Attacking and Defending Azure – Advanced Edition

Nikhil Mittal – Keanu Nys

alteredsecurity.com

Nikhil Mittal

- Twitter - @nikhil_mitt
- Founder of Altered Security - alteredsecurity.com
- GitHub - github.com/samratashok
- Creator of Nishang, Deploy-Deception, RACE toolkit and more
- Interested in Active Directory and Azure security
- Previous Talks and/or Trainings
  - DEF CON, BlackHat, BruCON and more.

About me

- Keanu Nys – Offensive Security Lead in Belgium
- Socials
  - Twitter - @RedByte1337
  - LinkedIn - linkedin.com/in/keanunys/
  - GitHub - github.com/RedByte1337
- Creator of the M365 and Entra attack toolkit GraphSpy
- Passion for Active Directory security, Azure security & Social Engineering
- Certifications
  - eCPPTv2, OSEP, CRTE, CARTP, CARTE, MS-500, …

Altered Security

- Trained more than 30000 security professionals from more than 130 countries!
- Our Red Team Labs Platform enables labs to be:
  - Affordable
  - Easy to Access
  - Stable and provide great user experience
  - Fun to Solve
  - Big enough to feel enterprise-like

ALTERED SECURITY

Course Content

- Introduction to attack methodology and tools
- APIs and endpoints
- Abusing Microsoft Graph API
- Initial Access Attacks - Device Code Phishing, Illicit Consent Grant, Attacker In The Middle, Abusing JWT Signing, Abusing Custom Claims, Abusing GitHub Actions and Workflow Discovery and Recon
- Privilege Escalation - Abusing Family of Client IDs, Certificate Based Authentication, Attribute Based Access Control, Privileged Identity Management, Tampering with Logic Apps, Authentication Cookies, Traffic Interception and more

Course Content

- Lateral Movement - Abusing Azure Lighthouse, Cross Tenant Access Settings, Kerberos in Entra ID, Trust between tenants, Multi-Cloud Management, Azure ARC, Token Extraction, Authentication Cookie Forging and Replay etc.
- Bypassing Defenses - Advanced Conditional Access Policies, Multiple ways to bypass MFA that is enforced using different methods, Privileged Identity Management (PIM) and Microsoft Defender for Cloud.
- Detecting and Stopping the attacks used in the class using Log Analysis and MS tools like Identity Protection, MFA, Conditional Access and Defender for Cloud.

Goal

- The course expects prior knowledge of Azure and Entra (Azure Active Directory).
- This course introduces a concept, demonstrates how an attack can be executed and then have Learning Objective section where students can practice on the lab.
- The course is split in four kill chains, and we complete all the attacks in one kill chain at a time.
- We may make some assumptions for smoothly executing attacks in the lab.
- Everything is not in the slides :)

How to use the course content

- You have access to the slides, slides notes, lab manual, walk-through videos, Kill Chain diagrams,Lab Diagram and Tools used in the course OneDrive.

- Access the OneDrive using the lab portal - https://azureadvanced.enterprisesecurity.io/

- Keeping an eye on the Lab diagram and Kill chain diagrams will help if you feel lost.

Word of Caution

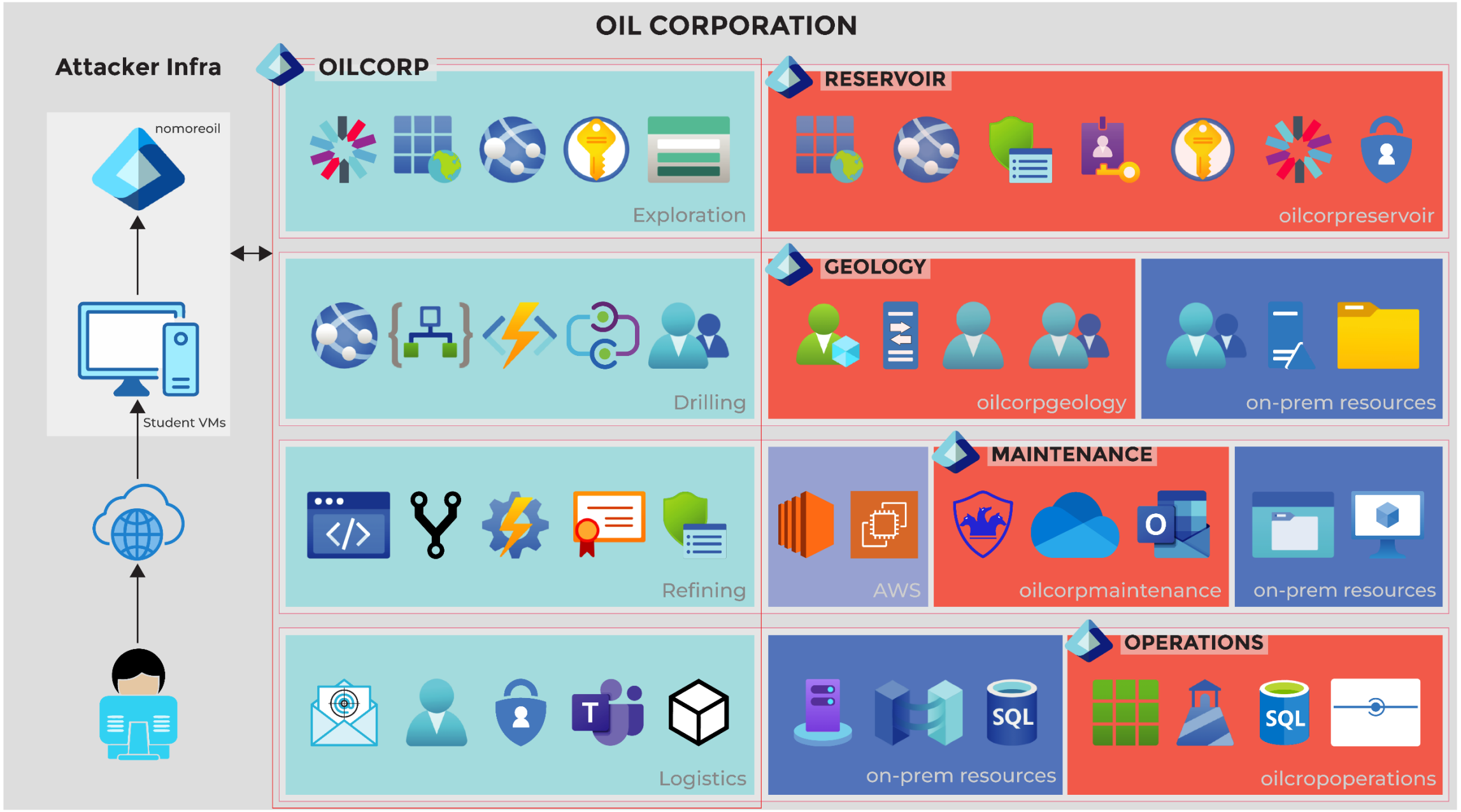
- In scope:
  - Only the explicitly specified on-prem and Azure resources and users are in scope.
- Everything else is NOT in scope.
- Any abuse of the lab internet or resources - attempts of unauthorized access or attacks on external infrastructure - will result in immediate disqualification from the class without refund.
- Please treat the lab network as a dangerous environment and take care of yourself!

Tools

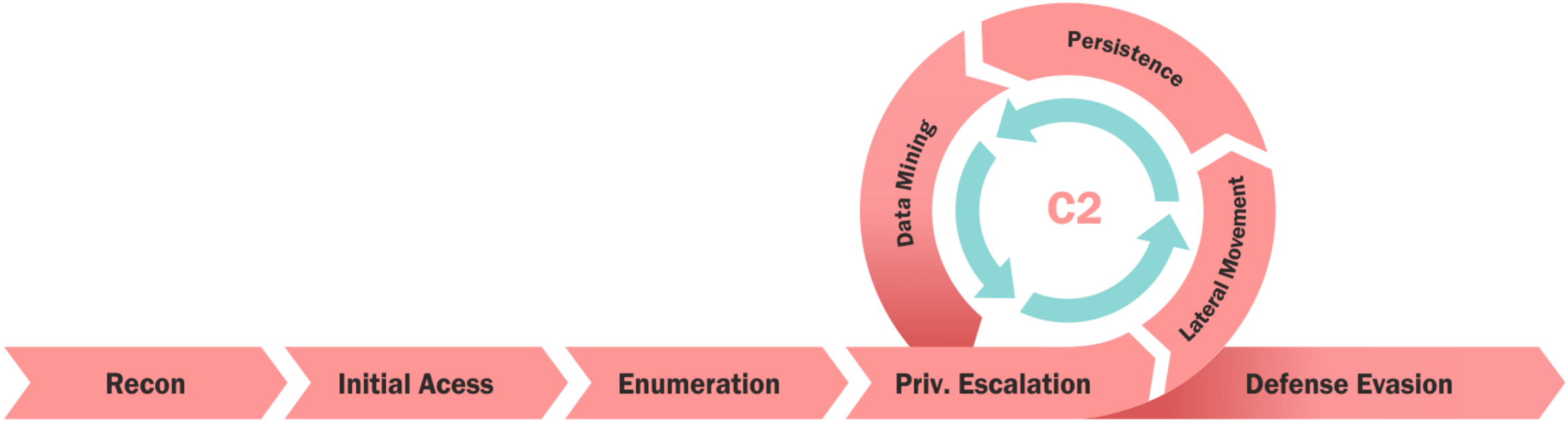
- Microsoft's tools
  - Microsoft Graph PowerShell module (MG Module)
  - Az PowerShell module
  - REST APIs (MSGraph, ARM and others)
  - Some Microsoft portals (A comprehensive list is at - https://msportals.io/)
- Open source PowerShell, .NET and C++ tools

The Lab

- Our target - Oil Corporation - is a fictitious big oil company that is exploiting oil resources across the globe.
- OilCorp is a multi-tenant organization in Azure with resources and applications in multiple tenants related to their departments.
- We are 'No More Oil' - An organization dedicated to stopping big oil!
- Our goal is to compromise OilCorp and gain access to their next big move in exploiting this planet.
- You can access the lab at - https://azureadvanced.enterprisesecurity.io/

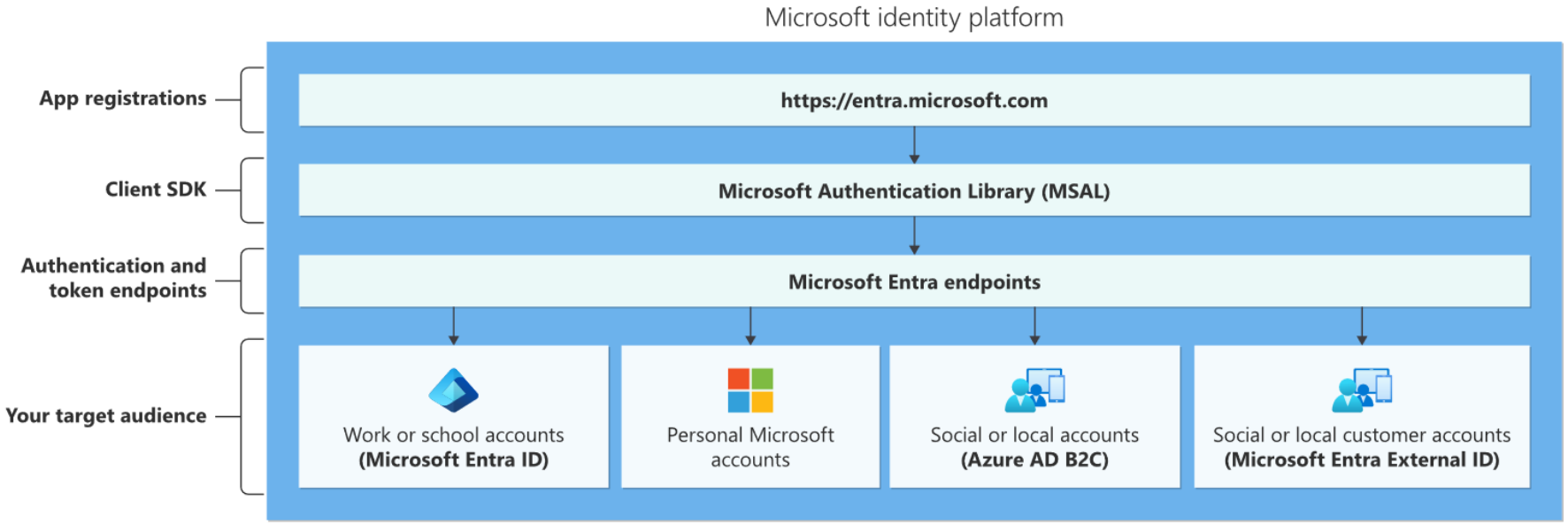

OIL CORPORATION
Attacker Infra
nomoreoil
Student VMs
OILCORP
Exploration
Drilling
Refining
Logistics
RESERVOIR
oilcorpreservoir
GEOLOGY
oilcorpgeology
on-prem resources
AWS
MAINTENANCE
oilcorpmaintenance
on-prem resources
on-prem resources
OPERATIONS
SQL
SQL
oilcropoperations

Azure Kill Chain



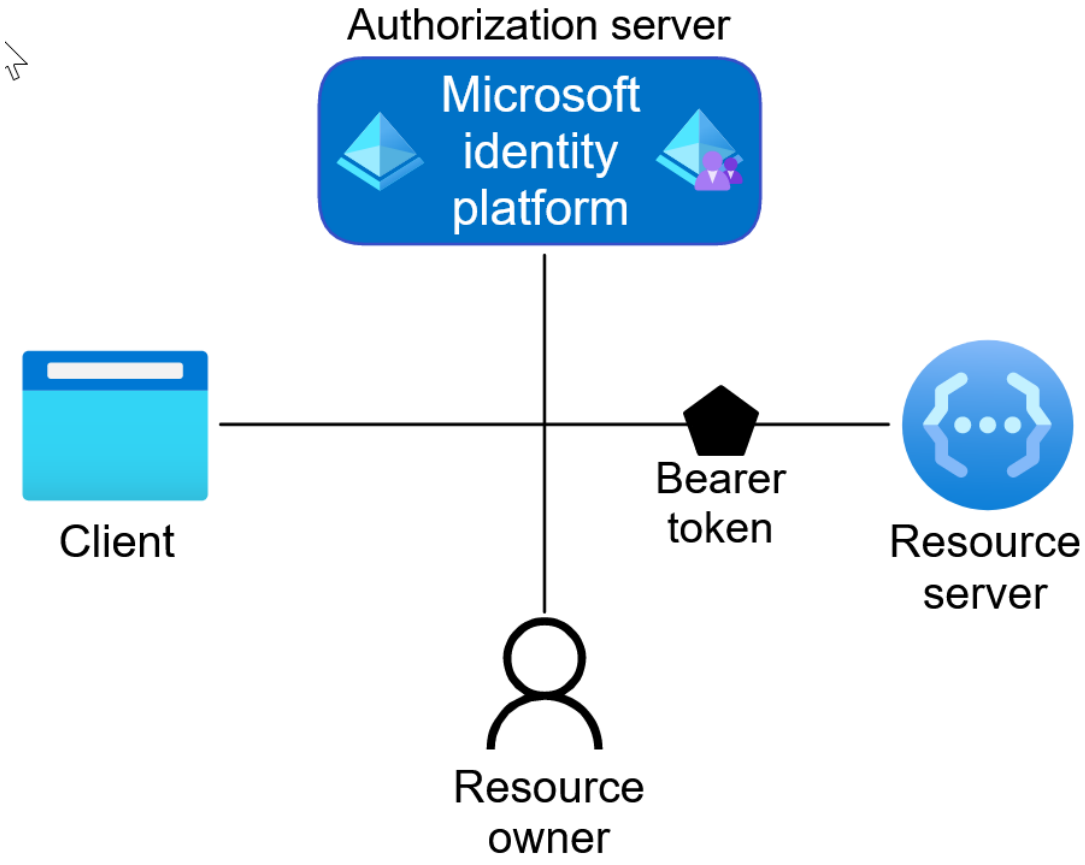
Microsoft Identity Platform

- Microsoft Identity Platform is a cloud identity service that provides authentication and authorization services.
- It is used to access Microsoft Cloud services and custom applications.
- Microsoft Identity Platform supports industry-standard protocols such as OAuth 2.0 and OpenID Connect.



Microsoft Identity Platform - OAuth

- OAuth (Open Authentication) is an open-standard authorization protocol / framework designed for granting access to the resources.
- It enables third-party applications to obtain limited access on resources, either on behalf of the resource owner, or by obtaining the access on its own behalf.
- It allows user's account information to be shared with third-party without exposing the user's credentials.



Microsoft Identity Platform - OAuth

Microsoft Identity Platform - OAuth - Tokens

- The authentication flow uses bearer tokens.
- Identity Platform uses three types of bearer tokens
  - ID token - Contains basic information about the user.
  - Access token - Used to get access to a resource. Prone to token replay attacks. Microsoft struggles to manage them securely. Expriy ranges from 70 minutes to more than 24 hours depending on the type.
  - Refresh token - Can be used to request new Access and ID tokens. Expiry is 90 days for inactive tokens and no expiry for active tokens.

Microsoft Identity Platform - OAuth – Access Tokens

- Valid for limited lifetime (usually 1-2 hour(s))
- Bound to **1 specific user**
- Can be used for **1 specific resource** only
- Can include **MFA claims**!
- Can NOT be revoked before it expires
  - Unless CAE is in use (only available for certain apps)

Microsoft Identity Platform - OAuth – Refresh Tokens

- Valid for **90 days** by default (can be reduced)
- Bound to **1 specific user**
- Used to **obtain new access tokens**
- Can be **refreshed itself infinitely** for an additional 90 days!
  - As long as it is not expired or revoked
- **Can be revoked**
  - When the user resets their password, or manually revokes all active sessions

Microsoft Graph

- The MSGraph is the gateway to Microsoft 365 and Entra ID.
- Provides access to
  - Microsoft 365 core services - Office, OneDrive, Outlook/Exchange, Teams and more.
  - Enterprise Mobility + Security Services - Entra ID, Intune, Defender 365 etc.
  - Windows services - Activities, devices, notifications etc.
  - Dynamics 365 Business Central services - ERP services
- Single API endpoint - https://graph.microsoft.com

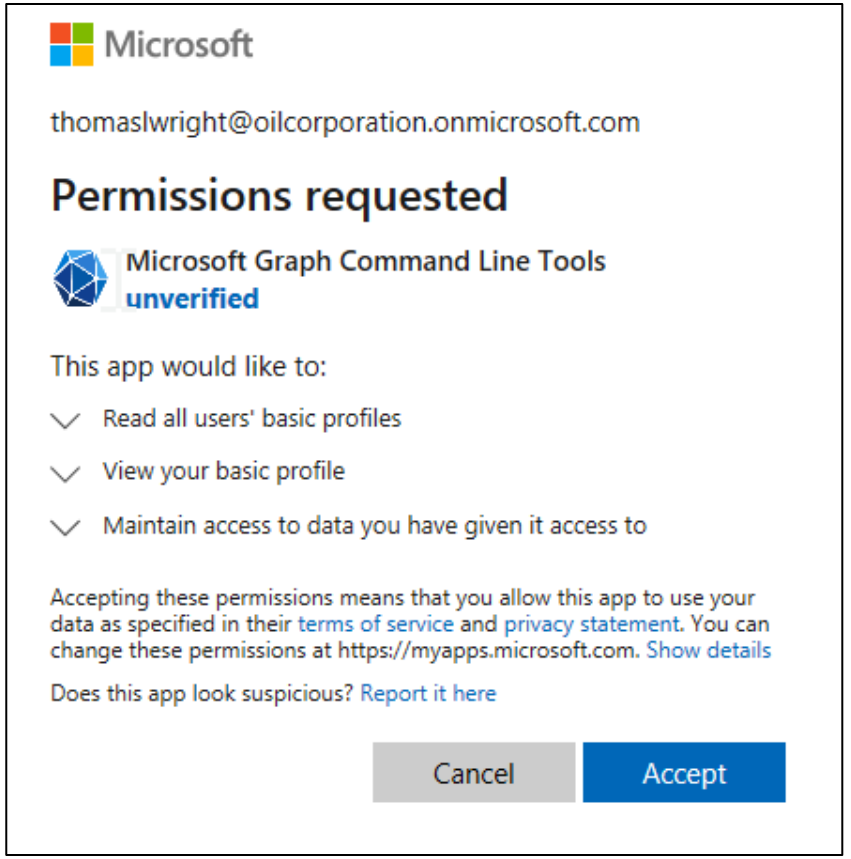
Microsoft Graph PowerShell Module (Mg Module)

- The MSGraph PowerShell module (Mg module) is an API wrapper for MSGraph API.
- Can be used to manage Entra ID and other Microsoft 365 services like SharePoint, OneDrive, Exchange, Teams, Outlook etc.
- Commands in MG module are autogenerated from the Graph API.
- MG module replaces Azure AD module and MSOnline modules.
- Use `Install-Module Microsoft.Graph` to install.

- Not to be confused with Microsoft Graph CLI :P

Mg Module - Authentication

- Mg Module supports delegated access and app-only access.
- For Delegated access (act on behalf of user), consent is required when using the module for the first time.
- For app-only access, admin consent is required.



