SPECTEROPS

It's Raining Shells

How to Find New Attack Primitives in Azure

Permalink to this deck: https://bit.ly/3ItRi6u

Hello!

My name is Andy Robbins

Co-creator of BloodHound
Product Architect of BloodHound Enterprise
I work at @SpecterOps
You can find me at @_wald0

Agenda

Why abuses, not bugs?

Crash Course Through The Basics

Finding New Attack Primitives: MS Graph Case Study

Where to find research ideas

Conclusion

Agenda

Why abuses, not bugs?

Crash Course Through The Basics

Finding New Attack Primitives: MS Graph Case Study

Where to find research ideas

Conclusion

Abuse primitives:

- Generally enjoy a dramatically longer shelf life

Abuse primitives:

- Generally enjoy a dramatically longer shelf life
- Are much cheaper to maintain over time

Abuse primitives:

- Generally enjoy a dramatically longer shelf life
- Are much cheaper to maintain over time
- Exist in almost every instance of a given platform

Abuse primitives:

- Generally enjoy a dramatically longer shelf life
- Are much cheaper to maintain over time
- Exist in almost every instance of a given platform
- Present a notorious challenge to detection engineers

thaddeus e. grugq
@thegrugq

Give a man an 0day and he'll have access for a day, teach a man to phish and he'll have access for life.

11:35 PM · Feb 6, 2015 · Tweetbot for iOS

5,269 Retweets **211** Quote Tweets **8,877** Likes

https://twitter.com/thegrugq/status/563964286783877121

https://twitter.com/mattifestation/status/809105346870460416

Agenda

Why Azure abuses?

Crash Course Through The Basics

Finding New Attack Primitives: MS Graph Case Study

Where to find research ideas

Conclusion

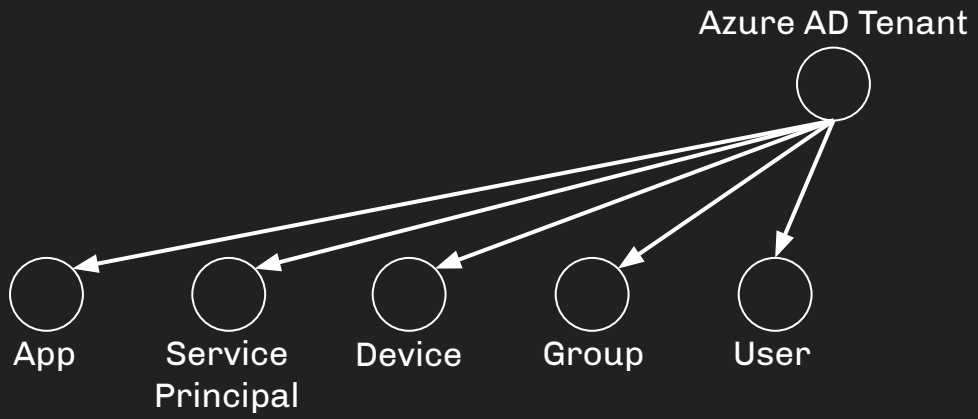
What (exactly) is Azure?

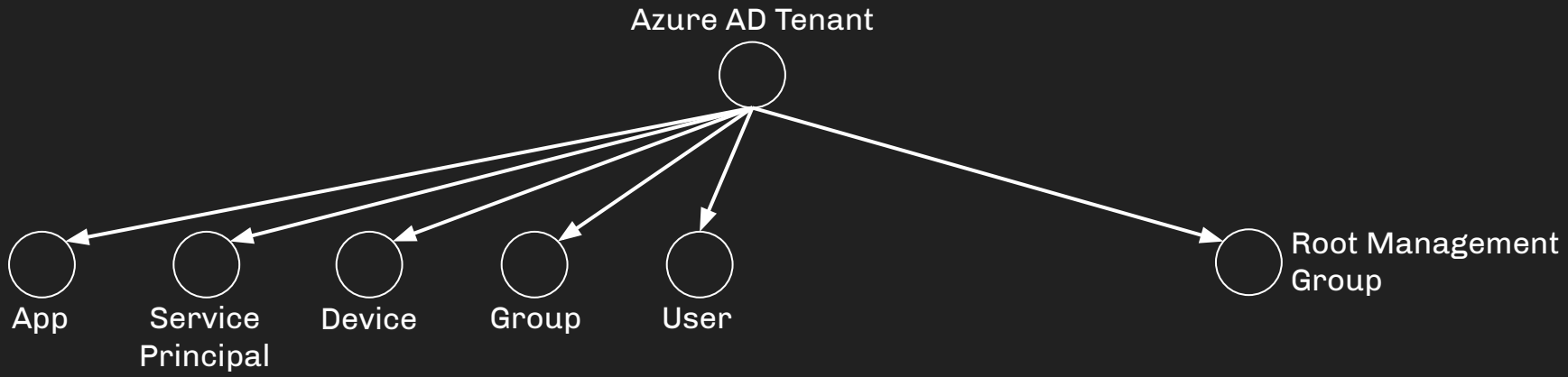
Simply put: Azure is Microsoft's cloud computing product.

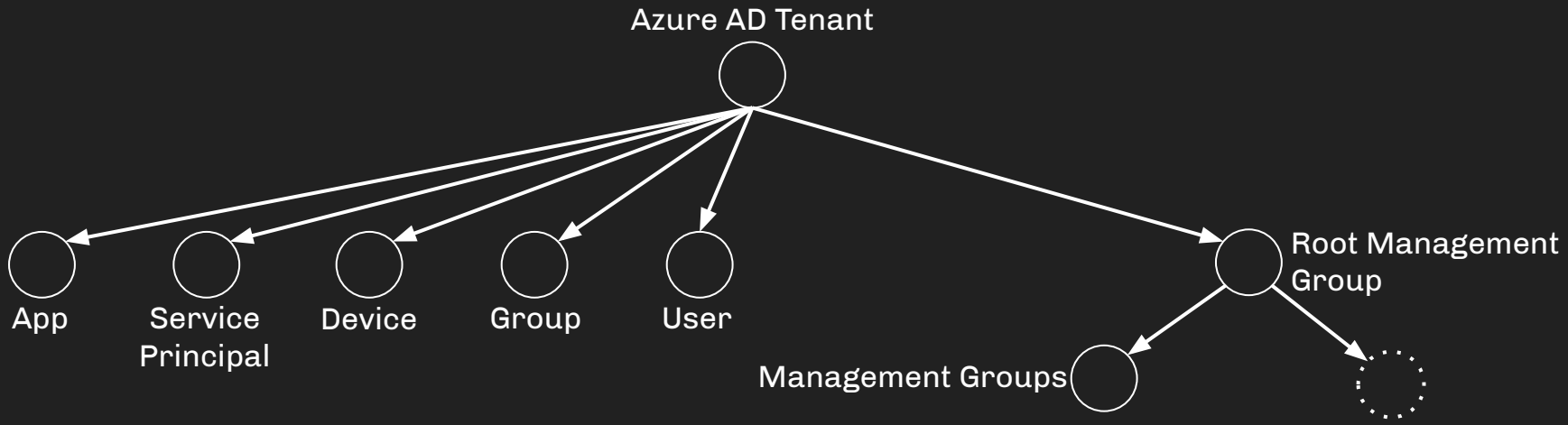
Azure is comprised of more than 600 distinct services that cover:

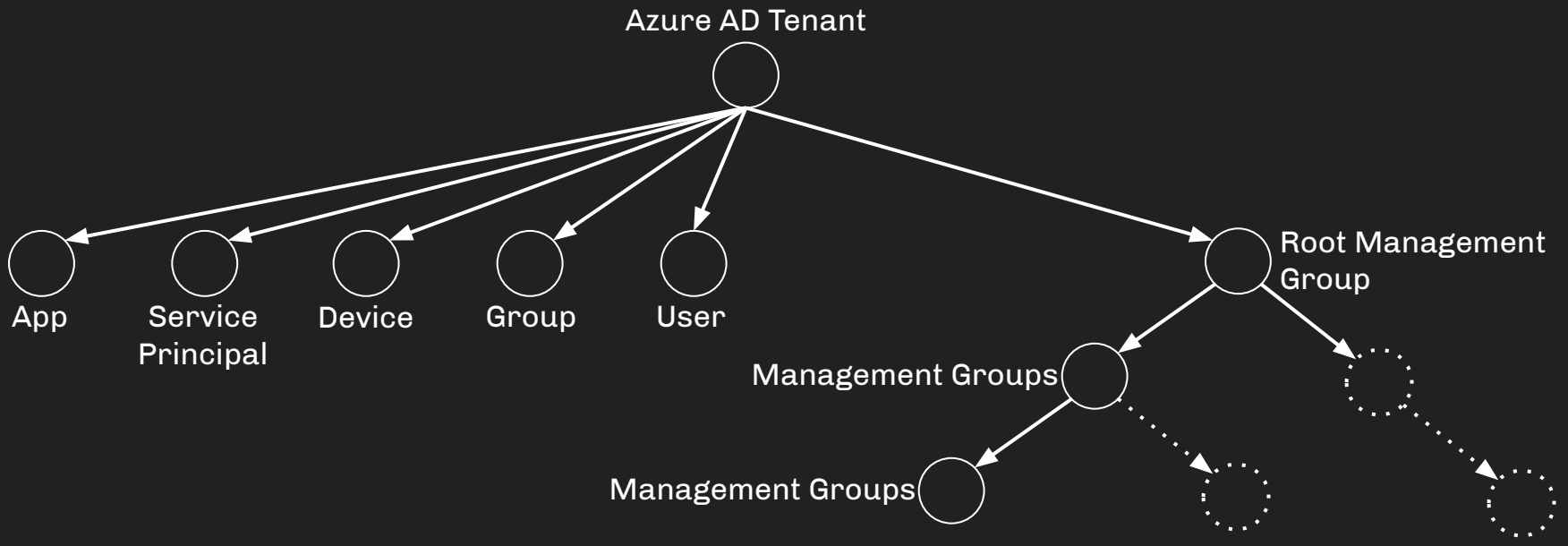
- Identity
- Computing
- Storage
- Data management
- Messaging
- DevOps
- IoT
- etc

