BLUEHAT

SEATTLE 2019

I’m in your cloud…

A YEAR OF HACKING AZURE AD

DIRK-JAN MOLLEMA / @_dirkjan

Whoami

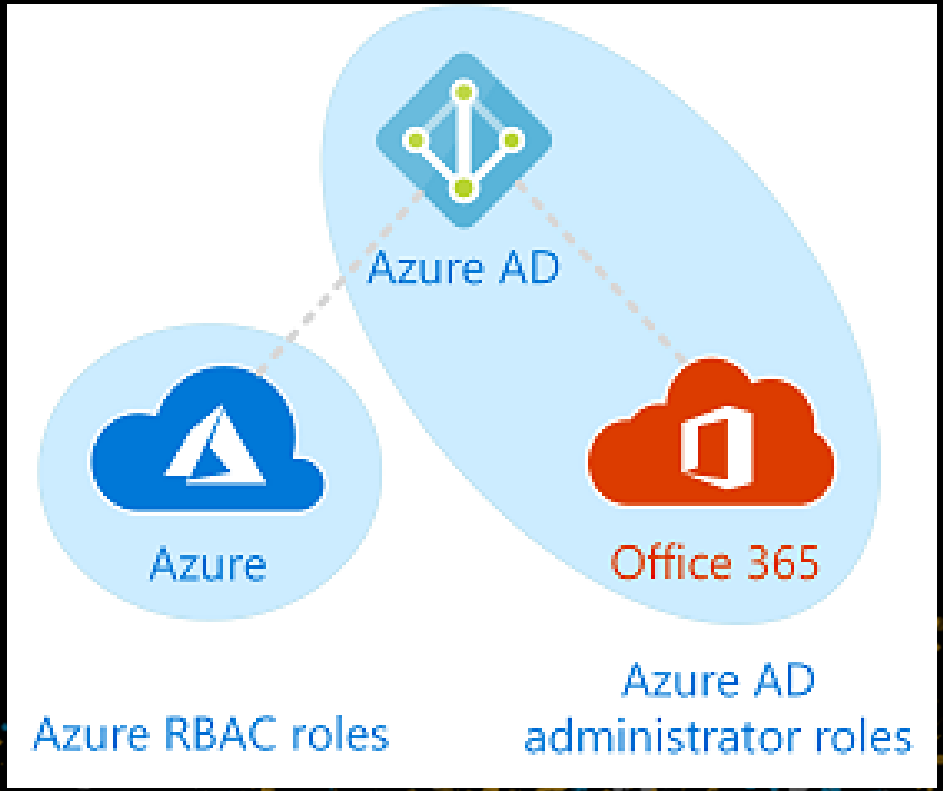
- Lives in The Netherlands
- Hacker / Red Teamer / Researcher @ Fox-IT since 2016
- Author of several Active Directory tools
  - Mitm6
  - ldapdomaindump
  - BloodHound.py
  - aclpwn.py
  - Co-author of ntlmrelayx
- One of the MSRC Most Valuable Security Researchers 2018/2019
- Blogs on dirkjanm.io
  - PrivExchange
- Tweets stuff on @_dirkjan

This talk

- Azure AD terminology – Portal vs API
- “Reversing” Azure AD via undocumented APIs
- Digging into service principals
- Linking up cloud and on-premise

Azure AD

- Not related to on-premise Active Directory
- Source of authentication for Office 365, Azure Resource Manager, and anything else you integrate with it.