Mg Module - Sign in with Delegated Access

- To sign in with delegated access, we can use the `Connect-MgGraph` cmdlet.
- Interactive authentication

```
Connect-MgGraph -Scopes "User.Read.All"
```

- Device code flow

```
Connect-MgGraph -Scopes "User.Read.All" -UseDeviceAuthentication
```

- Access token

```
Connect-MgGraph -AccessToken $AccessToken
```

Mg Module - Permission scopes

- For delegated access, we need to specify permission scope and the user consent is required.
- Find required permissions for `Get-MgUser` cmdlet:

```
Find-MgGraphCommand -command Get-MgUser | Select -First 1 -ExpandProperty Permissions
```

Mg Module - Sign in with App-Only

- Using Certificate (`CertificateThumbPrint`, `CertificateName` or `Certificate` from store supported)

```
Connect-MgGraph -ClientId "<AppID>" -TenantId "<TenantID>" -
CertificateThumbprint "<CERT_THUMBPRINT>"
```

- Using Client Secret (Use `Get-Credential` or a PSCredenital object)

```
Connect-MgGraph -TenantId "<TenantID>" -ClientSecretCredential
$ClientSecretCredential
```

- Using Managed Identity

```
Connect-MgGraph -Identity
```

Mg Module - Discovering Commands

- PowerShell help system can be used to discover cmdlets and syntax.

```
Get-Help Get-MgUser -Detailed
```

- PowerShell help system can be used to discover cmdlets and syntax.

```
Get-Command -Module Microsoft.Graph* -Verb Get -Noun *user*
```

Mg Module - Find-MgGraphCommand

- Use `Find-MgGraphCommand` cmdlet to list the cmdlets and API URI that can be used to interact with objects.
- Also lists the required permissions for some cmdlets. Wildcards supported.

```
Find-MgGraphCommand -Command *User*
```

- Use URI wildcard

```
Find-MgGraphCommand -Uri .*users -Method Get
```

Mg Module - Find-MgGraphPermission

- This can be used to list delegated and application permissions for different actions or domain (like application or directory)

```
Find-MgGraphPermission application
```

KC1 Starts

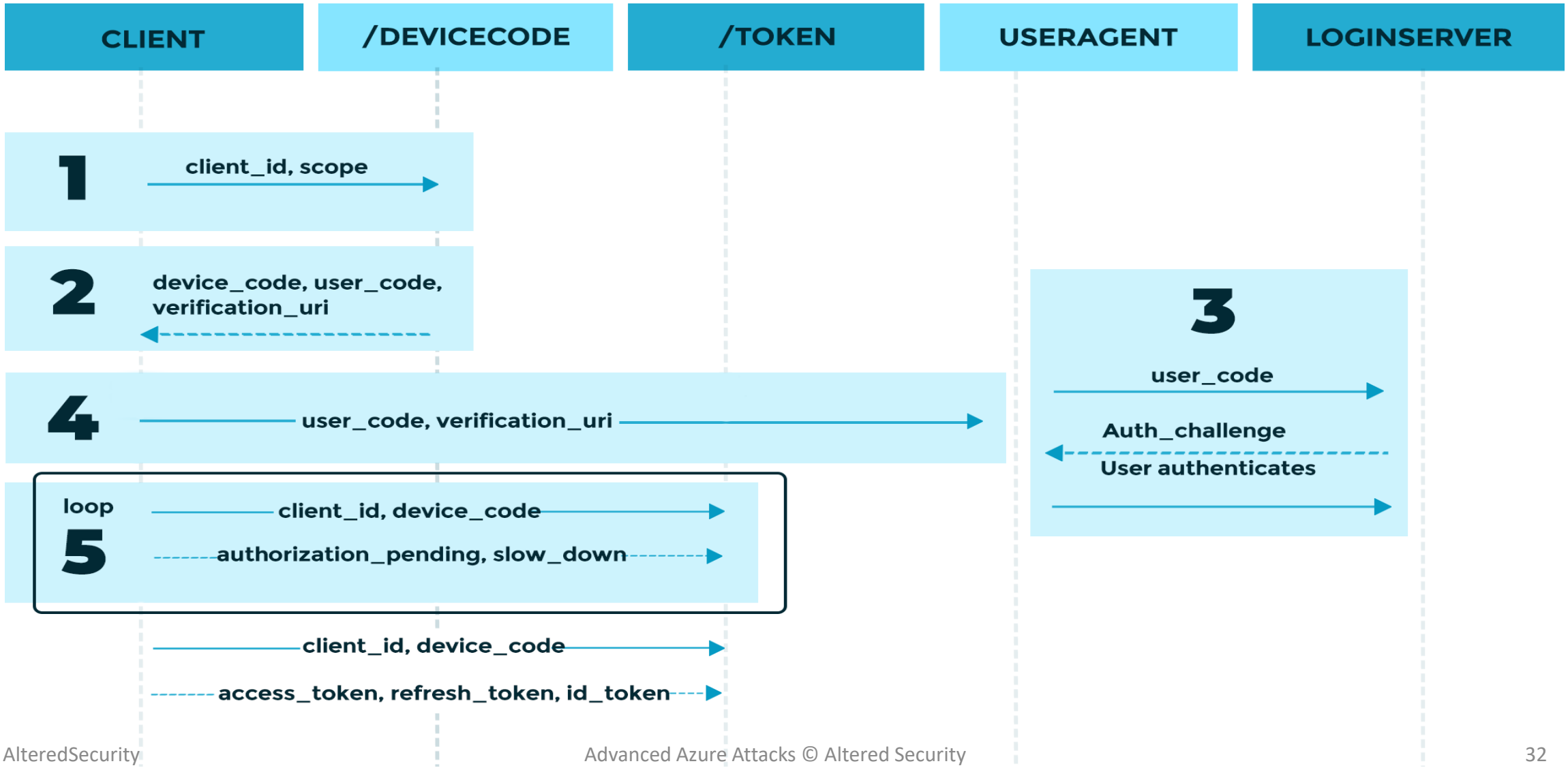
KC1 - Device Authorization Grant Flow

- Used to login to devices that has input limitations (no browser/simple remotes) - like a smart TV!
- An overview of how it works:
  - When the device needs authentication, it provides a code to the user.
  - The user uses another device - a computer or mobile to browse to https://microsoft.com/devicelogin and enter the code.
  - Once the code is entered, the user performs normal authentication including any Consent or MFA is needed.
  - On successful login, the device with input limitations gets the access and refresh tokens.

KC1 - Device Authorization Grant Flow

1. An app that supports device code flow is started on the device and sends the application **client_id** and scope to the **/devicecode** endpoint.
2. The endpoint responds with a JSON object with **device_code**, **user_code** and **verification_uri** (https://microsoft.com/devicelogin) and a few more parameters.
3. The **user_code** and **verification_uri** are displayed to the user.
4. User browses to **verification_uri** on a computer or mobile, enters the **user_code** and completes the normal authentication process.
5. The device keeps polling the **/token** endpoint for **access_token** and **refresh_token** using the **device_code**.

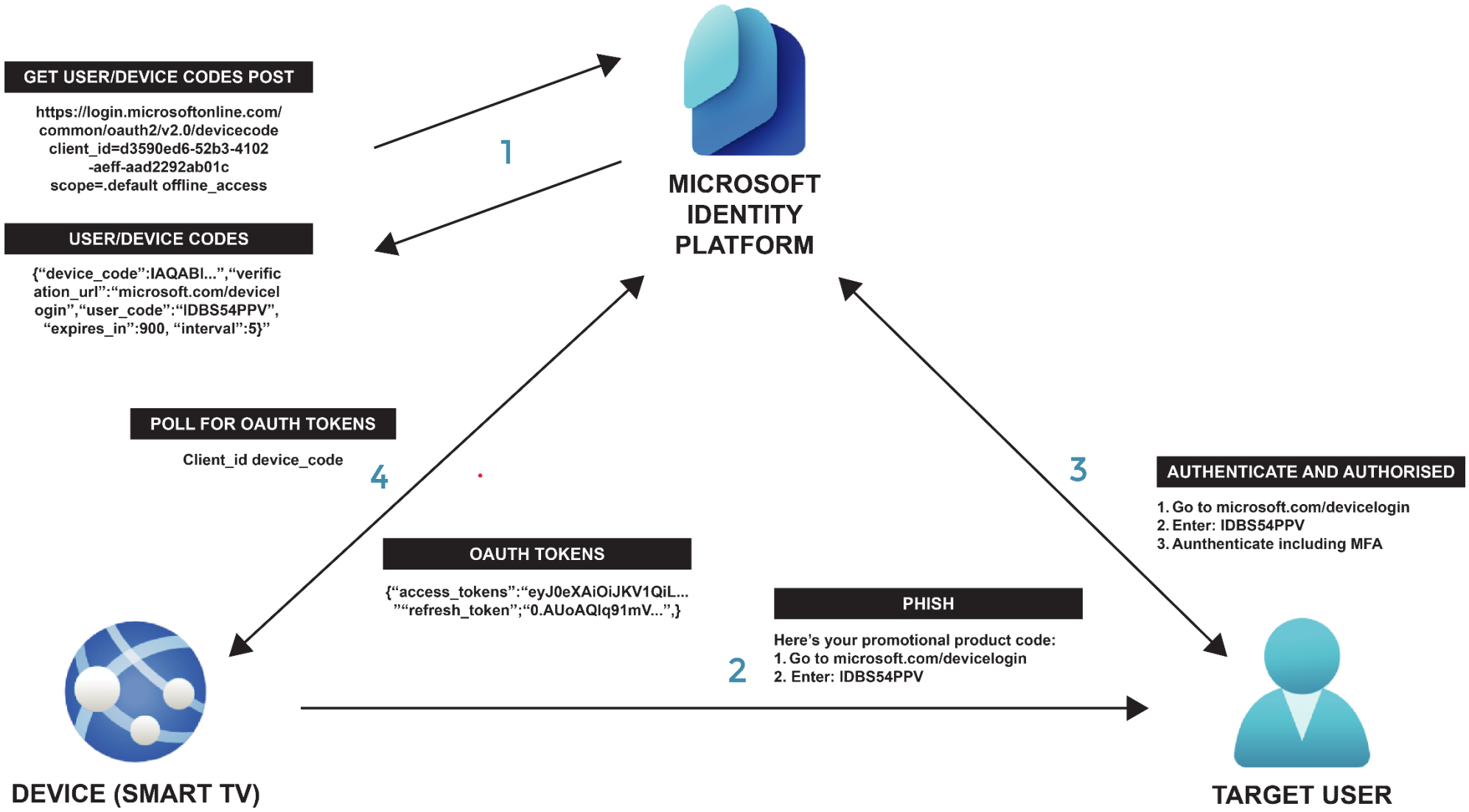
KC1 - Device Authorization Grant Flow



KC1 - Device Code Phishing

- Device Code Flow can be abused for phishing as follows:
  1. Connect to the **/devicecode** endpoint and use a legit **client_id** and **scope** to generate a code.
  2. Send the received **verification_uri** and **user_code** to the victim.
  3. Hope that the victim opens the link, enters code and completes sign-in within 15 minutes.
  4. Request access token and refresh token for the victim.

KC1 - Device Code Phishing



KC1 - Device Code Phishing - Generate Code

- Following PowerShell code can be used to connect to /devicecode endpoint and get a code:

```
$ClientID = "d3590ed6-52b3-4102-aeff-aad2292ab01c"
$Scope = ".default offline_access"
$GrantType = "urn:ietf:params:oauth:grant-type:device_code"

$body = @{
"client_id" = $ClientID
"scope" = $Scope
}

$authResponse = Invoke-RestMethod -UseBasicParsing -Method Post -Uri
"https://login.microsoftonline.com/{tenant}/oauth2/v2.0/devicecode" -Body $body
Write-Output $authResponse
```

Parameter	Value	Description
client_id	d3590ed6-52b3-4102-aeff-aad2292ab01c	We are using "Microsoft Office" client_id. After entering code, victim will see that "You're signing in to **Microsoft Office** on another device located in.."
scope	.default offline_access	We are requesting all permissions (.default) that Microsoft Office requires and we want refresh token too (offline_access)
grant_type	urn:ietf:params:oauth:grant-type:device_code	Must be urn:ietf:params:oauth:grant-type:device_code
tenant	common	Can be common, consumers, organizations, tenant ID or tenant domain

KC1 - Device Code Phishing - Receive Response

- We would get a response like below:

```
user_code        : EBLD8ZFJ8
device_code      : EAQABAAEAAAAtyolDObpQQ5VtlI4uGjEPO0WC1YL5_EzO2qtpwfoRt9xImysn3LGBQAOHR-
CKdQqoS9nblpOeHY1p8isH_R54zYxx05sRjIgXgt8DWWEj1CyDCeFiQ8KT9U4LmNRCHaudjdF3
                   kqo4hIZzOSJfUefu4hcU6smwdaXAO_R6HaEMinmzyFMK4pX-TdigQSpuFg8gAA
verification_uri : https://microsoft.com/devicelogin
expires_in       : 900
interval         : 5
message          : To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code
EBLD8ZFJ8 to authenticate.
```

Parameter	Value	Description
user_code	Received in response	**This needs to be sent to the victim**
device_code	Received in response	Will be used by the device for polling and to request access token
verification_url	Received in response	**This needs to be sent to the victim**
expires_in	Received in response	Expiration time of user_code (in seconds)
interval	Received in response	Polling request interval before device requests for tokens (in seconds)
message	Received in response	Preset message shown to the user.

KC1 - Device Code Phishing - Request Tokens

- Following PowerShell code can be used to request access token using **device_code**:

```
$body=@{
"client_id" = $ClientID
"grant_type" = $GrantType
"code" = $authResponse.device_code
}
$Tokens = Invoke-RestMethod -UseBasicParsing -Method Post -Uri
"https://login.microsoftonline.com/common/oauth2/v2.0/token" -Body $body -ErrorAction SilentlyContinue
$GraphAccessToken = $Tokens.access_token
```

Parameter	Value	Description
tenant	organizations	Must be same as the initial request
grant_type	urn:ietf:params:oauth:grant-type:device_code	Must be urn:ietf:params:oauth:grant-type:device_code
client_id	d3590ed6-52b3-4102-aeff-aad2292ab01c	Must be same as the initial request
device_code	Received in response	Use from the response

KC1 - Device Code Phishing

- Device Code Phishing has some advantages!
  - The target always interacts with legit Microsoft web pages.
  - Bypasses MFA - MFA enforcement can’t stop the attack as a valid user authenticates.
  - Multiple client_id supported - We can mimic Office Applications, Az PowerShell and more.
  - Multiple resources supported! We can use it to request access tokens for Microsoft Graph, ARM and many more services.
  - When using public client_id - No consent is required from the user and a new access token for a different family client application can be requested.

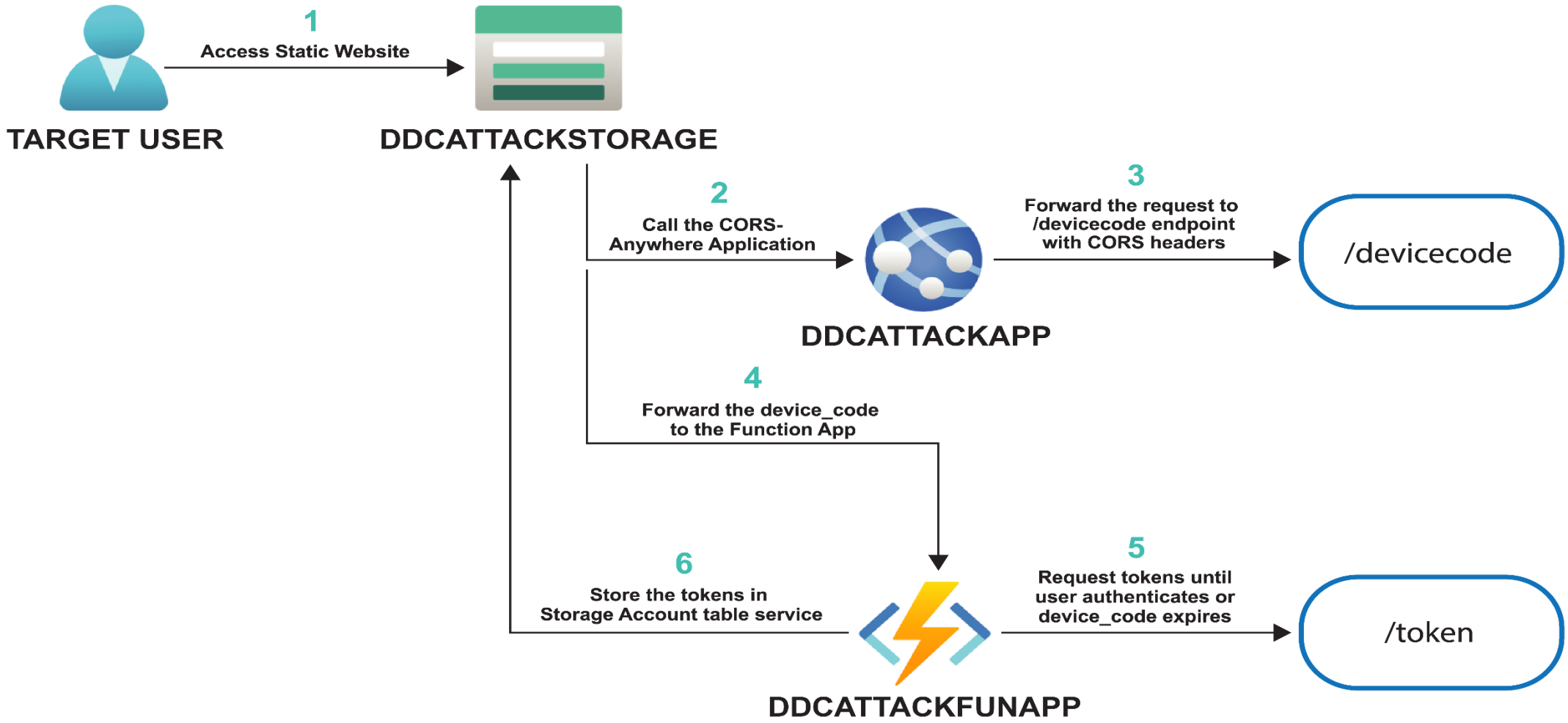
KC1 - Dynamic Device Code Phishing

- Device Code Phishing has one huge limitation - 15 minutes window!
- Dynamic Device Code Phishing solves that!
- In place of generating the user_code, we send a link to the target that generates user_code when the target visits the link.
- So, the 15 minutes window starts when the target opens the link. This means increased chances of success.

KC1 - Dynamic Device Code Phishing - Azure Infrastructure

- To perform Dynamic Device Code Phishing we need to setup the following:
  - Storage Account - To host a static website that fetches device code and stores tokens.
  - App Service - To host CORS-Anywhere (https://github.com/Rob--W/cors-anywhere) application. This application helps us to overcome the CORS (Cross-origin Resource Sharing) limitation. The limitation is that we will trigger CORS while sending the HTTP request via jQuery to the Function App.
  - Function App - To request the tokens and store them to the Storage Account Table service.

KC1 - Dynamic Device Code Phishing - Azure Infrastructure



KC1 - Family Of Client IDs (FOCI)

- Family of Client IDs (FOCI) is a set of Microsoft client applications that are "compatible" with each other.
- Obtaining a "Family Refresh Token (FRT)" can be used to request refresh and access tokens for any other client applications in the family.
- Currently, there is only one family and it contains many "First-Party" Microsoft Applications with "public" client IDs - Office, Teams, Az CLI, Az PowerShell, Microsoft support, OneDrive and so on!
- Many First Party applications are automatically provisioned in tenants and have public client IDs - no consent is required (implied consent or pre-consent)

KC1 - Family Of Client IDs (FOCI) - FRT Abuse

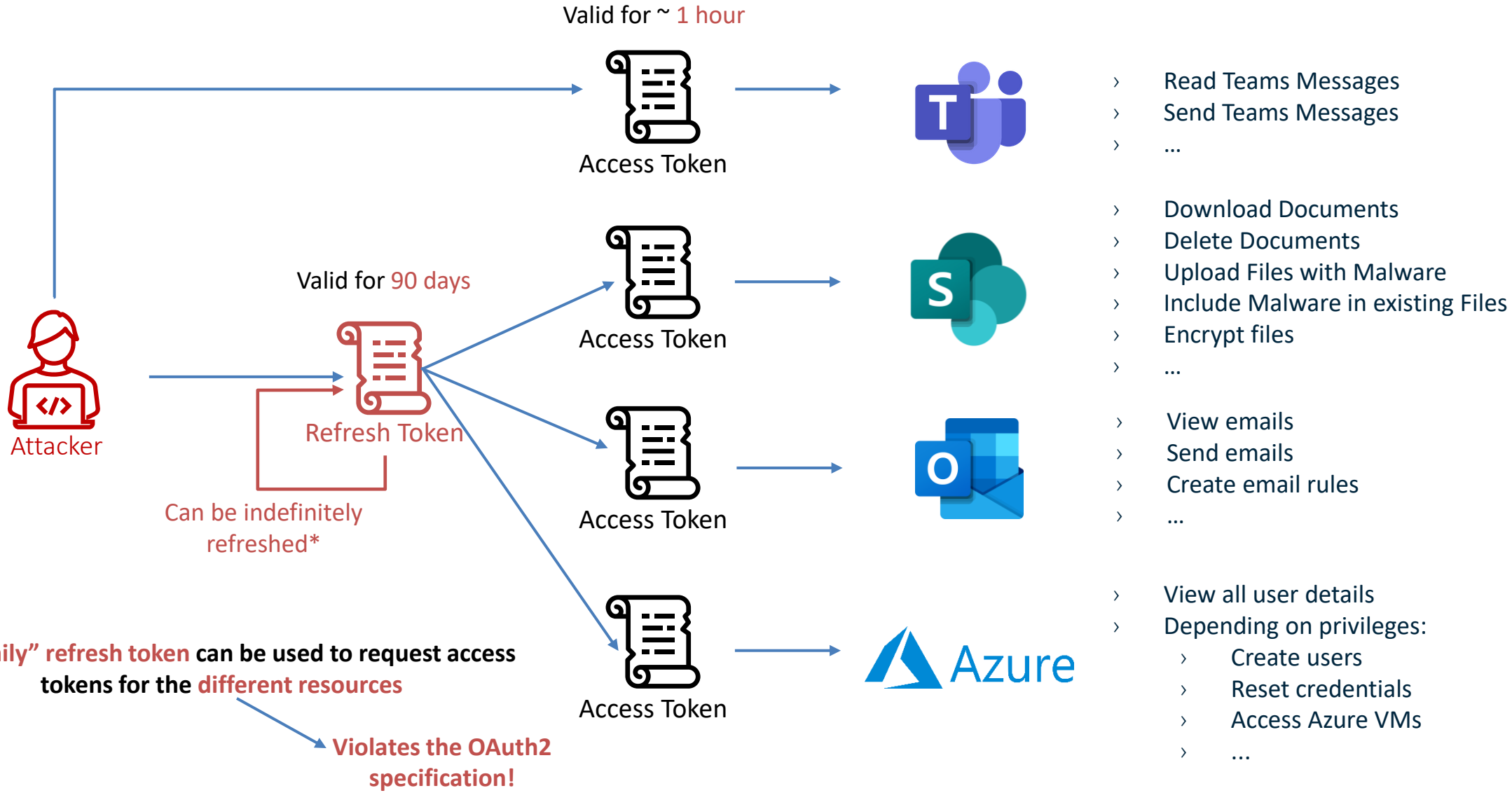
- A "Family Refresh Token (FRT)" is a special refresh token that is NOT bound to client_id or scope.
- That is, it can be used to request refresh and access tokens for any other client applications in the family.
- However, Azure AD role-based level of access cannot be changed (We cannot become a Global Admin using MS Graph token).

KC1 - Family Of Client IDs (FOCI) - FRT Abuse

- This means that using a refresh token for ARM for Az PowerShell client, we can request a refresh token and access token for MSGraph for Office client. This will allow us to access office documents for the user.

- Note that even using a normal refresh token, we can request access token for different scopes. For example, using a MSGraph refresh token to request an access token for ARM.

KC1 - Family Of Client IDs (FOCI) - FRT Abuse



KC1 - Family Of Client IDs (FOCI) - Different Client IDs

- Resource:
  - Microsoft Graph
    - (https://graph.microsoft.com)
- Client ID:
  - **Microsoft Office**
    - (d3590ed6-52b3-4102-aeff-aad2292ab01c)

```
AuditLog.Create AuditLog.Read.All Calendar.ReadWrite Calendars.Read.Shared
Calendars.ReadWrite Contacts.ReadWrite DataLossPreventionPolicy.Evaluate
Directory.AccessAsUser.All Directory.Read.All Files.Read Files.Read.All
Files.ReadWrite.All Group.Read.All Group.ReadWrite.All
InformationProtectionPolicy.Read Mail.ReadWrite Mail.Send Notes.Create
Organization.Read.All People.Read People.Read.All Printer.Read.All
PrinterShare.ReadBasic.All PrintJob.ReadWriteBasic
SensitiveInfoType.Detect SensitiveInfoType.Read.All
SensitivityLabel.Evaluate Tasks.ReadWrite TeamMember.ReadWrite.All
TeamsTab.ReadWriteForChat User.Read.All User.ReadBasic.All
User.ReadWrite Users.Read
```

- Resource:
  - Microsoft Graph
    - (https://graph.microsoft.com)
- Client ID:
  - **Microsoft Teams**
    - (1fec8e78-bce4-4aaf-ab1b-5451cc387264)

```
AppCatalog.Read.All Channel.ReadBasic.All Contacts.ReadWrite.Shared
Files.ReadWrite.All InformationProtectionPolicy.Read Mail.ReadWrite
Mail.Send MailboxSettings.ReadWrite Notes.ReadWrite.All People.Read
Place.Read.All Sites.ReadWrite.All Tasks.ReadWrite Team.ReadBasic.All
TeamsAppInstallation.ReadForTeam TeamsTab.Create User.ReadBasic.All
```

A **"family" refresh token** can be used to request access tokens for the **same resource** but with **different client IDs**

Violates the OAuth2 specification!

KC1 - Device Code Phishing - Defense

- Note that it is the attacker IP and Device that is logged in Entra ID Sign-in logs - as the authentication is initiated from there!
- Look at User sign-ins with Authentication Protocol: Device Code.
- This information can be used for detection as well as prevention.
- Note that the access token acquisition is not logged.

- A location based Conditional Access Policy seems to be an effective defense in this case.
- A Conditional Access Policy based on Authentication flows -> Device code flow specifically targets this attack. You can block or allow Device code flow only for specific identities.

KC1 - Learning Objective - 1

- Find a target email from Oilcorp website - https://explorationportal.z13.web.core.windows.net/
- Use the Device Code Phishing to compromise the user Thomas.
- Use the FRT for Thomas to request access tokens for other Microsoft Applications.
- Extract keys from a Key Vault accessible to the user Thomas.