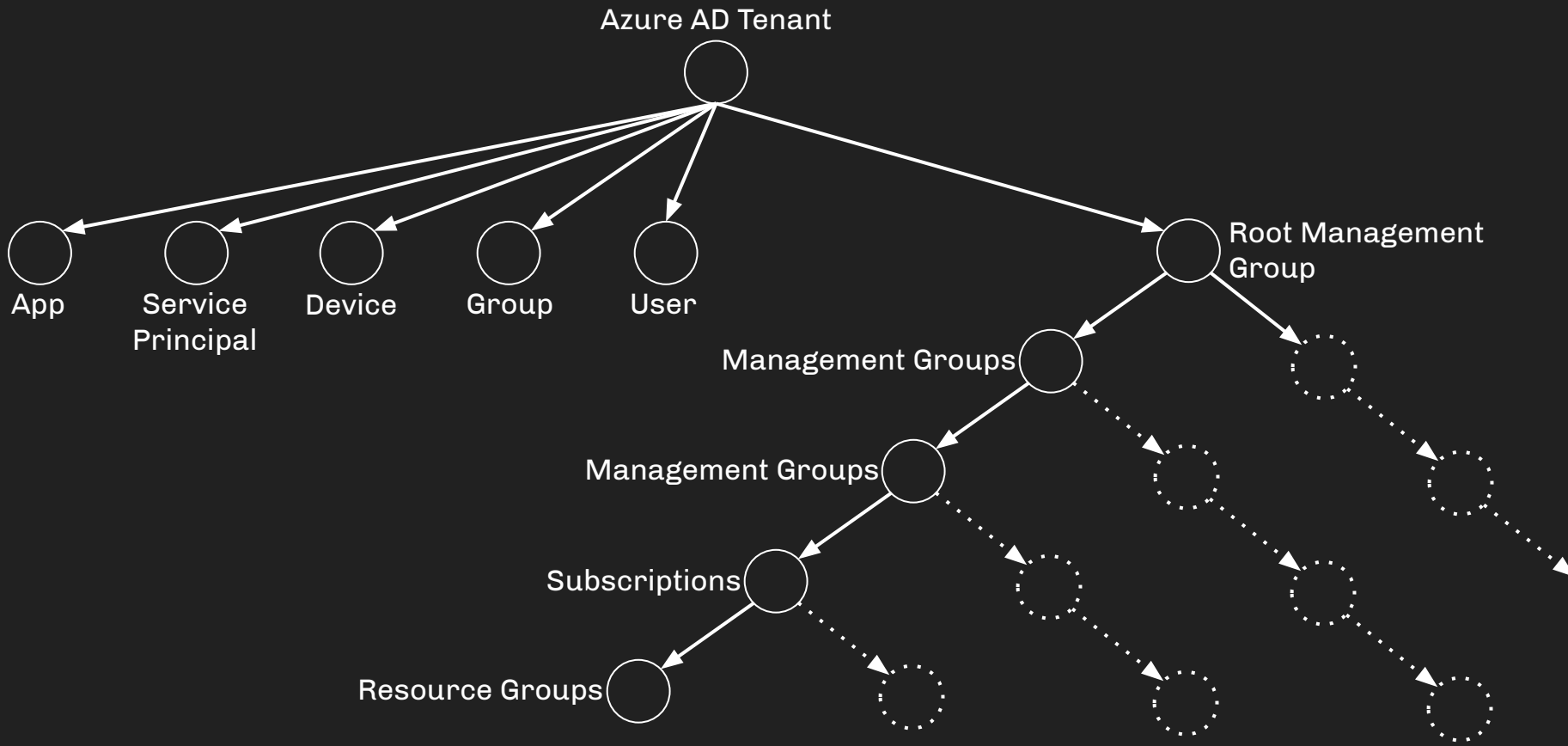
Securable Object Hierarchy

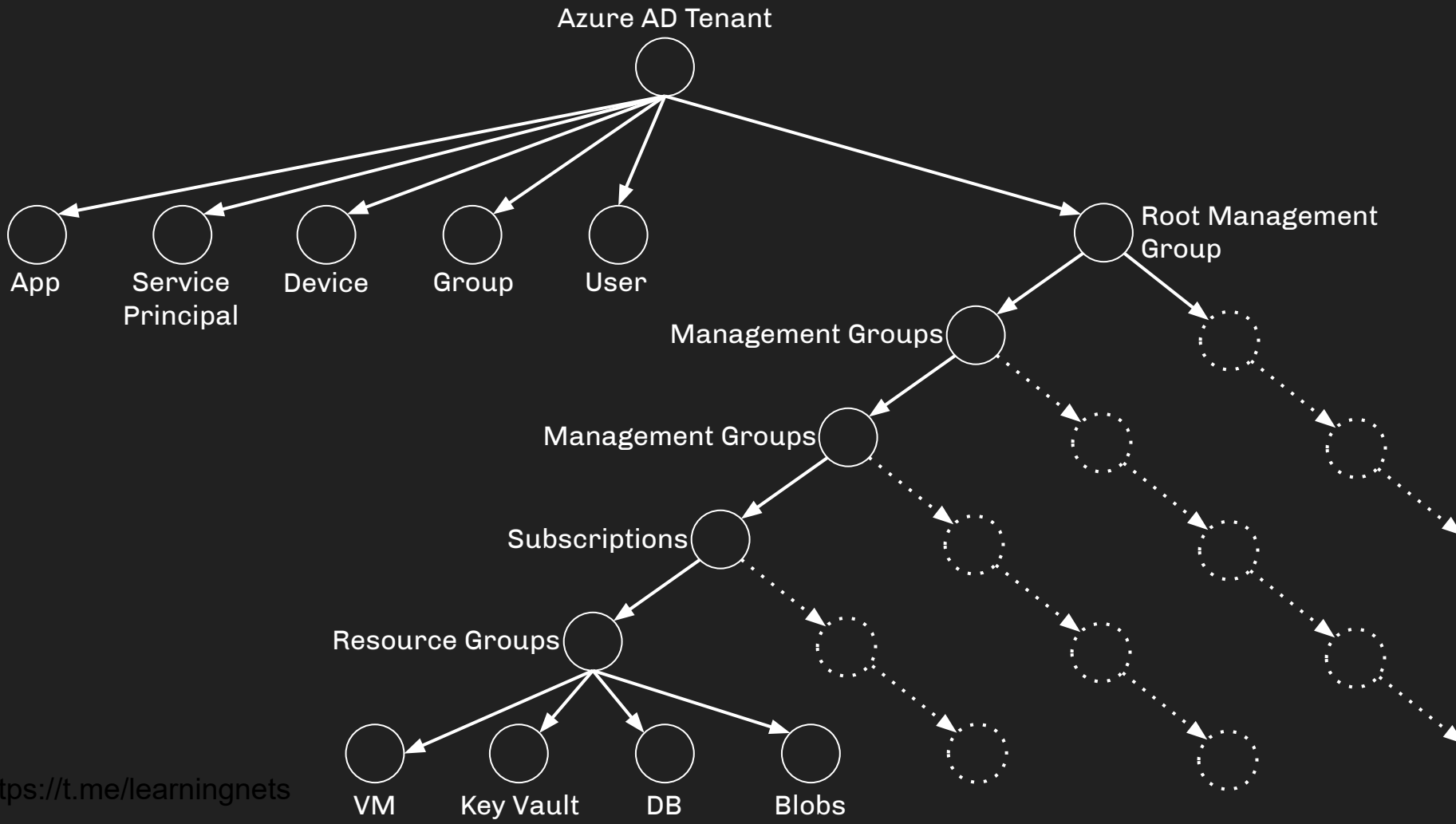




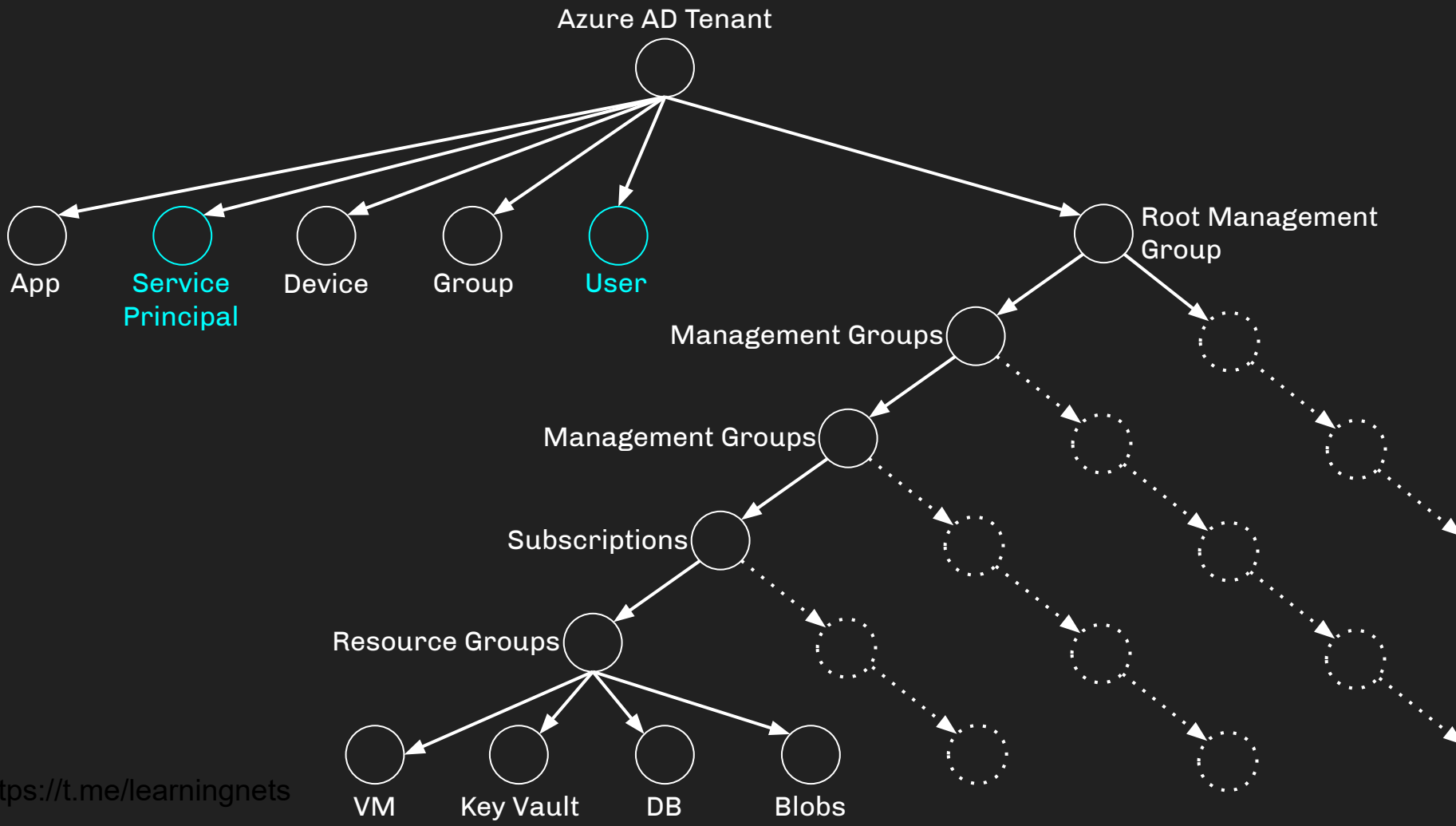




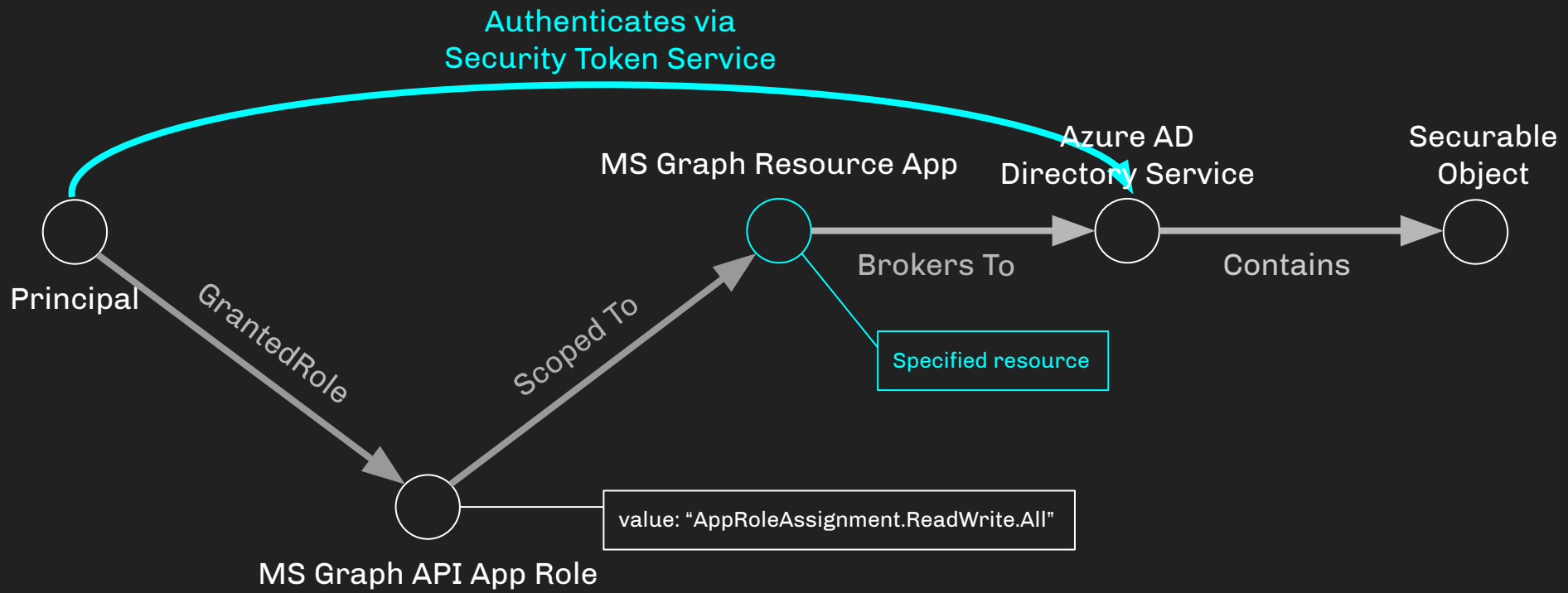




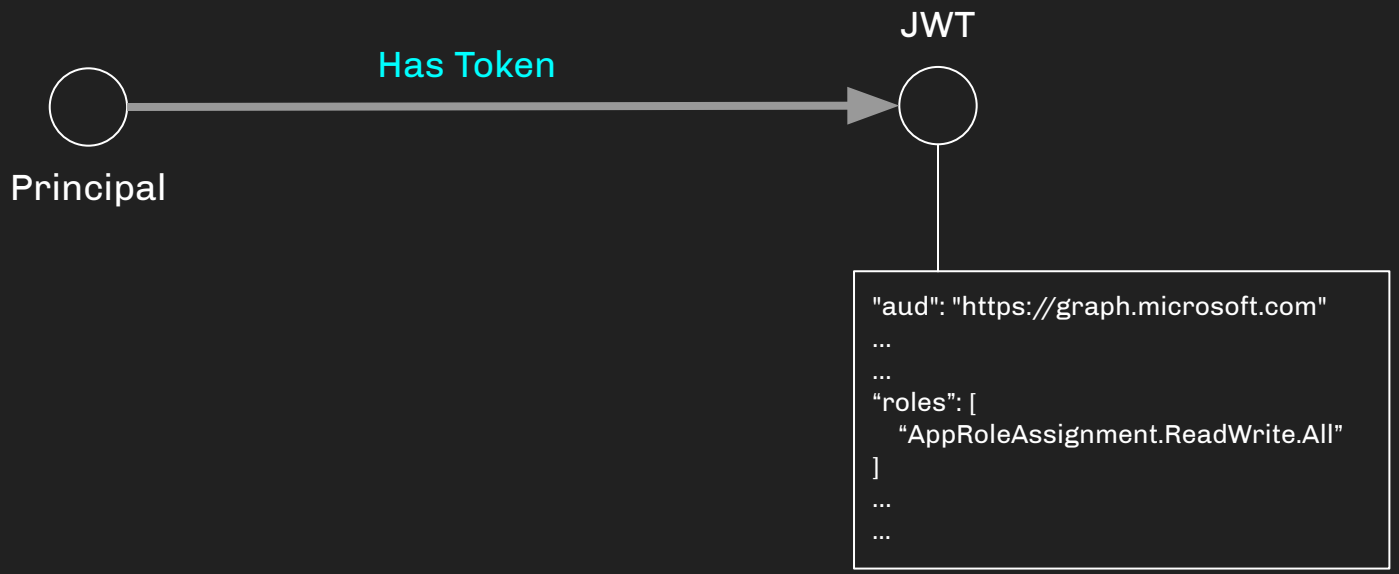
Authentication



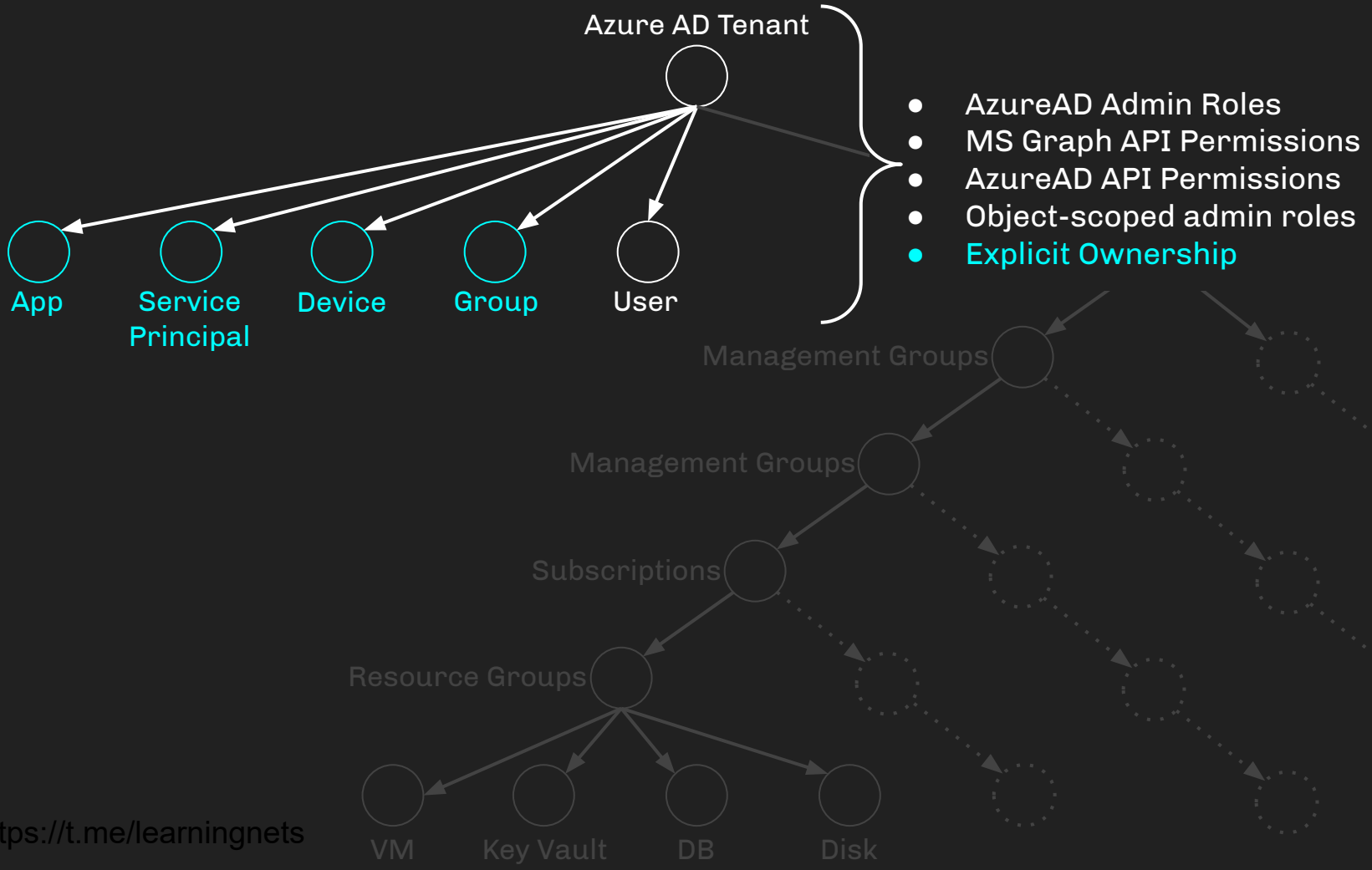
Token issuance

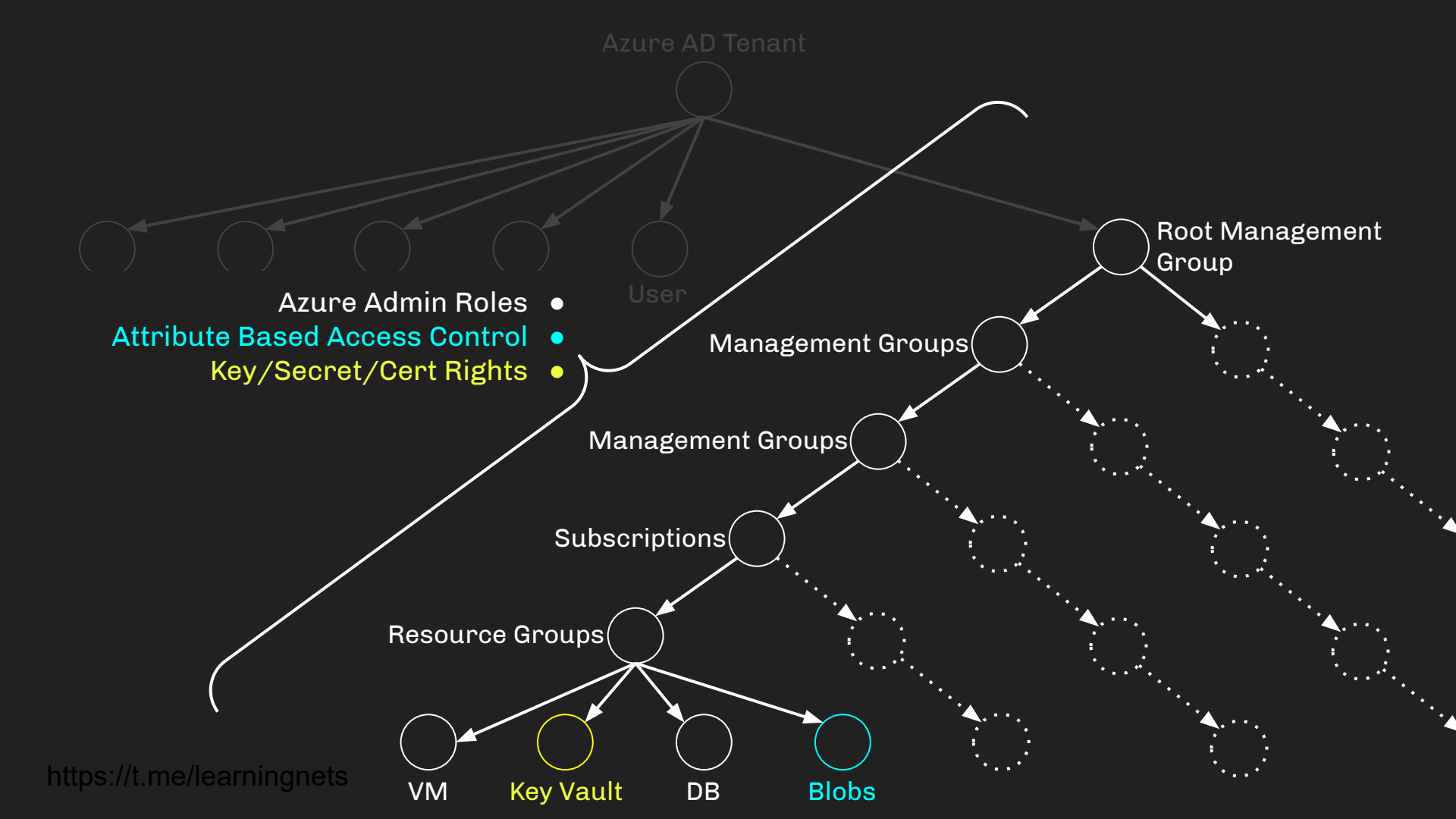


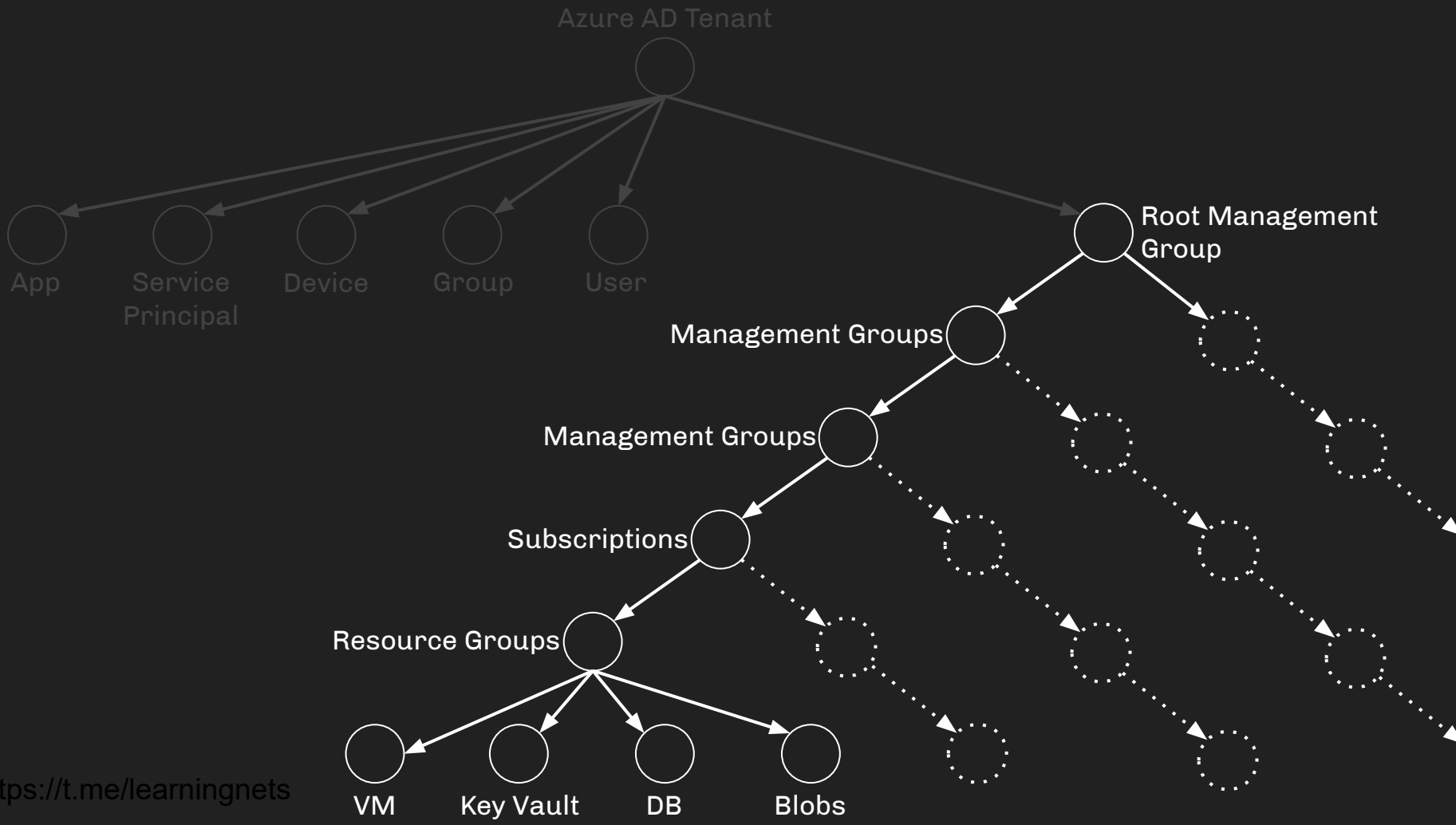
Token issuance

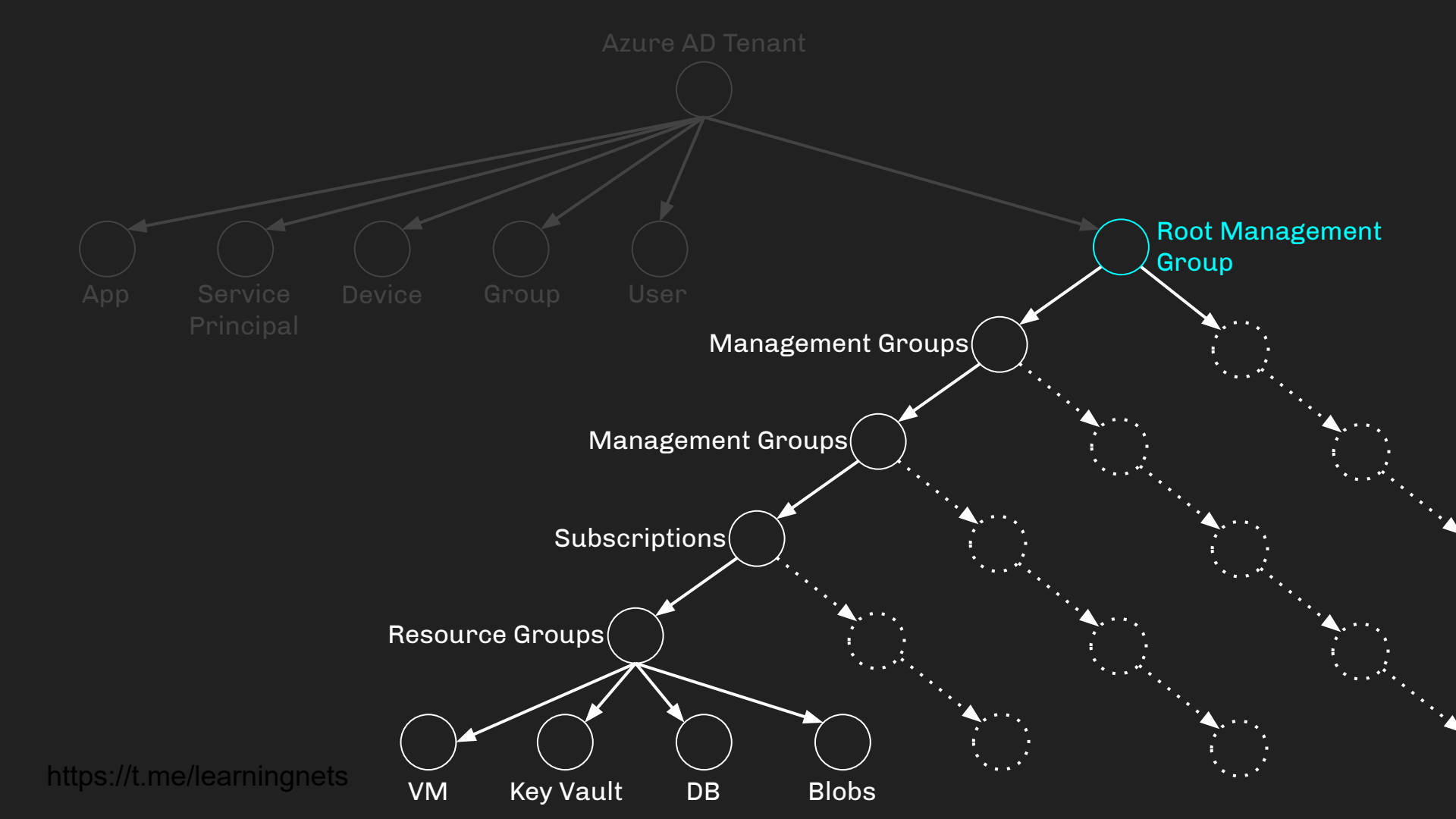


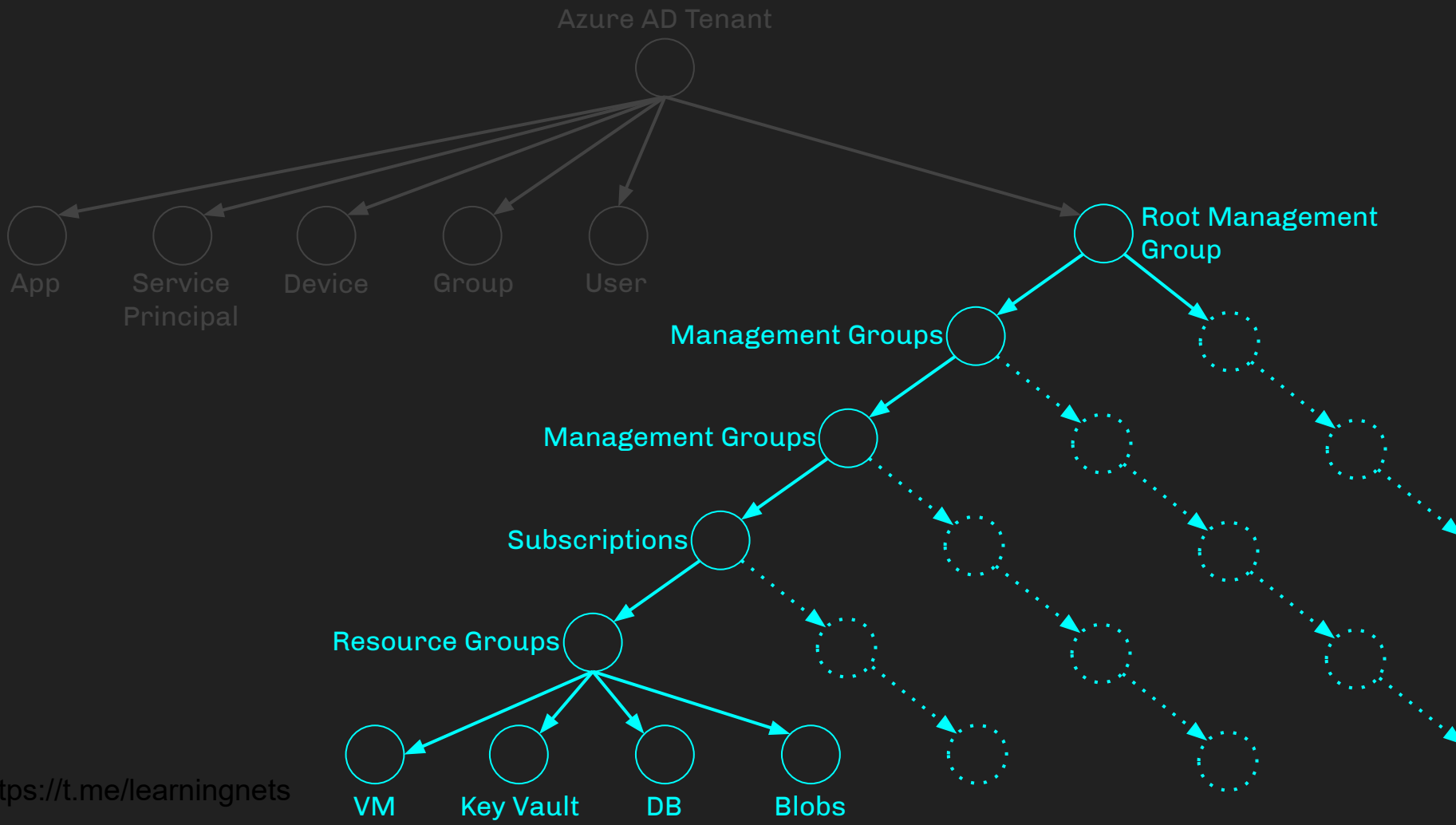
Access Control

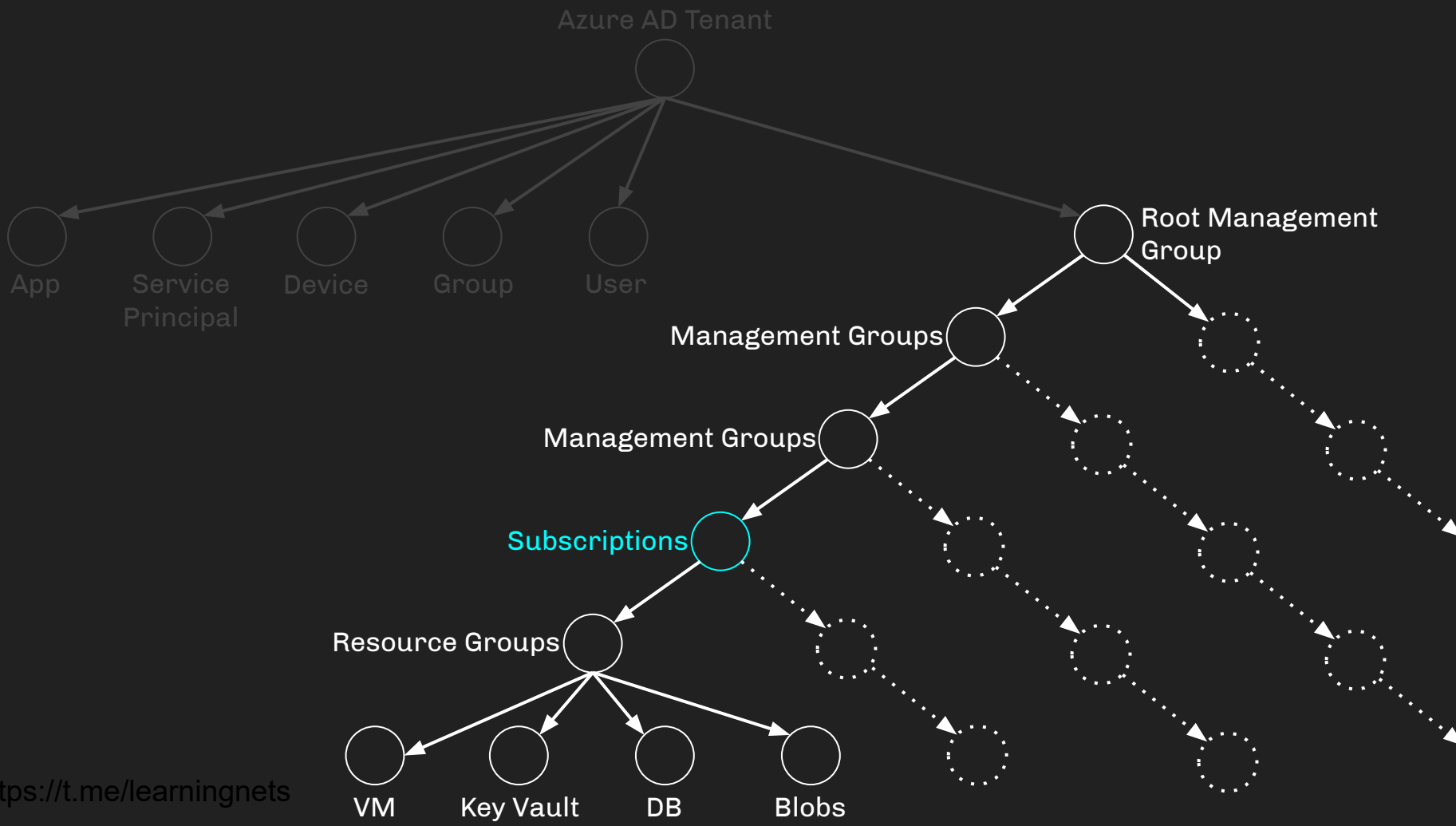


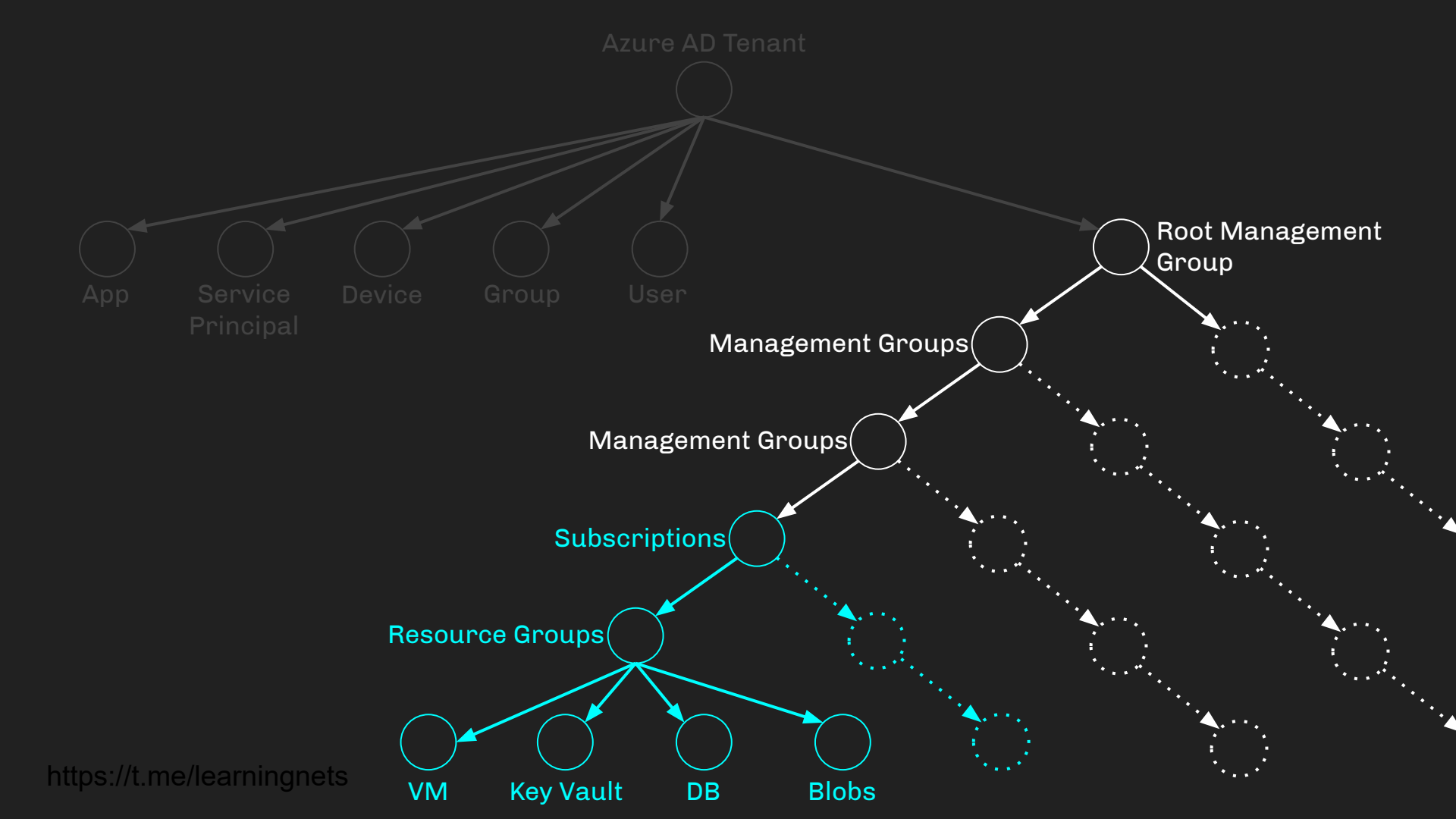


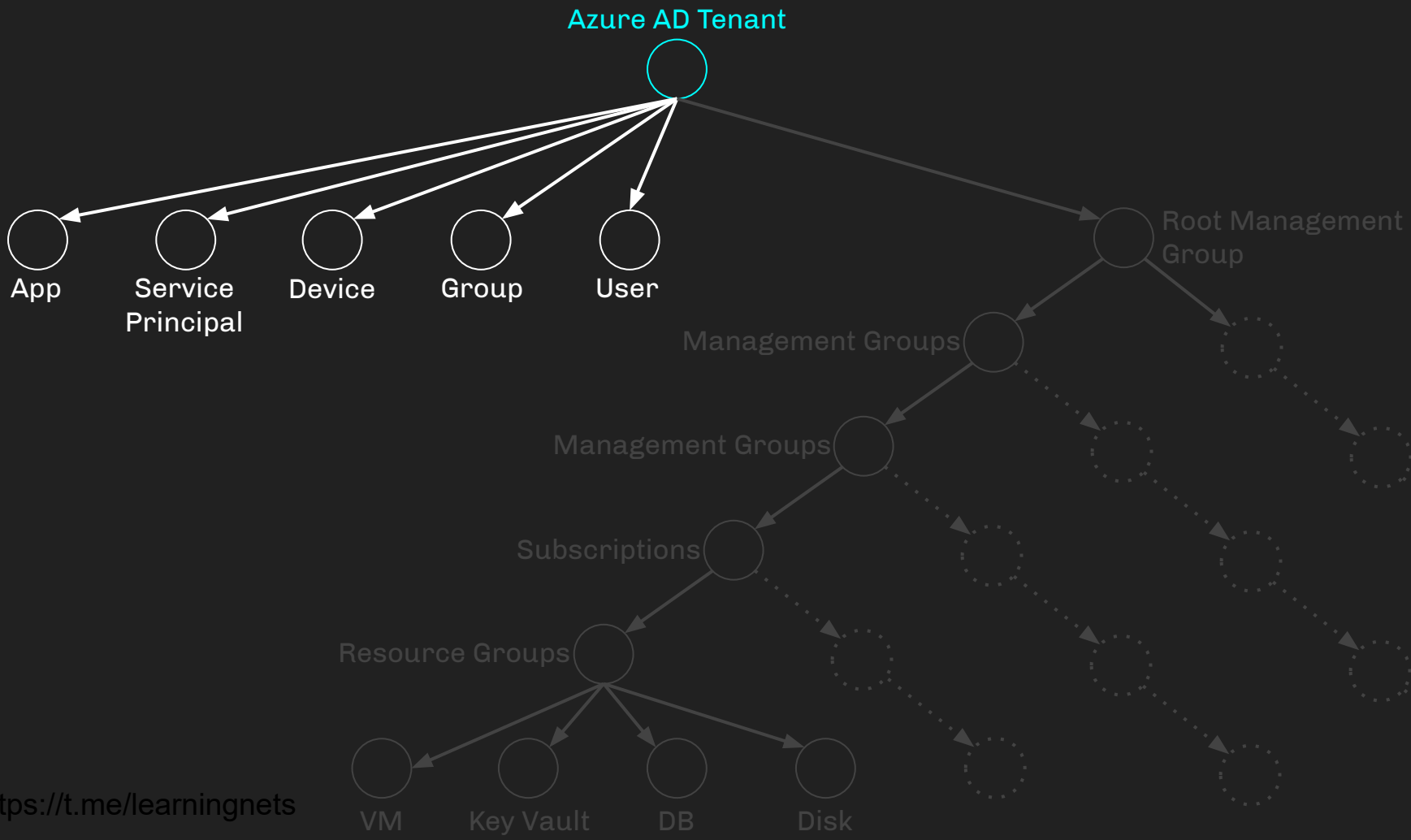


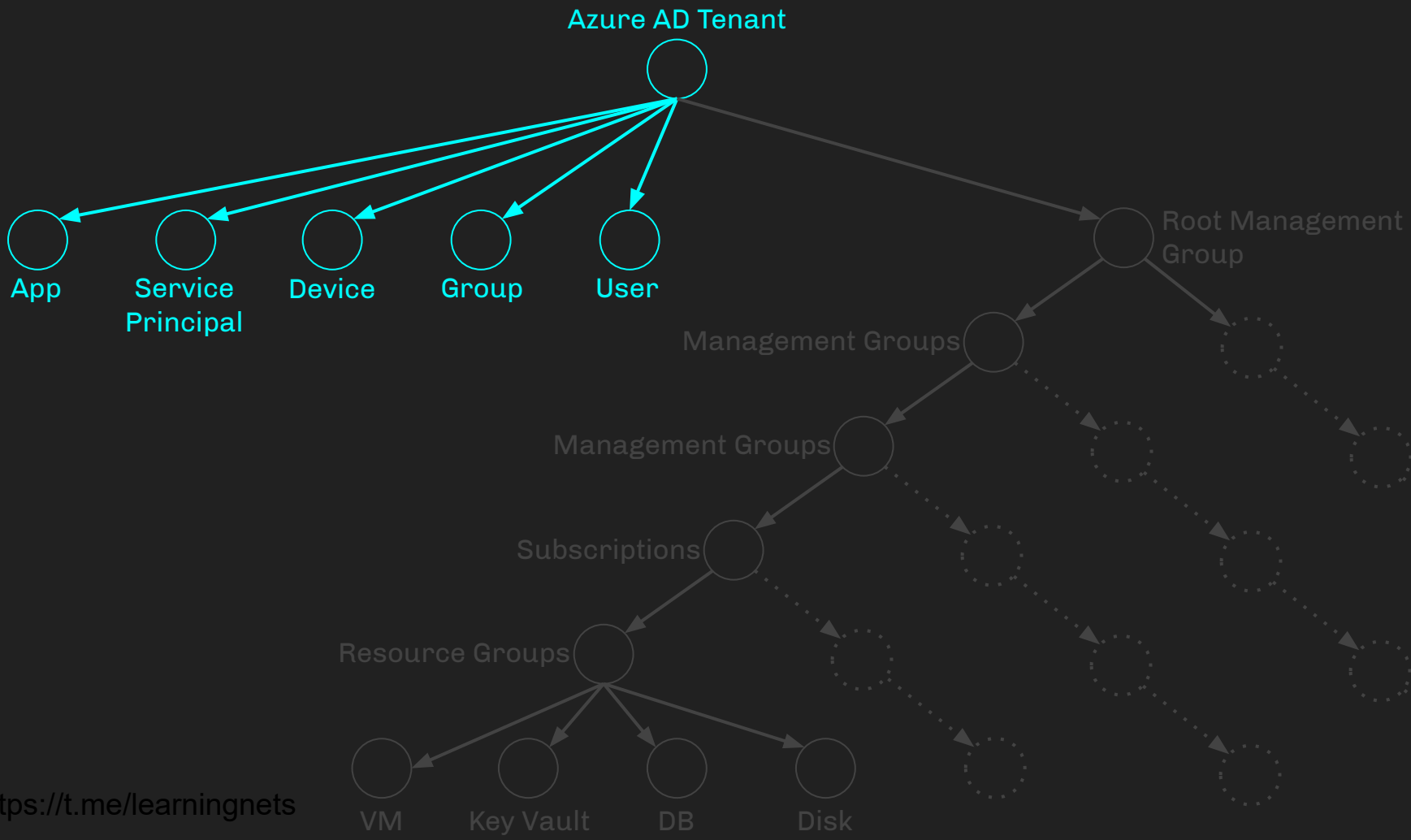


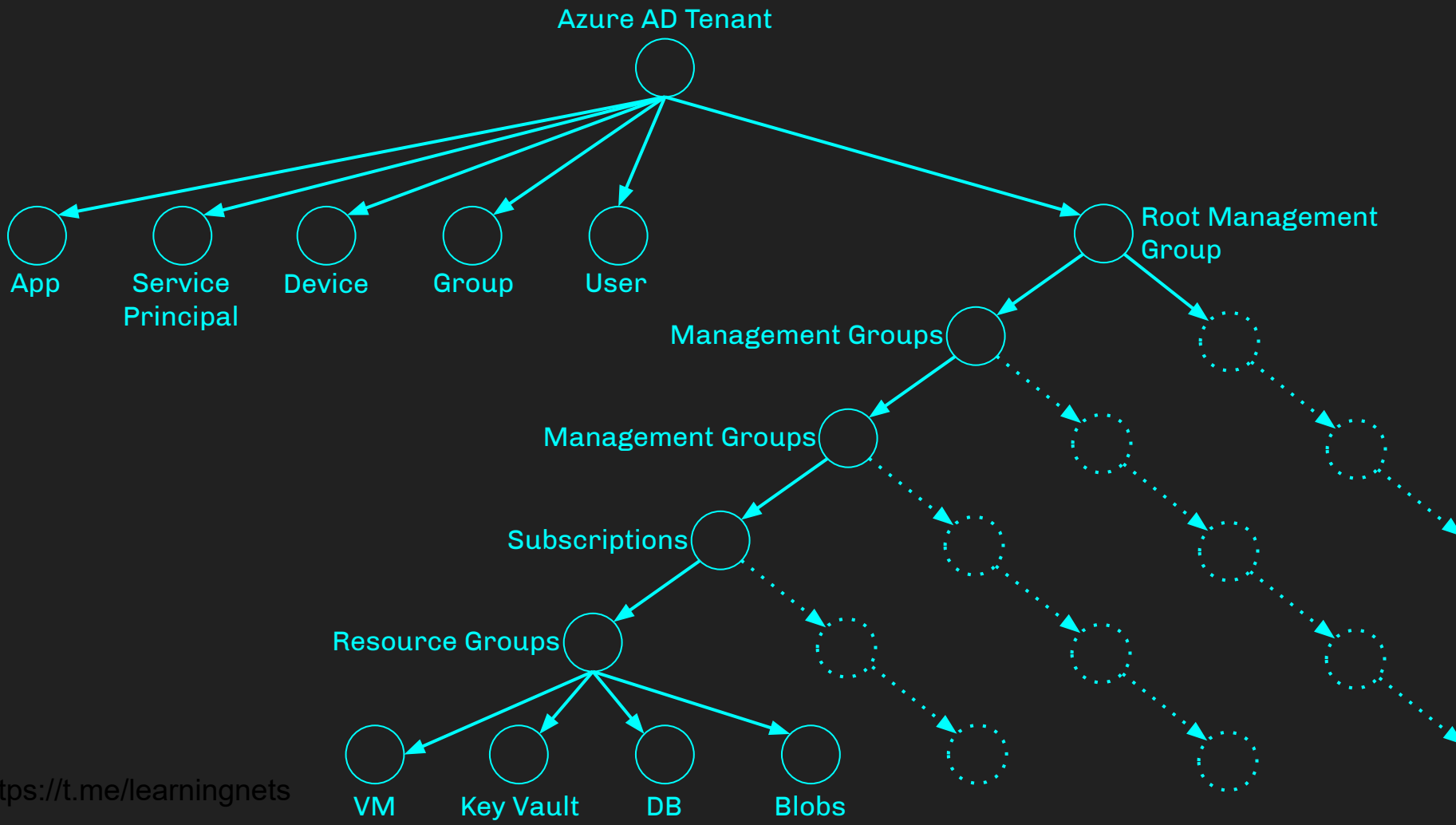












Agenda

Why Azure abuses?

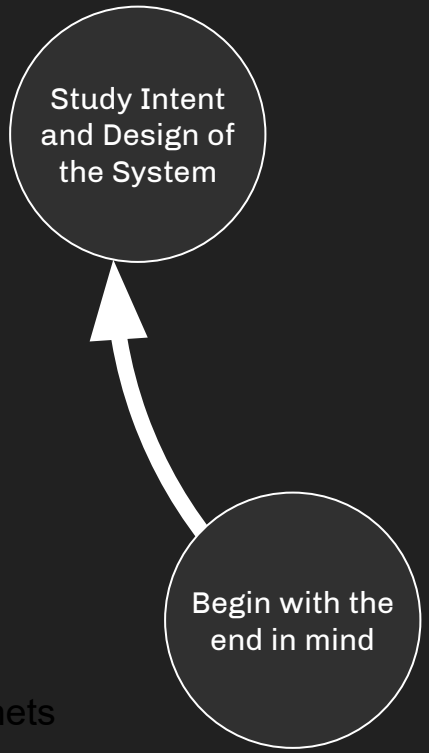
Crash Course Through The Basics

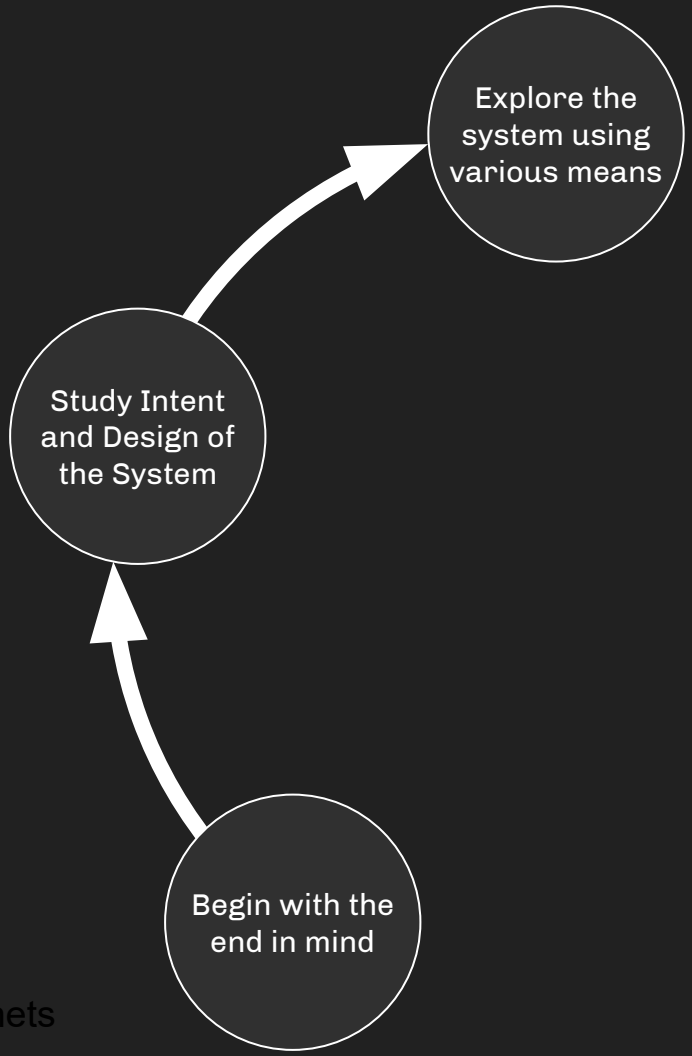
My Abuse Primitive Research Process: MS Graph Case Study

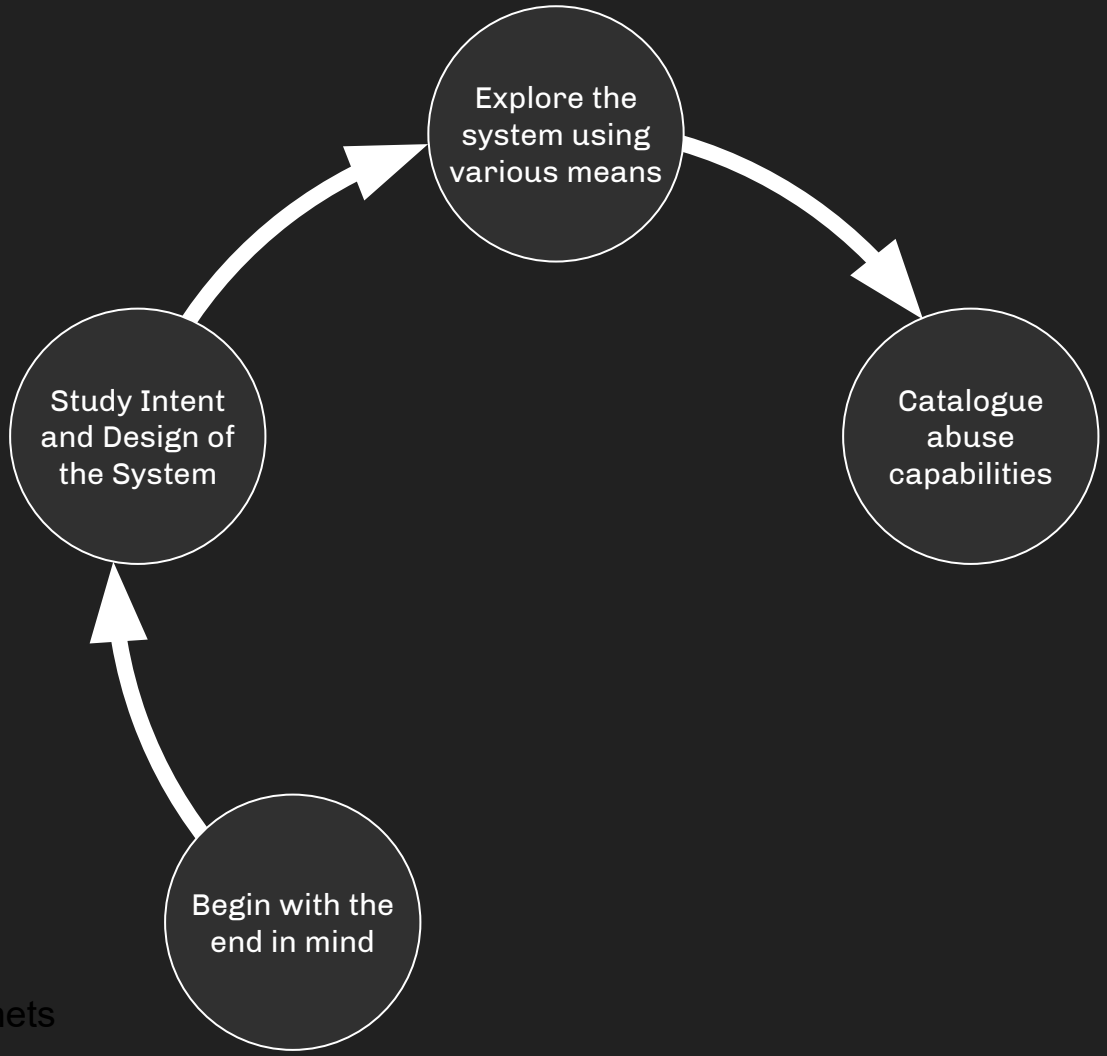
Where to find research ideas

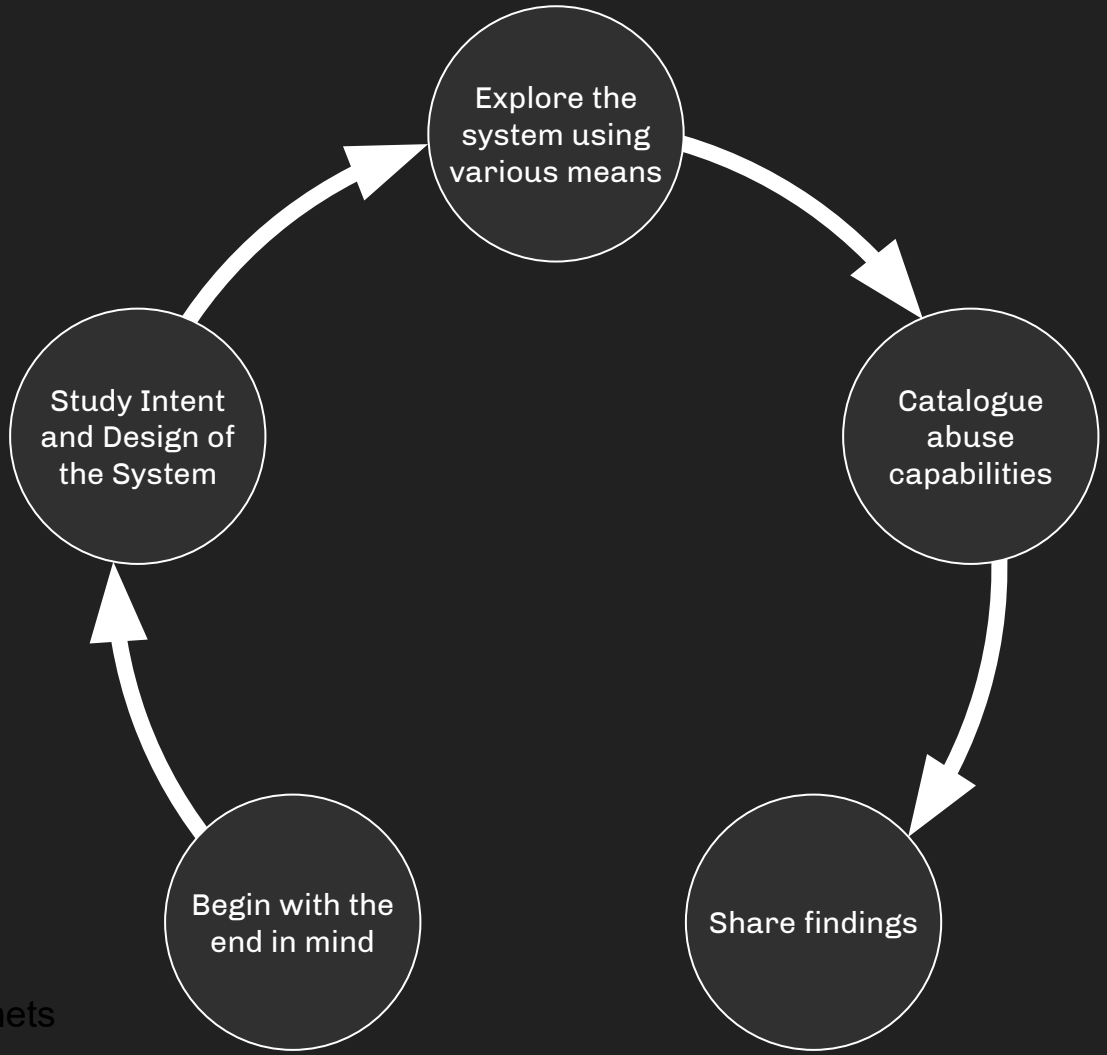
Conclusion

Begin with the end in mind









I want to understand:

- The fundamental mechanics the system
- How the system interacts with other systems
- How the system can be abused

Begin with the end in mind

I want to understand:

- The fundamental mechanics the system
- How the system interacts with other systems
- How the system can be abused

I want to produce:

- A blog/talk for others to understand and build on
- Example audit and abuse code
- Practical remediation guidance

Begin with the end in mind

I want to understand:

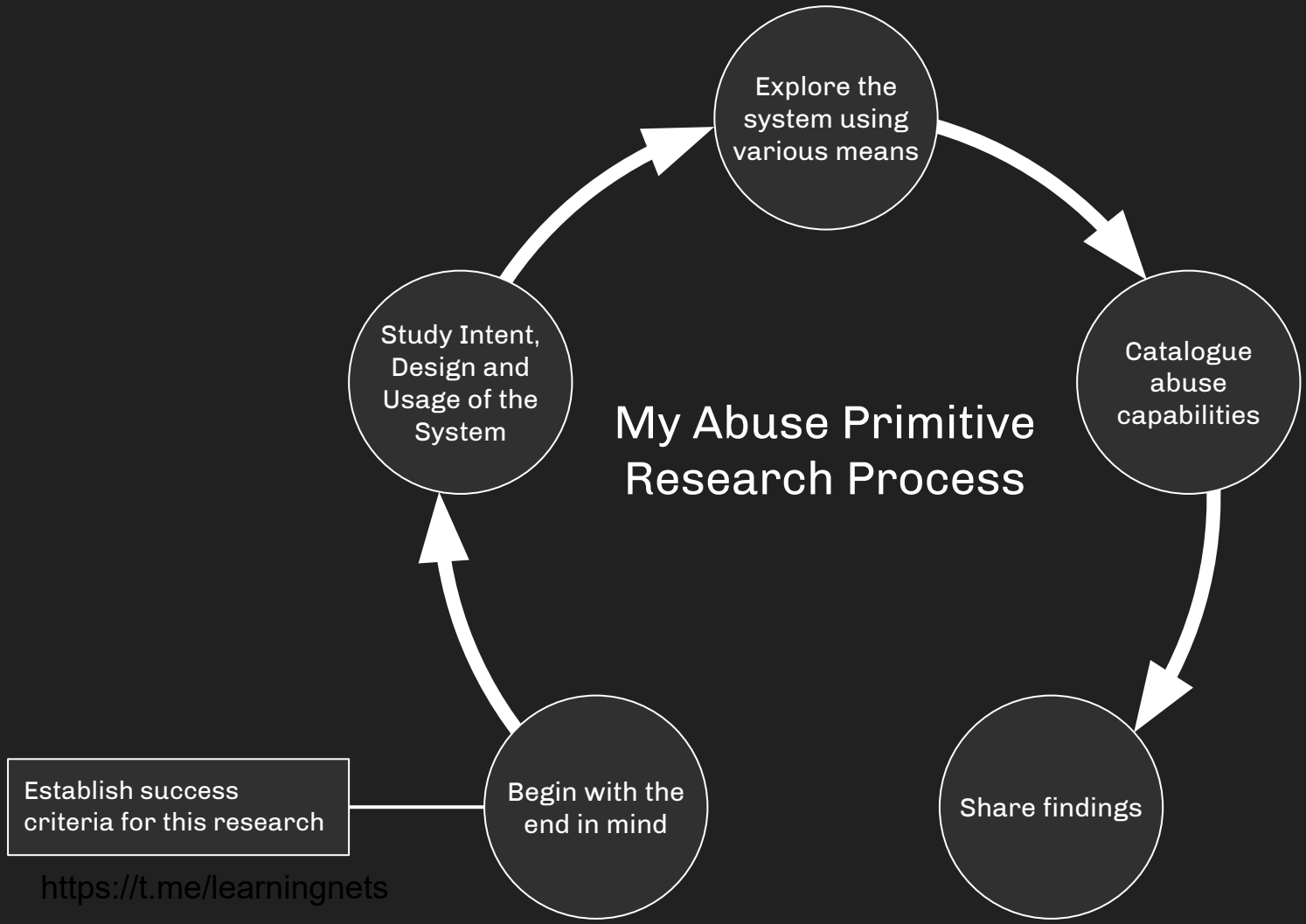
- The fundamental mechanics the system
- How the system interacts with other systems
- How the system can be abused

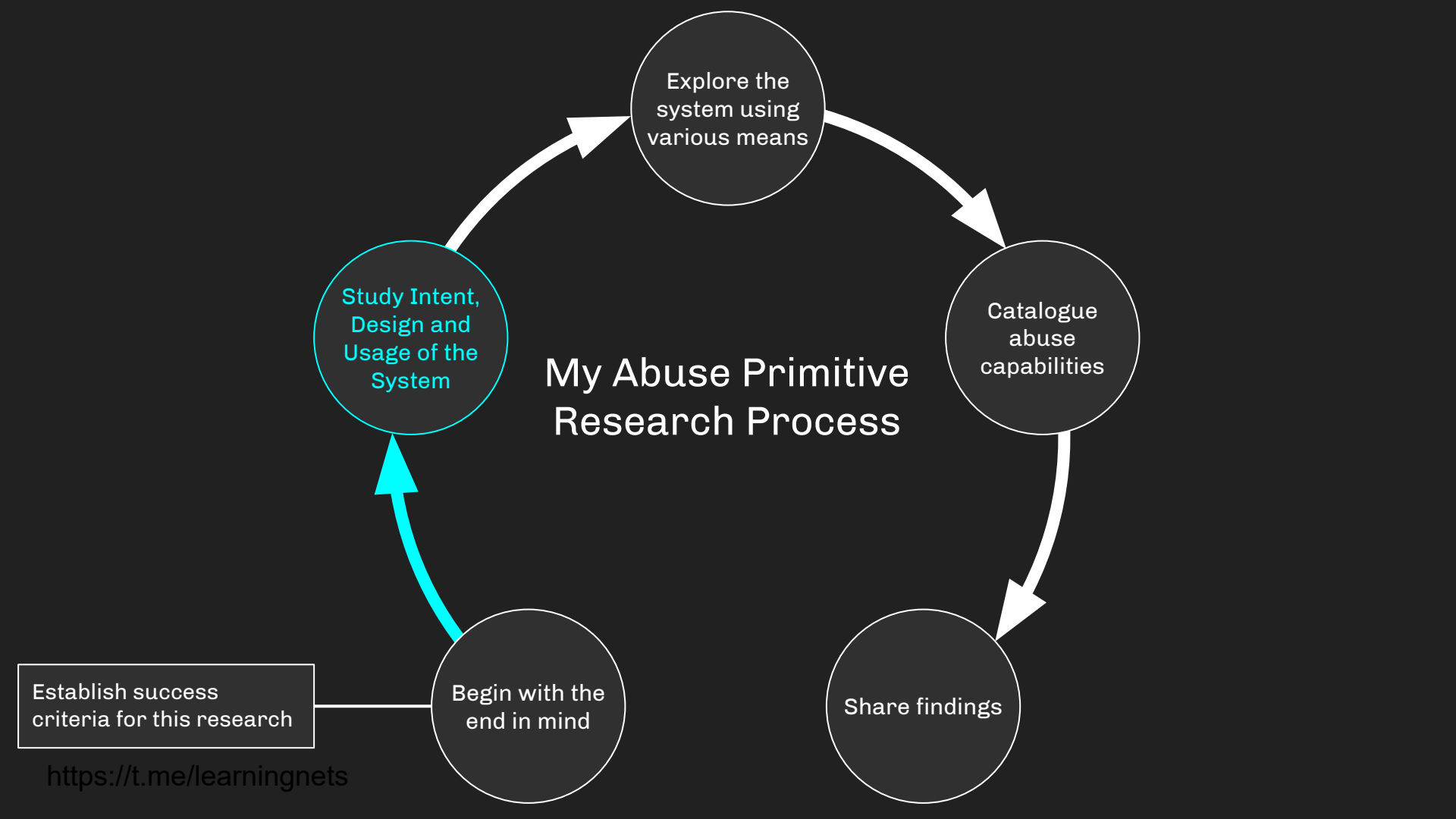
I want to produce:

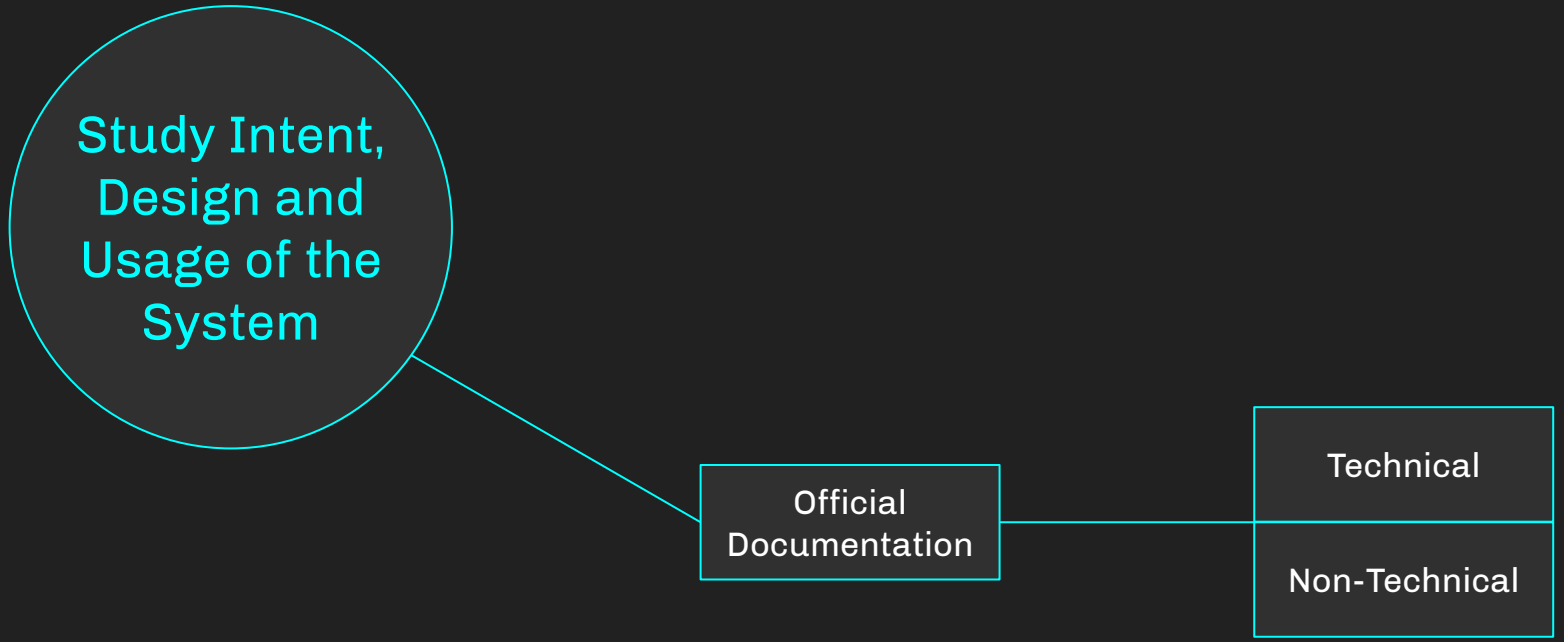
- A blog/talk for others to understand and build on
- Example audit and abuse code
- Practical remediation guidance