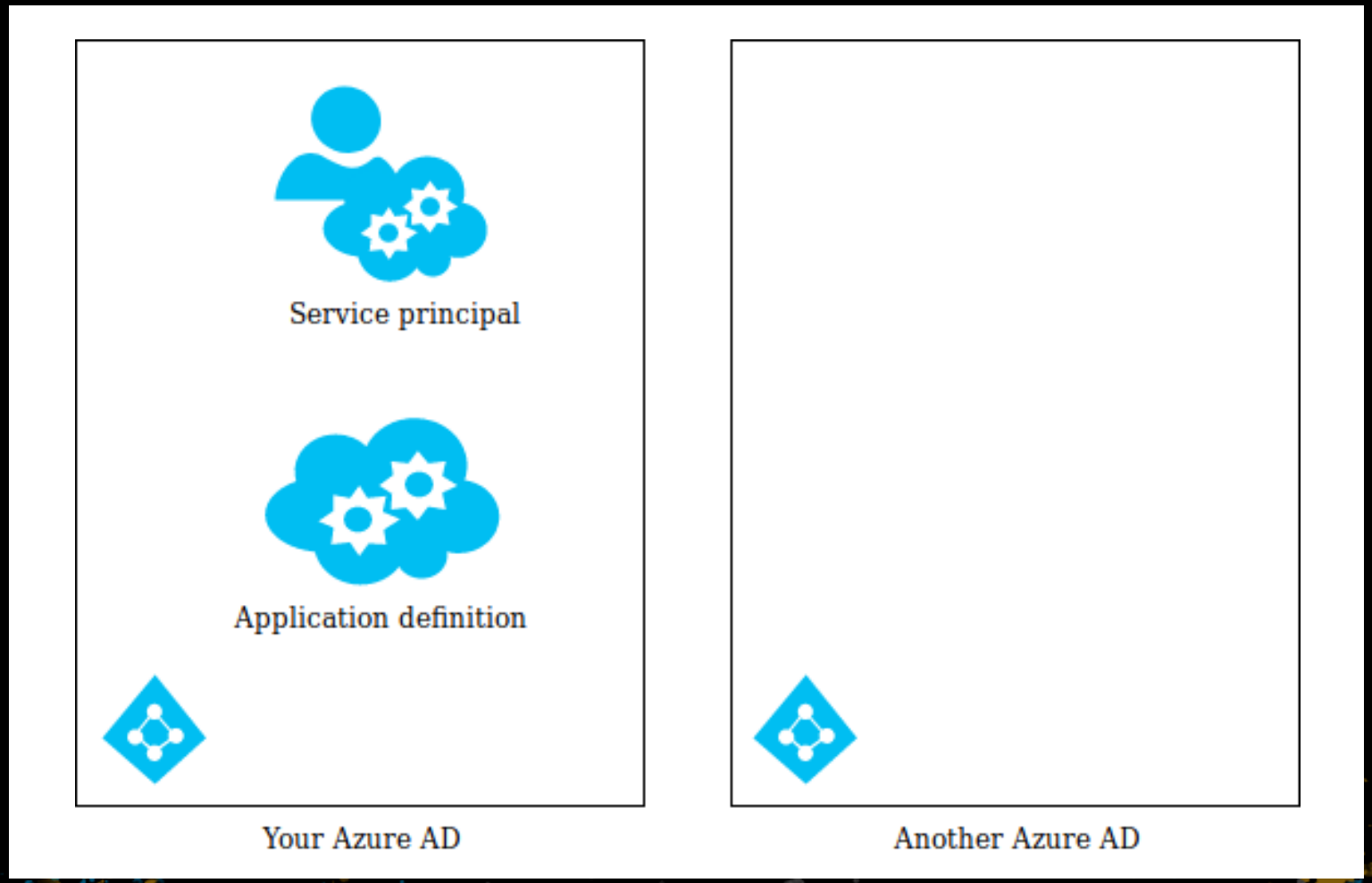
Azure AD Principals

- Users
- Devices
- Applications

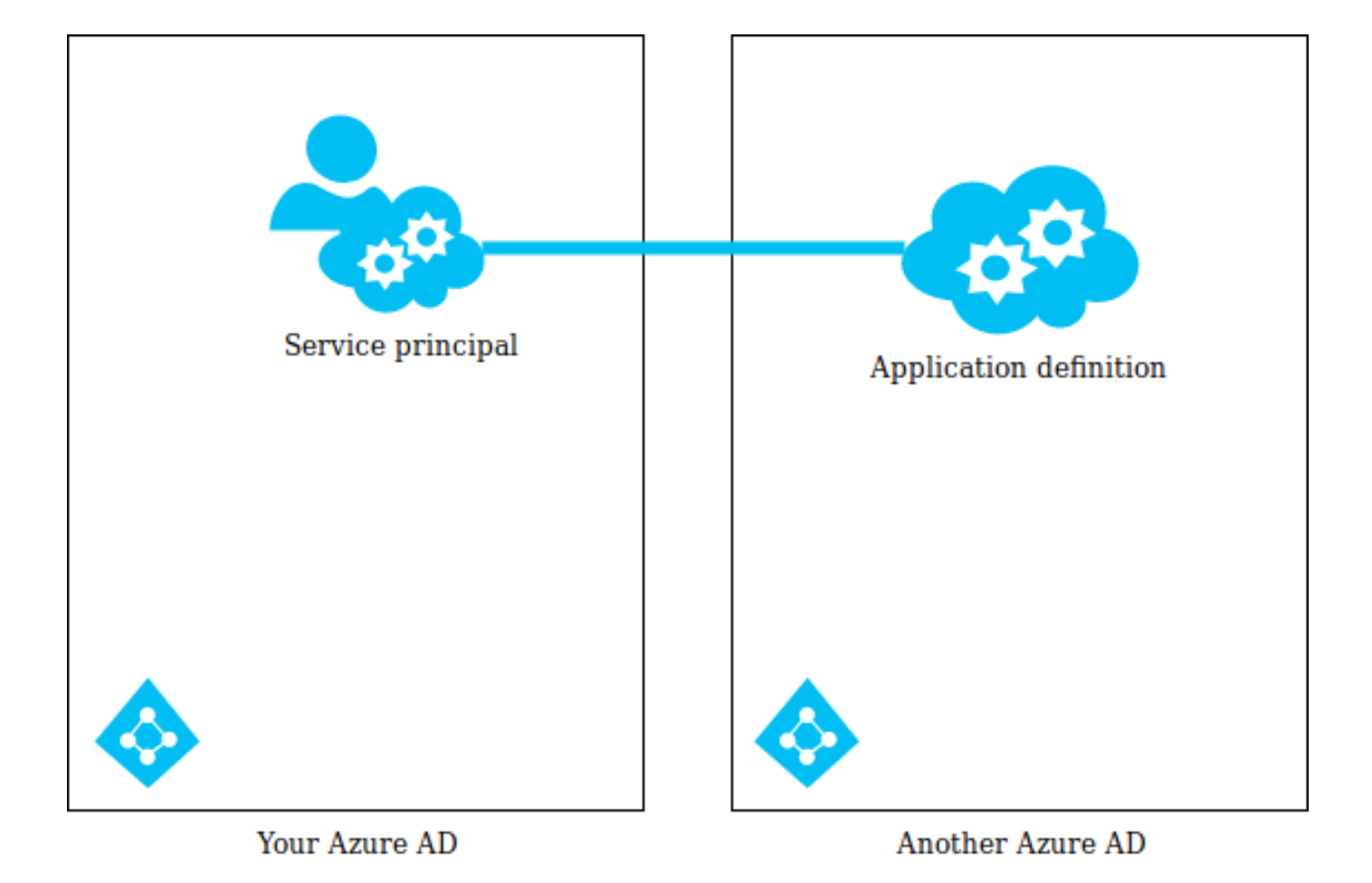
Everything is an application

- Examples:
  - Microsoft Graph
  - Azure Multi-Factor Auth Client
  - Azure Portal
  - Office 365 portal
  - Azure ATP
- A default Office 365 Azure AD has about 200 service principals (read: applications)

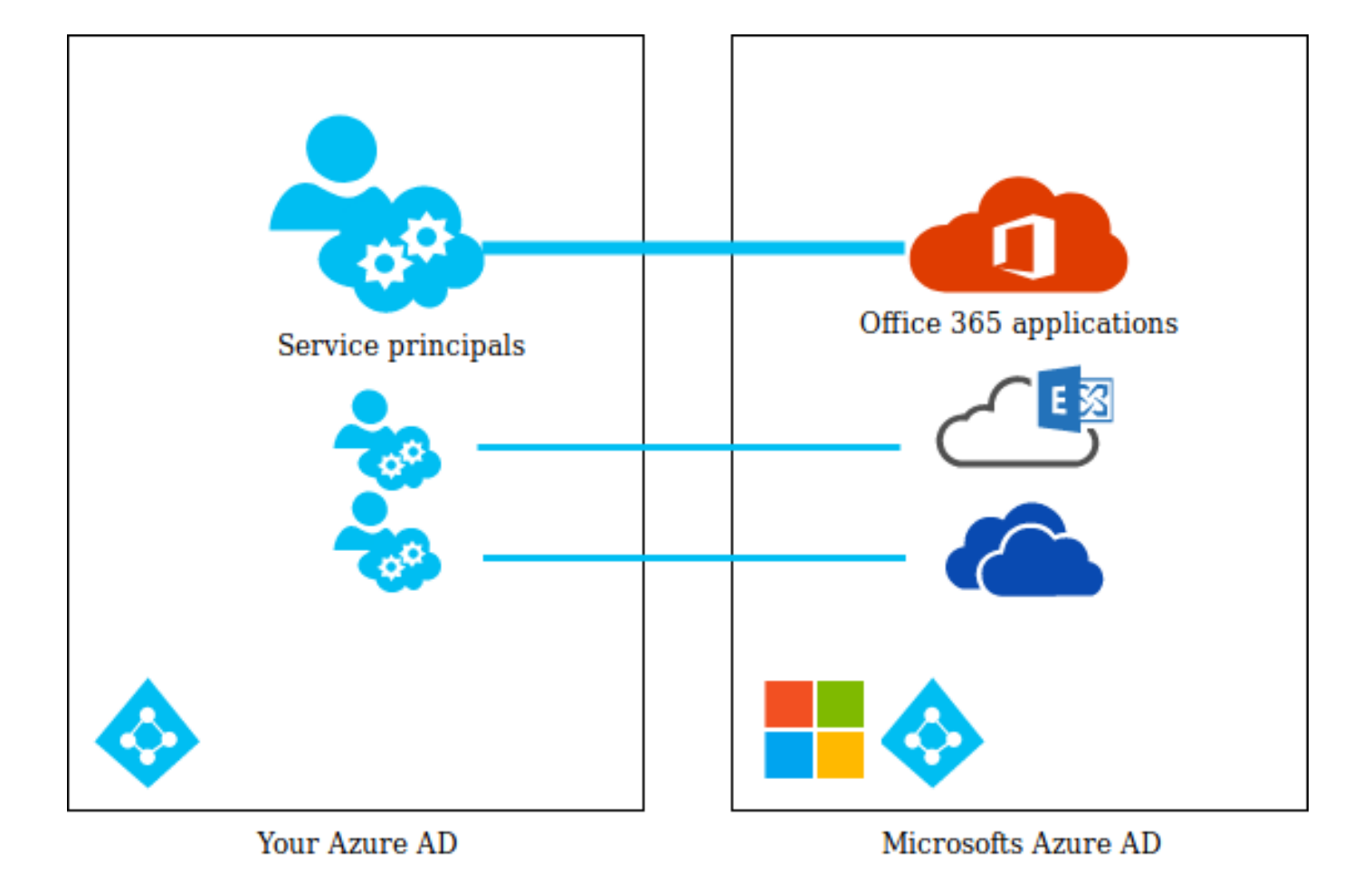
Applications and multitenancy – your apps



Applications and multitenancy – third party apps



Applications and multitenancy – Microsoft apps



Application privileges

- Two types of privileges:
  - Delegated permissions
    - Require signed-in user present to utilize
  - Application permissions
    - Are assigned to the application, which can use them at any time
- These privileges are assigned to the service principal

Permissions model

- Every application defines permissions
- Can be granted to Service Principals
- Commonly used:
  - Microsoft Graph permissions
  - Azure AD Graph permissions

Example: Application permissions

Home > MSOBB - App registrations > appadmintest - API permissions

appadmintest - API permissions

Search (Ctrl+/)

- Overview
- Quickstart

Manage

- Branding
- Authentication
- Certificates & secrets
- API permissions
- Expose an API
- Owners
- Manifest

API permissions

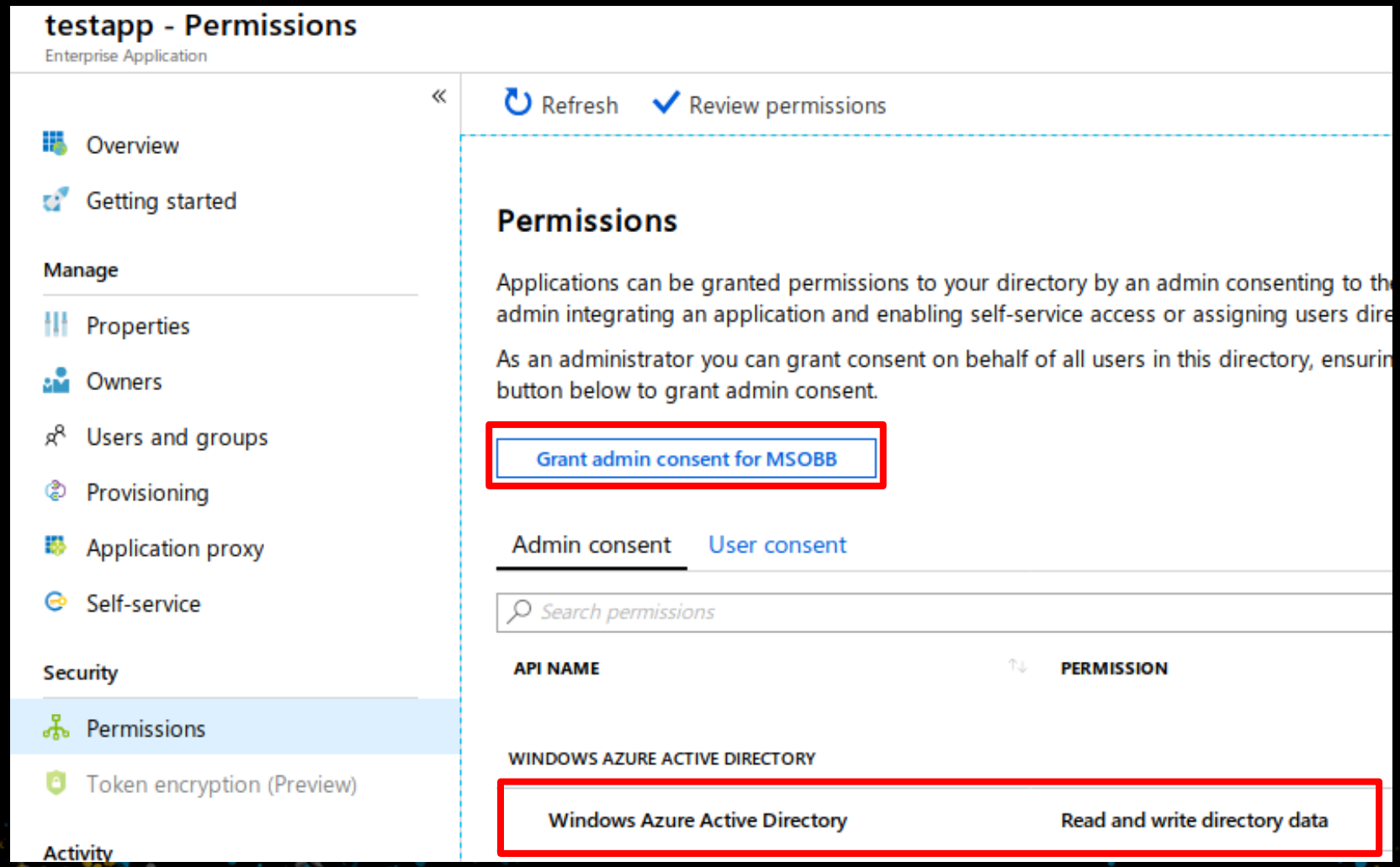
Applications are authorized to use APIs by requesting permissions. These permissions show up during the consent process where users are given the opportunity to grant/deny access.