Part of - Kill Chain 1

Topics covered - Initial Access, Authenticated Enumeration, FOCI and Data Mining

KC1 - JWT Assertion

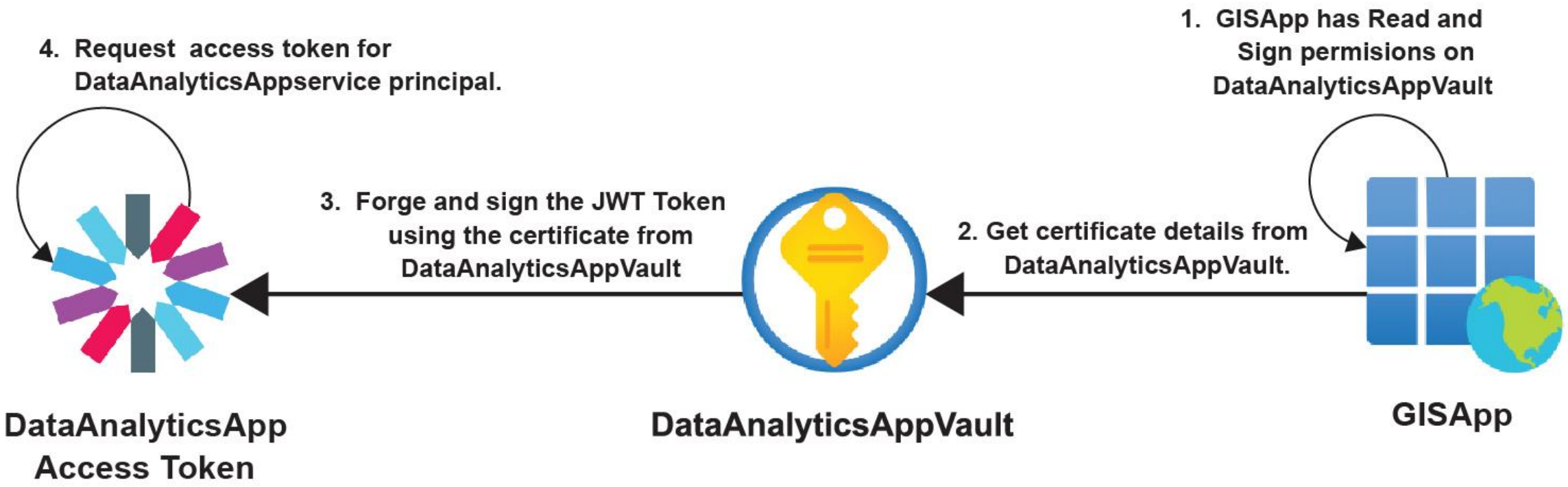
- "The Microsoft identity platform allows an application to use its own credentials for authentication anywhere a client secret could be used"
- JWT assertion signed with an application owned certificate is one such credential.
- An assertion can be used in place of client secret in any of the Token grant flows. For example, to request access tokens.

KC1 - JWT Assertion - Abuse

- If we compromise an application’s certificate, we can sign assertions using the private key.
- A benefit of using the certificate over client secret (maximum life 2 years) is longer expiry time.
- Using JWT assertion is an alternative way of using an application certificate.
- Look at "Key Vault JWT Officer" role on DataAnalyticsAppVault in the lab - It allows signing JWTs using a certificates key but can't export the certificate.

- Note that using JWT assertion does not bypass Conditional Access or Network based restrictions.

KC1 - JWT Assertion - Abuse



KC1 - Learning Objective - 2

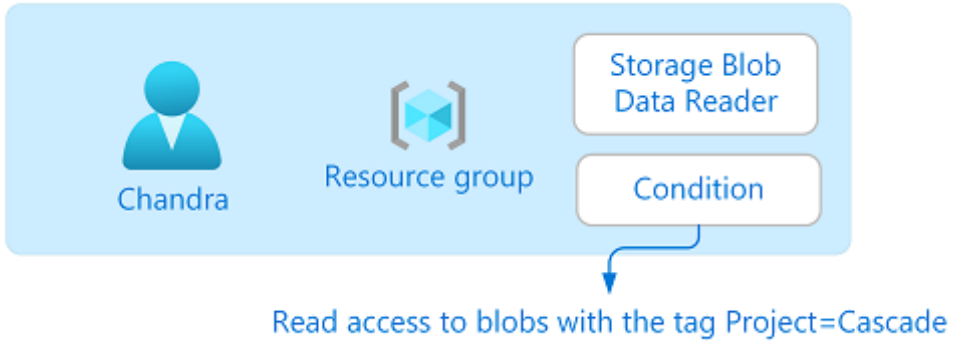
- Find out the application that uses the certificate extracted from GISAppVault key vault.
- Using the certificate, craft and sign a JWT assertion for GISApp and use it request access tokens.
- Enumerate the permissions that GISApp has on DataAnalyticsAppVault key vault.
- Abuse the custom role assigned to the GISApp on the DataAnalyticsAppVault to craft and sign a JWT assertion for DataAnalyticsApp and use it to request access tokens.
- Enumerate the resources that DataAnalyticsApp can access.

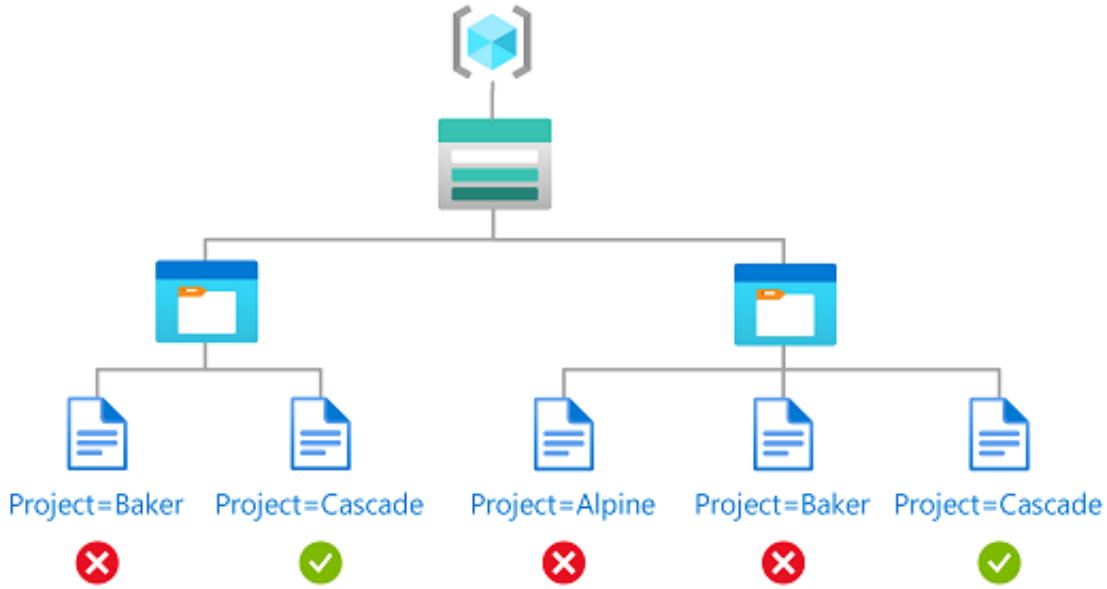
Part of - Kill Chain 1

Topics covered - Authenticated Enumeration, JWT Assertion, Privilege Escalation and Data Mining

KC1 - Attribute-based Access Control (ABAC)

- ABAC builds on RBAC and provides fine-grained access control based on attributes of a resource, security principal and environment.
- Implemented using role assignment condition.





KC1 - ABAC - Abuse

- In case of a storage account, a principal with permission of "Microsoft.Storage/storageAccounts/blobServices/containers/blobs/tags/write" can modify tags.

- This can be abused to get unauthorized access to the stored data if the principal has roles/permissions to read data from the storage account.

KC1 - Application Permissions

- We already know that Entra ID applications need permissions to access resources.

- There are two types of permissions:
  - Delegated permissions - Used for Delegated Access (access on behalf of a user). Needs user or admin consent.
  - Application permissions - Used for App-only access (access without a user). Needs admin consent.

KC1 - Application Permissions

- Application permissions are usually granted to applications that need automation, backup of multiple resources and when "You find yourself tempted to store credentials locally and allow the app to sign in as the user or admin."
- It allows applications to access APIs using its own identity.
- Also called App role assignments.

KC1 - Application Permissions - Abuse

- An application would rarely have MFA or Conditional Access.
- Compromise of an Azure resource that is using an application would result in abuse of application permissions.
- Compromise of any overly permissive application would result in access to more resources and roles.
- In the lab, GeologyApp is one such overly permissive application.

KC1 - Learning Objective - 3

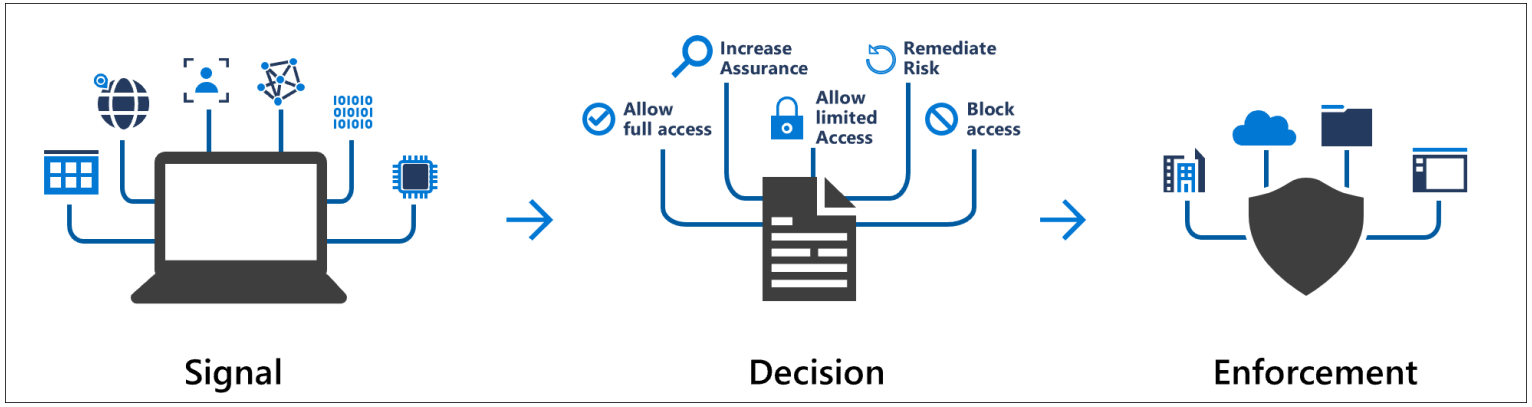
- Enumerate attribute-based access control on a storage account using DataAnalyticsApp.
- Extract secrets for a service principal from a storage account after abusing ABAC.
- Enumerate the Entra ID roles, Azure roles and application permissions assigned to the service principal GeologyApp.

Part of - Kill Chain 1

Topics covered - Authenticated Enumeration, ABAC, Application Permission, Privilege Escalation and Data Mining

KC1 - Conditional Access

- Uses Signals to take Decision on access when an identity wants to access all resources or specified resources.
- Common signals:
  - User or group membership
  - IP Location
  - Device Platform
  - Device State
  - Risk detection
- Common Decisions:
  - Block
  - Grant
  - Grant but require - MFA / Authentication Strength / Compliant Device / Hybrid Joined Device / Approved Client App / App Protection Policy / Password Change / Terms of use



KC1 - Conditional Access - Abuse

- Microsoft recommends use of conditional access very strongly and is a critical part of their "Zero Trust Security Model"
- Conditional access blocks access even if you have correct credentials but do not satisfy conditions.
- It is one of the ways to effectively enforce MFA!
- The ability to enumerate CAPs is important for executing any attack once we have a foothold.

KC1 - Conditional Access - Enumeration

- It is not possible to enumerate CAPs without authentication.
- Even with authentication, you need certain roles or Microsoft Graph API permissions to be able to enumerate CAPs.

- Using the 1.61-internal version of AADGraph API, the deprecated (probably retiring on June 30, 2025) can be used to enumerate CAPs with privileges of any user (workload identities can’t enumerate CAPs).

  "https://graph.windows.net/myorganization/conditionalAccessPolicies?api-version=1.61-internal"
- Roadtools and AADInternals make use of the "1.61-internal" version.

KC1 - Conditional Access - Enumeration

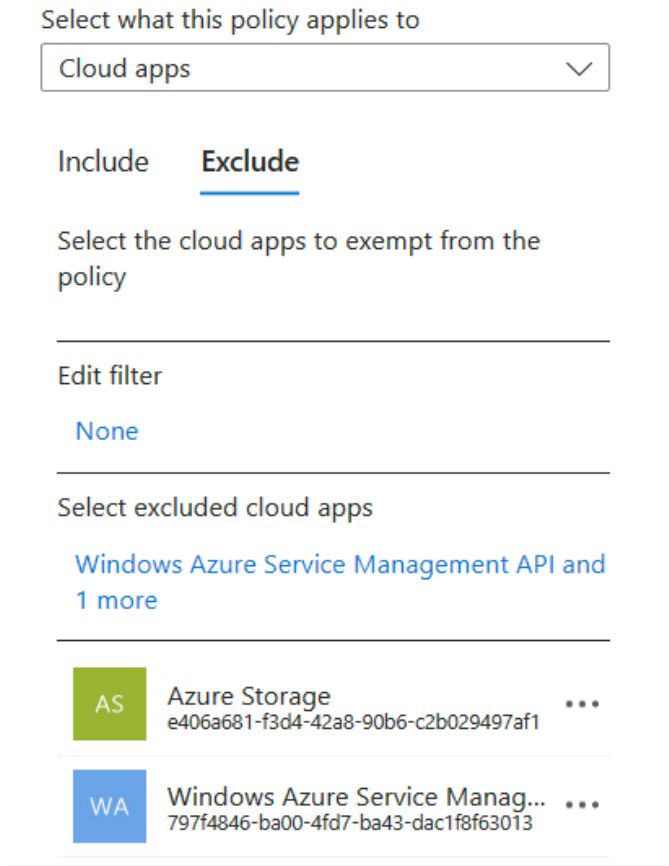
- Following roles can read CAPs:
  - Security Reader
  - Global Reader
  - Security Administrator
  - Conditional Access Administrator
  - Global Administrator

- An application with any of the following permissions:
  - Policy.Read.ConditionalAccess
  - Policy.ReadWrite.ConditionalAccess
  - Policy.Read.All

- Microsoft Graph for PowerShell all of the following:
  - Policy.Read.ConditionalAccess
  - AuditLog.Read.All
  - Directory.Read.All

KC1 - Conditional Access - Target resources

- Due to multiple options in CAPs, numerous scenarios are possible.
- For example, you can configure a policy that allows access only to a certain set of resources or cloud apps.
- This provides granular control. However, if we can enumerate CAPs, the allowed cloud apps can be accessed.



KC1 - Learning Objective - 4

- Abuse the roles and permissions assigned to GeologyApp to:
  - Reset password of StorageMapper**X**.
  - Add client secret to ExpStorageAppSP
- Enumerate the roles assigned to ExpStorageAppSP and abuse that to modify group membership of the StorageAccess group.
- Enumerate the permissions that the StorageAccess group has on Azure resources.
- Enumerate the conditional access policies using GeologyApp.
- Evade MFA enforced by a CAP and access a storage account using membership of the StorageAccess group.

Part of - Kill Chain 1

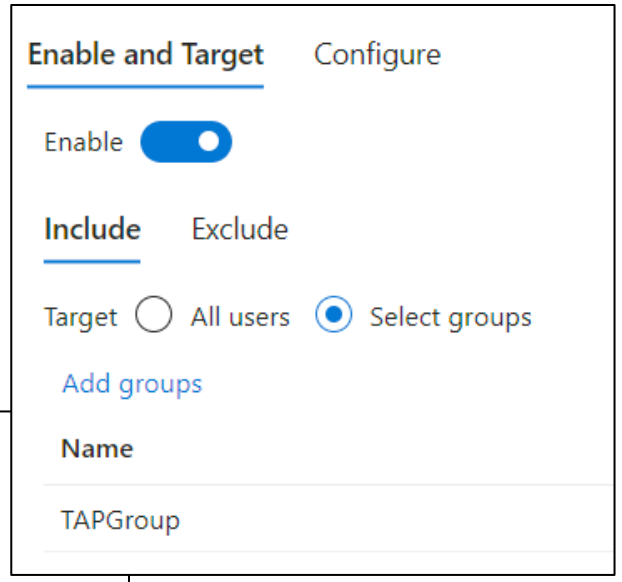
Topics covered - Authenticated Enumeration, Conditional Access Evasion, MFA Evasion and Data Mining

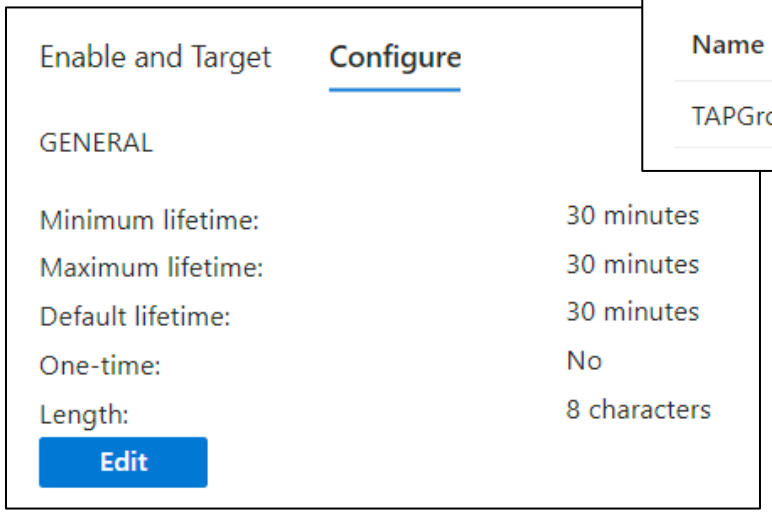
KC1 - Temporary Access Pass (TAP)

- "A Temporary Access Pass is a time-limited passcode that can be configured for single use or multiple."
- By design, TAP:
  - Allows users to access their account _without_ password
  - Satisfies MFA requirement (considered strong authentication in CAPs)
  - Allows users to configure other authentication methods including passwordless methods.
  - Is preferred over Identity Provider (IdP) in federated domains.
  - Works across tenants.

KC1 - Temporary Access Pass (TAP) - Policy

- A Tap Policy defines what users or groups can use TAP and the lifetime of TAP.
- Following roles can enable create TAP policy:
  - Global Administrators
  - Authentication Policy Administrators



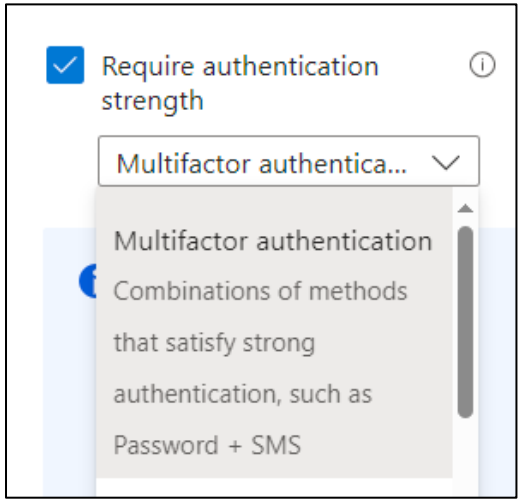


KC1 - Temporary Access Pass (TAP) - Abuse

- Following roles can create a temporary access pass:
  - Global Administrators
  - Privileged Authentication Administrators
  - Authentication Administrators
  - UserAuthenticationMethod.ReadWrite.All
- A Global Reader or role with "Policy.Read.All" can read TAP Policy.

KC1 - Temporary Access Pass (TAP) - Abuse

- Compromising any of the roles with rights to create TAP can be abused!
- Any Conditional Access Policy forcing MFA or authentication strength can be bypassed.
- Look at ‘MFAOnSyncedUsers’ in Oil Corporation.



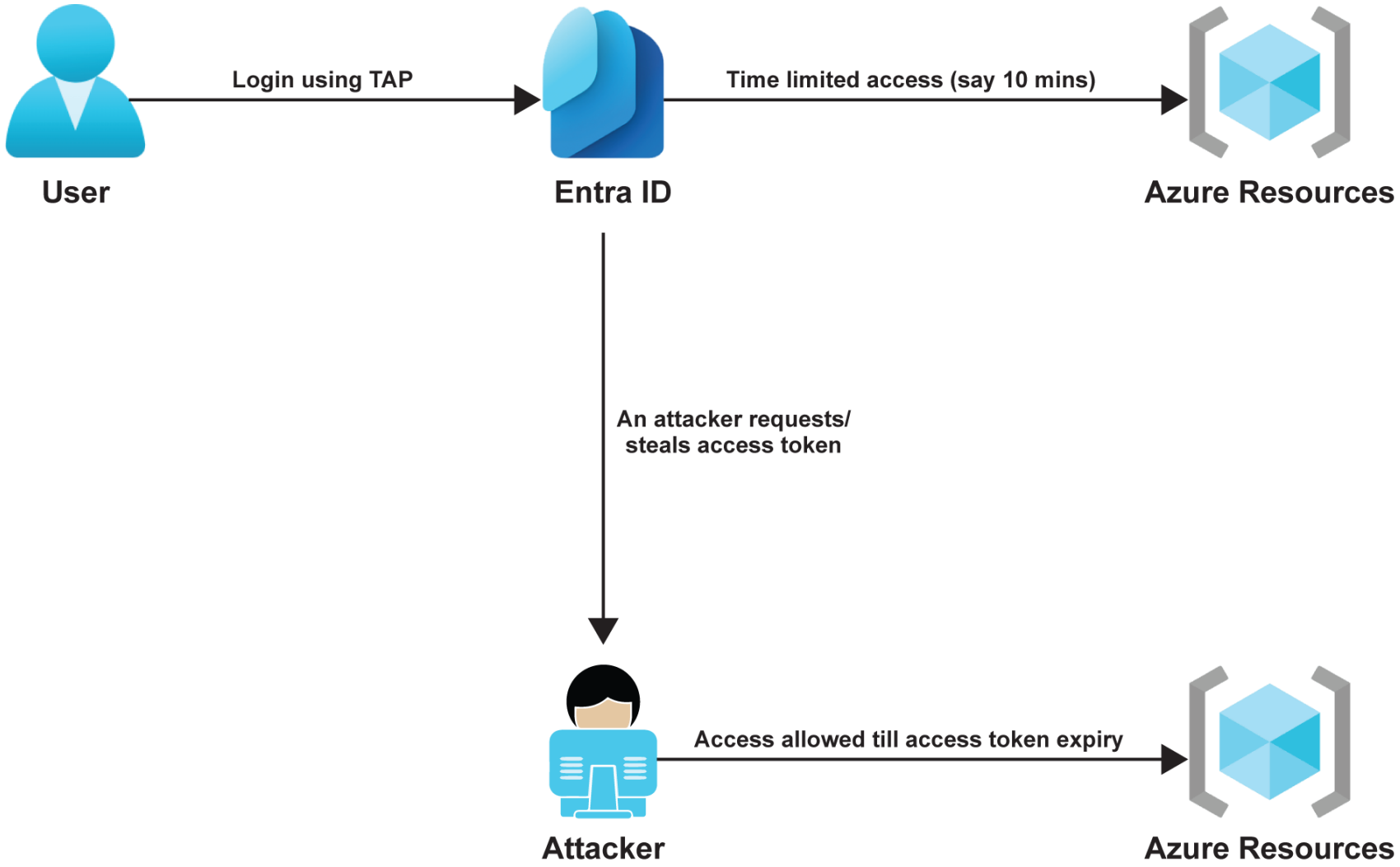
KC1 - Temporary Access Pass (TAP) - Abuse

- Microsoft document mentions this about token expiration:

  "The token lifetime (session token, refresh token, access token, and so on) obtained by using a Temporary Access Pass login is limited to the Temporary Access Pass lifetime. When a Temporary Access Pass expires, it leads to the expiration of the associated token."

- Interestingly, in our testing, that is not the case. The TAP access token for ARM has a validity of 70-75 minutes even if the TAP policy limits it to 10 minutes.
- This defeats the purpose of TAP expiry!

KC1 - Temporary Access Pass (TAP) - Abuse



KC1 - External Identities

- An External identity or External ID refers to scenarios where external users can access resources in a tenant using their own credentials or identity.
- Following are the supported scenarios:
  - B2B Collaboration - Typical guest access. External users visible in current tenant. Enabled by default.
  - B2B direct connect - Mutual, two-way trust is established but external users are not visible in the current tenant. Disabled by default.
  - Azure AD B2C - Similar to Entra ID but a separate service that targets external customer identities.
  - Multi-tenant organization - Create a group of tenants to manage access.

