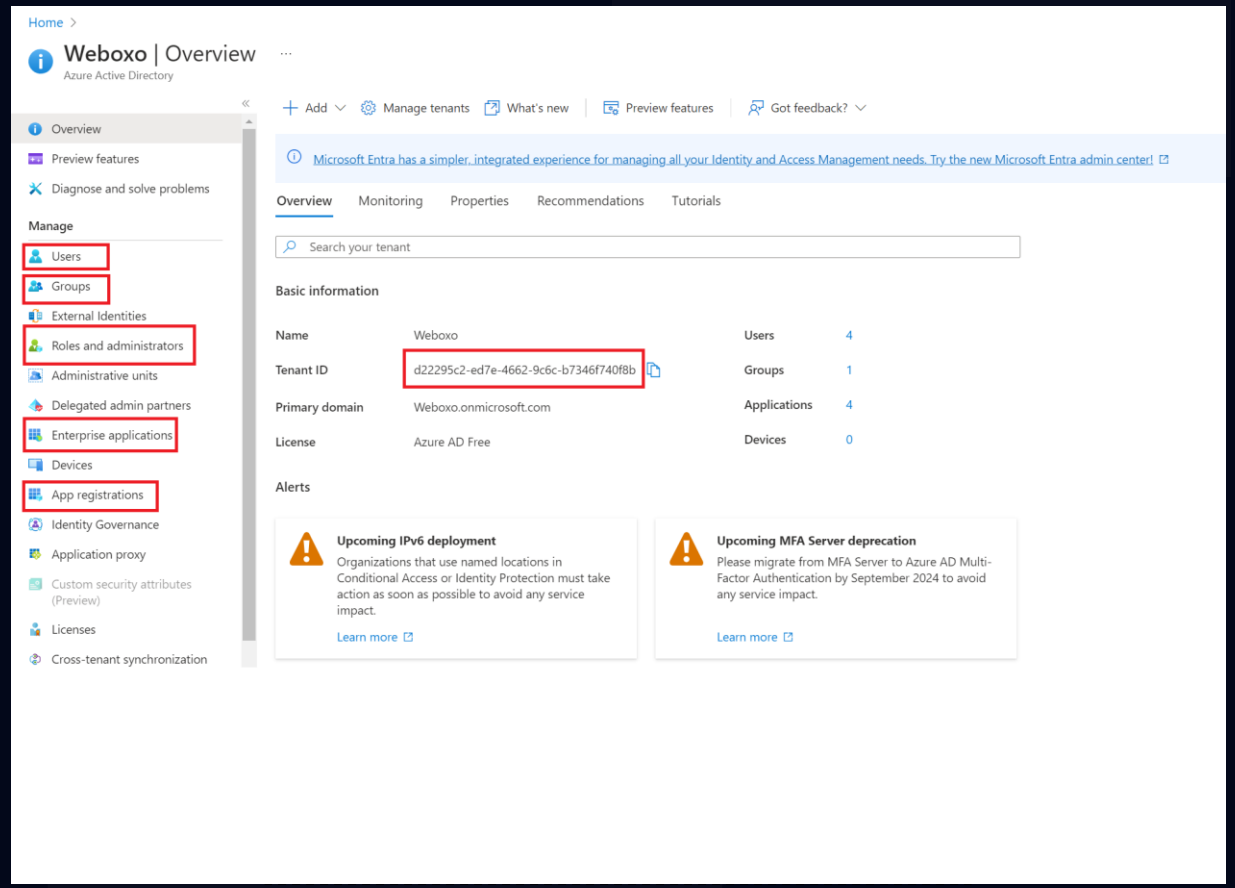
Azure Active Directory

- Cloud-based directory services and IAM platform
- Microsoft Online Apps and Azure
- SSO to SaaS
- Custom integration via Identity Platform