+ Add a permission

API / PERMISSIONS NAME	TYPE	DESCRIPTION	ADMIN CONSENT REQUIRED
Microsoft Graph (3)			
Directory.AccessAsUser.All	Delegated	Access directory as the signed in user	Yes Granted for MSOBB
Directory.ReadWrite.All	Application	Read and write directory data	Yes Not granted for MSOBB
User.Read	Delegated	Sign in and read user profile	- Granted for MSOBB

These are the permissions that this application requests statically. You may also request user consent-able permissions dynamically through code. See best practices for requesting permissions

Service principal permissions



How permissions actually work

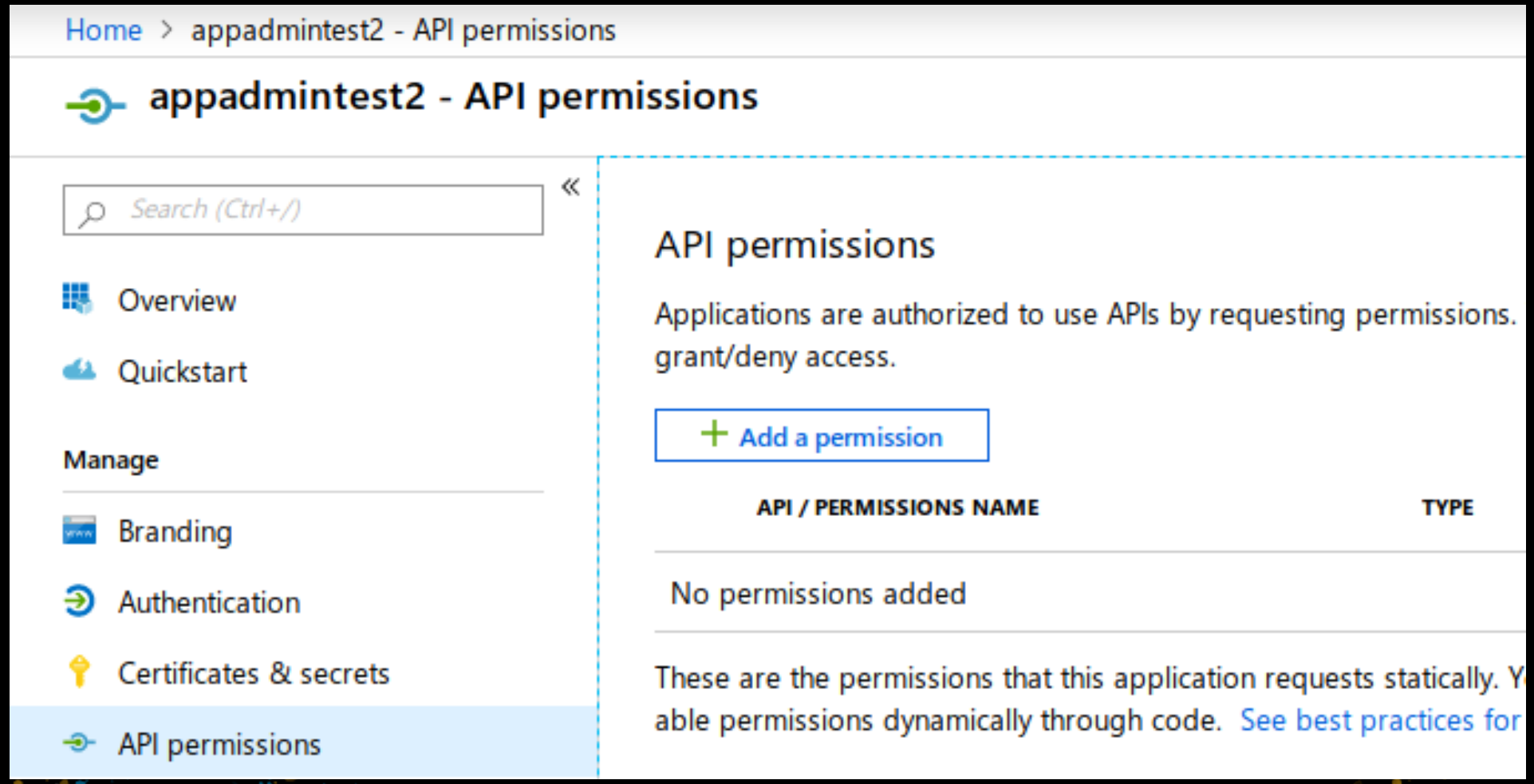
API definition	Portal terminology
Every application defines: - OAuth2 permissions - Application roles	App registration: - Delegated permissions - Application permissions
An application requires: - Resource access	App registration: - API permissions
A service principal has: - OAuth2 permission grants - Application roles	An enterprise application has: - Delegated permissions - Application permissions

Hiding in plain sight

- Normal flow:
  - Define required permissions in application
  - Approve permissions

- Alternative flow:
  - Assign a service principal to a role in MS Graph/AAD Graph directly

Application view



Service Principal view

Home > MSOBB > Enterprise applications - All applications > appadmintest2 - Permissions

appadmintest2 - Permissions

Enterprise Application

- Overview
- Getting started

Manage

- Properties
- Owners
- Users and groups
- Provisioning
- Application proxy
- Self-service

Security

- Conditional Access
- Permissions
- Token encryption (Preview)

Activity

Refresh | Review permissions

Permissions

Applications can be granted permissions to your directory by an admin consenting to the application for all users, a user consenting to the application for him or herself, or an admin integrating an application and enabling self-service access or assigning users directly to the application.

The ability to consent to this application is disabled as the app does not require consent. Granting consent only applies to applications requiring permissions to access your resources.

Grant admin consent for MSOBB

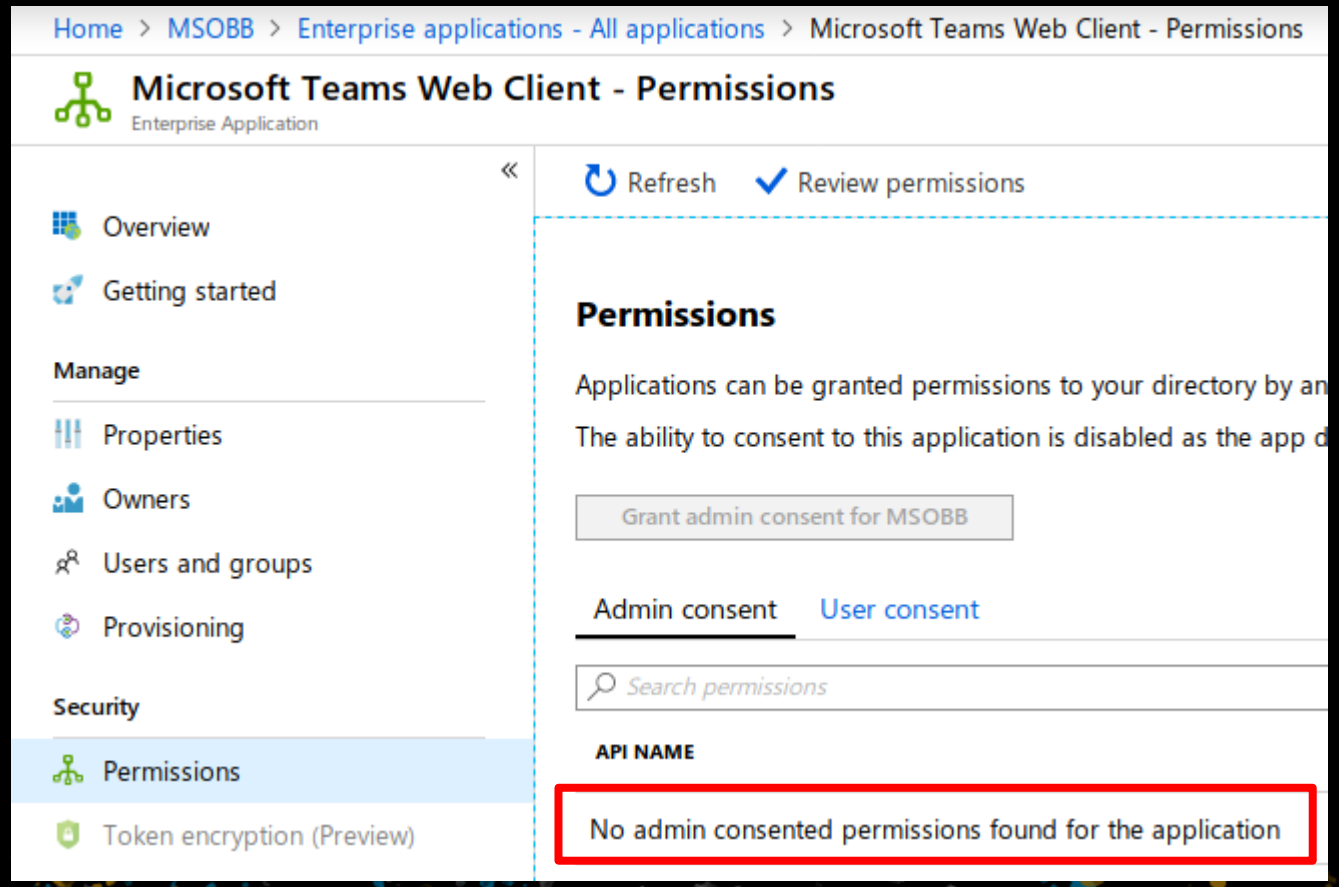
Admin consent | User consent

Search permissions

API NAME	PERMISSION	TYPE	GRANTED THR...	GRAN...
MICROSOFT GRAPH				
Microsoft Graph	Sign in and read user profile	Delegated	Admin consent	An admin
Microsoft Graph	Read and write directory data	Application	Admin consent	An admin

The exception: Microsoft applications…

- No way to tell from portal or API which permissions they have



JWT

PAYLOAD: DATA


```
{
  "aud": "https://outlook.office365.com",
  "iss": "https://sts.windows.net/50ad18e1-bb23-4466-9154-
bc92e7fe3fbb/",
  "iat": 1562755035,
  "nbf": 1562755035,
  "amr": [
    "pwd",
    "mfa"
  ],
  "app_displayname": "Microsoft Teams Web Client",
  "appid": "5e3ce6c0-2b1f-4285-8d4b-75ee78787346",
  "appidacr": "0",
  "enfpolids": [],
  "family_name": "Headinclouds",
  "given_name": "Eric",
  "ipaddr": "              ",
  "name": "Eric",
  "oid": "e0cd1b1c-d57a-4d31-a52b-50eee61836f3",
  "puid": "100320004A8144BA",
  "scp": "Calendars.ReadWrite Contacts.ReadWrite
EWS.AccessAsUser.All Mail.ReadWrite Mail.Send User.Read
User.ReadBasic.All",
  "sid": "1129f3ce-ee18-4295-ada0-b1004f6a36f9",
```

Why does this matter?