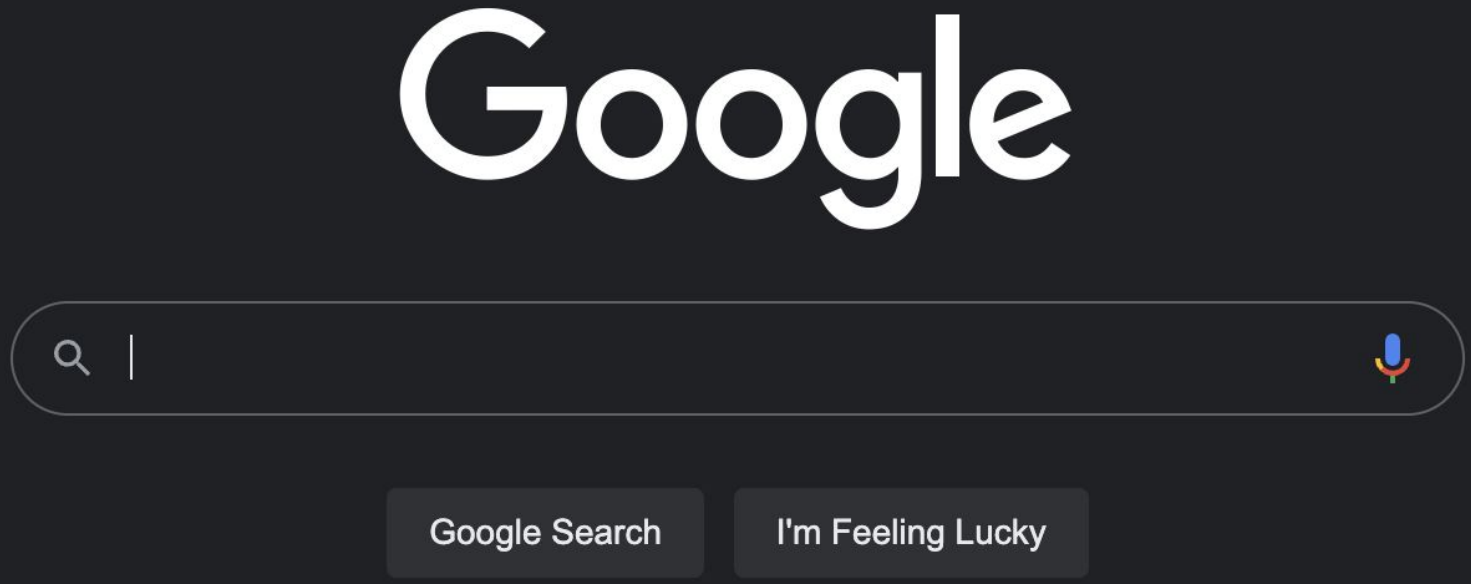
If appropriate for BloodHound, I want to prepare for:

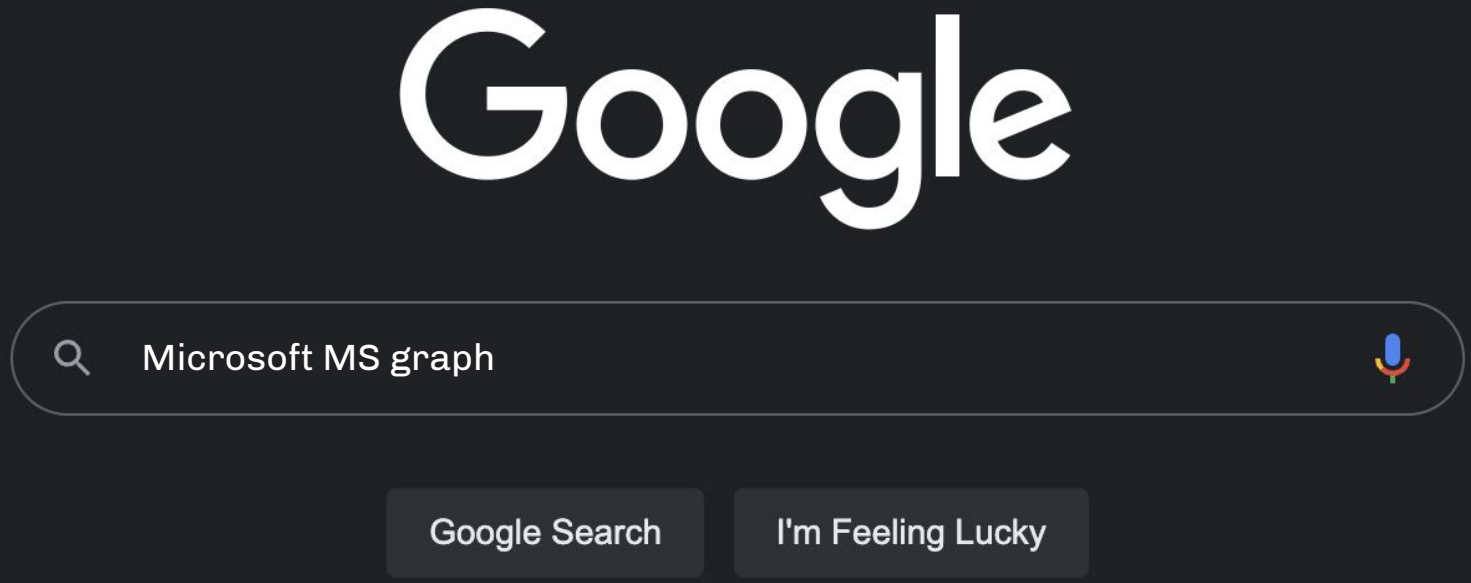
- The impact on the existing graph model
- How to expand the graph model
- What data to collect and ingest, and how to get that data







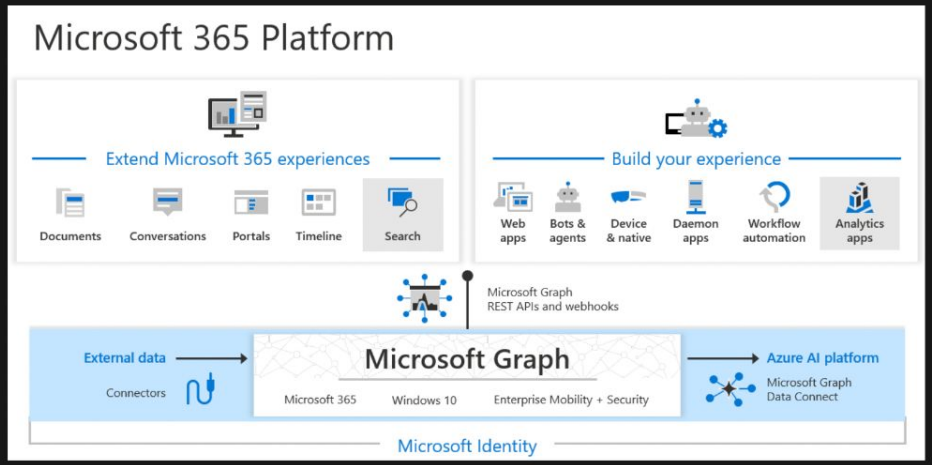




Overview of Microsoft Graph

Article • 11/11/2021 • 6 minutes to read • 16 contributors

Microsoft Graph is the gateway to data and intelligence in Microsoft 365. It provides a unified programmability model that you can use to access the tremendous amount of data in Microsoft 365, Windows 10, and Enterprise Mobility + Security. Use the wealth of data in Microsoft Graph to build apps for organizations and consumers that interact with millions of users.

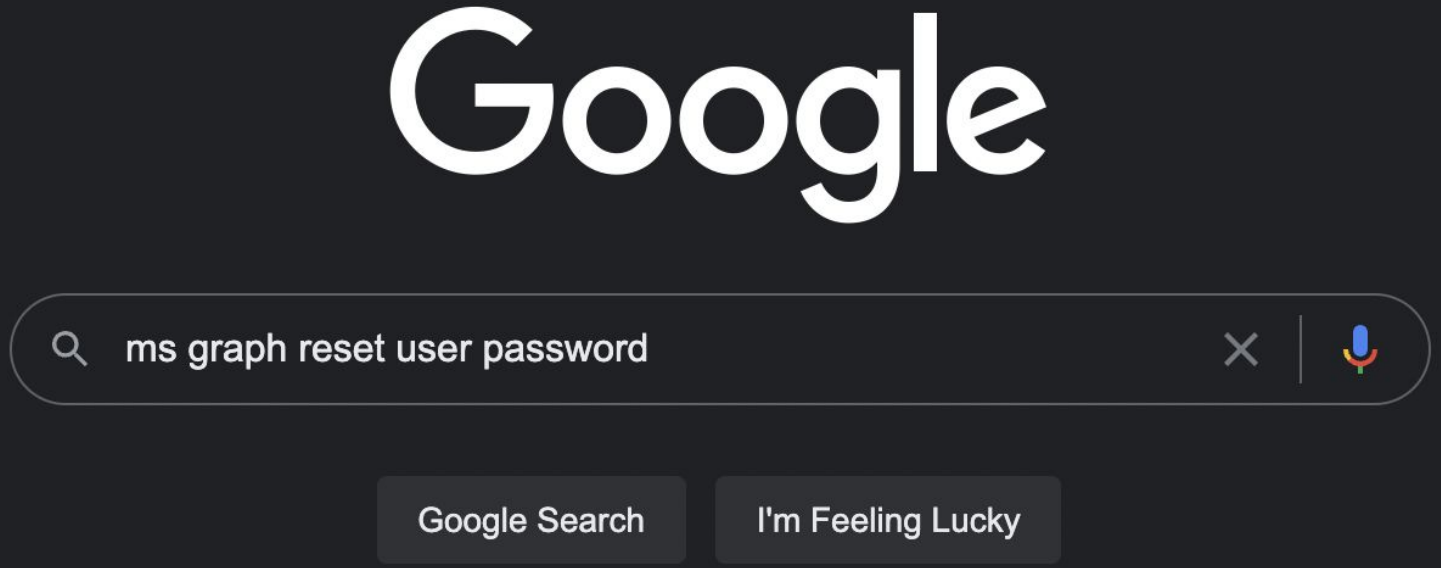


Data and services powering the Microsoft 365

What can you do with Microsoft Graph?

Use Microsoft Graph to build experiences around the user's unique context to help them be more productive. Imagine an app that...

- Looks at your next meeting and helps you prepare for it by providing profile information for attendees, including their job titles and managers, as well as information about the latest documents they're working on, and people they're collaborating with.
- Scans your calendar, and suggests the best times for the next team meeting.
- Fetches the latest sales projection chart from an Excel file in your OneDrive and lets you update the forecast in real time, all from your phone.
- Subscribes to changes in your calendar, sends you an alert when you're spending too much time in meetings, and provides recommendations for the ones you can miss or delegate based on how relevant the attendees are to you.
- Helps you sort out personal and work information on your phone; for example, by categorizing pictures that should go to your personal OneDrive and business receipts that should go to your OneDrive for Business.
- Analyzes at-scale Microsoft 365 data so that decision makers can unlock valuable insights into time allocation and collaboration patterns that improve business productivity.



Version
Microsoft Graph REST API Beta
Filter by title
Get
Reset
Get operation status
Certificate-based auth configuration
Conditional Access
Cross-tenant access (preview)
Data policy operation
Identity protection
Identity provider
Identity provider (deprecated)
Invitation manager
Organizational branding
Policies
Trust Framework Keyset
Trust Framework policy
User flows
Governance
Mail
Notes
Notifications (deprecated)

passwordAuthenticationMethod: resetPassword

Article • 11/29/2021 • 6 minutes to read • 8 contributors

Namespace: microsoft.graph

> **Important**
>
> APIs under the `/beta` version in Microsoft Graph are subject to change. Use of these APIs in production applications is not supported. To determine whether an API is available in v1.0, use the **Version** selector.

Initiate a reset for the password associated with a password authentication method object. This can only be done by an administrator with appropriate permissions and cannot be performed on a user's own account.

This flow writes the new password to Azure Active Directory and pushes it to on-premises Active Directory if configured using password writeback. The admin can either provide a new password or have the system generate one. The user is prompted to change their password on their next sign in.

This reset is a long-running operation and will return a link in the `Location` header where the caller can periodically check for the status of the reset.

Permissions

One of the following permissions is required to call this API. To learn more, including how to choose permissions, see Permissions.

Version

Microsoft Graph REST API Beta

Filter by title

Get

Reset

Get operation status

Certificate-based auth configuration

Conditional Access

Cross-tenant access (preview)

Data policy operation

Identity protection

Identity provider

Identity provider (deprecated)

Invitation manager

Organizational branding

Policies

Trust Framework Keyset

Trust Framework policy

User flows

Governance

Mail

Notes

Notifications (deprecated)

People and workplace intelligence

Personal contacts

Reports

Search

Permissions

One of the following permissions is required to call this API. To learn more, including how to choose permissions, see Permissions.

Permissions acting on self

The operation cannot be performed on a user's own account.

Permission type	Permissions (from least to most privileged)
Delegated (work or school account)	Not supported.
Delegated (personal Microsoft account)	Not supported.
Application	Not supported.

Permissions acting on other users

Only an administrator with the appropriate permissions can perform this operation.

Permission type	Permissions (from least to most privileged)
Delegated (work or school account)	UserAuthenticationMethod.ReadWrite.All
Delegated (personal Microsoft account)	Not supported.
Application	Not supported.

For delegated scenarios where an admin is acting on another user, the admin needs one of the following Azure AD roles:

- Global administrator
- Privileged authentication administrator
- Authentication administrator

In this article

Permissions

HTTP request

Request headers

Request body

Response

Examples

Microsoft Graph permissions reference

Article • 02/17/2022 • 123 minutes to read • 103 contributors

For your app to access data in Microsoft Graph, the user or administrator must grant it the correct permissions via a consent process. This topic lists the permissions associated with each major set of Microsoft Graph APIs. It also provides guidance about how to use the permissions.

> **Note**
>
> As a best practice, request the least privileged permissions that your app needs in order to access data and function correctly. Requesting permissions with more than the necessary privileges is poor security practice, which may cause users to refrain from consenting and affect your app's usage.

To learn more about how permissions work, see Authentication and authorization basics, and watch the following video.

Initial MS Graph Notes

- MS Graph is some kind of Azure-related service

Initial MS Graph Notes

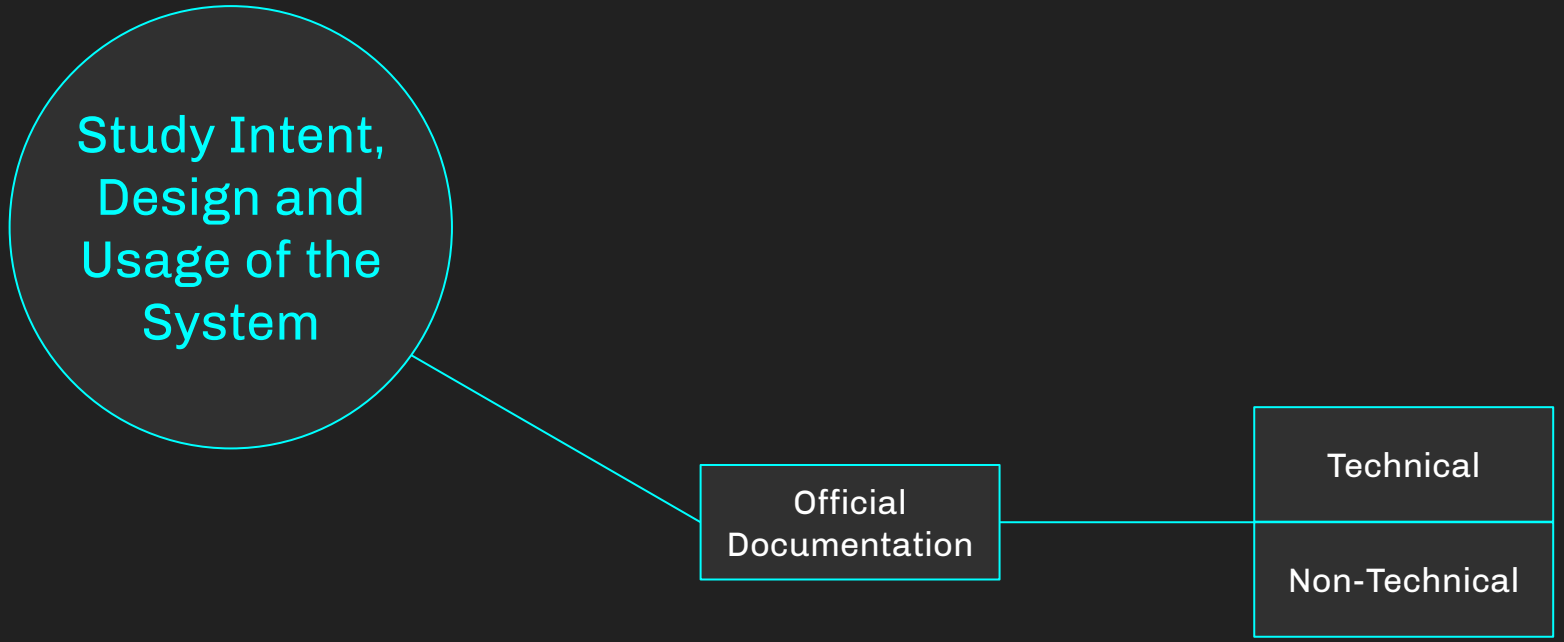
- MS Graph is some kind of Azure-related service
- It exposes a REST API

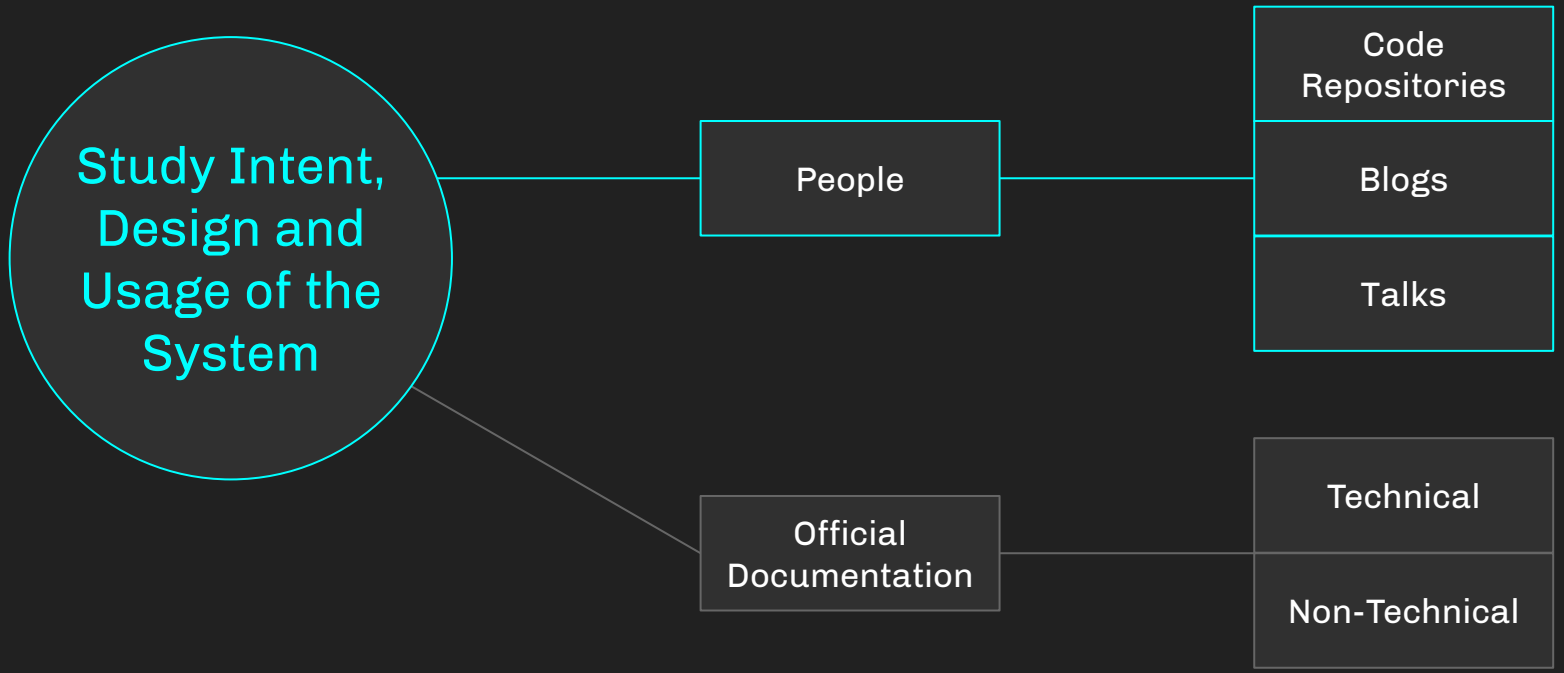
Initial MS Graph Notes

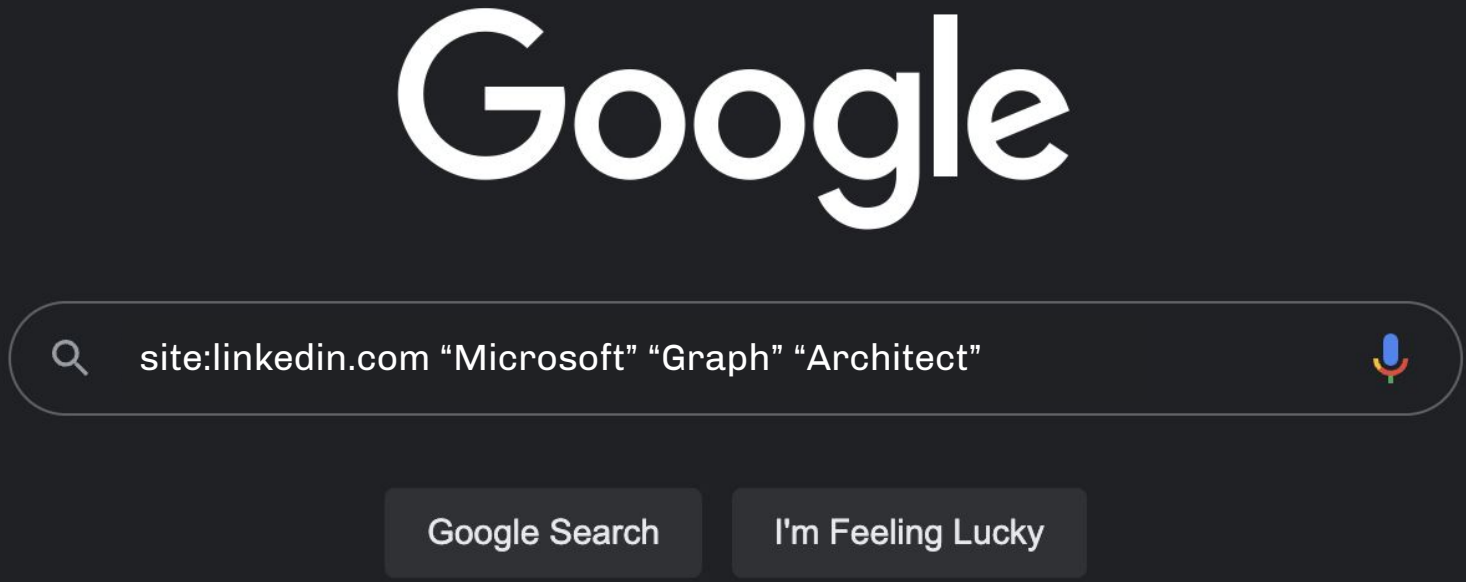
- MS Graph is some kind of Azure-related service
- It exposes a REST API
- It can let you reset other users' passwords if you have the right permissions

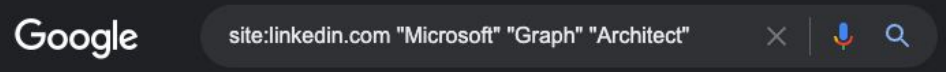
Initial MS Graph Notes

- MS Graph is some kind of Azure-related service
- It exposes a REST API
- It can let you reset other users' passwords if you have the right permissions
- It seems to have a distinct permissions system versus Azure AD and Azure RM









All Images News Shopping Videos More Tools

About 40,400 results (0.44 seconds)

https://www.linkedin.com › garethalunjones

Gareth Jones - Principal API Architect for Microsoft Graph

Redmond, Washington, United States · Principal API Architect for Microsoft Graph · Microsoft
As the API **Architect** for the **Microsoft Graph**, I'm responsible for the all-up design choices and technical direction of Microsoft's most strategic API.

https://ca.linkedin.com › darrelmiller

Darrel Miller - API Architect - Microsoft | LinkedIn

API **Architect** on **Microsoft Graph** ... As a member of the **Microsoft Graph** Developer Experience team, ... Senior Program Manager - Graph Developer Tooling.

https://www.linkedin.com › ...

Joseph Cortese - Architect - Microsoft | LinkedIn

New Alexandria, Virginia, United States · Architect · Microsoft
Architect at Microsoft. Microsoft ... Senior Program Manager - **Microsoft Graph** at Microsoft ... Azure **Architect**, Principal Consultant at Microsoft.

https://www.linkedin.com › steveanonsen

Steve Anonsen - Graduate Student - Biola University | LinkedIn

Fargo, North Dakota, United States · Graduate Student · Biola University
I'm API **Architect** for the **Microsoft Graph** (http://aka.ms/graph), driving API strategy, conventions and runtime for cloud APIs from Office and other ...

Announcing “30 Days of Microsoft Graph” Blog Series

Brian

October 30th, 2018

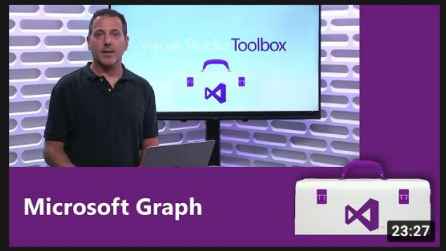
Throughout the month of November 2018, we are publishing daily articles (30 total) that aim to introduce developers to Microsoft Graph. We’ll have content that covers 0-level to 200-level topics. Each post should take you 5-15 mins to read and try out the sample exercises. No prior knowledge of Microsoft Graph is required. We hope that beginners will quickly pick up the content and that experts will also learn a few new things.

Please be sure to bookmark this page (https://aka.ms/30DaysMSGraph) as the below list of topics will be updated as each day’s post is published.

youtube.com/results?search_query=microsoft+graph

YouTube

microsoft graph



Microsoft Graph

41K views • 2 years ago

Microsoft Visual Studio

Microsoft Graph exposes REST APIs and client libraries to access data on a number of Microsoft 365 services. In this episode ...

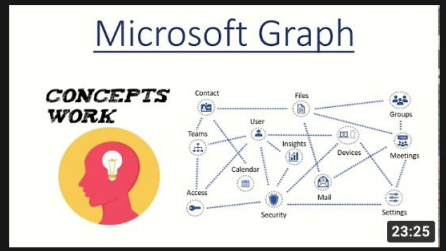
CC

Start calling the Microsoft Graph API in under 5 minutes! | Tips & Tricks

25K views • 10 months ago

Microsoft 365 Developer

Learn tips & tricks to start calling the **Microsoft Graph** API in under 5 minutes. Microsoft Cloud Advocate, Dan Wahlin ...



What is Microsoft Graph?

16K views • 1 year ago

Concepts Work

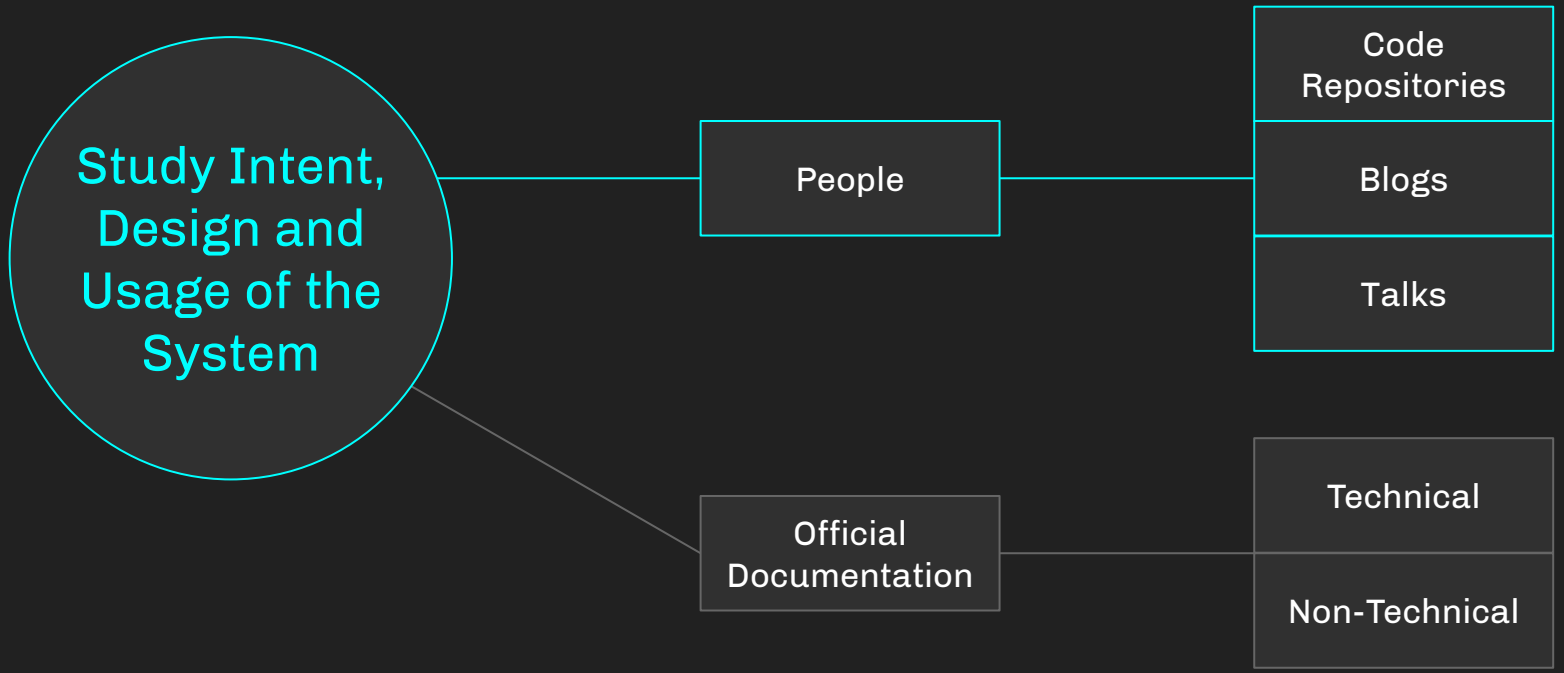
Microsoft #Graph #MicrosoftGraph What is **Microsoft Graph**? What is **Microsoft Graph** Api? What is **Microsoft Graph** Explorer?

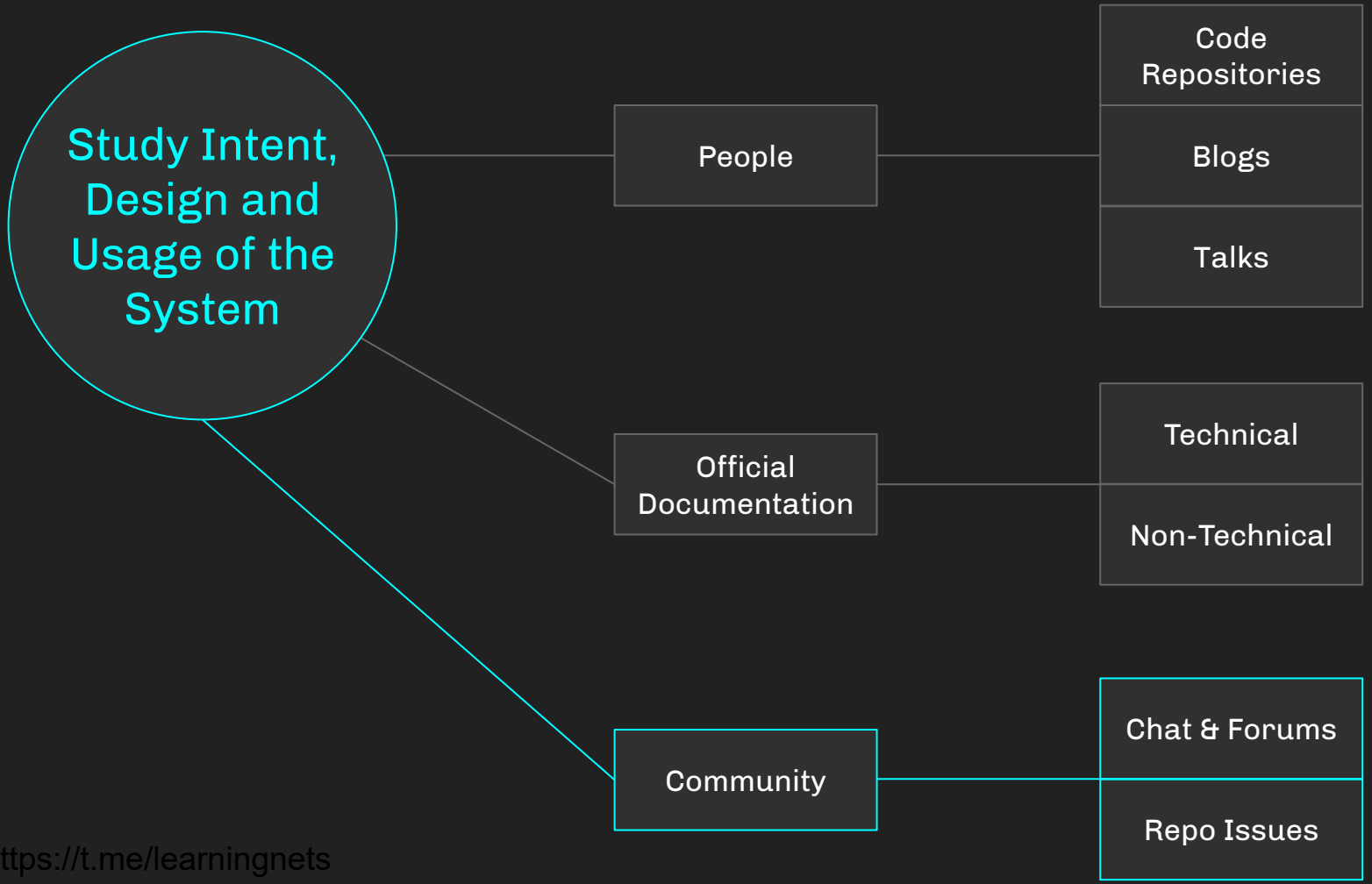
What is Microsoft Graph?

24K views • 3 years ago

Pragmatic Works

Learn about **Microsoft Graph**, an API that you can use to connect to various Office 365 services. It's a single endpoint that gets you ...





github.com/microsoftgraph/msgraph-sdk-powershell/issues

185 Open ✓ 505 Closed Author Label Projects Milestones Assignee Sort

- Powershell Microsoft.Graph- How to set MaxRetry ToTriage
  #1106 opened 2 hours ago by MarkDordoy
- Continuous access evaluation error ToTriage
  #1105 opened 5 hours ago by rickjansen-dev
- Users should use Authentication Providers to securely connect to Microsoft Graph ToTriage
  #1104 opened yesterday by sebastienlevert
- Authentication Providers ToTriage
  #1103 opened yesterday by sebastienlevert
- Set-MgApplicationLogo requires Content-Type to be image/* ToTriage
  #1097 opened 4 days ago by peombwa
- Clean-up AzDo Pipeline for Continuous Release 1
  #1096 opened 4 days ago by peombwa
- Selecting fields with Get-MgSiteListItem ToTriage
  #1094 opened 5 days ago by lkubat
- Invoke-MgInvalidateUserRefreshToken throwing exception, although operation succeeds ToTriage
  #1092 opened 7 days ago by khant414
- get-mguser doesn't always return Id ToTriage
  #1091 opened 7 days ago by anwarmahmood1
- Searching for equivalent command ToTriage
  #1090 opened 7 days ago by Tiberriver256
- Issue adding an application to Catalog through PowerShell SDK - New-MgEntitlementManagementAccessPackageResourceRequest ToTriage
  #1089 opened 7 days ago by purish87
- Create documentation for the cmdlets Authentication module Request: Documentation 3
  #1088 opened 8 days ago by maisarissi 03/22

Initial MS Graph Notes

- MS Graph is some kind of Azure-related service
- It exposes a REST API
- It can let you reset other users' passwords if you have the right permissions
- It seems to have a distinct permissions system versus Azure AD and Azure RM

Updated MS Graph Notes

- Microsoft is investing into MS Graph as a sort of “API for APIs”, a unifying endpoint that may eventually allow for indirectly interacting with any object in any service just by interfacing with MS Graph.

Updated MS Graph Notes

- Microsoft is investing into MS Graph as a sort of “API for APIs”, a unifying endpoint that may eventually allow for indirectly interacting with any object in any service just by interfacing with MS Graph.

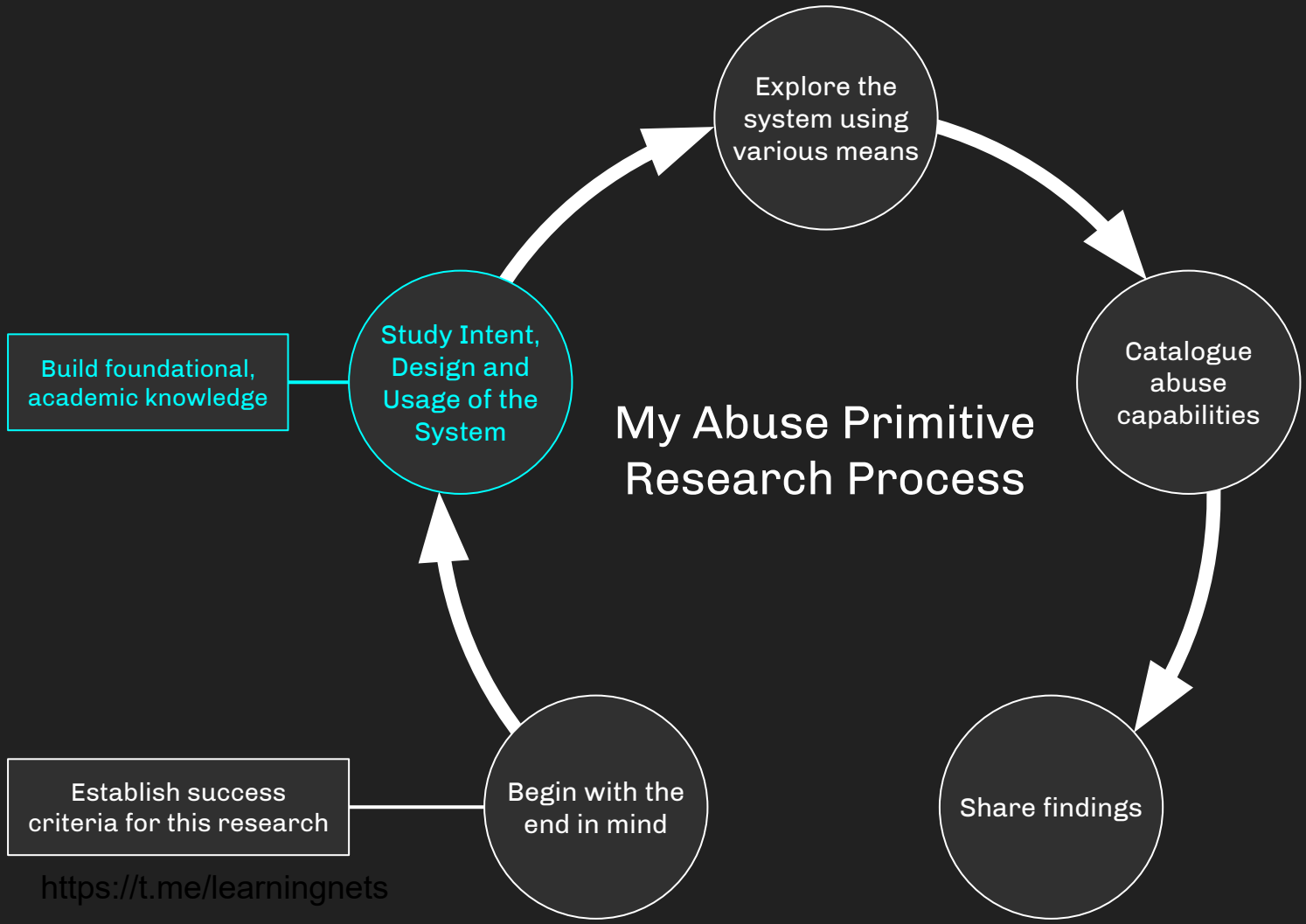
- MS Graph’s REST API is instantiated into every Azure tenant as a Resource App (aka Enterprise App, aka Service Principal, aka First Party App).

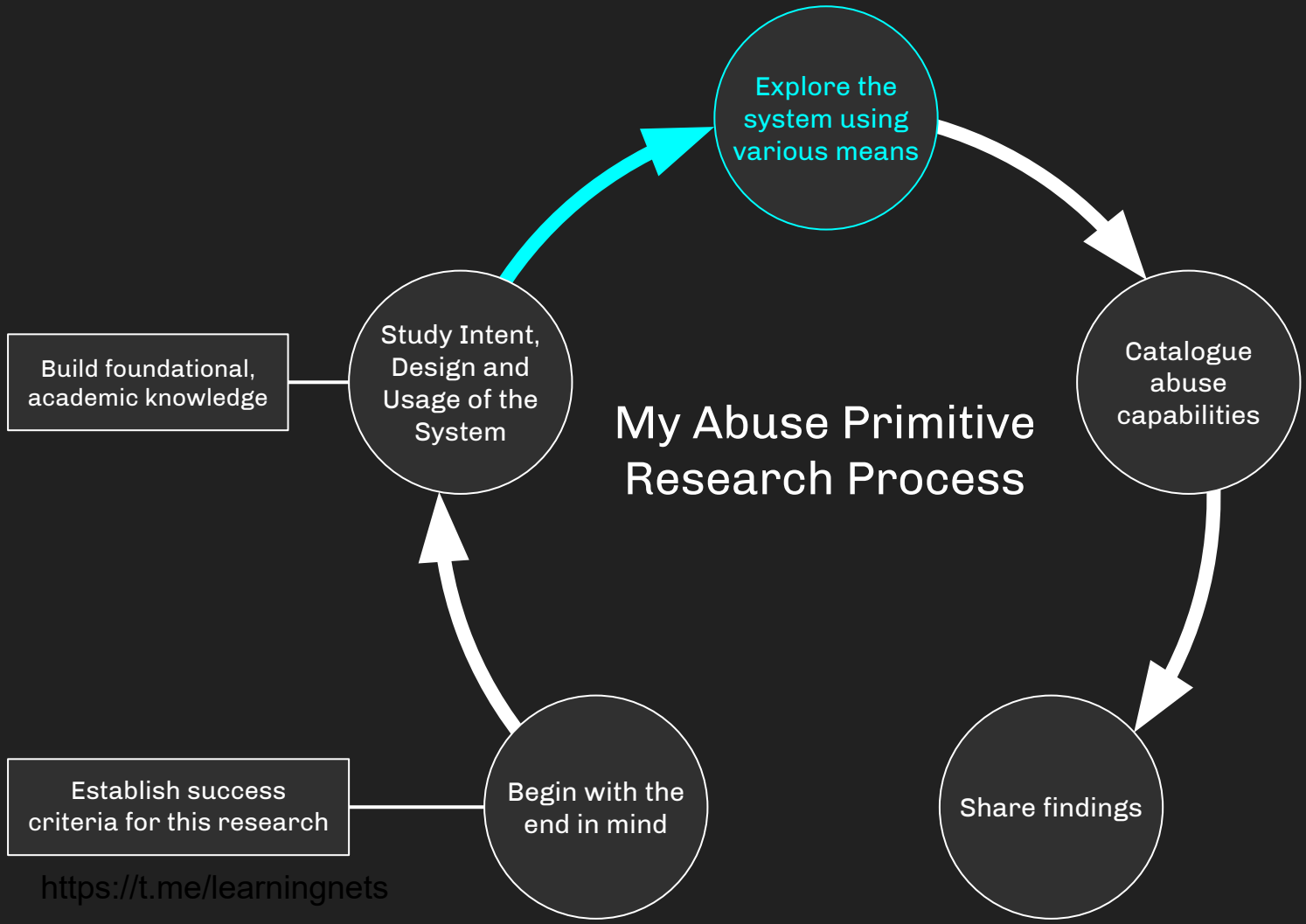
Updated MS Graph Notes

- Microsoft is investing into MS Graph as a sort of “API for APIs”, a unifying endpoint that may eventually allow for indirectly interacting with any object in any service just by interfacing with MS Graph.

