Copilot for Security

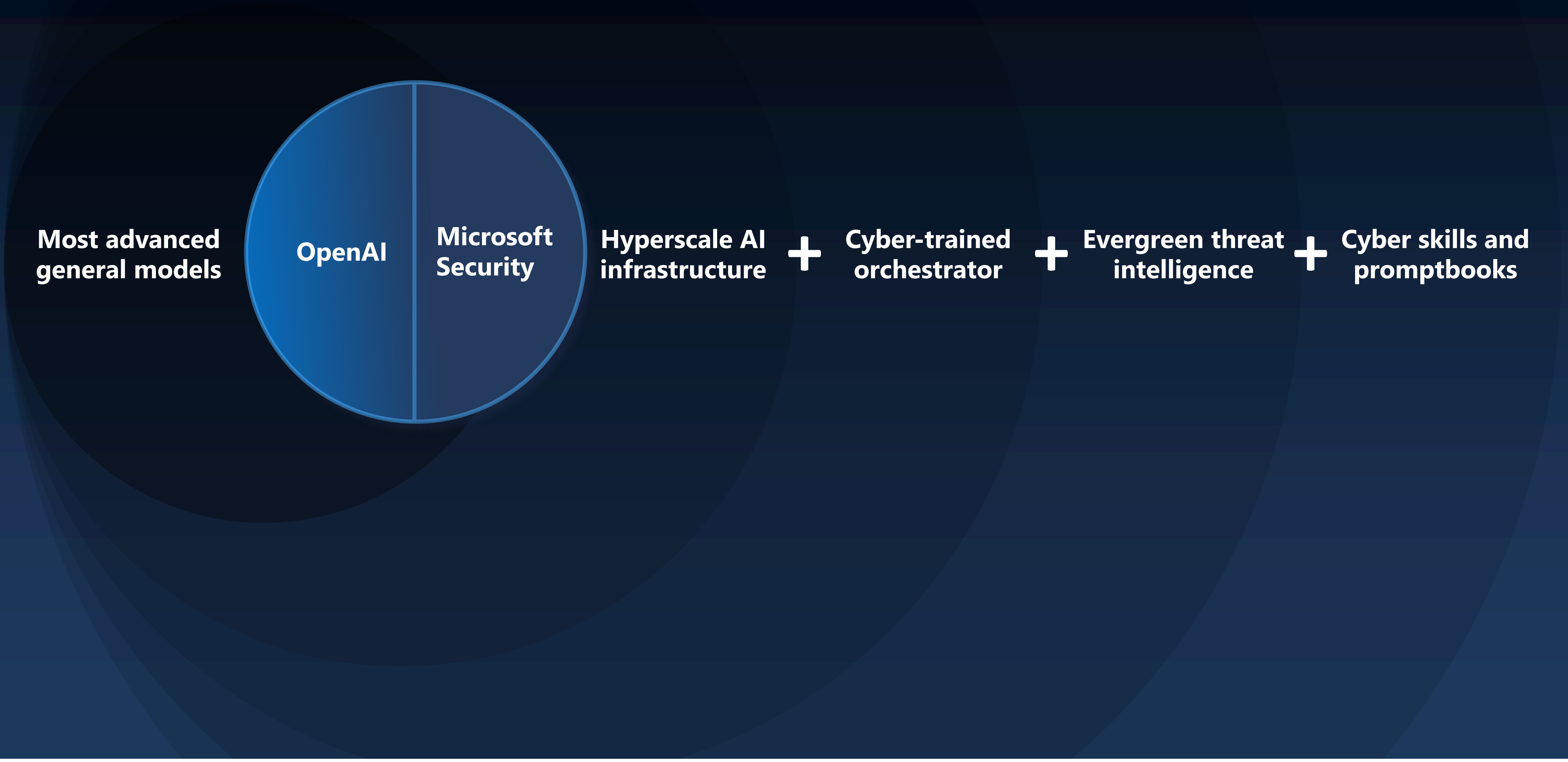
Agenda

- Copilot for Security Overview

- Extensibility
  - Plugins (API, KQL, GPT)
  - Knowledge Base Integtation
  - Promptbooks
  - Automated Workflows via Logic Apps