- Some admin roles allow managing all applications
  - Global Administrator
  - (Cloud) Application Administrator
- Including assigning credentials
- Possibility for backdooring Azure AD
  - No MFA for Service Principals
- Possible to escalate privileges
  - If you control an application with more privileges than you
- Default applications with more permissions than Application Administrator

Default app permissions

Application name	Access
Microsoft Forms	**Sites.ReadWrite.All**
Microsoft Forms	**Files.ReadWrite.All**
Microsoft Cloud App Security	**Sites.ReadWrite.All**
Microsoft Cloud App Security	**Sites.FullControl.All**
Microsoft Cloud App Security	**Files.ReadWrite.All**
Microsoft Cloud App Security	**Group.ReadWrite.All**
Microsoft Cloud App Security	**User.ReadWrite.All**
Microsoft Cloud App Security	**IdentityRiskyUser.ReadWrite.All**
Microsoft Teams	**Sites.ReadWrite.All**
Microsoft StaffHub	**Directory.ReadWrite.All**
Microsoft StaffHub	**Group.ReadWrite.All**
Microsoft.Azure.SyncFabric	**Group.ReadWrite.All**
Microsoft Teams Services	**Sites.ReadWrite.All**
Microsoft Teams Services	**Group.ReadWrite.All**
Office 365 Exchange Online	**Group.ReadWrite.All**
Microsoft Office 365 Portal	**User.ReadWrite.All**
Microsoft Office 365 Portal	**AuditLog.Read.All**
Azure AD Identity Governance Insights	**AuditLog.Read.All**
Kaizala Sync Service	**Group.ReadWrite.All**

https://dirkjanm.io/azure-ad-privilege-escalation-application-admin/

Logging?

- Log shows actions were performed by application

DATE	SERVICE	CATEGORY	ACTIVITY	STATUS	TARGET(S)	INITIATED BY (ACTOR)
3/13/2019, 9:53:56 PM	Core Directory	GroupManagement	Add member to group	Success	user@bbqmeatlovers.co...	testapp
3/13/2019, 9:53:40 PM	Core Directory	GroupManagement	Remove member from gr...	Success	user@bbqmeatlovers.co...	testapp
3/13/2019, 9:30:04 PM	Core Directory	GroupManagement	Add member to group	Success	user@bbqmeatlovers.co...	testapp

OAuth2 permissions – password grant

Application	API	Permissions
Microsoft.MileIQ	https://graph.windows.net/	**user_impersonation**
SharePoint Online Client Extensibility	https://graph.windows.net/	**user_impersonation**
Microsoft Teams - Device Admin Agent	https://graph.windows.net/	**user_impersonation**
Microsoft Stream Mobile Native	https://graph.windows.net/	**user_impersonation**
SharePoint Online Client	https://graph.windows.net/	**user_impersonation**
Outlook Online Add-in App	https://graph.windows.net/	**user_impersonation**
Microsoft.MileIQ	https://graph.microsoft.com/	**user_impersonation**
SharePoint Online Client Extensibility	https://graph.microsoft.com/	**user_impersonation**
Outlook Online Add-in App	https://graph.microsoft.com/	**user_impersonation**

Abusing password grant permissions

- OAuth2 password grant does not require verification
- Any APP ID can be used
- Interact with API’s with full user permissions
- Run AAD PowerShell without the PS App ID
  - Makes defender’s life harder

Details

Date: 2019-10-13 21:01:12

IP address:

User: will@willswindows.onmicrosoft.com

Activity: User logged in

Item: 00000003-0000-0000-c000-000000000000

Detail:

More information

Actor:


```
[
  {
    "ID": "639550db-b488-4664-a728-e05de4377461",
    "Type": 0
  },
  {
    "ID": "will@willswindows.onmicrosoft.com",
    "Type": 5
  },
  {
    "ID": "10032000728C3905",
    "Type": 3
  }
]
```

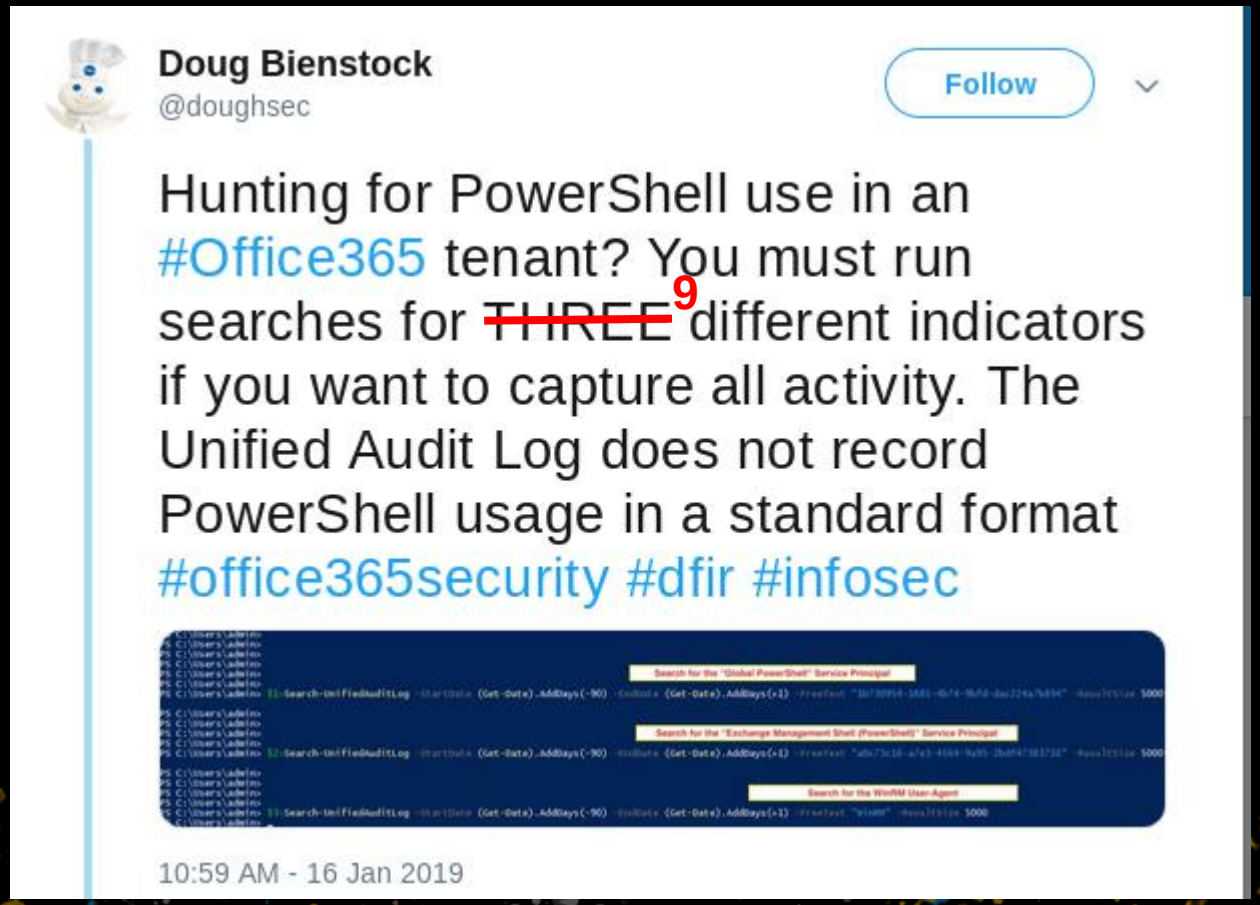

ActorContextId: 649a4dbf-4925-496d-a3e6-7abb28010f43

ActorIpAddress:

ApplicationId: a25dbca8-4e60-48e5-80a2-0664fdb5c9b6

How many different PowerShell usage records?

- At least nine depending on how many apps have impersonate privs



“Reversing” Azure AD

How does it actually work?

