Utilizing Microsoft Graph API and Office 365 Management Activity API during security investigations

Kirill Bogdanov,
Security TSP, Microsoft

Agenda:

- Audit options available in Office 365 and Azure AD
- Office 365 Management Activity API
- Azure AD Audit API
- DIY PowerShell script to download logs
- Investigating attack

Key takeaways:

- Understand audit options in Office 365 and Azure
- Get basic understanding of APIs for retrieving audit events
- Observe example investigation utilizing the APIs

Introduction

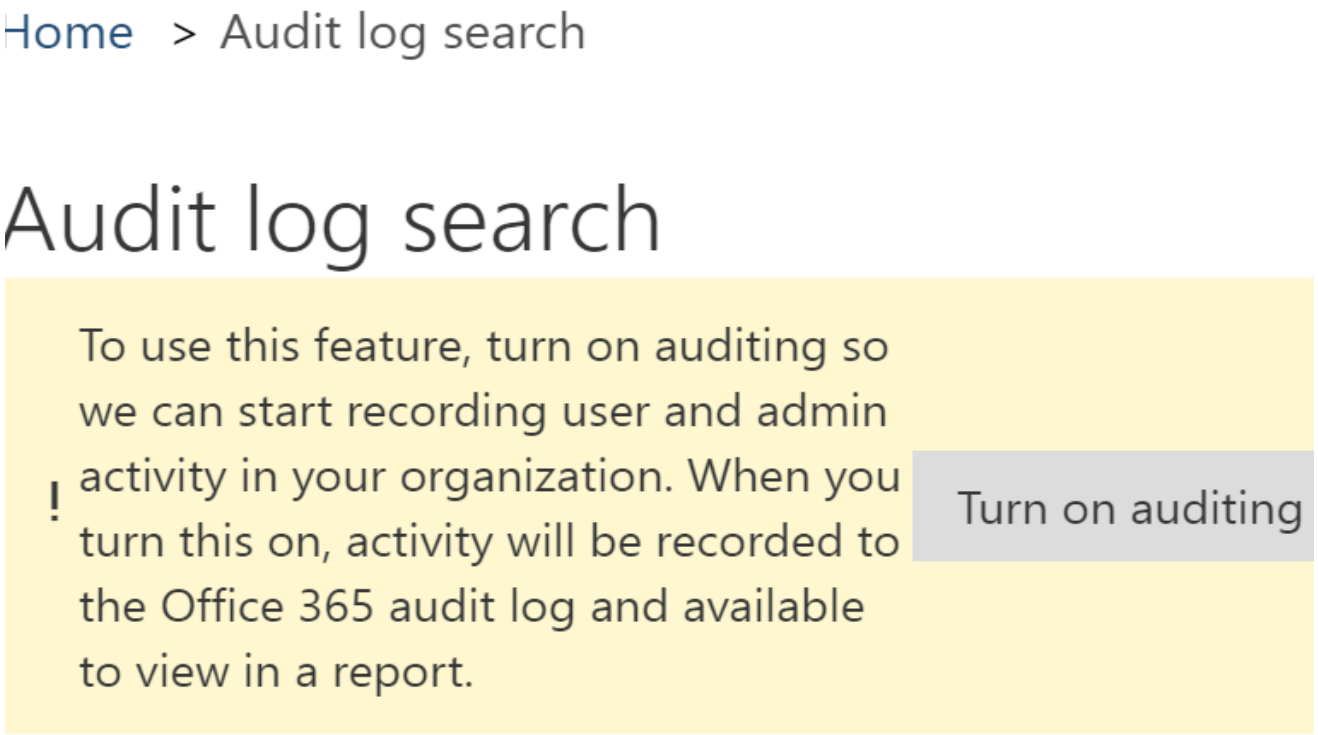
Kirill Bogdanov

```
PS C:\>
PS C:\> Get-Speaker $kirbogd
Name:   Kirill Bogdanov
Role:   Security TSP
Job Description:   5 years of helping enterprise customers stay secure in cloud
Previous experience
    :  Architect
    :  System Engineer
    :  Field Engineer
@Twitter:   @Kirbogd
Description:   Passionate InfoSecurity noob
```

Audit options available in Office 365 and Azure AD

What activities are available?

- Administrative actions
- Sign-on information
- User actions
- DLP alerts
- eDiscovery requests
- Depending on subscription



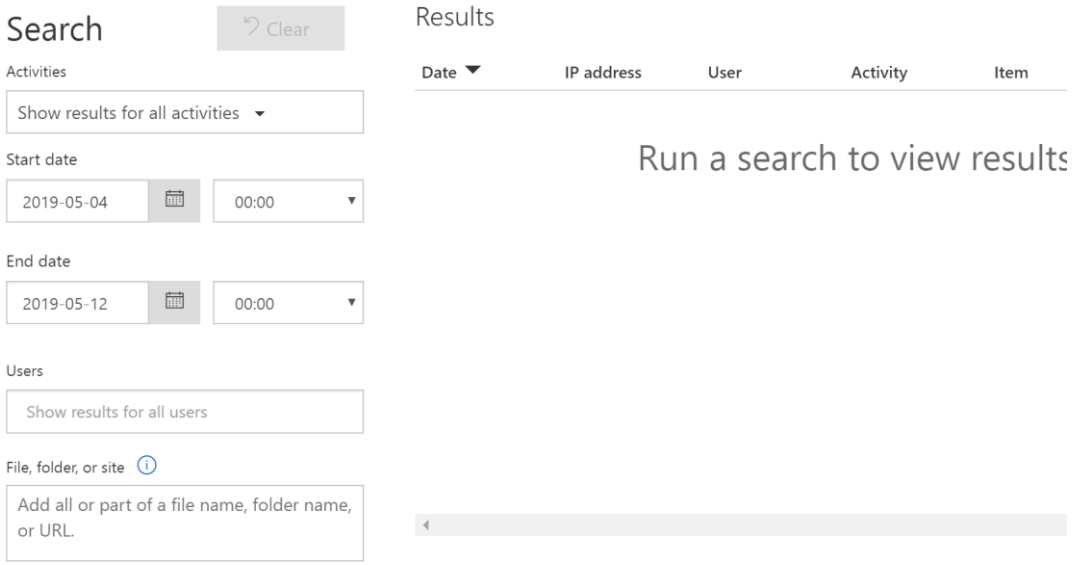
Retention

- Events are stored for 90 days
- Office 365 E5 users’ actions will be retained for 365 days (Private preview)
- Office 365 Management Activity API exposes only 7 days of history

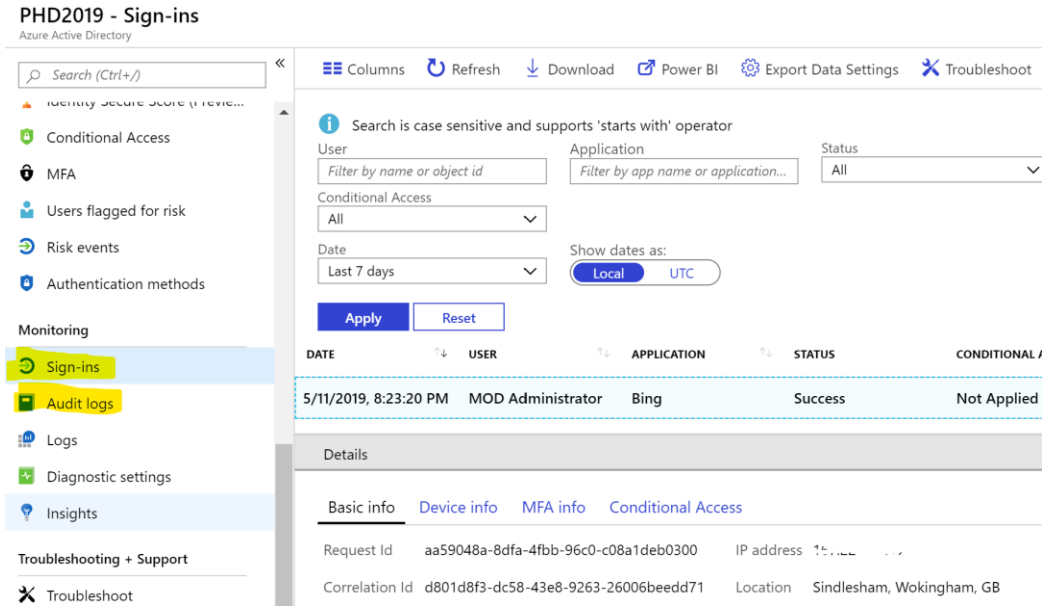
Ways to access events

Administrative portals

Security and Compliance Center
https://protection.office.com



Azure Portal
https://aad.portal.azure.com

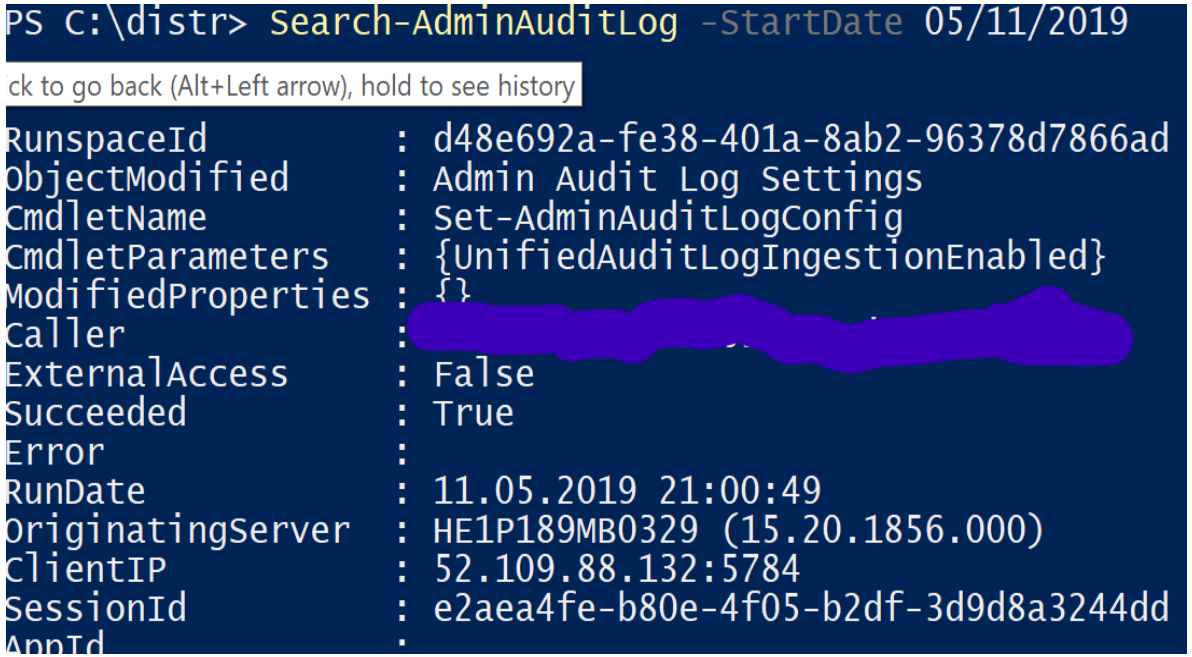


Exchange Online PowerShell Admin tool

Search-AdminAuditLog

Search-MailboxAuditLog

Search-UnifiedAuditLog



«Get into local infrastructure»

Download using native SIEM connectors and Office 365 Management Activity API

Microsoft Cloud App Security SYSLOG SIEM connector

Create a connector by yourself!

Office 365 Management Activity API

Office 365 Management Activity API

REST web service

Uses Azure AD and OAuth2 for authentication and Authorization

SO…

1. Register App in Azure AD
2. Get JWT Token for the App and activity API and craft a header
   (use https://login.windows.net/{TenantID}/oauth2/token/{TenantID} for Authority)
3. Use REST methods to get or post info from/to API
   (Use https://manage.office.com/api/v1.0/{TenantID}/activity/feed for requests)

Office 365 Management Activity API

5 content types:

- Audit.AzureActiveDirectory
- Audit.Exchange
- Audit.Sharepoint
- Audit.General
- DLP.All

Office 365 Management Activity API

Step1: Enable Content type (optionally add webhook)

POST {root}/subscriptions/stop?contentType=Audit.SharePoint

Step2: Retrieve available content

GET {root}/subscriptions/content?contentType=Audit.SharePoint

Step3: Retrieve events in JSON form from content URI

GET {root}/audit/20129900723 1$301299007231

Office 365 Management Activity API

Data is aggregated from different sources and is not aligned in time

Some sources can take up to 24h to provide data

Request throttling – 60K req/min via PublisherIdentifier

By default returns content for the last 24 hours. You can specify any interval less then 24h within last 7 days