How Azure AD Works

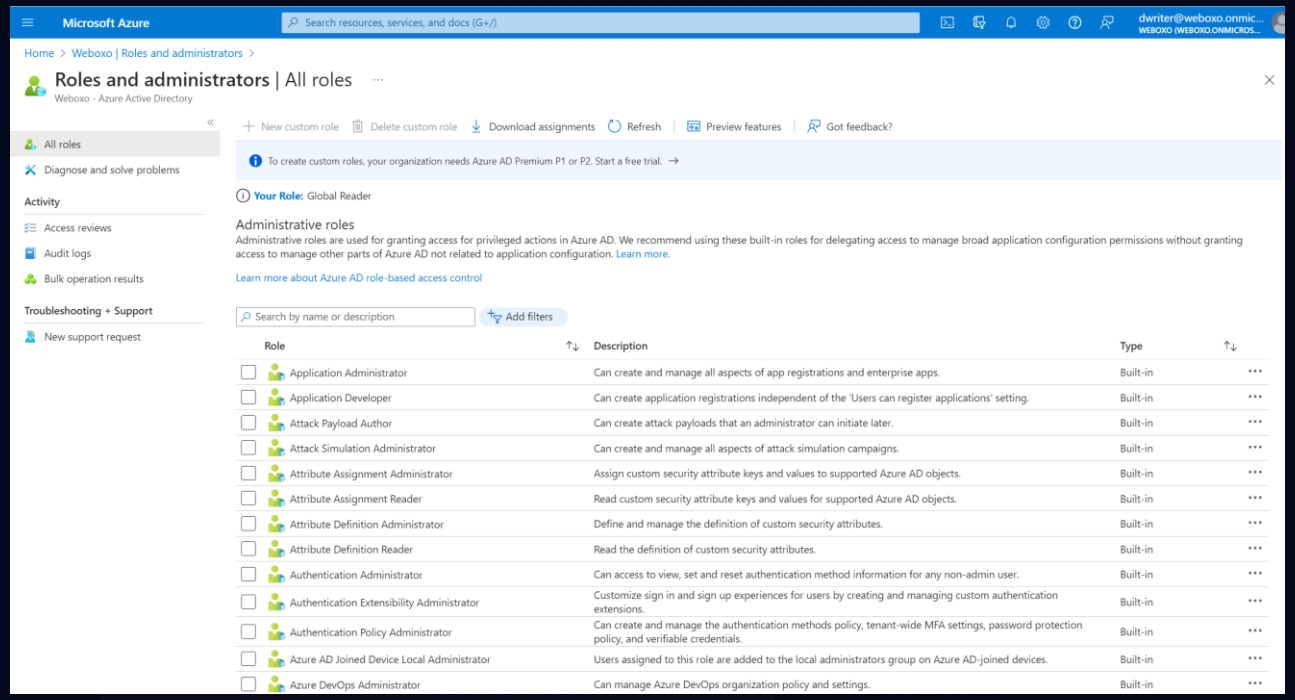
- Organizations have a unique tenant
- Directory objects:
  - Users
  - Groups
  - Service Principals
  - Application Objects
  - Devices
- To manage a tenant you need an *Azure AD Role* assigned to your user



Azure AD in Azure Portal

How Azure AD Works - Roles

- *Azure AD Roles* contain permissions to modify objects in the Microsoft Online ecosystem.
- Administration of Azure AD and M365 with limited permissions
- Principle of least privilege
  - Limit Global Admins

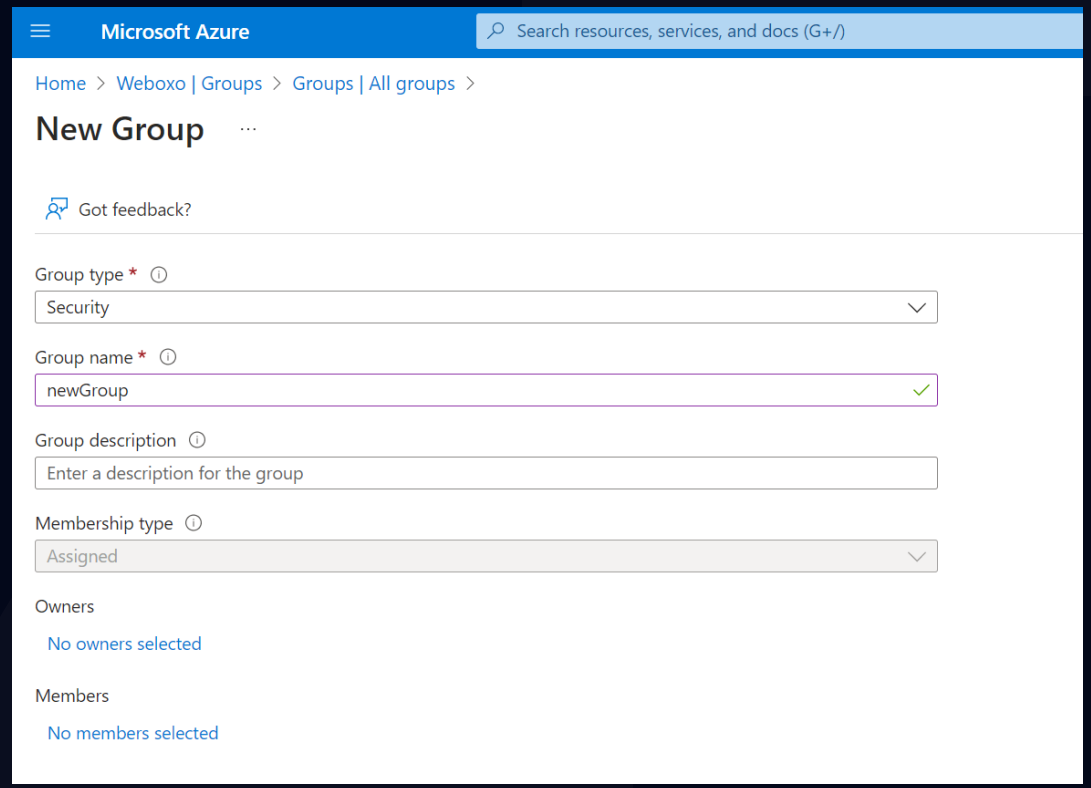


Azure AD Roles

Under the hood

What happens when you click save?

POST https://graph.microsoft.com/v1.0/groups



Creating a group

Azure AD management APIs

Create Group, update group, add user to group...

