Microsoft

An Introduction to the Microsoft Graph Security API

A technical explanation

Microsoft
Published: September 2018
Version: 1.0
Authors: Jean-Yves Grasset, Sarah Fender, Preeti Krishna, Michael Shalev
Reviewers: Kasia Kaplinska

For the latest information, please see
https://aka.ms/graphsecurityapi

Abstract

With the advent of the cloud and mobility, organizations are facing increased threats; intrusions and compromises of information systems have become the daily life of security managers. Threat detection and security management rely on dedicated teams within Security Operation Centers (SOC) who must monitor alerts in real time from all parts of the extended information system. The teams of analysts responsible for dealing with security incidents must switch between the various interfaces of the detection solutions which complicates the work of the analyst and makes difficult the correlation between alerts coming from the different systems.

The Microsoft Graph Security API provides a unified interface to connect security solutions from multiple providers (Microsoft or third party), simplifying integration of alerts and contextual data across security tools and workflows. The solution is open and allows all partners to make their security solution available through this interface and/or take advantage of the Microsoft Graph Security API to build or adapt their security solution.

Table of Contents

Notice

Microsoft makes no commitments and no warranties, express or implied, with respect to the information provided here.

Feedback

For any feedback or comment regarding this document, please send an e-mail to graphsecfeedback@microsoft.com.

Introduction

Objectives of this paper

The objectives of this white paper are:

- Define the Microsoft Graph Security API, including how it relates to the Microsoft Graph;
- Introduce the Microsoft Graph Security API and how it works;
- Explain the Microsoft Graph Security API features and benefits;
- Allow integrators, partners, ISVs... to understand the benefits of using the Microsoft Graph Security API to share security insights;
- Enable customers to understand how they can benefit directly from the Microsoft Graph Security API, or indirectly through the choice of solutions that rely on it.

Important note: Throughout the document, the terms "Microsoft Graph Security API" or "Security API for Microsoft Graph" or the abbreviated version "Security API" will be used interchangeably to designate the same service.

Non-objectives of this paper

This white paper is not a complete documentation of the Microsoft Graph Security API or an implementation guide.

Organization of this paper

The white paper is built in three main parts:

- The first part briefly introduces the Microsoft Graph to help understand how Microsoft Graph Security API is positioned.
- The second part focuses on the description of the Microsoft Graph Security API, its benefits, usage scenarios, and who can benefit from it.
- The last part introduces opportunities for integration with an existing SIEM and integration opportunities for partners.

About the audience

This document is intended for CISOs, CSOs, Developers, MSSP (Managed Security Service Providers), Security System Integrators, IT professionals and developers who are interested in understanding what is the Microsoft Security Graph API, what it can bring to them as benefits and opportunities, and how to integrate to leverage its features.

The Microsoft Graph

The Microsoft Graph (graph.microsoft.com) offers a **unified interface for accessing information from Microsoft online services** (Azure AD, Office 365, OneDrive, OneNote, SharePoint, Planner, PowerBI, Intune, etc.) in the form of APIs that are simple to implement, share a common authentication framework, and support a wide variety of platforms with easy to use SDKs and code samples.

The Graph provides access to information related to identities (users, groups, and login activity), contacts, emails, documents, tasks, etc. to support user-centric applications[1]. For example, for all Office 365 documents stored in OneDrive – personal or shared –, using the Graph facilitates access in a simple syntax and provides the ability to share documents for more complex scenarios.

The Microsoft Graph can be accessed through a single endpoint https://graph.microsoft.com and adopts a standard schema for authentication, based on OpenID Connect and OAuth 2.0, and the use of Web REST API with standard JSON response formats.

> **Note** For a complete description of the Microsoft Graph API usage, please refer to MICROSOFT REST API GUIDELINES on Github.

This single connection point is the root for the other namespaces that are assigned to the different services accessible through the Graph. For example, the REST requests concerning the current user will be done on the URL https://graph.microsoft.com/v1.0/me, V1.0 allowing to implement versioning and "me" representing the space related to the current user.

In return for this call, the response, in a JSON format, provides information about the profile of the current user.

```
{
    "@odata.context":
"https://graph.microsoft.com/v1.0/$metadata#users/$entity",
"id": "48d31887-5fad-4d73-a9f5-3c356e68a038",
    "businessPhones": [
    "+1 412 555 0109"
    ],
    "displayName": "Megan Bowen",
    "givenName": "Megan",
    "jobTitle": "Auditor",
    "mail": "MeganB@M365x214355.onmicrosoft.com",
    "mobilePhone": null,
    "officeLocation": "12/1110",
    "preferredLanguage": "en-US",
    "surname": "Bowen",
    "userPrincipalName": "MeganB@M365x214355.onmicrosoft.com"
}
```

Then to access the mails, you have to add to the URL "/ messages", to access the OneDrive of the user "/ drive", to OneNote "/ onenote", etc. and then a filter if necessary.

[1] Overview of users in Microsoft Graph https://developer.microsoft.com/en-us/graph/docs/concepts/azuread-users-concept-overview

To get the list of important messages for the current user, simply send the POST request https://graph.microsoft.com/v1.0/me/messages?$filter=importance eq 'high'.