- No low-level access to Azure AD backend
- No way to use traditional reversing to find out more

Interacting with Azure AD

- Portal
- PowerShell modules
- API’s

Portal

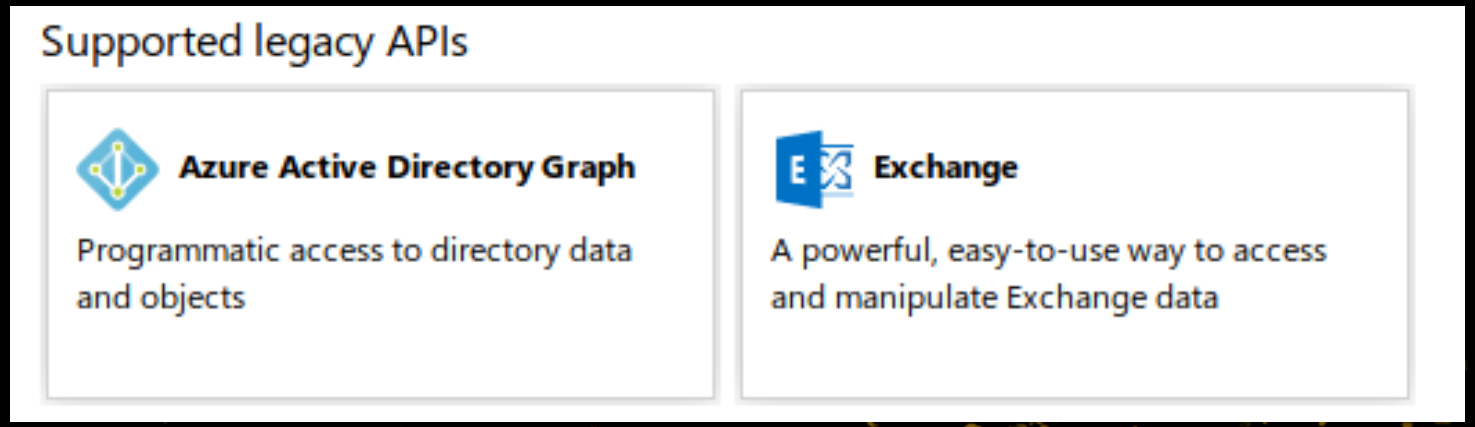
- Nice and shiny
- Offers (almost) all configuration options
- Does simplify concepts

API’s

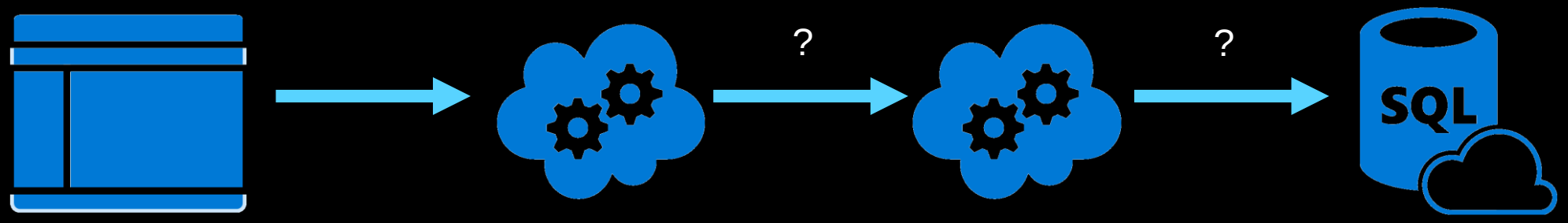
- Azure AD Graph
- Microsoft Graph
- Exchange Provisioning service

Which one to use?

- All of them have limitations
- Unique features, yet deprecated
- Different authentication methods supported
- Different terminology



Front-end vs backend (Azure AD)



Azure portal

Azure portal API

Internal API?

Azure AD graph?

Microsoft graph?

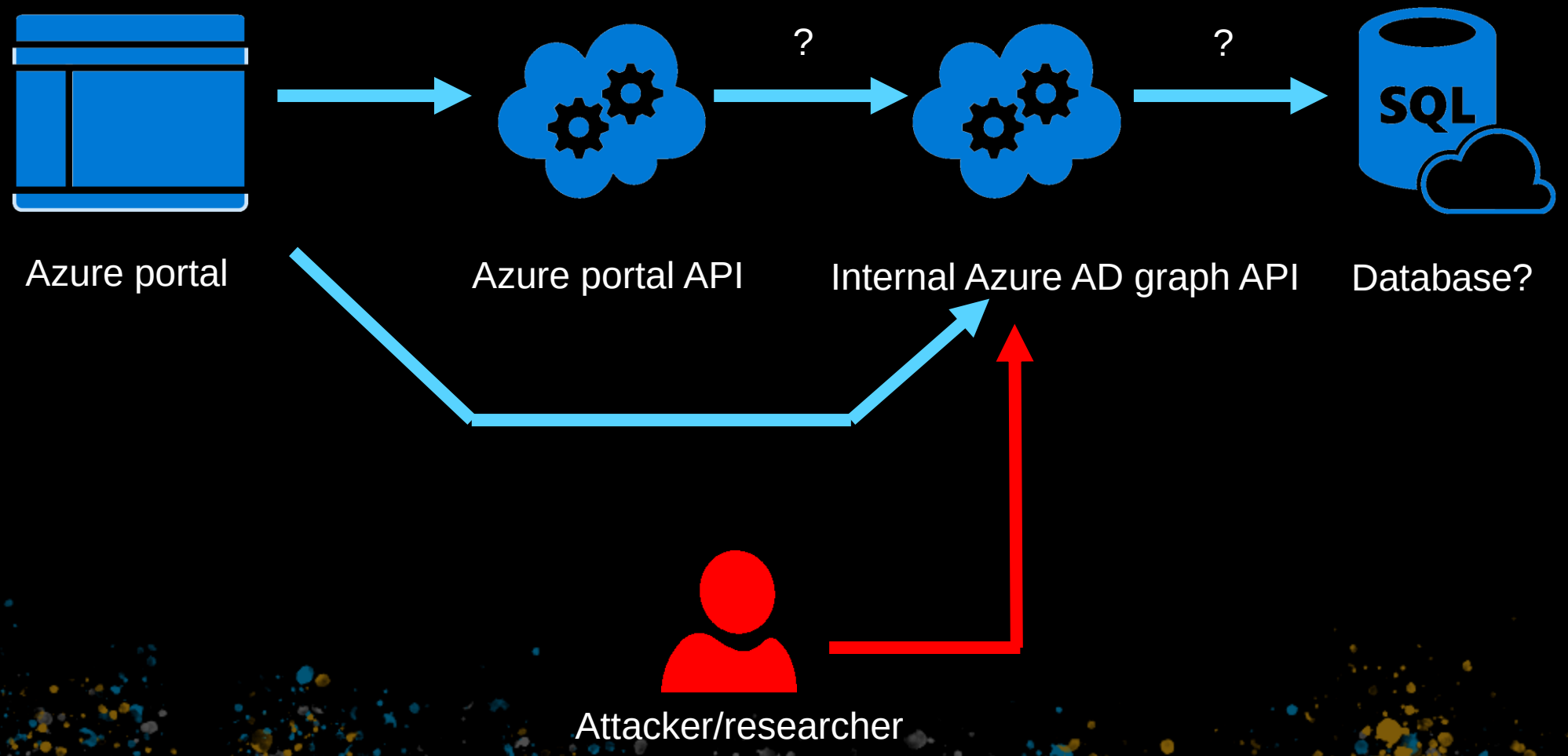
Database?

Once upon a time in the Azure Portal

Status	Method	Domain	File	Cause	Type	Transferred	Size	0 ms
200	GET	afd.hosting.portal.a...	zwB7yYcvLudD.js	fetch	js	6.03 KB	24.14 KB	44 ms
200	POST	portal.azure.com	DelegationToken?feature.refreshtokenbinding=true&featur...	xhr	json	2.93 KB	6.21 KB	317 ms
200	POST	portal.azure.com	DelegationToken?feature.refreshtokenbinding=true&featur...	xhr	json	3.48 KB	6.36 KB	152 ms
200	GET	graph.windows.net	roleDefinitions?api-version=1.61-internal&$top=500	xhr	json	68.92 KB	68.89 KB	143 ms
200	OPTIONS	main.iam.ad.ext.az...	CurrentContext	xhr	plain	39 B	0 B	95 ms
200	GET	main.iam.ad.ext.az...	CurrentContext	xhr	json	992 B	99 B	57 ms
200	OPTIONS	main.iam.ad.ext.az...	RoleAssignments?scope=undefined	xhr	plain	752 B	0 B	62 ms
200	GET	main.iam.ad.ext.az...	RoleAssignments?scope=undefined	xhr	json	2.08 KB	4.50 KB	395 ms

Status	Method	Domain		Type	Transferred	Size	0 ms
200	GET	afd.hosting		js	6.03 KB	24.14 KB	44 ms
200	POST	portal.azure		json	2.93 KB	6.21 KB	317 ms
200	POST	portal.azure		json	3.48 KB	6.36 KB	152 ms
200	GET	graph.wind	nitions?api-version=1.61-internal&s	json	68.92 KB	68.89 KB	143 ms
200	OPTIONS	main.iam.a		plain	39 B	0 B	95 ms
200	GET	main.iam.a		json	992 B	99 B	57 ms
200	OPTIONS	main.iam.a		plain	752 B	0 B	62 ms
200	GET	main.iam.a		json	2.08 KB	4.50 KB	395 ms

Front-end vs backend (Azure AD)



Azure AD graph metadata – internal version

```
133933 Jun 28 14:13 $metadata_internal.xml
 72527 Jun 21 21:40 $metadata.xml
```

File size


```
-<edmx:Edmx Version="3.0">
 -<edmx:DataServices m:DataServiceVersion="3.0" m:MaxDataServiceVersion="3.0">
  -<Schema Namespace="Microsoft.DirectoryServices">
   -<EntityType Name="DirectoryObject" OpenType="true">
    -<Key>
      <PropertyRef Name="objectId"/>
    </Key>
    <Property Name="objectType" Type="Edm.String"/>
    <Property Name="objectId" Type="Edm.String" Nullable="false"/>
    <Property Name="deletionTimestamp" Type="Edm.DateTime"/>
    <NavigationProperty Name="createdOnBehalfOf" Relationship="Microsoft.DirectoryServices.Mic
    <NavigationProperty Name="createdObjects" Relationship="Microsoft.DirectoryServices.Microsc
    <NavigationProperty Name="manager" Relationship="Microsoft.DirectoryServices.Microsoft_Dir
    <NavigationProperty Name="directReports" Relationship="Microsoft.DirectoryServices.Microsof
    <NavigationProperty Name="members" Relationship="Microsoft.DirectoryServices.Microsoft_Dir
    <NavigationProperty Name="transitiveMembers" Relationship="Microsoft.DirectoryServices.Mic
    <NavigationProperty Name="memberOf" Relationship="Microsoft.DirectoryServices.Microsoft_D
    <NavigationProperty Name="transitiveMemberOf" Relationship="Microsoft.DirectoryServices.Mi
    <NavigationProperty Name="owners" Relationship="Microsoft.DirectoryServices.Microsoft_Direc
    <NavigationProperty Name="ownedObjects" Relationship="Microsoft.DirectoryServices.Microsof
   </EntityType>
   -<EntityType Name="ExtensionProperty" BaseType="Microsoft.DirectoryServices.DirectoryObject" Op
    <Property Name="appDisplayName" Type="Edm.String"/>
    <Property Name="name" Type="Edm.String"/>
    <Property Name="dataType" Type="Edm.String"/>
    <Property Name="isSyncedFromOnPremises" Type="Edm.Boolean"/>
    <Property Name="targetObjects" Type="Collection(Edm.String)" Nullable="false"/>
   </EntityType>
```