- MS Graph’s REST API is instantiated into every Azure tenant as a Resource App (aka Enterprise App, aka Service Principal, aka First Party App).

- MS Graph brokers requests to particular Azure services, including privileged action requests like resetting passwords or adding users to security groups.





You must go beyond the documentation.

- These systems are interconnected in undocumented and non-public ways

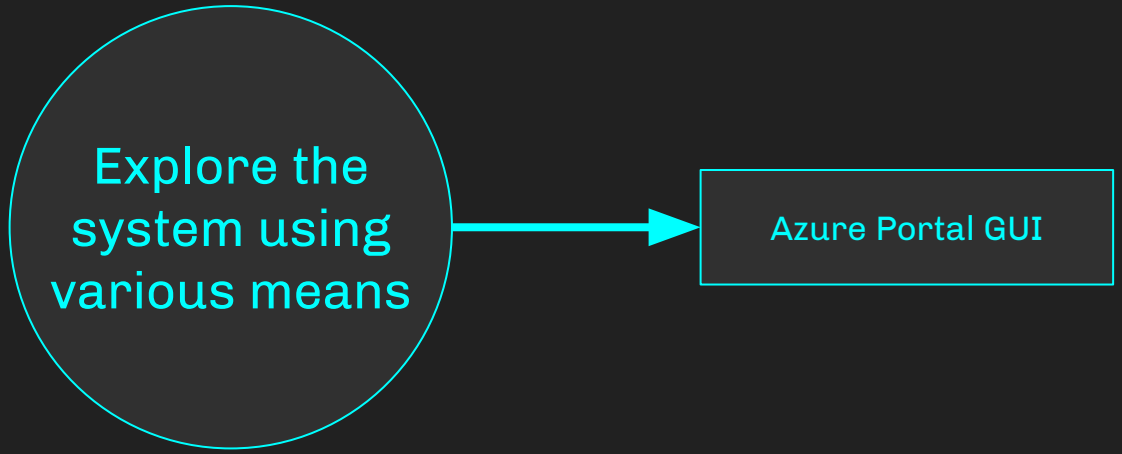
You must go beyond the documentation.

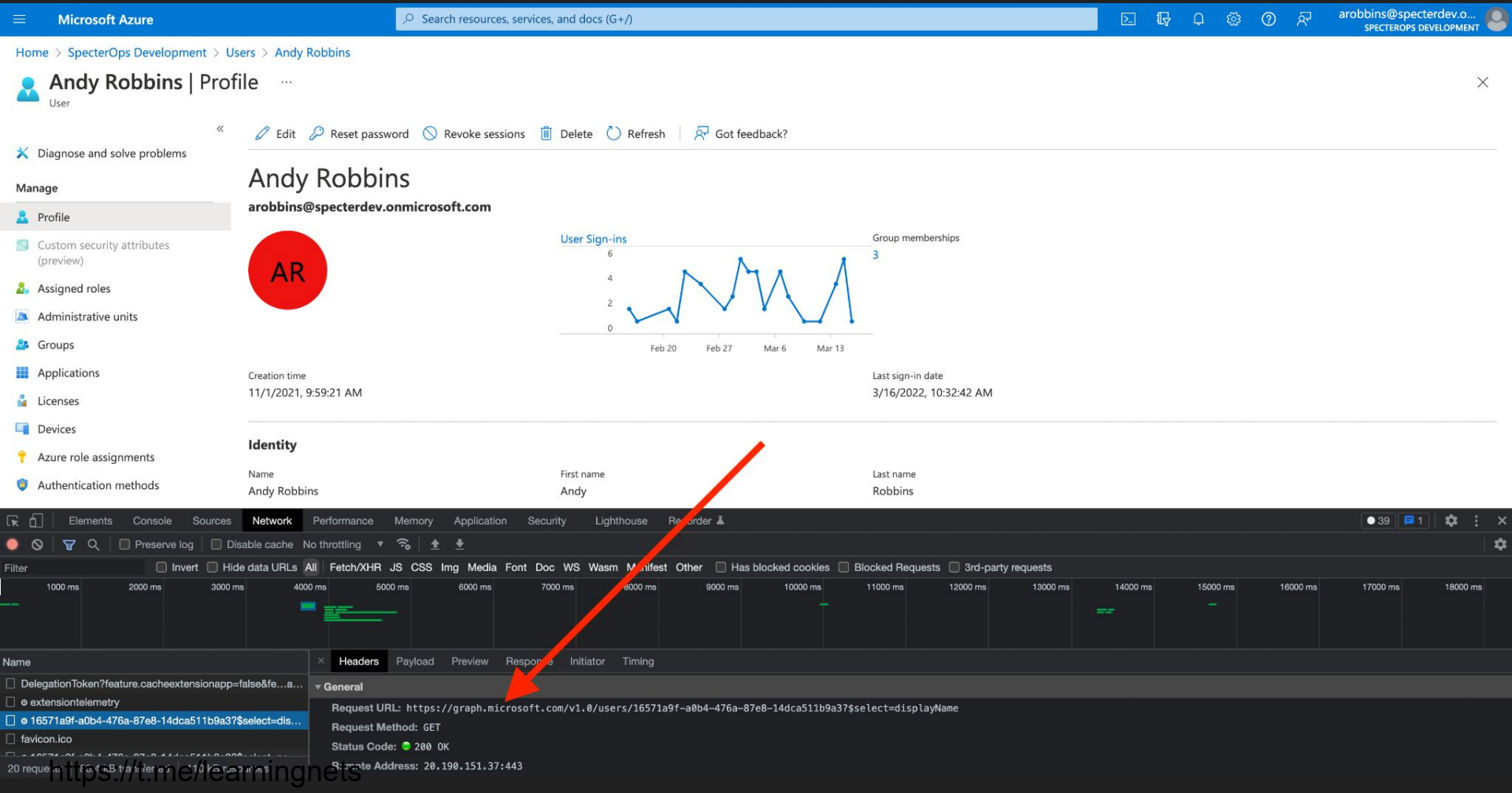
- These systems are interconnected in undocumented and non-public ways
- Documentation often doesn’t keep up with changes

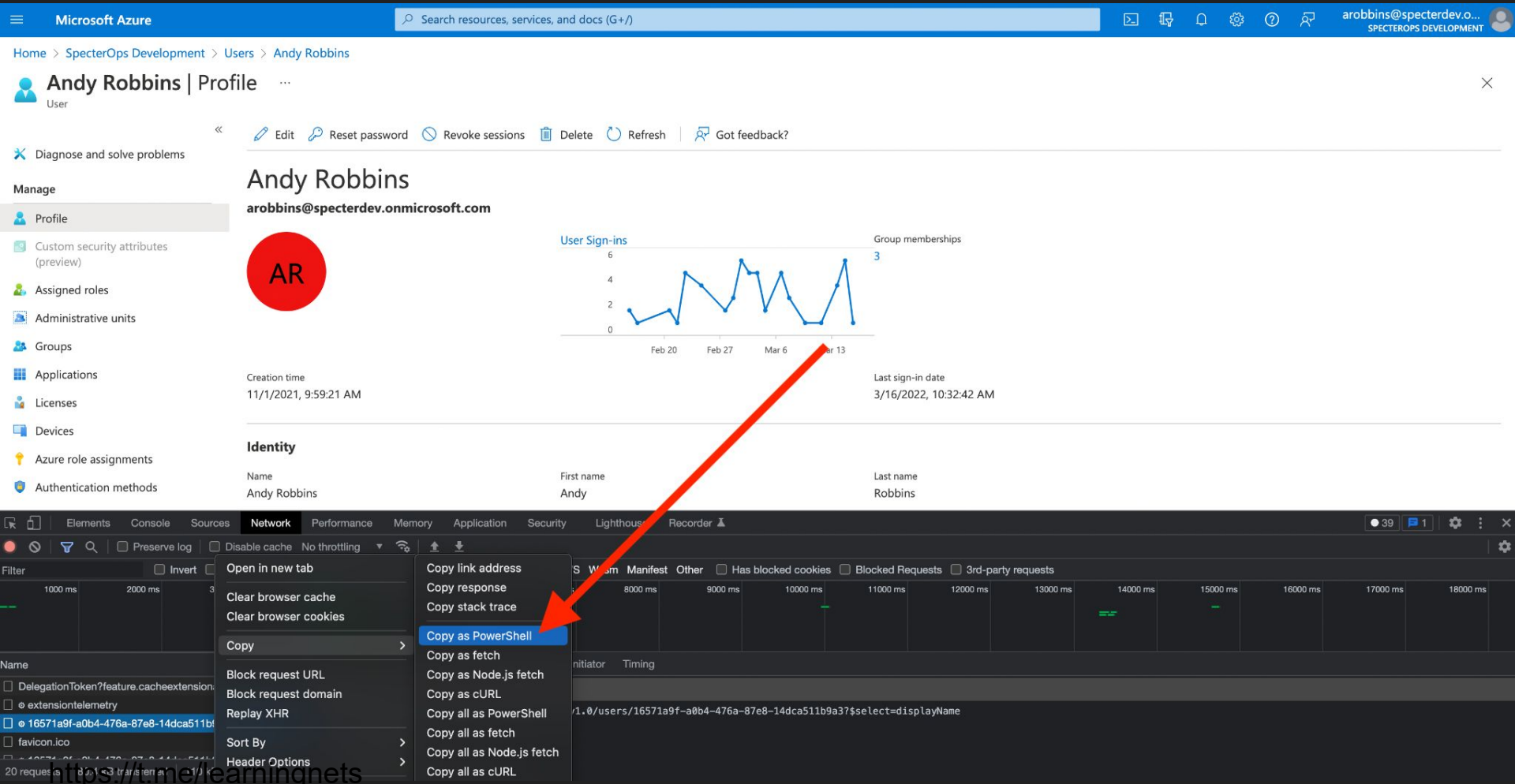
You must go beyond the documentation.

- These systems are interconnected in undocumented and non-public ways
- Documentation often doesn't keep up with changes
- Tooling based only on documentation is almost always inaccurate, unreliable tooling.

Explore the system using various means

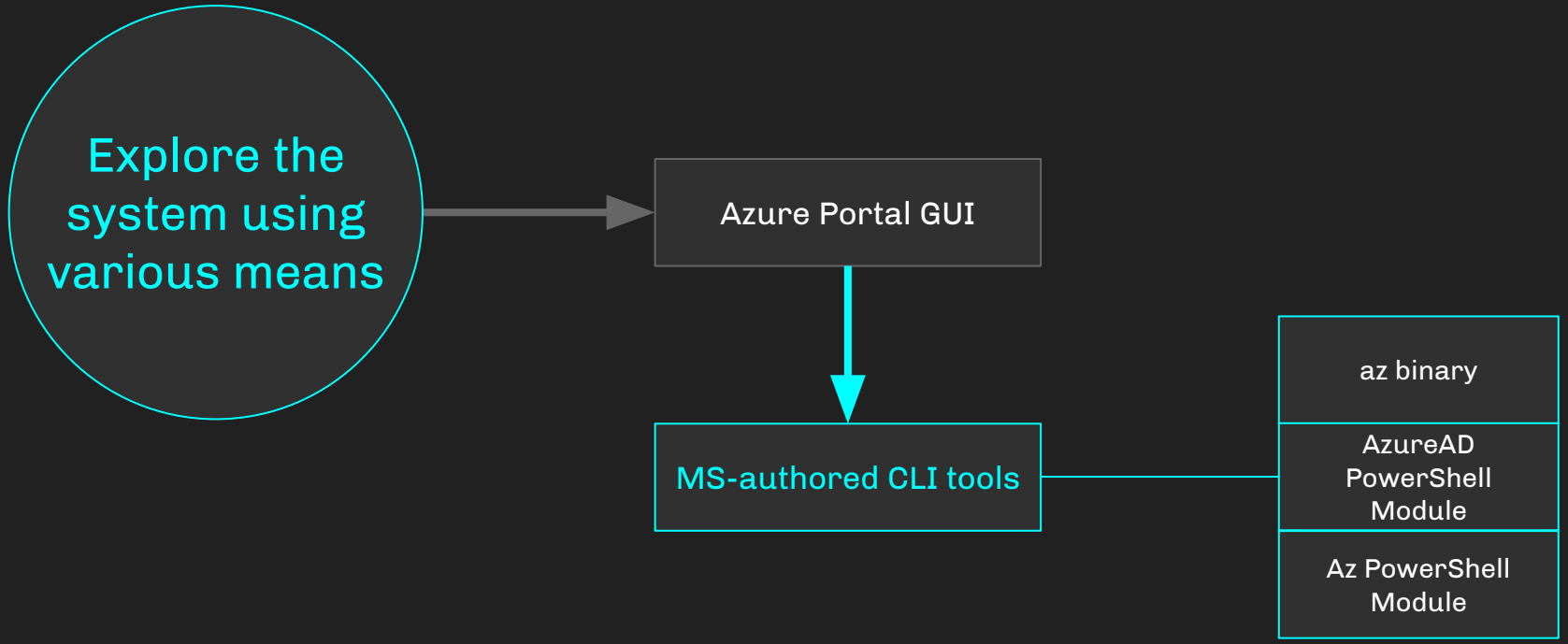


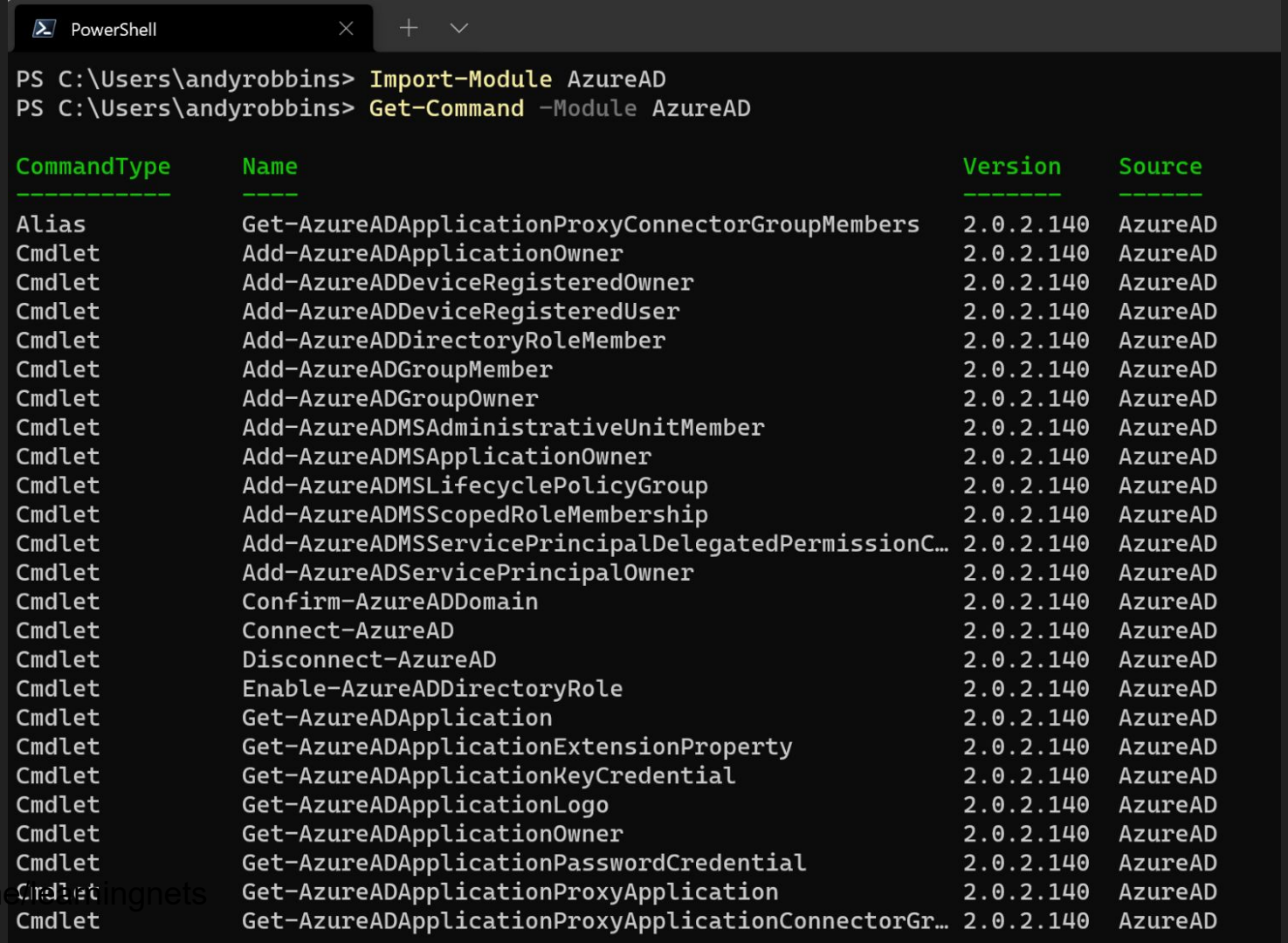




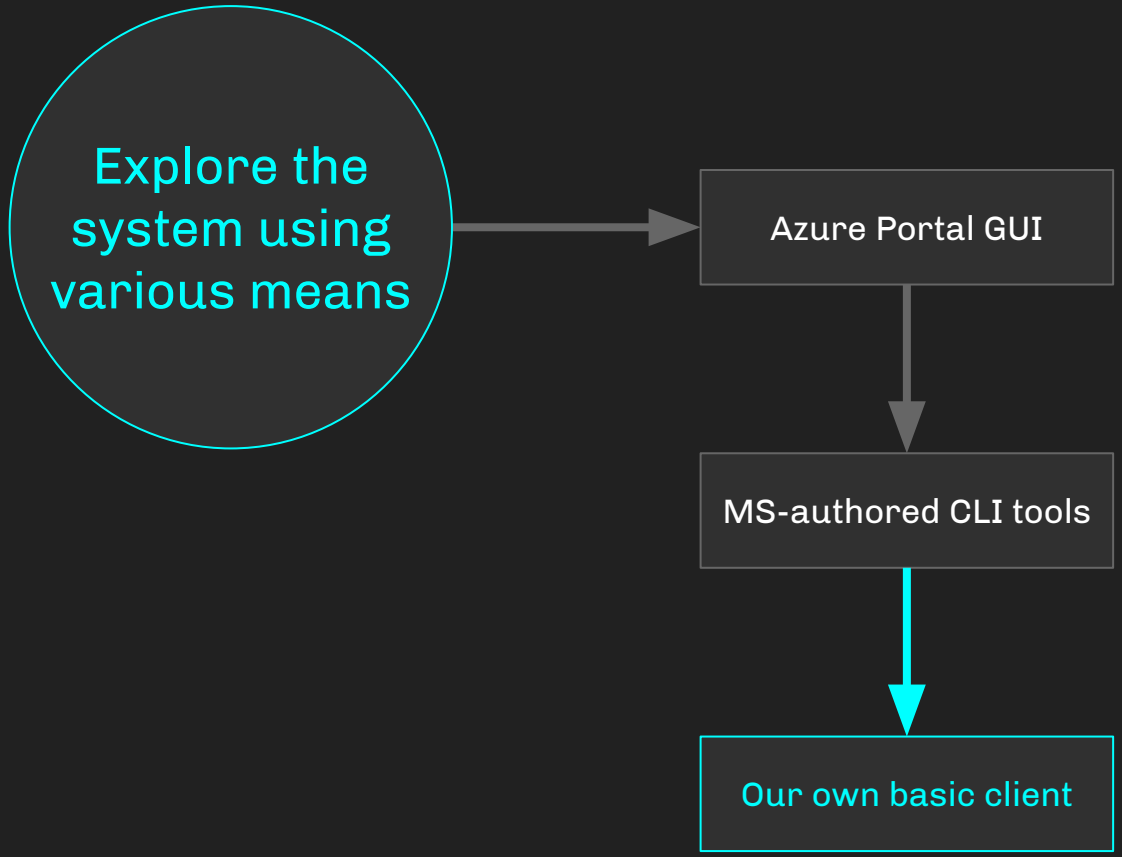
```
$session = New-Object Microsoft.PowerShell.Commands.WebRequestSession
$session.UserAgent = "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/
  99.0.4844.51 Safari/537.36"
Invoke-WebRequest -UseBasicParsing -Uri "https://graph.microsoft.com/v1.0/users/16571a9f-a0b4-476a-87e8-14dca511b9a3?`$
  select=displayName" `
-WebSession $session `
-Headers @{
"x-ms-client-session-id"="f9bdae195136432c82b5f27fdddf1ad2"
  "DNT"="1"
  "x-ms-command-name"="UserManagement%20-%20UserAllPropertiesDataMsGraph"
  "Accept-Language"="en"
  "Authorization"="Bearer eyJ0eXAiOiJKV1QiLCJub25jZSI6Ikl3WHA3cy14MHV6RWJValVRb0NWZ1JiSEdXbU9GOW9zUllwbUltek53REkiLCJhbGciO
  iJSUzI1NiIsIng1dCI6ImpTMVhvMU9XRGpfNTJ2YndHTmd2UU8yVnpNYyIsImtpZCI6ImpTMVhvMU9XRGpfNTJ2YndHTmd2UU8yVnpNYyJ9.eyJhdWQiOiJod
  HRwczovL2dyYXBoLm1pY3Jvc29mdC5jb20vIiwiaXNzIjoiaHR0cHM6Ly9zdHMud2luZG93cy5uZXQvNmMxMmIwYjAtYjJjYy00YTczLTgyNTItMGI5NGJmY2
  EyMTQ1LyIsImlhdCI6MTY0NzU1MjU4NSwibmJmIjoxNjQ3NTUyNTg1LCJleHAiOjE2NDc1NTc4MDIsImFjY3QiOjAsImFjciI6IjEiLCJhaW8iOiJFMlpnWUl
  qYWZxUmxlKytaajlLVHRTVFdybGY3d2VTWjFLQ3p0L1JVL1BhR200SU96bXNCIiwiYW1yIjpbInB3ZCJdLCJhcHBfZGlzcGxheW5hbWUiOiJBREliaXphVVgi
  LCJhcHBpZCI6Ijc0NjU4MTM2LTE0ZWMtNDYzMC1hZDliLTI2ZTE2MGZmMGZjNiIsImFwcGlkYWNyIjoiMiIsImNvbnRyb2xzIjpbImNhX2VuZiJdLCJmYW1pb
  HlfbmFtZSI6IlJvYmJpbnMiLCJnaXZlbl9uYW1lIjoiQW5keSIsImlkdHlwIjoidXNlciIsImlwYWRkciI6IjIwOC43MS4xNjQuMTk1IiwibmFtZSI6IkFuZH
  kgUm9iYmlucyIsIm9pZCI6IjE2NTcxYTlmLWEwYjQtNDc2YS04N2U4LTE0ZGNhNTExYjlhMyIsInBsYXRmIjoiNSIsInB1aWQiOiIxMDAzMjAwMUEwQThFQzR
  EIiwicmgiOiIwLkFWRUFzTEFTYk15eWMwcUNVZ3VVdjhvaFJRTUFBQUFBQUFBQXdBQUFBQUFBQUFCUkFLYy4iLCJzY3AiOiJBY2Nlc3NSZXZpZXcuUmVhZFdy
  aXRlLkFsbCBBdWRpdExvZy5SZWFkLkFsbCBEaXJlY3RvcnkuQWNjZXNzQXNVc2VyLkFsbCBEaXJlY3RvcnkuUmVhZC5BbGwgRGlyZWN0b3J5LlJlYWRXcml0Z
  S5BbGwgRGlyZWN0b3J5UmVjb21tZW5kYXRpb25zLlJlYWQuQWxsIGVtYWlsIEVudGl0bGVtZW50TWFuYWdlbWVudC5SZWFkLkFsbCBHcm91cC5SZWFkV3JpdG
  UuQWxsIElkZW50aXR5UHJvdmlkZXIuUmVhZFdyaXRlLkFsbCBJZGVudGl0eVJpc2tFdmVudC5SZWFkV3JpdGUuQWxsIElkZW50aXR5VXNlckZsb3cuUmVhZC5
  BbGwgb3BlbmlkIFBvbGljeS5SZWFkLkFsbCBQb2xpY3kuUmVhZFdyaXRlLkF1dGhlbnRpY2F0aW9uRmxvd3MgUG9saWN5LlJlYWRXcml0ZS5BdXRoZW50aWNh
  dGlvbk1ldGhvZCBQb2xpY3kuUmVhZFdyaXRlLkNvbmRpdGlvbmFsQWNjZXNzIHByb2ZpbGUgUmVwb3J0cy5SZWFkLkFsbCBSb2xlTWFuYWdlbWVudC5SZWFkV
  3JpdGUuRGlyZWN0b3J5IFNlY3VyaXR5RXZlbnRzLlJlYWRXcml0ZS5BbGwgVHJ1c3RGcmFtZXdvcmtLZXlTZXQuUmVhZC5BbGwgVXNlci5FeHBvcnQuQWxsIF
  VzZXIuUmVhZFdyaXRlLkFsbCBVc2VyQXV0aGVudGljYXRpb25NZXRob2QuUmVhZFdyaXRlLkFsbCIsInN1YiI6InhidUhwTHRfbVRHRURPcldXOFJUMUhBdW5
  pWi0xbDV2S0h5cEtUT3gzMFUiLCJ0ZW5hbnRfcmVnaW9uX3Njb3BlIjoiTkEiLCJ0aWQiOiI2YzEyYjBiMC1iMmNjLTRhNzMtODI1Mi0wYjk0YmZjYTIxNDUi
  LCJ1bmlxdWVfbmFtZSI6ImFyb2JiaW5zQHNwZWN0ZXJkZXYub25taWNyb3NvZnQuY29tIiwidXBuIjoiYXJvYmJpbnNAc3BlY3RlcmRldi5vbm1pY3Jvc29md
  C5jb20iLCJ1dGkiOiI3Q3lGbTZ5R0xVNlkxNzRKaTdZUUFBIiwidmVyIjoiMS4wIiwid2lkcyI6WyI2MmU5MDM5NC02OWY1LTQyMzctOTE5MC0wMTIxNzcxND
  VlMTAiLCJiNzlmYmY0ZC0zZWY5LTQ2ODktODE0My03NmIxOTRlODU1MDkiXSwieG1zX3N0Ijp7InN1YiI6IldfaGRJWDlmRjR5SlpRRVRyOHN6N21YMFZzeDB
  nSVg0Wkl2Q3FDRXpMTFUifSwieG1zX3RjZHQiOjE1ODg2MDI4NzN9.ahiEMnLxJusjxBa4VAfJE64hkI7bpAFJJOXMQ5WzRprLck2hBjvPIbUat2nOLJlL4Za
  gsqVW-mojxdcCtme0MIKYWPebYnK9hWnFHLSmkAYSGPyuSkYPDGCxHAHeYV0oiKQoAPL58lgv4pia32YkAjXdccnqvOamKalMA089kUcv13tZqYOyPaAIN0iM
  O--5og78M0BV8RYQHP3WinvT6Jg8P7NxDpgiUK22kHvlJXj-D_EMMsLEC20Ey7ImwESpNblvkrGHCu4IhUKN6ljoSkytlvxJ6JCtK7gksqCWbtDr81e5fohn5
  SI5mRnysxq-WE8ZT8wrrtrTNArFlkQ00g"
  "x-ms-effective-locale"="en.en-us"
  "Accept"="*/*"
  "Referer"=""
  "x-ms-client-request-id"="b0864d22-023e-4e77-afcf-9b36de5c7059"
}
```

```
PS /Users/andyrobbins> $session = New-Object Microsoft.PowerShell.Commands.WebRequestSession
PS /Users/andyrobbins> $session.UserAgent = "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.51 Safari/537.36"
PS /Users/andyrobbins> Invoke-WebRequest -UseBasicParsing -Uri "https://graph.microsoft.com/v1.0/users/16571a9f-a0b4-476a-87e8-14dca511b9a3?`$select=displayName" `
>> -WebSession $session `
>> -Headers @{
>> "x-ms-client-session-id"="f9bdae195136432c82b5f27fdddf1ad2"
>>   "DNT"="1"
>>   "x-ms-command-name"="UserManagement%20-%20UserAllPropertiesDataMsGraph"
>>   "Accept-Language"="en"
>>   "Authorization"="Bearer eyJ0eXAiOiJKV1QiLCJub25jZSI6Ikl3WHA3cy14MHV6RWJValVRb0NWZ1JiSEdXbU9GOW9zUllwbUltek53REkiLCJhbGciOiJSUzI1NiIsIng1dCI6ImpTMVhvMU9XRGpfNTJ2YndHTmd2UU8yVnpNYyIsImtpZCI6ImpTMVhvMU9XRGpfNTJ2YndHTmd2UU8yVnpNYyJ9.eyJhdWQiOiJodHRwczovL2dyYXBoLm1pY3Jvc29mdC5jb20vIiwiaXNzIjoiaHR0cHM6Ly9zdHMud2luZG93cy5uZXQvNmMxMmIwYjAtYjJjYy00YTczLTgyNTItMGI5NGJmY2EyMTQ1LyIsImlhdCI6MTY0NzU1MjU4NSwibmJmIjoxNjQ3NTUyNTg1LCJleHAiOjE2NDc1NTc4MDIsImFjY3QiOjAsImFjciI6IjEiLCJhaW8iOiJFMlpnWUlqYWZxUmxlKytaajlLVHRTVFdybGY3d2VTWjFLQ3p0L1JVL1BhR200SU96bXNCIiwiYW1yIjpbInB3ZCJdLCJhcHBfZGlzcGxheW5hbWUiOiJBRElaXphVVgiLCJhcHBpZCI6Ijc0NjU4MTM2LTE0ZWMtNDYzMC1hZDliLTI2ZTE2MGZmMGZjNiIsImFwcGlkYWNyIjoiMiIsImNvbnRyb2xzIjpbImNhX2VuZiJdLCJmYW1pbHlfbmFtZSI6IlJvYmJpbnMiLCJnaXZlbl9uYW1lIjoiQW5keSIsImlkdHlwIjoidXNlciIsImlwYWRkciI6IjIwOC43MS4xNjQuMTk1IiwibmFtZSI6IkFuZHkgUm9iYmlucyIsInBsYXRmIjoiNSIsInB1aWQiOiIxMDAzMjAwMUEwQThFQzREIiwicmgiOiIwLkFWRUFzTEFTYk15eWMwcUNVZ3VVdjhvaFJRTUFBQUFBQUFBQXdBQUFBQUFCUkFLYy4iLCJzY3AiOiJBY2Nlc3NSZXZpZXcuUmVhZFdyaXRlLkFsbCBBdWRpdExvZy5SZWFkLkFsbCBEaXJlY3RvcnkuQWNjZXNzQXNVc2VyLkFsbCBEaXJlY3RvcnkuUmVhZC5BbGwgRGlyZWN0b3J5LlJlYWRXcml0ZS5BbGwgRGlyZWN0b3J5LlJlYWRXcml0ZS5BbGwgRGlyZWN0b3J5IFNlY3VyaXR5RXZlbnRzLlJlYWRXcml0ZS5BbGwgVHJ1c3RGcmFtZXdvcmtLZXlTZXQuUmVhZC5BbGwgVXNlci5FeHBvcnQuQWxsIFVzZXIuUmVhZFdyaXRlLkFsbCBVc2VyQXV0aGVudGljYXRpb25NZXRob2QuUmVhZFdyaXRlLkFsbCIsInN1YiI6InhidUhwTHRfbVRHRURPcldXOFJUMUhBdW5pWi0xbDV2S0h5cEtUT3gzMFUiLCJ0ZW5hbnRfcmVnaW9uX3Njb3BlIjoiTkEiLCJ0aWQiOiI2YzEyYjBiMC1iMmNjLTRhNzMtODI1Mi0wYjk0YmZjYTIxNDUiLCJ1bmlxdWVfbmFtZSI6ImFyb2JiaW5zQHNwZWN0ZXJkZXYub25taWNyb3NvZnQuY29tIiwidXBuIjoiYXJvYmJpbnNAc3BlY3RlcmRldi5vbm1pY3Jvc29mdC5jb20iLCJ1dGkiOiI3Q3lGbTZ5R0xVNlkxNzRKaTdZUUFBIiwidmVyIjoiMS4wIiwid2lkcyI6WyI2MmU5MDM5NC02OWY1LTQyMzctOTE5MC0wMTIxNzcxNDVlMTAiLCJiNzlmYmY0ZC0zZWY5LTQ2ODktODE0My03NmIxOTRlODU1MDkiXSwieG1zX3N0Ijp7InN1YiI6IldfaGRJWDlmRjR5SlpRRVRyOHN6N21YMFZzeDBnSVg0Wkl2Q3FDRXpMTFUifSwieG1zX3RjZHQiOjE1ODg2MDI4NzN9.ahiEMnLxJusjxBa4VAfJE64hkI7bpAFJJOXMQ5WzRprLck2hBjvPIbUat2nOLJlL4ZagsqVW-mojxdcCtme0MIKYWPebYnK9hWnFHLSmkAYSGPyuSkYPDGCxHAHeYV0oiKQoAPL581gv4pia32YkAjXdccnqvOamKalMA089kUcv13tZqYOyPaAIN0iMO--5og78M0BV8RYQHP3WinvT6Jg8P7NxDpgiUK22kHvlJXj-D_EMMsLEC20Ey7ImwESpNblvkrGHCu4IhUKN6ljoSkytlvxJ6JCtK7gksqCWbtDr81e5fohn5SI5mRnysxq-WE8ZT8wrrtrTNArFlkQ00g"
>>   "x-ms-effective-locale"="en.en-us"
>>   "Accept"="*/*"
>>   "x-ms-client-request-id"="b0864d22-023e-4e77-afcf-9b36de5c7066"
>> }

StatusCode        : 200
StatusDescription : OK
Content           : {"@odata.context":"https://graph.microsoft.com/v1.0/$metadata#users(displayName)/$entity","displayName":"Andy Robbins"}
RawContent        : HTTP/1.1 200 OK
                    Cache-Control: no-cache
                    Transfer-Encoding: chunked
                    Strict-Transport-Security: max-age=31536000
                    request-id: f5b23cf0-4f1e-47b3-8c0e-7a97eb1101b9
                    client-request-id: f5b23cf0-4f1e-47b3-8c…
Headers           : {[Cache-Control, System.String[]], [Transfer-Encoding, System.String[]], [Strict-Transport-Security, System.String[]], [request-id,
                    System.String[]]…}
Images            : {}
InputFields       : {}
Links             : {}
RawContentLength  : 119
RelationLink      : {}

PS /Users/andyrobbins>
```





```powershell
function Invoke-AzureHound {
    [CmdletBinding()]
    Param(
    [Parameter(Mandatory=$False)][String]$TenantID = $null,
    [Parameter(Mandatory=$False)][String]$OutputDirectory = $(Get-Location),[ValidateNotNullOrEmpty()]
    [Parameter(Mandatory=$False)][Switch]$Install = $null)

    if ($Install){
      Install-Module -Name Az -AllowClobber
      Install-module -Name AzureADPreview -AllowClobber
    }

    $Modules = Get-InstalledModule
    if ($Modules.Name -notcontains 'Az.Accounts' -and $Modules.Name -notcontains 'AzureAD'){
      Write-Host "AzureHound requires the 'Az' and 'Azure AD PowerShell module, please install by using the -Install switch."
      exit
    }

    $date = get-date -f yyyyMMddhhmmss

    #Login Check
    $APSUser = Get-AzContext *>&1
    if ($APSUser -eq $null){
        Connect-AzAccount
        $APSUser = Get-AzContext *>&1
        if ($APSUser -eq $null){
            Write-Host "Login via Az PS Module failed."
            exit
            }
    }
    Connect-AADUser
    $Headers = Get-AzureGraphToken

    If(!$TenantID){
        $TenantObj = Invoke-RestMethod -Headers $Headers -Uri 'https://graph.microsoft.com/beta/organization'
        $Tenant = $TenantObj.value
        $TenantId = $Tenant.id
        }

    # Enumerate users
    $Coll = New-Object System.Collections.ArrayList
    Write-Info "Building users object, this may take a few minutes."
        $AADUsers = Get-AzureADUser -All $True | Select UserPrincipalName,OnPremisesSecurityIdentifier,ObjectID,TenantId
```



Get a token

Get data from the API

Work with the data

```powershell
$Body = @{
    Grant_Type      =   "client_credentials"
    Scope           =   "https://graph.microsoft.com/.default"
    client_Id       =   "5cb4d011-ef3d-4c4a-afe4-1189dd1808ae"
    Client_Secret   =   "3qh7Q~WIbzpawgelEoU1QKA~.VPMBWK3vIgpU"
}
$Token = Invoke-RestMethod `
    -URI        "https://login.microsoftonline.com/specterdev.onmicrosoft.com/oauth2/v2.0/token" `
    -Method     POST `
    -Body       $Body

# List users in my tenant
$Users = Invoke-RestMethod `
    -Headers    @{Authorization = "Bearer $($Token.access_token)" } `
    -Uri        "https://graph.microsoft.com/v1.0/users/" `
    -Method     GET

$Users.value
```

Client execution

```
PS /Users/andyrobbins> $Body = @{
>>     Grant_Type      =   "client_credentials"
>>     Scope           =   "https://graph.microsoft.com/.default"
>>     client_Id       =   "5cb4d011-ef3d-4c4a-afe4-1189dd1808ae"
>>     Client_Secret   =   "3qh7Q~WIbzpawgelEoU1QKA~.VPMBWK3vIgpU"
>> }
PS /Users/andyrobbins> $Token = Invoke-RestMethod `
>>     -URI        "https://login.microsoftonline.com/specterdev.onmicrosoft.com/oauth2/v2.0/token" `
>>     -Method     POST `
>>     -Body       $Body
PS /Users/andyrobbins>
PS /Users/andyrobbins> # List users in my tenant
PS /Users/andyrobbins> $Users = Invoke-RestMethod `
>>     -Headers    @{Authorization = "Bearer $($Token.access_token)" } `
>>     -Uri        "https://graph.microsoft.com/v1.0/users/" `
>>     -Method     GET
PS /Users/andyrobbins>
```

The data

```
PS /Users/andyrobbins> $Users.value | Select -First 2

businessPhones    : {}
displayName       : Andew Chiles
givenName         : Andrew
jobTitle          : Simulation
mail              : achiles@specterdev.onmicrosoft.com
mobilePhone       :
officeLocation    :
preferredLanguage :
surname           : Chiles
userPrincipalName : achiles@specterdev.onmicrosoft.com
id                : d17d05dd-cbe9-41de-ac4d-93df58de0792

businessPhones    : {}
displayName       : Andy Robbins
givenName         : Andy
jobTitle          :
mail              :
mobilePhone       :
officeLocation    :
preferredLanguage :
surname           : Robbins
userPrincipalName : arobbins@specterdev.onmicrosoft.com
id                : 16571a9f-a0b4-476a-87e8-14dca511b9a3

PS /Users/andyrobbins>
```

```powershell
$Body = @{
    Grant_Type      =   "client_credentials"
    Scope           =   "https://graph.microsoft.com/.default"
    client_Id       =   "5cb4d011-ef3d-4c4a-afe4-1189dd1808ae"
    Client_Secret   =   "3qh7Q~WIbzpawgelEoU1QKA~.VPMBWK3vIgpU"
}
$Token = Invoke-RestMethod `
    -URI        "https://login.microsoftonline.com/specterdev.onmicrosoft.com/oauth2/v2.0/token" `
    -Method     POST `
    -Body       $Body

# Get list of our test service principals by their ID
$URI = "https://graph.microsoft.com/v1.0/users"
$Results = $null
$Users = $null
do {
    $Results = Invoke-RestMethod `
        -Headers @{Authorization = "Bearer $($Token.access_token)"} `
        -URI $URI `
        -UseBasicParsing `
        -Method "GET" `
        -ContentType "application/json"
    if ($Results.value) {
        $TestSPObjects += $Results.value
    } else {
        $TestSPObjects += $Results
    }
    $uri = $Results.'@odata.nextlink'
} until (!($uri))

```

Token gotchas

- JWTs facilitate stateless authentication but only partly facilitate stateless authorization
- But not all authorization is stored in the JWT

Token gotchas

- JWTs facilitate stateless authentication but only partly facilitate stateless authorization
- But not all authorization is stored in the JWT

- AzureAD roles and MS Graph roles are stored in JWTs
- But these tokens do not include AzureRM role assignments or the various other access control configs

Token gotchas

- JWTs facilitate stateless authentication but only partly facilitate stateless authorization
- But not all authorization is stored in the JWT

- AzureAD roles and MS Graph roles are stored in JWTs
- But these tokens do not include AzureRM role assignments or the various other access control configs