The Graph Explorer online tool https://developer.microsoft.com/en-us/graph/graph-explorer provides examples of simple REST queries and also lets you create and test your own queries. Without authentication, you access a sample user account, but you need to sign in with a Microsoft account to run the queries in your context.

> **Note** Inside Graph Explorer, click on the "show more samples" link to select examples from the categories you are interested in (Users, Mails, OneDrive, Insights, etc.).

The benefit to the application developer is evident because from a single point of access, with a Web standards-based access interface, Microsoft Graph provides unified access to an ever-growing set of services. One of the latest is the support of security through the **Microsoft Graph Security API** which is the subject of this white paper.

The Microsoft Graph Security API

Introduction

Let's start with a definition: The Microsoft Graph Security API can be defined as an **intermediary service (or broker)** that provides a **single programmatic interface** to connect multiple security providers. Requests to the graph are **federated to all applicable providers**. The results are aggregated and returned to the requesting application in a common schema.

There is no additional cost to use the Security API: An Azure subscription is all that is required. However, access to data from service providers may depend on any necessary subscriptions. Read the Roadmap section for current and future direction of Microsoft Graph Security API.

Microsoft Graph Security API has introduced the alerts entity as a first step to provide a unified gateway to security insights across Microsoft and non-Microsoft security providers. The following chapters will focus primarily on security alerts as these are available today from a wide range of security providers. Just a quick reminder of what a security alert is.

A **security alert** is triggered by an event or sequence of events that is **characteristic of a suspicious behavior**. The alert is intended to bring the attention of an operator or system to initiate a remediation action. Either the analysis shows a real threat that requires a reaction to counter it, or it is a false alarm – a false positive – and the alert will be cleared.

Let's take the example of Azure Security Center: this service collects the Azure resource logs (VM, network ...) and partner solutions that are used (firewalls, endpoint protection solutions ...) and analyzes them to detect possible threats or attacks.

For example, Azure Security Center will detect brute force attacks on an RDP access, the execution of a suspicious process on a virtual machine, an outgoing communication to an IP address considered malicious (a command-control server of a botnet), a system binary alteration, and so on. All these detections will generate alerts visible in the Azure Security Center console, to be processed by a security analyst who will be responsible for investigating and remedying the incident. For a brute force RDP access attempt, action may be required to reinforce the password or to impose multi-factor authentication; for detecting a suspicious process or a binary system modification, to disinfect or rebuild the machine.

Azure Security Center is the typical case of what is referred to as a security provider: it analyzes security events using a combination of behavioral and rules-based analysis (an attack scheme, a particular sequence) and reports the detection of potential threats as an alert.

> **More information** Managing and responding to security alerts in Azure Security Center https://docs.microsoft.com/en-us/azure/security-center/security-center-managing-and-responding-alerts

Before going any further, and not to disappoint the reader, it is necessary to understand what the Microsoft Graph Security API is NOT:

- It is not a collector of security logs coming from multiple sources and formats that can then be analyzed. In a broker role, the security APIs do not store any information;
- It is not a management portal or a security dashboard: APIs are hidden entities to ordinary mortals;
- It is not a Microsoft SIEM;

- It is not a threat analysis platform: The Security API does not integrate intelligence, strictly speaking, but plays the role of unifying service.

Customers' challenges

Increased threats

With the advent of the cloud and mobility, corporate environments have become increasingly open, extended and complex. Businesses must deal with amplified threats and intrusions, and compromises of information systems have become the day-to-day for security managers.

Threat detection and security management rely on dedicated teams within the SOC who must monitor alerts issued from all parts of the extended information system in real time. Smartphones or desktops, cloud services, cloud or on-premises applications, users' identities, systems protection, etc. All these elements must be monitored to make sure to limit the spread of a compromise, the exfiltration of data, or unauthorized access to sensitive data - but at the cost of an increasingly large number of alerts from disparate and disconnected detection systems.

To manage this flow of data, detection systems are becoming more and more powerful by relying on artificial intelligence that allows them, by implementing their own correlation algorithms, to generate ever more precise alerts while limiting the number of false-positives.

Lack of unification and difficulty to analyze

However, each solution is adapted to its own context. For example Microsoft security solutions are multiple: Microsoft Azure Security Center is designed to protect cloud and on-premises infrastructure and issue its own alerts, Windows Defender ATP detects on a Windows 10 client and gives insights and tools to close incidents quickly, Azure Active Directory Information Protection helps identify and alert on anomalous data access. Third-party solutions also provide their own unique alert detection and management systems, and it is not uncommon to see that most large organizations' SOCs are equipped with dozens of different security solutions.

Analysts responsible for handling security incidents face several difficulties in their day-to-day work: they have to juggle between the different interfaces of the detection solutions which complicates the work of the analyst and makes it difficult to correlate alerts from different systems. In general, working with **disconnected alerts** makes the investigation work more complex and tedious. **The analyst does not have a complete and consolidated vision on all security solutions**.