Unannounced features…

```
<EntityType Name="Group" BaseType="Microsoft.DirectoryServices.DirectoryObject" OpenType="true">
  <Property Name="appMetadata" Type="Microsoft.DirectoryServices.AppMetadata"/>
  <Property Name="classification" Type="Edm.String"/>
  <Property Name="cloudSecurityIdentifier" Type="Edm.String"/>
  <Property Name="createdDateTime" Type="Edm.DateTime"/>
  <Property Name="description" Type="Edm.String"/>
  <Property Name="dirSyncEnabled" Type="Edm.Boolean"/>
  <Property Name="displayName" Type="Edm.String"/>
  <Property Name="exchangeResources" Type="Collection(Edm.String)" Nullable="false"/>
  <Property Name="expirationDateTime" Type="Edm.DateTime"/>
  <Property Name="externalGroupIds" Type="Collection(Microsoft.DirectoryServices.AlternativeSecurityId)" Nullable="false"/>
  <Property Name="externalGroupProviderId" Type="Edm.String"/>
  <Property Name="externalGroupState" Type="Edm.String"/>
  <Property Name="creationOptions" Type="Collection(Edm.String)" Nullable="false"/>
  <Property Name="groupTypes" Type="Collection(Edm.String)" Nullable="false"/>
  <Property Name="isAssignableToRole" Type="Edm.Boolean"/>
  <Property Name="isMembershipRuleLocked" Type="Edm.Boolean"/>
  <Property Name="isPublic" Type="Edm.Boolean"/>
  <Property Name="lastDirSyncTime" Type="Edm.DateTime"/>
  <Property Name="licenseAssignment" Type="Collection(Microsoft.DirectoryServices.LicenseAssignment)" Nullable="false"/>
  <Property Name="mail" Type="Edm.String"/>
  <Property Name="mailNickname" Type="Edm.String"/>
  <Property Name="mailEnabled" Type="Edm.Boolean"/>
  <Property Name="membershipRule" Type="Edm.String"/>
  <Property Name="membershipRuleProcessingState" Type="Edm.String"/>
  <Property Name="membershipTypes" Type="Collection(Edm.String)" Nullable="false"/>
  <Property Name="onPremisesSecurityIdentifier" Type="Edm.String"/>
  <Property Name="preferredDataLocation" Type="Edm.String"/>
```

More unannounced features (DPAPI)

```
user@localhost:~$ diff metadef_internal_prev_pp.xml metadef_oct10_pp
297a298
>                               <Property Name="isAssignableToRole" Type="Edm.Boolean"/>
322a324
>                               <NavigationProperty FromRole="eligibleMemberOfPartner" Name="eligibleMem
_Group_eligibleMemberOf_Microsoft_DirectoryServices_DirectoryObject_eligibleMemberOfPartner" ToRole="eli
422a425,441
>                       <EntityType Name="DpapiData">
>                               <Key>
>                                       <PropertyRef Name="objectId"/>
>                               </Key>
>                               <Property Name="objectId" Nullable="false" Type="Edm.String"/>
>                               <Property Name="keyData" Type="Edm.Binary"/>
>                       </EntityType>
>                       <EntityType Name="SecuredEncryptedData">
>                               <Key>
>                                       <PropertyRef Name="objectId"/>
>                               </Key>
>                               <Property Name="objectId" Nullable="false" Type="Edm.String"/>
>                               <Property Name="shardId" Nullable="false" Type="Edm.Int32"/>
>                               <Property Name="version" Nullable="false" Type="Edm.Int32"/>
>                               <Property Name="encryptionAlgorithm" Type="Edm.String"/>
>                               <Property Name="encryptedData" Type="Edm.Binary"/>
```

Interesting things

https://docs.microsoft.com/en-us/azure/active-directory/conditional-ac

Search

Filter by title

Conditional Access Documentation

- Overview
- Quickstarts
- Tutorials

management settings.

Are Graph APIs available for configuring Conditional Access policies?

Currently, no.

GET AzureAD | GET https://graph.windows.net/wills... | GET https://graph.windows.net/649...

GET https://graph.windows.net/myorganization/policies?api-version=1.6

Pretty | Raw | Preview | Visualize BETA | JSON

```
1 {
2     "odata.metadata": "https://graph.windows.net/myorganization/$metadata#directoryObjects",
3     "value": []
4 }
```

GET https://graph.windows.net/myorganization/policies?api-version=1.61-internal Send Save

Pretty Raw Preview Visualize BETA JSON


```
18      {
19          "odata.type": "Microsoft.DirectoryServices.Policy",
20          "objectType": "Policy",
21          "objectId": "96298eaf-85c1-4911-9fd1-0c3d85dfede3",
22          "deletionTimestamp": null,
23          "displayName": "Baseline policy: Require MFA for admins",
24          "keyCredentials": [],
25          "policyType": 10,
26          "policyDetail": [
27              "{\"PolicyKind\":\"RequireMfaForAdmins\",\"Version\":4,
                    \"CreatedDateTime\":\"2019-05-29T14:25:40Z\",
                    \"ModifiedDateTime\":\"2019-05-30T14:25:40Z\",\"State\":\"Enabled\",\"Conditions\":
                    {\"Users\":{\"Include\":[{\"Roles\":[\"194ae4cb-b126-40b2-bd5b-6091ba8097ad\",
                    \"29232cdf-a323-42fd-ade2-1d097af3e4df\",\"62e90394-69f5-4237-9a90-012177145e1a\",
                    \"b1be1c3e-b65d-4a19-8427-f6fa0d97feba\",\"f28a1f50-f6e7-4571-818b-6a12f2af6baa\"]}]}},
                    \"Controls\":[{\"Control\":[\"Mfa\"]}],
                    \"IncludeOtherLegacyClientTypeForEvaluation\":true}"
28          ],
29          "policyIdentifier": null,
30          "tenantDefaultPolicy": null
31      },
```


Can be queried by any authenticated user

The Access Policy that wasn't

- Change "Mfa" control to "Allow"
- Invisible in portal

Microsoft Azure
Search resources, services, and docs
eric@ericsengines.on...
MSOBB

Home > Conditional Access - Policies

Create a resource
Home
Dashboard
All services
FAVORITES
All resources
Resource groups
App Services
Function App
SQL databases

Conditional Access - Policies
Azure Active Directory

Policies
Manage
Named locations
Custom controls (preview)
Terms of use
VPN connectivity
Classic policies

New policy | What If | Got feedback?

Interested in understanding the impact of the policies on a user sign-in? Check out the "What If" tool.

POLICY NAME	ENABLED	
Baseline policy: Block legacy authentication (Preview)	✓	...
Baseline policy: Require MFA for admins (Preview)	✓	...
Baseline policy: End user protection (Preview)		...
Baseline policy: Require MFA for Service Management (Pre...		...

Logs

- No details on properties

DATE	SERVICE	CATEGORY	ACTIVITY	STATUS	TARGET(S)	INITIATED BY (ACTOR)
7/22/2019, 2:36:54 PM	Core Directory	Policy	Update policy	Success	Baseline policy: Require...	notsync@ericsengines.o...
7/22/2019, 2:35:16 PM	Core Directory	Policy	Update policy	Success	Default Policy 10	eric@ericsengines.onmi...
7/22/2019, 2:35:16 PM	Core Directory	Policy	Add policy	Success	Baseline policy: Require...	eric@ericsengines.onmi...
7/22/2019, 2:34:50 PM	Core Directory	Policy	Delete policy	Success	Baseline policy: Require...	eric@ericsengines.onmi...

Details

Activity Target(s) Modified Properties

TARGET	PROPERTY NAME	OLD VALUE	NEW VALUE
Baseline policy: Require MFA for admins	Included Updated Properties		""

Sign-in logs

- Passes checks

Apply Reset

DATE	USER	APPLICATION	STATUS	IP ADDRESS	CONDITIONAL ACCESS
7/22/2019, 2:43:21 PM	Eric	Azure Portal	Success		Success
7/22/2019, 2:43:07 PM	Eric	Azure Portal	Interrupted		Success
7/22/2019, 2:42:44 PM	Eric	Azure Portal	Success		Success

Details

Basic info | Device info | MFA info | Conditional Access

POLICY NAME	GRANT CONTROLS	SESSION CONTROLS	RESULT
Baseline policy: Require MFA for admins			Unknown Future Value

A sign-in can also be interrupted (e.g. blocked, MFA challenged) because of a user risk policy or sign-in risk policy. Currently, this tab only lists Conditional Access policies.