- Tips and Resources

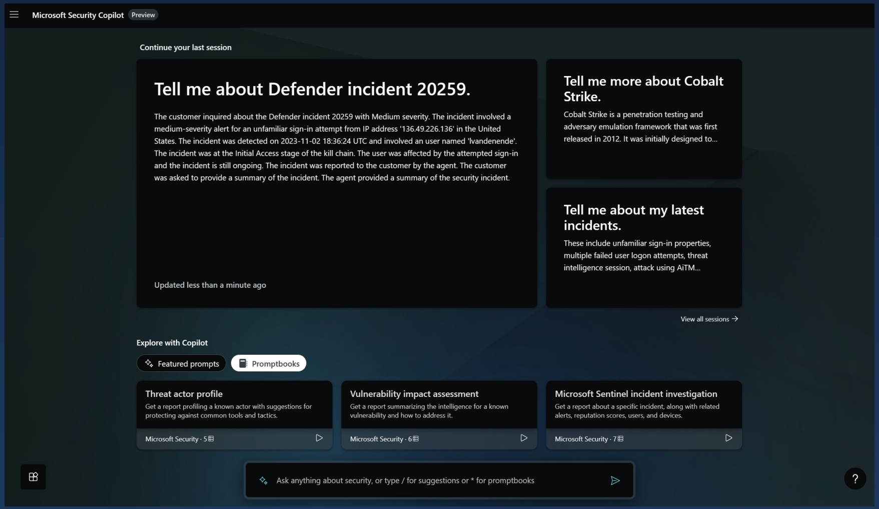
The Microsoft Copilot for Security advantage


Most advanced general models
OpenAI
Microsoft Security
Hyperscale AI infrastructure
+
Cyber-trained orchestrator
+
Evergreen threat intelligence
+
Cyber skills and promptbooks

Experiences to meet you where and how you work

Standalone

Helps teams gain a **broader context** to troubleshoot and remediate incidents faster within Security Copilot itself, with **all use cases in one place**, enabling **enriched cross-product guidance**.



Tell me about Defender incident 20259.

The customer inquired about the Defender incident 20259 with Medium severity. The incident involved a medium-severity alert for an unfamiliar sign-in attempt from IP address '136.49.226.136' in the United States. The incident was detected on 2023-11-02 18:36:24 UTC and involved an user named 'lvandenende'. The incident was at the Initial Access stage of the kill chain. The user was affected by the attempted sign-in and the incident is still ongoing. The incident was reported to the customer by the agent. The customer was asked to provide a summary of the incident. The agent provided a summary of the security incident.

Updated less than a minute ago

Tell me more about Cobalt Strike.

Cobalt Strike is a penetration testing and adversary emulation framework that was first released in 2012. It was initially designed to...

Tell me about my latest incidents.

These include unfamiliar sign-in properties, multiple failed user logon attempts, threat intelligence session, attack using AiTM...

Threat actor profile
Get a report profiling a known actor with suggestions for protecting against common tools and tactics.
Microsoft Security · 5

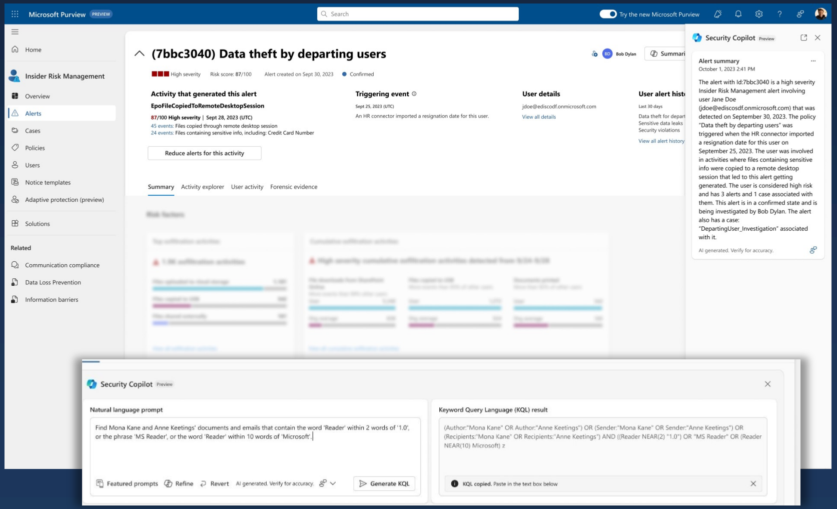
Vulnerability impact assessment
Get a report summarizing the intelligence for a known vulnerability and how to address it.
Microsoft Security · 6

Microsoft Sentinel incident investigation
Get a report about a specific incident, along with related alerts, reputation scores, users, and devices.
Microsoft Security · 7

Ask anything about security, or type / for suggestions or * for promptbooks

Embedded

Offers the **intuitive experience** of getting Security Copilot guidance **natively** within the products that your team members already work from and are familiar with.