KC1 - External ID - Cross-tenant access settings

- Cross-tenant access settings govern the collaboration with other tenants.
- Some useful terms:
  - Outbound access - If users from current tenant can access resources in other tenants.
  - Inbound access - If users from other tenants can access resources in the current tenant.
  - Trust settings - If claims (MFA, Compliance Device, Hybrid joined devices) would be trusted from other tenants.

KC1 - External ID - Cross-tenant access settings - Default Settings

- Applies to all tenants except the ones configured in Organizational settings.
- B2B collaboration is enabled in Default settings whereas B2B direct connect is disabled.
- Cross-tenant synchronization is not enabled. No sync of users from other tenants to the current tenant.

KC1 - External ID - Cross-tenant access settings - Organizational Settings - Automatic Redemption

- Organizational Settings is used to configure tenant specific settings.
- Both the inbound and outbound trust settings allow Automatic redemption.
- Automatic redemption can be used to suppress the Consent prompt if both the source tenant (outbound) and resource tenant (inbound) has it configured.
- In the lab, this is configured for Oil Corporation (outbound) and Oil Corporation - Geology (inbound). This means that consent experience is suppressed for users from Oil Corp when accessing resources in Oil Corp Geology.

KC1 - External ID - Cross-tenant synchronization

- Cross-tenant synchronization automates lifecycle management of B2B users across tenants.
- It is built upon B2B collaboration and uses the existing B2B cross-tenant access settings.
- It allows provisioning of all or selected users and groups across tenant.
- In the lab, "SyncGroup" of Oil Corp is configured to sync with Oil Corp Geology.

KC1 - Learning Objective - 5

- Using the permissions assigned to GeologyApp, enumerate TAP policy on OilCorp tenant and create TAP for explorationsyncuser**X**.
- Abuse cross-tenant access from Oil Corp to Oil Corp Geology and access the tenant as explorationsyncuser**X**.

Part of - Kill Chain 1

Topics covered - Authenticated Enumeration, Conditional Access Evasion, MFA Evasion and Lateral Movement across tenants

KC1 - Privileged Identity Management (PIM)

- PIM allows just-in-time, time-bound and approval-based access to Entra ID and Azure roles.
- In place of permanent role assignments, using PIM, a user can activate a role for privileged tasks.
- PIM can be used for
  - Entra roles
  - Azure roles
  - Groups - Access to member and Owner role of a security group.
- Note that the classic subscription administrator roles (Account Administrator, Service Administrator and Co-Administrator) can’t be managed using PIM.

KC1 - Privileged Identity Management (PIM)

- Depending on role settings, a role activation can be configured to require:
  - MFA
  - Conditional access authentication context
  - Justification/Approval by selected Approvers/Ticket information

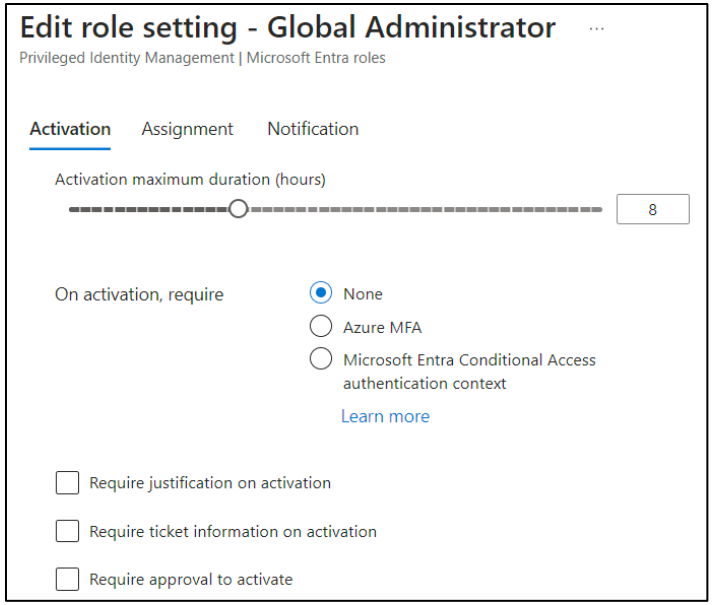
- PIM also enables notifications when roles are assigned and activated.
- Access Reviews helps in removing dormant role assignments.

KC1 - PIM - Types of assignments

- Eligible assignment - A user can activate the role for privileged tasks.
- Active assignment - A user has the role assigned.
- Both the assignments could be permanent or time-bound.
  - Permanent eligible
  - Permanent active
  - Time-bound eligible
  - Time-bound active

KC1 - PIM - Abuse

- Role activation needs to be configured properly.
- A user with Global Administrator or Privileged Role administrator can make changes to role settings in PIM.
- No notifications are sent on the above changes but logs are created in "Resource Audit".

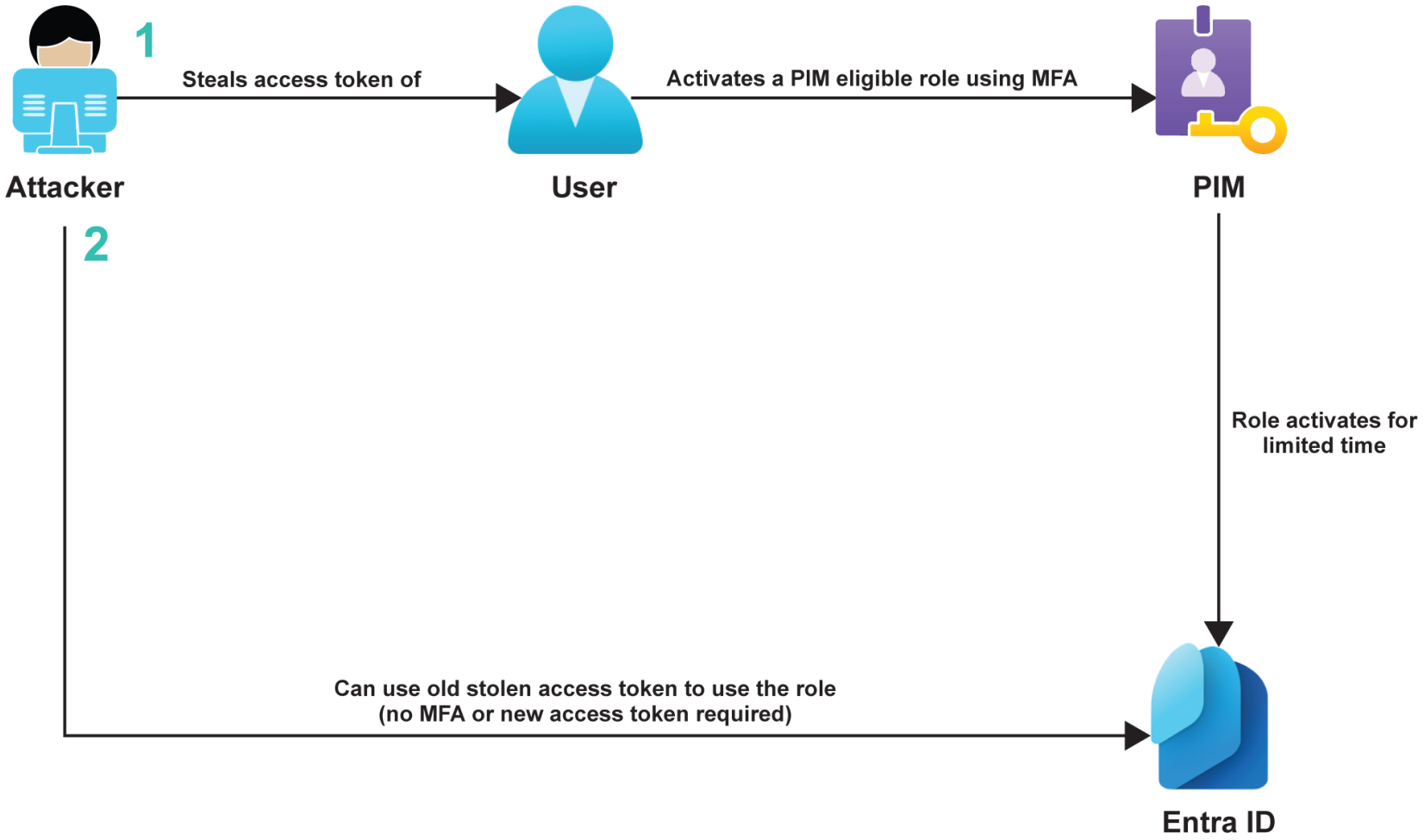


- Roles can still be assigned outside of PIM.
- However, above results in an email notification.

KC1 - PIM - Access token

- We found out two issues during our testing of PIM. Both related to access token.
- If a user's access token is stolen and then they activate a role using PIM
  1. MFA is not enforced by PIM on the stolen access token.
  2. Privileges of the new activated role are available with the stolen access token!

KC1 - PIM - Access token



KC1 - Learning Objective - 6

- Enumerate eligible role assignments for explorationsyncuser**X** in the OilCorp Geology tenant.
- Steal the access token of privuser by using overly permissive web app (https://secureiam.azurewebsites.net/) in OilCorp Geology and use it to evade PIM based privileges and MFA.
- Use the access token to access secrets from a key vault.

Part of - Kill Chain 1

Topics covered - Authenticated Enumeration, PIM bypass, MFA Evasion and Lateral Movement across tenants

KC2 Starts

KC2 - Entra ID Applications

- An Entra ID application is required to use Entra ID’s identity and access management service.
- An application or "App Registration" enables use of Entra ID features for applications like Authentication, Authorization, Role-based access, JWTs with custom claims, SSO, API access and so on.
- Required to integrate an application with Entra ID and Microsoft identity platform.

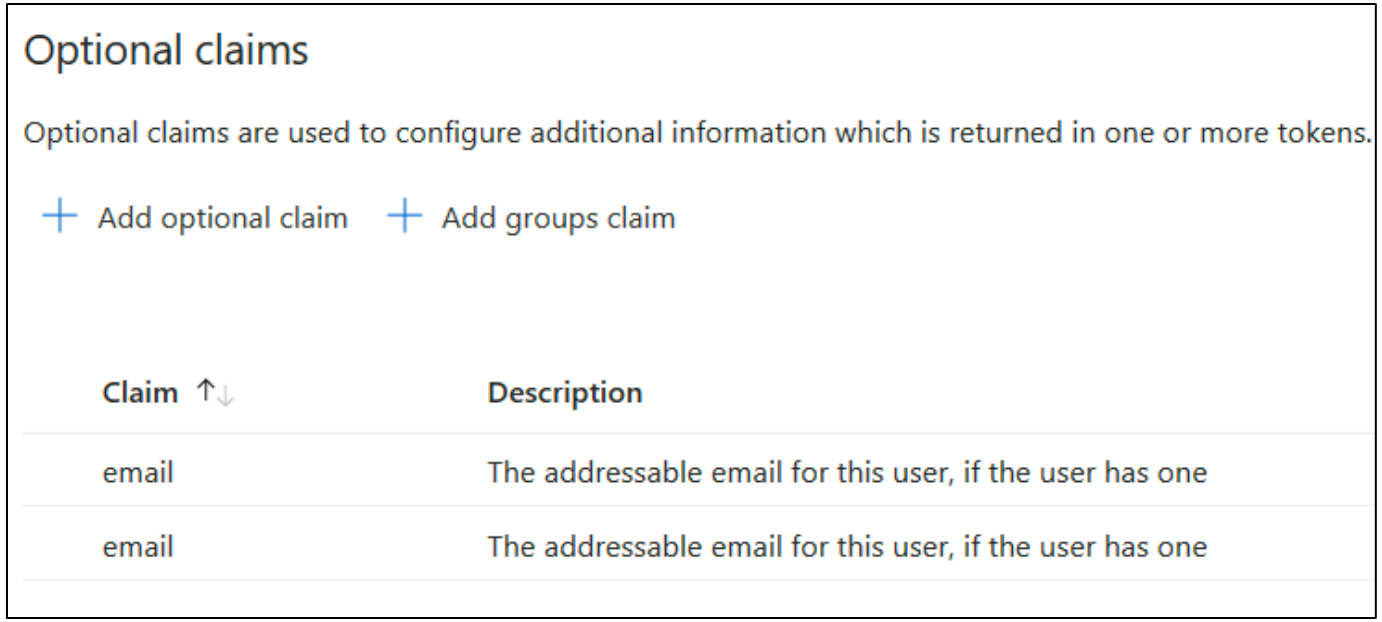
KC2 - Entra ID Applications - Claims

- Claims are included in JWTs and contain information about an identity.
- "Claims are name or value pairs that relay facts about the token subject."
- For example, a claim for a user identity may have upn, unique_name, roles, email etc.
- Claims are used for role-based access, authentication and authorization, SSOs and more.


```
  "unique_name":
"ThomasLWright@oilcorporation.onmicrosoft.com",
  "upn":
"ThomasLWright@oilcorporation.onmicrosoft.com",
  "uti": "ppgudHO_XUuBTUA_nrQOAA",
  "ver": "1.0",
```

KC2 - Entra ID Applications - Optional Claims

- Optional Claims are not included by default but are used to add functionality for a particular application.
- Common optional claims are:
  - Role Claims
  - Group Claims
  - Email Claims
  - Custom Claims



KC2 - Entra ID Applications - Abusing Claims

- Abusing claims is possible if they are not properly handled.
  - A mutable claim like email, preferred_username or unique_name is used for user identification or authorization (termed false identifier anti-pattern by MS).
  - A misconfigured or malicious claims transformation (customizing use of claims by specific application) can be abused for account takeover or privilege escalation.
  - Optional claims may also end up having sensitive information and unnecessary high privileges.

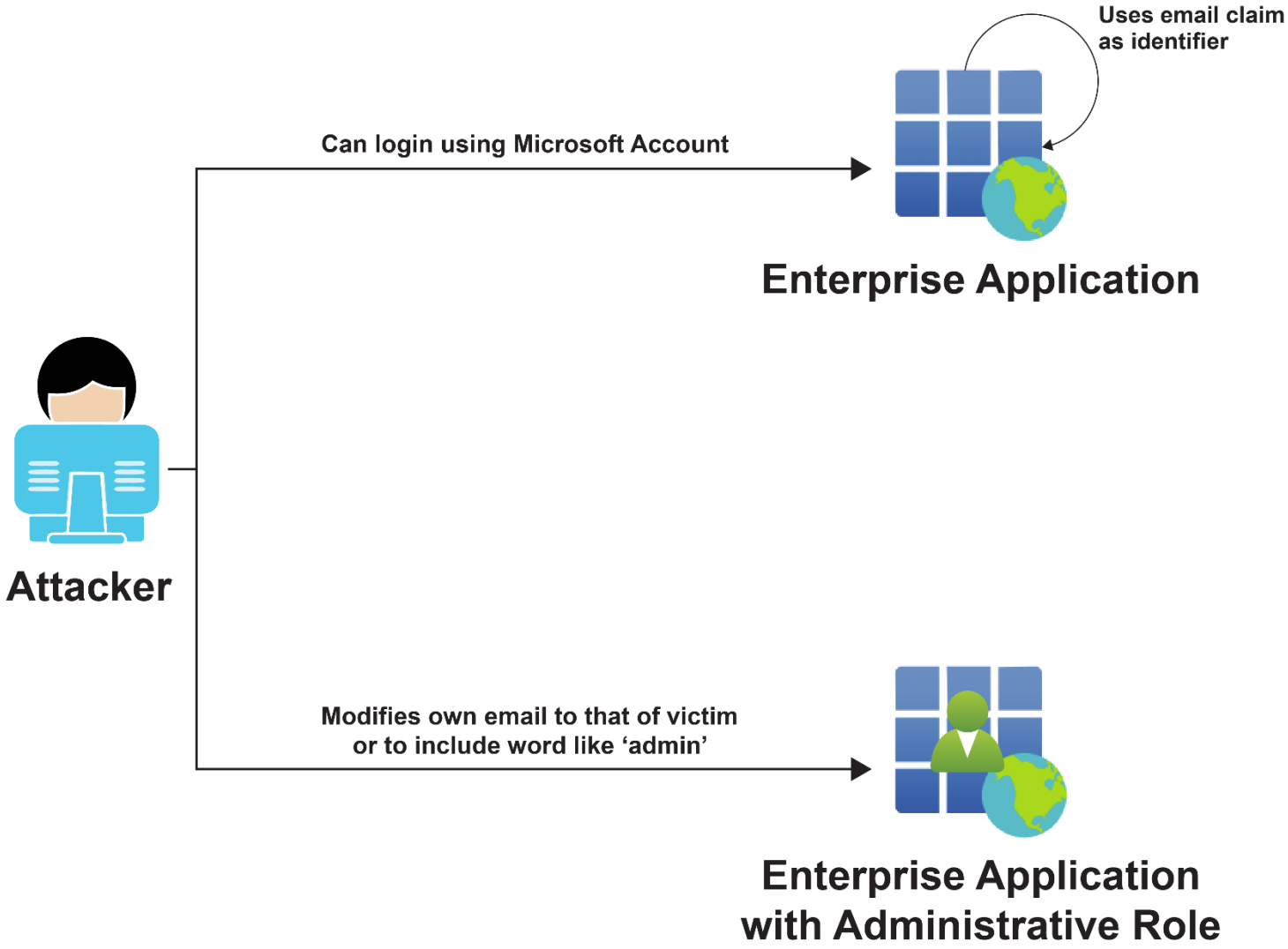
KC2 - Entra ID Applications - Abusing Claims - nOAuth

- "In Microsoft Azure AD, the email claim is both mutable and unverified so it should never be trusted or used as an identifier"

- If an Azure AD application uses email as identifier, it is vulnerable to nOAuth.

- An attacker can assign a target’s email to their user in their own Azure AD tenant and use that to access the target Azure AD application in another tenant as the target user.

KC2 - Entra ID Applications - Abusing Claims - email



KC2 - Entra ID Applications - Abusing Claims - Defense

- Use of "sub" Subject claim as the unique identifier.
- Use "xms_edov" optional claim to check for email verification.

- Note that since June 2023, email addresses with unverified domain are removed from the tokens. However, if required it can be turned off.

- In the lab, we have a vulnerable web application that uses email as a unique identifier.

KC2 - Learning Objective - 7

- Access the Drill Planning (https://drillplanning.azurewebsites.net) web application as student**X**@nomoreoil.onmicrosoft.com from the attacker tenant.
- Modify email of your own user in the attacker tenant to get higher privileges on the Drill Planning Application

Part of - Kill Chain 2

Topics covered - Initial Access and Claims Abuse

KC2 - Azure Logic Apps

- One of the (currently) four workflow creation services in Azure - "where you can create and run automated workflows with little to no code."
- Provides a visual designer to create workflows without code.

- As per MS documentation, logic apps can be used for:
  - Email notifications when a specific event happens
  - Moving uploaded files from a FTP server to Azure Storage
  - Monitor social media

KC2 - Azure Logic Apps - Terms

- Definition - A JSON document that contains the structure of Logic App.
- Workflow - A series of steps that the logic app wants to execute.
- Trigger - A condition that starts the workflow - like a HTTP request, new file upload, new email etc.
- Action - The operation/task that is performed.
- Connector - Used to work with other resources and data.
- Parameter - Used to trigger different action based on input value.
- Callback URL/Workflow URL - Used to trigger Logic Apps using HTTP request.

KC2 - Azure Logic Apps - Workflow

- Logic app workflow supports many statements to control workflow execution. Note that these are like many programming languages.
  - Conditional statements - To run an action based on true or false returned by the statement.
  - Switch statements - To run an action based on a case.
  - Branches - To run actions parallelly.
  - Loops - To repeat actions.
  - Scopes - To run actions only after another group of actions succeed or fail.

KC2 - Azure Logic Apps - Authorization - Consumption Workflows

- Following are specific roles for Consumption workflows (Pay as you go):

Role	Description
Logic App Contributor	Can manage workflows. Can’t change access to them.
Logic App Operator	Can read, enable or disable workflows. Can’t edit or update them.
Contributor	Full access but can’t assign roles.

KC2 - Azure Logic Apps - Authorization - Standard Workflows (Preview)

- Following are specific roles for Standard workflows (based on App Service Plan):

Role	Description
Logic Apps Standard Reader	Full read access.
Logic Apps Standard Operator	Can enable, resubmit and disable workflows and create connections. Can't edit workflows or settings.
Logic Apps Standard Developer	Can create and edit workflows, connections and settings. Can't make application-wide changes like changes to network access control.
Logic Apps Standard Contributor	Full access but can't assign roles.

KC2 - Azure Logic Apps - Abuse

- Run history may contain sensitive information like passwords and secrets (can be obfuscated and/or IP restrcited).
- If IP based restriction and authentication is not enabled for HTTP triggers, anyone who knows the callback URL can trigger the logic app.
- A user with "Microsoft.Logic/workflows/read" permissions can read the logic app workflow that may contain sensitive information like passwords, secrets and input parameters .

KC2 - Learning Objective - 8

- Using the privileges of WellPlannerX user, read the workflow of a logic app.
- Using the information from the workflow, trigger another logic app that creates simulationuser_X.
- As simulationuser_X, bypass MFA enforced by CAP by registering MFA.
- As simulationuser_X, access https://reservoirmgmtapp.azurewebsites.net to abuse B2B Collaboration between Oil Corp and Oil Corp Reservoir tenant.

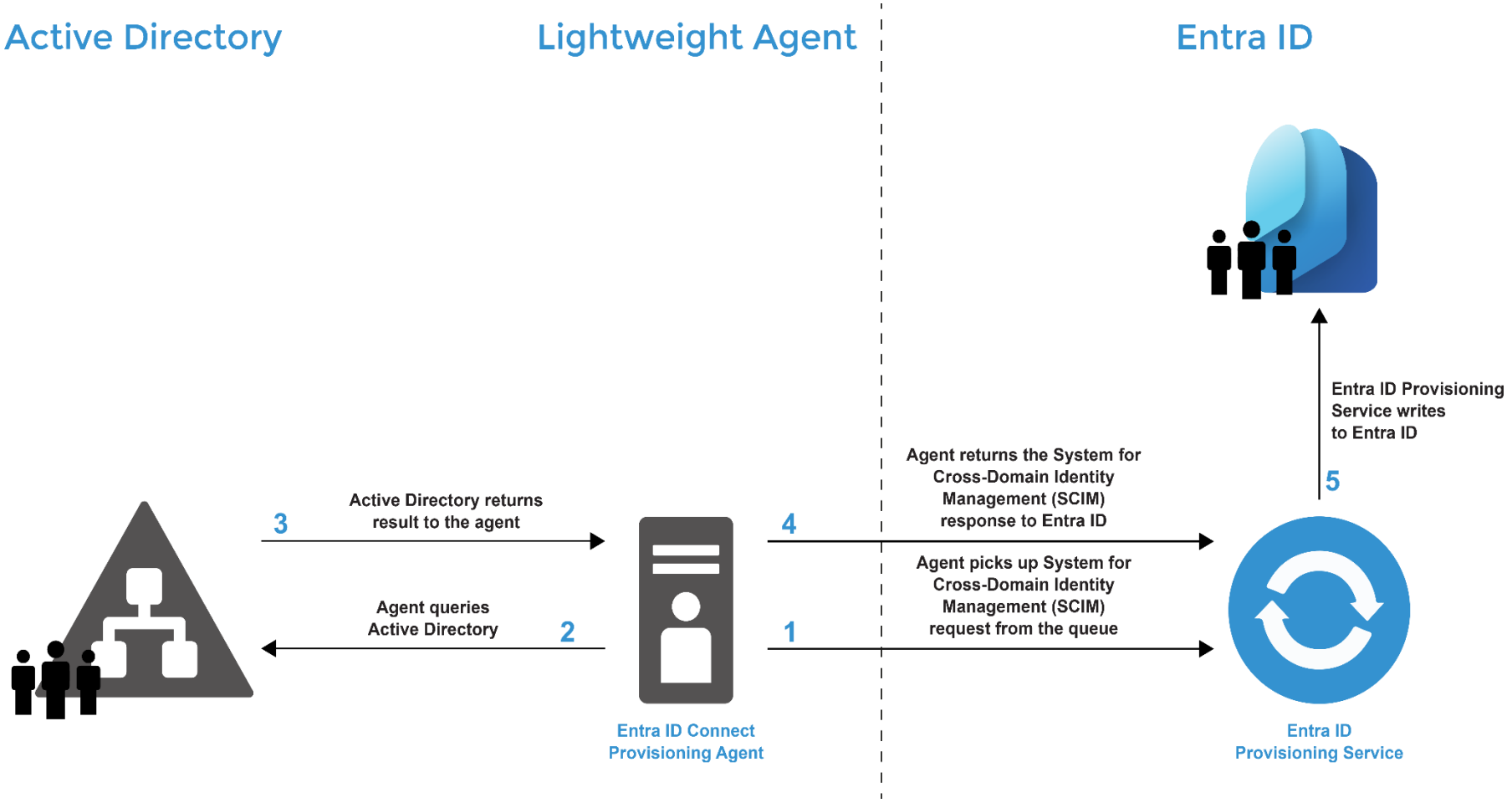
Part of - Kill Chain 2

Topics covered - Logic Apps, Privilege Escalation, Authenticated Enumeration, MFA Evasion and Lateral Movement across tenants

KC2 - Hybrid Identity- Microsoft Entra Cloud Sync

- Cloud sync is an alternative to Connect sync (Azure AD Connect) to get a hybrid identity.
- Cloud sync supports Password Hash Sync (PHS) and Federation for hybrid identity.
- It uses a provisioning agent that works as a bridge between on-prem AD and Entra ID. Similar to app proxy and Pass Through Authentication agent.

