21

Understanding Microsoft 365 Defender APIs

Application programming interfaces (**APIs**) are a way to programmatically connect applications to other applications. In the context of Microsoft 365 Defender, this could mean connecting your PowerShell scripts to query data; an **independent software vendor** (**ISV**) connecting to provide additional value in their tool; or using a service such as Azure Logic Apps to automate a workflow based on Microsoft 365 Defender triggers.

In this chapter, you'll learn the fundamentals regarding APIs for Microsoft 365 Defender and its related services, such as MDE, MDO, MDA, and MDVM. Specifically, we're going to cover the following:

- The different APIs available, including their differences and when to use each
- Accessing the APIs and their permissions
- An example scenario, where we'll use PowerShell to perform an MDE operation using APIs

Based on this chapter, you'll start to think of your own creative possibilities for automation using Microsoft 365 Defender APIs. Without further ado, let's start exploring them.

Making sense of the different APIs

Microsoft has made five APIs available related to Microsoft 365 Defender and its associated services. Some of these APIs are exclusively related to Microsoft 365 Defender, and others are part of a broader collection. Let's check them out.

Microsoft Graph security API

The **Microsoft Graph security API** is part of the **Microsoft Graph API**. Its production request URLs are found at `graph.microsoft.com/v1.0/security` and beta request URLs at `graph.microsoft.com/beta/security`, with the beta endpoint subject to change and therefore not recommended by Microsoft for production use.

Microsoft is investing heavily in Microsoft Graph, and it's reasonable to say this should be your starting point and where to expect future investment. This means if an operation is available in the Microsoft Graph security API and another API, you may want to opt for the Graph version.

The following capabilities related to Microsoft 365 Defender are exposed in this API:

- **Action** (list, create, get, and cancel)
- **Advanced hunting** (run queries)
- **Alerts and incidents** (list, get, update, view evidence, and add comments)
- **Attack simulation and training** (run, list, and view simulation and report data)
- **Secure score** (list tenant and individual control results, and update individual control properties)
- **Threat intelligence indicators** – aka **indicators of compromise**, or simply, **indicators** (list, create, get, update, and delete for files, IPs, and URLs/domains)
- **Threat submission** (list, create, and get for files, URLs, emails, and email submission policies)

Over time, you can expect to see more Microsoft 365 Defender (and Microsoft 365 security capabilities in general) exposed through the Microsoft Graph security API. Now that we've reviewed it, we're on to the next important one – this time, one dedicated only to Microsoft 365 Defender.

Microsoft 365 Defender APIs

Also known as **Microsoft Threat Protection APIs** (the old name still seen in many interfaces), the **Microsoft 365 Defender APIs** are found at `api.security.microsoft.com`. There are three supported APIs within this:

- The **Streaming API**, which you would use to send events somewhere else for long-term retention, such as Azure Event Hubs or an Azure storage account. These events would be things such as `DeviceEvents`, `DeviceNetworkEvents`, `DeviceLogonEvents`, `EmailEvents`, or `IdentityLogonEvents`, which you can use for hunting.
- The **Incidents API**, which has feature parity with the incidents APIs available in the Microsoft Graph security API.
- The **Advanced Hunting API**, which is equivalent to the data exposed in the Graph version.

To summarize the Microsoft 365 Defender APIs, the Incidents and Advanced Hunting capabilities are probably now best handled by the Microsoft Graph security API, but if you have long-term data retention or need events stored elsewhere, the Streaming API is not currently available in the Graph API.

Microsoft Defender for Endpoint APIs

Also referred to as the **WindowsDefenderATP** API (historical names are hard to ditch!), the **Microsoft Defender for Endpoint APIs** are related purely to MDE and use the following URL: `api.securitycenter.microsoft.com`.

So, why would you need to use these instead of the last two we reviewed? As the name suggests, these are refined to MDE capabilities, also including some of MDVM. The full list of what's exposed is extensive and there-

fore referenced in the *Further reading* section, but the highlights include the following:

- Retrieve MDVM data, such as recommendations, vulnerabilities, and associated devices; installed software/extensions/certificates; and security baseline assessments
- Retrieve device health data, such as the **Microsoft Defender Antivirus** (**MDAV**) client running mode, engine update status, and signature status
- Perform machine actions, such as offboarding, isolation, initiating a scan, or running live response
- Manage the live response file library, indicators, automated investigations, and authenticated scan profiles

You can leverage the MDE APIs for the automation and scripting of many operations, and we'll return to them in our *API example scenario* section. Operationally, our capabilities for API access continue into MDA, which we'll review next.

Microsoft Defender for Cloud Apps API

Like MDE, MDA also has its own API, the **Microsoft Defender for Cloud Apps API**. The URL for this one differs by tenant, following the format of `<tenant>.<region>.portal.cloudappsecurity.com/api`.