(7bbc3040) Data theft by departing users

High severity   Risk score: 87/100   Alert created on Sept 30, 2023   Confirmed

Activity that generated this alert
EpoFileCopiedToRemoteDesktopSession
87/100 High severity | Sept 28, 2023 (UTC)
45 events: Files copied through remote desktop session
24 events: Files containing sensitive info, including: Credit Card Number

Triggering event
Sept 25, 2023 (UTC)
An HR connector imported a resignation date for this user.

User details
jdoe@ediscodf.onmicrosoft.com
View all details

User alert hist
Last 30 days
Data theft for depart
Sensitive data leaks
Security violations
View all alert history

Security Copilot Preview

Alert summary
October 1, 2023 2:41 PM

The alert with Id:7bbc3040 is a high severity Insider Risk Management alert involving user Jane Doe (jdoe@ediscosdf.onmicrosoft.com) that was detected on September 30, 2023. The policy "Data theft by departing users" was triggered when the HR connector imported a resignation date for this user on September 25, 2023. The user was involved in activities where files containing sensitive info were copied to a remote desktop session that led to this alert getting generated. The user is considered high risk and has 3 alerts and 1 case associated with them. This alert is in a confirmed state and is being investigated by Bob Dylan. The alert also has a case: "DepartingUser_Investigation" associated with it.

AI generated. Verify for accuracy.

Security Copilot Preview

Natural language prompt

Find Mona Kane and Anne Keetings' documents and emails that contain the word 'Reader' within 2 words of '1.0', or the phrase 'MS Reader', or the word 'Reader' within 10 words of 'Microsoft'.

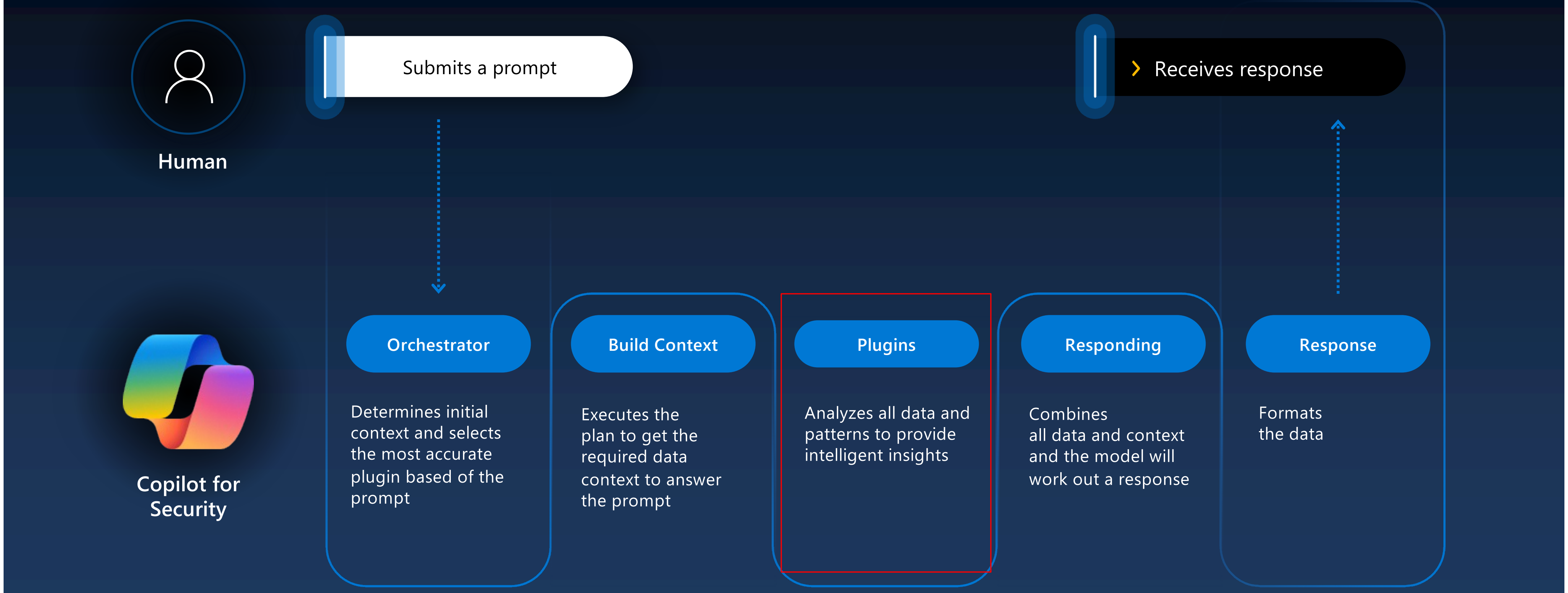
Featured prompts   Refine   Revert   AI generated. Verify for accuracy.   Generate KQL