KC2 - Hybrid Identity- Microsoft Entra Cloud Sync

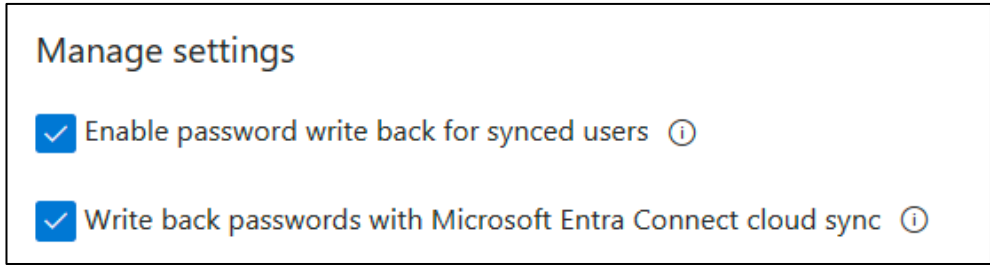


KC2 - Hybrid Identity- Microsoft Entra Cloud Sync - Abuse

- When the provisioning agent is installed, its service runs as a General Managed Service Account (GMSA) with the name pGMSA_<isntallationID>
- If Password Hash Sync is used, this account has Replication rights to the forest that is synced.
- An 'On-Premises Directory Synchornization Service Account' (ADToAADSyncServiceAccount@tenant) is created in Entra ID.

KC2 - Hybrid Identity- Microsoft Entra Cloud Sync - Cloud to On-Prem Lateral Movement

- Cloud sync supports password write-back to on-prem AD.
- If a user with password reset role in Entra ID is compromised, they can reset password for synced user.
- If the synced users has interesting permissions in the on-prem AD, it can be abused.



- Default security groups are not synchronized.
- In the lab, an Entra ID group 'OU Admins Sync' has User Administrator role that can be abused to reset password of synced users.

KC2 - Hybrid Identity- Microsoft Entra Cloud Sync - Persistence

- When the provisioning agent is installed, its service runs as a General Managed Service Account (GMSA) with the name pGMSA_<isntallationID>
- If Password Hash Sync is used, this account has Replication rights (DCSync) to the forest that is synced.
- By extracting the pGMSA user credentials, it is possible to persist in the on-prem environment and get DA on-demand.
- With DA privileges in the on-prem, it could be possible to move back to cloud - for example, by backdooring the provisioning agent.

KC2 - Learning Objective - 9

- As simulationuser_**X**, reset password of hybriduser**X** in the OilCorp Reservoir tenant.
- Enumerate the permissions of hybriduser**X** and execute the DCSync attack against the on-prem domain reservoirone.corp
- Enumerate and compromise an on-prem forest trust for reservoirone.corp and move to reservoirtwo.corp
- Compromise the pGMSA_f9d2bf93$ account in reservoirone.corp and replay its credentials for persistence.

Part of - Kill Chain 2

Topics covered - Hybrid Identity, Cloud Sync, Cloud to on-prem lateral movement, Persistence and Lateral Movement across on-prem forests.

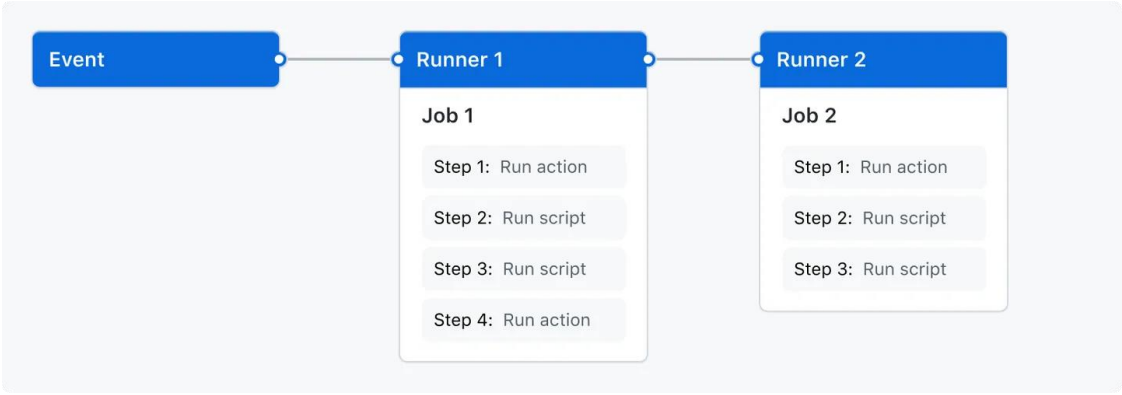
KC3 Starts

KC3 - Initial Access - GitHub

- GitHub Repos are always open for abuse because of the huge attack surface.
- Some of the most common abuses are
  - Secrets (credentials, API Keys, Tokens) in repos
  - Compromising a user with commit rights
  - Hosting malware/Use as C2 channel
  - Abusing GitHub actions and workflows to trigger builds/execute code/perform an action.

KC3 - Initial Access - GitHub Actions

- "GitHub Actions goes beyond just DevOps and lets you run workflows when other events happen in your repository."
- A workflow is an automated process that will run when triggered by an event and will run one or more jobs.
- GitHub provides Linux, Windows and macOS VMs to run the workflows.



KC3 - Initial Access - GitHub Actions - Components

- Event - Triggers a workflow. Could be a pull request, an issue is opened, a commit, a REST API call or a schedule.
- Runner - VM that runs a workflow. Could be Ubuntu, Windows and macOS.
- Job - A step in a workflow. Coule be a shell script or a custom action.

KC3 - Initial Access - GitHub Actions - Abuse

- For a repo with interesting or privileged workflows, the ability to even open an issue could be abused.
- An attacker can open an issue in such a repository that could result in a privileged activity depending on the jobs in the workflow.

- In our lab, we are going to abuse a GitHub repository that triggers a workflow when an issue is opened.

KC3 - Learning Objective - 10

- Access the Refining Wiki (https://refiningwiki.z13.web.core.windows.net) web application.
- Try to find secrets in the Wiki.
- Use the secrets from the Wiki to abuse GitHub actions on one of the repositories of the Oilcorp organization.

Part of - Kill Chain 3

Topics covered - Initial Access and GitHub actions abuse

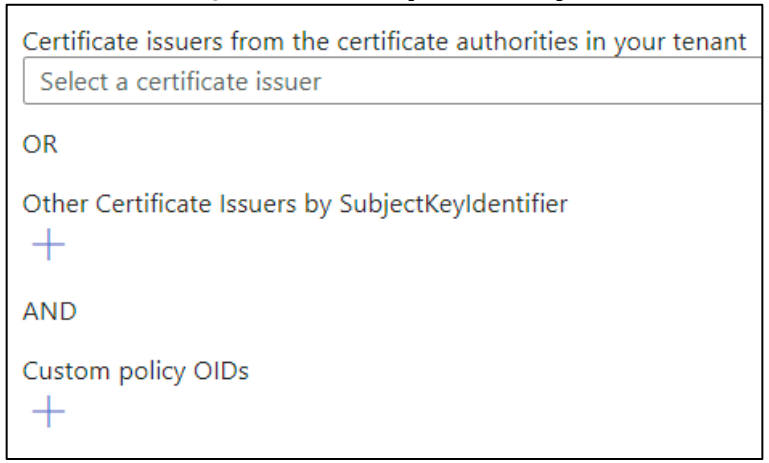
KC3 - Authentication Strength

- A CAP can enforce authentication strength for MFA (discussed in KC1 for TAP).
- There are three built-in authentication strength in increasing order:
  - Multifactor authentication (like Password + SMS, TAP, Federated MFA)
  - Passwordless MFA (doesn’t require password like - Microsoft Authenticator)
  - Phishing-resistant MFA (FIDO2 security key, Windows Hello for Business and Certificate-based Authentication Multi-Factor)
- Custom authentication strength can also be created.

KC3 - Authentication Strength - Phishing-resistant MFA

- One of the Phishing-resistant MFA methods is Certificate-based Authentication.
- We can use ‘Advanced options’ of Certificate-based authentication to allow following configurations (in decreasing order of security
  - Allowed certificate issuer (CA must be defined in tenant) AND policy OID (both required)
  - Allowed certificate issuer OR policy OID
  - Other certificate issuer by SubjectKeyIdentifier

    (no need to have the CA defined in tenant)



KC3 - Authentication Strength - Phishing-resistant MFA - Abuse

- To abuse Certificate-based authentication we need to steal an identity's certificate.
- A certificate can be found in
  - User/machine certificate stores or files (both standalone machines and Active Directory Certificate Services)
  - Key vaults
  - Storage Accounts, Automation Accounts, App Services and more

KC3 - Abusing Automation Accounts - Read Permission

- Automation accounts and its components like runbook and job output may have sensitive information like credentials, API Keys, connection strings, certificates etc.
- Even only with the ability to read them could be very fruitful.
- Some interesting permissions are:
  - Microsoft.Automation/automationAccounts/read
  - Microsoft.Automation/automationAccounts/jobs/read
  - Microsoft.Automation/automationAccounts/jobs/output/read
  - Microsoft.Automation/automationAccounts/runbooks/read
  - Microsoft.Automation/automationAccounts/runbooks/content/read

KC3 - Abusing Automation Accounts - Read Permission

- Following built-in roles have reader permissions on Automation Accounts. (Not listing the Privileged Administrator roles - Owner, Contributor, Role Based Access Control Administrator and User Access Administrator)

Role	Description
Automation Contributor	Full access but can’t assign roles.
Automation Operator	Can read job outputs.
Automation Job Operator	Can read job outputs.

KC3 - Learning Objective - 11

- Evade a CAP and access Oilcorp as the user ChristinaWBurrus
- Enumerate the Azure resources the user can access.
- Extract credentials from Job output of an automation account.

Part of - Kill Chain 3

Topics covered - Conditional Access Evasion, MFA bypass, Authenticated Enumeration and Data mining

KC3 - Token Extraction using AiTM

- Like any other application, Office 365 applications use JWTs to access resources in Azure.
- Using mitmdump tool, custom certificates and ability to modify proxy on a target machine, we can intercept the SSL/TLS traffic and extract the tokens.
- We will be interested in capturing tokens for Graph API as it also provides access to Office 365 services (OneDrive, Outlook/Exchange, Sharepoint and more).
- Note that traffic interception always needs to be done carefully!

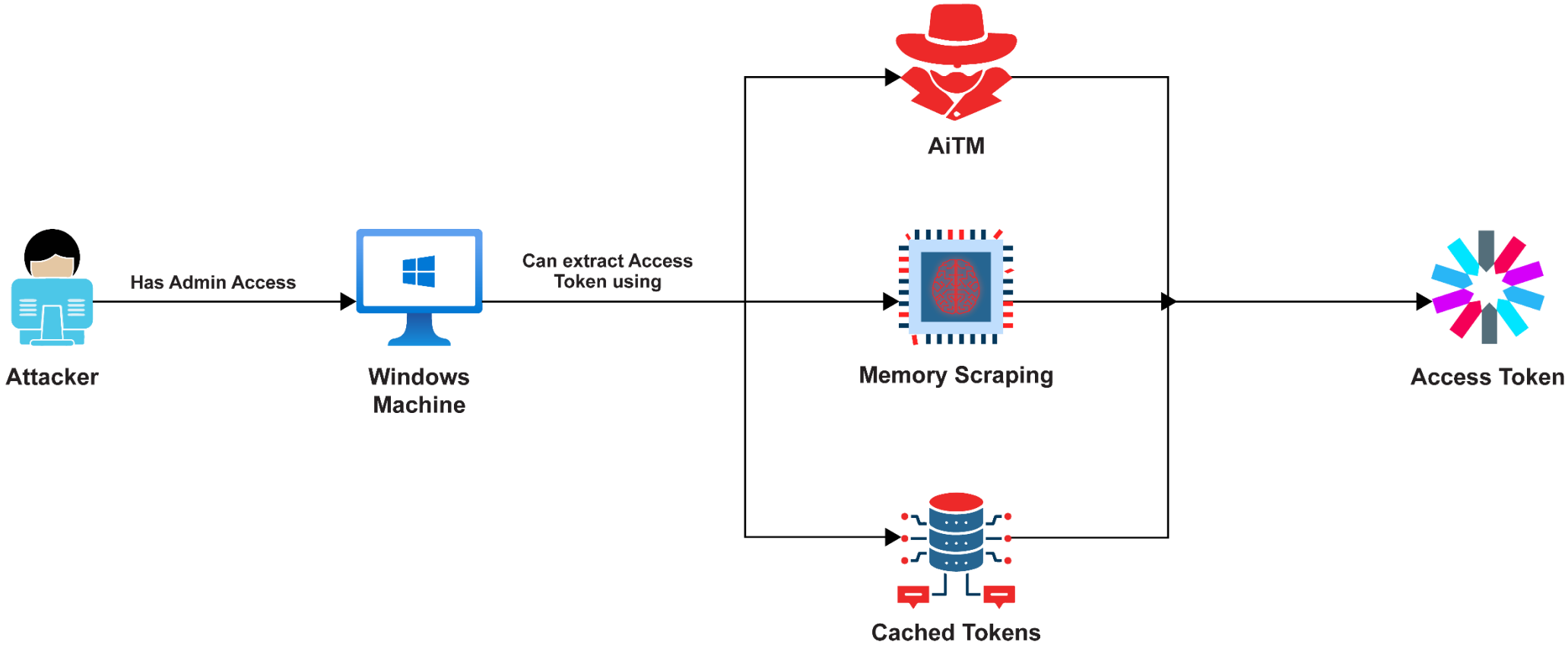
KC3 - Token Extraction using Memory Scrapping

- Another way to extract credentials is to dump memory of an Office 365 application and scrapping for tokens.
- Using Procdump, we can create a memory dump of an Office 365 application and then analyze it for useful tokens.

KC3 - Token Extraction by Decrypting Cached Tokens

- One more method would be to decrypt cached tokens.
- Tokens used by Office applications are cached in the '%LOCALAPPDATA%\Microsoft\TokenBroker\Cache' directory in .TBRES files protected using DPAPI.
- Using the WAMBam (https://github.com/xpn/WAMBam) tool, we can decrypt the files to retrieve tokens in clear-text.

KC3 - Token Extraction methods



KC3 - Learning Objective - 12

- Use the credentials extracted earlier to access an AWS VM using PowerShell Remoting.
- Extract access tokens from Word application on the VM.
- Use the access tokens to extract credentials from OneDrive or Mailbox of the user Casey.

Part of - Kill Chain 3

Topics covered - Lateral Movement across cloud providers, Abuse user-data, Token stealing and Data Mining

KC3 - Hybrid Identity - Microsoft Entra Kerberos

- Microsoft made it possible to use Kerberos authentication in Azure.
- Why? - A lot of applications depend on Kerberos authentication and Entra Kerberos makes it easier to move them to the cloud.

- A specific use case of Azure Virtual Desktop explains it very well:
  - On virtual desktops (VD), user's roaming profiles are maintained on network shares to provide same experience from any VD.
  - Kerberos authentication allows Azure VD to load user profiles from network shares - Azure files. This removes the requirement of having network access to an on-prem DC.

KC3 - Hybrid Identity - Microsoft Entra Kerberos - Azure File Shares

- This essentially means that Entra Kerberos allows hybrid users to access Azure file shares over the internet using their on-prem credentials.
- However, there are perquisites:
  - Only one identity method can be used at a time from Entra Kerberos, Entra Domain Services or On-Prem AD.
  - Client machines must be either Entra joined or Hybrid joined.
  - Only hybrid users are supported.

KC3 - Hybrid Identity - Microsoft Entra Kerberos - Azure File Shares - Entra App

- An Entra app registration is auto-generated when configuring Entra Kerberos on a file share - [Storage Account] <storage-account-name>.file.core.windows.net
- The service principal for this app needs manual admin consent and must be excluded from any MFA.
- The service principal has no access to the storage account.
- "Storage Resource Provider" is the Owner of this app.

KC3 - Hybrid Identity - Microsoft Entra Kerberos - Azure File Shares - Entra App - Persistence

- The Entra App can be used for persistence.
  - With Global Admin, Application Administrator and Cloud App Administrator roles, we can add Client secret to service principal that represents the storage account.
  - If we also assign some interesting roles to the app, it can work as a nice backdoor.
  - Note that there would still be sing-in and other logs.
- Also, [Storage Account] <storage-account-name>.file.core.windows.net can be used as a name for any backdoor app registration.

KC3 - Hybrid Identity - Microsoft Entra Kerberos - Azure File Shares - Lateral Movement

- If we can compromise a hybrid user who has access to an Azure File Share we will get access to files or blobs in the file share.
- Cloud to on-prem Lateral Movement (Many prerequisites)
  1. Compromise a cloud-only user who can reset password of other users (GA, Authentication Admin, User Admin, Helpdesk Admin, Password Admin and Privileged Authentication Admin)
  2. Reset the password of a synced user who can access an Azure File Share.
  3. Access the file share from a Entra joined or Hybrid joined machine.

KC3 - Learning Objective - 13

- Use the credentials extracted earlier to check if fileadmin user is a cloud-only user or synced user and their permissions on any resources.
- Use the credentials extracted earlier to access an Entra joined machine.
- Access an Azure File Share as fileadmin user from the Entra joined machine.

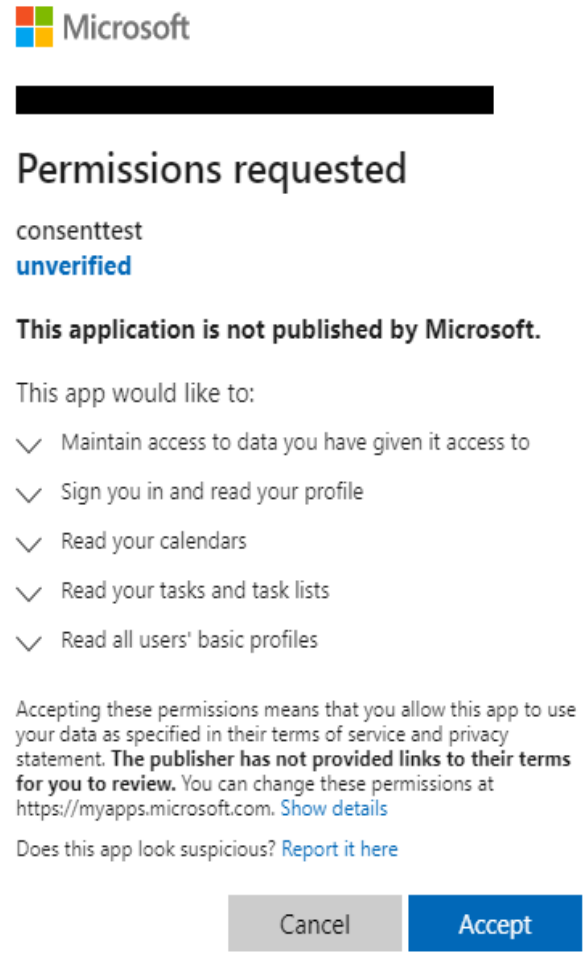
Part of - Kill Chain 3

Topics covered - Entra Kerberos, Azure File Share, Lateral Movement and Data mining

KC4 Starts

KC4 - Consent and Permissions

- Applications need to ask user for permissions to access their data. For example, for basic sign-in.
- Turned-on by-default, a normal user can grant consent only for "Low Impact" permissions. In all other cases, admin consent is required.
- Certain administrative roles can allow tenant-wide admin consent.



KC4 - Consent and Permissions - Low Impact

- Only the permissions that don't need admin consent can be classified as low impact.
- Permissions required for basic sign-in are openid, profile, email, User.Read and offline_access.
- That means, if an organization allows user consent for all apps, an employee can grant consent to an app to read the above from their profile.
- There are some very interesting low impact permissions. For example: User.ReadBasic.All that allows the app to read display name, first and last name, email address, open extensions and photo for all the users!

KC4 - Consent and Permissions - Admin Consent

- Most 'fun' permissions, understandably, need Admin consent.
- An admin can provide tenant-wide consent, that is, consent on behalf of organization.
- Global Administrator, Privileged Role Administrator, can provide tenant-wide admin consent for any permission for any API.
- Application Administrator and Cloud Application Administrator can provide tenant-wide consent except application permissions for Graph API.
- A custom role including 'permission to grant permissions to applications' can provide tenant-wide admin consent -
  microsoft.directory/servicePrincipals/managePermissionGrantsForAll.{id}
  {id} is ID of app consent policy

KC4 - Consent and Permissions - User Consent Settings

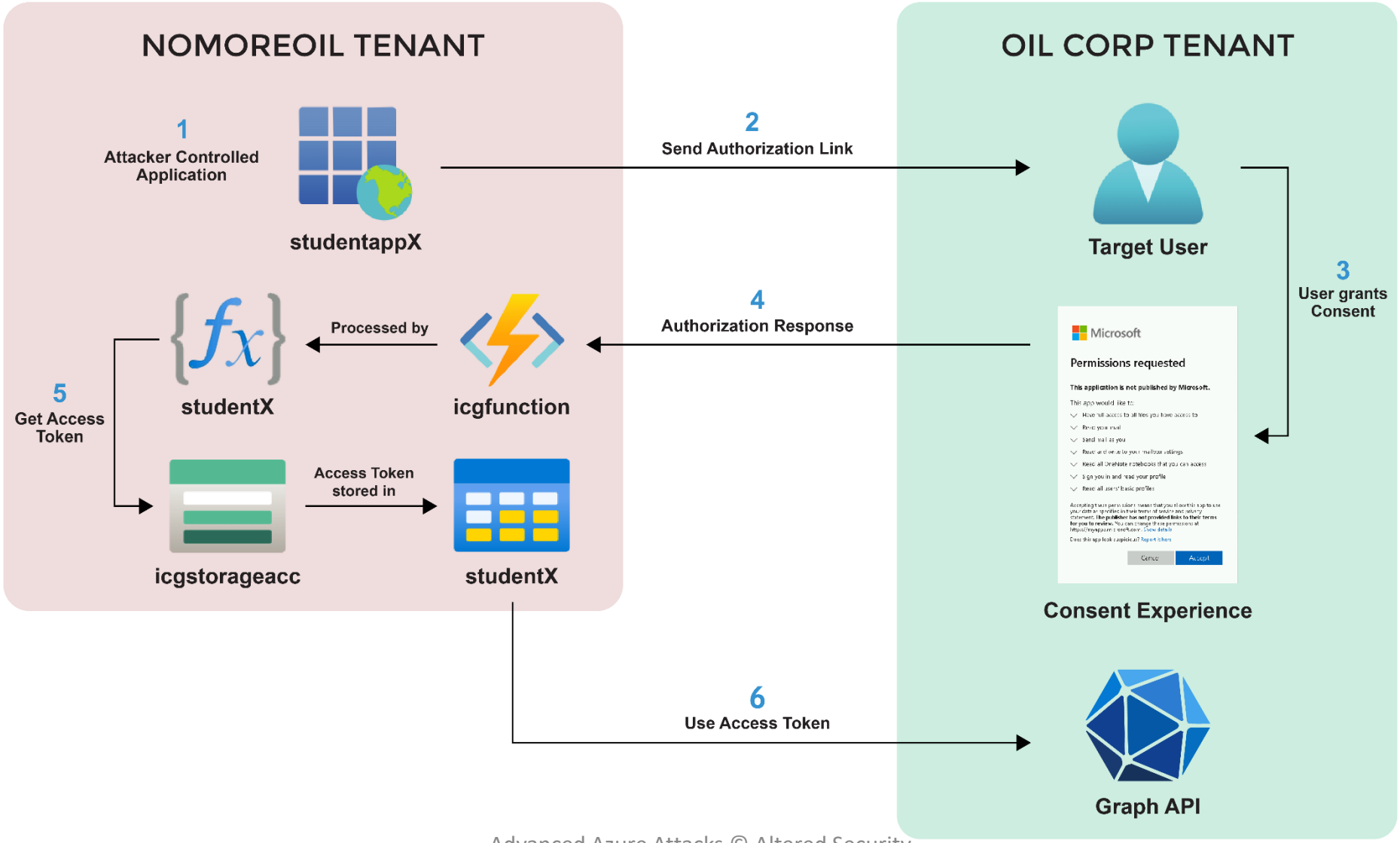
- Do not allow user consent
- Allow user consent for apps from verified publishers, for selected permissions - Only for "Low Impact" permissions for apps from same tenant and verified publisher
- Allow user consents for all apps - Allows consent for apps from other tenants and unverified publishers for Low Impact permissions **(Default)**
- Custom app consent policy

- 'Allow user consent for all apps' is what Oil Corporation is using. Note that this is the default setting (not Recommended).