- Tl;dr: You can have more privileges than what your JWT states

```powershell
# Get a token for MS Graph API as the "MyCoolApp" service principal
$clientId       =   "4a2d703b-ec39-4fe4-98a9-e298d5a0f540"
$tenantName     =   "specterdev.onmicrosoft.com"
$clientSecret   =   "PYi7Q~LCQud5LChhdUYU9rMjYDQModARxDa0O"
$resource       =   "https://graph.microsoft.com/"

$Body = @{
    Grant_Type      =   "client_credentials"
    Scope           =   "https://graph.microsoft.com/.default"
    client_Id       =   $clientID
    Client_Secret   =   $clientSecret
}

$token = Invoke-RestMethod `
    -Uri        "https://login.microsoftonline.com/$TenantName/oauth2/v2.0/token" `
    -Method     POST `
    -Body       $Body

# $token.access_token is our bearer token
$token.access_token

# Now that we have a bearer token, we can interface with the MS Graph API
$URI        =   'https://graph.microsoft.com/v1.0/Groups/'
$Request    =   Invoke-RestMethod `
                    -Headers @{Authorization = "Bearer $($token.access_token)"} `
                    -URI $URI `
                    -Method GET
$Request.value | Select -First 1
```

```
PS C:\Users\andyrobbins> # Get a token for MS Graph API as the "MyCoolApp" service principal
PS C:\Users\andyrobbins> $clientId       =   "4a2d703b-ec39-4fe4-98a9-e298d5a0f540"
PS C:\Users\andyrobbins> $tenantName     =   "specterdev.onmicrosoft.com"
PS C:\Users\andyrobbins> $clientSecret   =   "PYi7Q~LCQud5LChhdUYU9rMjYDQModARxDa0O"
PS C:\Users\andyrobbins> $resource       =   "https://graph.microsoft.com/"
PS C:\Users\andyrobbins>
PS C:\Users\andyrobbins> $Body = @{
>>      Grant_Type      =   "client_credentials"
>>      Scope           =   "https://graph.microsoft.com/.default"
>>      client_Id       =   $clientID
>>      Client_Secret   =   $clientSecret
>> }
PS C:\Users\andyrobbins>
PS C:\Users\andyrobbins> $token = Invoke-RestMethod `
>>      -Uri        "https://login.microsoftonline.com/$TenantName/oauth2/v2.0/token" `
>>      -Method     POST `
>> -Body        $Body
PS C:\Users\andyrobbins>
PS C:\Users\andyrobbins> # $token.access_token is our bearer token
PS C:\Users\andyrobbins> $token.access_token
eyJ0eXAiOiJKV1QiLCJub25jZSI6Im5xRVdUQjZMOVJWcjNIY0RRaVA2VmpWaTFFeHFrcGlkRkhiUHBtUkFfVHciLCJhbGciOiJSUzI1NiIsIng1dCI6Ik1y
NS1BVWliZkJpaTdOZDFqQmViYXhib1hXMCIsImtpZCI6Ik1yNS1BVWliZkJpaTdOZDFqQmViYXhib1hXMCJ9.eyJhdWQiOiJodHRwczovL2dyYXBoLm1pY3J
vc29mdC5jb20iLCJpc3MiOiJodHRwczovL3N0cy53aW5kb3dzLm5ldC82YzEyYjBiMC1iMmNjLTRhNzMtODI1Mi0wYjk0YmZjYTIxNDUvIiwiaWF0IjoxNjQ
1NTcwNTU3LCJuYmYiOjE2NDU1NzA1NTcsImV4cCI6MTY0NTU3NDQ1NywiYWlvIjoiRTJaZ1lPQXhiNzNTVnUvMzBMbXhxRVZ2bmRKL0FBPT0iLCJhcHBfZGl
zcGxheW5hbWUiOiJNeUNvb2xBcHAiLCJhcHBpZCI6IjRhMmQ3MDNiLWVjMzktNGZlNC05OGE5LWUyOThkNWEwZjU0MCIsImFwcGlkYWNyIjoiMSIsImlkcCI
6Imh0dHBzOi8vc3RzLndpbmRvd3MubmV0LzZjMTJiMGIwLWIyY2MtNGE3My04MjUyLTBiOTRiZmNhMjE0NS8iLCJpZHR5cCI6ImFwcCIsIm9pZCI6ImQxNDY
0NjRmLTUyM2EtNGQyNC1iZmNlLTE3ZDIxNTY4NjQ3ZSIsInJoIjoiMC5BVkVBc0xBU2JNeXljMHFDVWd1VXY4b2hSUU1BQUFBQUFBQUF3QUFBQUFBQUFBQlJ
BQUEuIiwicm9sZXMiOlsiRGlyZWN0b3J5LlJlYWQuQWxsIl0sInN1YiI6ImQxNDY0NjRmLTUyM2EtNGQyNC1iZmNlLTE3ZDIxNTY4NjQ3ZSIsInRlbmFudF9
yZWdpb25fc2NvcGUiOiJOQSIsInRpZCI6IjZjMTJiMGIwLWIyY2MtNGE3My04MjUyLTBiOTRiZmNhMjE0NSIsInV0aSI6IkExSWFrWk9CVFV5czRzTndQSEF
UQWciLCJ2ZXIiOiIxLjAiLCJ3aWRzIjpbIjA5OTdhMWQwLTBkMWQtNGFjYi1iNDA4LWQ1Y2E3MzEyMWU5MCJdLCJ4bXNfdGNkdCI6MTU4ODYwMjg3M30.rWl
1jDm5z1EH3QQVCiva6Fd1z-FSpw6VZpFGOhETT-gJLY0LiMfTJwi8f9ghl4XUgiQGbazFMfx85-xbaPerbHGW3dCjFh1EfXXloLXB8TJ9GTvrJNGa0jX2Xj5
T6tAYUp75ci06BxPF8eXH0a-7f0q24hhKWcqG5QXhMkg4Dv5mtzQ7yrA4W0Fzc2XbPKALFq5Xz579zJyvfRAeNup_TkVv6s4h_eLEnA-dzxSvq8OuTXD7gmu
1d[illegible]vdQUCpHWikgZGB7AI4ovjmLcGueK0jZdAmDVjo7li93T1AOxf6UV9SJddk74bX7224DFi9zw
PS C:\Users\andyrobbins>
```

jwt.ms

jwt.ms

eyJ0eXAiOiJKV1QiLCJub25jZSI6Im5xRVdUQjZMOVJWcjNIY0RRaVA2VmpWaTFFeHFrcGlkRkhiUHBtUkFfVHciLCJhbGciOiJSUzI1NiIsIng1dCI6Ik1yNS1BVWliZkJpaTdOZDFqQmViYXhib1hXMCIsImtpZCI6Ik1yNS1BVWliZkJpaTdOZDFqQmViYXhib1hXMCJ9.eyJhdWQiOiJodHRwczovL2dyYXBoLm1pY3Jvc29mdC5jb20vLCJpc3MiOiJodHRwczovL3N0cy53aW5kb3dzLm5ldC82YzEyYjBiMC1iMmNjLTRhNzMtODI1Mi0wYjk0YmZjYTIxNDUvIiwiaWF0IjoxNjQ1NTcwNTU3LCJuYmYiOjE2NDU1NzA1NTcsImV4cCI6MTY0NTU3NDQ1NywiYWlvIjoiRTJaZ1lPQXhiNzNTVnUvMzBMbXhxRVZ2bmRKL0FBPT0iLCJhcHBfZGlzcGxheWSbWUiOiJNeUnVb2xBcHAiLCJhcHBpZCI6IjRhMmQ3MDNiLWVjMzktNGZlNC05OGE5LWUyOThkNWEwZjU0MCIsImFwcGlkYWNyIjoiMSIsImlkcCI6Imh0dHBzOi8vc3RzLndpbmRvd3MubmV0LzZjMTJiMGIwLWIyY2MtNGE3My04MjUyLTBiOTRiZmNhMjE0NS8iLCJpZHR5cCI6ImFwcCIsIm9pZCI6ImQxNDY0NjRmLTUyM2EtNGQyNC1iZmNlLTE3ZDIxNTY4NjQ3ZSIsInJoIjoiMC5BVkVBc0xBU2JNeXljMHFDVWd1VXY4b2hSUU1BQUFBQUFBQUF3QUFBQUFBQUFBQlJBQUEuIiwicm9sZXMiOlsiRGlyZWN0b3J5LlJlYWQuQWxsIl0sInN1YiI6ImQxNDY0NjRmLTUyM2EtNGQyNC1iZmNlLTE3ZDIxNTY4NjQ3ZSIsInRlbmFudF9yZWdpb25fc2NvcGUiOiJOQSIsInRpZCI6IjZjMTJiMGIwLWIyY2MtNGE3My04MjUyLTBiOTRiZmNhMjE0NSIsInV0aSI6IkExSWFrWk9CVFV5czRzTndQSEFUQWciLCJ2ZXIiOiIxLjAiLCJ3aWRzIjpbIjA5OTdhMWQwLTBkMWQtNGFjYi1iNDA4LWQ1Y2E3MzEyMWU5MCJdLCJ4bXNfdGNkdCI6MTU4ODYwMjg3M30.rWl1jDm5z1EH3QQVCiva6Fd1z-FSpw6VZpFGOhETT-gJLY0LiMfTJwi8f9ghl4XUgiQGbazFMfx85-xbaPerbHGW3dCjFh1EfXXloLXB8TJ9GTvrJNGa0jX2Xj5T6tAYUp75ci06BxPF8eXH0a-7f0q24hhKWcqG5QXhMkg4Dv5mtzQ7yrA4W0Fzc2XbPKALFq5Xz579zJyvfRAeNup_TkVv6s4h_eLEnA-dzxSvq8OuTXD7gmu1dFAcVf4rJARgpGRmL1aR5ObePbvdQUCpHWikgZGB7AI4ovjmLcGueK0jZdAmDVjo7li93T1AOxf6UV9SJddk74bX7224DFi9zw

This token was issued by Azure Active Directory.

Decoded Token | Claims


```
{
  "typ": "JWT",
  "nonce": "nqEWTB6L9RVr3HcDQiP6VjVi1ExqkpidFHbPpmRA_Tw",
  "alg": "RS256",
  "x5t": "Mr5-AUibfBii7Nd1jBebaxboXW0",
  "kid": "Mr5-AUibfBii7Nd1jBebaxboXW0"
}.{
  "aud": "https://graph.microsoft.com",
  "iss": "https://sts.windows.net/6c12b0b0-b2cc-4a73-8252-0b94bfca2145/",
  "iat": 1645570557,
  "nbf": 1645570557,
  "exp": 1645574457,
  "aio": "E2ZgYOAxb73SVu/30LmxqEVvndJ/AA==",
  "app_displayname": "MyCoolApp",
  "appid": "4a2d703b-ec39-4fe4-98a9-e298d5a0f540",
  "appidacr": "1",
  "idp": "https://sts.windows.net/6c12b0b0-b2cc-4a73-8252-0b94bfca2145/",
  "idtyp": "app",
  "oid": "d146464f-523a-4d24-bfce-17d21568647e",
  "rh": "0.AVEAsLASbMyyc0qCUguUv8ohRQMAAAAAAAAAwAAAAAAAABRAAA.",
  "roles": [
    "Directory.Read.All"
  ],
```

Audience

Decoded Token | Claims


```
{
  "typ": "JWT",
  "nonce": "nqEWTB6L9RVr3HcDQiP6VjVi1ExqkpidFHbPpmRA_Tw",
  "alg": "RS256",
  "x5t": "Mr5-AUibfBii7Nd1jBebaxboXW0",
  "kid": "Mr5-AUibfBii7Nd1jBebaxboXW0"
}.{
  "aud": "https://graph.microsoft.com",
  "iss": "https://sts.windows.net/6c12b0b0-b2cc-4a73-8252-0b94bfca2145/",
  "iat": 1645570557,
  "nbf": 1645570557,
  "exp": 1645574457,
  "aio": "E2ZgYOAxb73SVu/30LmxqEVvndJ/AA==",
  "app_displayname": "MyCoolApp",
  "appid": "4a2d703b-ec39-4fe4-98a9-e298d5a0f540",
  "appidacr": "1",
  "idp": "https://sts.windows.net/6c12b0b0-b2cc-4a73-8252-0b94bfca2145/",
  "idtyp": "app",
  "oid": "d146464f-523a-4d24-bfce-17d21568647e",
  "rh": "0.AVEAsLASbMyyc0qCUguUv8ohRQMAAAAAAAAAwAAAAAAAAABRAAA.",
  "roles": [
    "Directory.Read.All"
  ],
  "sub": "d146464f-523a-4d24-bfce-17d21568647e",
  "tenant_region_scope": "NA",
  "tid": "6c12b0b0-b2cc-4a73-8252-0b94bfca2145",
  "uti": "A1IakZOBTUys4sNwPHATAg",
  "ver": "1.0",
  "wids": [
    "0997a1d0-0d1d-4acb-b408-d5ca73121e90"
  ],
  "xms_tcdt": 1588602873
}.[Signature]
```

Issuer

Decoded Token | Claims


```
{
  "typ": "JWT",
  "nonce": "nqEWTB6L9RVr3HcDQiP6VjVi1ExqkpidFHbPpmRA_Tw",
  "alg": "RS256",
  "x5t": "Mr5-AUibfBii7Nd1jBebaxboXW0",
  "kid": "Mr5-AUibfBii7Nd1jBebaxboXW0"
}.{
  "aud": "https://graph.microsoft.com",
  "iss": "https://sts.windows.net/6c12b0b0-b2cc-4a73-8252-0b94bfca2145/",
  "iat": 1645570557,
  "nbf": 1645570557,
  "exp": 1645574457,
  "aio": "E2ZgYOAxb73SVu/30LmxqEVvndJ/AA==",
  "app_displayname": "MyCoolApp",
  "appid": "4a2d703b-ec39-4fe4-98a9-e298d5a0f540",
  "appidacr": "1",
  "idp": "https://sts.windows.net/6c12b0b0-b2cc-4a73-8252-0b94bfca2145/",
  "idtyp": "app",
  "oid": "d146464f-523a-4d24-bfce-17d21568647e",
  "rh": "0.AVEAsLASbMyyc0qCUguUv8ohRQMAAAAAAAAAwAAAAAAAAABRAAA.",
  "roles": [
    "Directory.Read.All"
  ],
  "sub": "d146464f-523a-4d24-bfce-17d21568647e",
  "tenant_region_scope": "NA",
  "tid": "6c12b0b0-b2cc-4a73-8252-0b94bfca2145",
  "uti": "A1IakZOBTUys4sNwPHATAg",
  "ver": "1.0",
  "wids": [
    "0997a1d0-0d1d-4acb-b408-d5ca73121e90"
  ],
  "xms_tcdt": 1588602873
}.[Signature]
```

MS Graph scoped App Roles

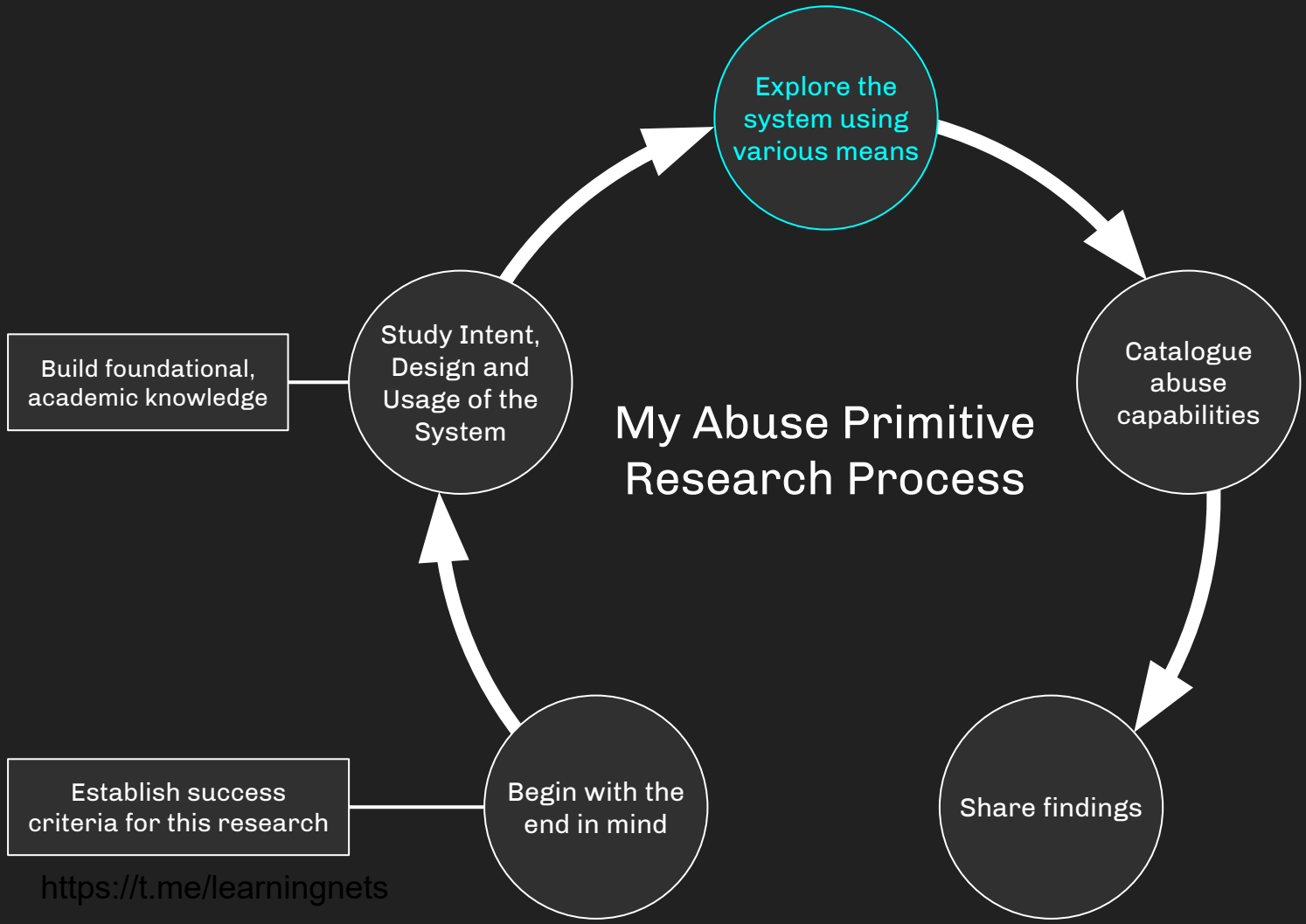
Decoded Token | Claims

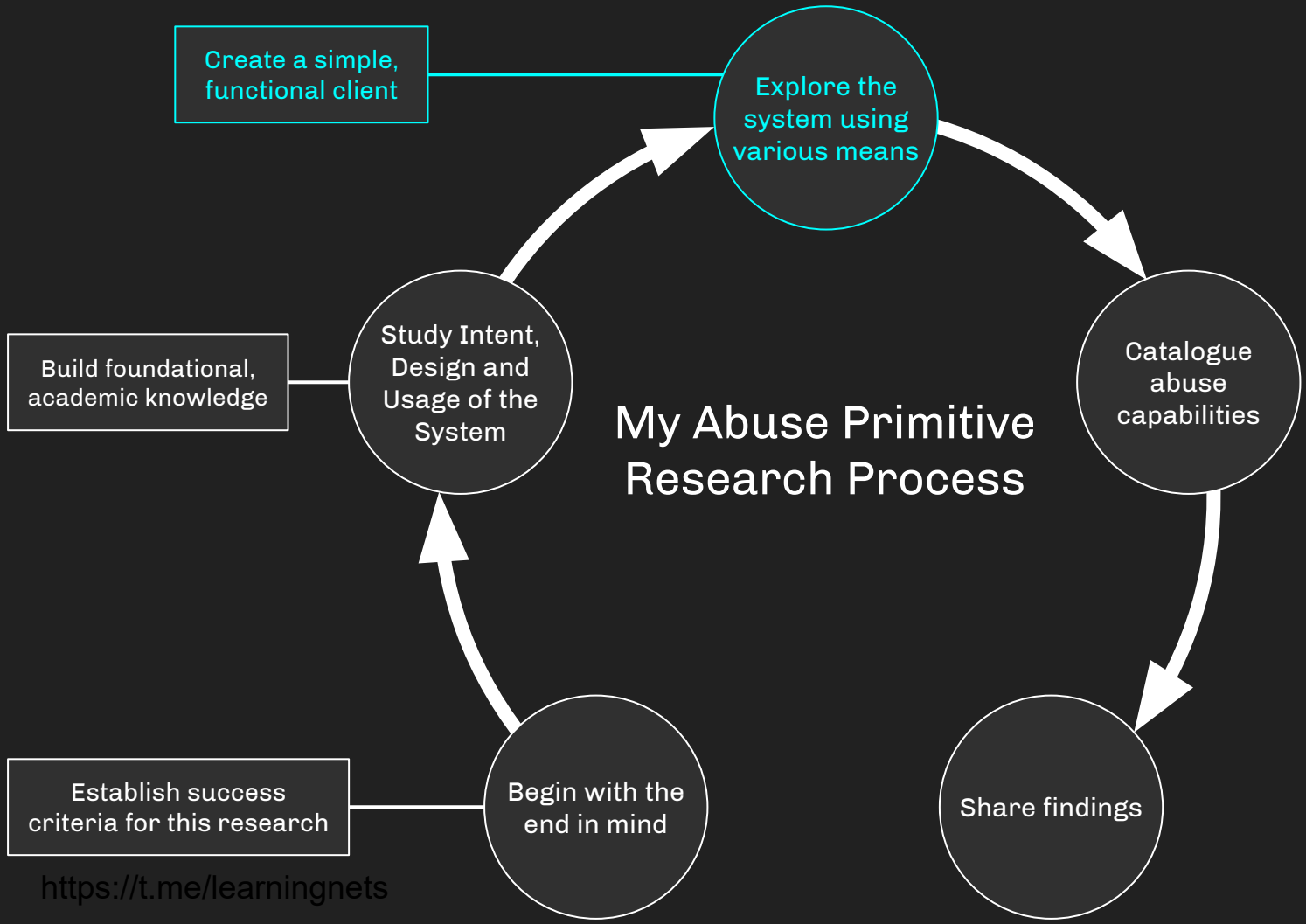

```
{
  "typ": "JWT",
  "nonce": "nqEWTB6L9RVr3HcDQiP6VjVi1ExqkpidFHbPpmRA_Tw",
  "alg": "RS256",
  "x5t": "Mr5-AUibfBii7Nd1jBebaxboXW0",
  "kid": "Mr5-AUibfBii7Nd1jBebaxboXW0"
}.{
  "aud": "https://graph.microsoft.com",
  "iss": "https://sts.windows.net/6c12b0b0-b2cc-4a73-8252-0b94bfca2145/",
  "iat": 1645570557,
  "nbf": 1645570557,
  "exp": 1645574457,
  "aio": "E2ZgYOAxb73SVu/30LmxqEVvndJ/AA==",
  "app_displayname": "MyCoolApp",
  "appid": "4a2d703b-ec39-4fe4-98a9-e298d5a0f540",
  "appidacr": "1",
  "idp": "https://sts.windows.net/6c12b0b0-b2cc-4a73-8252-0b94bfca2145/",
  "idtyp": "app",
  "oid": "d146464f-523a-4d24-bfce-17d21568647e",
  "rh": "0.AVEAsLASbMyyc0qCUguUv8ohRQMAAAAAAAAAwAAAAAAAAABRAAA.",
  "roles": [
    "Directory.Read.All"
  ],
  "sub": "d146464f-523a-4d24-bfce-17d21568647e",
  "tenant_region_scope": "NA",
  "tid": "6c12b0b0-b2cc-4a73-8252-0b94bfca2145",
  "uti": "A1IakZOBTUys4sNwPHATAg",
  "ver": "1.0",
  "wids": [
    "0997a1d0-0d1d-4acb-b408-d5ca73121e90"
  ],
  "xms_tcdt": 1588602873
}.[Signature]
```

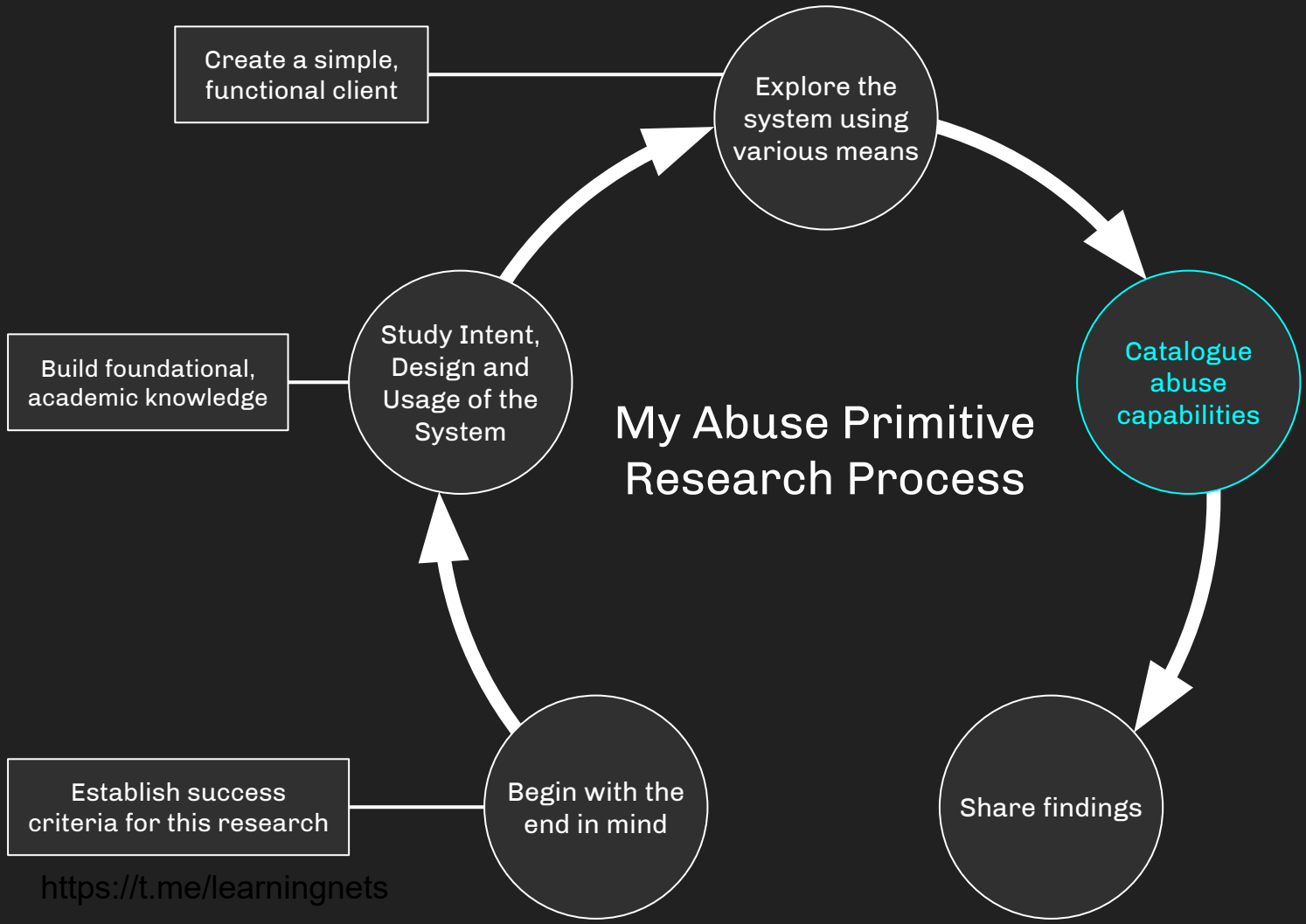
Decoded Token | Claims

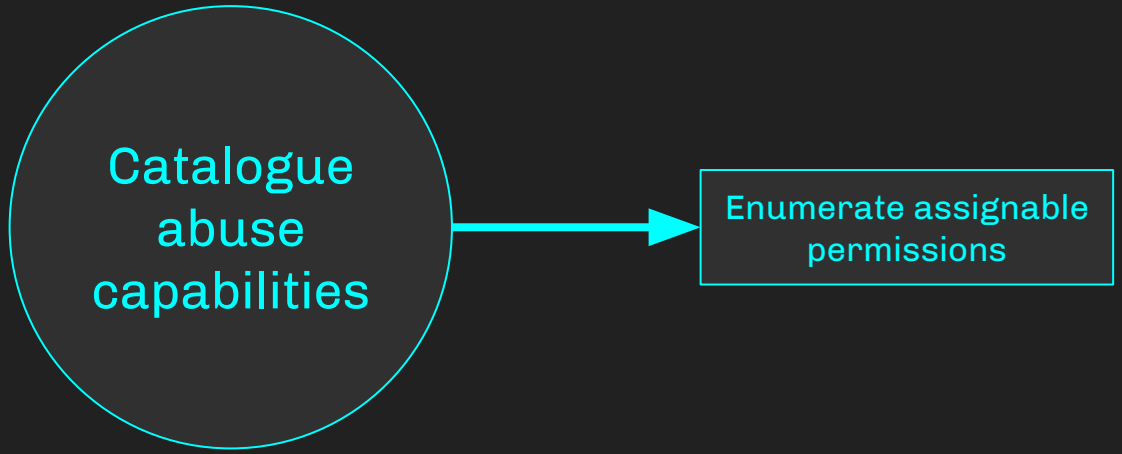

```
{
  "typ": "JWT",
  "nonce": "nqEWTB6L9RVr3HcDQiP6VjVi1ExqkpidFHbPpmRA_Tw",
  "alg": "RS256",
  "x5t": "Mr5-AUibfBii7Nd1jBebaxboXW0",
  "kid": "Mr5-AUibfBii7Nd1jBebaxboXW0"
}.{
  "aud": "https://graph.microsoft.com",
  "iss": "https://sts.windows.net/6c12b0b0-b2cc-4a73-8252-0b94bfca2145/",
  "iat": 1645570557,
  "nbf": 1645570557,
  "exp": 1645574457,
  "aio": "E2ZgYOAxb73SVu/30LmxqEVvndJ/AA==",
  "app_displayname": "MyCoolApp",
  "appid": "4a2d703b-ec39-4fe4-98a9-e298d5a0f540",
  "appidacr": "1",
  "idp": "https://sts.windows.net/6c12b0b0-b2cc-4a73-8252-0b94bfca2145/",
  "idtyp": "app",
  "oid": "d146464f-523a-4d24-bfce-17d21568647e",
  "rh": "0.AVEAsLASbMyyc0qCUguUv8ohRQMAAAAAAAAAwAAAAAAAAABRAAA.",
  "roles": [
    "Directory.Read.All"
  ],
  "sub": "d146464f-523a-4d24-bfce-17d21568647e",
  "tenant_region_scope": "NA",
  "tid": "6c12b0b0-b2cc-4a73-8252-0b94bfca2145",
  "uti": "A1IakZOBTUys4sNwPHATAg",
  "ver": "1.0",
  "wids": [
    "0997a1d0-0d1d-4acb-b408-d5ca73121e90"
  ],
  "xms_tcdt": 1588602873
}.[Signature]
```


AzureAD Roles





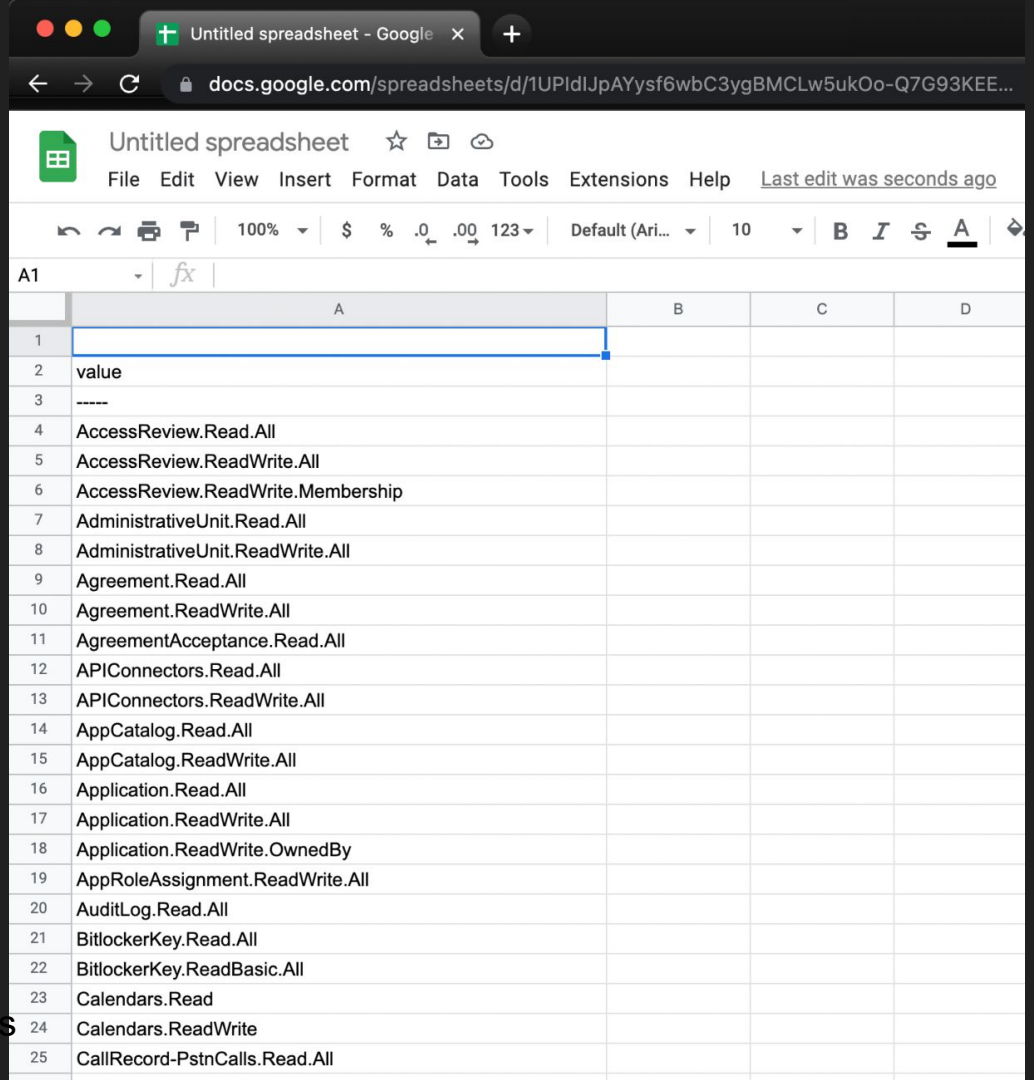


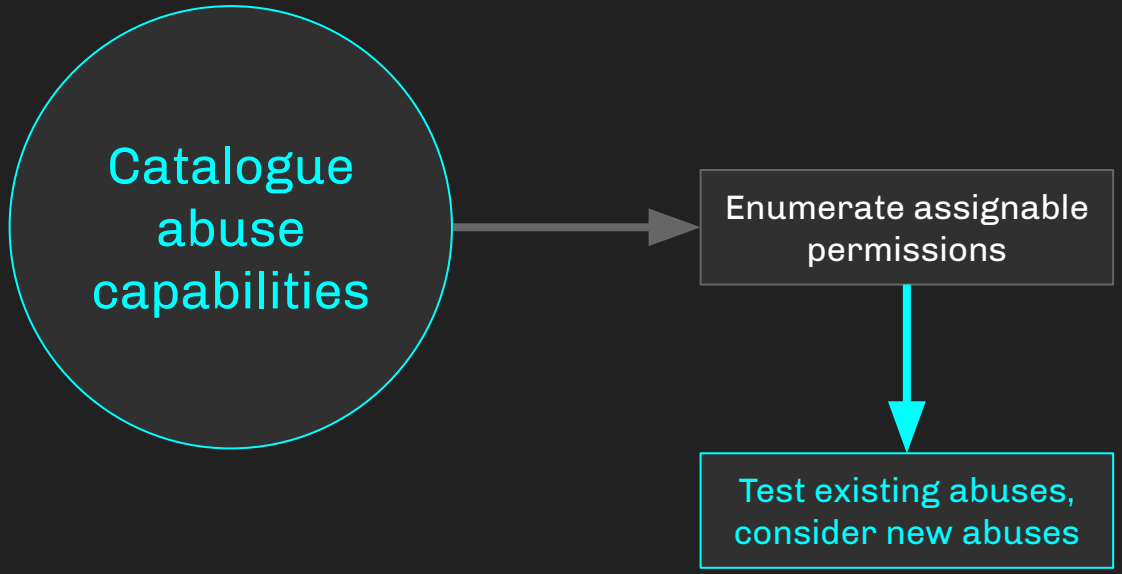


```
# Collect all service principal objects and put them into $ServicePrincipals
$URI = "https://graph.microsoft.com/v1.0/servicePrincipals"
$Results = $null
$ServicePrincipals = $null
do {
    $Results = Invoke-RestMethod `
        -Headers @{Authorization = "Bearer $($token.access_token)" } `
        -URI $URI `
        -UseBasicParsing `
        -Method "GET" `
        -ContentType "application/json"
    if ($Results.value) {
        $ServicePrincipals += $Results.value
    } else {
        $ServicePrincipals += $Results
    }
    $uri = $Results.'@odata.nextlink'
} until (!($uri))
```

C:\Users\andyrobbins\AppData\

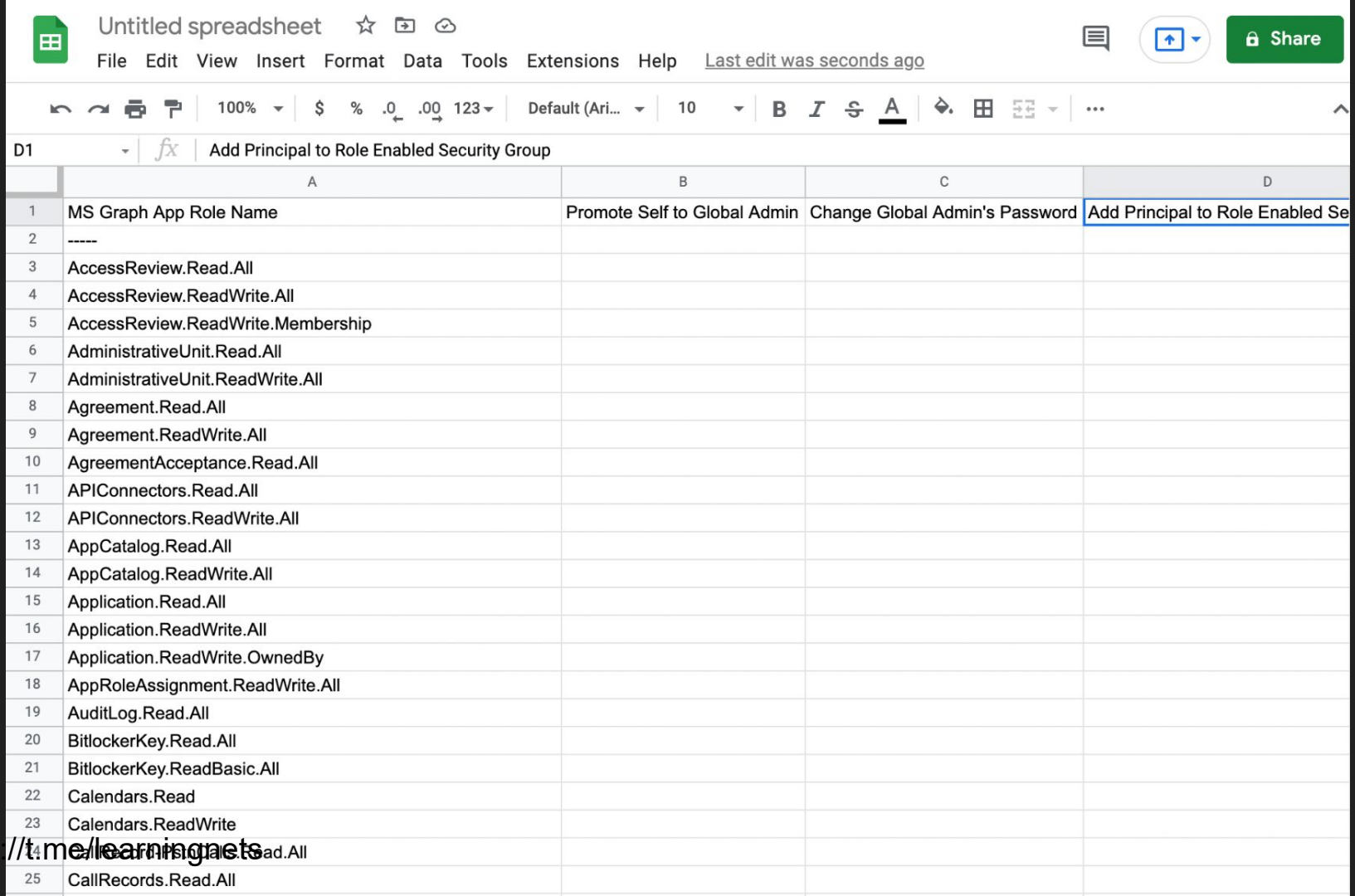
```
PS C:\Users\andyrobbins> $ServicePrincipals | ?{$_.appId -Match '0000000
 appRoles | sort-object -property value | select value

value
-----
AccessReview.Read.All
AccessReview.ReadWrite.All
AccessReview.ReadWrite.Membership
AdministrativeUnit.Read.All
AdministrativeUnit.ReadWrite.All
Agreement.Read.All
Agreement.ReadWrite.All
AgreementAcceptance.Read.All
APIConnectors.Read.All
APIConnectors.ReadWrite.All
AppCatalog.Read.All
AppCatalog.ReadWrite.All
Application.Read.All
Application.ReadWrite.All
Application.ReadWrite.OwnedBy
AppRoleAssignment.ReadWrite.All
AuditLog.Read.All
```