With the MDA API, you can do the following:

- Review activities that MDA has discovered. For example, it exposes the entities performing actions, the action itself and its type, and other information you find in the MDA UI filter such as user agents, IPs, and tags.
- List or perform actions on MDA alerts, such as closing them.
- Perform cloud discovery log uploads, review continuous report data, or generate block scripts based on unsanctioned apps.
- List, create, update, and delete IP ranges as part of data enrichment.
- Retrieve files and entities (users and accounts).

Consider using the MDA API to scale your operations across multiple tenants or help in your general scripting and automation needs. Four down, one to go; let's review our last API for Microsoft 365 Defender information.

Office 365 Management Activity API

The **Office 365 Management Activity API** is used across Microsoft 365 to track activities, either automated or by users and admins. Found at `manage.office.com`, in the context of Microsoft 365 Defender, you can use this to retrieve several activities relevant to MDO and EOP:

- As part of its **MDO and Threat Investigation and Response schema**, find out about email message events, and the detection systems behind emails, attachments, and URL events

- As part of its **Submission schema**, retrieve data about admin and user email submissions
- As part of its **Automated investigation and response (AIR) events schema**, pull down MDO AIR data such as the threats found, any pending actions, timing, associated entities, and the investigation type
- As part of its **Hygiene events schema**, review outbound spam events

The Office 365 Management Activity API is worth reviewing in additional depth if you're responsible for other areas of Microsoft 365, as it logs activity across the suite.

Now that you've learned about the five core Microsoft 365 Defender-related APIs and the types of reasons you'd use each, let's have a look at how you can get access to them.

Accessing the APIs

As APIs expose a lot of sensitive data and, in many cases, give you the ability to change significant settings, their exposure is protected by default. To get access, you'll need to open up access, which is typically protected behind an access token or certificate.

For the APIs discussed in the last section, you'll manage permissions to the APIs using **Azure AD app registrations**. So, we'll guide you through creating an app registration, then use PowerShell as an example of using the APIs.

Creating an app registration for API access

As you create the app registration, you'll get to choose which API permissions are exposed, then obtain the credentials to access those APIs. We're going to use **application permissions**, which means they can run without a dedicated signed-in user (compared to **delegated permissions**, which access the API as an authenticated user).

To perform the following, you should be an Azure AD global administrator. Let's get going!

1. Head to **aad.portal.azure.com** | **App registrations** | + **New registration**.
2. Enter a name for the app and then click **Register**.
3. You'll be taken to your new app's page in Azure AD. Head to the **API permissions** page.
4. Click + **Add a permission**.
5. In the box that pops up, click on the **APIs my organization users** tab. From here, you can search for and select the APIs you learned about in the *Making sense of the different APIs* section to get the permissions you need. The specific names available are **Microsoft Graph**, **Microsoft Threat Protection**, **WindowsDefenderATP**, **Microsoft Defender for Cloud Apps**, and **Office 365 Management APIs**. You can only add one at a time, but you can return to add more to the same app later.

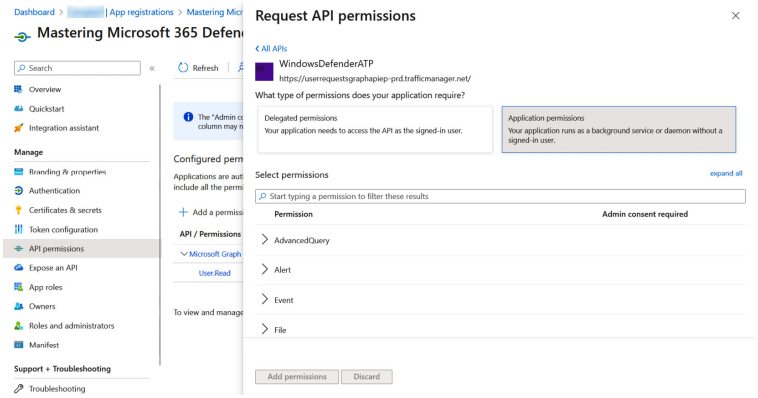

Figure 21.1 – Adding API permissions for an Azure AD app registration

6. After you select your API, you can scroll through the list of permissions and the boxes your app needs before choosing **Add permissions**. In *Figure 21.2*, you'll see that **Machine.Offboard** from **WindowsDefenderATP** was selected. You can add as many or as few as needed for your use case.
7. On the **API permissions** page, click **Grant admin consent for <your tenant>** and approve the prompt.

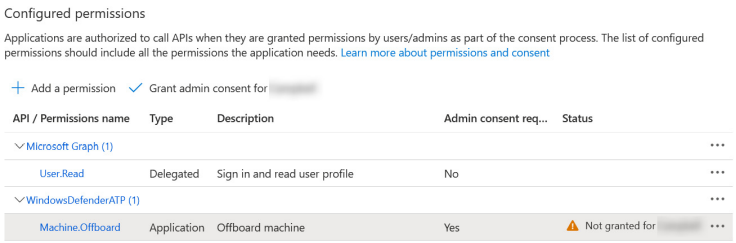

Figure 21.2 – Granting admin consent for permissions

8. Head to **Certificates & secrets** to get a client secret or certificate for your application to access the APIs you've now given it permission to access. In our example, we'll click + **New client secret** to create a secret valid for 3 months.
9. Securely record a note of the secret value. Note that it becomes unavailable after you leave this page.