KC4 - Initial Access - Illicit Consent Grant - Azure Infrastructure

- We need to setup the following in attacker tenant to be able to process responses from the target:
  - App registration - A multi-tenant application that the target needs to consent to.
  - Function App - To receive and process the Authorization response from a target. This response contains the access token.
  - Storage Account - To store the access tokens in a table.

KC4 - Initial Access - Illicit Consent Grant



KC4 - Initial Access - Illicit Consent Grant - Defense

- Configure the recommend User Consent setting - "Allow user consent for apps from verified publishers, for selected permissions" (Note that this doesn't stop consent for applications from the same tenant as the target)
- Disable Application registration in your tenant for normal users - Set "Users can register applications" to No.
- Note that a consent grant is logged in Entra ID activity logs and can be used for detection of illicit consent grant.

KC4 - Learning Objective - 14

- Find a target user from the OilCorp website https://explorationportal.z13.web.core.windows.net/
- Compromise the user using an Illicit Consent Grant attack.
- Access Team Chats of the user to enumerate more information from the target environment.

Part of - Kill Chain 4

Topics covered - Initial Access, Illicit Consent Grant and Teams Chat

KC4 - Basic Authentication and MFA

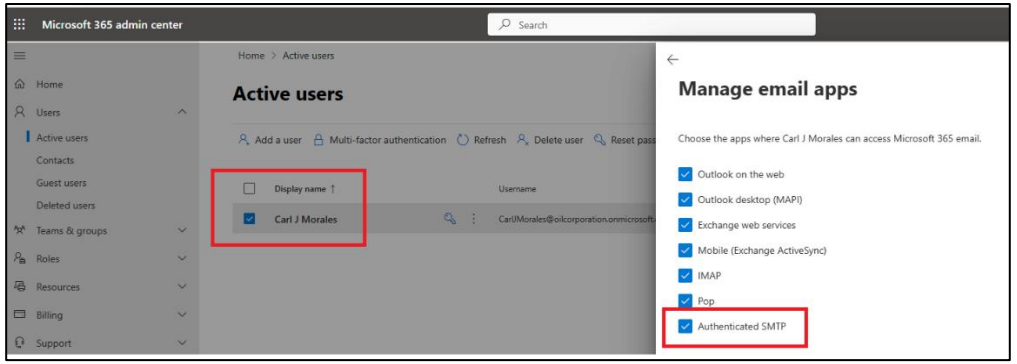
- For many years, Basic authentication (use of username and password) has been used to access web applications, services and more.
- In Office 365, Exchange Online used to be accessible using basic authentication.
- Basic authentication is considered risky as you cannot have MFA and Conditional Access on that.
- Microsoft disabled basic authentication (including Exchange Online) from 1st October 2022.

KC4 - SMTP and MFA Bypass

- SMTP is a legacy protocol used for sending emails (can't receive emails using this protocol).
- Since it is still widely used (printers, scanners etc.), it is possible to have SMTP AUTH (Basic authentication) in Azure.
- Please note that SMTP Auth can be configured on per-user basis. That is, even if it is disabled at the tenant level, it can be turned on for a particular user.
- This means that if we compromise a user's credentials, it is possible to bypass MFA by sending emails as that user using SMTP.

KC4 - SMTP and MFA Bypass

- In the lab, SMTP Auth is enabled only for the user carljmorales.
- A CAP "MFAPolicyForCarlJMorales" enforces MFA for user Carl if they try to access any cloud app.
- However, SMTP can still be used for the user.
- Note that this bypasses all the CAPs because - legacy auth.



KC4 - AiTM Phishing - Evilginx 3.0

- Evilginx 3.0 (https://github.com/kgretzky/evilginx2) is an Attacker in The Middle phishing framework.
- To target a specific service, Evilginx uses phishlets. These are YAML files that specify conditions like hosts, filters, structure of authentication cookies and credentials.

KC4 - AiTM Phishing - Evilginx 3.0 - MFA Bypass

- The captured session cookies can be replayed for MFA Bypass!
- No special tools are required for cookie replay. A browser plugin that allows editing cookies (ESTSAUTHPERSISTENT) is enough.

- Recall our discussion on Authentication strengths? Every MFA other than Phishing-resistant MFA - FIDO2 keys, Windows Hello and Certificate-based authentication - can be bypassed.
- There is a user simulation in the lab, that can complete MFA (with software OATH token) as the user Adam Jelder of OilCorp.

KC4 - Learning Objective - 15

- Use the credentials of carljmorales@oilcorporation.onmicrosoft.com with SMTP to bypass MFA and send an email to adamjelder@oilcorporation.onmicrosoft.com
- Send a lure from Evilginx in the email to phish adamjelder and capture their session cookie.
- Use the session cookie to bypass MFA and access Azure portal as adamjelder.

Part of - Kill Chain 4

Topics covered - Basic Authentication, MFA Bypass, AiTM Phishing and Cookie Replay

KC4 - Partners and Solution Providers (CSPs)

- Organizations can work with Microsoft-certified solution providers or partners to buy and manage products and services.
- A solution provider could be a reseller, advisor, indirect reseller, OEM and more.
- A Global Admin or Billing Admin can look for partners (https://appsource.microsoft.com/marketplace/partner-dir) and establish a contact.

KC4 - Partners and Solution Providers

- Depending on their type, a CSP may have administrative access to your tenant.
- Note that even a normal B2B Collaboration guest can be used to manage Entra, Azure and Microsoft 365 resources (more dangerous).

Partner Type	Description	Roles
Advisor	Partners can reset passwords and handle support incidents.	Global Admin and Helpdesk Admin
Granular delegated administrator privileges (GDAP)	Successor to much abused DAP (Global Admin by-default). Use to manage Azure resources but limited access in the Microsoft 35 admin centre.	Any Entra role (including Global Admin but not by-default)
Partner	Used to manage services in Microsoft 365 admin centre.	Any admin role in Microsoft 365 admin centre (including Global Admin but not by-default)

KC4 - Partners and Solution Providers

- If a partner with high privileged roles is compromised, it can lead to compromise of all the tenants managed by them. Classic Supply Chain attack!
- This has been abused by multiple threat actors. Most noted one is Nobelium/APT29/Midnight Blizzard.

KC4 - Partners and Solution Providers (CSPs)

- There are two ways to manage CSPs:
  - Partner Centre - https://partner.microsoft.com/dashboard/home - Provides the Granular DAP and other access.
  - Azure Lighthouse - Provides more granular access to Azure resources.

KC4 - Azure Lighthouse

- Uses Azure delegated resource management which makes it possible to manage customer’s Azure resources without having to sign in to the customer’s tenant.
- Provides more gradual control as subscriptions and resource groups can be delegated to specific users and roles in service provider’s tenant.
- "Customers maintain control over who has access to their tenant, which resources they can access, and what actions can be taken."
- Can also be used by a multi-tenant organization.
- Not to be confused with Microsoft 365 Lighthouse which is for M365 services.

KC4 - Azure Lighthouse - On-Boarding

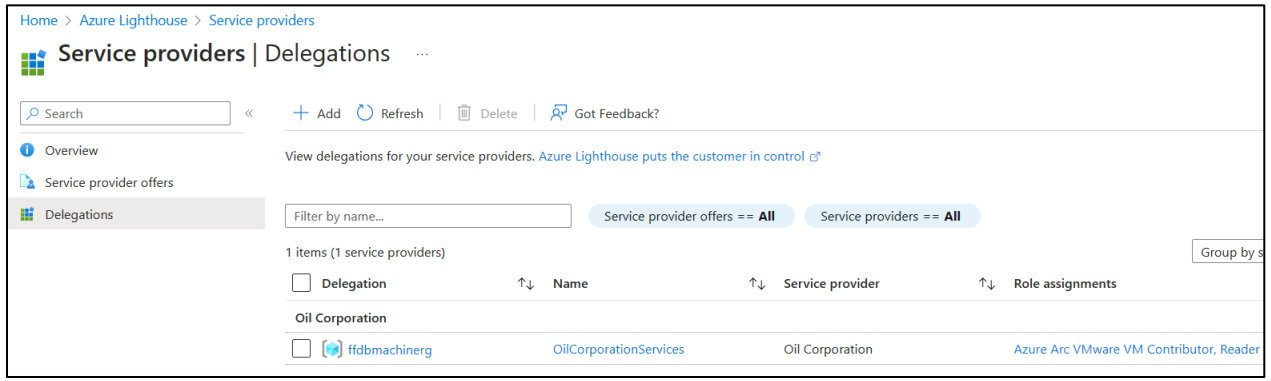
- A subscription or a resource group from a customer tenant needs to be onboarded to the service provider tenant.
- Only a subscription Owner or a user with "Microsoft.Authorization/roleAssignments/write" can onboard the subscription or any resource group in it.
- To see the delegated roles and customers, a user must be granted the Reader role during onboarding.

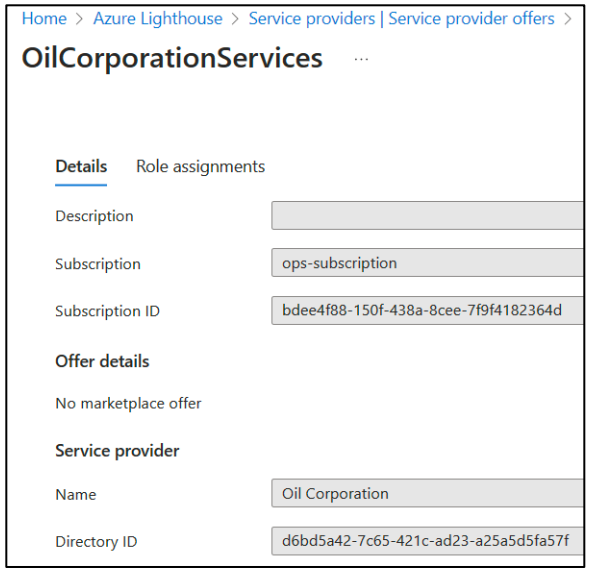
KC4 - Azure Lighthouse - Authorization

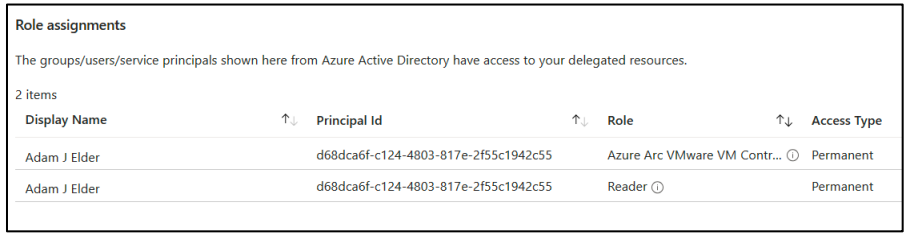
- Once onboarding is done, an authorization needs to be defined.
- Roles must be assigned to users, groups and service principals to the onboarded resources.
- Only the built-in Azure roles are supported.
- Following roles are not supported
  - Custom roles, Owner and Classic subscription administrator roles
  - User Access Administrator role is supported only to assign roles to Managed Identities in the customer tenant.
  - Any roles with DataActions
  - Any roles with */write, */delete or permissions that allow write or delete on Microsoft.Authorization
- Privileged Identity Management (PIM) is supported for activating role delegations.

KC4 - Azure Lighthouse - Authorization

- In the lab, Oil Corporation tenant is configured as a service provider for the Oil Corporation - Operations tenant.







KC4 - Azure Lighthouse - Abuse

- Overly permissive roles can be abused. For example, Contributor to an entire subscription.
- Roles with unintentional data access. Microsoft’s docs provides an example of the "Virtual Machine Contributor" role can read access keys of the storage account as the "Microsoft.Storage/storageAccounts/listKeys/action" action is included.
- As Azure Lighthouse is used for management, compromising a user with authorizations could result in compromise of resources they manage.

KC4 - Azure Lighthouse - Abuse

- In the lab, the user Adam Elder of OilCorp tenant has "Azure Arc VMWare Contributor" and Reader role on a resource group in the OilCorp Maintenance tenant.
- Compromising the user provides ability to install extensions on any VM in the target resource group - "Microsoft.HybridCompute/machines/extensions/write".
- Note the unintentional data access "Microsoft.Resources/deployments/read" will allow the user to read Deployment templates.
- See - https://azure.permissions.cloud/builtinroles/Azure%20Arc%20VMware%20VM%20Contributor

KC4 - Learning Objective - 16

- Use access token of the user adamjelder to connect to the Oil Corp tenant.
- Abuse the permissions that adamjhelder has on a resource group to execute commands on the FF-machine VM.
- Gather more information from FF-machine.

Part of - Kill Chain 4

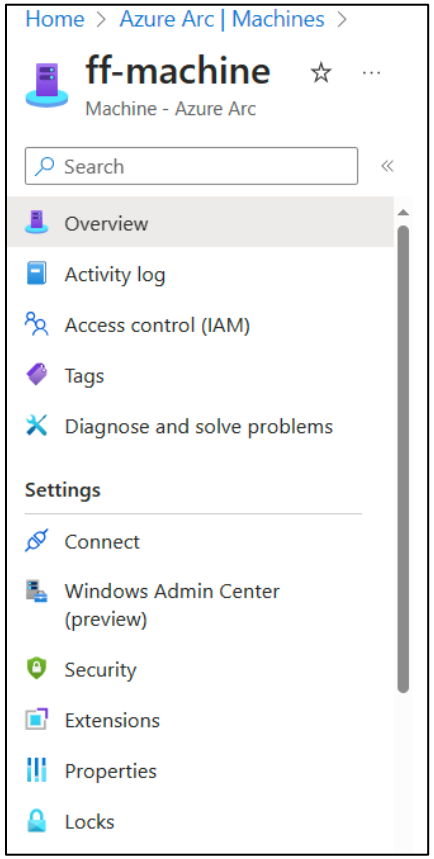
Topics covered - Azure Lighthouse, Privilege Escalation and Authenticated Enumeration

KC4 - Azure Arc

- Azure Arc allows to view and manage resources hosted outside Azure using Azure Resource Manager.
- Following major resources are supported
  - Servers and Application Services (App Services, Azure Functions etc.)
  - Kubernetes clusters
  - SQL Servers
  - VMs from VMWare, Hyper-V and Azure Stack HCI
- Management using resource manager means resources can be managed using API, Portal, Az PowerShell, Az CLI etc.

KC4 - Azure Arc-enabled Servers

- A Windows or Linux based server can be on-boarded to ARC using Connected Machine agent.
- An Arc-enabled server (aka hybrid machine) can:
  - Have VM extensions
  - Be on-boarded to Defender for Cloud
  - Be managed using ARM API and tools
- The agent uses three services:
  - Hybrid Instance Metadata Service (himds) - NT Service
  - Guest Configuration (GCService) and Guest Configuration Extension services - Local System



KC4 - Azure Arc-enabled Servers - Extensions

- VM extensions can be installed on Arc-enabled servers.
- Depending on the type, an extension on a Windows VM can be used to:
  - Collect logs and monitoring - Microsoft Monitoring Agent
  - Performance insights - Dependency Agent Windows
  - Certificate life-cycle management in a key vault - Key Vault for Windows
  - Post deployment configuration - Custom Script Extension
- Check the slide notes for a comprehensive list.
- Arc-enabled servers can have extensions allowlist and blocklist.

KC4 - Azure Arc-enabled Servers - Extensions

- Arc-enabled servers can have extensions allowlist and blocklist.
- Microsoft highlights that Custom Script extension could be blocked.
- Extensions like OpenSSH and Admin Center can still be abused to access an Arc-enabled server.

- Note that with ability to run commands on a server, it is possible to modify or remove allowlists and blocklists.

KC4 - Azure Arc-enabled Servers - Manged Identity

- When a server is on-boarded to Arc, a system-assigned managed identity is assigned to it.
- The details are added to Azure Instance Metadata Service (IMDS) on the server.
- Any process on the server can request an access token using the IDENTITY_ENDPOINT - http://localhost:40342/metadata/identity/oauth2/token

KC4 - Azure Arc-enabled Servers - Abuse

- Arc-enabled servers can be managed using ARM.
- This means that any Azure Role that provides ability to run commands or install extensions on the server can be abused.
- This can be abused to execute attacks against other on-prem resources.
- Ability to run commands on a server will allow an attacker to modify extensions allowlists and blocklists.
- Any Azure role assignments to the managed identity of an Arc-enabled server can be abused to attack other Azure resources.

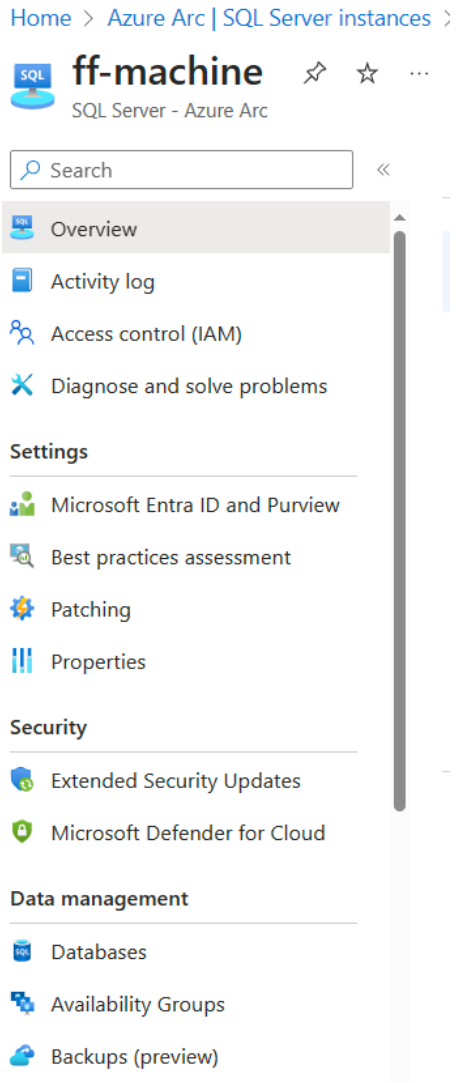
KC4 - Azure Arc-enabled Servers - Defense

- Avoiding overly permissive role assignment is the key! - to protect from run commands abuse, extensions and managed identity.
- Keep in mind that on-boarding a server to Arc **increases** the attack surface.
- For extensions, it is possible to create allowlists and blocklists that can be modified only locally from the server.
- It is also possible to disable the extension manager. However, it can be re-enabled locally.

KC4 - SQL Servers in Azure Arc

- SQL Servers can be on-boarded to Arc using the SQL Server extension.
- This allows SQL Server 2022 to use Entra ID for authentication.
- Like other Arc-enabled servers, Microsoft sells this as an easier way to manage your SQL Server inventory and easier on-boarding to Defender for Cloud.

- The extension creates a server role - SQLArcExtensionServerRole and a database role - SQLArcExtensionUserRole.
- Both are assigned SYSTEM at the database level.



KC4 - Azure SQL

- "Azure SQL is a family of managed, secure, and intelligent products that use the SQL Server database engine in the Azure cloud"
- There are three products in the Azure SQL family:
  - Azure SQL Database - Manged database service (Database-as-a-service).
  - Azure SQL Managed Instance - Instance as a service (supports Arc)
  - SQL Server on Azure VMs

KC4 - Azure SQL Database

- Compatible with most SQL Server features based on the latest stable Enterprise Edition of SQL Server.
- Microsoft manages patching, backups and availability of the database.
- Supports Single database or Elastic pool of multiple databases. Both can be managed using a Database logical server.
- Supports Entra ID authentication and can be assigned a managed identity.
- Can be on-boarded to Defender for Cloud.

KC4 - SQL Server Links

- Using SQL Server links or Linked servers, a SQL server instance can read data and execute commands against remote database servers and OLE DB data sources.
- Azure SQL Database supports SQL Server links. Links cannot be created to a logical server but only to a database.
- SQL authentication, managed identity and Entra authentication is supported.

KC4 - Learning Objective - 17

- Check for Linked servers for the SQL instance on Arc-enabled server FF-Machine
- Discover the entire Linked server chain to access an Azure SQL database.
- Extract sensitive information from the Azure SQL database.