Results are paginated with NextPageUri returned in the response

Webhook will get content address as soon as content is available

Azure AD audit log API

Azure AD Audit Log API

Subset of Microsoft Graph REST API

Supports OData query parameters for response customization

Sign-on data requires Azure AD P1 or higher

Information contained depends on available subscriptions

Uses Azure AD and OAuth2 for authentication and Authorization

SO…

Azure AD Audit Log API

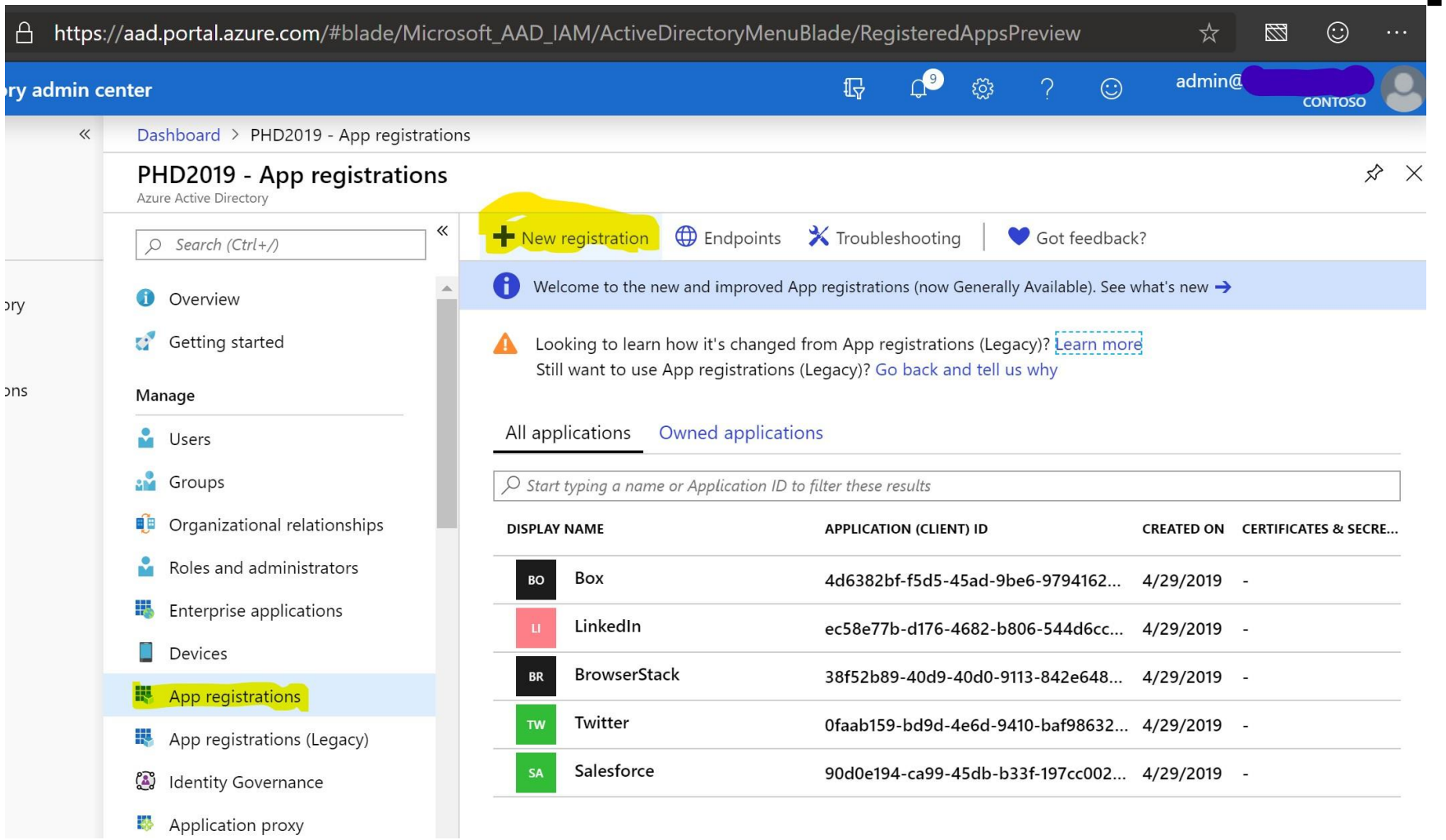
1. Register App in Azure AD

2. Get JWT Token for the App and activity API and craft a header
   (use https://login.Microsoftonline.com/{TenantID} for Authority)

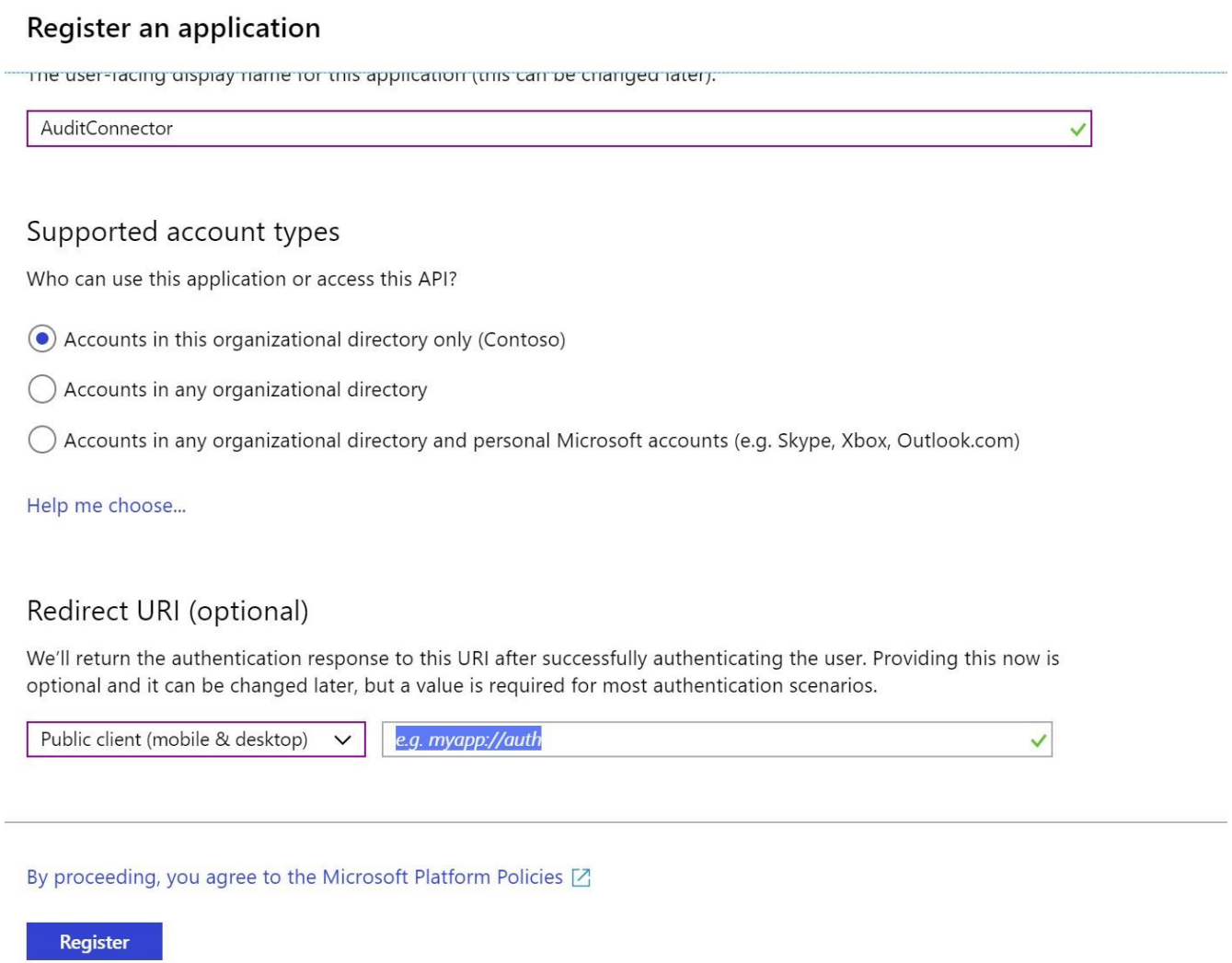
3. Use REST methods to get or post info from/to API
   (Use https://graph.microsoft.com/{version}/{resource}?query-parameters for requests)

Crafting connector

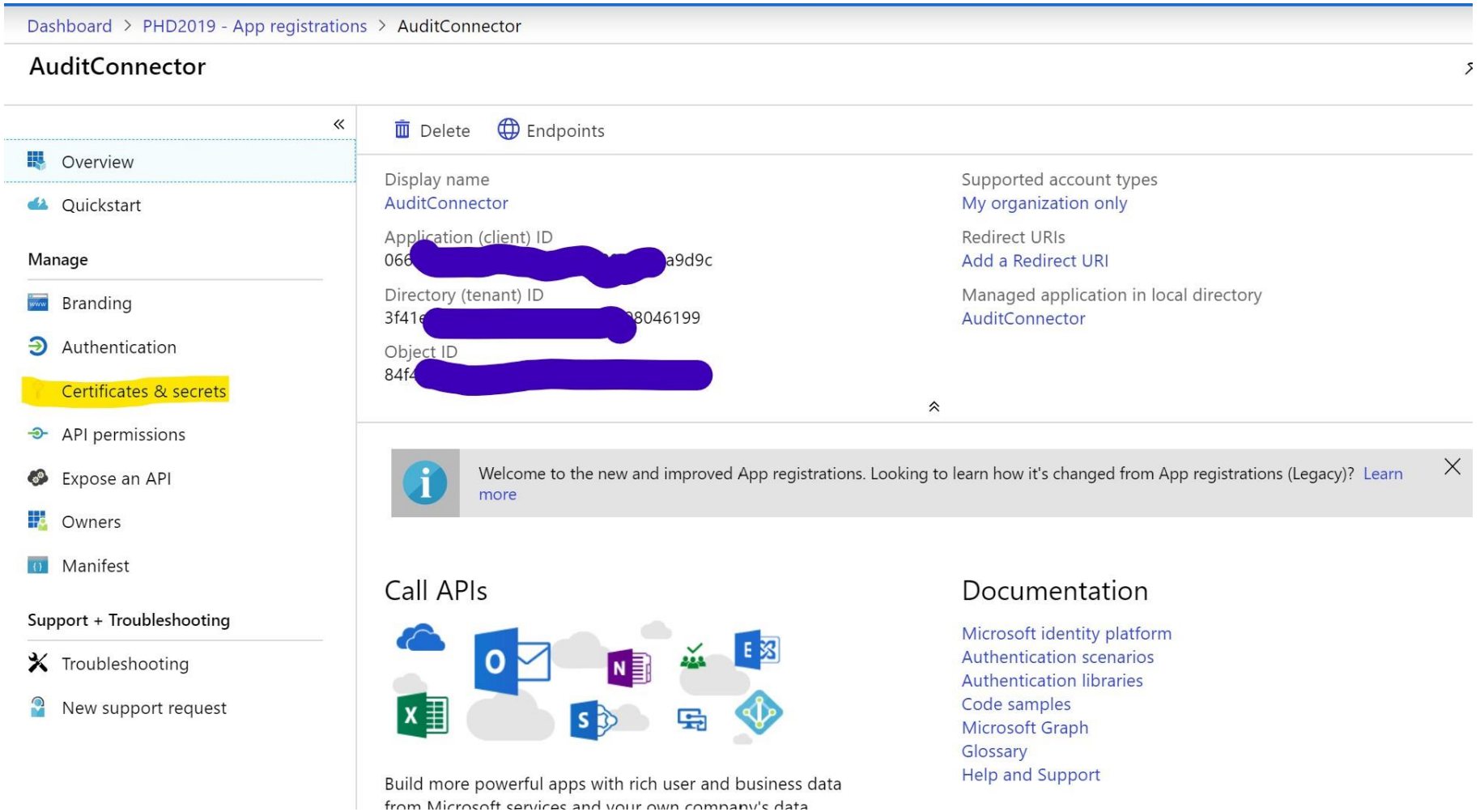
Demo


https://aad.portal.azure.com/#blade/Microsoft_AAD_IAM/ActiveDirectoryMenuBlade/RegisteredAppsPreview
admin center
admin@
CONTOSO
Dashboard > PHD2019 - App registrations
PHD2019 - App registrations
Azure Active Directory
Search (Ctrl+/)
New registration
Endpoints
Troubleshooting
Got feedback?
Overview
Getting started
Manage
Users
Groups
Organizational relationships
Roles and administrators
Enterprise applications
Devices
App registrations
App registrations (Legacy)
Identity Governance
Application proxy
Welcome to the new and improved App registrations (now Generally Available). See what's new
Looking to learn how it's changed from App registrations (Legacy)? Learn more
Still want to use App registrations (Legacy)? Go back and tell us why
All applications
Owned applications
Start typing a name or Application ID to filter these results
DISPLAY NAME
APPLICATION (CLIENT) ID
CREATED ON
CERTIFICATES & SECRE...
BO Box 4d6382bf-f5d5-45ad-9be6-9794162... 4/29/2019 -
LI LinkedIn ec58e77b-d176-4682-b806-544d6cc... 4/29/2019 -
BR BrowserStack 38f52b89-40d9-40d0-9113-842e648... 4/29/2019 -
TW Twitter 0faab159-bd9d-4e6d-9410-baf98632... 4/29/2019 -
SA Salesforce 90d0e194-ca99-45db-b33f-197cc002... 4/29/2019 -

Demo


Register an application
The user-facing display name for this application (this can be changed later).
AuditConnector
Supported account types
Who can use this application or access this API?
Accounts in this organizational directory only (Contoso)
Accounts in any organizational directory
Accounts in any organizational directory and personal Microsoft accounts (e.g. Skype, Xbox, Outlook.com)
Help me choose...
Redirect URI (optional)
We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.
Public client (mobile & desktop)
e.g. myapp://auth
By proceeding, you agree to the Microsoft Platform Policies
Register