Group Admin

POST
https://graph.microsoft.com/v1.0/groups

Authorization:

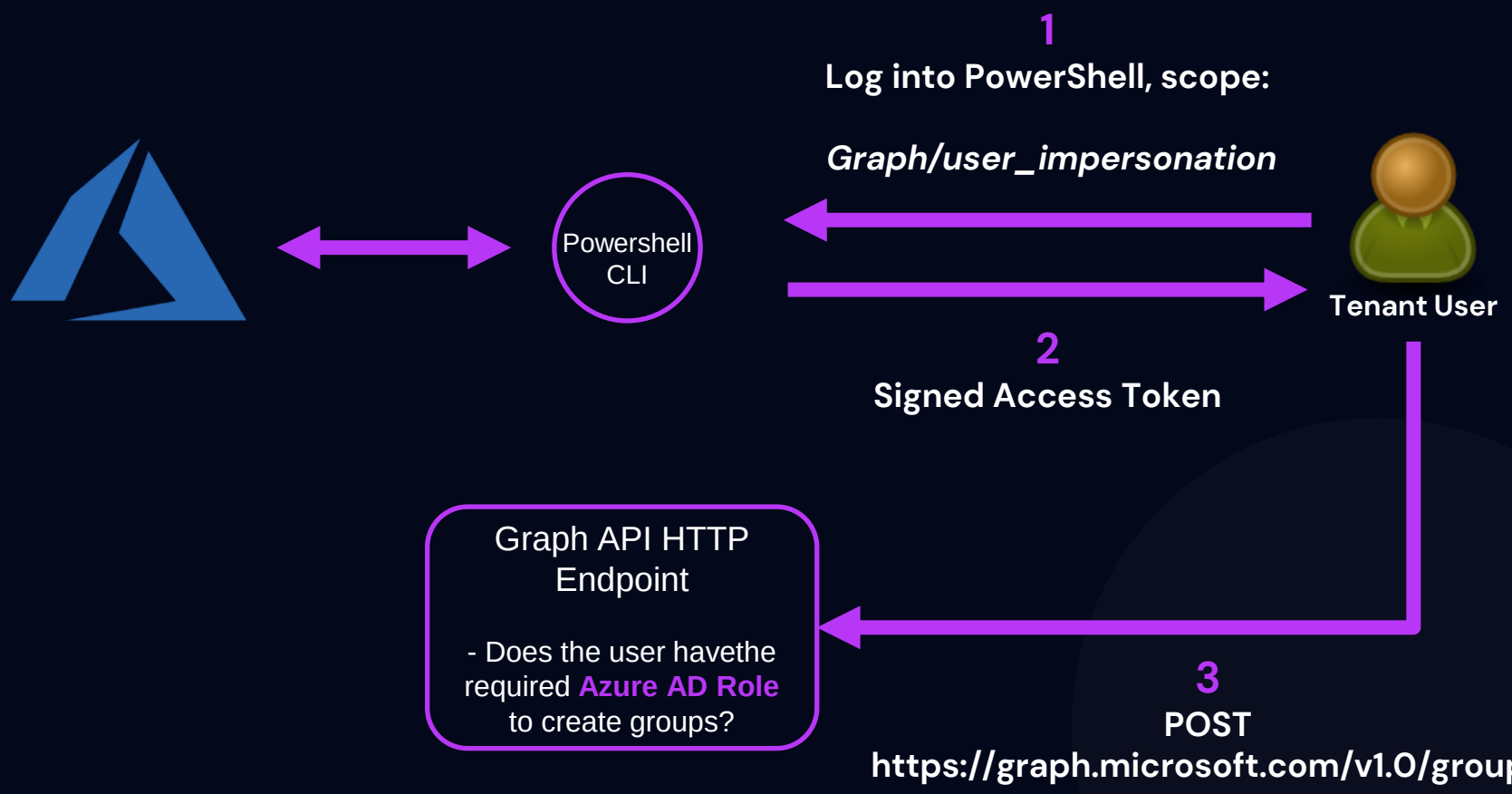
{
...jwt bearer token...
}

Graph API

- Authenticates based on Azure AD role and graph scope

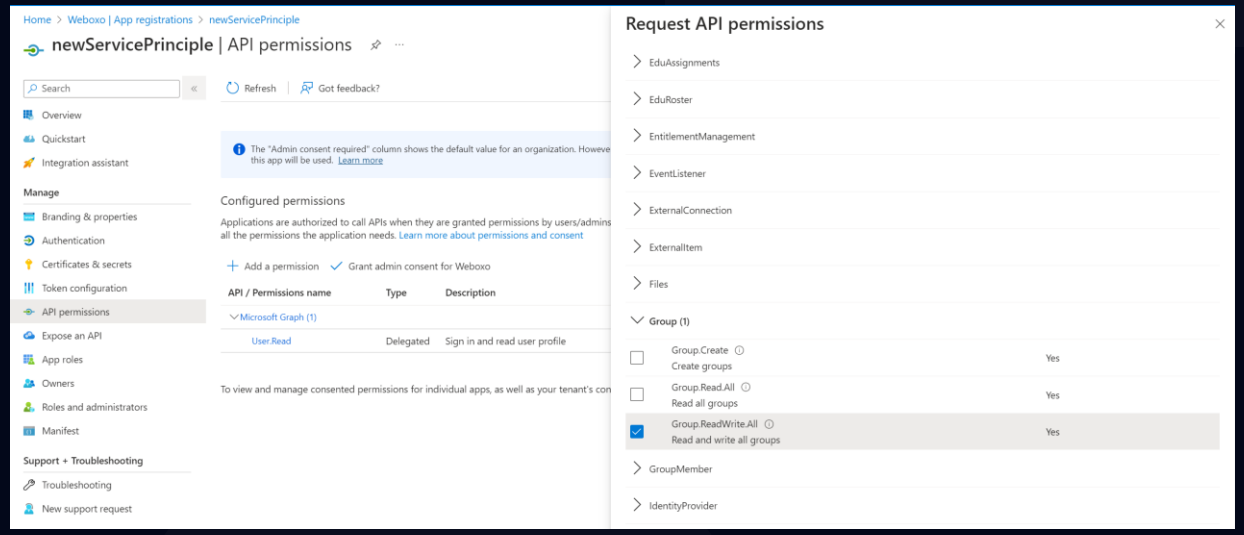
Does my user have access?