Take the case of an alert raised by a solution concerning a specific user: the analyst wants to check if the alert is only generated by this solution or if in a more global way, several alerts were triggered for this person by different security systems. Indeed, the priority of treatment will be much higher in the second case. The analyst will have to interrogate all the security systems to get a global view of the user and quickly decide whether it is a single tinkling or whether all the bells ring at the same time. This means an inefficiently used time in the processing of the alert because of the additional effort on the part of the analyst, the possibility of minimizing the criticality of an alert because of an unconsolidated vision of the entire landscape and most importantly, delays in mitigating a real threat.

Heterogeneous schemas

Some customers will want to take care of the consolidation of alerts by themselves but will face a new pitfall because each solution works with its own description of the alerts (we will speak more specifically of a schema) i.e. that **there is no common schema for all alert providers**.

As a matter of fact, alerts coming from an Intrusion Detection System (IDS), an Endpoint Detection System) (EDS) or a Cloud Access Security Broker (CASB) will not present the same fields simply because the information is specific to each of the solutions. As a result, interfacing between the solutions will be complex and any addition or replacement of a solution will require additional effort.

Lack of context

The other limitation that we will face is the **lack of context**. The alert contains information on the initiative events or triggers (detection of malware during a scan, suspicious authentication, etc.) and some contextual elements that the security system knows (the name and type of malware, the IP address of the post, the processes involved ...). The work of investigation may require access to additional information that may not be available from this provider, but which is distributed among the different providers and which will enrich the entire context around this alert.

The SIEM allows to consolidate logs coming from different providers to obtain interesting information by setting up correlation. But, in the context of an investigation, it is possible that the contextual elements brought by the logs are not sufficient and the interface of the SIEM will not be able to give the capability to the analyst to question the providers. The path is only one way: the services (providers) send their information through the logs but the SIEM cannot request the services to retrieve additional context information. This again hinders and complicates the work of investigation since the analyst will have to switch between different consoles to try to recover this additional context information while themselves performing the necessary correlation between the different sources.

Operational complexity

In addition, the response to a threat detected through the tools available in the SOC is not necessarily easy: remediation actions must be performed to the protection solutions manually or automatically through runbooks and workflows. Here again, the multiplicity of solutions brings an **operational complexity** which slows down the efficiency and the reaction time to address a threat. One could imagine the possibility through a single runbook to act immediately on several systems such as for example block a malicious IP address at a perimeter firewall and do the same thing at the endpoints.

Finally, access to the configuration of security solutions that are deployed is not be possible through the consoles. It may be necessary to verify that security policies have been correctly applied for either protection or compliance reasons.

Microsoft Graph Security API: Architecture overview

The aim of the Microsoft Graph Security API is to provide answers to these challenges that are facing our customers.

In the case of security alerts, the Microsoft Graph Security API provides a **unified interface for managing security alerts** in a consistent manner coming from a set of security service providers for the benefit of applications.

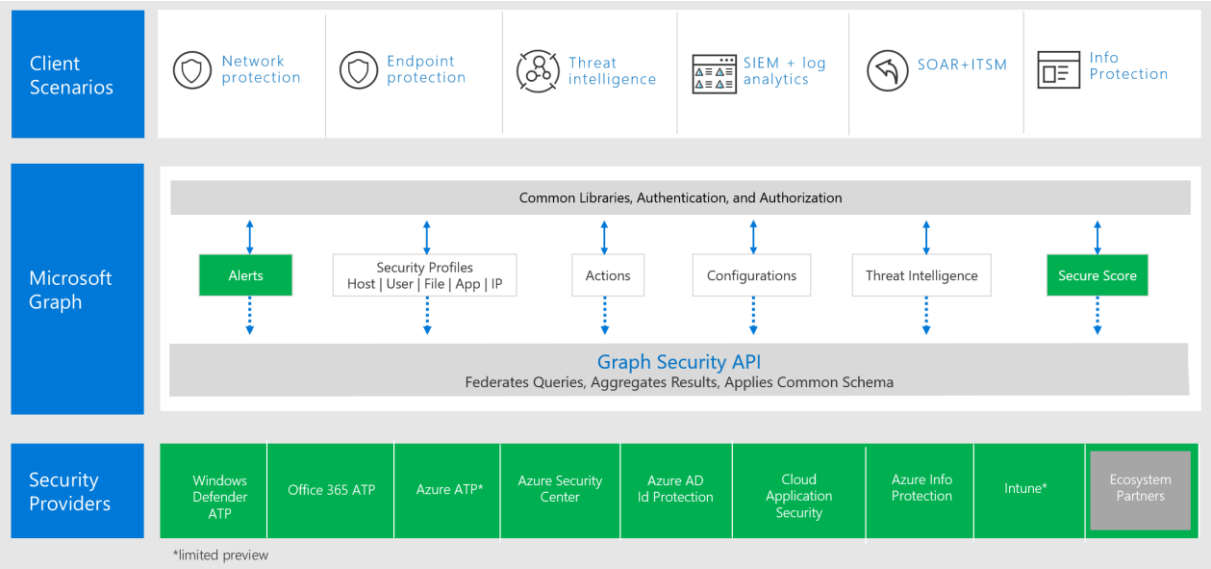
On one side we have **security providers** (see diagram) that generate alerts: these security providers can be linked to Microsoft services, such as Azure Security Center, Azure Active Directory Identity Protection, Windows Defender ATP, or third-party solutions providers such as endpoint and network protection solutions, application and email security solutions, etc. Providers implement their own detection logic that generates alerts and are responsible for managing alert lifecycle. Since we are onboarding providers all the time, please see the Microsoft Graph Security documentation for the latest list.