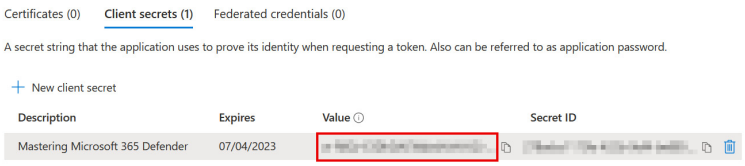

Figure 21.3 – Getting the client secret to access the API

10. In addition to the secret ID, you'll need two more pieces of information for our example scenario: the tenant ID and the application (client) ID. Head to the **Overview** page for your app registration and copy these from the **Essentials** section.

Our foundations have been laid! You've now got an app registration with a secret key that will let you work with the APIs you chose; let's use it in PowerShell.

Using the app registration for API access

Now, we're going to use our app and secret to get a token, giving us permission to the app to then offboard a device using the API. You can, of course, use this token for any other operations you granted permission for. Follow these steps:

1. Lee Ford, MVP, has a script for obtaining tokens on GitHub, so get a copy here: **gist.github.com/leeford/de7dd733161c3b37bca7f2c4ac45fc33#file-azureadgraphapiapplicationtoken-ps1**.
2. In this script, update the **`$clientID`**, **`$tenantID`**, and **`$clientSecret`** variables with those for your app registration.
3. Change the **`scope`** URL to that needed for your API. In our example, we're going to use **`scope = "https://api.securitycenter.microsoft.com/.default"`**. This is the API URL for the Microsoft Defender APIs, as you learned in the *Making sense of the different APIs* section. Amend it to different API URLs as required.
4. After executing this script, we have our token saved to the **`$token`** variable.
5. Now, you can use **`Invoke-WebRequest`** to perform the API actions. We're going for the example of offboarding a device, so we need the device ID from MDE. After we've got that, we can do the operation using the following:

```
Invoke-WebRequest -Method POST -Uri https://api.securitycenter.microsoft.com/api/machines/3ee
```

This is just one example of how APIs can be used. In this case, you accessed a machine action not exposed in the UI, and only available programmatically.

That concludes our dive into the APIs available for Microsoft 365 Defender. In the next section, we'll summarize what you need to know.

Summary

If you're going to be managing multiple tenants, or want to otherwise programmatically manage as much as possible, you're going to want to start mastering the APIs available for Microsoft 365 Defender and its associated services.

That's why, in this chapter, you learned about the five main APIs available, and the kinds of operations each exposes:

- The Microsoft Graph security API
- Microsoft 365 Defender APIs
- Microsoft Defender for Endpoint APIs
- The Microsoft Defender for Cloud Apps API
- The Office 365 Management Activity API

As a rule, if it's in Microsoft Graph, use it there to centralize efforts and stick with where Microsoft appears to be focusing most. However, you'll still find lots in the other APIs only. Most of the APIs focus on operations of Microsoft 365 Defender rather than settings management, but hopefully, we'll see this change over time so more things can be automated and managed as code.

Challenges

Rather than have questions for this chapter like the others, we've posed two challenges that can be attempted using the APIs learned about in this chapter:

1. Using an Azure Logic Apps app, try to update your team every time a new MDE alert is generated.
2. You have several scripts you want to add to the live response library for multiple tenants. Try to upload these using PowerShell.

Further reading

To continue your learning path of the world of Microsoft 365 Defender APIs, you can refer to the following resources:

- When you're using APIs, you should always refer to the latest official documentation, which for each of the types discussed in this chapter is available at the following URLs:
  - *Microsoft Graph security API*: **learn.microsoft.com/en-us/graph/api/resources/security-api-overview?view=graph-rest-1.0**
  - *Microsoft 365 Defender APIs*: **learn.microsoft.com/en-us/microsoft-365/security/defender/api-supported?view=o365-worldwide**
  - *Microsoft Defender for Endpoint APIs*: **learn.microsoft.com/en-us/microsoft-365/security/defender-endpoint/exposed-apis-list?view=o365-worldwide**
  - *Microsoft Defender for Cloud Apps API*: **learn.microsoft.com/en-us/defender-cloud-apps/api-introduction**
  - *Office 365 Management APIs*: **learn.microsoft.com/en-us/office/office-365-management-api/office-365-management-activity-api-reference**
- At the time of publication, Microsoft does not officially support an API for Microsoft Defender for Identity, but Raymond Roethof, MVP, has made available some very interesting research and tools into it. You can find his first blog post on it here: **thalpius.com/2022/12/08/microsoft-defender-for-identity-sensordeployment-api**.