Conditional access policies backdooring TL;DR

- Details only available via undocumented API
- Impossible to see in the portal
- Not really visible in logs
- Attack possibilities:
  - Exclude specific users
  - Disable entire policy
  - Change trusted networks
- Fixed in October 2019

Digging into Service Principals

ReplyUrls

- Used for OAuth2 implicit grant (web)
- Whitelist of URLs
- Sends access token to URL in fragment (#)

ReplyUrls don’ts

- Non-https URL
  - Portal refuses, API accepts
- Expired domain
- Relative URL

Demo

You're not connected

Mail – deviceuser@eric:

https://outlook.office.com/owa/?realm=ericsengines.onmicrosoft.com

Office 365 | Outlook

Search Mail and People

New | Mark all as read

Undo

Try the new Outlook

Folders

Inbox 2

Sent Items

Drafts

More

Groups New

Groups give teams a shared space for email, documents, and scheduling events.

Discover

Create

Focused Other

Filter

Next: No events for the next two days.

Agenda

Eric

New password Wed 17:09

Hi, Your new password is: ZI4uV5Jje#5#WTSn Kind regar...

Eric

Email from ADMIN Wed 16:51

Hi!

Select an item to read

Click here to always select the first item in the list

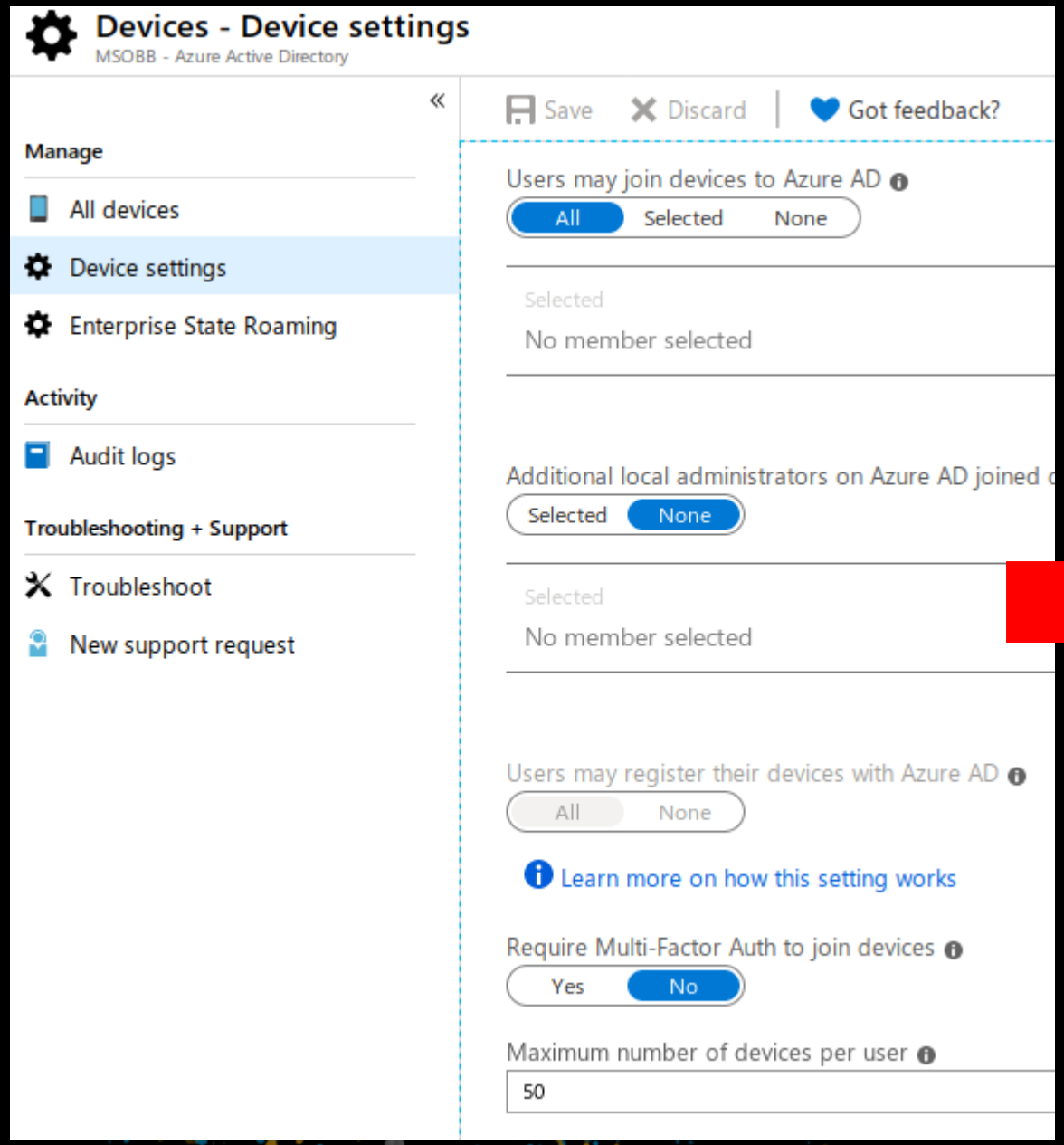
ENG 01:55 11/07/2019

Office 365 insecure Reply URLs

- Microsoft Teams Web Client
  - Whitelisted http://dev.local
- Has read/write access to Email, SharePoint, OneDrive
- Allows for man/person in the middle attack
- Not possible to identify in logs (Reply URL is not logged)
- Fixed for new Office 365 tenants in September 2019, existing tenants in October 2019
- More details: https://dirkjanm.io/office-365-network-attacks-via-insecure-reply-url/

appMetaData

- Hidden property “appMetaData” – only visible in internal API
- Mostly for custom apps (Federated Certificate Storage)
  - Only used for a few apps by default
- Interesting case “Device Registration Service”




```
{
   "version":2,
   "serviceOn":true,
   "allowAdminOnOffControl":false,
   "registrationPolicies":[
      {
         "action":"Workplace",
         "scope":"All",
         "adminConfigurable":false
      },
      {
         "action":"DeviceJoin",
         "scope":"All",
         "adminConfigurable":true
      }
   ],
   "logonPolicies":[
      {
         "action":"LogonAsUser",
         "scope":"All",
         "adminConfigurable":true
      },
      {
         "action":"LogonAsAdmin",
         "scope":"None",
         "adminConfigurable":true,
         "sids":[
            "S-1-12-1-3002916598-1200172303-198343323-2875385887",
            "S-1-12-1-2653652498-1307445353-1462749103-3254925571"
         ]
      }
   ]
}
```

Device settings

- Property of service principal
- Can be edited by Application Administrator

```
"logonPolicies":[
    {
        "action":"LogonAsUser",
        "scope":"All",
        "adminConfigurable":true
    },
    {
        "action":"LogonAsAdmin",
        "scope":"None",
        "adminConfigurable":true,
        "sids":[
            "S-1-12-1-3002916598-1200172303-198343323-2875385887",
            "S-1-12-1-2653652498-1307445353-1462749103-3254925571"
        ]
    }
]
}
```

Security Identifiers in Azure AD

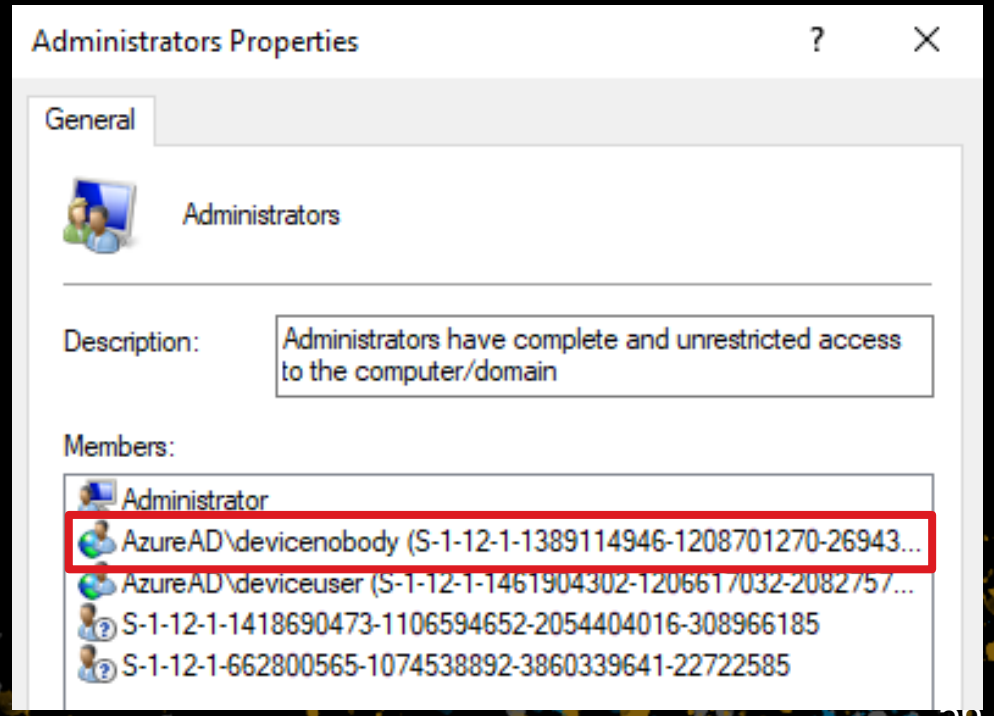
- Following principals have security identifiers
  - Users
  - Groups
  - Roles

- Stored in “cloudSecurityIdentifier” property (internal API only)