On the other hand, there are **consumers or clients** relying on the Security API to consume alerts from multiple security vendors in their own solutions. These solutions include SIEMs, incident ticket management systems, security monitoring systems, threat intelligence, Managed Security Service providers (MSSP), plus a variety of threat protection solutions.

The Microsoft Graph Security API service is accessible through a Web-based interface (REST) with a reduced set of functions to manage alerts. Applications can do the following operations:

- Query the API to get all or filtered set of alerts.
- Update or PATCH alerts to get different security solutions in sync.
- Subscribe to and receive notifications about updates to Microsoft Graph webhooks.

Microsoft Graph Security API libraries are available in few languages - .NET, JavaScript and expected to be made available on the most common mobile platforms (iOS, Android).



Authentication is supported by the Azure Active Directory (AAD) through the OpenID Connect and oAuth 2.0 protocols. Requests sent from a specific user in AAD tenant is authenticated by Microsoft Security Graph and then federated to multiple providers the application has consented to. Graph Security API

aggregates the results, and response is sent back in json format specified by the schema. Application level authorization is also supported including Role-Based Access Control (RBAC) managed by the application.

Key concepts and benefits

Let's see the benefits and features the Microsoft Graph Security API has to offer.

Unification, federation and standardization of alerts

The first advantage of the Microsoft Graph Security API is to **unify and standardize the management of alerts**. The implementation of the Security API ensures the **federation** of all security providers connected to them; all security providers can be accessed through a single interface: The Graph Security API. The application that relies on the Security API can query each security provider with a **single language**: for example, the same query will be used to retrieve the top 5 alerts from each provider.

But the Security API can go further by offering the possibility in one request to query all the security providers: for example, it is possible to request the top 5 alerts on all providers and the service will be responsible for fanning out the query on each of the providers and then concatenating all the answers in a **single result**. When the number of suppliers tends to grow, the benefit of having a complete view of the information on alerts from all connected providers is obvious.

Alerts life cycle management

Alert management also needs ensure that the **complete lifecycle of alerts** is implemented. The Microsoft Graph Security API allows to update the information associated with the alerts through attributes such as the status, the analyst in charge of processing, the date of the alert acknowledgment, for each connected provider. It is also possible to synchronize the status of an alert against all security providers: a single status update request and the update is performed automatically for all providers.

The Security API provides an abstraction layer that can "standardize" access to suppliers **that drastically simplifies the integration** of a new source of alerts: a new security provider connecting to the APIs will be immediately recognized and will enter the list of providers. It will be solicited by any new request to get the information on alerts or other information that it knows and manages (e.g. "entities" of type host, user, IP address, etc.).

This provider must of course be able to integrate with the API (for more information, see the paragraph INTEGRATING YOUR SOLUTION AS A SECURITY PROVIDER.). According to the same principle, if you want to switch to a security provider from another security vendor, nothing is easier than to delete a provider and to replace it with another one.

Using a single schema

The schema defines the structure of the information that is transmitted and manipulated. Each object is defined by a structure of its own and composed of a set of properties. The schema used by the Microsoft Security API defines the structure of alerts – the richest structure – and other entities related to security profiles.

> **Note** Five security profiles have been defined corresponding to information about users (userSecurityProfile), hosts (hostSecurityProfile, files (fileSecurityProfile), IP addresses (IPSecurityProfile) and applications (applicationSecurityProfile).

However, each security provider has its own schema to store its specific information. Consequently, it is necessary to agree on a **common definition of information that must be found in each alert** to provide a **unified view** for applications that rely on the Security API. This is the main interest of

complying with the common schema defined by the Security API. Each provider is then in charge of providing its information through the prism of this common schema.

The benefit of a single schema is to be able to query with filters that will apply to all providers. For example, a single request will be able to retrieve from all the providers, the alerts whose status is "InProgress" by filtering on the common attribute "status" of the schema. The analyst may also retrieve hyperlinks to access additional information from the providers by querying the "sourceMaterials" attribute. In addition, the developer who codes his application in C # or the analyst who wants to develop his own scripts in Python will have to develop **only one version of code** to access the information from all security providers associated with the Microsoft Graph Security API.

A common schema also makes it easier to correlate the events related to an alert (for example the user, computer, IP address, file, or detected malware) and alerts or events that have occurred and relating to these elements.

Access through Graph APIs: the same paradigm

The Security API is accessible through the Microsoft Graph, that is, using the single point of connection https://graph.microsoft.com, with the same syntax REST API and response format JSON (see chapter THE MICROSOFT GRAPH).