Keyword Query Language (KQL) result

```
(Author:"Mona Kane" OR Author:"Anne Keetings") OR (Sender:"Mona Kane" OR Sender:"Anne Keetings") OR (Recipients:"Mona Kane" OR Recipients:"Anne Keetings") AND ((Reader NEAR(2) "1.0") OR "MS Reader" OR (Reader NEAR(10) Microsoft) z
```

KQL copied. Paste in the text box below

System Architecture



Human

Submits a prompt

Receives response

Copilot for Security

Orchestrator

Determines initial context and selects the most accurate plugin based of the prompt

Build Context

Executes the plan to get the required data context to answer the prompt

Plugins

Analyzes all data and patterns to provide intelligent insights

Responding

Combines all data and context and the model will work out a response

Response

Formats the data

Extensibility: tailor to YOUR business

Knowledge Base Integration

- Upload a file
- Azure AI Search

Plugins

- Microsoft
- Non-Microsoft (3rd Party)
- Custom

Promptbooks

- Templates of bundled prompts
- Customizable
- Manual Trigger

Logic Apps

- Automated workflows
- Trigger based
- Copilot for Security Connector

Demo

- User Experience
  - Embedded/Standalone
- Plugin UI
- Knowledge bases
- Promptbooks
- Logic Apps

Plugins

How can someone in Security Copilot leverage plugins?

→Pre-built plugins from 1st & 3rd Party applications & services
→Invoke **skills** within **plugins** using natural-language prompts or direct skill invocation
→Enable Plugins for all users or just you; Admins have the control!

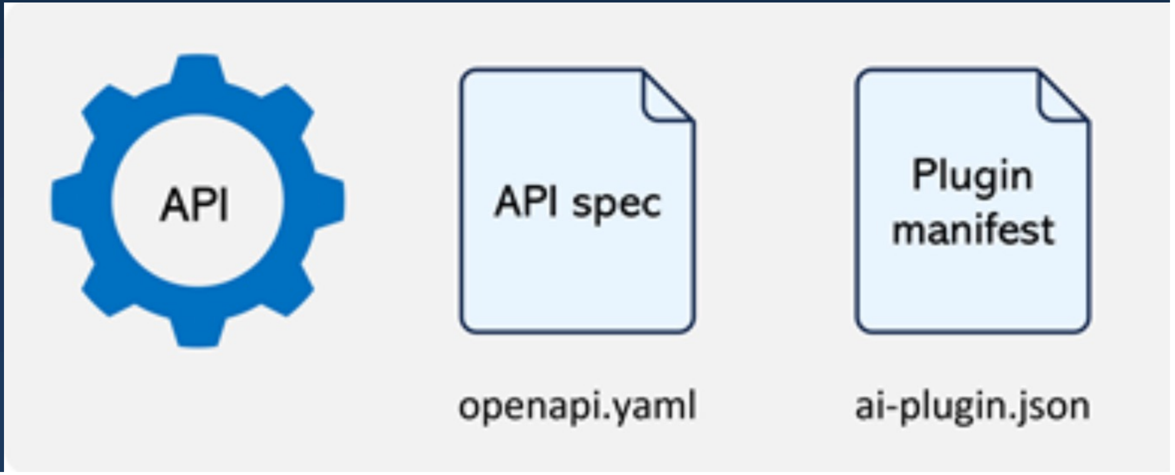
How can I build/leverage plugins?