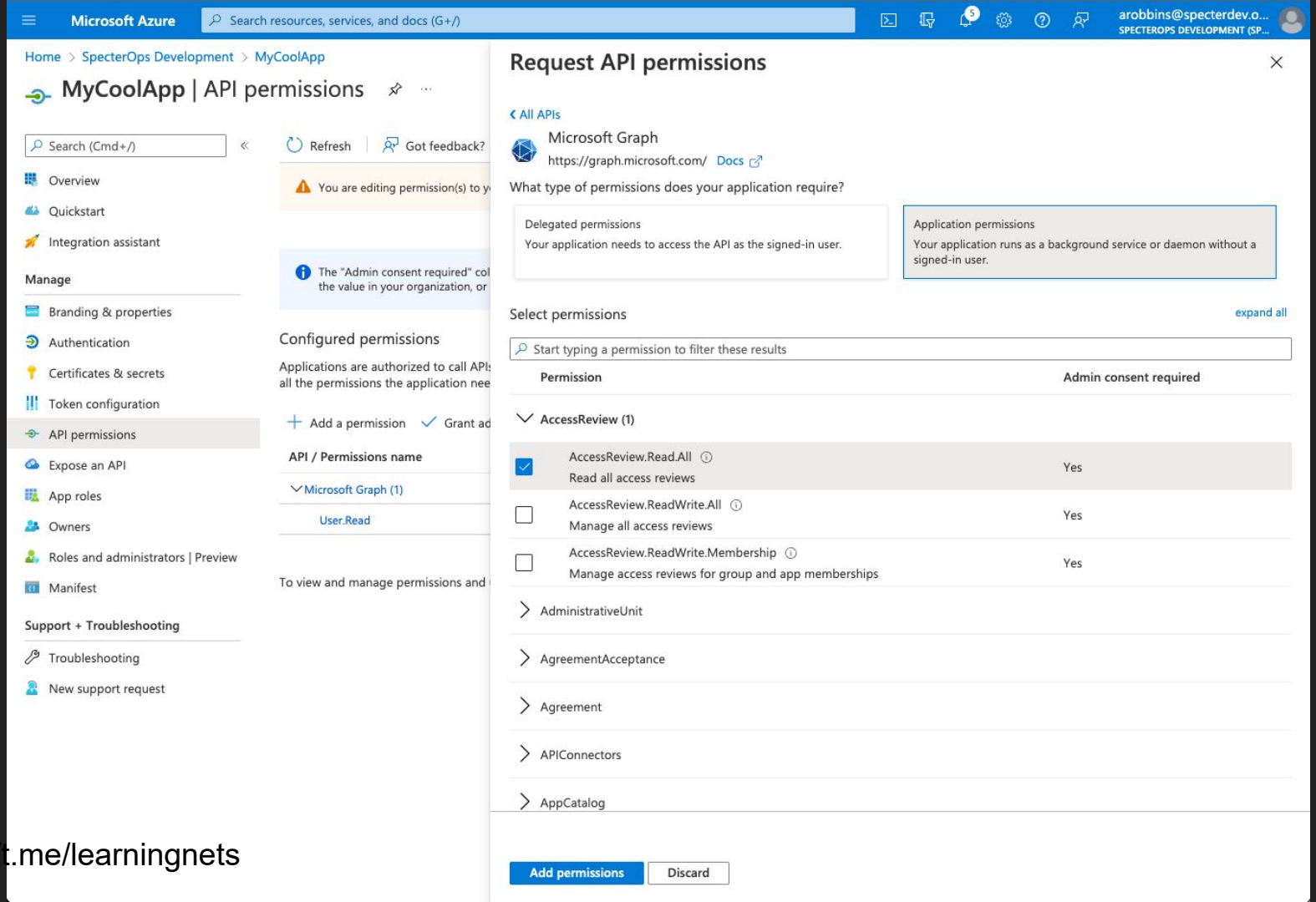


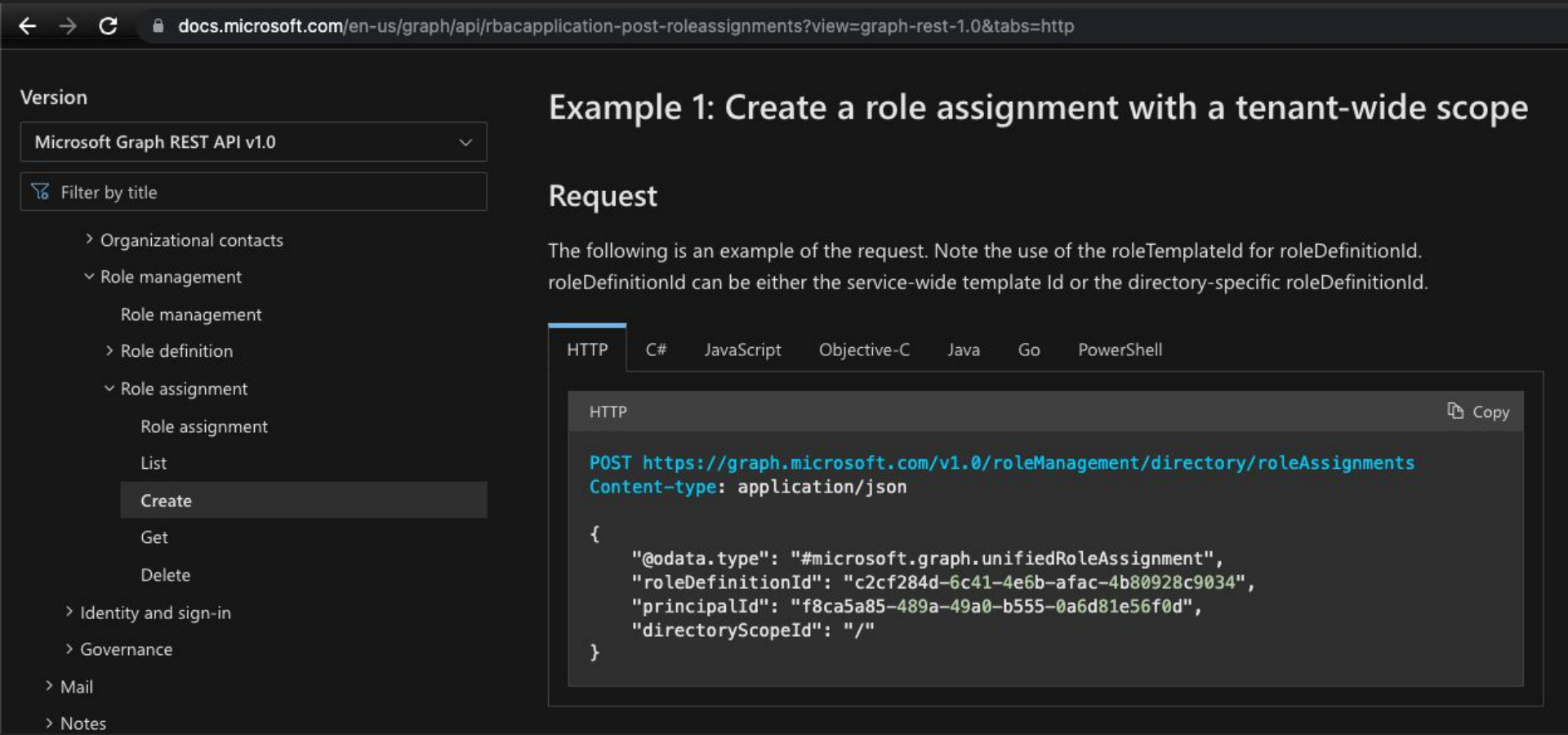
Application permissions

Permission	Display String	Description	Admin Consent Required
Directory.Read.All	Read directory data	Allows the app to read data in your organization's directory, such as users, groups and apps, without a signed-in user.	Yes
Directory.ReadWrite.All	Read and write directory data	Allows the app to read and write data in your organization's directory, such as users, and groups, without a signed-in user. Does not allow user or group deletion.	Yes



	A	B	C	D
1	MS Graph App Role Name	Promote Self to Global Admin	Change Global Admin's Password	Add Principal to Role Enabled Se
2	-----			
3	AccessReview.Read.All			
4	AccessReview.ReadWrite.All			
5	AccessReview.ReadWrite.Membership			
6	AdministrativeUnit.Read.All			
7	AdministrativeUnit.ReadWrite.All			
8	Agreement.Read.All			
9	Agreement.ReadWrite.All			
10	AgreementAcceptance.Read.All			
11	APIConnectors.Read.All			
12	APIConnectors.ReadWrite.All			
13	AppCatalog.Read.All			
14	AppCatalog.ReadWrite.All			
15	Application.Read.All			
16	Application.ReadWrite.All			
17	Application.ReadWrite.OwnedBy			
18	AppRoleAssignment.ReadWrite.All			
19	AuditLog.Read.All			
20	BitlockerKey.Read.All			
21	BitlockerKey.ReadBasic.All			
22	Calendars.Read			
23	Calendars.ReadWrite			
24	CallRecords.Read.All			
25	CallRecords.Read.All			





```
PS C:\Users\andyrobbins> # Try to grant "MyCoolApp" the "Global Admin" role:
PS C:\Users\andyrobbins> $body = @{
>>     "@odata.type" = "#microsoft.graph.unifiedRoleAssignment"
>>     principalId = "d146464f-523a-4d24-bfce-17d21568647e"
>>     roleDefinitionId = "62e90394-69f5-4237-9190-012177145e10"
>>     directoryScopeId = "/"
>> }
PS C:\Users\andyrobbins> $req = $null
PS C:\Users\andyrobbins> $req = Invoke-RestMethod -Headers @{Authorization = "Bearer $($token.access_token)" } `
>>     -Uri "https://graph.microsoft.com/v1.0/roleManagement/directory/roleAssignments" `
>>     -Method POST `
>>     -Body $($body | ConvertTo-Json) `
>>     -ContentType 'application/json'
Invoke-RestMethod : The remote server returned an error: (403) Forbidden.
At line:1 char:8
+ $req = Invoke-RestMethod -Headers @{Authorization = "Bearer $($token. ...
+        ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : InvalidOperation: (System.Net.HttpWebRequest:HttpWebRequest) [Invoke-RestMethod], WebExc
   eption
    + FullyQualifiedErrorId : WebCmdletWebResponseException,Microsoft.PowerShell.Commands.InvokeRestMethodCommand
```

Untitled spreadsheet

Saved to Drive

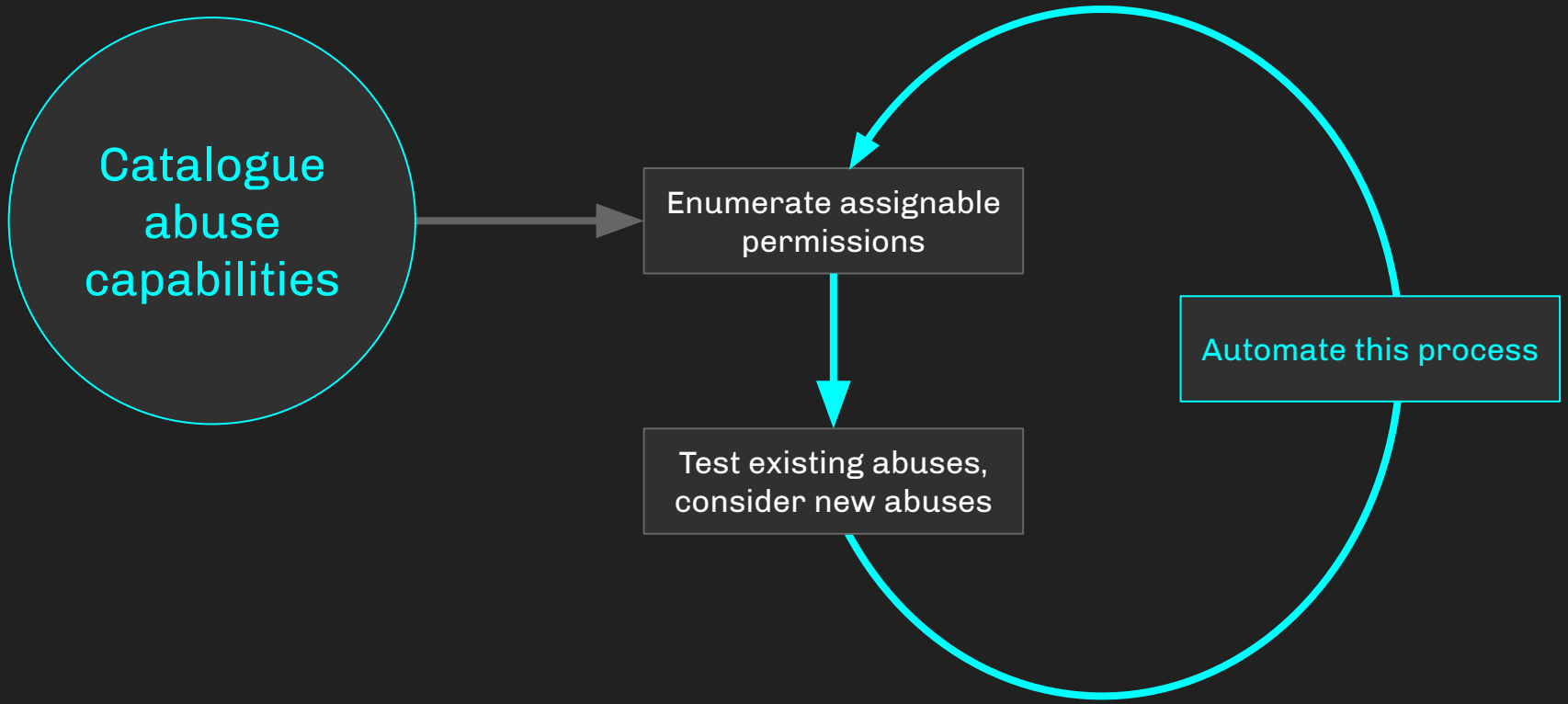
File Edit View Insert Format Data Tools Extensions Help Last edit was seconds ago

Share

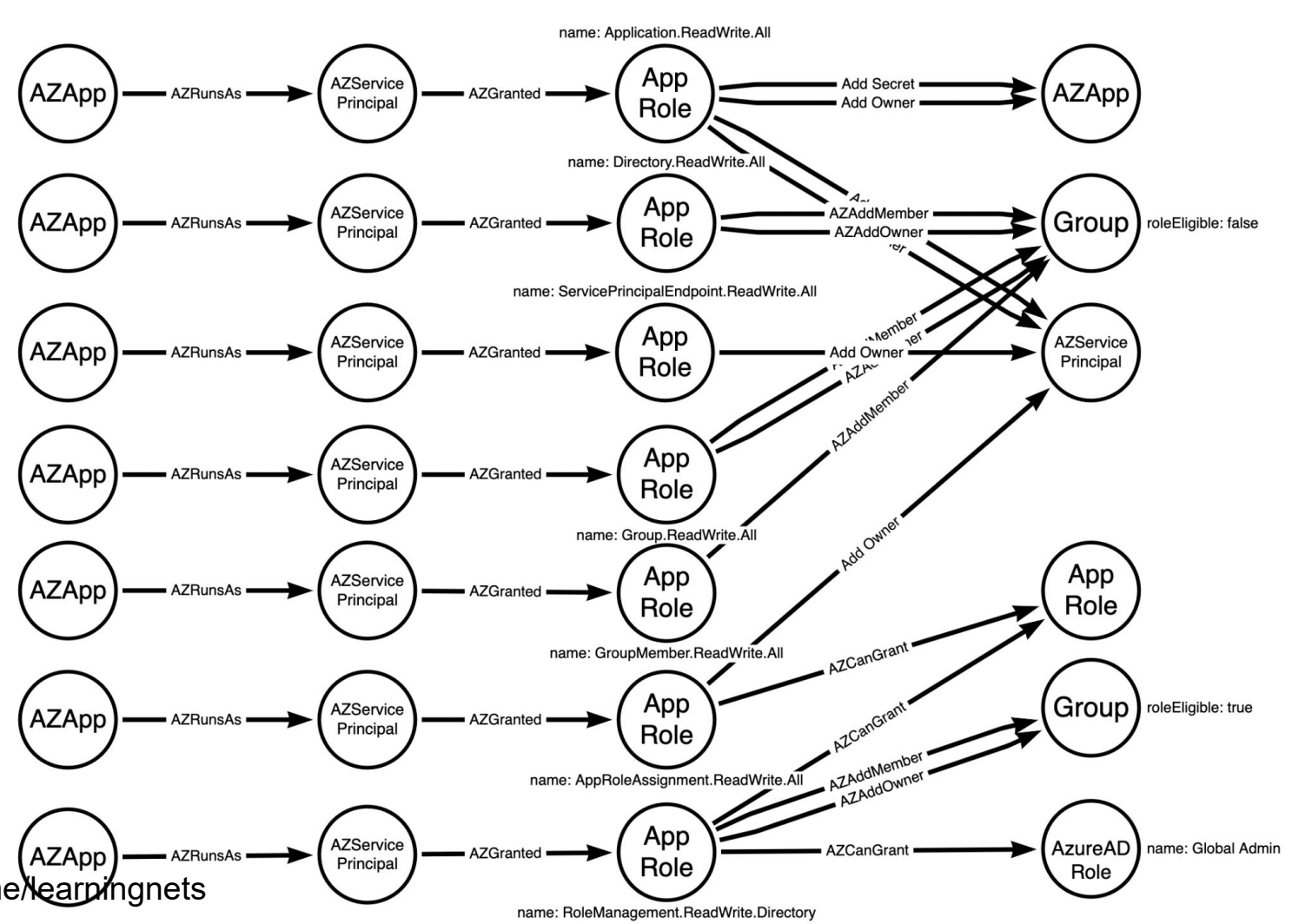
100% $ % .0 .00 123 Default (Ari... 10

B4

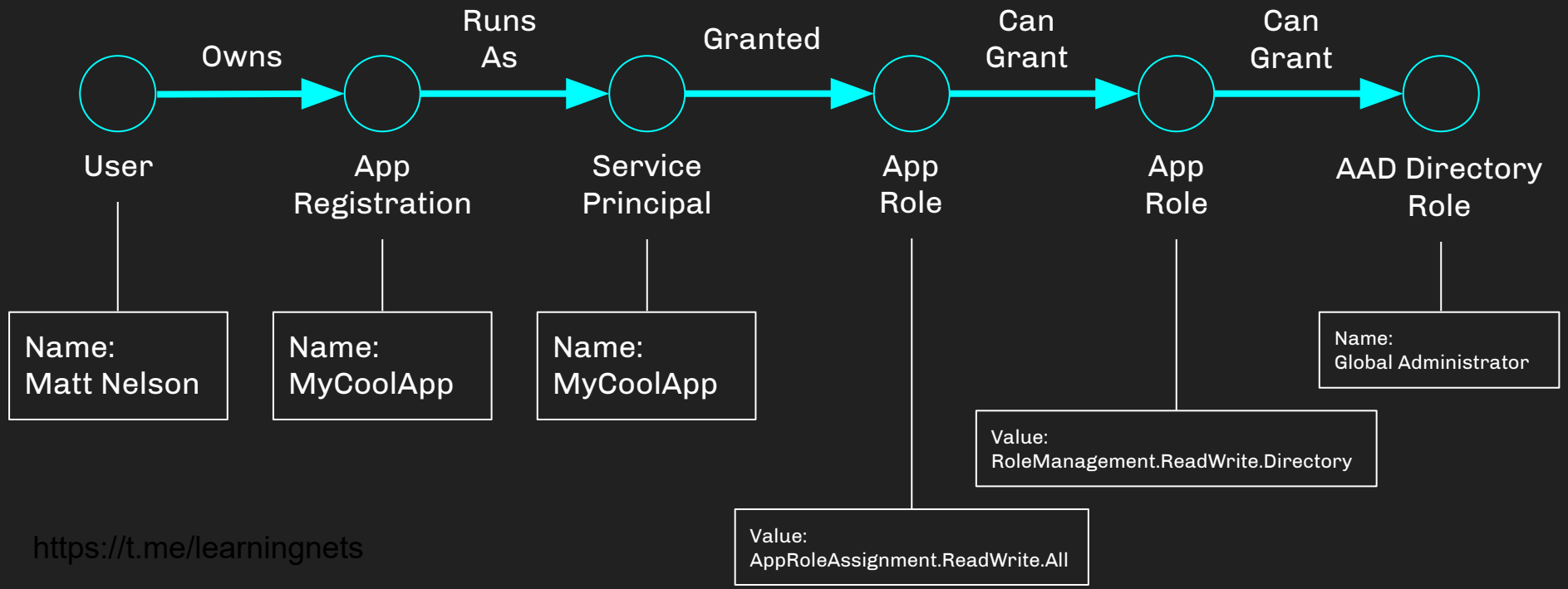
	A	B	C	D
1	MS Graph App Role Name	Promote Self to Global Admin	Change Global Admin's Password	Add Principal to Role Enabled Se
2	-----			
3	AccessReview.Read.All	No		
4	AccessReview.ReadWrite.All			
5	AccessReview.ReadWrite.Membership			
6	AdministrativeUnit.Read.All			
7	AdministrativeUnit.ReadWrite.All			
8	Agreement.Read.All			
9	Agreement.ReadWrite.All			
10	AgreementAcceptance.Read.All			
11	APIConnectors.Read.All			
12	APIConnectors.ReadWrite.All			
13	AppCatalog.Read.All			
14	AppCatalog.ReadWrite.All			
15	Application.Read.All			
16	Application.ReadWrite.All			
17	Application.ReadWrite.OwnedBy			
18	AppRoleAssignment.ReadWrite.All			
19	AuditLog.Read.All			
20	BitlockerKey.Read.All			
21	BitlockerKey.ReadBasic.All			
22	Calendars.Read			
23	Calendars.ReadWrite			
24	CallRecords.Read.All			
25	CallRecords.Read.All			



	Is that app permission abusable?											
	Abuses											
	Groups				Service Principals / Apps				Directory Roles		App Roles	
	Add owner to role eligible group	Add member to role-eligible group	Add member to non-role eligible group	Add owner to non role-eligible group	Add secret to SP	Add owner to SP	Add secret to App Reg	Add Owner to App Reg	Promote a user to Global Admin		Grant an app role	
Microsoft Graph API Permissions:												
Application.ReadWrite.All	No	No	No	No	Yes	Yes	Yes	Yes	No		No	
AppRoleAssignment.ReadWrite.All	No	No	No	No	No	Yes	No	No	No		Yes	
DelegatedPermissionGrant.ReadWrite.All	No	No	No	No	No	No	No	No	No		No	
Directory.ReadWrite.All	No	No	Yes	Yes	No	No	No	No	No		No	
Group.ReadWrite.All	No	No	Yes	Yes	No	No	No	No	No		No	
GroupMember.ReadWrite.All	No	No	Yes	No	No	No	No	No	No		No	
RoleManagement.ReadWrite.Directory	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes		Yes	
ServicePrincipalEndpoint.ReadWrite.All	No	No	No	No	No	Yes	No	No	No		No	



Example Attack Path



Permission types

The Microsoft identity platform supports two types of permissions: *delegated permissions* and *application permissions*.

- **Delegated permissions** are used by apps that have a signed-in user present. For these apps, either the user or an administrator consents to the permissions that the app requests. The app is delegated with the permission to act as a signed-in user when it makes calls to the target resource.

  Some delegated permissions can be consented to by nonadministrators. But some high-privileged permissions require administrator consent. To learn which administrator roles can consent to delegated permissions, see Administrator role permissions in Azure Active Directory (Azure AD).

- **Application permissions** are used by apps that run without a signed-in user present, for example, apps that run as background services or daemons. Only an administrator can consent to application permissions.

Permission types

The Microsoft identity platform supports two types of permissions: *delegated permissions* and *application permissions*.

- **Delegated permissions** are used by apps that have a signed-in user present. For these apps, either the user or an administrator consents to the permissions that the app requests. The app is delegated with the permission to act as a signed-in user when it makes calls to the target resource.

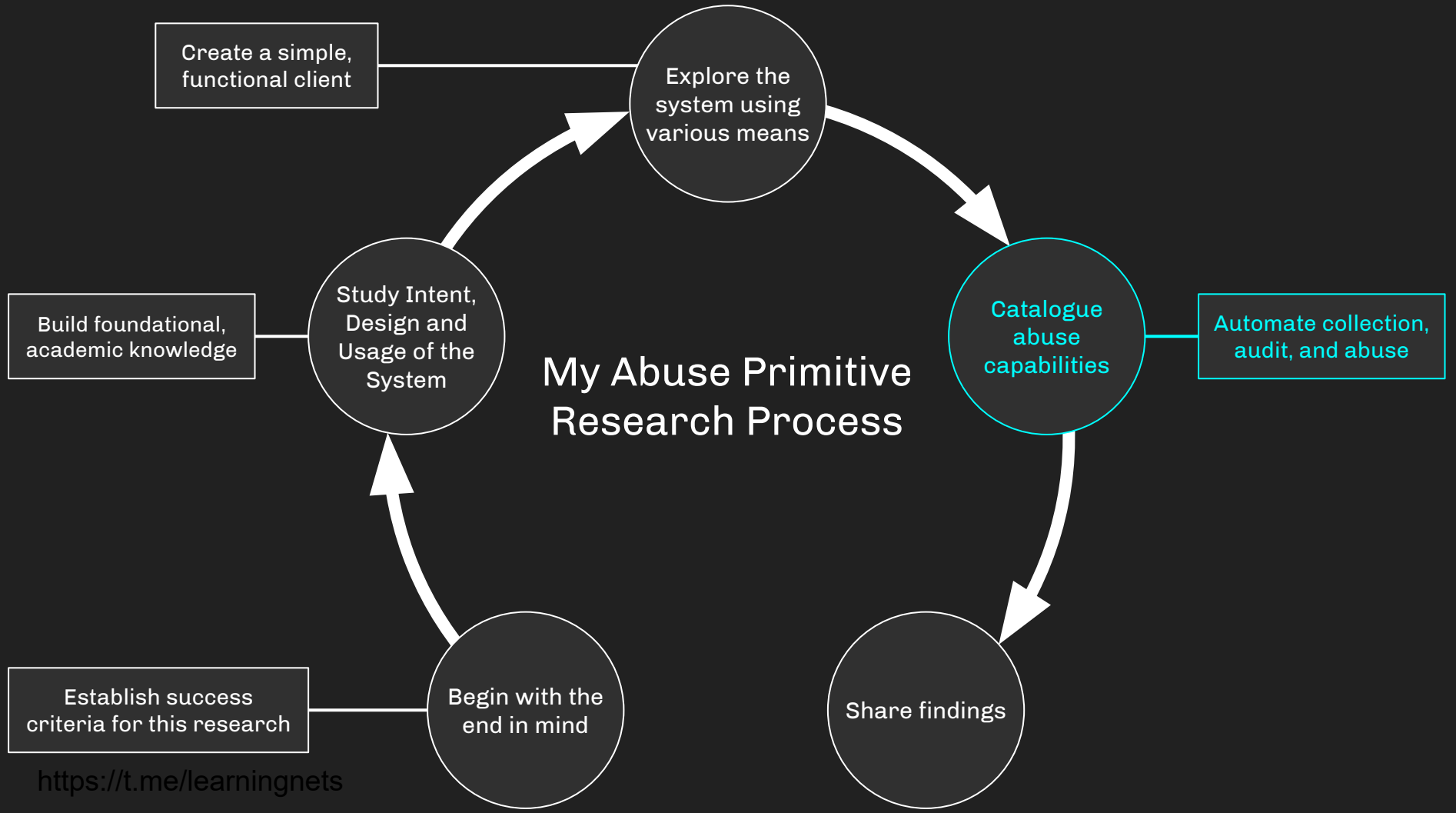
  Some delegated permissions can be consented to by nonadministrators. But some high-privileged permissions require administrator consent. To learn which administrator roles can consent to delegated permissions, see Administrator role permissions in Azure Active Directory (Azure AD).

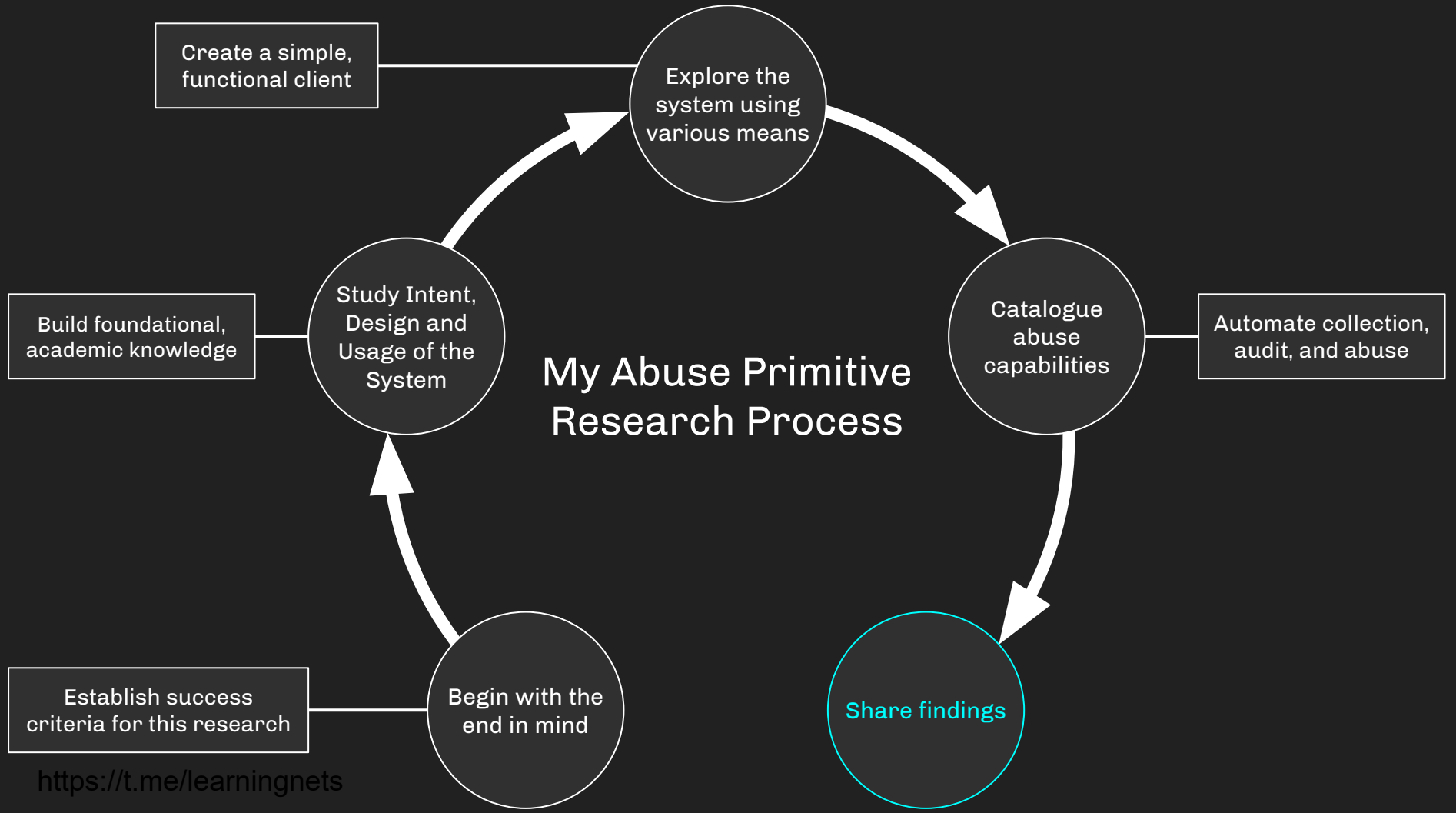
- **Application permissions** are used by apps that run without a signed-in user present, for example, apps that run as background services or daemons. Only an administrator can consent to application permissions.

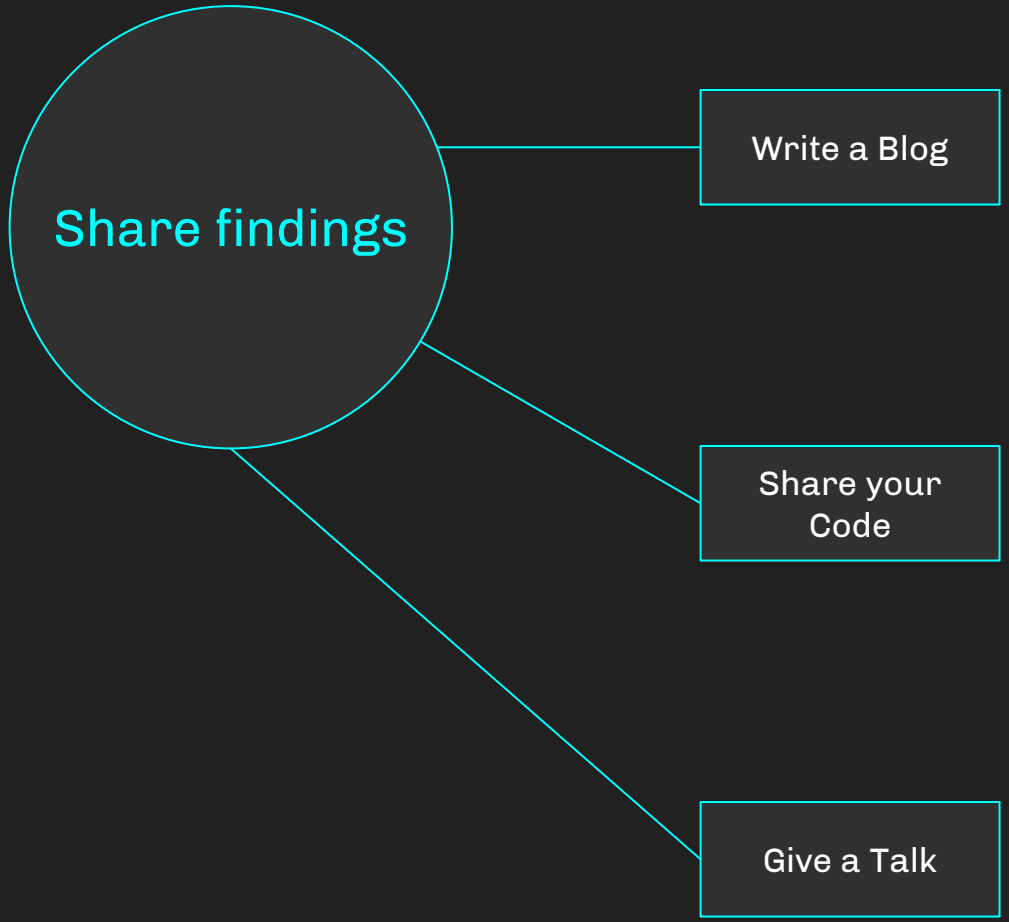
Coming soon: Atomic Azure Tests

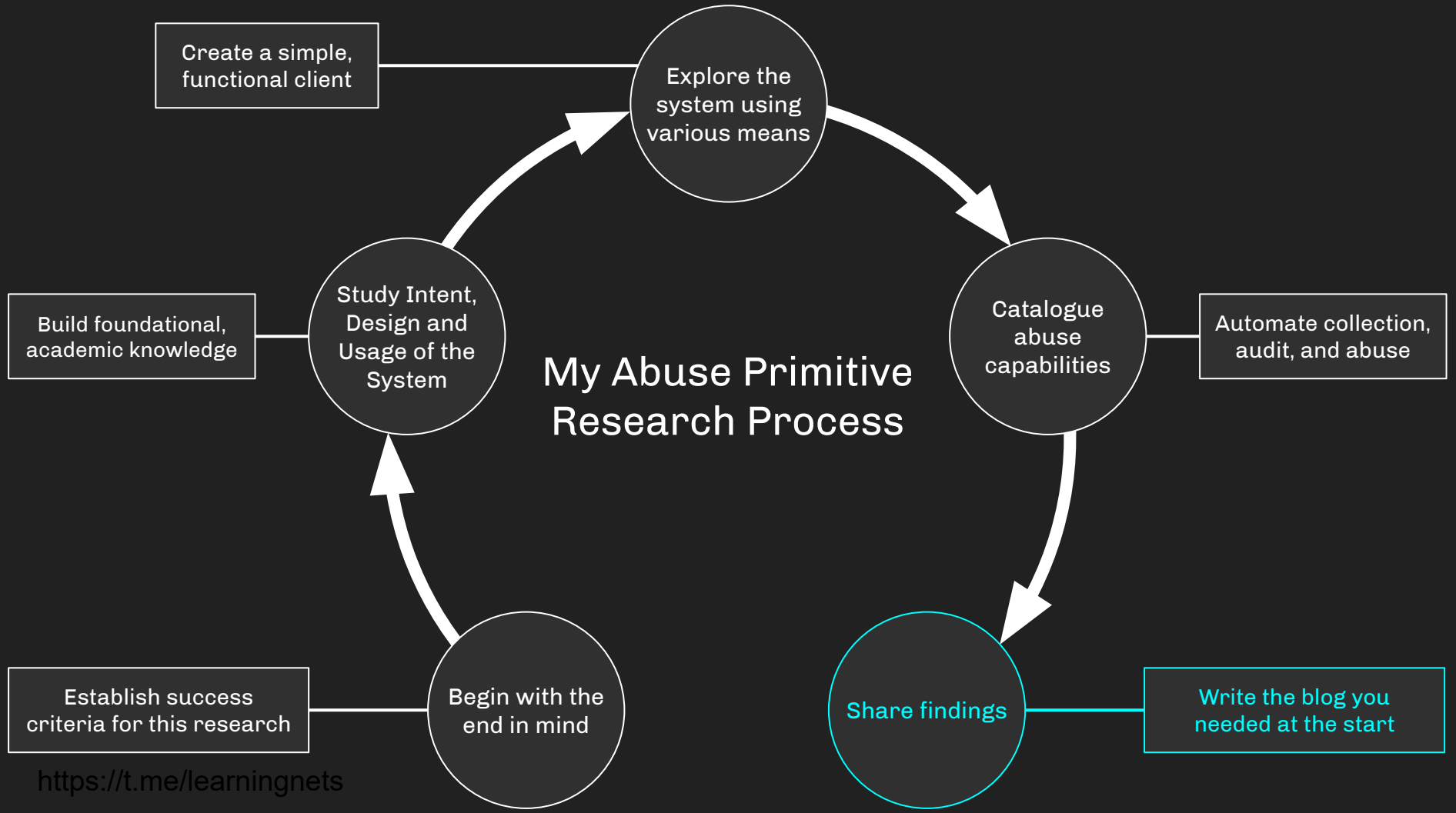
- Inspired by Red Canary's Atomic Red Team
- But with no dependence on existing PowerShell modules
- Can be easily expanded to cover other Azure services

- End result: daily automatic permission->abuse mapping available for anyone to see.









Begin with the end in mind

I want to understand:

- The fundamental mechanics the system
- How the system interacts with other systems
- How the system can be abused

I want to produce:

- A blog/talk for others to understand and build on
- Example audit and abuse code
- Practical remediation guidance

If appropriate for BloodHound, I want to prepare for:

- The impact on the existing graph model
- How to expand the graph model
- What data to collect and ingest, and how to get that data

Begin with the end in mind

I want to understand:
- ✅ The fundamental mechanics the system
- ✅ How the system interacts with other systems
- ✅ How the system can be abused

I want to produce:
- ✅ A blog/talk for others to understand and build on
- ✅ Example audit and abuse code
- ✅ Practical remediation guidance

If appropriate for BloodHound, I want to prepare for:
- ✅ The impact on the existing graph model
- ✅ How to expand the graph model
- ✅ What data to collect and ingest, and how to get that data

Agenda

Why Azure abuses?

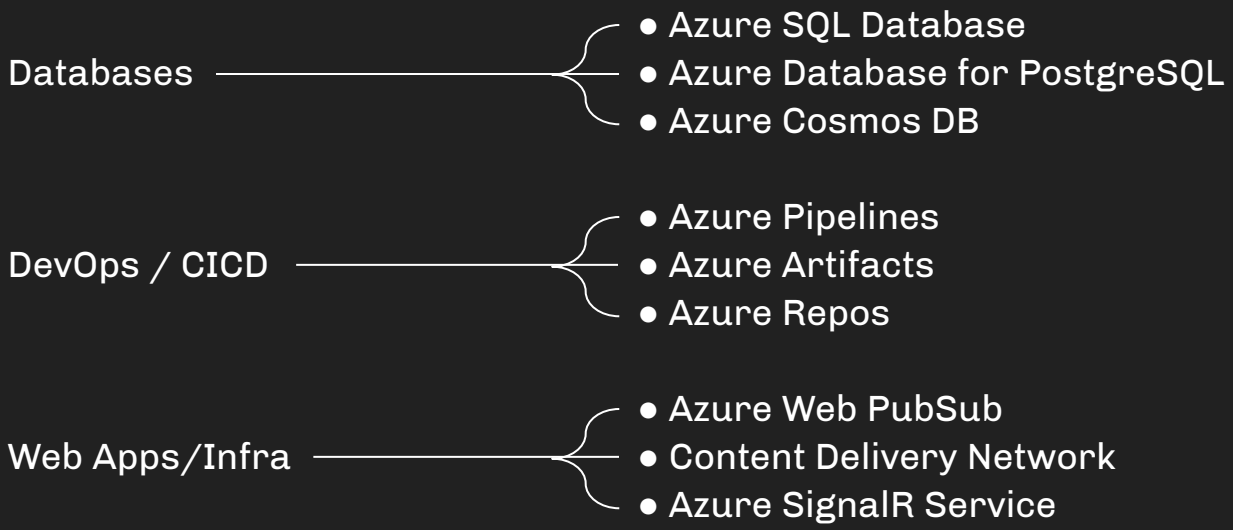
Crash Course Through The Basics

Finding New Attack Primitives: MS Graph Case Study

Where to find research ideas

Conclusion

Build on your existing expertise



See the full (?) directory of services here:
https://azure.microsoft.com/en-us/services/

Follow Microsoft Leaders on Twitter

@JefTek - Jef Kazimer

Principal Program Manager - Azure Active Directory

@BaileyBercick - Bailey Bercick

Program Manager - Azure Active Directory Product Group

@Sue_Bohn - Sue Bohn

Vice President of Program Management in the Identity & Network Access Division

@Alex_A_Simons - Alex Simons

Corporate Vice President of Program Management, Microsoft Identity Division

Follow These People on Twitter

@mariussmellum - Marius Solbakken

Principal - TietoEVRY

@inversecos - Lina Lau

Principal Incident Response Consultant - Secureworks

@DrAzureAD - Dr. Nestori Syynimaa

Senior Principal Security Researcher - Secureworks

@asegunlolu - David Okeyode

EMEA Chief Technology Officer, Azure Cloud - Palo Alto Networks

Bookmark these pages

https://www.thelazyadministrator.com

https://goodworkaround.com/

https://www.azadvertizer.net/

https://thomasvanlaere.com/

https://msportals.io/?search=

Agenda

Why abuses, not bugs?

Crash Course Through The Basics

Finding New Attack Primitives: MS Graph Case Study

Where to find research ideas

Conclusion

Conclusion

There has never been a better time than right now to get involved in Azure abuse research.

Conclusion

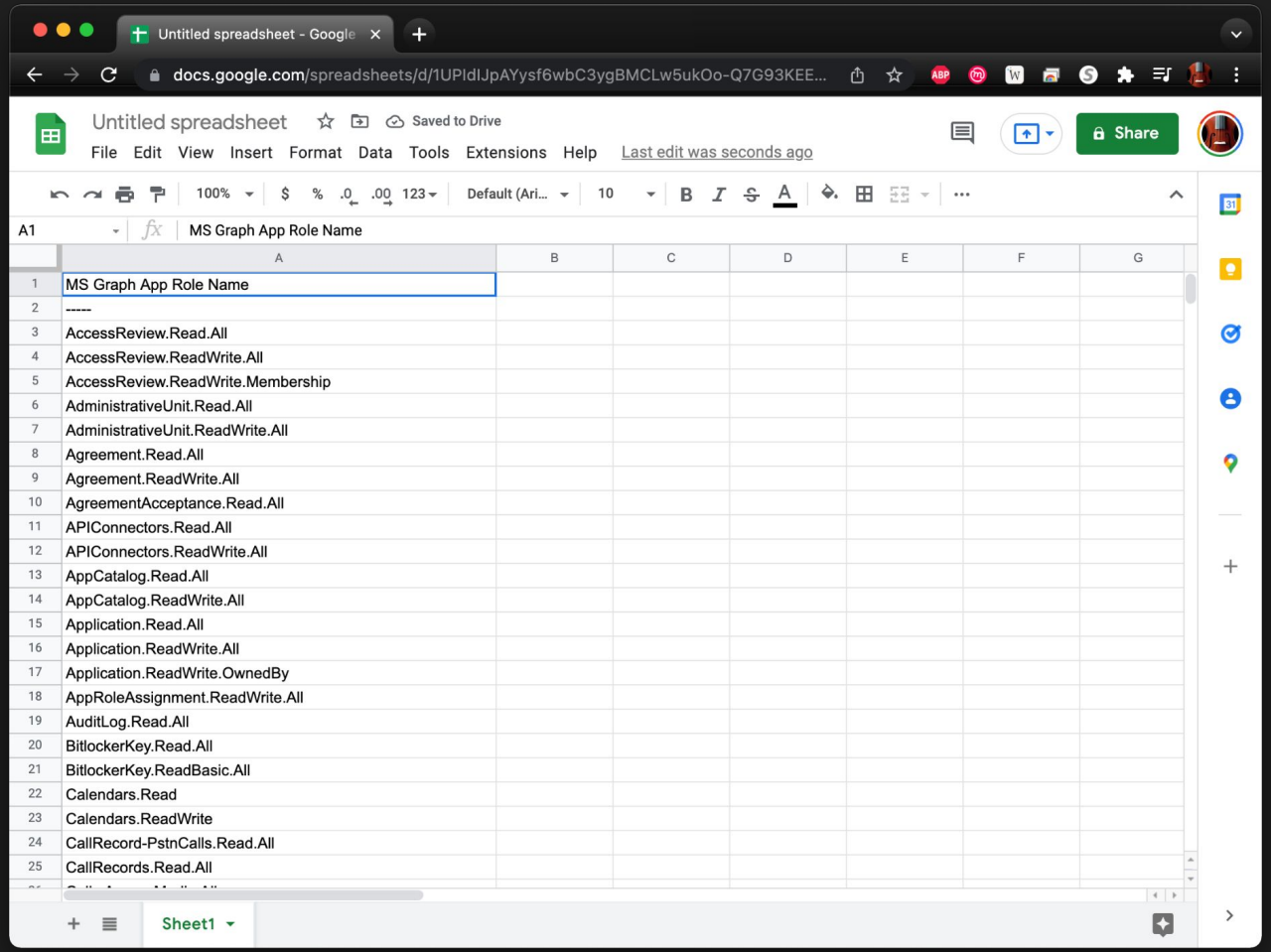
There has never been a better time than right now to get involved in Azure abuse research.