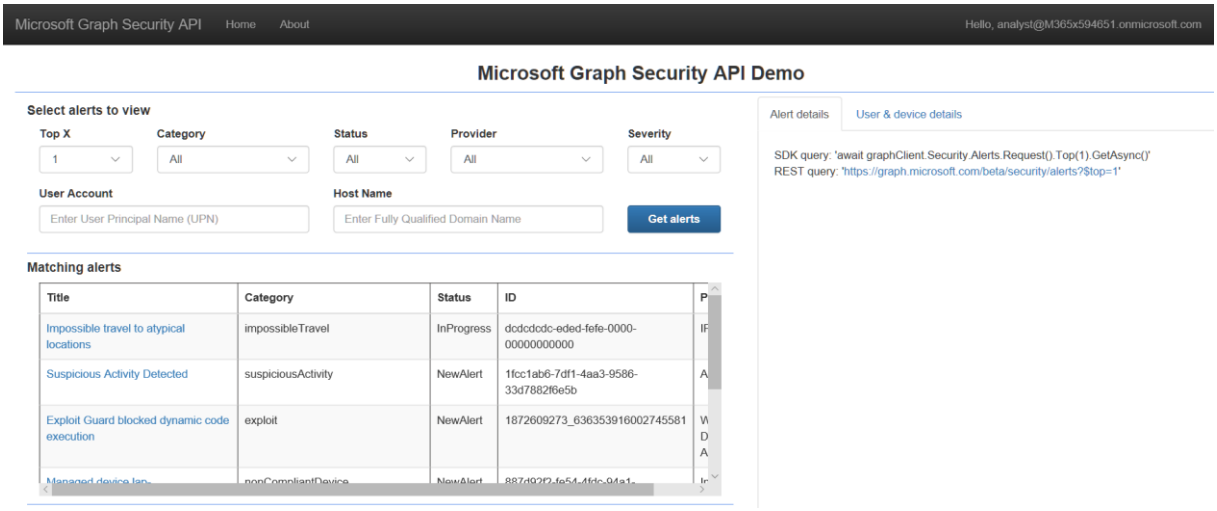
The Security API has its own namespace, currently graph.microsoft.com/v1.0/security, which allows for example to access all the "top 1" alerts of each service provider by the request:

GET https://graph.microsoft.com/v1.0/security/alerts?$top=1

The Microsoft Graph Security Demo application, available on GitHub microsoftgraph/aspnet-security-api-sample, is a sample ASP.NET application that allows to easily run and test queries to Security API. The procedure for registering the application, setting permissions and compiling the application in Visual Studio (or the free Visual Studio Code version) are detailed. The demo sample is also available in:

- Python
- Node.js (JavaScript)

The screen of the demo application shown below allows to view, on the right, the request made and, in the table "Matching alerts", alerts coming from each of the security providers.



Note You cannot use GraphExplorer to test queries on the Security API. Indeed, Graph Explorer currently does not ask for the permissions needed to access the Security API. See https://stackoverflow.com/questions/50254712/https-graph-microsoft-com-beta-security-alerts-not-returning-any-data-value.

Authentication and Access

Security data accessible via the Microsoft Graph Security API is protected using both permissions and Azure AD (AAD) roles. Because the Graph is authenticated to your domain through AAD, only people with the appropriate permissions can get responses or take actions on your security data using the Security API.

Security API can be accessed either by an application – for example in the case of a SIEM or a daemon – or in the context of an authenticated user (for example through Graph Explorer). First of all, the application must be registered at the tenant level, which will be commonly the responsibility of the developer or the person in charge of the application. Next, the tenant's administrator must assign the necessary authorizations to the application. If users need to access the Security API, the administrator will need to assign them access permissions.

More information For detailed information please consult AUTHORIZATION AND THE SECURITY API IN MICROSOFT GRAPH https://developer.microsoft.com/en-us/graph/docs/concepts/security-authorization and the blog post UNDERSTANDING AUTHORIZATION WHEN CALLING THE MICROSOFT GRAPH SECURITY API https://techcommunity.microsoft.com/t5/Using-Microsoft-Graph-Security/Authorization-and-Microsoft-Graph-Security-API/m-p/184376.

In fact, the control is done at 3 levels:

- Authentication at the Graph level (authentication with the tenant's user account);
- At the Security API level (Application or user);
- At the Security provider level which also integrates its own Role based access control (RBAC system).

Please note that the 3 levels of authentication are managed by the customer's tenant admin and that consequently the customer always owns the control of access to the data. At any moment, the permissions can be removed, and the application will lose access.

An open solution

Finally, the implementation of APIs is a **completely open solution** since any provider of a security solution can decide to create its own provider that will be aggregated to all available providers by integrating into this " Security Fabric ". The connection APIs are documented and all you need to do to easily integrate is implement the queries and make an endpoint available. Microsoft provides access to its own solutions (see the roadmap chapter for more details) but opens the possibility for all partners to rely on the Microsoft Graph Security API.

Usage scenarios

Some examples scenarios

Below are some common usage scenarios to illustrate the benefits of the Graph Security API.

- The analyst receives an alert from a provider about a user and queries all security providers through the Microsoft Graph Security API for a complete view of the user context: if other providers also send alerts, the analyst has a broader view of the extent of the threat or compromise for effective correlation.