Demo

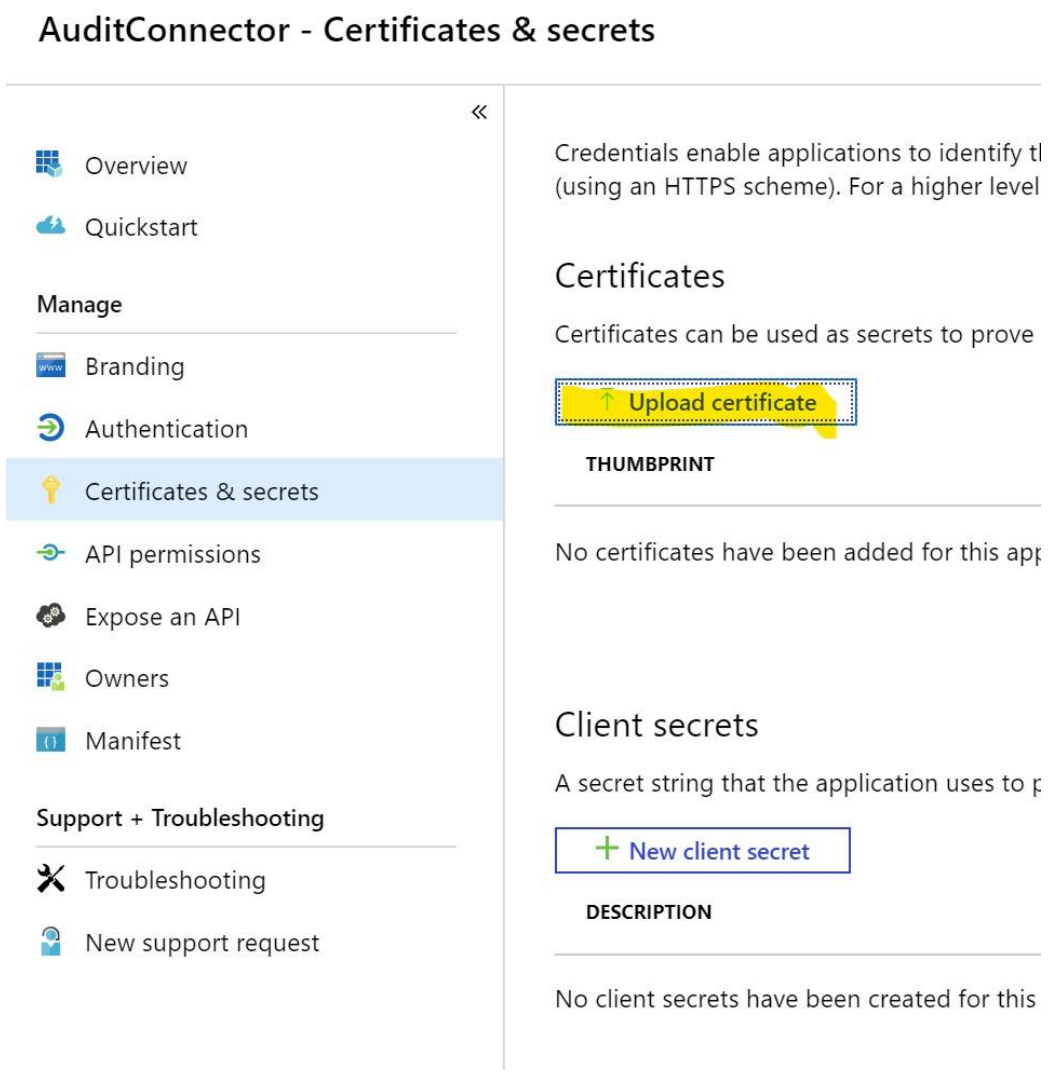

Dashboard > PHD2019 - App registrations > AuditConnector
AuditConnector
Delete
Endpoints
Overview
Quickstart
Manage
Branding
Authentication
Certificates & secrets
API permissions
Expose an API
Owners
Manifest
Support + Troubleshooting
Troubleshooting
New support request
Display name
AuditConnector
Application (client) ID
Directory (tenant) ID
Object ID
Supported account types
My organization only
Redirect URIs
Add a Redirect URI
Managed application in local directory
AuditConnector
Welcome to the new and improved App registrations. Looking to learn how it's changed from App registrations (Legacy)? Learn more
Call APIs
Build more powerful apps with rich user and business data
Documentation
Microsoft identity platform
Authentication scenarios
Authentication libraries
Code samples
Microsoft Graph
Glossary
Help and Support

Demo

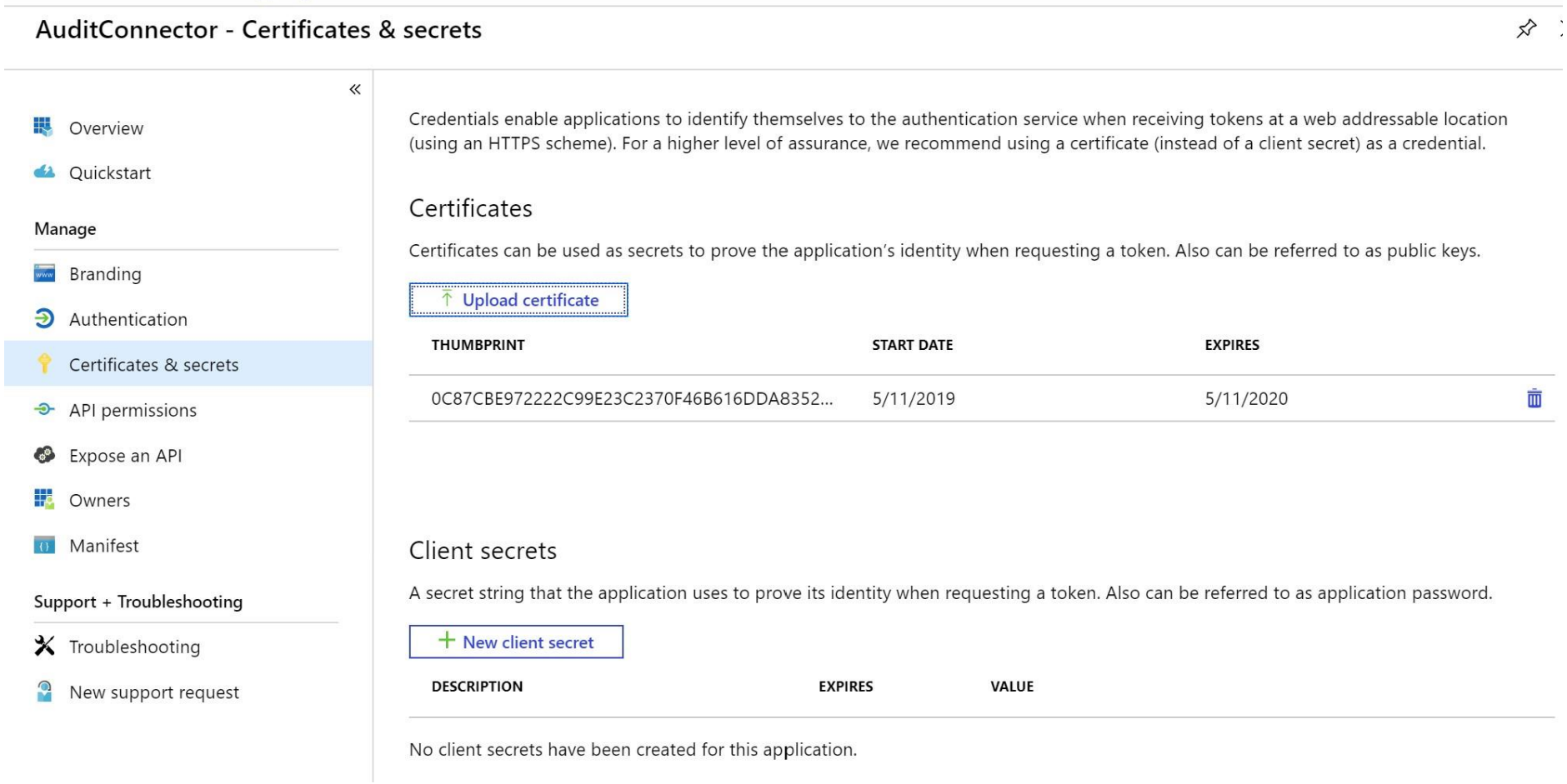
```
PS C:\distr> $cert=New-SelfSignedCertificate -Subject "CN=AuditAPIConnector" -CertStoreLocation "Cert:\CurrentUser\My"  -KeyEx
portPolicy Exportable -KeySpec Signature
PS C:\distr> $cert


   PSParentPath: Microsoft.PowerShell.Security\Certificate::CurrentUser\My

Thumbprint                                Subject
----------                                -------
0C87CBE972222C99E23C2370F46B616DDA835244  CN=AuditAPIConnector
```

```
PS C:\distr> Export-Certificate -Cert $cert -Type CERT -FilePath C:\distr\AuditAPIConnector.cer


    Directory: C:\distr


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        11.05.2019     21:58            790 AuditAPIConnector.cer


PS C:\distr>
```

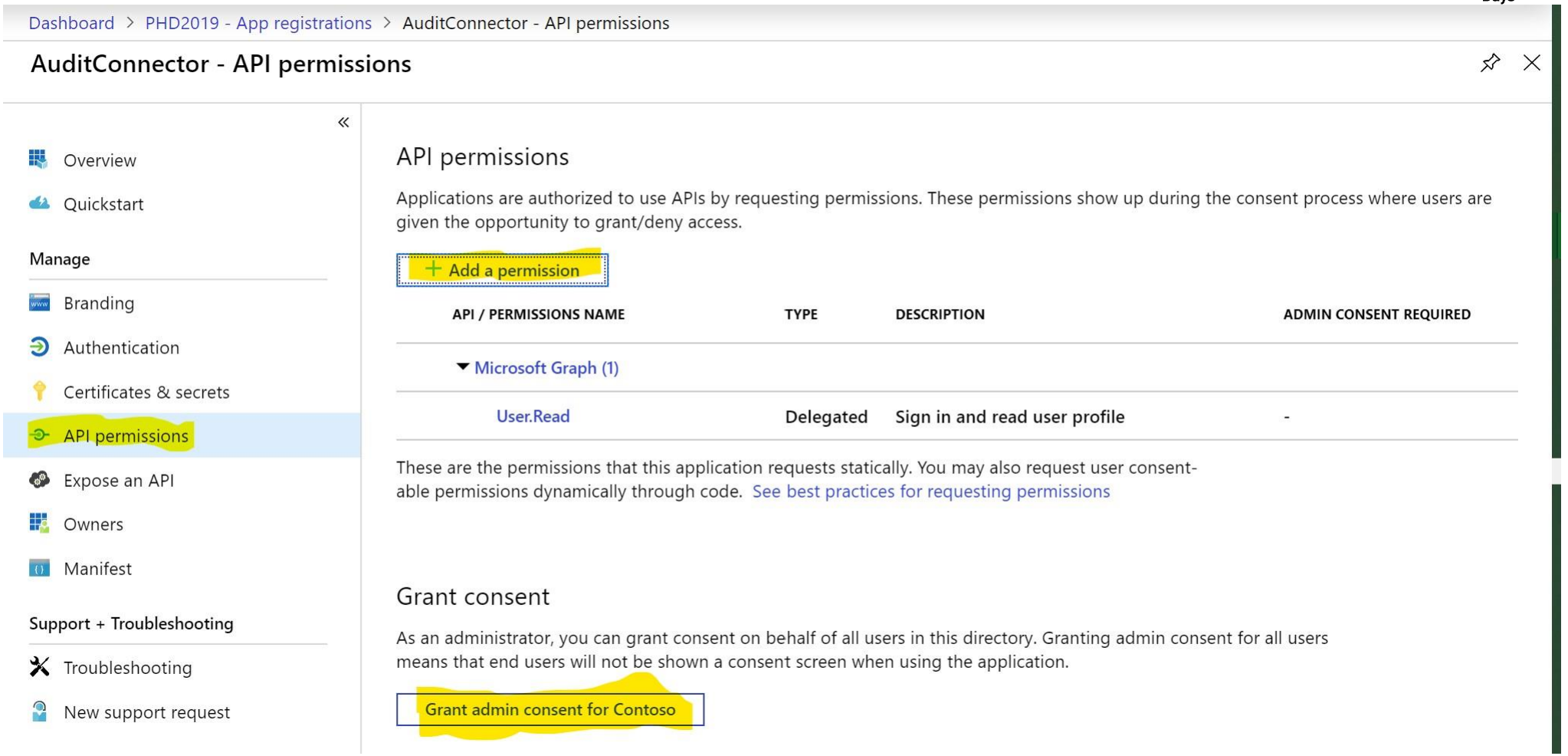
Demo


AuditConnector - Certificates & secrets
Overview
Quickstart
Manage
Branding
Authentication
Certificates & secrets
API permissions
Expose an API
Owners
Manifest
Support + Troubleshooting
Troubleshooting
New support request
Credentials enable applications to identify tl
(using an HTTPS scheme). For a higher level
Certificates
Certificates can be used as secrets to prove
Upload certificate
THUMBPRINT
No certificates have been added for this app
Client secrets
A secret string that the application uses to p
New client secret
DESCRIPTION
No client secrets have been created for this