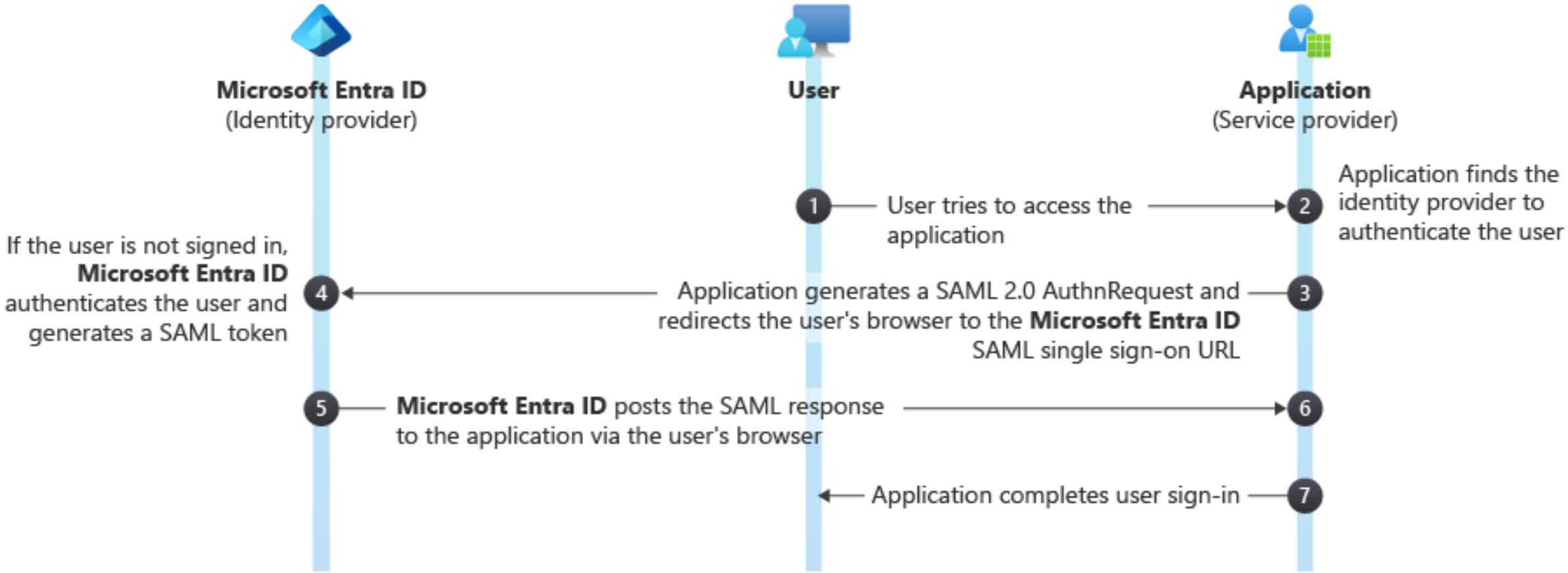
Part of - Kill Chain 4

Topics covered - Azure Arc, VM Extensions, Azure SQL, Cloud to On-prem Lateral movement, Linked Servers, On-Prem to Cloud Lateral Movement, Data Mining

KC4 - SAML SSO

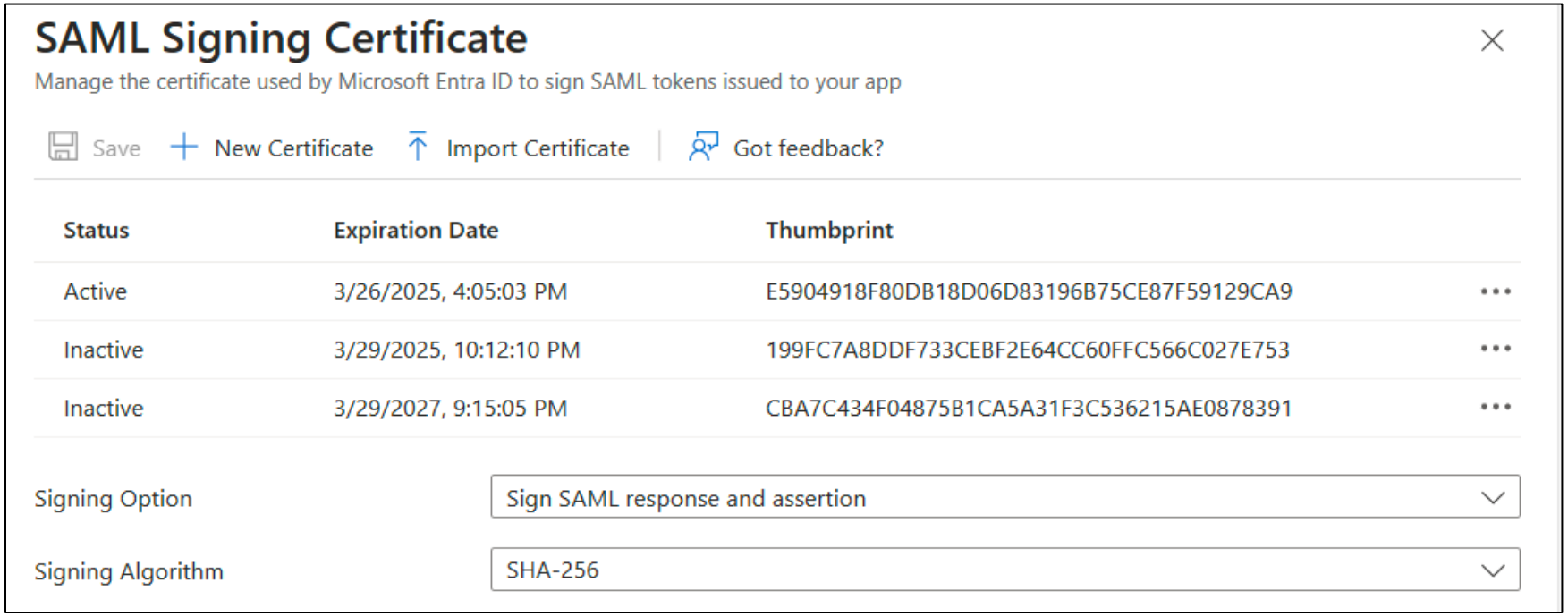
- Enterprises run on SSO - A single set of credentials to access applications!
- Entra ID provides for Single sign-on (SSO) for enterprise applications - SAML, Password-based and Linked.
- SAML SSO is the most popular method.
- A Global Administrator, Cloud Application Administrator, Application Administer or Owner of the application can configure SSO.

KC4 - SAML SSO in Entra ID



KC4 - SAML SSO in Entra ID

- When SAML SSO is configured, Entra ID provides a self-signed certificate for SAML response signing.
- It is possible to use an external self-signed certificate.



KC4 - Silver SAML

- If an attacker gets the external certificate that is configured to sign the SAML response, they can access the application as any user!
- That is, the attacker can forge the SAML response for the target user.
- This is both a privilege escalation and persistence technique.
- Depending on the application that is targeted, it can also be useful in lateral movement.

KC4 - Learning Objective - 18

- Access the MIRO application as the user Lyndia in Oil Corp - Operations and capture the SSO request.
- Execute the Silver SAML attack by abusing the SAML signing certificate to access the MIRO application as eatoceo user.

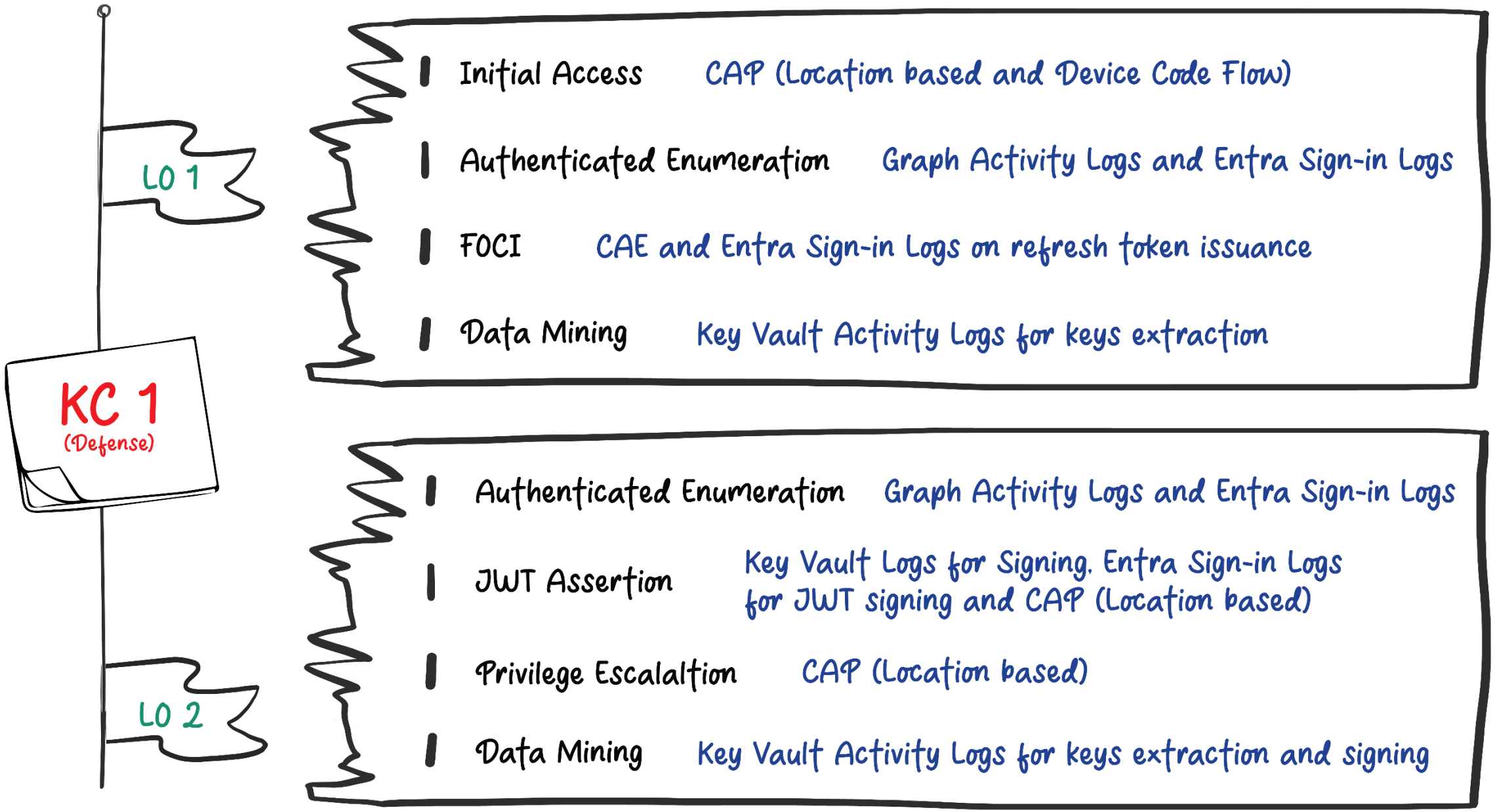
Part of - Kill Chain 4

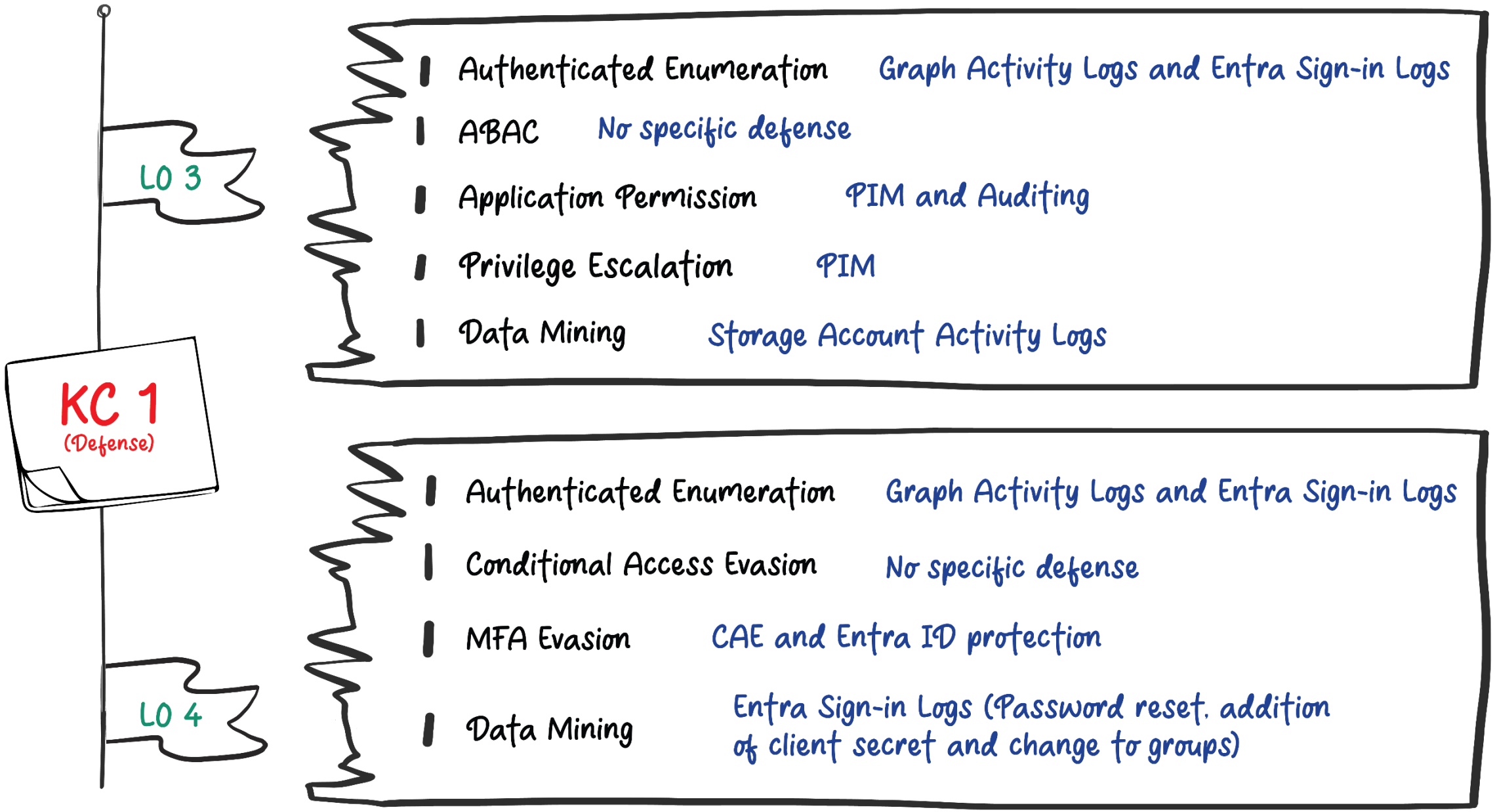
Topics covered - Authenticated Enumeration, SAML SSO and Silver SAML

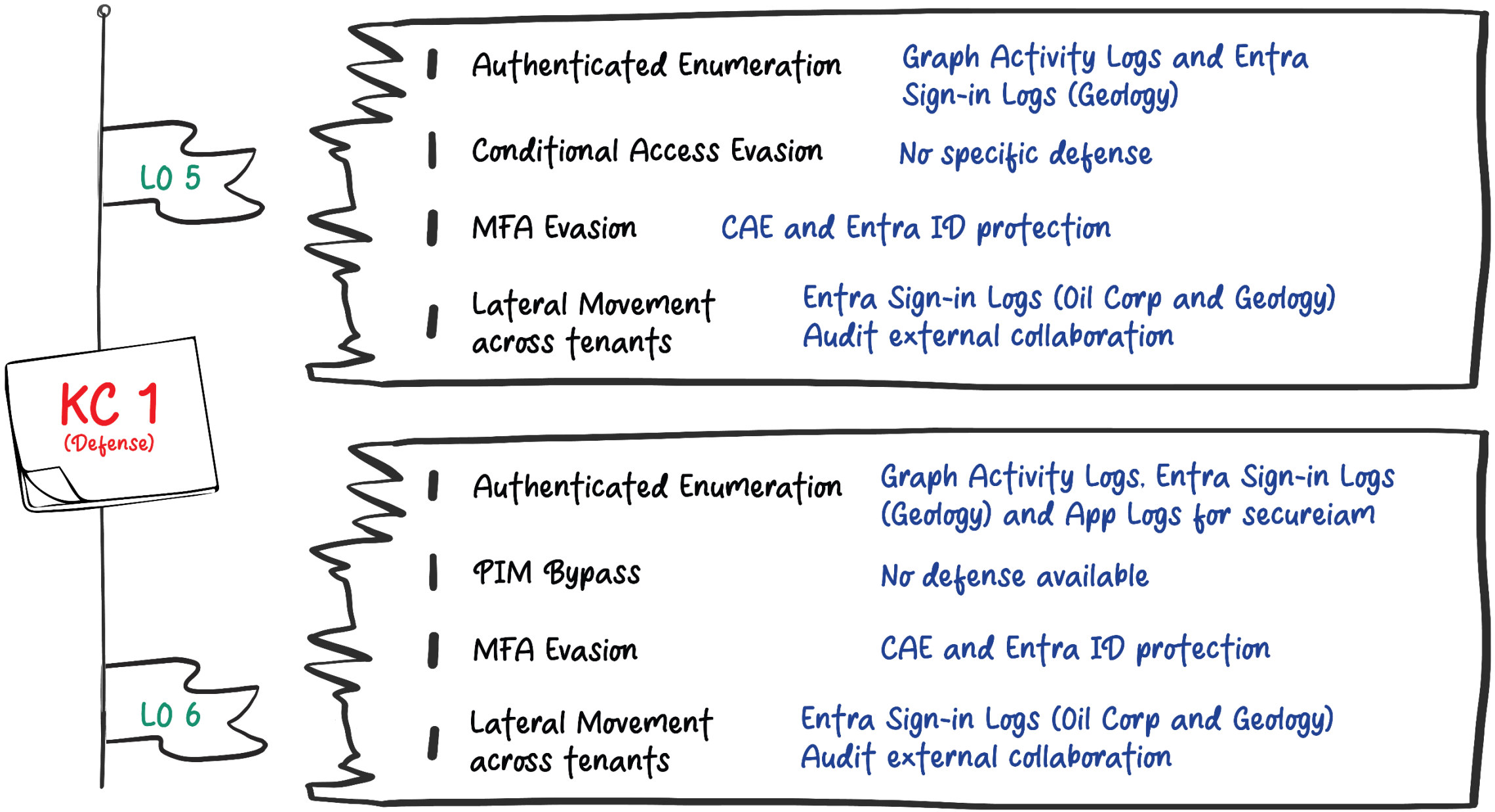
Defense Summary

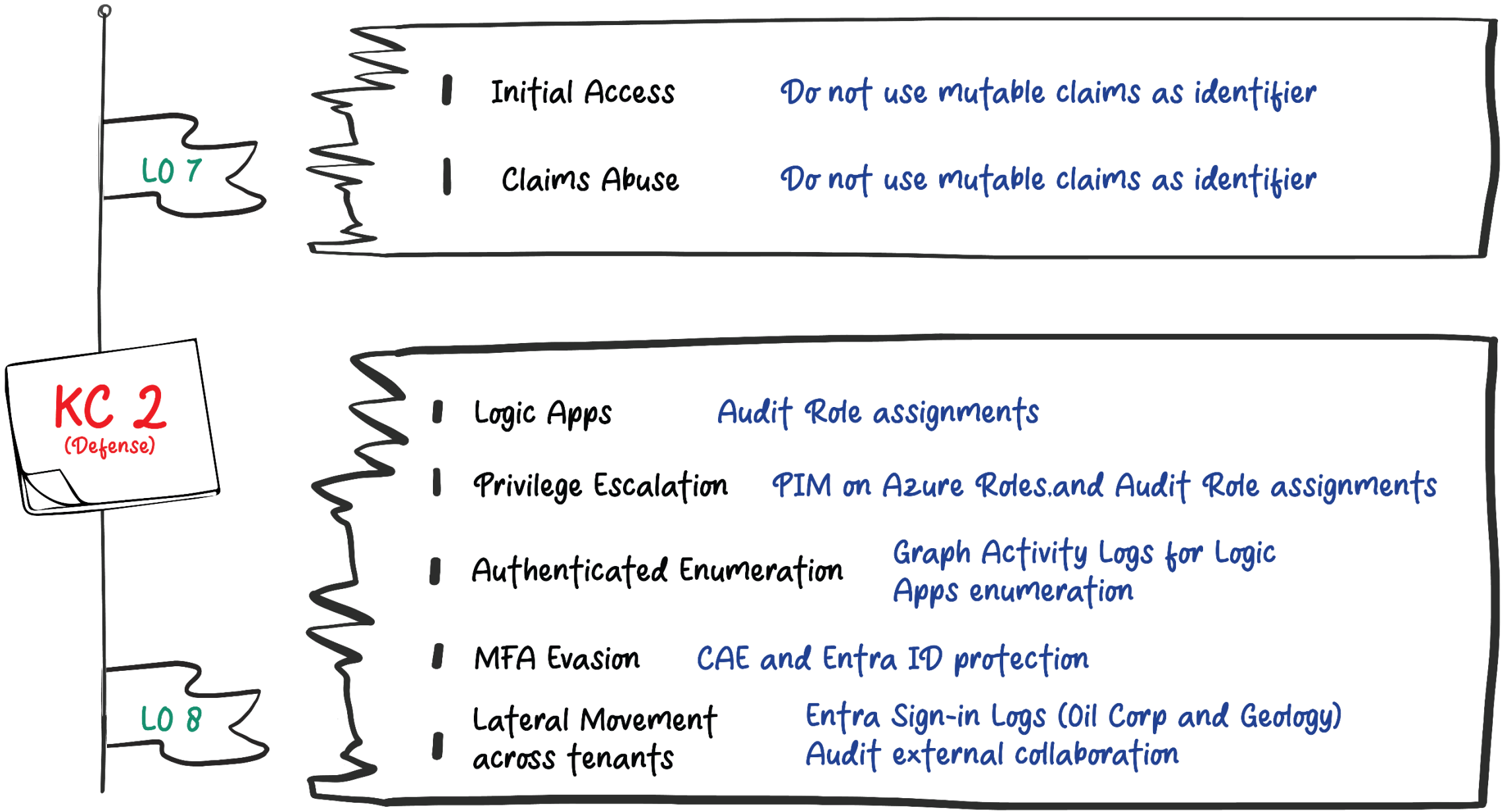
Defense Summary

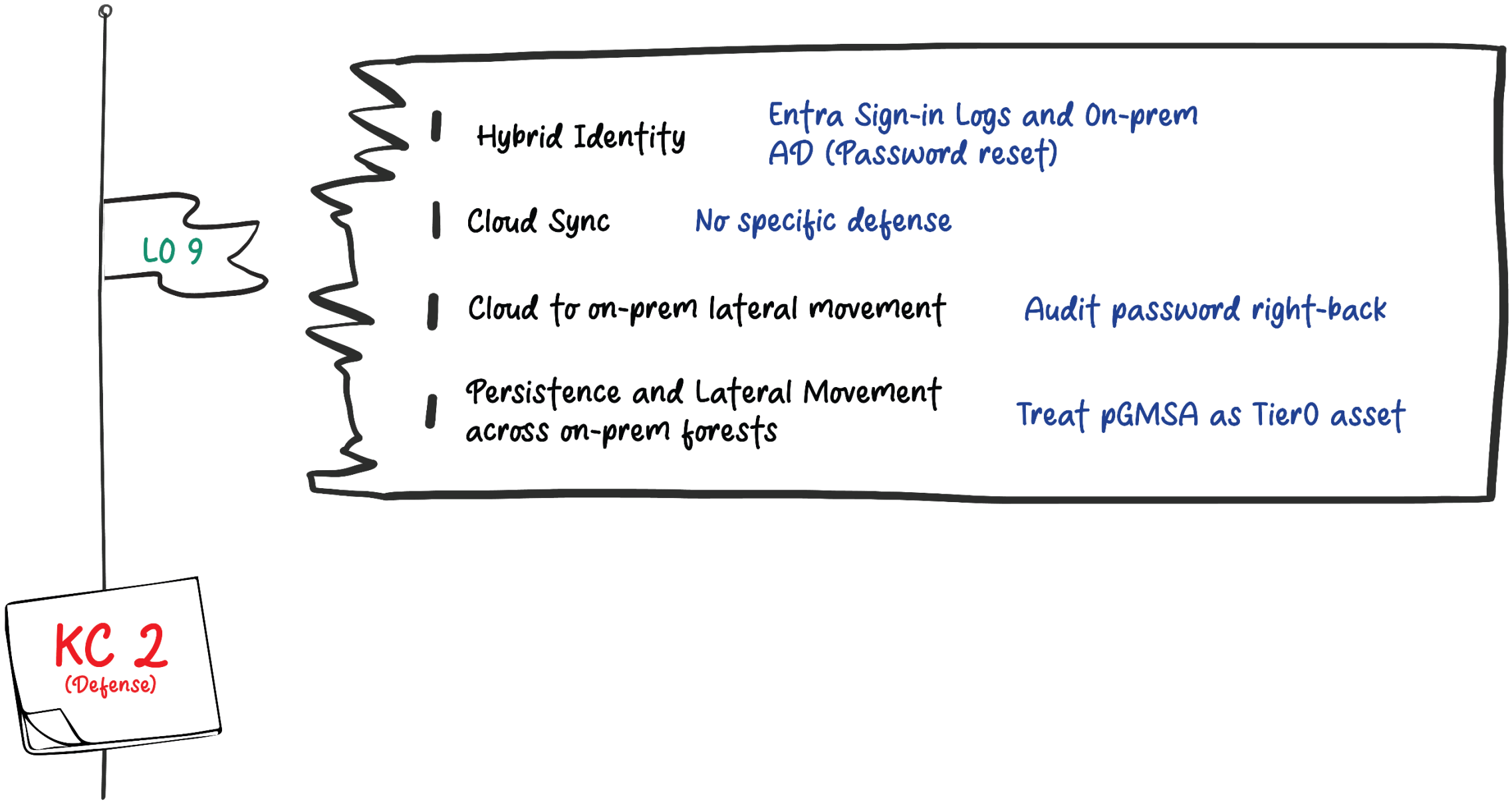
- Conditional Access Policies, Continuous Access Evaluation, Entra ID Protection and Network restrictions would be the most effective protection against the attacks executed in the lab.
- Please note that we deliberately focused on user identities in the class as it is comparatively easier to protect them. Workload identities are much harder to protect!
- We also have a CAP that mimics Security Defaults but with some exceptions for the sake of the lab.