- The analyst receives a list of alerts from providers through the Microsoft Graph Security API and assigns these for investigation after initial investigation. The ticketing system that files bugs associated with these alerts is also integrated with Microsoft Graph Security API. The assignments are automatically reflected in the ticketing system without the analyst having to login and update the tickets with alert assignments to keep it in sync.

- Multiple analysts are working on a critical alert from a security product integrated with Graph Security API. One of these analysts is waiting for her turn to analyze the alert pertaining to her area. She has to go for a meeting, however would like to be notified about the status of this alert. To do so, she subscribes to receive notifications on this alert via Microsoft Graph webhooks. The alert gets assigned to her while she is in the meeting. She gets promptly notified about the change and excuses herself from the meeting to complete the investigation.

- An alert is issued after Azure Identity Protection detects an "Impossible travel to atypical location" and sent to the analyst. Even before conducting the investigation, the analyst can enforce a multi-factor authentication. This action can be automated by a runbook.

- An alert is raised after the discovery that a computer is establishing a communication to a malicious IP address. The analyst launches a runbook that will block the connection from the workstation to this IP address on the company's firewall and then launch (through Windows Defender) the disinfection of the workstation.

Detailed scenario

Gary is an analyst working in a SOC and his role is to deal with security alerts, conduct the necessary investigations and, if necessary, implement remediation solutions. It uses a SIEM solution integrated with the Security API (see INTEGRATING ALERTS WITH YOUR SIEM) and displays all new alerts on his dashboard.

He is just receiving an alert concerning the user JaneDoe: "Account janedoe targeted by attacker via e-mail". He wants to get more information about this user and makes a request through the Security API on the security profile. He gets the user's UPN (janedoe@contoso.com).

```
GET /security/alerts/1234?$expand=userStates &$select=userPrincipalName
```

He gets the user's UPN (janedoe@contoso.com).

Gary wants to identify from which computer originates the attack, and he just needs to make a query through the Microsoft Graph for the devices associated with Jane Doe.

```
GET users/janedoe@contoso.com/registeredDevices?$select=displayName
```

This answer is that Jane's device is named BLUEFINANCE05.

All Gary has to do is run the following command to retrieve the list of applications executing on BLUEFINANCE05.

```
GET /security/hostStates/?$filter=NetBiosName eq
'BLUEFINANCE05'&$select=applications
```

Gary realizes that Jane uses the critical app "Contoso Payroll Application" and decides to block access to this SaaS application through Microsoft Cloud App Security (MCAS).

We already see that with a few basic commands, it is easy for Gary to conduct an investigation. However, rather than having to enter these commands manually, the SIEM could easily integrate them in its interface to view the elements associated with the alert by simply querying Microsoft Graph and Microsoft Graph Security API.

Who could use the Microsoft Graph Security API?

After understanding the features and benefits of the Security APIs, it is interesting to know who among the ecosystem or the customers may be interested in using them.

- **Security vendors**[2]: they can integrate their security solution as a security provider to make their alerts and contextual information visible through the Security API. They will be automatically visible by applications or solutions that rely on the Security APs. Their security solution can also integrate as a consumer of the API to process the alerts coming from their solution but also from all the Microsoft and other third-party solutions connected to the Security Graph.

  Take the example of the Palo Alto Networks, which has integrated its App Framework as a security provider. App Framework users can also call the Security API to get alerts from other providers, enabling them to correlate alerts across their systems.

  In addition, security solution providers that offer alert management and task automation in response to these alerts can immediately take advantage of the Security API.

- The **managed security service providers** (MSSP) offer monitoring and security management to their customers through their SOC (Security Operation Center). They can immediately benefit from the Security API to integrate various security solutions, take advantage of the insights of different security providers, aggregate alerts into their own dashboards and enrich them with contextual information.

- Partners that offer **IT Service Management** platforms can leverage the Security API to integrate it into incident management, automation, dashboards and reports. Applications that are not directly related to security – for example HR or financial applications – can also use the Security Graph to benefit from an additional security context.

- Finally, **customers or clients** who have their own SOC will be able to use the Security API to integrate it into their own in-house solutions, enjoy the ease of integration they provide in already existing SOC procedures and tools (SIEM) and the capability of creating automation playbooks.

[2] Please note the difference between the provider that designates the product or service (e.g. Windows Defender ATP) and the vendor that corresponds to the company that makes the provider available. For more information see https://developer.microsoft.com/en-us/graph/docs/api-reference/beta/resources/securityvendorinformation

Integrating alerts with your SIEM

As a reminder, the Security API federates the alerts coming from the different security providers connected to it but does not consolidate logs as a SIEM does. On the other hand, it can be interesting to interface an existing SIEM so that it retrieves the alerts generated by the Security API. The SIEM can therefore integrate these alerts in its management of events to be able to correlate them and include the result its dashboards. Azure Monitor will serve as a natural channel for integrating the Security Graph with third-party SIEMs by taking advantage of already existing connectors.

Introduction to Azure Monitor

Azure Monitor is the basic infrastructure for managing metrics and logs from Azure services. It acts as a pipeline to consolidate all Azure service logs. Most Azure services use Azure Monitor, including Azure Security Center, and all Azure services will be integrated in the near future.