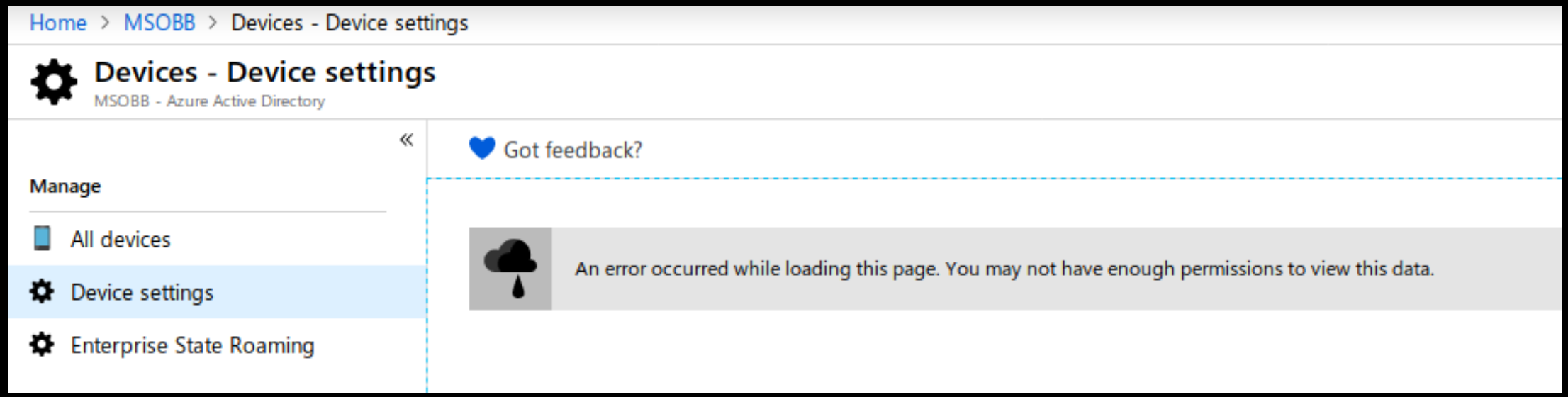
displayName	cloudSecurityIdentifier
Filter	Filter
Company Administrator	S-1-12-1-3002916598-1200172303-198343323-2875385887
Device Administrators	S-1-12-1-2653652498-1307445353-1462749103-3254925571

Application Administrator to local Admin on devices

- Application Administrator can add SIDs to policy in metadata
  - Can be users/groups/roles
- New device joined? User gets added to Administrators group
- Rogue user is now admin on device
- Not yet fixed (by design)



Portal doesn’t seem to like it…




```
[Microsoft_AAD_IAM]  8:10:48 PM MsPortalFx/Base/Base.Net 1 Base.Net: readyState: 4
responseJSON: {"ClassName":"Microsoft.Portal.Framework.Exceptions.ClientException","Message":"ArgumentNull","Data":{},"HResult":-
2146233088,"XMsServerRequestId":null,"Source":null,"HttpStatusCode":500,"ClientData":
{"errorCode":"InternalServerError","localizedErrorDetails":null,"operationResults":null,"timeStampUtc":"2019-10-13T18:10:50.1940254Z","clientRequ
estId":"af16586e-4b3a-4636-a101-60684c96a051","internalTransactionId":"525ce9e5-93ce-44ea-8699-f5ab1ca246df","tenantId":"50ad18e1-bb23-4466-9154-
bc92e7fe3fbb","userObjectId":"e0cd1b1c-d57a-4d31-a52b-50eee61836f3","exceptionType":"CommunicationException"}}
```

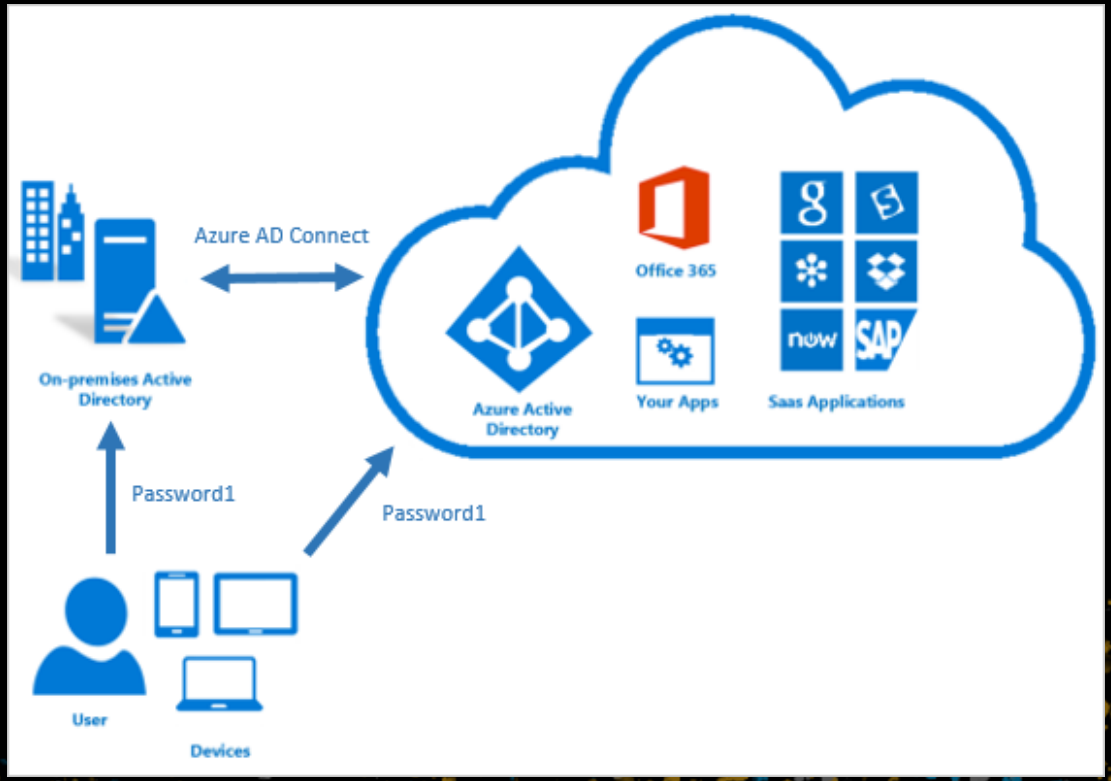
Linking up Cloud an on-prem

Exploiting the link with on-premise

- Application administrator is high-privilege cloud account
  - Hopefully protected with MFA
- What about on-premise?

Azure AD connect

- Tool that resides on-premise and syncs AD data to Azure AD
- Installed in both Password Hash Synchronization and ADFS scenario's



Source: https://docs.microsoft.com/en-us/azure/active-directory/hybrid/whatis-phs

Previous vulnerability – Password Hash Sync

- Possible to link new on-premise account to existing cloud-only account
- Anyone with user creation privileges on-premise could overwrite the password of (admin) accounts in the cloud

https://blog.fox-it.com/2019/06/06/syncing-yourself-to-global-administrator-in-azure-active-directory/

Sync account privileges

- If Password Hash Synchronization is in use, the Sync account can sync all password hashes
  - Means it’s basically Domain Admin on-premise
- Both with PHS and ADFS sync account has high privileges in the cloud
- Cloud assets may extend beyond the AD Domain

Azure AD Connect password extraction

- Adconnectdump: 3 ways to dump the password on-premises
- Technical explanation: see my Troopers presentation

Tool	Requires code execution on target	DLL dependencies	Requires MSSQL locally	Requires python locally
ADSyncDecrypt	Yes	Yes	No	No
ADSyncGather	Yes	No	No	Yes
ADSyncQuery	No (network RPC calls only)	No	Yes	Yes

https://github.com/fox-it/adconnectdump

AD Sync account privileges in Azure AD

Directory Synchronization Accounts permissions

Only used by Azure AD Connect service.

Actions	Description
microsoft.directory/organization/dirSync/update	Update organization.dirSync property in Azure Active Directory.
microsoft.directory/policies/create	Create policies in Azure Active Directory.
microsoft.directory/policies/delete	Delete policies in Azure Active Directory.
microsoft.directory/policies/basic/read	Read basic properties on policies in Azure Active Directory.
microsoft.directory/policies/basic/update	Update basic properties on policies in Azure Active Directory.
microsoft.directory/policies/owners/read	Read policies.owners property in Azure Active Directory.

microsoft.aad.directory/servicePrincipals /appRoleAssignments/update	Update servicePrincipals.appRoleAssignments property in Azure Active Directory.
microsoft.aad.directory/servicePrincipals/audience /update	Update servicePrincipals.audience property in Azure Active Directory.
microsoft.aad.directory/servicePrincipals/authentication /update	Update servicePrincipals.authentication property in Azure Active Directory.
microsoft.aad.directory/servicePrincipals/basic/read	Read basic properties on servicePrincipals in Azure Active Directory.
microsoft.aad.directory/servicePrincipals/basic/update	Update basic properties on servicePrincipals in Azure Active Directory.
microsoft.aad.directory/servicePrincipals/create	Create servicePrincipals in Azure Active Directory.
microsoft.aad.directory/servicePrincipals/credentials /update	Update servicePrincipals.credentials property in Azure Active Directory.
microsoft.aad.directory/servicePrincipals /memberOf/read	Read servicePrincipals.memberOf property in Azure Active Directory.

~~Fun~~ bad stuff to do with the Sync account

- Dump all on-premise password hashes (if PHS is enabled)
- Log in on the Azure portal (since it’s a user)
- Bypass conditional access policies for admin accounts
- Add credentials to service principals
- Modify service principals properties
- Modify/backdoor/remove conditional access policies (internal API)

Azure Resource Manager RBAC

- RBAC roles can be assigned to service principals
- These can be managed by Application Administrators
- Also by the on-premise sync account
- High privilege applications might need an account
  - Example: Terraform
- Service principals credentials can be assigned by these accounts
  - Control over cloud resources

Conclusions

Conclusions / recommendations

- Internal API version gives some insight into inner Azure AD workings
- Application Administrators are more powerful than you'd think
  - Avoid using "global" Application Administrators, use scoped/custom roles instead
- Service Principals can be backdoored and abused
  - Monitor for credential modification
  - Review credentials/owners
  - Review permissions and reply URLs for security issues

Conclusions / recommendations (2)

- Enforce MFA for all admin accounts
  - (Preferably for all accounts)
- Use conditional access policies
  - Monitor modifications made
- The AD Sync account is highly privileged on-prem and in the cloud
  - Treat it's system as Tier 0
  - Monitor for sign-ins from strange IP addresses
- Implement recommendations from Sean Metcalf and Mark Morowczynski's talk "Attacking and Defending the Microsoft Cloud"