→You can build your own plugins (API, GPT, KQL) to enrich your investigations
→Bring existing OpenAI plugins to or build your own using what you learn today ☺
→Encourage your favorite SaaS tools & ISVs to build integrations with Security Copilot

Custom Plugin architecture

OpenAPI Specification:
This component describes how the API of the plugin works. It defines the endpoints, request and response formats, and any other relevant details about the plugin's API.

Plugin Manifest:
The plugin manifest file is used to explain to Copilot how to use the plugin. It provides metadata and configuration information about the plugin, such as its name, version, author, dependencies, and authentication.



Demo

- API- Plugin – Shodan IP Info
- KQL Plugin - Sentinel Cost
- KQL Plugin – GCP Custom Log

Tip and tricks

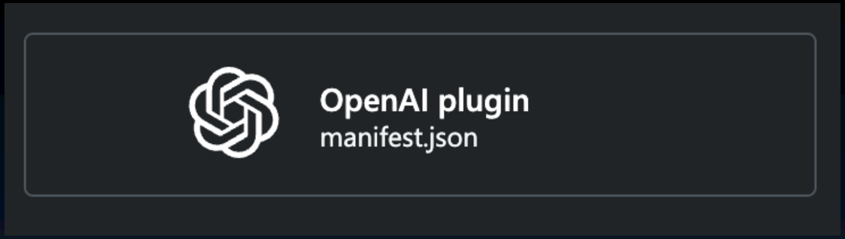
- Only retrieve what you need – save your tokens $$$
  - Response length (max token size)
- Use Example prompts in the description
- Use CI/CD to keep versioning
- Validate network connectivity and Copilot IP's allow list
- Get familiar with APIs and authentication, for example by using Postman
- API Terms and conditions, costs

Resources

- Public Doc - Plugins overview Microsoft Copilot for Security | Microsoft Learn
- How to build a Copilot for Security API Plugin - How to build a Copilot for Security API Plugin – Part 1 - Microsoft Community Hub
- Microsoft Copilot for Security Repo – link
- Join the Copilot for Security community: https://aka.ms/JoinCCP (requires signup and NDA)
  - Webinar- Link