I hope I've shown you just how easy (if tedious) it actually is. Happy hunting!

Thank you!

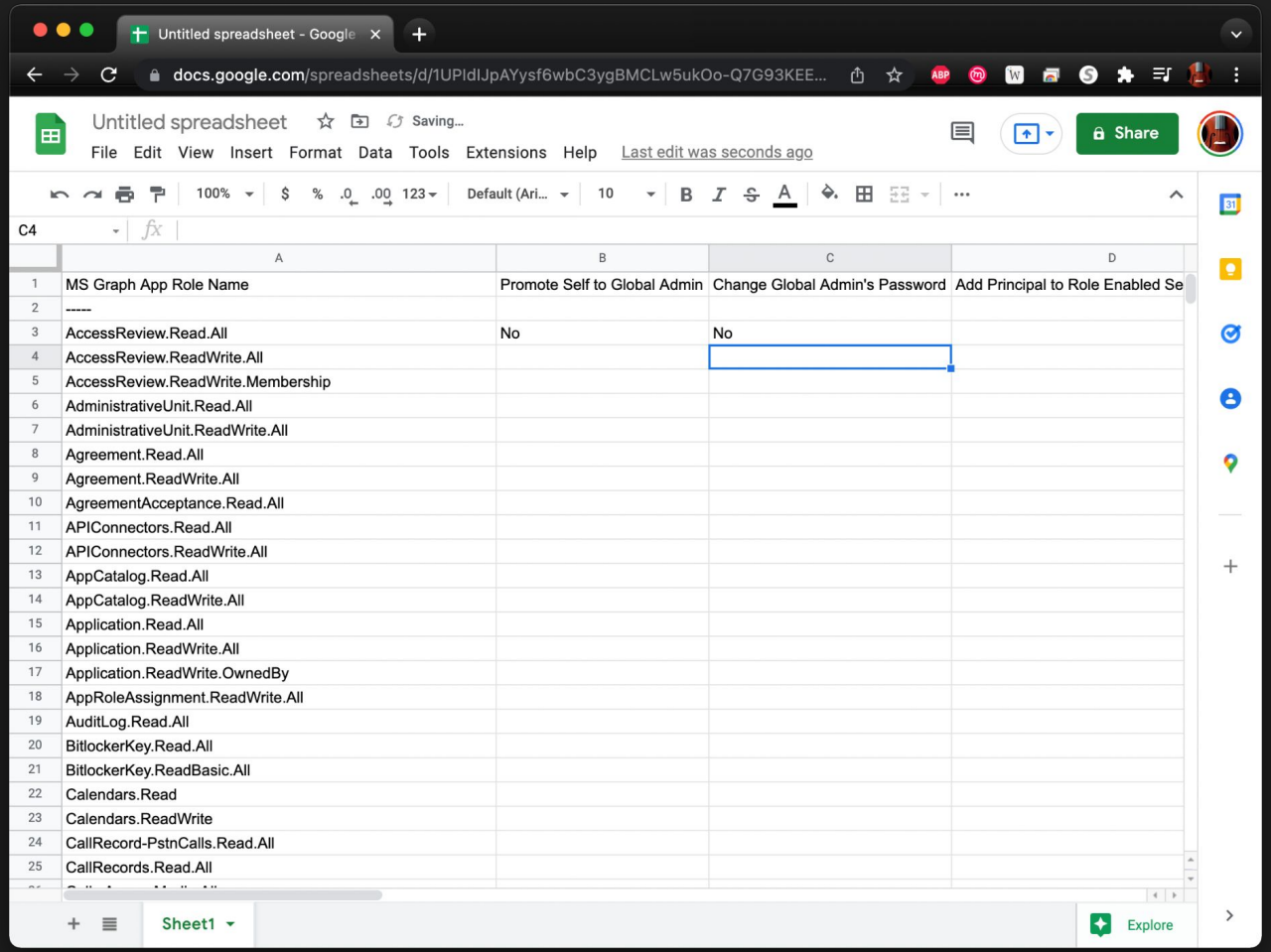
You can find me at @_wald0

Appendix Slides

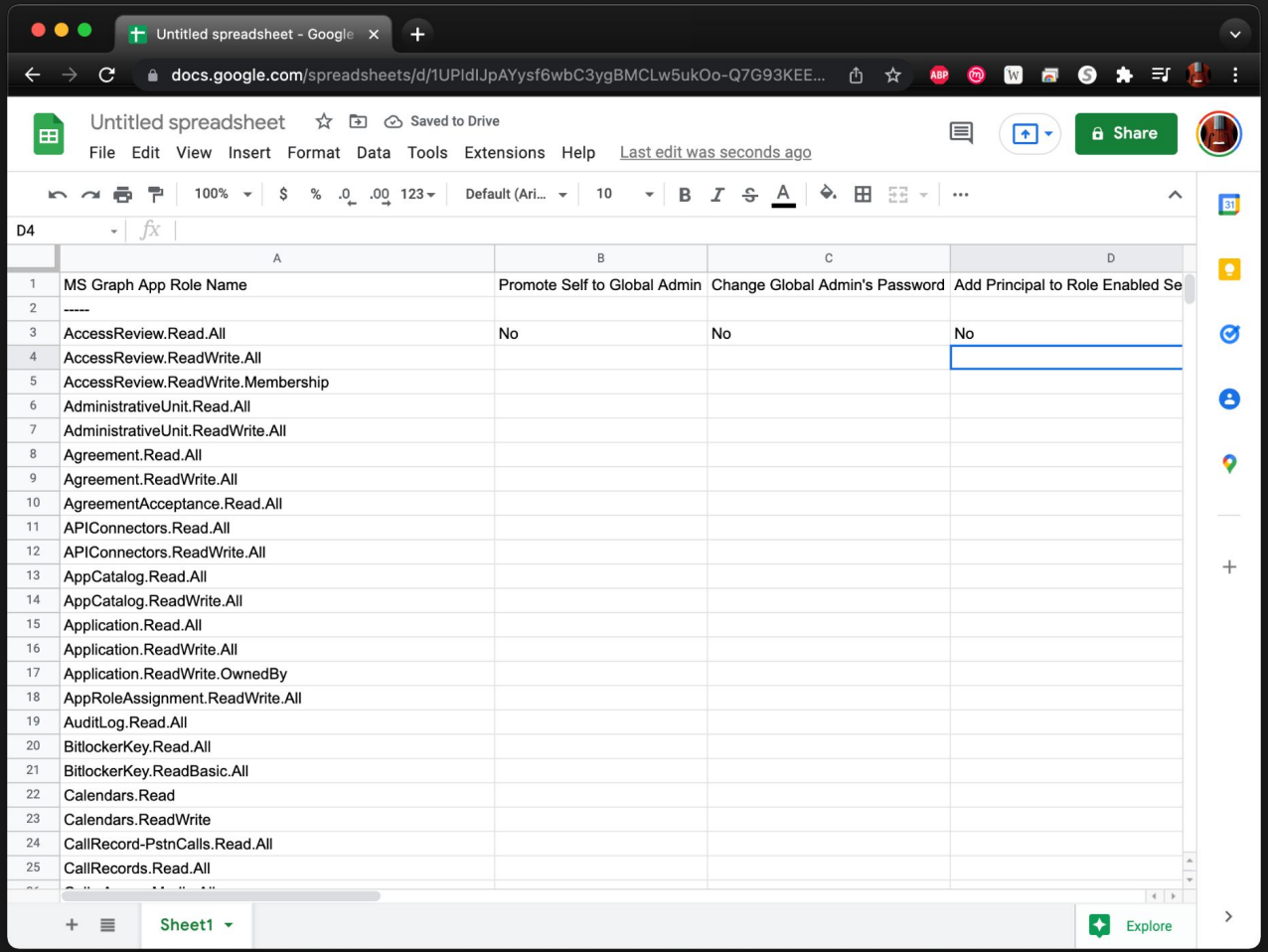


	A
1	MS Graph App Role Name
2	-----
3	AccessReview.Read.All
4	AccessReview.ReadWrite.All
5	AccessReview.ReadWrite.Membership
6	AdministrativeUnit.Read.All
7	AdministrativeUnit.ReadWrite.All
8	Agreement.Read.All
9	Agreement.ReadWrite.All
10	AgreementAcceptance.Read.All
11	APIConnectors.Read.All
12	APIConnectors.ReadWrite.All
13	AppCatalog.Read.All
14	AppCatalog.ReadWrite.All
15	Application.Read.All
16	Application.ReadWrite.All
17	Application.ReadWrite.OwnedBy
18	AppRoleAssignment.ReadWrite.All
19	AuditLog.Read.All
20	BitlockerKey.Read.All
21	BitlockerKey.ReadBasic.All
22	Calendars.Read
23	Calendars.ReadWrite
24	CallRecord-PstnCalls.Read.All
25	CallRecords.Read.All

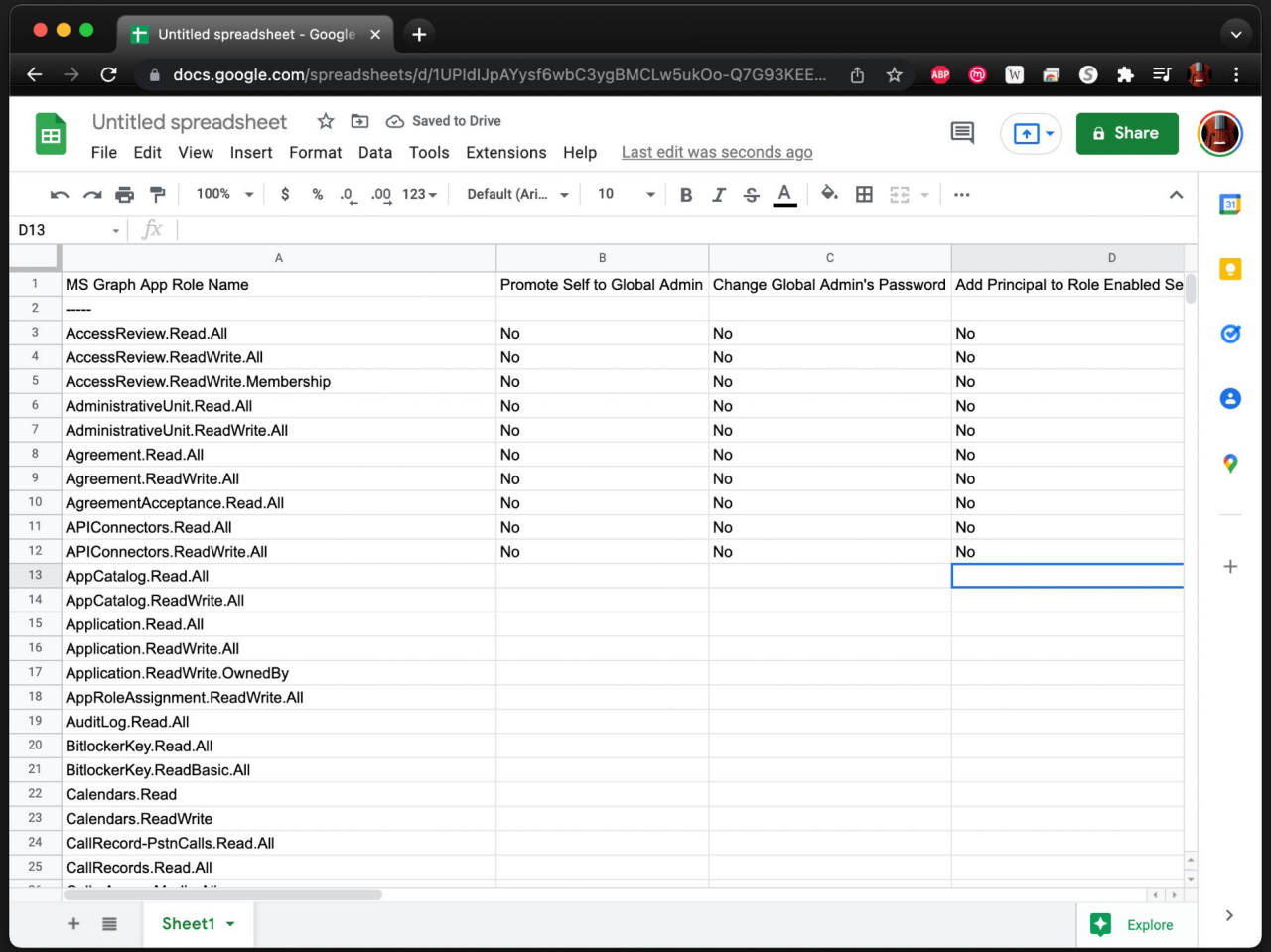
```
# Try to grant "MyCoolApp" the "Global Admin" role:
$body = @{
    "@odata.type" = "#microsoft.graph.unifiedRoleAssignment"
    principalId = "d146464f-523a-4d24-bfce-17d21568647e"
    roleDefinitionId = "62e90394-69f5-4237-9190-012177145e10"
    directoryScopeId = "/"
}
$req = $null
$req = Invoke-RestMethod -Headers @{Authorization = "Bearer $($token.access_token)" } `
    -Uri "https://graph.microsoft.com/v1.0/roleManagement/directory/roleAssignments" `
    -Method POST `
    -Body $($body | ConvertTo-Json) `
    -ContentType 'application/json'
$req
```



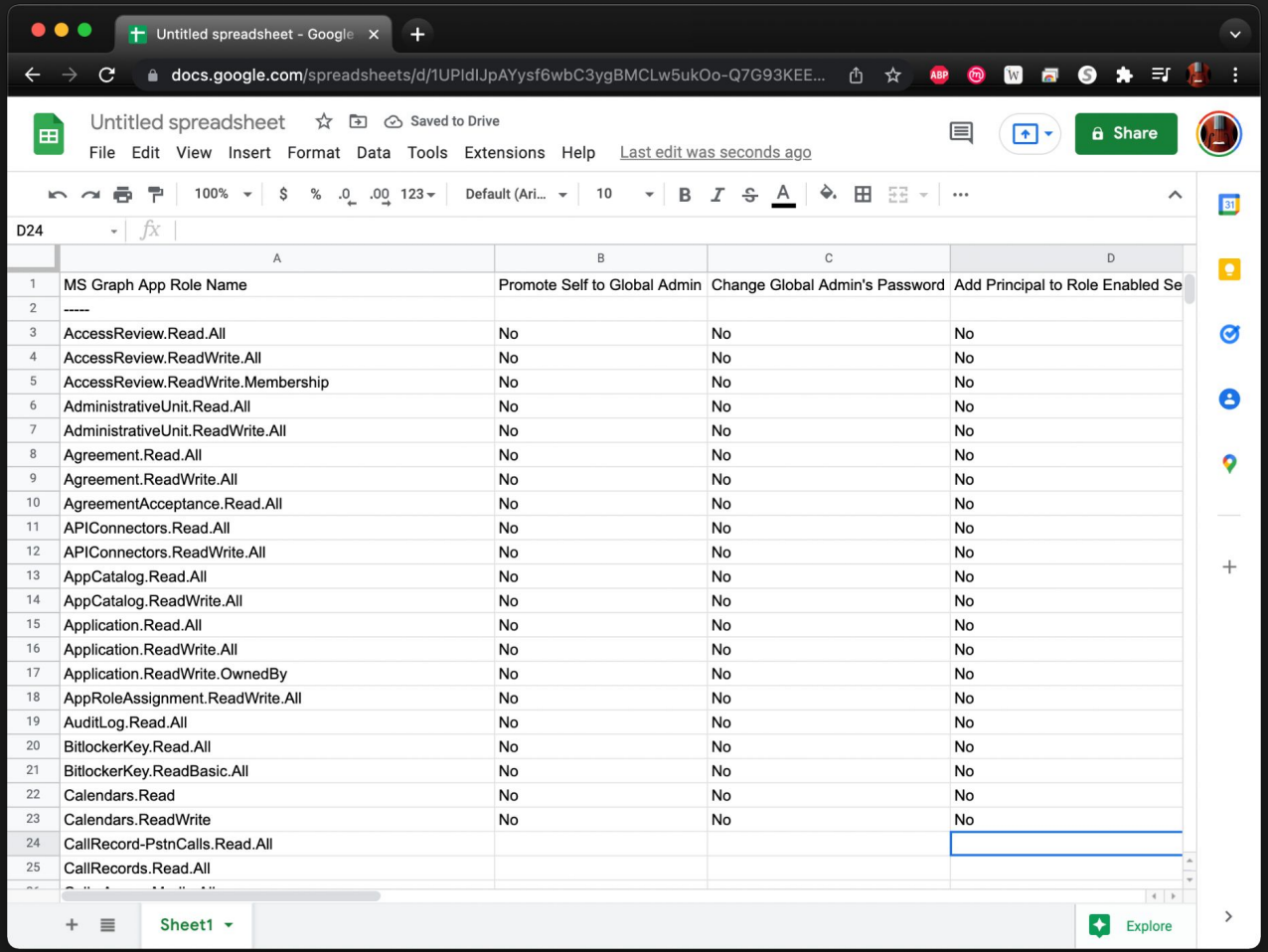
	A	B	C	D
1	MS Graph App Role Name	Promote Self to Global Admin	Change Global Admin's Password	Add Principal to Role Enabled Se
2	-----			
3	AccessReview.Read.All	No	No	
4	AccessReview.ReadWrite.All			
5	AccessReview.ReadWrite.Membership			
6	AdministrativeUnit.Read.All			
7	AdministrativeUnit.ReadWrite.All			
8	Agreement.Read.All			
9	Agreement.ReadWrite.All			
10	AgreementAcceptance.Read.All			
11	APIConnectors.Read.All			
12	APIConnectors.ReadWrite.All			
13	AppCatalog.Read.All			
14	AppCatalog.ReadWrite.All			
15	Application.Read.All			
16	Application.ReadWrite.All			
17	Application.ReadWrite.OwnedBy			
18	AppRoleAssignment.ReadWrite.All			
19	AuditLog.Read.All			
20	BitlockerKey.Read.All			
21	BitlockerKey.ReadBasic.All			
22	Calendars.Read			
23	Calendars.ReadWrite			
24	CallRecord-PstnCalls.Read.All			
25	CallRecords.Read.All			



	A	B	C	D
1	MS Graph App Role Name	Promote Self to Global Admin	Change Global Admin's Password	Add Principal to Role Enabled Se
2	-----			
3	AccessReview.Read.All	No	No	No
4	AccessReview.ReadWrite.All			
5	AccessReview.ReadWrite.Membership			
6	AdministrativeUnit.Read.All			
7	AdministrativeUnit.ReadWrite.All			
8	Agreement.Read.All			
9	Agreement.ReadWrite.All			
10	AgreementAcceptance.Read.All			
11	APIConnectors.Read.All			
12	APIConnectors.ReadWrite.All			
13	AppCatalog.Read.All			
14	AppCatalog.ReadWrite.All			
15	Application.Read.All			
16	Application.ReadWrite.All			
17	Application.ReadWrite.OwnedBy			
18	AppRoleAssignment.ReadWrite.All			
19	AuditLog.Read.All			
20	BitlockerKey.Read.All			
21	BitlockerKey.ReadBasic.All			
22	Calendars.Read			
23	Calendars.ReadWrite			
24	CallRecord-PstnCalls.Read.All			
25	CallRecords.Read.All			



	A	B	C	D
1	MS Graph App Role Name	Promote Self to Global Admin	Change Global Admin's Password	Add Principal to Role Enabled Se
2	-----			
3	AccessReview.Read.All	No	No	No
4	AccessReview.ReadWrite.All	No	No	No
5	AccessReview.ReadWrite.Membership	No	No	No
6	AdministrativeUnit.Read.All	No	No	No
7	AdministrativeUnit.ReadWrite.All	No	No	No
8	Agreement.Read.All	No	No	No
9	Agreement.ReadWrite.All	No	No	No
10	AgreementAcceptance.Read.All	No	No	No
11	APIConnectors.Read.All	No	No	No
12	APIConnectors.ReadWrite.All	No	No	No
13	AppCatalog.Read.All			
14	AppCatalog.ReadWrite.All			
15	Application.Read.All			
16	Application.ReadWrite.All			
17	Application.ReadWrite.OwnedBy			
18	AppRoleAssignment.ReadWrite.All			
19	AuditLog.Read.All			
20	BitlockerKey.Read.All			
21	BitlockerKey.ReadBasic.All			
22	Calendars.Read			
23	Calendars.ReadWrite			
24	CallRecord-PstnCalls.Read.All			
25	CallRecords.Read.All			



	A	B	C	D
1	MS Graph App Role Name	Promote Self to Global Admin	Change Global Admin's Password	Add Principal to Role Enabled Se
2	-----			
3	AccessReview.Read.All	No	No	No
4	AccessReview.ReadWrite.All	No	No	No
5	AccessReview.ReadWrite.Membership	No	No	No
6	AdministrativeUnit.Read.All	No	No	No
7	AdministrativeUnit.ReadWrite.All	No	No	No
8	Agreement.Read.All	No	No	No
9	Agreement.ReadWrite.All	No	No	No
10	AgreementAcceptance.Read.All	No	No	No
11	APIConnectors.Read.All	No	No	No
12	APIConnectors.ReadWrite.All	No	No	No
13	AppCatalog.Read.All	No	No	No
14	AppCatalog.ReadWrite.All	No	No	No
15	Application.Read.All	No	No	No
16	Application.ReadWrite.All	No	No	No
17	Application.ReadWrite.OwnedBy	No	No	No
18	AppRoleAssignment.ReadWrite.All	No	No	No
19	AuditLog.Read.All	No	No	No
20	BitlockerKey.Read.All	No	No	No
21	BitlockerKey.ReadBasic.All	No	No	No
22	Calendars.Read	No	No	No
23	Calendars.ReadWrite	No	No	No
24	CallRecord-PstnCalls.Read.All			
25	CallRecords.Read.All			

	Is that app permission abusable?											
	Abuses											
	Groups				Service Principals / Apps				Directory Roles		App Roles	
	Add owner to role eligible group	Add member to role-eligible group	Add member to non-role eligible group	Add owner to non role-eligible group	Add secret to SP	Add owner to SP	Add secret to App Reg	Add Owner to App Reg	Promote a user to Global Admin		Grant an app role	
Microsoft Graph API Permissions:												
Application.ReadWrite.All	No	No	No	No	Yes	Yes	Yes	Yes	No		No	
AppRoleAssignment.ReadWrite.All	No	No	No	No	No	Yes	No	No	No		Yes	
DelegatedPermissionGrant.ReadWrite.All	No	No	No	No	No	No	No	No	No		No	
Directory.ReadWrite.All	No	No	Yes	Yes	No	No	No	No	No		No	
Group.ReadWrite.All	No	No	Yes	Yes	No	No	No	No	No		No	
GroupMember.ReadWrite.All	No	No	Yes	No	No	No	No	No	No		No	
RoleManagement.ReadWrite.Directory	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes		Yes	
ServicePrincipalEndpoint.ReadWrite.All	No	No	No	No	No	Yes	No	No	No		No	

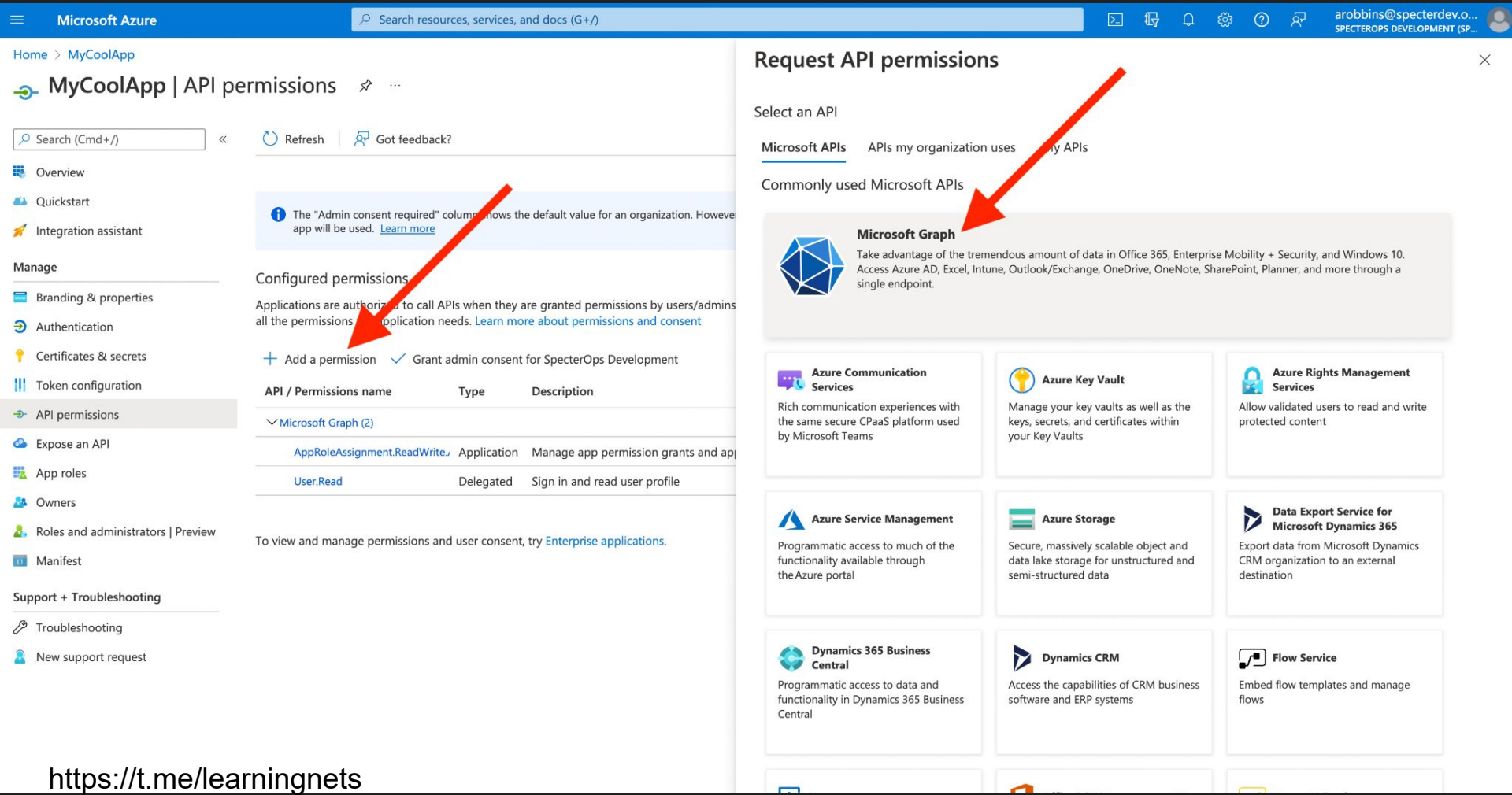
Review: Catalogue the system's abuse capabilities

Actions:

- Determine the scope of objects that can be acted upon through the system
- For each object type, attempt to "abuse" the object through the system using various privileges
- Document your findings
- If possible, automate your abuse tests

Results:

- You should understand where the system materially differs from documentation
- You should now know at least some abuse primitives against the system
- You should now know what information to collect, and how, to find abusable configurations in the system
- You should be able to publish your findings for industry colleagues to understand and build on top of



Microsoft Azure | Search resources, services, and docs (G+/) | arobbins@specterdev.o... SPECTEROPS DEVELOPMENT (SP...

Home > MyCoolApp

MyCoolApp | API permissions

Search (Cmd+/)

- Overview
- Quickstart
- Integration assistant

Manage

- Branding & properties
- Authentication
- Certificates & secrets
- Token configuration
- API permissions
- Expose an API
- App roles
- Owners
- Roles and administrators | Preview
- Manifest

Support + Troubleshooting

- Troubleshooting
- New support request

Refresh | Got feedback?

The "Admin consent required" column shows the default value for an organization. Howeve
app will be used. Learn more

Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins
all the permissions the application needs. Learn more about permissions and consent

Add a permission | Grant admin consent for SpecterOps Development

API / Permissions name	Type	Description
Microsoft Graph (2)		
AppRoleAssignment.ReadWrite.	Application	Manage app permission grants and ap
User.Read	Delegated	Sign in and read user profile

To view and manage permissions and user consent, try Enterprise applications.

Request API permissions

All APIs

Microsoft Graph
https://graph.microsoft.com/ Docs

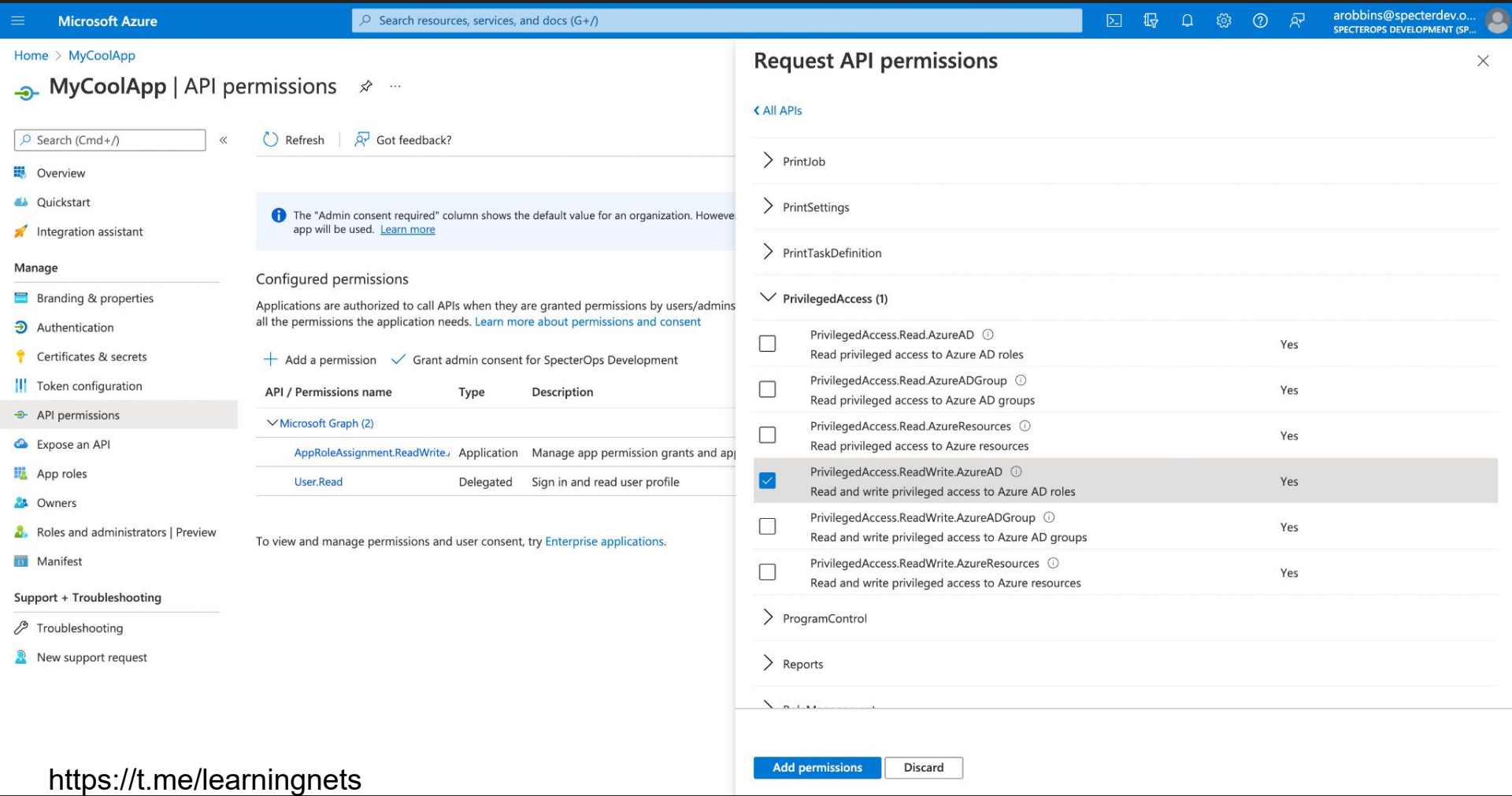
What type of permissions does your application require?

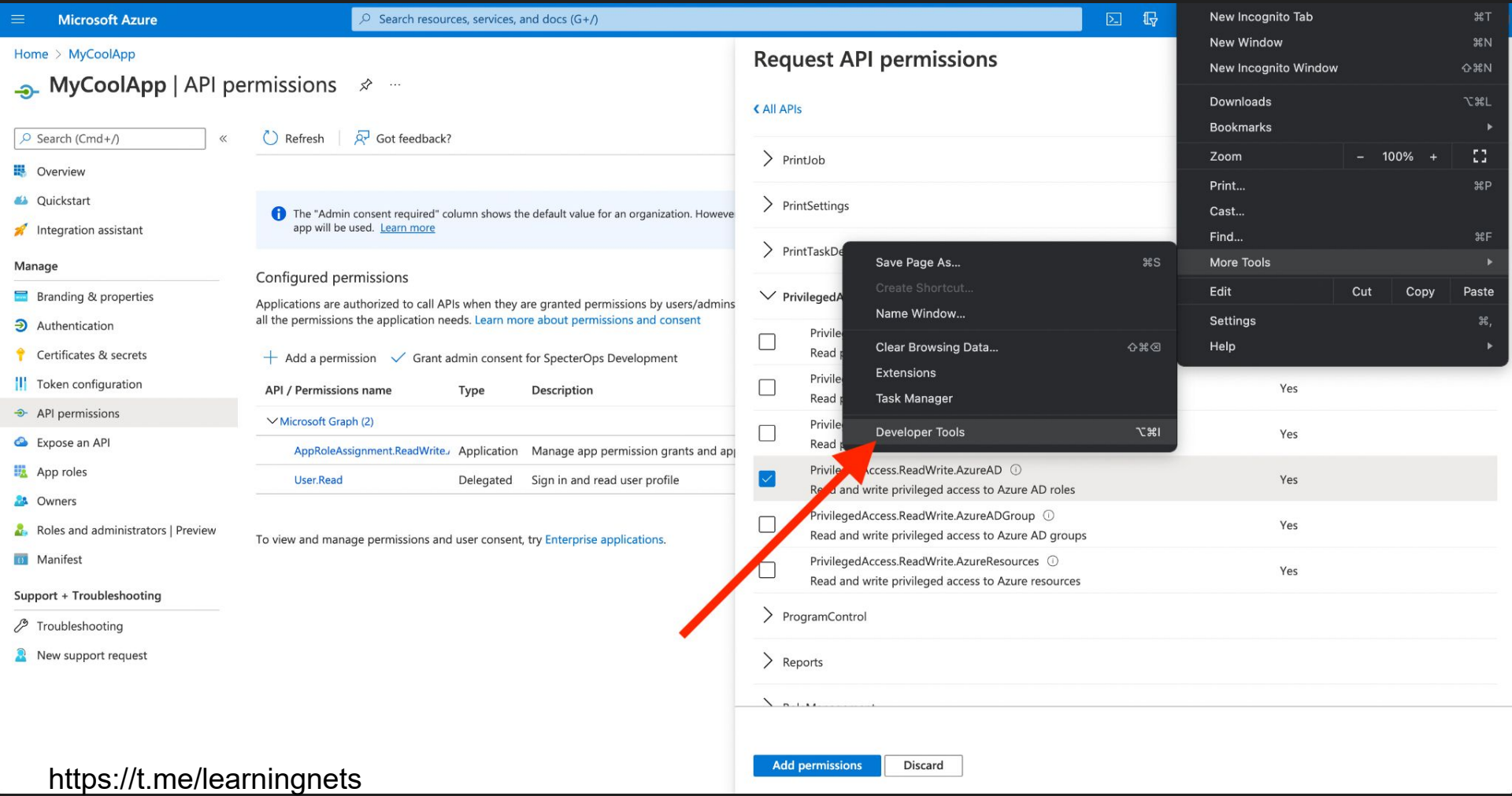
Delegated permissions
Your application needs to access the API as the signed-in user.

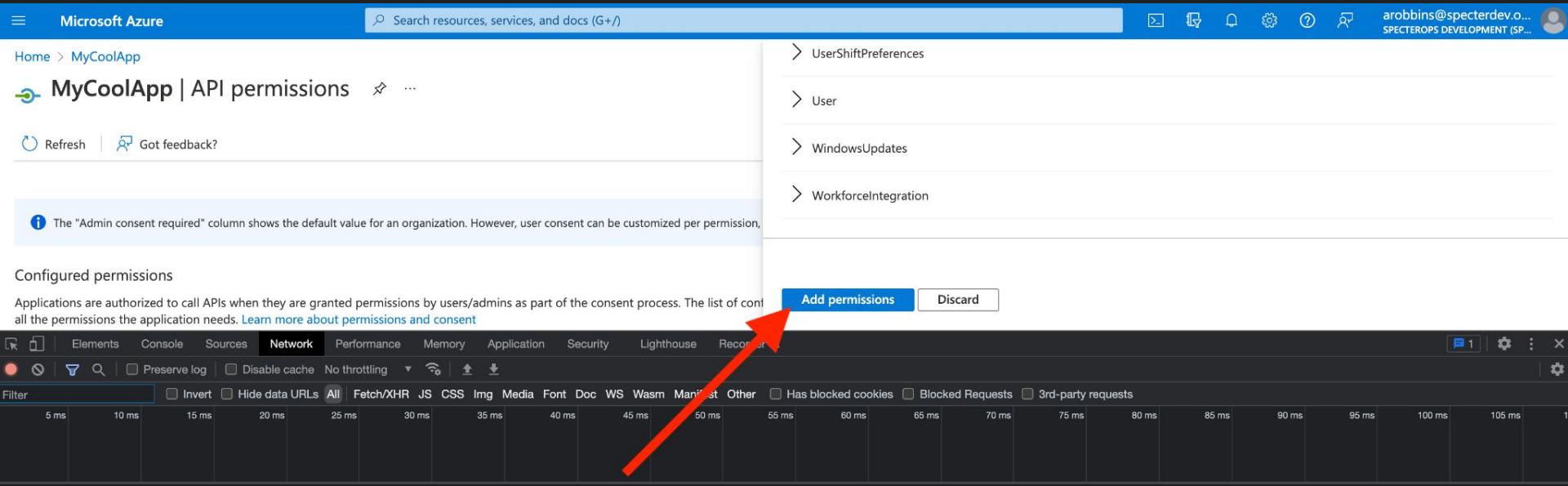
Application permissions
Your application runs as a background service or daemon without a signed-in user.

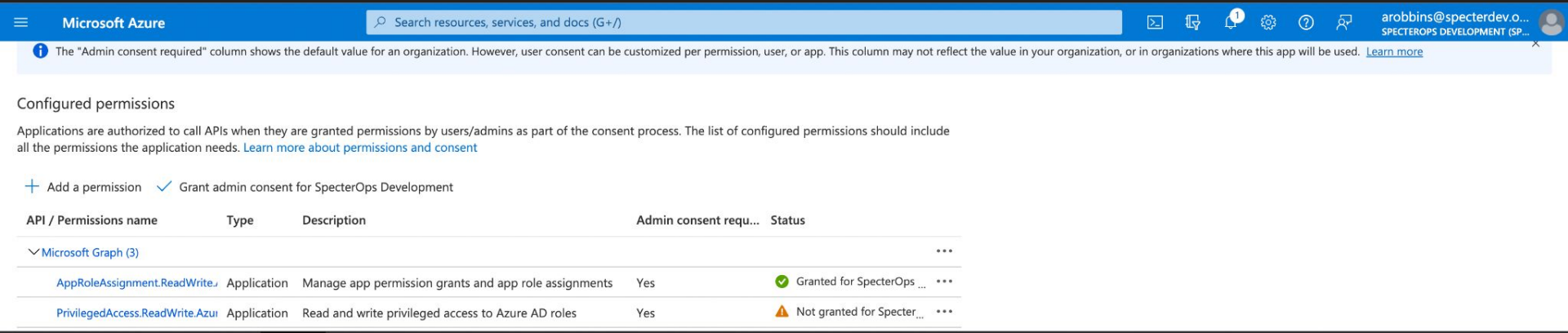
Your application runs as a background service or daemon without a signed-in user.