Azure Monitor can retrieve information from applications (logs, performance counters, diagnostics), VMs (metrics and diagnostics), and Azure infrastructure activity logs. This data is kept for a period of 90 days (except for diagnostic logs) but they can be stored in Azure Storage if you want to keep them longer or for archival purposes. Information can be transferred to Application Insights for analysis or forwarded to third-party solutions through Event Hubs[3].

Microsoft has worked with leading SIEM solution partners to enable integration of their platforms by leveraging Event Hubs. Solutions like Splunk, IBM QRadar, Arcsight or others like ELK stack and SumoLogic integrate with Azure Monitor through connectors whose implementation is significantly different depending on the product.

> **More information** For additional information please refer to USE AZURE MONITOR TO INTEGRATE WITH SIEM TOOLS https://azure.microsoft.com/en-us/blog/use-azure-monitor-to-integrate-with-siem-tools/, STREAM AZURE MONITORING DATA TO AN EVENT HUB FOR CONSUMPTION BY AN EXTERNAL TOOL https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitor-stream-monitoring-data-event-hubs and AZURE MONITOR ADD-ON FOR SPLUNK https://splunkbase.splunk.com/app/3534/#/details.

Azure Monitor and Microsoft Graph Security API integration

Azure Monitor now offers the ability to interface with the Security API to transfer alerts generated through Event Hubs. As a result, all SIEM solutions that have a connector (see above) can retrieve all the alerts sent by the security providers and federated by the Security API. This provides the benefit of having a tenant level single pipeline for events and alerts.

The principle of implementation is simple: after creating an Event Hubs namespace to receive alerts, Azure Monitor is configured to send alerts to Event Hubs. Once the connector to the SIEM is set up and the SIEM configured, the SIEM will receive security alerts to process them.

---

[3] Overview of Azure Monitor https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-overview-azure-monitor

More information As an example, the detailed procedure for integrating the SIEM Splunk to allow it to receive alerts related to the Security API is documented in the article INTEGRATE SECURITY API ALERTS WITH YOUR SIEM USING AZURE MONITOR https://developer.microsoft.com/en-us/graph/docs/concepts/security_siemintegration.

It should be noted that the SIEM which receives the information on the alerts must be able to interpret the different fields of the event "alert" by relying on the Graph Security API schema to derive all the benefits. In addition, the information flows in one direction and the alerts generated by the Security Graph are transferred to the SIEM. But the SIEM can also rely on the Security API to retrieve additional elements of context acting as a consumer.

Note Check with your SIEM vendor to find out what additional level of integration with the Security API the solution might provide.

Integration scenarios for partners

Partners who want to rely on the Graph Security API can do so either as a consumer or client, or as a security provider, or both. In the chapter WHO COULD USE THE MICROSOFT GRAPH SECURITY API? are listed some use cases of the Security API. For example, an editor of security solution – software and possibly hardware – will have a strong interest in making his solution accessible through the API and will just need to develop a security provider that can be integrated and accessible by any consumer application. Other partners will develop solutions that will only consume the Security API, such as SIEM which, while retrieving alerts through the import of logs and alerts, would rely on the API for retrieving additional context information or trigger actions.

Building an application leveraging the Graph Security API

The Security API is accessible through the Microsoft Graph. To build a solution based on the Security API as a consumer, developers will rely on the development kits made available by Microsoft on GitHub. These SDKs are available in ASP.NET, Xamarin, Java, JavaScript, Angular, Php, etc. and include many examples of implementation. Each developer will be able to choose his preferred language, or one that best suits the solution he develops, to easily interface with the Security API and derive the most benefit of it.

> **More information** To obtain the SDKs for accessing the Microsoft Graph, please see the MICROSOFT GRAPH CODE SAMPLES AND SDKS https://developer.microsoft.com/en-us/graph/code-samples-and-sdks which redirects to the SDKs on Github.

Integrating your solution as a security provider

In general terms, the integration of a security provider must respect the following elements:

- Providers must define and declare an endpoint on which they are accessible in https.
- Authentication between the Graph Security service and the provider uses either Basic authentication or Service to Service (S2S) token-based authentication over an https channel.
- All https requests include a header containing, but not limited to, a query ID, the vendor name, the provider name, and the Azure tenant ID.
- Queries with the GET request respect the OData syntax, knowing that the queries are transferred as received by the Security API. The OData options that must be supported are $top, $skip, $orderby, $filter, ('<id>') and $count.
- The Provider service should always return standard OData response with content in JSON format, and ONLY return data for the requested tenant.

> **More information** For a detailed description of the OData syntax, please see URL CONVENTIONS (ODATA VERSION 3.0) https://www.odata.org/documentation/odata-version-3-0/url-conventions/

To help **Provider** services onboard to Graph Security, Microsoft provides a test tool to generate requests to the **Provider** endpoint and analyze the response to ensure Graph Security can understand and parse the response. You can reach out to the Microsoft Graph Security team via email at graphsecfeedback@microsoft.com to get help to onboard as a provider.