API Plugins

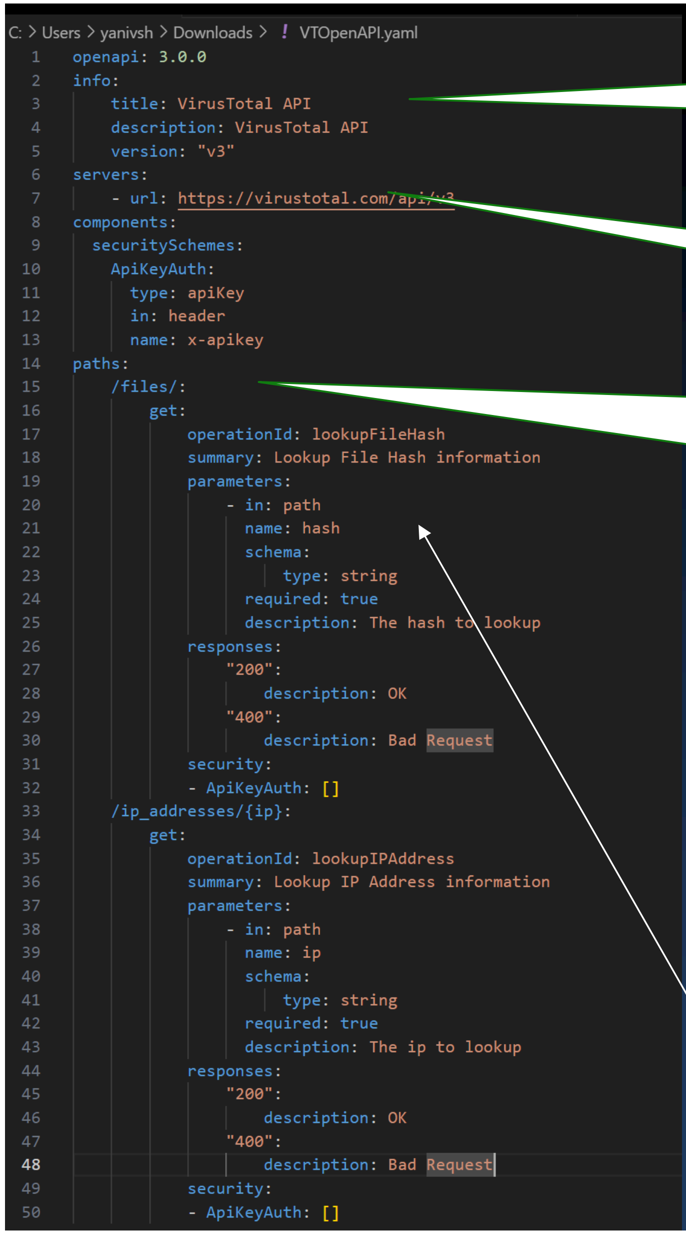
API Plugins



- API Plugins enable you to utilize existing interfaces and bring them to Copilot for Security!

- Support for many OpenAI Plugin Bring any OpenAI plugin to Security Copilot, today!
- Bring some existing APIs Security Copilot Plugins (Note: building purpose-built endpoints recommended)
- Various supported authentication mechanisms (Basic, API Key, Oauth, Entra ID, etc.)
- Customizable endpoint URL support

API Type Plugin



C: > Users > yanivsh > Downloads > ! VTOpenAPI.yaml

```yaml
openapi: 3.0.0
info:
    title: VirusTotal API
    description: VirusTotal API
    version: "v3"
servers:
    - url: https://virustotal.com/api/v3
components:
  securitySchemes:
    ApiKeyAuth:
      type: apiKey
      in: header
      name: x-apikey
paths:
    /files/:
        get:
            operationId: lookupFileHash
            summary: Lookup File Hash information
            parameters:
                - in: path
                  name: hash
                  schema:
                      type: string
                  required: true
                  description: The hash to lookup
            responses:
                "200":
                    description: OK
                "400":
                    description: Bad Request
            security:
            - ApiKeyAuth: []
    /ip_addresses/{ip}:
        get:
            operationId: lookupIPAddress
            summary: Lookup IP Address information
            parameters:
                - in: path
                  name: ip
                  schema:
                      type: string
                  required: true
                  description: The ip to lookup
            responses:
                "200":
                    description: OK
                "400":
                    description: Bad Request
            security:
            - ApiKeyAuth: []
```

In the **info object** is like a summary of key details about the API.
It usually includes things like the title, a brief description, the version number, and links to stuff like licenses and terms of service.
The only things that are required in there are the title and version

In the servers object, we can list one or more main paths used in requests to the API.
This section is in array format, so each path needs to be specified separately, unlike the values in the info section.
The basepath is the part of the web address that comes before the specific endpoint. The only required property is the url

The Paths object
The paths object can be seen as a roadmap for the endpoints available within the API.
It provided details on how to access the endpoints and what actions you can perform.
Each endpoint is represented by a unique path within the paths object
To break this section a bit more down into digestible chunks, I will describe the most important parts.

- path
- HTTP methods
- operation object …

Operation Object
For each HTTP method specified, there must be an associated operation object containing additional information about the operation.
Examples of these operation objects are parameters, request body, responses, etc.

The only required values are summary and the operation object responses
The summary field provides a brief, human-readable description of the endpoint.

In the responses section, we define the possible responses that the API can return when that endpoint is accessed. It contains the HTTP status codes and their corresponding descriptions.
Optionally, the responses section includes details about the response payload, like the format and schema definition.

Parameter Object
You don't have to use the parameter object, but it's handy when making a plugin for Copilot(s).
If you include the parameter object in the path item, the parameters will affect all actions on that path.
Alternatively, you can place parameters in the operations object, where they'll only affect that specific action.

Depending on the API, the parameters can reside in different locations, indicated by the in field.