Azure AD Auth – User Access



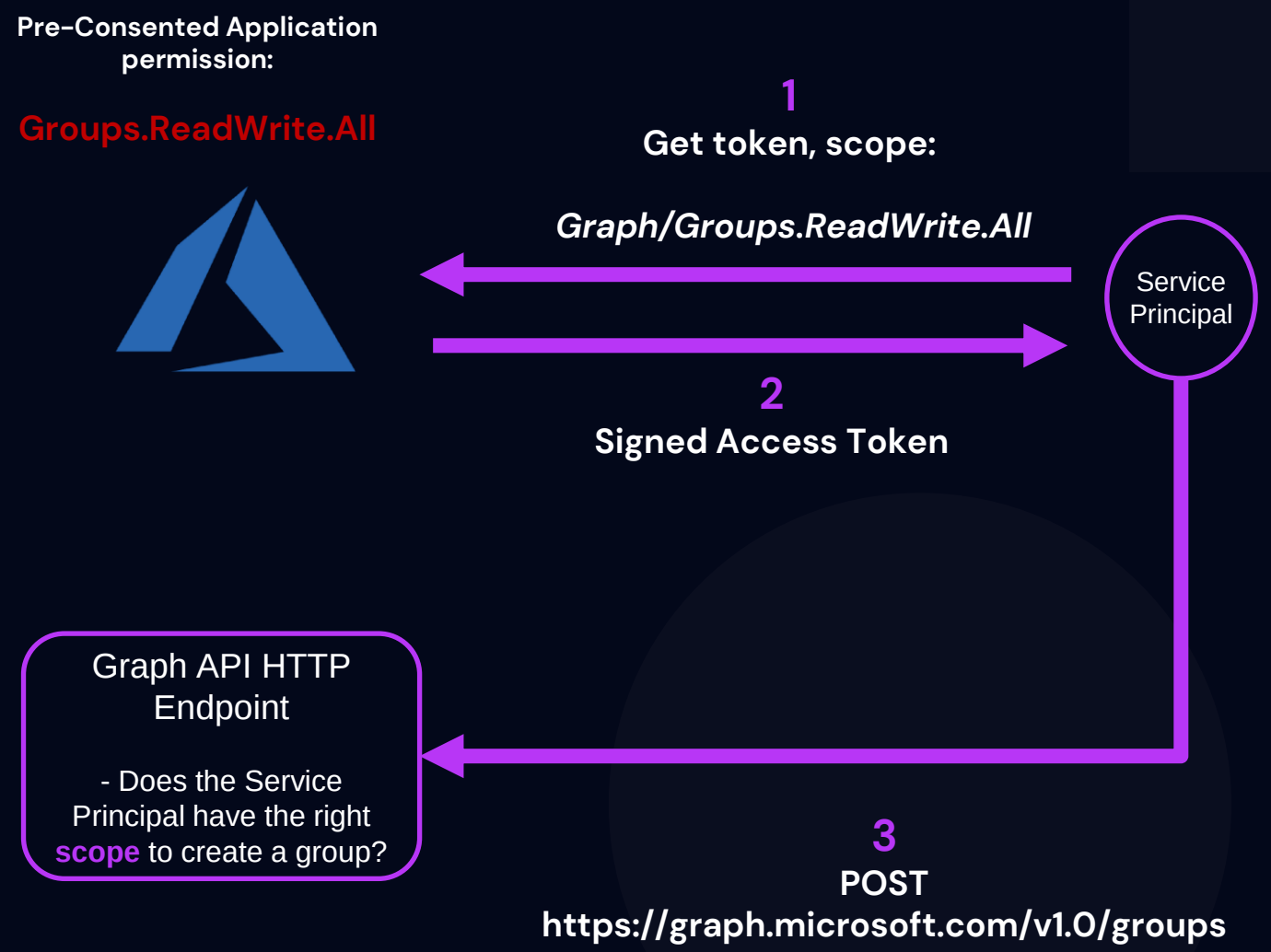
What about machine to API?

- Directory Automation?
- Reading directory objects from apps?
- Updating objects such as Groups from a SaaS app?
- Requires a service principle



Service Principle Permissions

Azure AD Auth – Service Principles



How Azure AD Works

In summary

- *Most* management requests now use *https://graph.microsoft.com* APIs
- To modify the directory, must authenticate to MS Graph APIs using some directory object
- Two authorization mechanisms that allows you to modify directory objects
  - Azure AD roles
  - Application permissions with service principals

How can you abuse Azure AD?