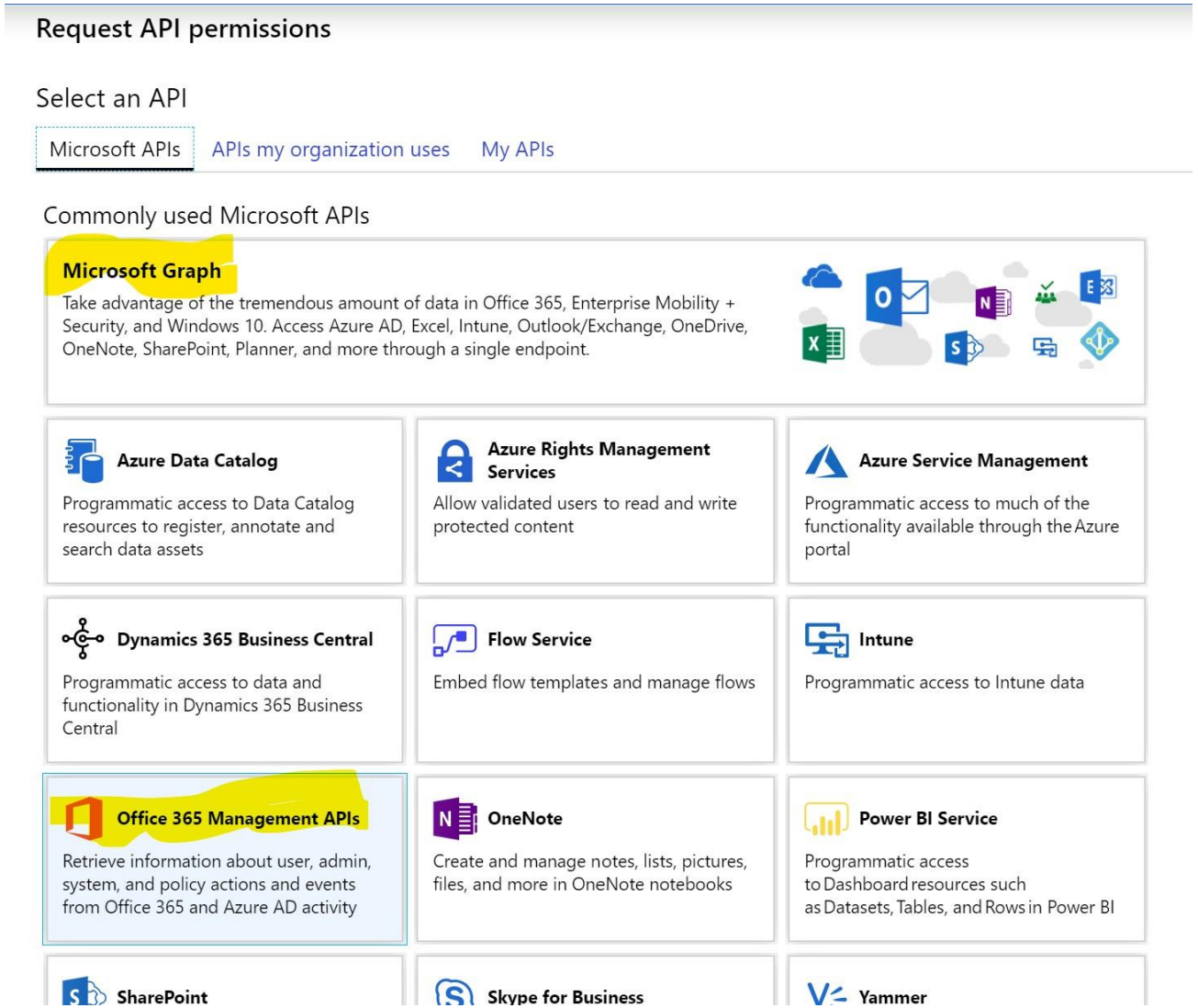
Demo


AuditConnector - Certificates & secrets
Overview
Quickstart
Manage
Branding
Authentication
Certificates & secrets
API permissions
Expose an API
Owners
Manifest
Support + Troubleshooting
Troubleshooting
New support request
Credentials enable applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.
Certificates
Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.
Upload certificate
THUMBPRINT
START DATE
EXPIRES
0C87CBE972222C99E23C2370F46B616DDA8352...
5/11/2019
5/11/2020
Client secrets
A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.
New client secret
DESCRIPTION
EXPIRES
VALUE
No client secrets have been created for this application.

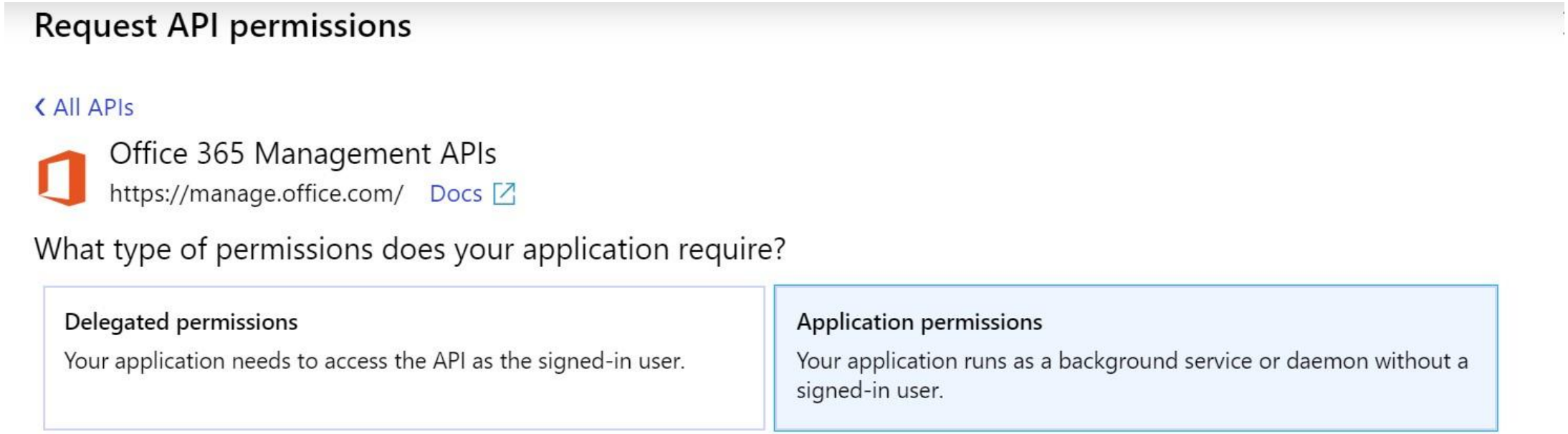
Demo


Dashboard > PHD2019 - App registrations > AuditConnector - API permissions
AuditConnector - API permissions
Overview
Quickstart
Manage
Branding
Authentication
Certificates & secrets
API permissions
Expose an API
Owners
Manifest
Support + Troubleshooting
Troubleshooting
New support request
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 (1)
User.Read
Delegated
Sign in and read user profile
-
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
Grant consent
As an administrator, you can grant consent on behalf of all users in this directory. Granting admin consent for all users means that end users will not be shown a consent screen when using the application.
Grant admin consent for Contoso

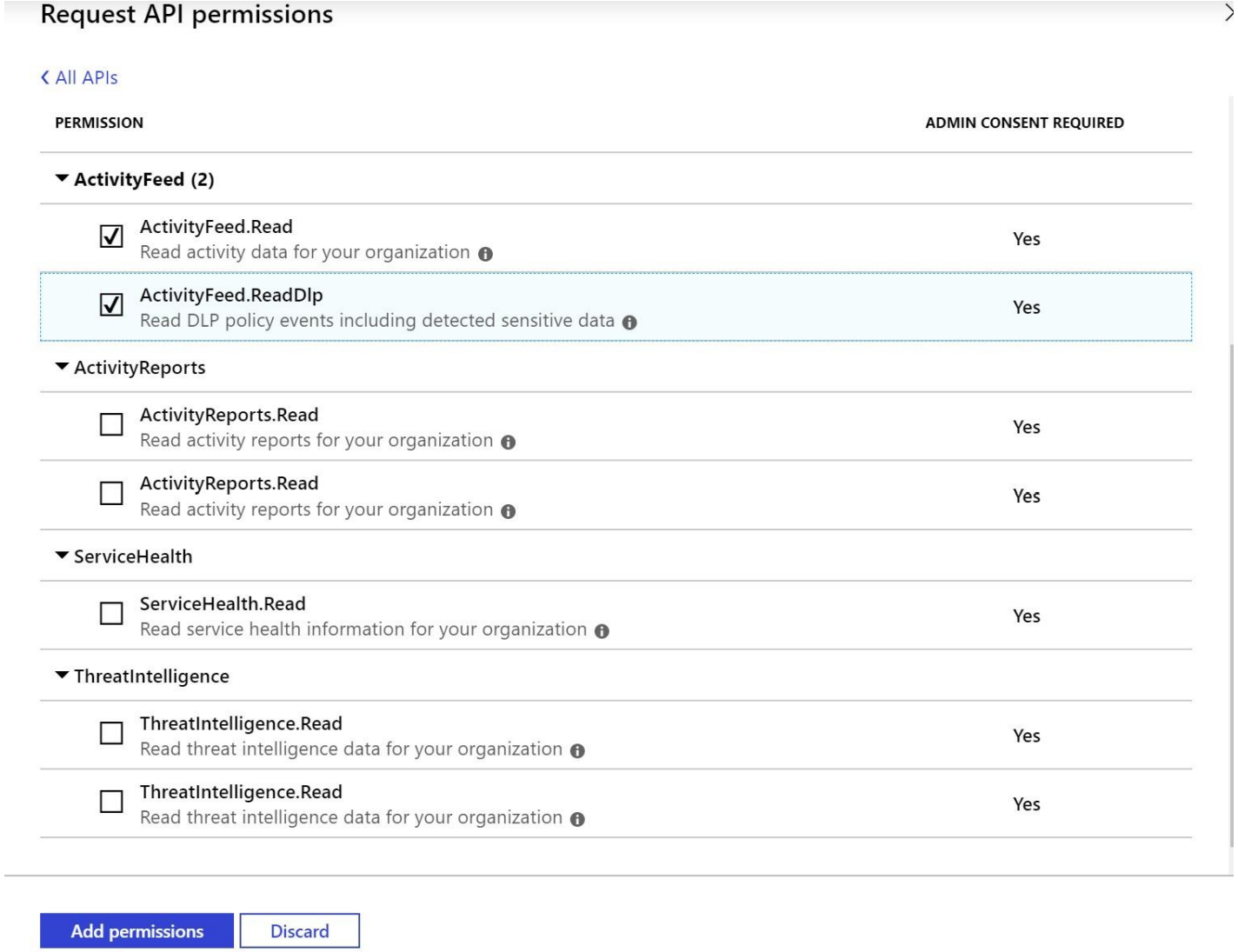
Demo


Request API permissions
Select an API
Microsoft APIs
APIs my organization uses
My APIs
Commonly used Microsoft APIs
Microsoft Graph
Take advantage of the tremendous amount of data in Office 365, Enterprise Mobility + Security, and Windows 10. Access Azure AD, Excel, Intune, Outlook/Exchange, OneDrive, OneNote, SharePoint, Planner, and more through a single endpoint.
Azure Data Catalog
Programmatic access to Data Catalog resources to register, annotate and search data assets
Azure Rights Management Services
Allow validated users to read and write protected content
Azure Service Management
Programmatic access to much of the functionality available through the Azure portal
Dynamics 365 Business Central
Programmatic access to data and functionality in Dynamics 365 Business Central
Flow Service
Embed flow templates and manage flows
Intune
Programmatic access to Intune data
Office 365 Management APIs
Retrieve information about user, admin, system, and policy actions and events from Office 365 and Azure AD activity
OneNote
Create and manage notes, lists, pictures, files, and more in OneNote notebooks
Power BI Service
Programmatic access to Dashboard resources such as Datasets, Tables, and Rows in Power BI
SharePoint
Skype for Business
Yammer

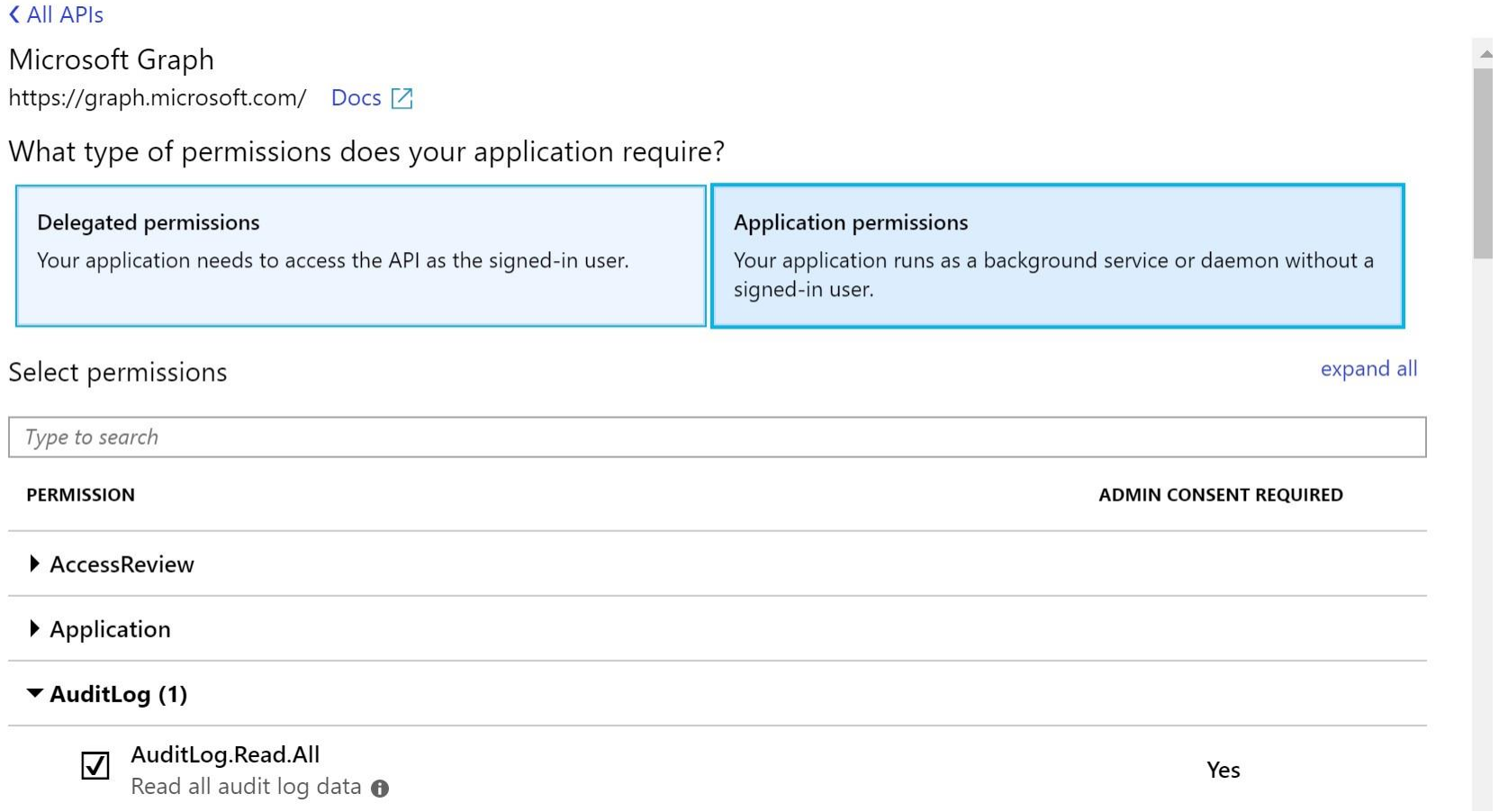
Demo


Request API permissions
All APIs
Office 365 Management APIs
https://manage.office.com/ Docs
What type of permissions does your application require?
Delegated permissions
Your application needs to access the API as the signed-in user.
Application permissions
Your application runs as a background service or daemon without a signed-in user.