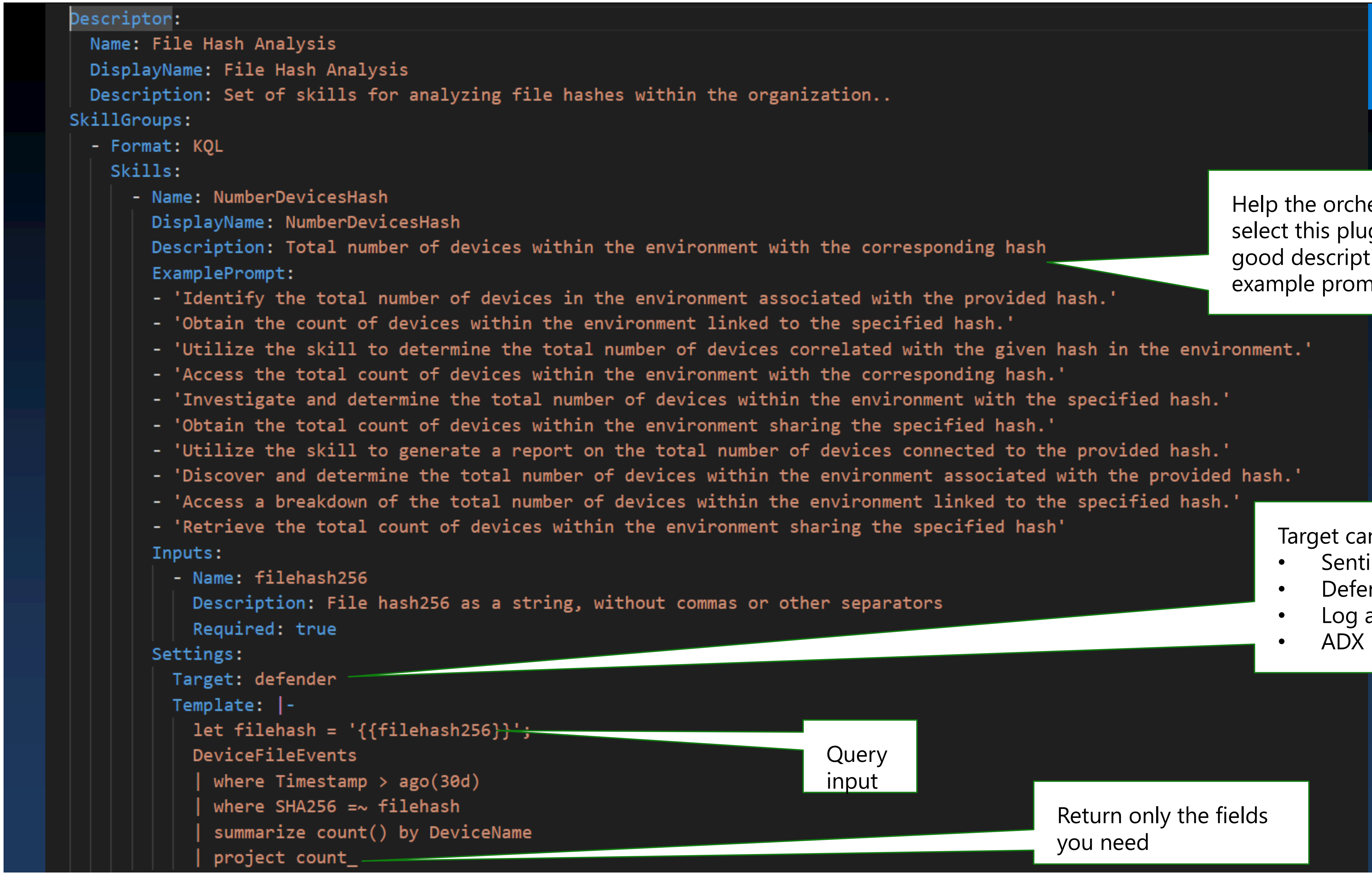
KQL Plugins

KQL Plugins

KQL Plugins enable users to analyze & query datasets at-scale, while having the ability to store useful queries as skills.

- Skills can be packaged as Kusto Query Language (KQL) queries to be run against a Kusto database, a Sentinel or Log Analytics instance or against Microsoft 365 Defender Advanced Hunting.
- Many customers have over-time developed libraries of hundreds of saved queries for their common query patterns. KQL Skills allow them to be easily turned into skills that Security Copilot can use.
- Tldr; powerful query language to analyze large volumes of data. Enable users to find trends, anomalies, patterns, etc.