Initial Access - attacks

- Illicit Consent Grant Attacks?
- Password Spraying?
- Credential Stuffing?
- General password brute forcing?
- Traditional phishing tactics – fake login page?
- Steal the tokens from browsers of victims (after initial compromise)?
- Dump and steal Primary Refresh Tokens (after initial compromise)?
- Leaked Service Principal Secret?

How can you abuse Azure AD?

Initial Access - attacks

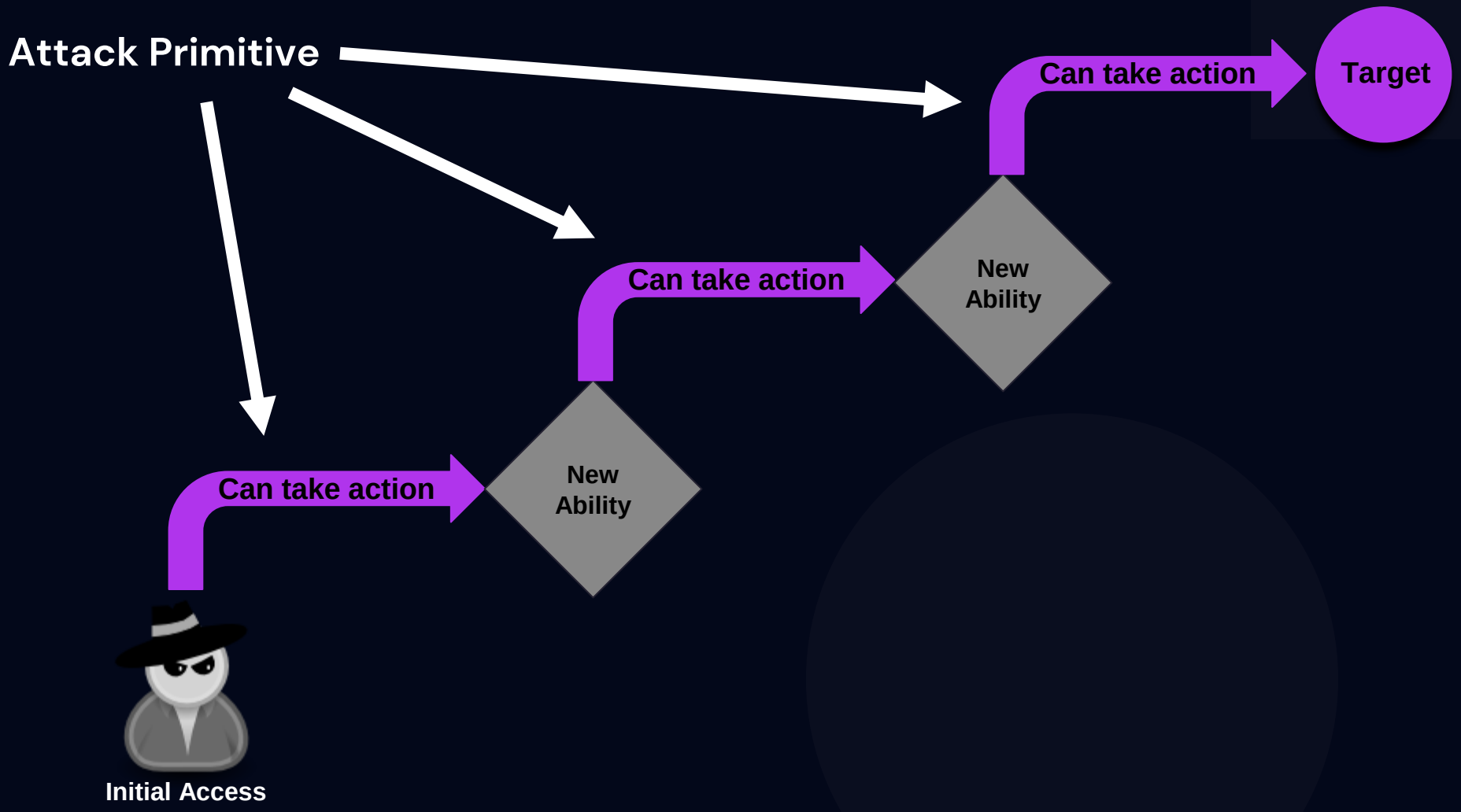
- Illicit Consent Gra… …ks?
- Password Spraying?
- Credential Stuffing?
- General password b…
- Traditional phis… …rake
- Steal the tokens fr… …owsers of victi… …fter initial compromise)?
- Dump and steal Primary Refresh Tokens (after initial compromise)?
- Leaked Service Principal Secret?

Not what this presentation is about

Azure AD attack primitives

The basis of attack path analysis

What is an “attack primitive”?