Demo


Request API permissions
All APIs
PERMISSION
ADMIN CONSENT REQUIRED
ActivityFeed (2)
ActivityFeed.Read
Read activity data for your organization
Yes
ActivityFeed.ReadDlp
Read DLP policy events including detected sensitive data
Yes
ActivityReports
ActivityReports.Read
Read activity reports for your organization
Yes
ActivityReports.Read
Read activity reports for your organization
Yes
ServiceHealth
ServiceHealth.Read
Read service health information for your organization
Yes
ThreatIntelligence
ThreatIntelligence.Read
Read threat intelligence data for your organization
Yes
ThreatIntelligence.Read
Read threat intelligence data for your organization
Yes
Add permissions
Discard

Demo


All APIs
Microsoft Graph
https://graph.microsoft.com/ Docs
What type of permissions does your application require?
Delegated permissions
Your application needs to access the API as the signed-in user.
Application permissions
Your application runs as a background service or daemon without a signed-in user.
Select permissions
expand all
Type to search
PERMISSION
ADMIN CONSENT REQUIRED
AccessReview
Application
AuditLog (1)
AuditLog.Read.All
Read all audit log data
Yes

```
PS C:\Users\kirbogd> $certificate=(Get-ChildItem -Path Cert:\CurrentUser\My\ | where {$_.Subject -like "*AuditAPI*"})
PS C:\Users\kirbogd> $cert
PS C:\Users\kirbogd> $certificate


   PSParentPath: Microsoft.PowerShell.Security\Certificate::CurrentUser\My

Thumbprint                                Subject
----------                                -------
0C87CBE972222C99E23C2370F46B616DDA835244  CN=AuditAPIConnector


PS C:\Users\kirbogd> $ClientID = "0668dabc-a4fb-4e0d-ae1e-088413ca9d9c"
PS C:\Users\kirbogd> $tenantid="3f41e687-4cc6-48ac-a20a-ffa798046199"
PS C:\Users\kirbogd> $Authroot = "https://login.windows.net/$TenantID/oauth2/token"
PS C:\Users\kirbogd> $Authority = "$Authroot/$TenantID"
PS C:\Users\kirbogd> $ResourceAppID = "https://manage.office.com"
PS C:\Users\kirbogd> $AadModule = Get-Module -Name "AzureAD" -ListAvailable
PS C:\Users\kirbogd> $adal = Join-Path $AadModule.ModuleBase "Microsoft.IdentityModel.Clients.ActiveDirectory.dll"
PS C:\Users\kirbogd> $adalforms = Join-Path $AadModule.ModuleBase "Microsoft.IdentityModel.Clients.ActiveDirectory.Platform.dll"
PS C:\Users\kirbogd> [System.Reflection.Assembly]::LoadFrom($adal)

GAC    Version        Location
---    -------        --------
False  v4.0.30319     C:\Users\kirbogd\Documents\WindowsPowerShell\Modules\AzureAD\2.0.2.2\Microsoft.IdentityModel.Clients.ActiveDirectory.dll


PS C:\Users\kirbogd> [System.Reflection.Assembly]::LoadFrom($adalforms)

GAC    Version        Location
---    -------        --------
False  v4.0.30319     C:\Users\kirbogd\Documents\WindowsPowerShell\Modules\AzureAD\2.0.2.2\Microsoft.IdentityModel.Clients.ActiveDirectory.Platform.dll
```

```
PS C:\Users> $authContext = New-Object "Microsoft.IdentityModel.Clients.ActiveDirectory.AuthenticationContext" -ArgumentList $authority
PS C:\Users> $ClientCred = New-Object "Microsoft.IdentityModel.Clients.ActiveDirectory.ClientAssertionCertificate" -ArgumentList ($ClientId, $certificate)
PS C:\Users> $authReturn = $authContext.AcquireTokenAsync($ResourceAppID,$ClientCred)
PS C:\Users> $authResult = $authReturn.Result
PS C:\Users>  $authHeader = @{
>>
>>                              'Content-Type'='application/json'
>>
>>                              'Authorization'="Bearer " + $authResult.AccessToken
>>
>>                              'ExpiresOn'=$authResult.ExpiresOn
>>
>>                              }
PS C:\Users\kirbogd> $authHeader

Name                           Value
----                           -----
Authorization                  Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6IkhCeGw5bUFlNmd4YXZDa2NvT1UyVEhzRE5hMCIsImtpZCI6IkhCeGw5bUFlNmd4YXZDa2NvT1UyVEhzRE5hMCJ9.eyJhdWQiO...
Content-Type                   application/json
ExpiresOn                      12.05.2019 17:13:05 +00:00


PS C:\Users> $BaseURI = "https://manage.office.com/api/v1.0/3f41e687-4cc6-48ac-a20a-ffa798046199/activity/feed"
PS C:\Users> $SubURI = "subscriptions/list"
PS C:\Users> $URI = "$BaseURI/$SubURI"
PS C:\Users> $Return = Invoke-RestMethod -Uri $URI -Headers $authHeader -Method Get -Verbose
VERBOSE: GET https://manage.office.com/api/v1.0/3f41e687-4cc6-48ac-a20a-ffa798046199/activity/feed/subscriptions/list with 0-byte payload
VERBOSE: received 2-byte response of content type application/json; charset=utf-8
PS C:\Users> $Return
PS C:\Users> $SubURI = "subscriptions/start?ContentType=Audit.AzureActiveDirectory"
PS C:\Users> $URI = "$BaseURI/$SubURI"
PS C:\Users> $Return = Invoke-RestMethod -Uri $URI -Headers $authHeader -Method Post -Verbose
VERBOSE: POST https://manage.office.com/api/v1.0/3f41e687-4cc6-48ac-a20a-ffa798046199/activity/feed/subscriptions/start?ContentType=Audit.AzureActiveDirectory with 0-byte payload
VERBOSE: received 78-byte response of content type application/json; charset=utf-8
PS C:\Users> $Return

contentType                status  webhook
-----------                ------  -------
Audit.AzureActiveDirectory enabled
```

Demo

```
PS C:\Users\[redacted]> $SubURI = "subscriptions/list"
PS C:\Users\[redacted]> $URI = "$BaseURI/$SubURI"
PS C:\Users\[redacted]> $Return = Invoke-RestMethod -Uri $URI -Headers $authHeader -Method get -Verbose
VERBOSE: GET https://manage.office.com/api/v1.0/3f41e687-4cc6-48ac-a20a-ffa798046199/activity/feed/subscriptions/list with 0-byte payload
VERBOSE: received 342-byte response of content type application/json; charset=utf-8
PS C:\Users\[redacted]> $Return

contentType                status  webhook
-----------                ------  -------
Audit.AzureActiveDirectory enabled
Audit.Exchange             enabled
Audit.general              enabled
Audit.SharePoint           enabled
dlp.all                    enabled
```

```
PS C:\> $Cert = (get-childitem 'Cert:\CurrentUser\My\' | where {$_.Subject -like "*Audit*"})
PS C:\> $TenantID = "3f41e687-4cc6-48ac-a20a-ffa798046199"
PS C:\> $ClientID = "0668dabc-a4fb-4e0d-ae1e-088413ca9d9c"
PS C:\> $authroot = "https://login.microsoftonline.com"
PS C:\> $resourceAppIdURI = "https://graph.microsoft.com"
PS C:\> $authority = "$authroot/$TenantID"
PS C:\> $AadModule = Get-Module -Name "AzureAD" -ListAvailable
PS C:\> $adal = Join-Path $AadModule.ModuleBase "Microsoft.IdentityModel.Clients.ActiveDirectory.dll"
PS C:\> $adalforms = Join-Path $AadModule.ModuleBase "Microsoft.IdentityModel.Clients.ActiveDirectory.Platform.dll"
PS C:\> [System.Reflection.Assembly]::LoadFrom($adal) | Out-Null
PS C:\> [System.Reflection.Assembly]::LoadFrom($adalforms) | Out-Null
PS C:\> $authContext = New-Object "Microsoft.IdentityModel.Clients.ActiveDirectory.AuthenticationContext" -ArgumentList
$authority
PS C:\> $ClientCred = New-Object "Microsoft.IdentityModel.Clients.ActiveDirectory.ClientAssertionCertificate" -ArgumentL
ist ($ClientId, $Cert)
PS C:\> $authReturn = $authContext.AcquireTokenAsync($resourceAppIdURI,$ClientCred)
PS C:\> $authResult = $authReturn.Result
PS C:\>
>>                              $authHeader = @{
>>
>>                              'Content-Type'='application/json'
>>
>>                              'Authorization'="Bearer " + $authResult.AccessToken
>>
>>                              'ExpiresOn'=$authResult.ExpiresOn
>>
>>                              }
```

```
PS C:\> $BaseURI = "https://graph.microsoft.com/beta"
PS C:\> $SubURI = "auditLogs/signIns"
PS C:\> $URI = "$BaseURI/$SubURI"
PS C:\> $Return = Invoke-RestMethod -Uri $URI -Headers $authHeader -Method Get -Verbose
VERBOSE: GET https://graph.microsoft.com/beta/auditLogs/signIns with 0-byte payload
VERBOSE: received -1-byte response of content type
application/json;odata.metadata=minimal;odata.streaming=true;IEEE754Compatible=false;charset=utf-8
PS C:\> $Return

@odata.context                                                  value
--------------                                                  -----
https://graph.microsoft.com/beta/$metadata#auditLogs/signIns {@{id=dfed8335-27e3-4f5f-95f0-555c6a7d0100; createdDate...
```

```
PS C:\> $Return.value[0]


id                                : dfed8335-27e3-4f5f-95f0-555c6a7d0100
createdDateTime                   : 2019-05-15T12:43:52.8248009Z
userDisplayName                   : MOD Administrator
userPrincipalName                 : admin@m365x413039.onmicrosoft.com
userId                            : 3780bb02-6858-49e8-9a46-2b426c7f0519
appId                             : c44b4083-3bb0-49c1-b47d-974e53cbdf3c
appDisplayName                    : Azure Portal
ipAddress                         : 167.220.196.13
clientAppUsed                     : Browser
correlationId                     : 0c91c07c-e0b1-4626-8514-f92bc84b6b69
conditionalAccessStatus           :
originalRequestId                 :
isInteractive                     : True
tokenIssuerName                   :
tokenIssuerType                   : AzureAD
processingTimeInMilliseconds      : 0
riskDetail                        : none
riskLevelAggregated               : none
riskLevelDuringSignIn             : none
riskState                         : none
riskEventTypes                    : {}
resourceDisplayName               : windows azure service management api
resourceId                        : 797f4846-ba00-4fd7-ba43-dac1f8f63013
authenticationMethodsUsed         : {}
mfaDetail                         :
status                            : @{errorCode=0; failureReason=; additionalDetails=}
deviceDetail                      : @{deviceId=; displayName=; operatingSystem=Windows 10; browser=Chrome 76.0.3782
                                    ompliant=; isManaged=; trustType=}
location                          : @{city=Sonning; state=Wokingham; countryOrRegion=GB; geoCoordinates=}
appliedConditionalAccessPolicies : {}
authenticationProcessingDetails  : {}
```

Using data in investigation

Investigating incident

During routine work, administrator founds out a Journaling rule forwarding all his mail to external mailbox.

It has not existed a week ago and we hope to get details (or we regularly download events and have them locally)

Hunting begins.

Positive Hack Days

Getting all events for last 7 days and saving it to XML file for backup

```
$Return = ""
$Content = ""
$source = ("Audit.AzureActiveDirectory","Audit.Exchange","Audit.SharePoint","Audit.General","DLP.All")
$result = @()
$BaseURI = "https://manage.office.com/api/v1.0/$TenantID/activity/feed"
$source | ForEach-Object {
    for ($i = 0; $i -lt "6";$i++)
        {
        $startday = (Get-Date (Get-Date).AddDays(-$i-1) -UFormat %d)
        $endday = (Get-Date (Get-Date).AddDays(-$i) -UFormat %d)
        $SubURI = "subscriptions/content?contentType="+$_+"&startTime=2019-05-"+$startday+"&endTime=2019-05-"+$endday
        $URI = "$BaseURI/$SubURI"
        $Return = Invoke-RestMethod -Uri $URI -Headers $authHeader -Method Get
        if ($Return)
            { $Return | ForEach-Object {
                $Content = Invoke-RestMethod -Uri $_.contentUri -Headers $authHeader -Method get -Verbose
                $result = $result+$Content
                }
            }
        }
    }

    $result.Count
    $result | Export-Clixml -Path C:\Distr\PHDays.xml
```

First step

$result | where {$_.operation –like “*Journl*”}

Who is NewUser ? We do not have him in organization…
What else has he done?

```
PS C:\Distr> $result | where {$_.operation -like "*Journ*"}


CreationTime      : 2019-05-13T08:16:15
Id                : 49ae8c1e-39a1-4697-6f01-08d6d77b44fa
Operation         : New-JournalRule
OrganizationId    : 3f41e687-4cc6-48ac-a20a-ffa798046199
RecordType        : 1
ResultStatus      : True
UserKey           : 1003200048049F65
UserType          : 2
Version           : 1
Workload          : Exchange
ClientIP          : 191.232.238.156:40504
ObjectId          :
UserId            : NewUser@m365x413039.onmicrosoft.com
AppId             :
ClientAppId       :
ExternalAccess    : False
OrganizationName  : M365x413039.onmicrosoft.com
OriginatingServer : HE1P18901MB0076 (15.20.1878.000)
Parameters        : {@{Name=Recipient; Value=admin@m365x413039.onmicrosoft.com}, @{Name=JournalEmailAddress; Value=EMSTest}, @{Name=Name; Value=Rule1}}
SessionId         : 67381be3-fe74-47d6-b657-88bdd091a547
```

What was done during the session?