The full detailed specifications will be available publicly when the APIs are available in final version or available in beta version after subscription to the program with the development team.

Roadmap

The roadmap for Microsoft Graph Security includes support for the following scenarios. A few of these scenarios are elaborated in the following sub-sections.

- Alert ecosystem - Continue onboarding security alerts from multiple Microsoft and non-Microsoft security providers to build a rich ecosystem for applications integrated with the API.
- Secure Score – Enable visibility into your security posture and how to improve it as an effective risk mitigation technique.
- Security profiles – Support contextual information pertaining to a specific alert to speed up investigations further.
- Actions – Enable to perform actions pertaining to specific alerts.
- Configurations – Support for policy enablement on a global / corporate scale.

Proactive Security Risk Management: Secure Score

Microsoft Secure Score helps provide visibility into your organization's security posture and get suggestions for how to improve it, and project an improved score after those suggestions are incorporated. With a single score, you can better understand what you have done to reduce your risk in Microsoft solutions. You can also compare your score with other organizations and see how your score has been trending over time. The Microsoft Graph Security secureScore and secureScoreControlProfiles entities help you balance your organization's security and productivity needs while enabling the appropriate mix of security features.

> **More information** Secure Score is in Preview – refer to the API documentation for Secure Score for further details.

Access to additional context: Security Profiles

The Security API implements the notion of security profiles that correspond to information stored by the providers as part of their processing but not available directly in the alert. Each provider will have its own information that it collects, aggregates and analyzes as part of its service: Windows Defender ATP, through its Windows 10 client, will have precise information about the computer, the processes, the connected users, the executables at risk; a CASB solution such as Microsoft Cloud App Security will integrate information about cloud applications, transferred files; Azure Active Directory will provide a view of users and their risk score, etc.

This information is accessible through Security Profiles, which allow to query the providers connected to the Security API to **obtain contextual information** related to a specific alert.

Imagine the case of malware detected on a workstation: the alert will contain information about the user logged on this machine (his UPN or User Principal Name), but the information present in the alert that characterizes the machine are relatively limited (fqdn, Netbios name, public and private IP addresses ...). It may be interesting for the analyst to access additional information for the purposes of the investigation. This information – we also talk about inventory data – will help him in his research and allow him to correlate alerts with contextual data associated with these events.

The analyst may be interested in looking for related information such as the list of devices from which that same user has logged on, or if other alerts have already been raised relative to that user. This information can be retrieved by a simple query through the Security API on all security providers.

A particularly interesting property is the riskScore assigned by each provider on the profile. Unlike other properties that are static (OS version, home domain name, etc.) **the riskScore is a computed value** that leverages the security provider's processing intelligence. By retrieving through the security profiles the riskscores associated with a user, a file, a host, the analyst will have a more precise idea of the overall risk level associated with this alert by taking advantage of the analysis of each security provider.

Other applications are possible that take advantage of the information included in the security profiles. For example, the OS version information in the Security profile can be used to check for system updates in a compliance context. Other more targeted solutions like Windows Analytics are available but the advantage of the Security API remains its capability to integrate in larger solutions.

Finally, access through a graph-based API – as for other Microsoft providers like Azure Active Directory for identities, Microsoft Intune for devices, Microsoft Office –, allows developers to easily reach additional context information while retaining the same paradigm of access.

Not all providers are required to implement security profiles. If a request is issued by an application to view a particular security profile, only the providers that implement it will return information back to this request.

> **More information** Five security profiles have been defined corresponding to user information (userSecurityProfile), hosts (hostSecurityProfile, file (fileSecurityProfile), IP addresses (IPSecurityProfile) and application (applicationSecurityProfile).) Each of these profiles contains more targeted information: for example, the hostSecurityProfile contains information such as the set of network interfaces (NICs) with their IPv4 and IPV6 addresses, the MAC address, the user who is logged on the machine, the OS version, and so on. For a full description of the data available for each Security profile, refer to the online schema description.

An automatable solution: Actions and Configuration

Data that flows through the Security API is mainly one-way: from the security providers to the applications that consume the alerts. We have seen previously that Security Profiles also allow consumers to retrieve contextual information.

But the detection of alerts may require the ability to perform remediation actions: The Security API allows to address the assets of the environment through the security providers with the implementation of the concept of *Actions*. Depending on the characteristics of the provider, the possibilities of actions will be different and related to the controlled assets.

For example, a simple REST API request will deny access to a file for a set of users with Azure Information Protection, will trigger the scan of a suspicious Windows 10 computer with Windows Defender ATP, impose the password change of a particular user with Azure Identity Protection, restrict a user's access to a cloud application with the Microsoft Cloud App Security CASB, or block an IP address on a partner's firewall solution.

The concept of *Configuration* is more global: this is no longer a question of modifying a particular parameter of a Security Profile, but of applying a security policy encompassing several parameters, for example the enforcement of a security policy on devices through Microsoft Intune.

The simplicity of access to the Security API and the different scripting languages that can be used like Python, Angular or Ruby, make it possible to consider a simple automation through the creation of "run books" to invoke the Security Profiles to look for additional information or remediation actions.

© 2018 Microsoft Corporation. All rights reserved.