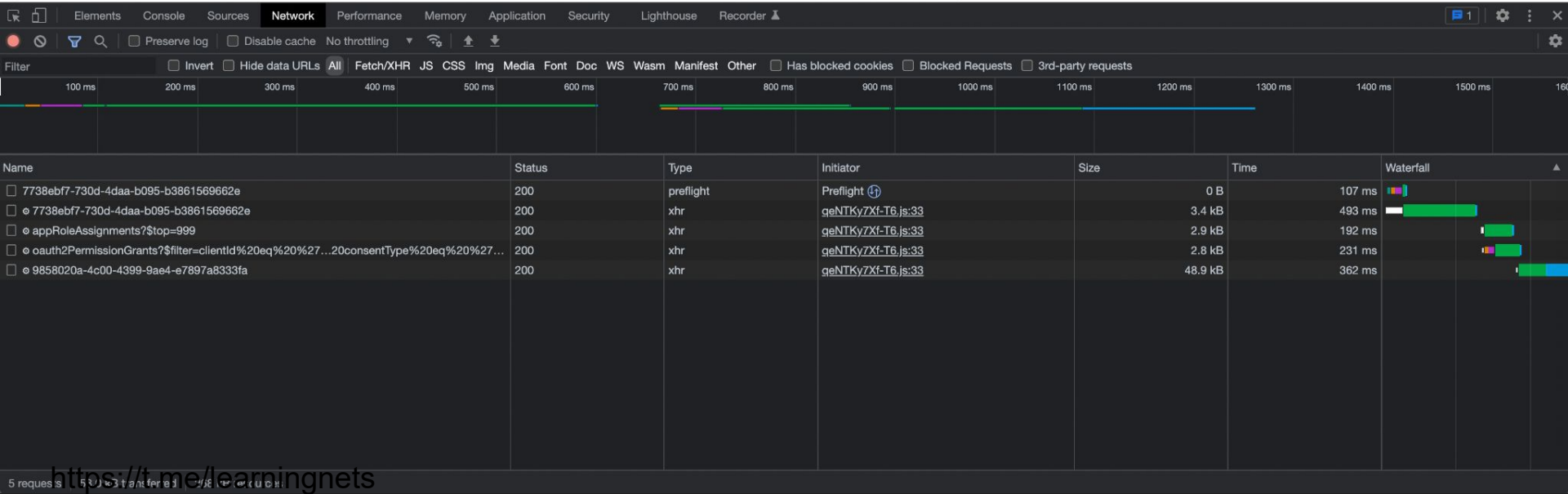
Add permissions | Discard

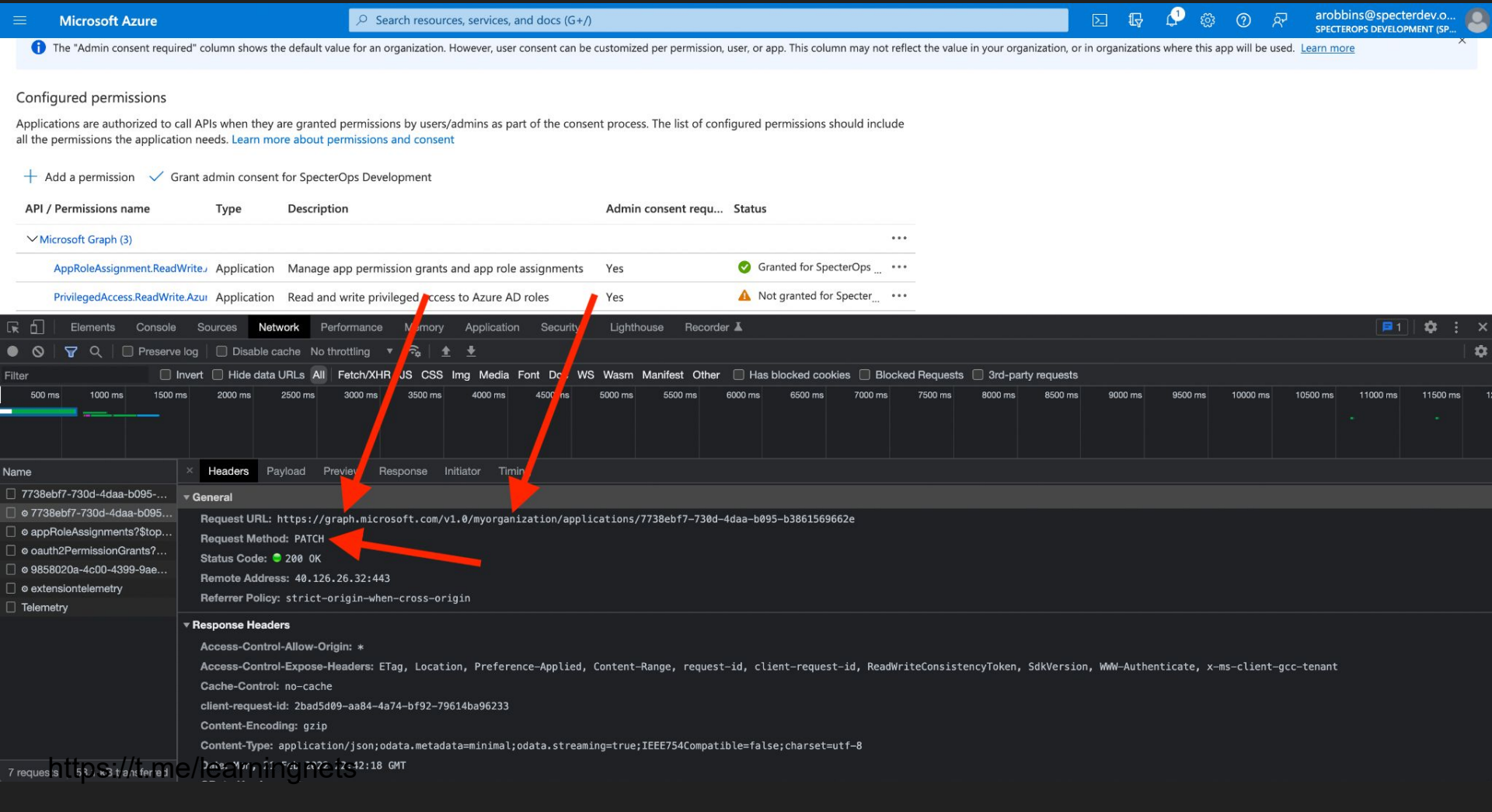


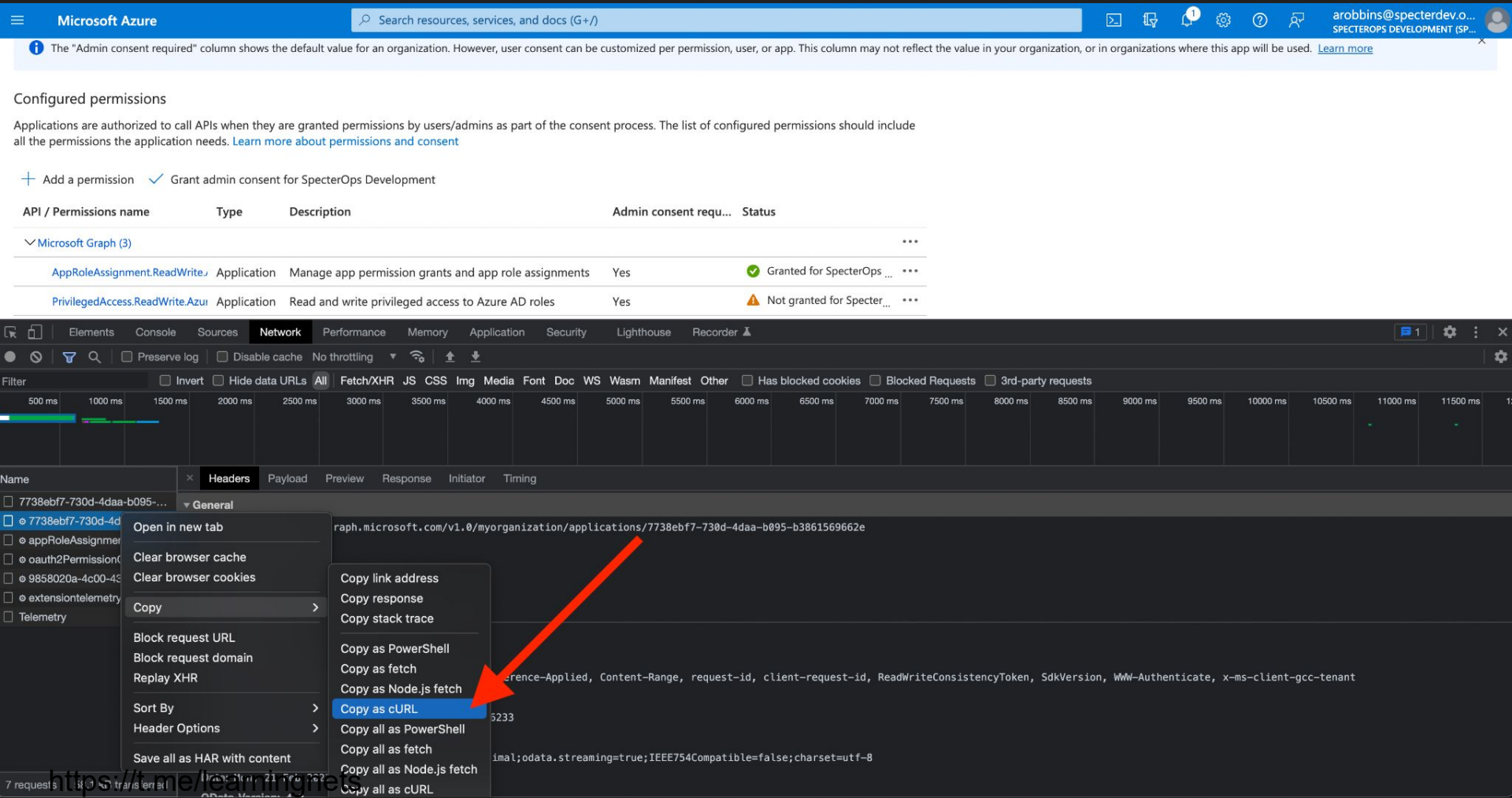


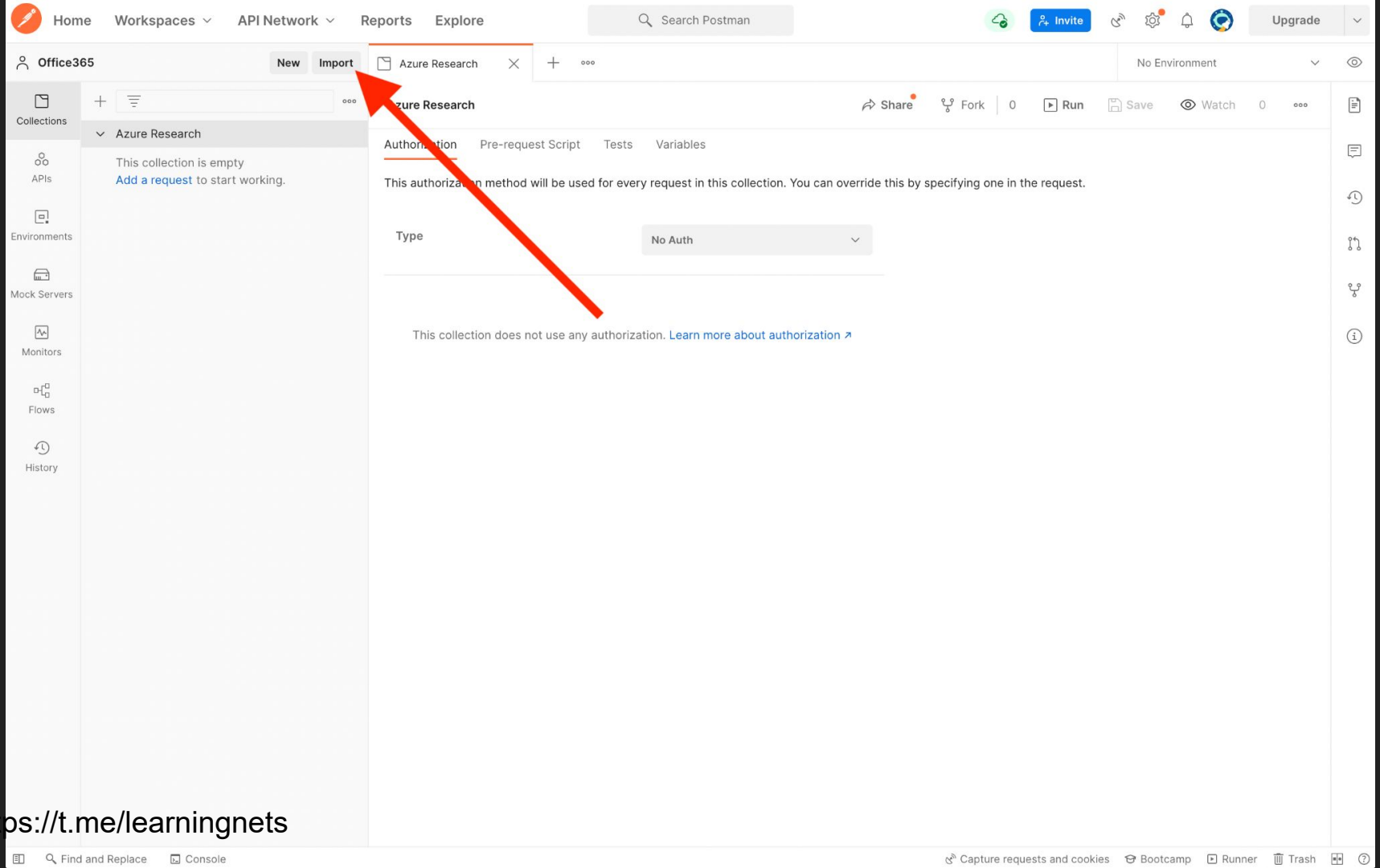


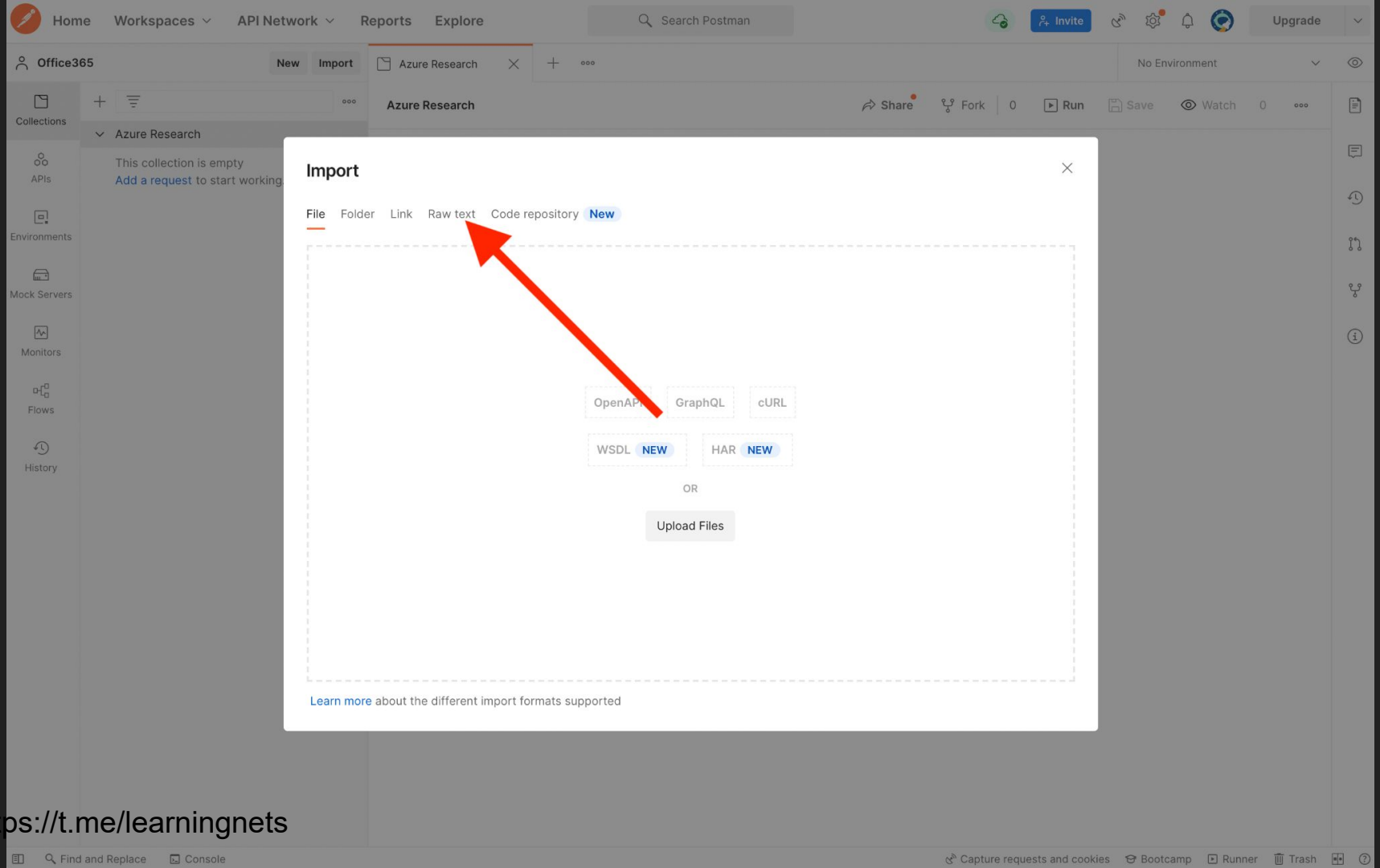


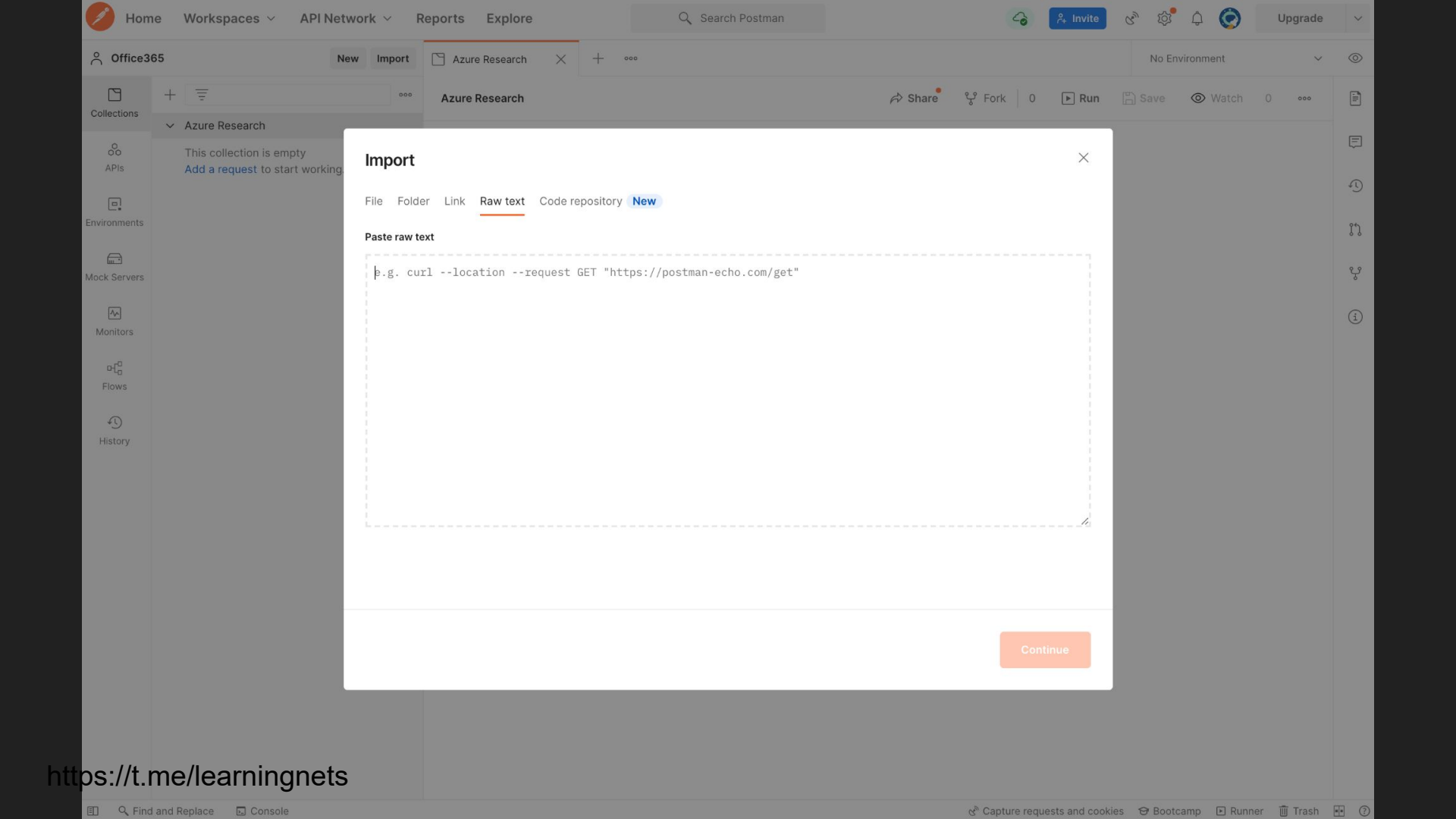


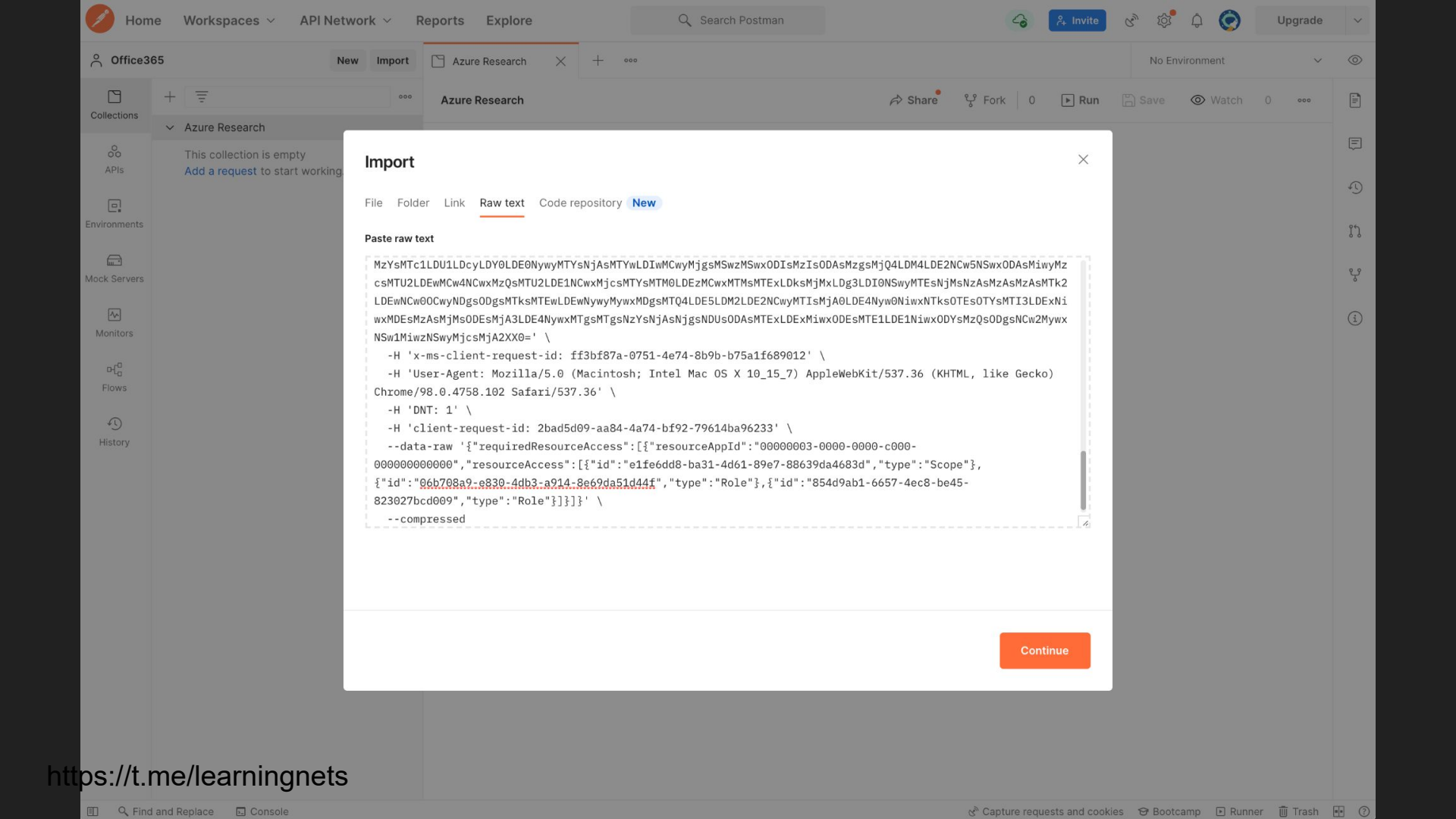


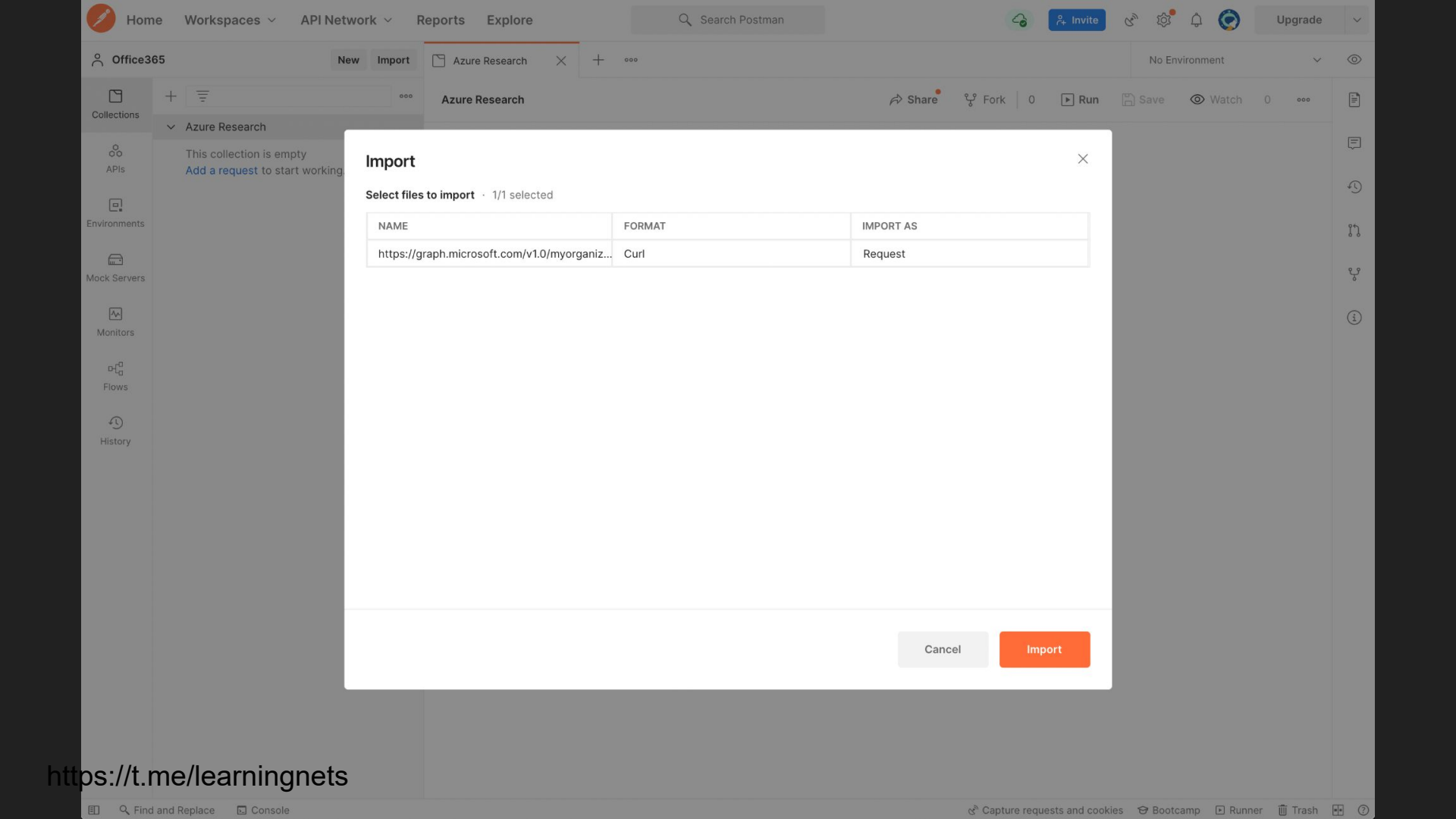

Microsoft Azure
Search resources, services, and docs (G+/)
arobbins@specterdev.o...
SPECTEROPS DEVELOPMENT (SP...
The "Admin consent required" column shows the default value for an organization. However, user consent can be customized per permission, user, or app. This column may not reflect the value in your organization, or in organizations where this app will be used. Learn more
Configured permissions
Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. Learn more about permissions and consent
Add a permission
Grant admin consent for SpecterOps Development
API / Permissions name
Type
Description
Admin consent requ...
Status
Microsoft Graph (3)
AppRoleAssignment.ReadWrite...
Application
Manage app permission grants and app role assignments
Yes
Granted for SpecterOps ...
PrivilegedAccess.ReadWrite.Azu...
Application
Read and write privileged access to Azure AD roles
Yes
Not granted for Specter...
Elements
Console
Sources
Network
Performance
Memory
Application
Security
Lighthouse
Recorder
Preserve log
Disable cache
No throttling
Filter
Invert
Hide data URLs
All
Fetch/XHR
JS
CSS
Img
Media
Font
Doc
WS
Wasm
Manifest
Other
Has blocked cookies
Blocked Requests
3rd-party requests
500 ms
1000 ms
1500 ms
2000 ms
2500 ms
3000 ms
3500 ms
4000 ms
4500 ms
5000 ms
5500 ms
6000 ms
6500 ms
7000 ms
7500 ms
8000 ms
8500 ms
9000 ms
9500 ms
10000 ms
10500 ms
11000 ms
11500 ms
Name
Headers
Payload
Preview
Response
Initiator
Timing
7738ebf7-730d-4daa-b095-...
General
Open in new tab
Clear browser cache
Clear browser cookies
Copy
Block request URL
Block request domain
Replay XHR
Sort By
Header Options
Save all as HAR with content
Copy link address
Copy response
Copy stack trace
Copy as PowerShell
Copy as fetch
Copy as Node.js fetch
Copy as cURL
Copy all as PowerShell
Copy all as fetch
Copy all as Node.js fetch
Copy all as cURL
appRoleAssignmer
oauth2Permission(
9858020a-4c00-43
extensiontelemetry
Telemetry
raph.microsoft.com/v1.0/myorganization/applications/7738ebf7-730d-4daa-b095-b3861569662e
erence-Applied, Content-Range, request-id, client-request-id, ReadWriteConsistencyToken, SdkVersion, WWW-Authenticate, x-ms-client-gcc-tenant
6233
imal;odata.streaming=true;IEEE754Compatible=false;charset=utf-8
7 requests
https://t.me/learningnets











Home Workspaces API Network Reports Explore Search Postman Invite Upgrade

Office365 New Import

Collections APIs Environments Mock Servers Monitors Flows History

Azure Research

This collection is empty
Add a request to start working.

Azure Research PATCH https://graph.mi... No Environment

https://graph.microsoft.com/v1.0/myorganization/applications/7738ebf7-730d-4daa-b095-b3861569662e Save

PATCH https://graph.microsoft.com/v1.0/myorganization/applications/7738ebf7-730d-4daa-b095-b3861569662e Send

Params Authorization Headers (21) Body Pre-request Script Tests Settings Cookies

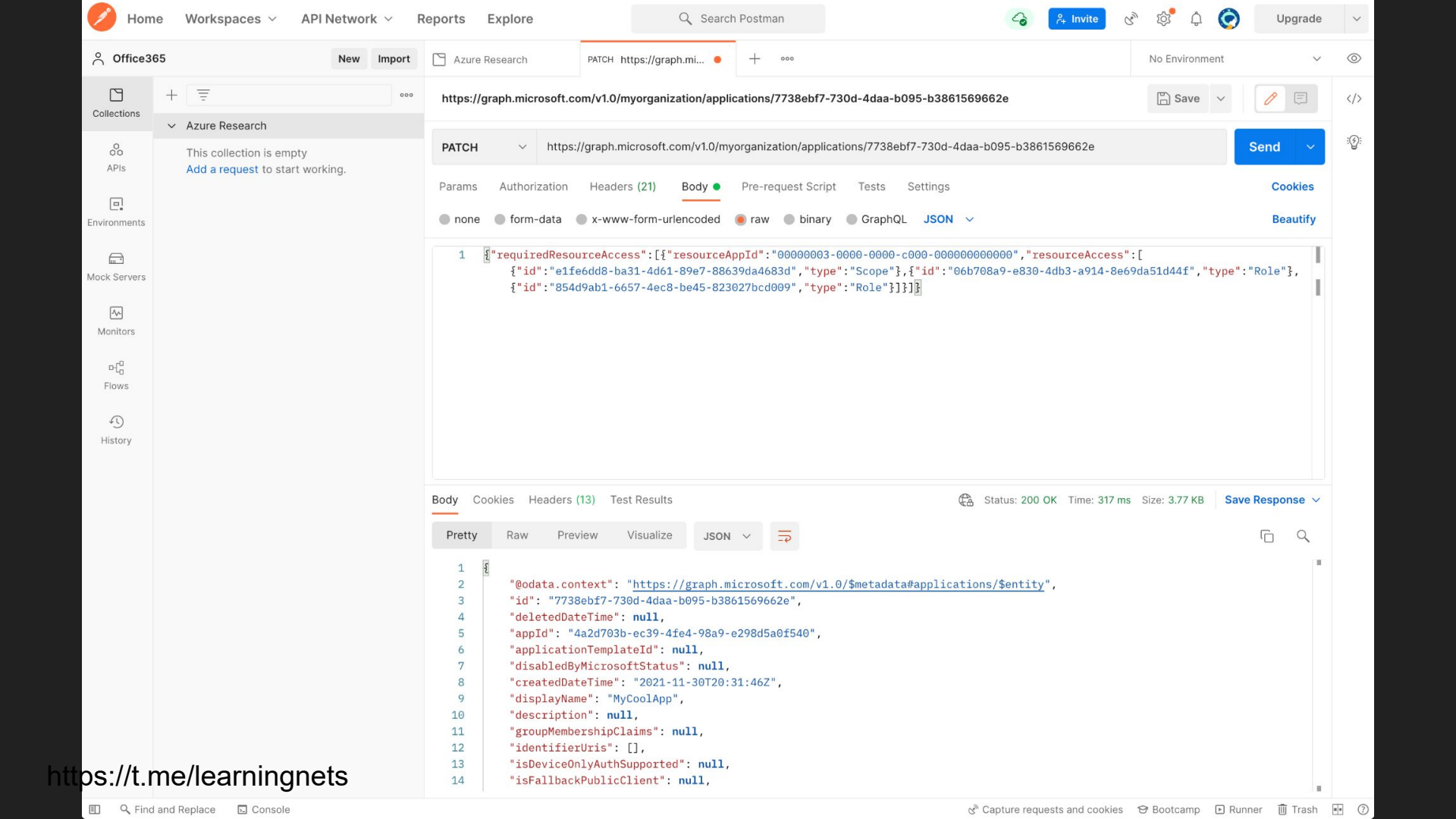
Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

Response

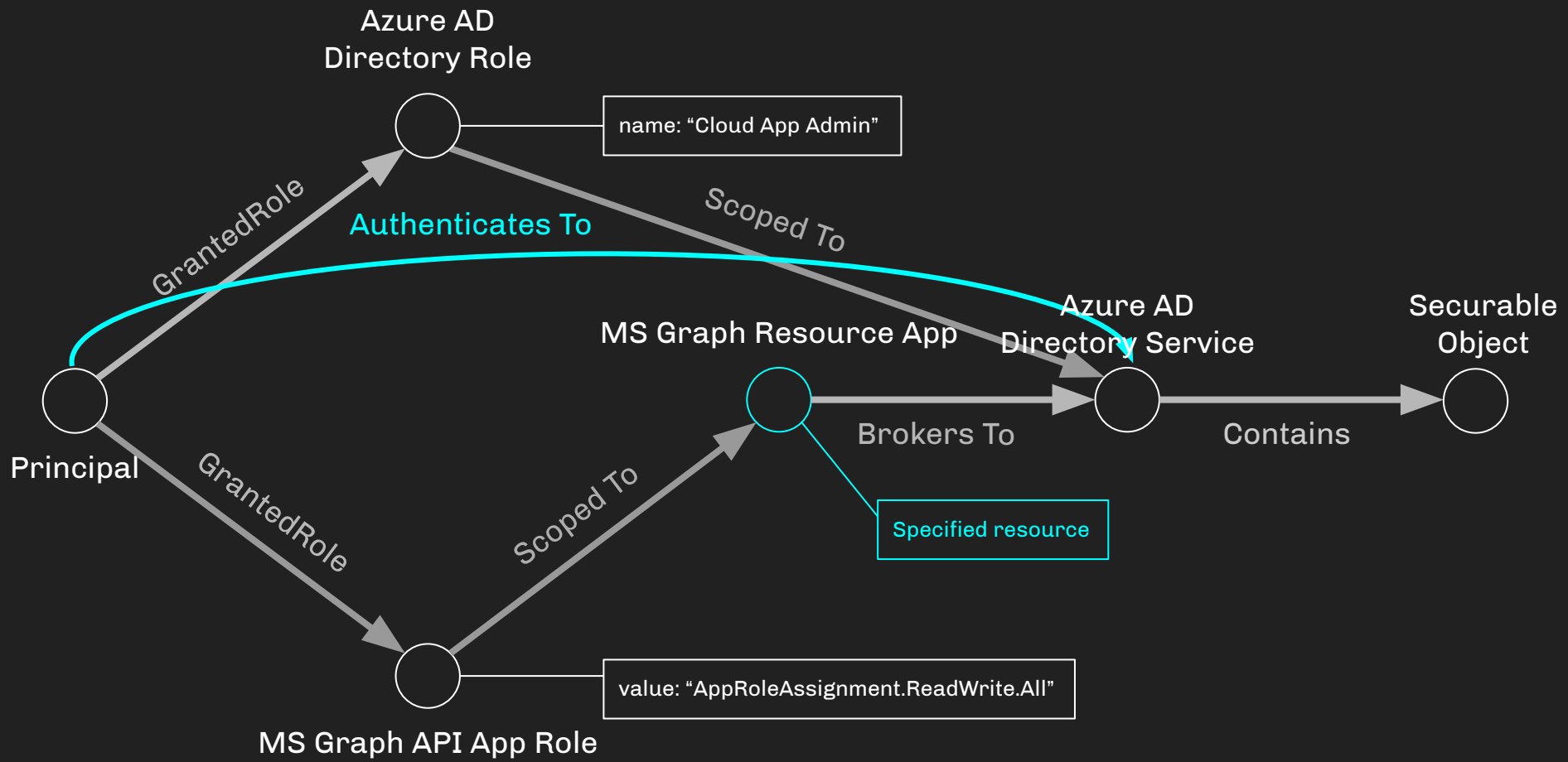
Click Send to get a response

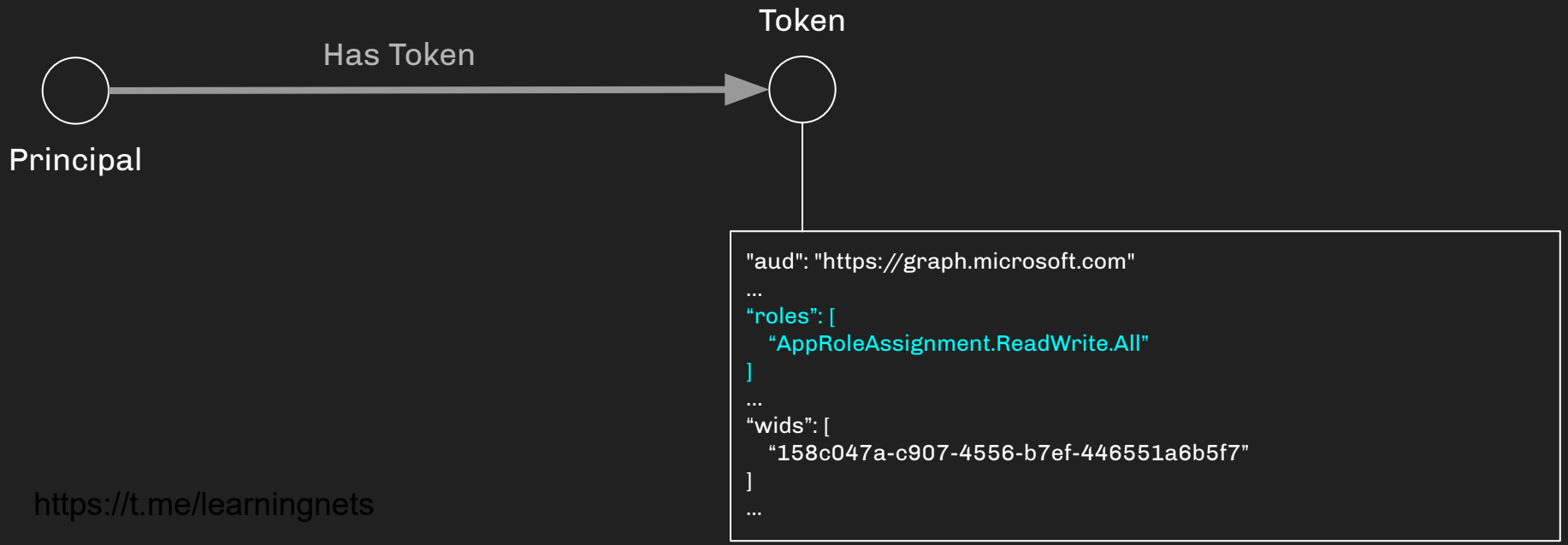
Find and Replace Console Capture requests and cookies Bootcamp Runner Trash



Fine-grained control lets you:

- Determine the minimum parameters actually required by the API
- Easily change request properties to test for CA bypasses
- Accurately automate your testing
- Build future tooling without needing 3rd party modules
- Discover how the system materially differs from its documentation
- VASTLY simplify getting data out of the system
- Get MORE data than what the GUI will ever show you





```
PS C:\Users\andyrobbins> # Now that we have a bearer token, we can interface with the MS Graph API
PS C:\Users\andyrobbins> $URI        =   'https://graph.microsoft.com/v1.0/Groups/'
PS C:\Users\andyrobbins> $Request    =   Invoke-RestMethod `
>>                     -Headers @{Authorization = "Bearer $($token.access_token)"} `
>>                     -URI $URI `
>>                     -Method GET
PS C:\Users\andyrobbins> $Request.value | Select -First 1


id                            : 0b57a21e-4fe0-4bb7-a7a8-4764bcc1b3bf
deletedDateTime               :
classification                :
createdDateTime               : 2021-11-03T22:33:56Z
creationOptions               : {}
description                   :
displayName                   : AAD DC Administrators
expirationDateTime            :
groupTypes                    : {}
isAssignableToRole            :
mail                          :
mailEnabled                   : False
mailNickname                  : AADDCAdministrators
membershipRule                :
membershipRuleProcessingState :
onPremisesDomainName          :
onPremisesLastSyncDateTime    :
onPremisesNetBiosName         :
onPremisesSamAccountName      :
onPremisesSecurityIdentifier  :
onPremisesSyncEnabled         :
preferredDataLocation         :
preferredLanguage             :
proxyAddresses                : {}
renewedDateTime               : 2021-11-03T22:33:56Z
resourceBehaviorOptions       : {}
resourceProvisioningOptions   : {}
securityEnabled               : True
securityIdentifier            : S-1-12-1-190292510-1270304736-1682417831-3216228796
theme                         :
visibility                    :
onPremisesProvisioningErrors  : {}
```

My High Level Methodology:

1. Begin with the end in mind
2. Understand the intent and design of the system
3. Explore the system from different perspectives
4. Catalogue the system’s abuse capabilities