$result | where {$_.SessionID –eq “” 67381be3-fe74-47d6-b657-88bdd091a547}

```
PS C:\Distr> $result | where {$_.SessionID -eq "67381be3-fe74-47d6-b657-88bdd091a547"}


CreationTime       : 2019-05-13T08:16:15
Id                 : 49ae8c1e-39a1-4697-6f01-08d6d77b44fa
Operation          : New-JournalRule
OrganizationId     : 3f41e687-4cc6-48ac-a20a-ffa798046199
RecordType         : 1
ResultStatus       : True
UserKey            : 1003200048049F65
UserType           : 2
Version            : 1
Workload           : Exchange
ClientIP           : 191.232.238.156:40504
ObjectId           :
UserId             : NewUser@m365x413039.onmicrosoft.com
AppId              :
ClientAppId        :
ExternalAccess     : False
OrganizationName   : M365x413039.onmicrosoft.com
OriginatingServer  : HE1P18901MB0076 (15.20.1878.000)
Parameters         : {@{Name=Recipient; Value=admin@m365x413039.onmicrosoft.com}, @{Name=JournalEmailAddress; Value=EMSTest}, @{Name=Name; Value=Rule1}}
SessionId          : 67381be3-fe74-47d6-b657-88bdd091a547

CreationTime       : 2019-05-13T08:15:50
Id                 : dc3baffd-e11f-45ae-06d5-08d6d77b3626
Operation          : New-MailContact
OrganizationId     : 3f41e687-4cc6-48ac-a20a-ffa798046199
RecordType         : 1
ResultStatus       : True
UserKey            : 1003200048049F65
UserType           : 2
Version            : 1
Workload           : Exchange
ClientIP           : 191.232.238.156:40504
ObjectId           : EURP189A001.PROD.OUTLOOK.COM/Microsoft Exchange Hosted Organizations/M365x413039.onmicrosoft.com/EMSTest
UserId             : NewUser@m365x413039.onmicrosoft.com
AppId              :
ClientAppId        :
ExternalAccess     : False
OrganizationName   : M365x413039.onmicrosoft.com
OriginatingServer  : HE1P18901MB0076 (15.20.1878.000)
Parameters         : {@{Name=DisplayName; Value=EMStest}, @{Name=ExternalEmailAddress; Value=smtp:emscloudonlytest@outlook.com}, @{Name=Name; Value=EMSTest}}
```

What else has this user done?

```
 $result | where {$_.UserID -eq
"NewUser@m365x413039.onmicrosoft.com"}
```

```
PS C:\Distr> $result | where {$_.UserID -eq "NewUser@m365x413039.onmicrosoft.com"} | select creationtime,operation,ClientIP | Sort-Object -Property CreationTime

CreationTime        Operation           ClientIP
------------        ---------           --------
2019-05-13T07:20:00 UserLoginFailed     191.232.238.156
2019-05-13T07:20:10 UserLoginFailed     191.232.238.156
2019-05-13T07:21:11 UserLoginFailed     191.232.238.156
2019-05-13T07:26:10 UserLoggedIn        191.232.238.156
2019-05-13T07:26:22 Update user.        <null>
2019-05-13T07:26:23 UserLoggedIn        191.232.238.156
2019-05-13T07:26:28 UserLoggedIn        191.232.238.156
2019-05-13T07:26:31 UserLoggedIn        191.232.238.156
2019-05-13T07:26:31 UserLoggedIn        191.232.238.156
2019-05-13T07:26:31 UserLoggedIn        191.232.238.156
2019-05-13T07:26:32 UserLoggedIn        191.232.238.156
2019-05-13T07:26:42 UserLoggedIn        191.232.238.156
2019-05-13T07:26:44 UserLoggedIn        191.232.238.156
2019-05-13T07:26:44 UserLoggedIn        191.232.238.156
2019-05-13T07:27:13 UserLoggedIn        191.232.238.156
2019-05-13T07:30:57 UserLoggedIn        191.232.238.156
2019-05-13T08:04:21 Set-TransportConfig 191.232.238.156:47828
2019-05-13T08:15:36 UserLoggedIn        191.232.238.156
2019-05-13T08:15:50 New-MailContact     191.232.238.156:40504
2019-05-13T08:16:15 New-JournalRule     191.232.238.156:40504
```

Who has deleted the user?

```
 ($result | where {($_.Operation -like "*Del*") -and
($_.Objectid -like "*NewUser@m365x413039*")} )
```

```
PS C:\Distr> ($result | where {($_.Operation -like "*Del*") -and ($_.Objectid -like "*NewUser@m365x413039*")} )


CreationTime                  : 2019-05-13T08:36:31
Id                            : 6aed79bb-1cb9-4e0a-bb74-cd21b6050adc
Operation                     : Delete user.
OrganizationId                : 3f41e687-4cc6-48ac-a20a-ffa798046199
RecordType                    : 8
ResultStatus                  : Success
UserKey                       : 1003200047F4BC37@m365x413039.onmicrosoft.com
UserType                      : 0
Version                       : 1
Workload                      : AzureActiveDirectory
ClientIP                      : <null>
ObjectId                      : 72234e7b0ba3486691cc0c607747b4a0NewUser@m365x413039.onmicrosoft.com
UserId                        : DemoAdm@m365x413039.onmicrosoft.com
AzureActiveDirectoryEventType : 1
ExtendedProperties            : {@{Name=resultType; Value=Success}, @{Name=auditEventCategory; Value=UserManagement}, @{Name=nCloud; Value=<null>}, @{Name=actor
                                ContextId; Value=3f41e687-4cc6-48ac-a20a-ffa798046199}...}
ModifiedProperties            : {@{Name=Is Hard Deleted; NewValue=False; OldValue=}}
Actor                         : {@{ID=DemoAdm@m365x413039.onmicrosoft.com; Type=5}, @{ID=1003200047F4BC37; Type=3}, @{ID=User_510eafcf-364d-405b-ab43-a32b0769fd
                                a4; Type=2}, @{ID=510eafcf-364d-405b-ab43-a32b0769fda4; Type=2}...}
ActorContextId                : 3f41e687-4cc6-48ac-a20a-ffa798046199
ActorIpAddress                : <null>
SupportTicketId               :
Target                        : {@{ID=User_72234e7b-0ba3-4866-91cc-0c607747b4a0; Type=2}, @{ID=72234e7b-0ba3-4866-91cc-0c607747b4a0; Type=2}, @{ID=User; Type=2}
                                , @{ID=72234e7b0ba3486691cc0c607747b4a0NewUser@m365x413039.onmicrosoft.com; Type=5}...}
TargetContextId               : 3f41e687-4cc6-48ac-a20a-ffa798046199
```

Who has created?

```
($result I where { ($_. operation -like "*add user*") -and ($_.objectid -like "*NewUser*")} )
```

```
($result | where {($_.operation -like "*update*") -and ($_.objectid -like
"*NewUser@m365x413039*")})| select creationtime,ObjectID,ModifiedProperties
```

```
$result | where {($_. Operation –like “*Member*”)  -and($_.objectID –eq
“NewUser@m365x413039.onmicrosoft.com”) } | sort-object –Property CreationTime -
Descending
```

```
PS C:\Distr> ($result | where {($_.Operation -like "*add user*") -and ($_.objectid -like "*NewUser*")} )


CreationTime                 : 2019-05-13T07:05:12
Id                           : e9549fe5-0200-4ccd-b7df-59303e79f522
Operation                    : Add user.
OrganizationId               : 3f41e687-4cc6-48ac-a20a-ffa798046199
RecordType                   : 8
ResultStatus                 : Success
UserKey                      : 1003200047F4BC37@m365x413039.onmicrosoft.com
UserType                     : 0
Version                      : 1
Workload                     : AzureActiveDirectory
ClientIP                     : <null>
ObjectId                     : NewUser@m365x413039.onmicrosoft.com
UserId                       : DemoAdm@m365x413039.onmicrosoft.com
AzureActiveDirectoryEventType : 1
ExtendedProperties           : {@{Name=actorContextId; Value=3f41e687-4cc6-48ac-a20a-ffa798046199}, @{Name=actorObjectId; Value=510eafcf-364d-405b-ab43-a32b076
                               9fda4}, @{Name=actorObjectClass; Value=User}, @{Name=actorUPN; Value=DemoAdm@m365x413039.onmicrosoft.com}...}
ModifiedProperties           : {@{Name=AccountEnabled; NewValue=[
                                 true
                               ]; OldValue=[]}, @{Name=StsRefreshTokensValidFrom; NewValue=[
                                 "2019-05-13T07:05:12Z"
                               ]; OldValue=[]}, @{Name=UserPrincipalName; NewValue=[
                                 "NewUser@m365x413039.onmicrosoft.com"
                               ]; OldValue=[]}, @{Name=UserType; NewValue=[
                                 "Member"
                               ]; OldValue=[]}...}
Actor                        : {@{ID=DemoAdm@m365x413039.onmicrosoft.com; Type=5}, @{ID=1003200047F4BC37; Type=3}, @{ID=User_510eafcf-364d-405b-ab43-a32b0769fd
                               a4; Type=2}, @{ID=510eafcf-364d-405b-ab43-a32b0769fda4; Type=2}...}
ActorContextId               : 3f41e687-4cc6-48ac-a20a-ffa798046199
ActorIpAddress               : <null>
SupportTicketId              :
Target                       : {@{ID=User_72234e7b-0ba3-4866-91cc-0c607747b4a0; Type=2}, @{ID=72234e7b-0ba3-4866-91cc-0c607747b4a0; Type=2}, @{ID=User; Type=2}
                               , @{ID=NewUser@m365x413039.onmicrosoft.com; Type=5}...}
TargetContextId              : 3f41e687-4cc6-48ac-a20a-ffa798046199
```

```
PS C:\Distr> ($result | where {($_.Operation -like "*update*") -and ($_.Objectid -like "*NewUser@m365x413039*")} ) | select creationtime,ObjectID,ModifiedPropertie

CreationTime        ObjectId                          ModifiedProperties
------------        --------                          ------------------
2019-05-13T07:25:41 NewUser@m365x413039.onmicrosoft.com {@{Name=MethodExecutionResult.; NewValue=Microsoft.Online.Workflows.WeakPasswordException; OldValue=}}
2019-05-13T07:25:41 NewUser@m365x413039.onmicrosoft.com {@{Name=TargetId.UserType; NewValue=Member; OldValue=}}
2019-05-13T07:25:52 NewUser@m365x413039.onmicrosoft.com {}
2019-05-13T07:25:52 NewUser@m365x413039.onmicrosoft.com {@{Name=TargetId.UserType; NewValue=Member; OldValue=}}
2019-05-13T07:26:22 NewUser@m365x413039.onmicrosoft.com {@{Name=Included Updated Properties; NewValue=; OldValue=}, @{Name=TargetId.UserType; NewValue=Member...
```

```
PS C:\Distr> $result | where  {($_.Operation -like "*Member*") -and ($_.ObjectId -eq "NewUser@m365x413039.onmicrosoft.com")} | Sort-Object -Property CreationTime -D


CreationTime                  : 2019-05-13T08:36:05
Id                            : 22195054-c5bd-4942-b7ec-f15e6a1c6eb7
Operation                     : Remove member from role.
OrganizationId                : 3f41e687-4cc6-48ac-a20a-ffa798046199
RecordType                    : 8
ResultStatus                  : Success
UserKey                       : 1003200047F4BC37@m365x413039.onmicrosoft.com
UserType                      : 0
Version                       : 1
Workload                      : AzureActiveDirectory
ClientIP                      : <null>
ObjectId                      : NewUser@m365x413039.onmicrosoft.com
UserId                        : DemoAdm@m365x413039.onmicrosoft.com
AzureActiveDirectoryEventType : 1
ExtendedProperties            : {@{Name=resultType; Value=Success}, @{Name=auditEventCategory; Value=RoleManagement}, @{Name=nCloud; Value=<null>}, @{Name=actor
                                ContextId; Value=3f41e687-4cc6-48ac-a20a-ffa798046199}...}
ModifiedProperties            : {@{Name=Role.ObjectID; NewValue=c31ac268-7a86-4f83-a51e-a9b65077cea1; OldValue=}, @{Name=Role.DisplayName; NewValue=Company Admi
                                nistrator; OldValue=}, @{Name=Role.TemplateId; NewValue=62e90394-69f5-4237-9190-012177145e10; OldValue=}, @{Name=Role.WellKnownO
                                bjectName; NewValue=TenantAdmins; OldValue=}}
Actor                         : {@{ID=DemoAdm@m365x413039.onmicrosoft.com; Type=5}, @{ID=1003200047F4BC37; Type=3}, @{ID=User_510eafcf-364d-405b-ab43-a32b0769fd
                                a4; Type=2}, @{ID=510eafcf-364d-405b-ab43-a32b0769fda4; Type=2}...}
ActorContextId                : 3f41e687-4cc6-48ac-a20a-ffa798046199
ActorIpAddress                : <null>
SupportTicketId               :
Target                        : {@{ID=User_72234e7b-0ba3-4866-91cc-0c607747b4a0; Type=2}, @{ID=72234e7b-0ba3-4866-91cc-0c607747b4a0; Type=2}, @{ID=User; Type=2}
                                , @{ID=NewUser@m365x413039.onmicrosoft.com; Type=5}...}
TargetContextId               : 3f41e687-4cc6-48ac-a20a-ffa798046199

CreationTime                  : 2019-05-13T07:16:35
Id                            : 6d279370-ca15-4d26-a1d6-1bbf49b42cd5
Operation                     : Add member to role.
OrganizationId                : 3f41e687-4cc6-48ac-a20a-ffa798046199
RecordType                    : 8
ResultStatus                  : Success
UserKey                       : 1003200047F4BC37@m365x413039.onmicrosoft.com
UserType                      : 0
```