Attack Primitives – definition

"A configuration state that may be abusable under certain conditions, but that is not a vulnerability"

Attack Primitives – definition

- Allows the attacker to “change state” in one way or another
  - Privilege Escalation
  - Lateral Movement
  - Abuse technique

- Building blocks of an attack path

Speed Round

Basic primitives

Abuse Azure AD Administrative Roles:

Azure AD Role	Primitive
Application Administrator	Assume Privileges of any Service Principle
Privileged Auth Administrator	Assign new Global Administrators
Groups Administrator	Assign self to privileged group
Hybrid Identity Administrator	Assign self as Owner to Application

Abuse object ownership:

Object Ownership	Primitive
Application Object	Assume Privileges of the Application (multi tenant)
Service Principal	Assume Privileges of the Service Principle (single tenant)
Group	Assign self to group

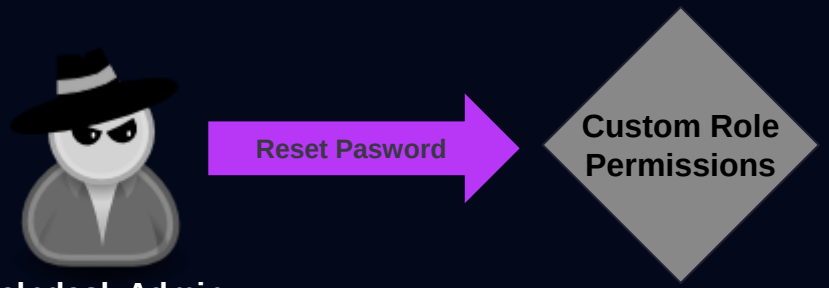
Basic primitives - Applications

Abuse Application Objects and Service Principals:

Graph Permission	Primitive
Application.ReadWrite.All	Assume Privileges of any Service Principle
DirectoryRole.ReadWrite.All	Assign new Global Administrators
Group.ReadWrite.All	Assign self to privileged group

Password reset primitives

Who can reset who's password?

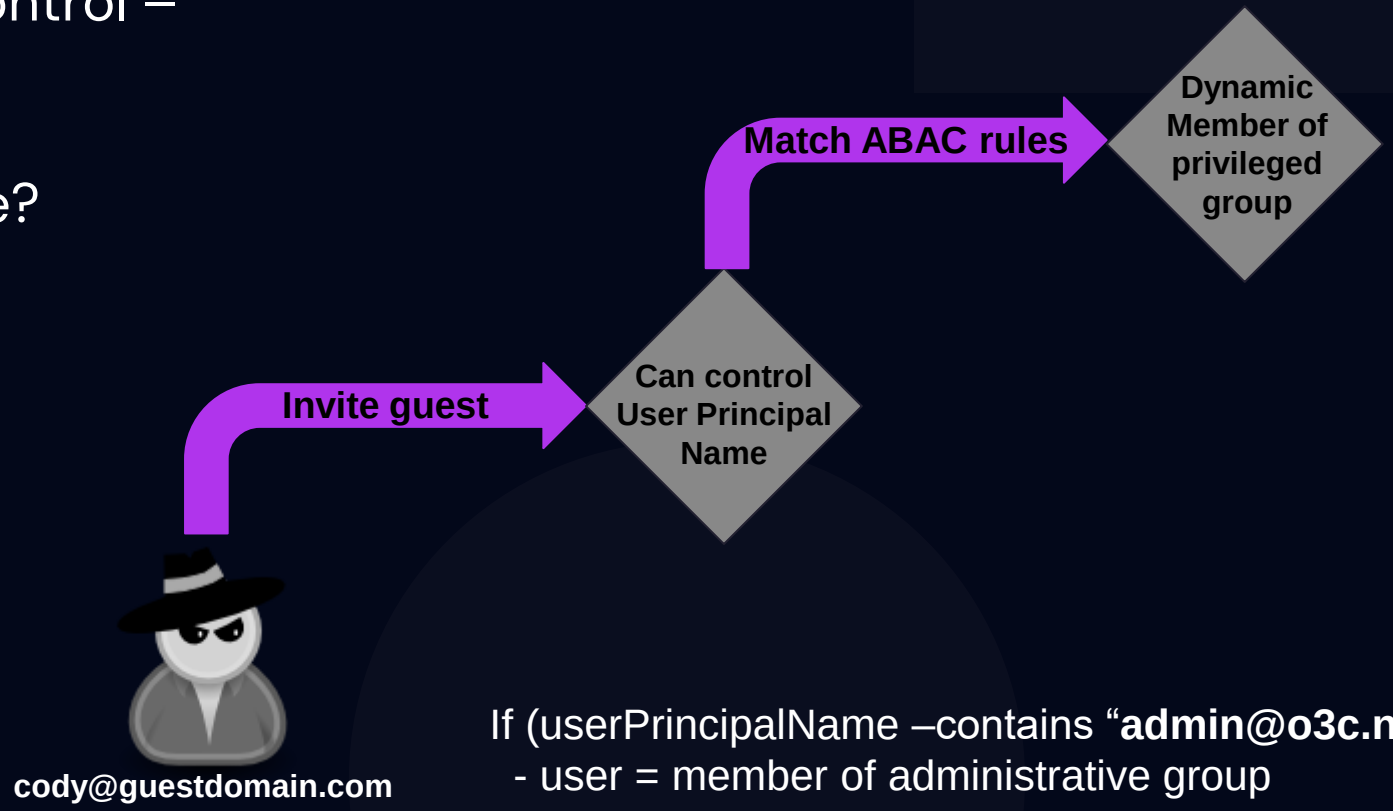


Role that password can be reset	Password Admin	Helpdesk Admin	Auth Admin	User Admin	Privileged Auth Admin	Global Admin
Auth Admin			✓		✓	✓
Directory Readers	✓	✓	✓	✓	✓	✓
Global Admin					✓	✓*
Groups Admin				✓	✓	✓
Guest Inviter	✓	✓	✓	✓	✓	✓
Helpdesk Admin		✓		✓	✓	✓
Message Center Reader		✓	✓	✓	✓	✓
Password Admin	✓	✓	✓	✓	✓	✓
Privileged Auth Admin					✓	✓
Privileged Role Admin					✓	✓
Reports Reader		✓	✓	✓	✓	✓
User (no admin role)	✓	✓	✓	✓	✓	✓
User (no admin role, but member or owner of a role-assignable group)					✓	✓
User Admin				✓	✓	✓
Usage Summary Reports Reader		✓	✓	✓	✓	✓
All custom roles	✓	✓	✓	✓	✓	✓

https://learn.microsoft.com/en-us/azure/active-directory/roles/permissions-reference#who-can-reset-passwords