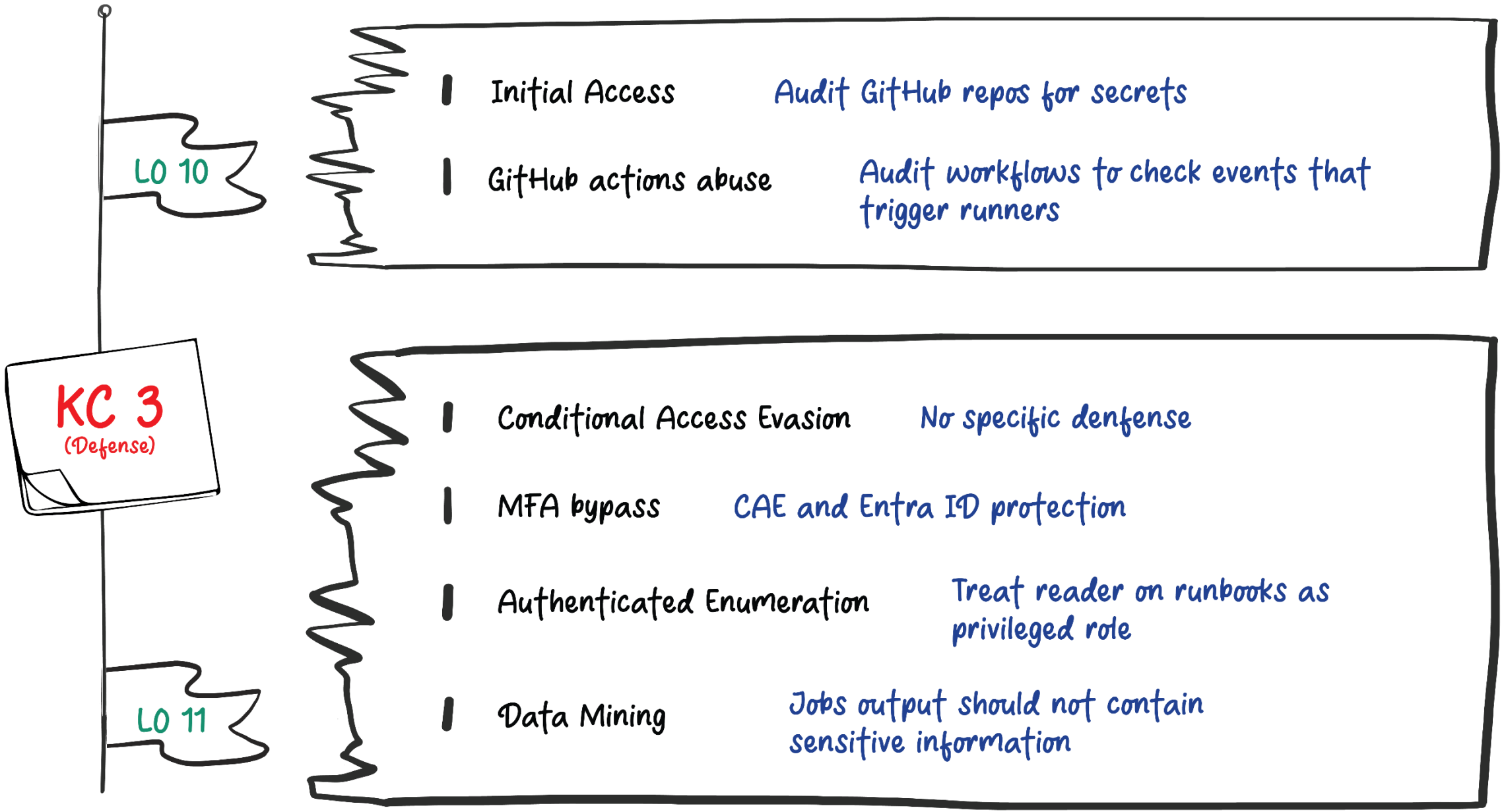

LO 1
KC 1
(Defense)
LO 2
Initial Access
CAP (Location based and Device Code Flow)
Authenticated Enumeration
Graph Activity Logs and Entra Sign-in Logs
FOCI
CAE and Entra Sign-in Logs on refresh token issuance
Data Mining
Key Vault Activity Logs for keys extraction
Authenticated Enumeration
Graph Activity Logs and Entra Sign-in Logs
JWT Assertion
Key Vault Logs for Signing, Entra Sign-in Logs for JWT signing and CAP (Location based)
Privilege Escalaltion
CAP (Location based)
Data Mining
Key Vault Activity Logs for keys extraction and signing

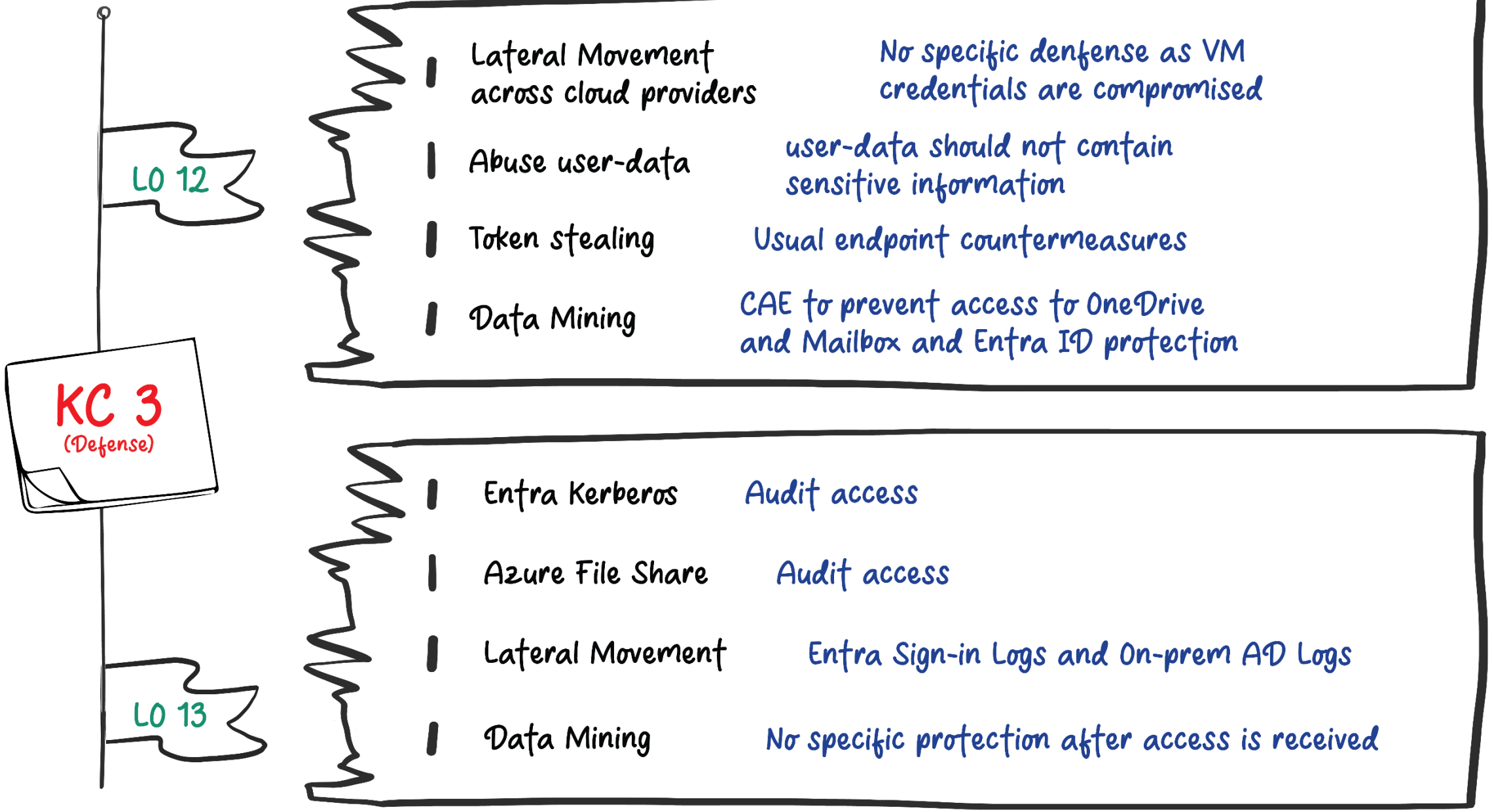

LO 3
KC 1
(Defense)
LO 4
Authenticated Enumeration
Graph Activity Logs and Entra Sign-in Logs
ABAC
No specific defense
Application Permission
PIM and Auditing
Privilege Escalation
PIM
Data Mining
Storage Account Activity Logs
Authenticated Enumeration
Graph Activity Logs and Entra Sign-in Logs
Conditional Access Evasion
No specific defense
MFA Evasion
CAE and Entra ID protection
Data Mining
Entra Sign-in Logs (Password reset, addition of client secret and change to groups)

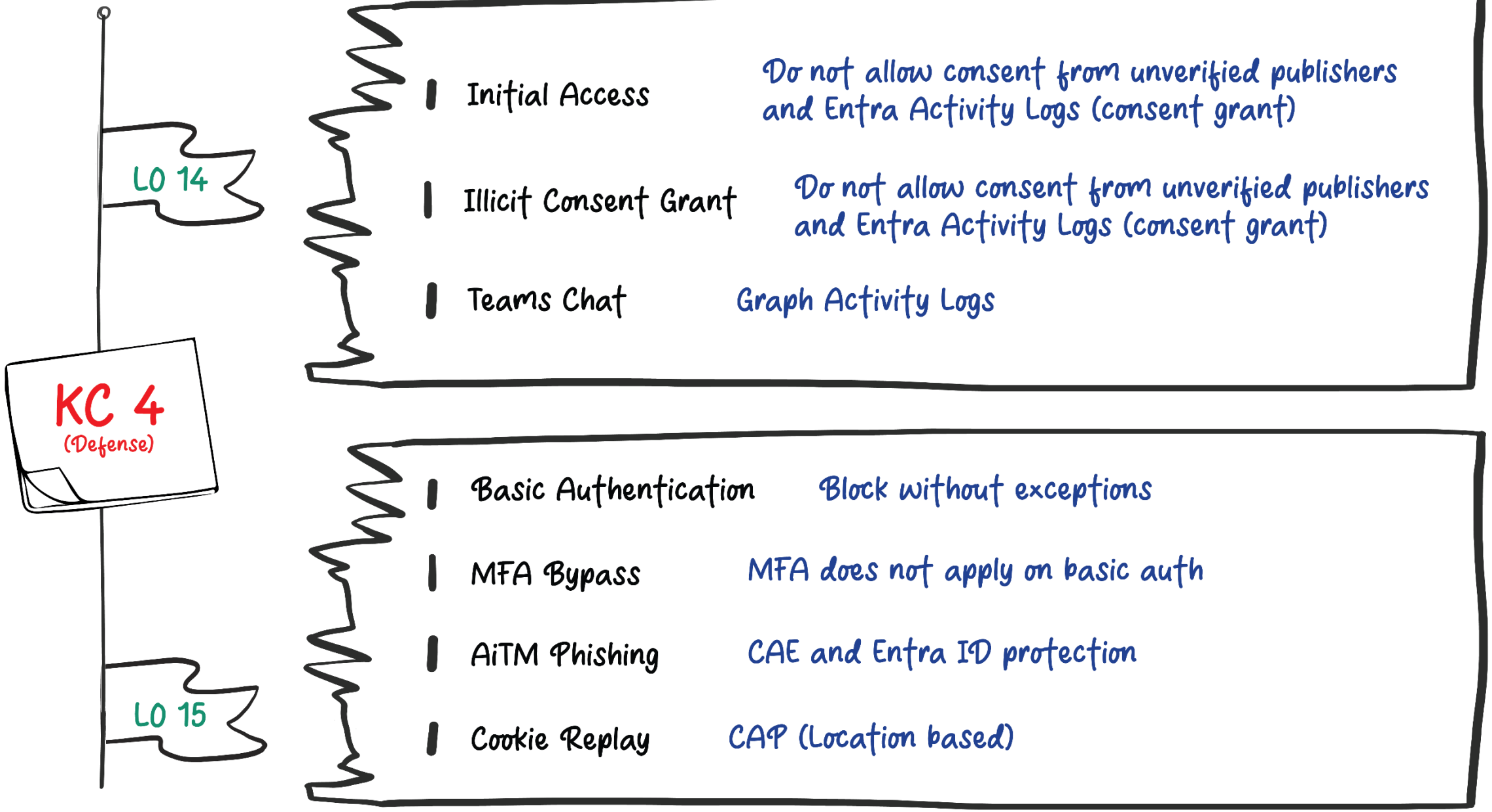

LO 5
KC 1
(Defense)
LO 6
Authenticated Enumeration
Graph Activity Logs and Entra Sign-in Logs (Geology)
Conditional Access Evasion
No specific defense
MFA Evasion
CAE and Entra ID protection
Lateral Movement across tenants
Entra Sign-in Logs (Oil Corp and Geology)
Audit external collaboration
Authenticated Enumeration
Graph Activity Logs, Entra Sign-in Logs (Geology) and App Logs for secureiam
PIM Bypass
No defense available
MFA Evasion
CAE and Entra ID protection
Lateral Movement across tenants
Entra Sign-in Logs (Oil Corp and Geology)
Audit external collaboration

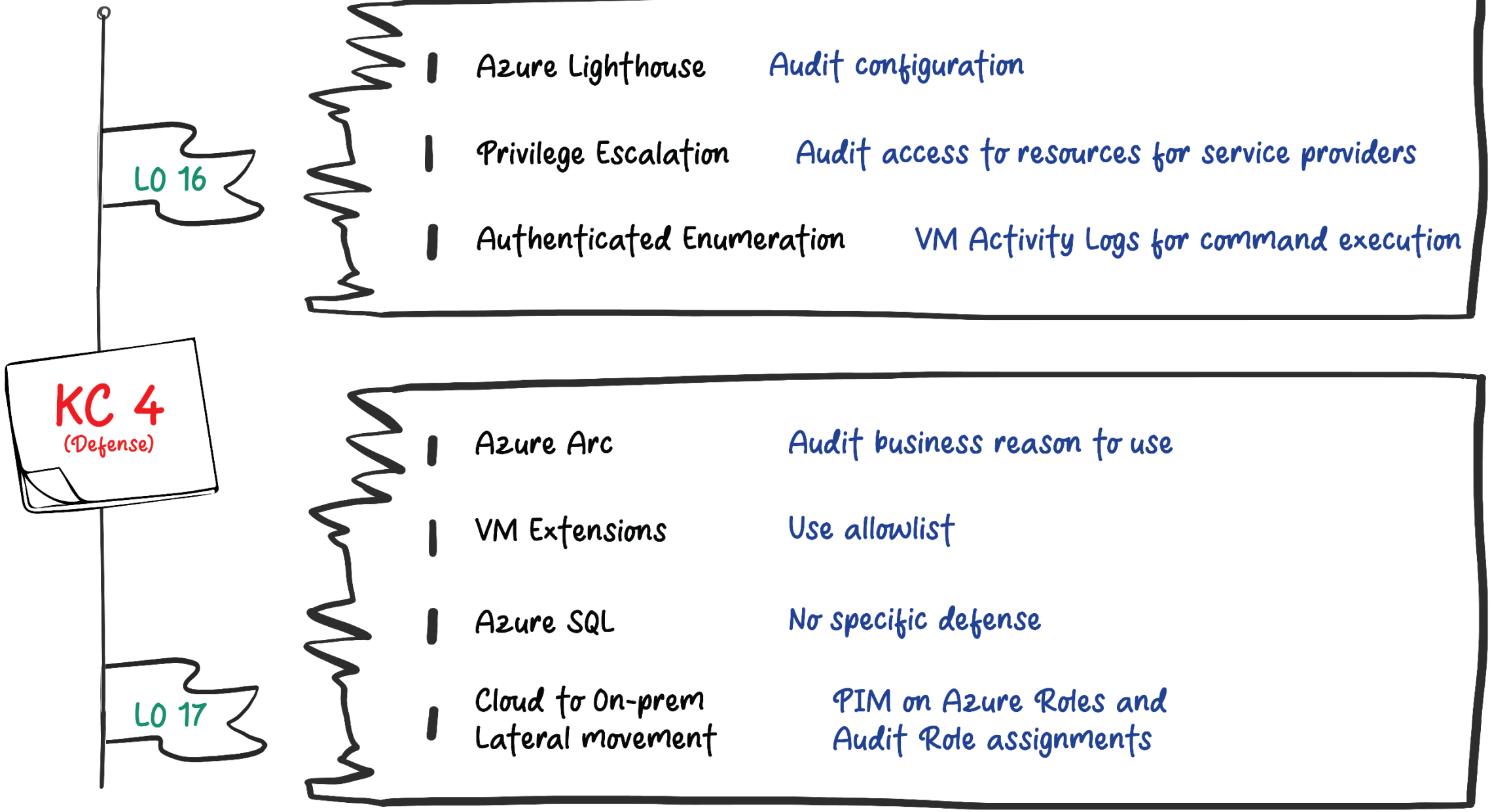

LO 7
Initial Access
Do not use mutable claims as identifier
Claims Abuse
Do not use mutable claims as identifier
KC 2
(Defense)
Logic Apps
Audit Role assignments
Privilege Escalation
PIM on Azure Roles.and Audit Role assignments
Authenticated Enumeration
Graph Activity Logs for Logic Apps enumeration
MFA Evasion
CAE and Entra ID protection
LO 8
Lateral Movement across tenants
Entra Sign-in Logs (Oil Corp and Geology)
Audit external collaboration

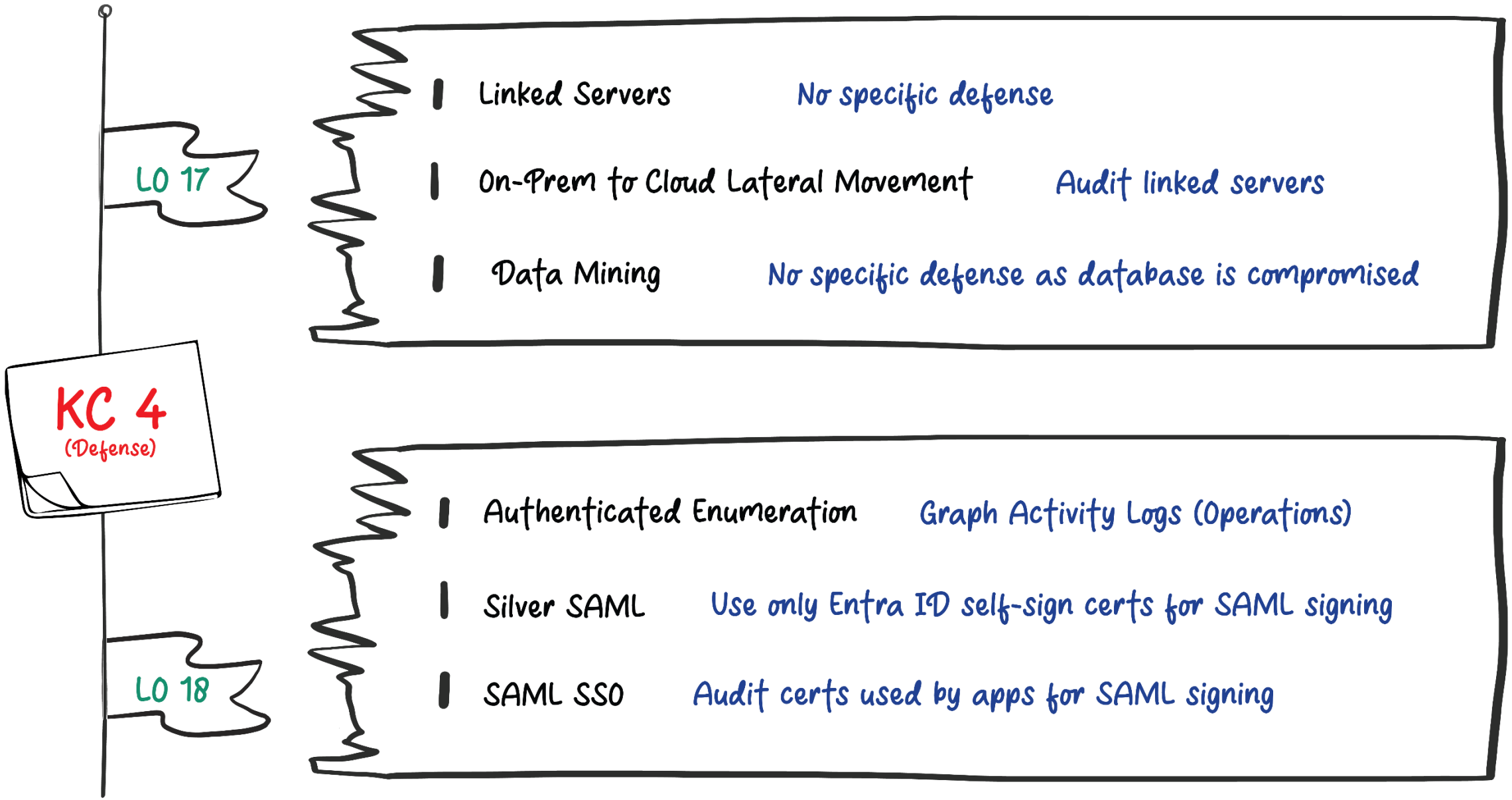

LO 9
Hybrid Identity
Entra Sign-in Logs and On-prem AD (Password reset)
Cloud Sync
No specific defense
Cloud to on-prem lateral movement
Audit password right-back
Persistence and Lateral Movement across on-prem forests
Treat pGMSA as Tier0 asset
KC 2
(Defense)


LO 10
Initial Access
Audit GitHub repos for secrets
GitHub actions abuse
Audit workflows to check events that trigger runners
KC 3
(Defense)
Conditional Access Evasion
No specific denfense
MFA bypass
CAE and Entra ID protection
Authenticated Enumeration
Treat reader on runbooks as privileged role
LO 11
Data Mining
Jobs output should not contain sensitive information


LO 12
KC 3
(Defense)
LO 13
Lateral Movement across cloud providers
No specific denfense as VM credentials are compromised
Abuse user-data
user-data should not contain sensitive information
Token stealing
Usual endpoint countermeasures
Data Mining
CAE to prevent access to OneDrive and Mailbox and Entra ID protection
Entra Kerberos
Audit access
Azure File Share
Audit access
Lateral Movement
Entra Sign-in Logs and On-prem AD Logs
Data Mining
No specific protection after access is received


LO 14
KC 4
(Defense)
LO 15
Initial Access
Do not allow consent from unverified publishers and Entra Activity Logs (consent grant)
Illicit Consent Grant
Do not allow consent from unverified publishers and Entra Activity Logs (consent grant)
Teams Chat
Graph Activity Logs
Basic Authentication
Block without exceptions
MFA Bypass
MFA does not apply on basic auth
AiTM Phishing
CAE and Entra ID protection
Cookie Replay
CAP (Location based)


LO 16
Azure Lighthouse
Audit configuration
Privilege Escalation
Audit access to resources for service providers
Authenticated Enumeration
VM Activity Logs for command execution
KC 4
(Defense)
Azure Arc
Audit business reason to use
VM Extensions
Use allowlist
Azure SQL
No specific defense
LO 17
Cloud to On-prem
Lateral movement
PIM on Azure Roles and
Audit Role assignments


LO 17
Linked Servers
No specific defense
On-Prem to Cloud Lateral Movement
Audit linked servers
Data Mining
No specific defense as database is compromised
KC 4
(Defense)
LO 18
Authenticated Enumeration
Graph Activity Logs (Operations)
Silver SAML
Use only Entra ID self-sign certs for SAML signing
SAML SSO
Audit certs used by apps for SAML signing

Thank you

- Please provide feedback.
- Follow me @nikhil_mitt
- nikhil@alteredsecurity.com
- For other red team labs: https://www.alteredsecurity.com/online-labs
- For bootcamps: https://www.alteredsecurity.com/bootcamps
- For lab extension/access/support, please contact : azadvanced@alteredsecurity.com
- Discord - https://discord.com/invite/vcEwaRMwJe