So we have DemoAdm compromised

```
 ($result | where {($_.Operation -like "*UserLoggedIn*") -and
($_.Userid -like "*demoadm*") -and ($_.CreationTime -like
"2019-05-13T*")} )| select
CreationTime,ClientIP,ResultStatus
```

CreationTime	ClientIP	ResultStatus
2019-05-13T04:46:57	109.195.105.130	Succeeded
2019-05-13T06:44:54	191.232.238.156	Succeeded
2019-05-13T06:46:17	191.232.238.156	Succeeded
2019-05-13T06:40:24	191.232.238.156	Succeeded
2019-05-13T07:09:12	109.195.105.130	Succeeded
2019-05-13T08:14:42	109.195.105.134	Succeeded

Who else?

```
($result | where {$_.ActorIPAddress -eq "191.232.238.156"})|
Sort-Object -Property CreationTime | select
creationtime,Operation,ResultStatus,UserID
```

CreationTime	Operation	ResultStatus	UserId
2019-05-13T06:10:31	UserLoginFailed	Failed	irvins@m365x413039.onmicrosoft.com
2019-05-13T06:10:38	UserLoginFailed	Failed	irvins@m365x413039.onmicrosoft.com
2019-05-13T06:10:49	UserLoginFailed	Failed	irvins@m365x413039.onmicrosoft.com
2019-05-13T06:10:57	UserLoginFailed	Failed	irvins@m365x413039.onmicrosoft.com
2019-05-13T06:11:09	UserLoggedIn	Succeeded	IrvinS@M365x413039.OnMicrosoft.com
2019-05-13T06:32:04	UserLoggedIn	Succeeded	IrvinS@M365x413039.OnMicrosoft.com
2019-05-13T06:32:24	UserLoggedIn	Succeeded	IrvinS@M365x413039.OnMicrosoft.com
2019-05-13T06:32:24	UserLoggedIn	Succeeded	IrvinS@M365x413039.OnMicrosoft.com
2019-05-13T06:32:24	UserLoggedIn	Succeeded	IrvinS@M365x413039.OnMicrosoft.com
2019-05-13T06:32:29	UserLoggedIn	Succeeded	IrvinS@M365x413039.OnMicrosoft.com
2019-05-13T06:32:32	UserLoggedIn	Succeeded	IrvinS@M365x413039.OnMicrosoft.com
2019-05-13T06:32:36	UserLoggedIn	Succeeded	IrvinS@M365x413039.OnMicrosoft.com
2019-05-13T06:32:39	UserLoggedIn	Succeeded	IrvinS@M365x413039.OnMicrosoft.com
2019-05-13T06:32:42	UserLoggedIn	Succeeded	IrvinS@M365x413039.OnMicrosoft.com
2019-05-13T06:32:45	UserLoggedIn	Succeeded	IrvinS@M365x413039.OnMicrosoft.com
2019-05-13T06:32:47	UserLoggedIn	Succeeded	IrvinS@M365x413039.OnMicrosoft.com
2019-05-13T06:34:22	UserLoggedIn	Succeeded	IrvinS@M365x413039.OnMicrosoft.com
2019-05-13T06:40:24	UserLoggedIn	Succeeded	DemoAdm@m365x413039.onmicrosoft.com
2019-05-13T06:44:54	UserLoggedIn	Succeeded	DemoAdm@m365x413039.onmicrosoft.com
2019-05-13T06:46:17	UserLoggedIn	Succeeded	DemoAdm@m365x413039.onmicrosoft.com
2019-05-13T07:20:00	UserLoginFailed	Failed	newuser@m365x413039.onmicrosoft.com
2019-05-13T07:20:10	UserLoginFailed	Failed	newuser@m365x413039.onmicrosoft.com
2019-05-13T07:21:11	UserLoginFailed	Failed	newuser@m365x413039.onmicrosoft.com
2019-05-13T07:26:10	UserLoggedIn	Succeeded	NewUser@m365x413039.onmicrosoft.com
2019-05-13T07:26:23	UserLoggedIn	Succeeded	NewUser@m365x413039.onmicrosoft.com
2019-05-13T07:26:28	UserLoggedIn	Succeeded	NewUser@m365x413039.onmicrosoft.com
2019-05-13T07:26:31	UserLoggedIn	Succeeded	NewUser@m365x413039.onmicrosoft.com
2019-05-13T07:26:31	UserLoggedIn	Succeeded	NewUser@m365x413039.onmicrosoft.com
2019-05-13T07:26:31	UserLoggedIn	Succeeded	NewUser@m365x413039.onmicrosoft.com
2019-05-13T07:26:32	UserLoggedIn	Succeeded	NewUser@m365x413039.onmicrosoft.com
2019-05-13T07:26:42	UserLoggedIn	Succeeded	NewUser@m365x413039.onmicrosoft.com
2019-05-13T07:26:44	UserLoggedIn	Succeeded	NewUser@m365x413039.onmicrosoft.com
2019-05-13T07:26:44	UserLoggedIn	Succeeded	NewUser@m365x413039.onmicrosoft.com
2019-05-13T07:27:13	UserLoggedIn	Succeeded	NewUser@m365x413039.onmicrosoft.com
2019-05-13T07:30:57	UserLoggedIn	Succeeded	NewUser@m365x413039.onmicrosoft.com
2019-05-13T08:15:36	UserLoggedIn	Succeeded	NewUser@m365x413039.onmicrosoft.com

What additional actions have been done?

```
($result | where {($_.ActorIPAddress -eq "$IPaddress") -or ($_.ClientIP -eq "$IPaddress")}) | Sort-Object -Property CreationTime | ft Operation,Workload,UserID,id
```

```
$result | where {$_.id -eq "523a0cd9-ece1-45d2-1973-08d6d76d0a52"}
```

```
$result | where {$_.objectID -eq "https://m365x413039-my.sharepoint.com/personal/irvins_m365x413039_onmicrosoft_com/Documents/file.com.txt"}
```

```
PS C:\Distr> ($result | where {($_.ActorIPAddress -eq "$IPaddress") -or ($_.ClientIP -eq "$IPaddress")}) | Sort-

Operation       Workload            UserId                          Id
---------       --------            ------                          --
UserLoginFailed AzureActiveDirectory irvins@m365x413039.onmicrosoft.com 662a74cd-2b0e-4c24-a44f-f2873f0eed60
UserLoginFailed AzureActiveDirectory irvins@m365x413039.onmicrosoft.com 505bbeab-ea55-4a9e-9440-c4e8de338192
UserLoginFailed AzureActiveDirectory irvins@m365x413039.onmicrosoft.com 3f49d915-f387-4b62-8cae-dd461c79553a
UserLoginFailed AzureActiveDirectory irvins@m365x413039.onmicrosoft.com 307d3f70-debf-48f3-854f-dacd0392e910
UserLoggedIn    AzureActiveDirectory IrvinS@M365x413039.OnMicrosoft.com b5f5b5aa-6d41-41a3-a893-fb484714ed5d
UserLoggedIn    AzureActiveDirectory IrvinS@M365x413039.OnMicrosoft.com 163bedd1-fa05-4a9b-ad9d-311cab6fa75c
UserLoggedIn    AzureActiveDirectory IrvinS@M365x413039.OnMicrosoft.com 54d015e7-5112-48e6-973c-18e9aa44046e
UserLoggedIn    AzureActiveDirectory IrvinS@M365x413039.OnMicrosoft.com 64f743a3-faf7-4ef4-b3aa-d464762e4a4c
UserLoggedIn    AzureActiveDirectory IrvinS@M365x413039.OnMicrosoft.com 2f3fcf85-2e35-46a3-ac67-c2c0b2afeb85
UserLoggedIn    AzureActiveDirectory IrvinS@M365x413039.OnMicrosoft.com b2614f72-e2ad-4d29-9a52-ccc2080ce6be
UserLoggedIn    AzureActiveDirectory IrvinS@M365x413039.OnMicrosoft.com fc8078db-1d5d-4a33-b258-5ef34832e4ee
UserLoggedIn    AzureActiveDirectory IrvinS@M365x413039.OnMicrosoft.com 33e5b2c9-3feb-4e79-aa16-1c1ea9a1ba0c
UserLoggedIn    AzureActiveDirectory IrvinS@M365x413039.OnMicrosoft.com 19b35d28-1413-469d-b1cd-a7d0ce586095
UserLoggedIn    AzureActiveDirectory IrvinS@M365x413039.OnMicrosoft.com 86180e48-c359-46d6-b4cf-36ee370843dd
UserLoggedIn    AzureActiveDirectory IrvinS@M365x413039.OnMicrosoft.com 651bd372-bfd7-4ba5-9261-82986a37b993
UserLoggedIn    AzureActiveDirectory IrvinS@M365x413039.OnMicrosoft.com 796d12d8-3666-486e-8295-0a2768647132
UserLoggedIn    AzureActiveDirectory IrvinS@M365x413039.OnMicrosoft.com a3491c43-2cbf-4ea7-90ed-b1cf4f03991c
PageViewed      OneDrive            irvins@m365x413039.onmicrosoft.com 523a0cd9-ece1-45d2-1973-08d6d76d0a52
PageViewed      OneDrive            irvins@m365x413039.onmicrosoft.com 523a0cd9-ece1-45d2-1973-08d6d76d0a52
FileAccessed    OneDrive            irvins@m365x413039.onmicrosoft.com 3318f661-d15a-49c5-c8c8-08d6d76d0f7b
FileAccessed    OneDrive            irvins@m365x413039.onmicrosoft.com c770e666-8c67-4503-8dcd-08d6d76d0f80
FileAccessed    OneDrive            irvins@m365x413039.onmicrosoft.com f4227297-34f2-45f8-525c-08d6d76d0f85
FileAccessed    OneDrive            irvins@m365x413039.onmicrosoft.com f4227297-34f2-45f8-525c-08d6d76d0f85
FileAccessed    OneDrive            irvins@m365x413039.onmicrosoft.com 3318f661-d15a-49c5-c8c8-08d6d76d0f7b
FileAccessed    OneDrive            irvins@m365x413039.onmicrosoft.com c770e666-8c67-4503-8dcd-08d6d76d0f80
FileDownloaded  OneDrive            irvins@m365x413039.onmicrosoft.com e64707d8-3e20-49e9-c181-08d6d76d21ff
FileDownloaded  OneDrive            irvins@m365x413039.onmicrosoft.com 0e03e9e3-ca92-4c0e-87e2-08d6d76d22d8
FileDownloaded  OneDrive            irvins@m365x413039.onmicrosoft.com edcf74eb-c6d7-4ed1-cfd8-08d6d76d228e
FileDownloaded  OneDrive            irvins@m365x413039.onmicrosoft.com 0a5cdd57-b3f0-41f8-e827-08d6d76d24dd
FileDownloaded  OneDrive            irvins@m365x413039.onmicrosoft.com b4d4e3b4-75c6-42fc-3933-08d6d76d248f
FileDownloaded  OneDrive            irvins@m365x413039.onmicrosoft.com 421bc10b-8b9b-4ed8-915a-08d6d76d24b5
FileDownloaded  OneDrive            irvins@m365x413039.onmicrosoft.com 67da295c-5d8d-4977-6d1d-08d6d76d2470
FileDownloaded  OneDrive            irvins@m365x413039.onmicrosoft.com 0c771fbc-e4c5-4293-c8f7-08d6d76d250d
FileDownloaded  OneDrive            irvins@m365x413039.onmicrosoft.com f741e1fb-4fb5-4ee3-573b-08d6d76d256f
FileDownloaded  OneDrive            irvins@m365x413039.onmicrosoft.com ec0bca4f-47c8-447b-e5c5-08d6d76d260c
FileDownloaded  OneDrive            irvins@m365x413039.onmicrosoft.com 76674b3b-e054-4b5d-2662-08d6d76d258c
FileDownloaded  OneDrive            irvins@m365x413039.onmicrosoft.com 11ddb690-5d02-45e4-8e59-08d6d76d25e6
FileDownloaded  OneDrive            irvins@m365x413039.onmicrosoft.com a1e36055-7d65-4b67-4d2e-08d6d76d25d1
FileDownloaded  OneDrive            irvins@m365x413039.onmicrosoft.com 584c4b5d-cd8e-4609-61d8-08d6d76d25a6
FileDownloaded  SharePoint          irvins@m365x413039.onmicrosoft.com e02e1d70-6172-4cc7-84b4-08d6d76d4dde
FileDownloaded  SharePoint          irvins@m365x413039.onmicrosoft.com e02e1d70-6172-4cc7-84b4-08d6d76d4dde
FileDownloaded  SharePoint          irvins@m365x413039.onmicrosoft.com 83deeb34-0c6e-4a69-f87b-08d6d76d4e12
FileDownloaded  SharePoint          irvins@m365x413039.onmicrosoft.com 83deeb34-0c6e-4a69-f87b-08d6d76d4e12
FileDownloaded  SharePoint          irvins@m365x413039.onmicrosoft.com 423e1f9a-772b-406b-18fd-08d6d76d4eb5
```