ABAC primitives

- Attribute-based access control – new attack surface!
- Who controls the attribute?
  - Dynamic Group Abuse
  - Azure ABAC abuse
    - *coming soon*

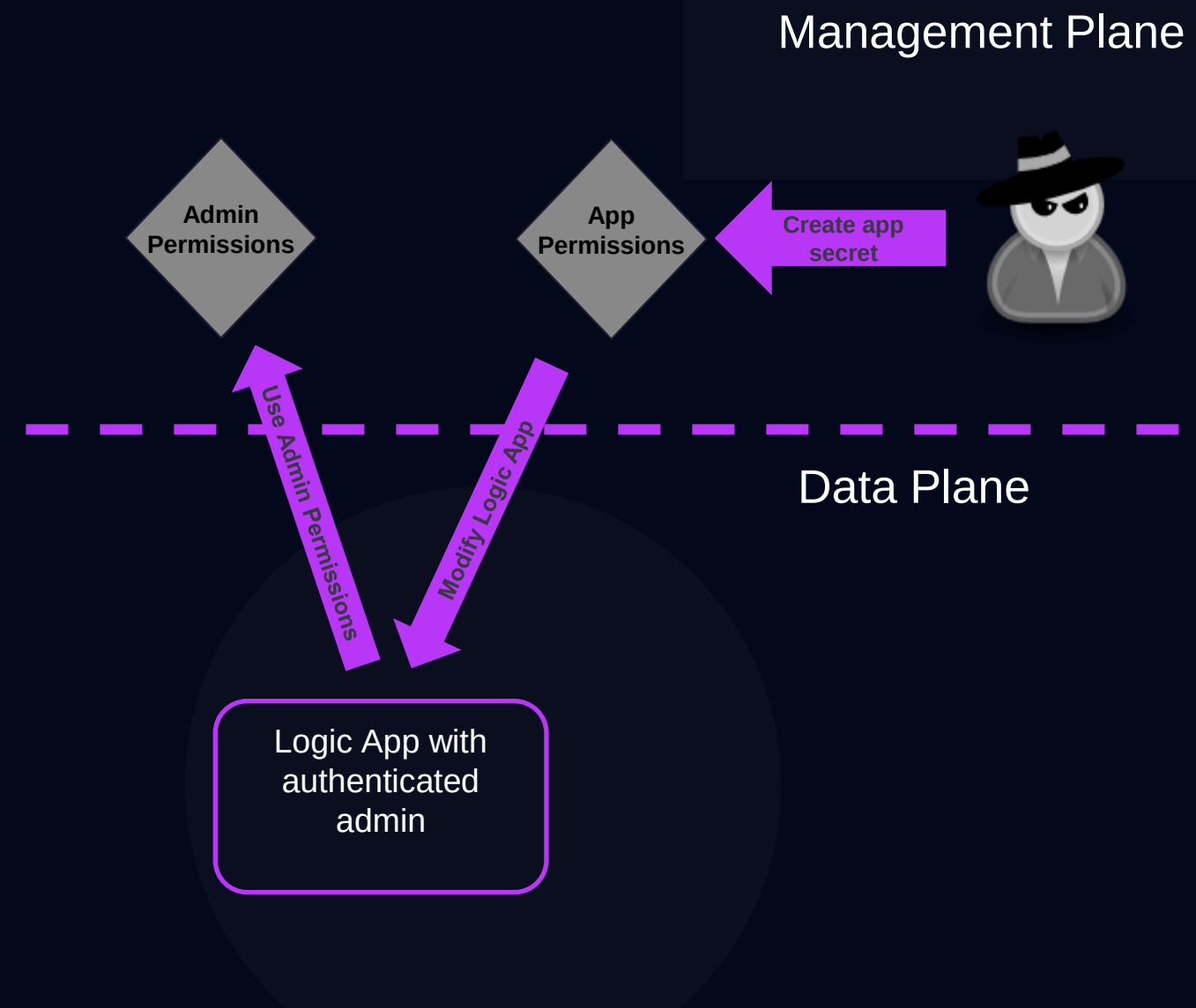


If (userPrincipalName –contains “**admin@o3c.no**”)
- user = member of administrative group

Invite **admin@o3c.no**.maliciousdomain.com

Data Plane “Bouncing”

- Secrets in Plaintext
- Cloud Shell Abuse
- Key Vault Secrets
- Automation Account Abuse
- Logic App Abuse
- Managed Identity Abuse



Application-level primitives

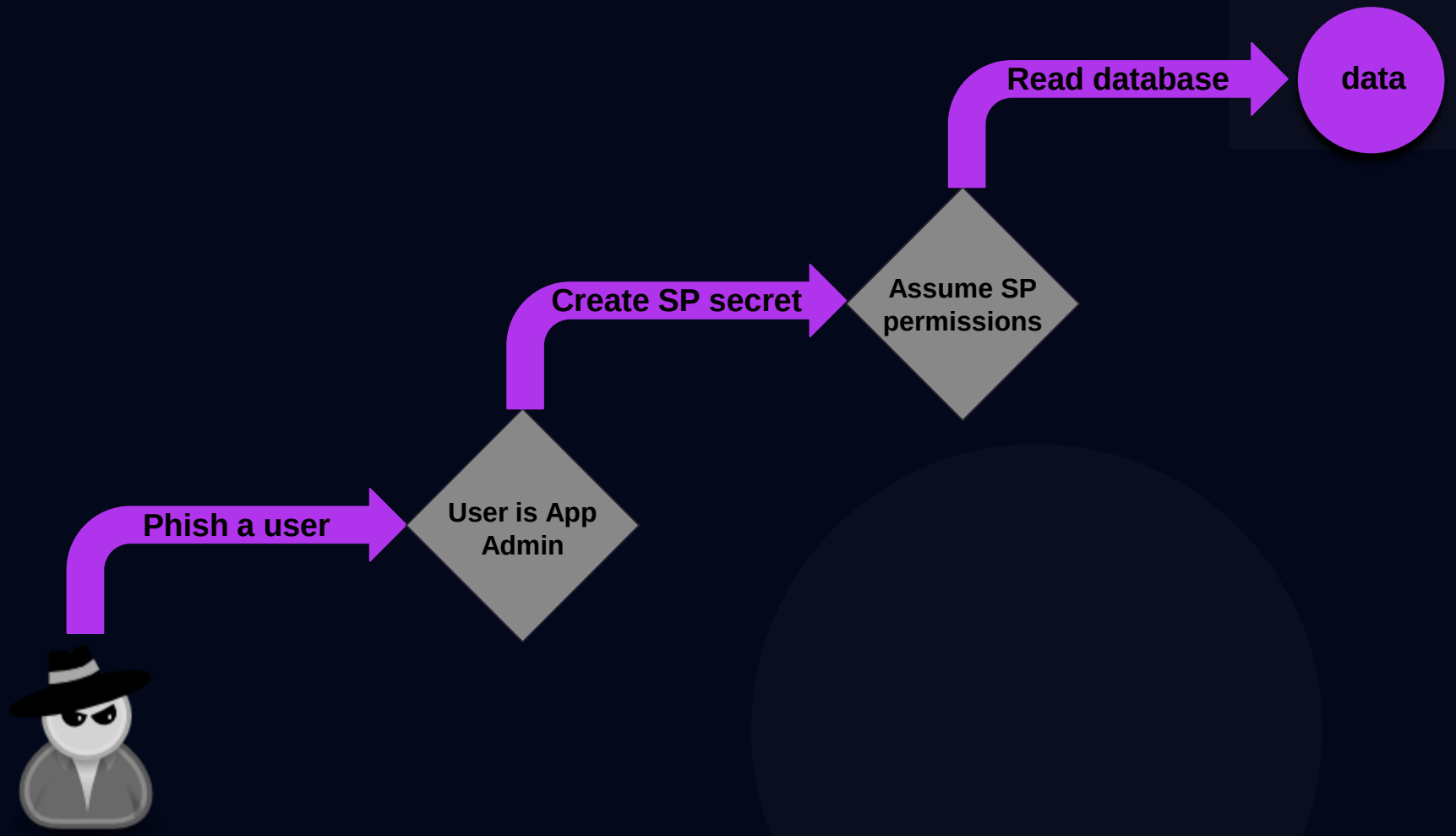
- Abuse admin portals on SaaS apps
- Abuse application OAuth vulnerabilities
  - Poor validation of tokens
  - Improper validation of scope or roles
- Capture auth codes from Oauth Clients
  - Open Redirects
  - Subdomain takeover of redirectUrl domain

Other interesting primitives

- Self-service signup flow in Azure AD
- Combine with traditional on-premise AD attack paths
  - Global admin in Azure may be a normal user on prem
- Bypass PIM with refresh tokens

Putting them together

Azure AD Attack Paths – example 1



Azure AD Attack Paths – example 2

Self service signup

Low-privilege Azure AD User

Dynamic Group gives access to SaaS App

Abuse Dynamic Group

Membership SaaS App Group

SaaS app has *Groups.ReadWrite.*All on Graph API

Abuse SaaS permissions

Added self to Azure admin group

Admin group is a contributor on a resource with a Managed Identity

Managed identity abuse

Access Cloud Shell Storage Account

Managed Identity can modify data in a cloud shell storage account

Cloud Shell Abuse

Global Admin permissions

Global Admin had file share in Cloud Shell Storage

Directory Role Assignment

Global Admin

Attacker assigns self to Global Admin role

Closing Thoughts

- ***Just because a primitive exists in your infra doesnt mean there is a security issue***
- «IAM hygiene» does help to eliminate *unnecessary* attack paths
- «IAM hygiene» will not prevent attack paths from being introduced
  - This is an entirely different topic
- Attack path analysis or testing may be worth including in IAM routines
- Research may yield interesting results in your unique setups

O3C.NO