What additional actions have been done?

```
PS C:\Distr> $result | where {$_.id -eq "523a0cd9-ece1-45d2-1973-08d6d76d0a52"}


CreationTime     : 2019-05-13T06:34:24
Id               : 523a0cd9-ece1-45d2-1973-08d6d76d0a52
Operation        : PageViewed
OrganizationId   : 3f41e687-4cc6-48ac-a20a-ffa798046199
RecordType       : 4
UserKey          : i:0h.f|membership|1003200045f60e72@live.com
UserType         : 0
Version          : 1
Workload         : OneDrive
ClientIP         : 191.232.238.156
ObjectId         : https://m365x413039-my.sharepoint.com/personal/irvins_m365x413039_onmicrosoft_com/_layouts/15/onedrive.aspx
UserId           : irvins@m365x413039.onmicrosoft.com
CorrelationId    : 3121dc9e-90a7-0000-9926-790c5a3fad7d
CustomUniqueId   : True
EventSource      : SharePoint
ItemType         : Page
ListItemUniqueId : 59a8433d-9bb8-cfef-6edc-4c0fc8b86875
Site             : a5a8b9b8-7d7f-49f4-9f0e-72a87296662a
UserAgent        : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140 Safari/537.36 Edge/18.17763
WebId            : 6b5bdf30-2c3d-4547-be42-d7b2222f8c1a
```

What additional actions have been done?

```
PS C:\Distr> $result | where {$_.id -eq "3318f661-d15a-49c5-c8c8-08d6d76d0f7b"}


CreationTime         : 2019-05-13T06:34:33
Id                   : 3318f661-d15a-49c5-c8c8-08d6d76d0f7b
Operation            : FileAccessed
OrganizationId       : 3f41e687-4cc6-48ac-a20a-ffa798046199
RecordType           : 6
UserKey              : i:0h.f|membership|1003200045f60e72@live.com
UserType             : 0
Version              : 1
Workload             : OneDrive
ClientIP             : 191.232.238.156
ObjectId             : https://m365x413039-my.sharepoint.com/personal/irvins_m365x413039_onmicrosoft_com/Documents/Forms/Upload.aspx
UserId               : irvins@m365x413039.onmicrosoft.com
CorrelationId        : 3321dc9e-e0c7-0000-99c2-159aad57fa9b
EventSource          : SharePoint
ItemType             : File
ListId               : d6961939-758a-4450-bc1a-408ae6bd9894
ListItemUniqueId     : 6798cd43-a1a3-44ef-a388-6df9746220b1
Site                 : a5a8b9b8-7d7f-49f4-9f0e-72a87296662a
UserAgent            : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140 Safari/537.36 Edge/18.17763
WebId                : 6b5bdf30-2c3d-4547-be42-d7b2222f8c1a
SourceFileExtension  : aspx
SiteUrl              : https://m365x413039-my.sharepoint.com/personal/irvins_m365x413039_onmicrosoft_com/
SourceFileName       : Upload.aspx
SourceRelativeUrl    : Documents/Forms
```

What additional actions have been done?

```
PS C:\Distr> $result | where {$_.id -eq "16192541-8a49-4b79-aa86-08d6d76d74a8"}


CreationTime        : 2019-05-13T06:37:22
Id                  : 16192541-8a49-4b79-aa86-08d6d76d74a8
Operation           : FileUploaded
OrganizationId      : 3f41e687-4cc6-48ac-a20a-ffa798046199
RecordType          : 6
UserKey             : i:0h.f|membership|1003200045f60e72@live.com
UserType            : 0
Version             : 1
Workload            : OneDrive
ClientIP            : 191.232.238.156
ObjectId            : https://m365x413039-my.sharepoint.com/personal/irvins_m365x413039_onmicrosoft_com/Documents/file.com.txt
UserId              : irvins@m365x413039.onmicrosoft.com
CorrelationId       : 5d21dc9e-f036-0000-99ba-24034251dcf1
EventSource         : SharePoint
ItemType            : File
ListId              : d6961939-758a-4450-bc1a-408ae6bd9894
ListItemUniqueId    : 1244ed6c-a711-4843-9c55-14324daff226
Site                : a5a8b9b8-7d7f-49f4-9f0e-72a87296662a
UserAgent           : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140 Safari/537.36 Edge/18.17763
WebId               : 6b5bdf30-2c3d-4547-be42-d7b2222f8c1a
ImplicitShare       : No
SourceFileExtension : txt
SiteUrl             : https://m365x413039-my.sharepoint.com/personal/irvins_m365x413039_onmicrosoft_com/
SourceFileName      : file.com.txt
SourceRelativeUrl   : Documents
```

What additional actions have been done?

Positive Hack Days

```
CreationTime         : 2019-05-13T06:39:20
Id                   : c16eaba5-c732-465e-9fe0-08d6d76dbade
Operation            : FileMalwareDetected
OrganizationId       : 3f41e687-4cc6-48ac-a20a-ffa798046199
RecordType           : 6
UserKey              : S-1-0-0
UserType             : 4
Version              : 1
Workload             : OneDrive
ClientIP             :
ObjectId             : https://m365x413039-my.sharepoint.com/personal/irvins_m365x413039_onmicrosoft_com/Documents/file.com.txt
UserId               : SHAREPOINT\system
CorrelationId        : 7821dc9e-4079-0000-9a1f-ccfe68fa71b1
EventSource          : SharePoint
ItemType             : File
ListId               : d6961939-758a-4450-bc1a-408ae6bd9894
ListItemUniqueId     : 1244ed6c-a711-4843-9c55-14324daff226
Site                 : a5a8b9b8-7d7f-49f4-9f0e-72a87296662a
UserAgent            :
WebId                : 6b5bdf30-2c3d-4547-be42-d7b2222f8c1a
SourceFileExtension  : txt
VirusInfo            : DOS/EICAR_Test_File
VirusVendor          : Default
SiteUrl              : https://m365x413039-my.sharepoint.com/personal/irvins_m365x413039_onmicrosoft_com/
SourceFileName       : file.com.txt
SourceRelativeUrl    : Documents
```

What can Graph API tell us?

```
$return2.value | where {$_.ipaddress -eq "191.232.238.156"}|
Sort-Object -Property CreatedDateTime |select
CreatedDateTime,AppDisplayName,ResourceDisplayName,
UserDisplayName
```

```
PS C:\Distr> $return2.value | where {$_.ipaddress -eq "191.232.238.156"} | Sort-Object -Property CreatedDateTime |select Cre
ame

createdDateTime             appDisplayName                          resourceDisplayName                userDisplayNar
---------------             --------------                          -------------------                --------------
2019-05-13T06:10:31.7844606Z Microsoft App Access Panel             windows azure active directory     Irvin Sayers
2019-05-13T06:10:38.3804161Z Microsoft App Access Panel             windows azure active directory     Irvin Sayers
2019-05-13T06:10:49.082495Z  Microsoft App Access Panel             windows azure active directory     Irvin Sayers
2019-05-13T06:10:57.9312485Z Microsoft App Access Panel             windows azure active directory     Irvin Sayers
2019-05-13T06:11:09.9864723Z Microsoft App Access Panel             windows azure active directory     Irvin Sayers
2019-05-13T06:11:17.1252296Z Microsoft App Access Panel             windows azure active directory     Irvin Sayers
2019-05-13T06:32:04.7975695Z Office 365 Exchange Online             office 365 exchange online         Irvin Sayers
2019-05-13T06:32:22.6383724Z Office365 Shell WCSS-Client                                               Irvin Sayers
2019-05-13T06:32:24.4453497Z Office365 Shell WCSS-Client            officeclientservice                Irvin Sayers
2019-05-13T06:32:24.4599355Z Office365 Shell WCSS-Client            office365 shell wcss-server        Irvin Sayers
2019-05-13T06:32:24.4977132Z Office365 Shell WCSS-Client            microsoft graph                    Irvin Sayers
2019-05-13T06:32:29.5087252Z Skype Web Experience On Office 365     skype for business online          Irvin Sayers
2019-05-13T06:32:32.8980058Z Skype Web Experience On Office 365     skype for business online          Irvin Sayers
2019-05-13T06:32:36.4580336Z Skype Web Experience On Office 365     skype for business online          Irvin Sayers
2019-05-13T06:32:39.7780392Z Skype Web Experience On Office 365     skype for business online          Irvin Sayers
2019-05-13T06:32:42.42991Z   Skype Web Experience On Office 365     skype for business online          Irvin Sayers
2019-05-13T06:32:45.0343907Z Skype Web Experience On Office 365     skype for business online          Irvin Sayers
2019-05-13T06:32:47.188992Z  Skype Web Experience On Office 365     microsoft graph                    Irvin Sayers
2019-05-13T06:34:22.4319952Z Office 365 SharePoint Online           office 365 sharepoint online       Irvin Sayers
2019-05-13T06:40:24.9879823Z Microsoft App Access Panel             windows azure active directory     DemoAdm
2019-05-13T06:40:29.0370025Z Microsoft App Access Panel             windows azure active directory     DemoAdm
2019-05-13T06:44:54.9036466Z Azure Portal                           windows azure service management api DemoAdm
2019-05-13T06:46:17.6810836Z Azure Active Directory PowerShell      windows azure active directory     DemoAdm
2019-05-13T07:20:00.7712931Z Microsoft Office 365 Portal            windows azure active directory     New User
2019-05-13T07:20:10.4993942Z Microsoft Office 365 Portal            windows azure active directory     New User
2019-05-13T07:21:11.1486127Z Microsoft Office 365 Portal            windows azure active directory     New User
2019-05-13T07:26:10.6822737Z Microsoft Office 365 Portal            windows azure active directory     New User
2019-05-13T07:26:19.6275444Z Microsoft Office 365 Portal            windows azure active directory     New User
2019-05-13T07:26:23.6499416Z O365 Suite UX                          windows azure active directory     New User
2019-05-13T07:26:28.9341448Z Office365 Shell WCSS-Client                                               New User
2019-05-13T07:26:31.5193973Z Microsoft Office 365 Portal            windows azure active directory     New User
2019-05-13T07:26:31.6297205Z Office365 Shell WCSS-Client            microsoft graph                    New User
2019-05-13T07:26:31.9498301Z Office365 Shell WCSS-Client            office365 shell wcss-server        New User
2019-05-13T07:26:32.8619851Z Office365 Shell WCSS-Client            office365 shell wcss-server        New User
2019-05-13T07:26:42.2355246Z Office365 Shell WCSS-Client                                               New User
2019-05-13T07:26:44.2054266Z Office365 Shell WCSS-Client            office365 shell wcss-server        New User
2019-05-13T07:26:44.657483Z  Office365 Shell WCSS-Client            microsoft graph                    New User
2019-05-13T07:27:13.5588303Z Office 365 Exchange Online             office 365 exchange online         New User
2019-05-13T07:30:57.8843463Z Microsoft Exchange Online Remote PowerShell office 365 exchange online    New User
2019-05-13T08:15:36.8517073Z Microsoft Exchange Online Remote PowerShell office 365 exchange online    New User
```

What can Graph API tell us?

```
$BaseURI = "https://graph.microsoft.com/beta"
$SubURI = "auditLogs/directoryaudits"
$URI = "$BaseURI/$SubURI"
$Return = Invoke-RestMethod -Uri $URI -Headers $authHeader -Method Get -Verbose
$Return
```

```
PS C:\Distr> $Return.value | where {$_.activityDateTime -match "2019-05-13T0[6-8]{1,1}(.)*"} | Sort-Object -Property activitydatetime | select activitydatetime,act

activityDateTime           activityDisplayName                    targetResources
---------------           -------------------                    ---------------
2019-05-13T06:58:55.8845469Z Add user                               {@{id=6c488ccf-c527-47d7-924e-eddcb78c0b8e; displayName=; type=User; userPrincipalName...
2019-05-13T07:05:12.538898Z  Add user                               {@{id=72234e7b-0ba3-4866-91cc-0c607747b4a0; displayName=; type=User; userPrincipalName...
2019-05-13T07:16:35.0994213Z Add member to role                     {@{id=72234e7b-0ba3-4866-91cc-0c607747b4a0; displayName=; type=User; userPrincipalName...
2019-05-13T07:25:41.9062624Z Reset user password                    {@{id=72234e7b-0ba3-4866-91cc-0c607747b4a0; displayName=; type=User; userPrincipalName...
2019-05-13T07:25:41.9062624Z Update user                           {@{id=72234e7b-0ba3-4866-91cc-0c607747b4a0; displayName=; type=User; userPrincipalName...
2019-05-13T07:25:41.9112713Z Update StsRefreshTokenValidFrom Timestamp {@{id=72234e7b-0ba3-4866-91cc-0c607747b4a0; displayName=; type=User; userPrincipalName...
2019-05-13T07:25:52.716434Z  Update user                           {@{id=72234e7b-0ba3-4866-91cc-0c607747b4a0; displayName=; type=User; userPrincipalName...
2019-05-13T07:25:52.8314345Z Reset user password                   {@{id=72234e7b-0ba3-4866-91cc-0c607747b4a0; displayName=; type=User; userPrincipalName...
2019-05-13T07:25:52.8314345Z Update StsRefreshTokenValidFrom Timestamp {@{id=72234e7b-0ba3-4866-91cc-0c607747b4a0; displayName=; type=User; userPrincipalName...
2019-05-13T07:26:22.2587455Z Update user                           {@{id=72234e7b-0ba3-4866-91cc-0c607747b4a0; displayName=; type=User; userPrincipalName...
2019-05-13T08:36:05.3932379Z Remove member from role               {@{id=72234e7b-0ba3-4866-91cc-0c607747b4a0; displayName=; type=User; userPrincipalName...
2019-05-13T08:36:31.267853Z  Delete user                           {@{id=72234e7b-0ba3-4866-91cc-0c607747b4a0; displayName=; type=User; userPrincipalName...
```

Findings:

Patient zero probably is Irvin who was hacked due to weak password

His OneDrive and SharePoint files leaked

Adversary tried to send malware through OneDrive synch

Skype web experience logins point to EWS connection attempts.

We can suppose use of MailSniper or similar tools

DemoAdm password probably leaked from Irvin

Adversary created a temp user, gave him GA role and tried to gain persistence through setting mail forwarding for administrator account using journaling rules

Temp user was deleted

Takeaways:

1. We have got a basic understanding of audit mechanisms in Office 365 and Azure AD
2. We have got basic skills working with O365 Management API and Graph API
3. We have demonstrated possible investigation using Powershell and API

Thank you!