KQL Type Plugin



```
Descriptor:
  Name: File Hash Analysis
  DisplayName: File Hash Analysis
  Description: Set of skills for analyzing file hashes within the organization..
SkillGroups:
  - Format: KQL
    Skills:
      - Name: NumberDevicesHash
        DisplayName: NumberDevicesHash
        Description: Total number of devices within the environment with the corresponding hash
        ExamplePrompt:
        - 'Identify the total number of devices in the environment associated with the provided hash.'
        - 'Obtain the count of devices within the environment linked to the specified hash.'
        - 'Utilize the skill to determine the total number of devices correlated with the given hash in the environment.'
        - 'Access the total count of devices within the environment with the corresponding hash.'
        - 'Investigate and determine the total number of devices within the environment with the specified hash.'
        - 'Obtain the total count of devices within the environment sharing the specified hash.'
        - 'Utilize the skill to generate a report on the total number of devices connected to the provided hash.'
        - 'Discover and determine the total number of devices within the environment associated with the provided hash.'
        - 'Access a breakdown of the total number of devices within the environment linked to the specified hash.'
        - 'Retrieve the total count of devices within the environment sharing the specified hash'
        Inputs:
          - Name: filehash256
            Description: File hash256 as a string, without commas or other separators
            Required: true
        Settings:
          Target: defender
          Template: |-
            let filehash = '{{filehash256}}';
            DeviceFileEvents
            | where Timestamp > ago(30d)
            | where SHA256 =~ filehash
            | summarize count() by DeviceName
            | project count_
```

Help the orchestrator to select this plugin with good description and example prompts

Target can be:
- Sentinel
- Defender
- Log analytics
- ADX

Query input

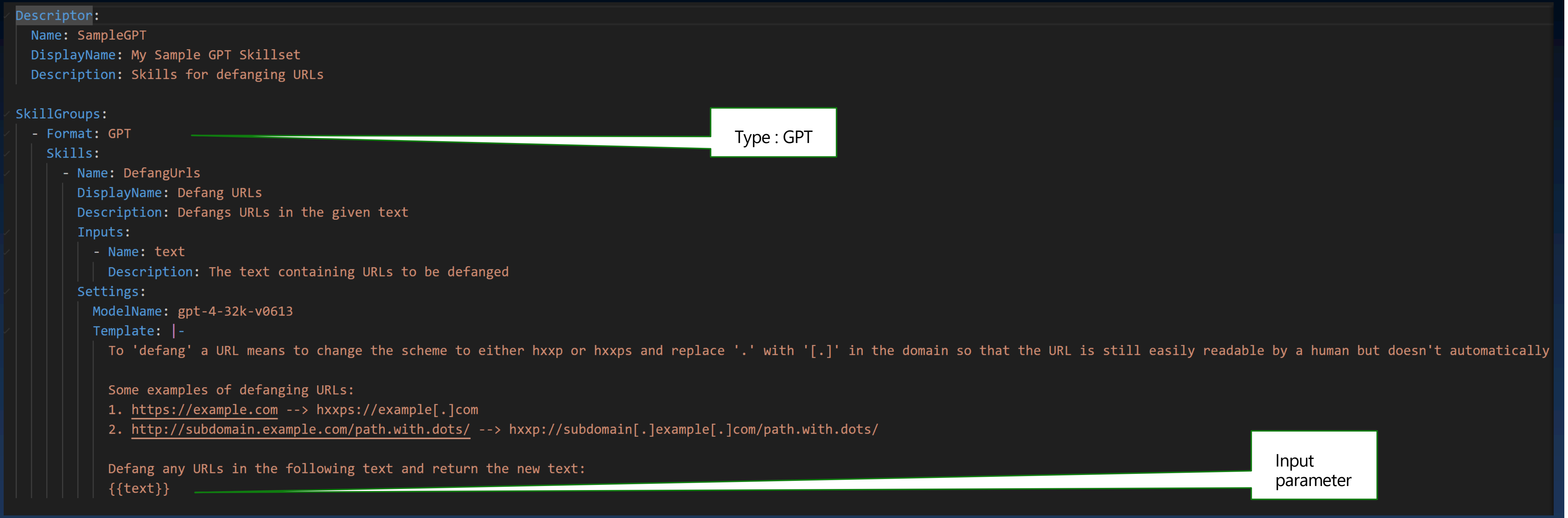
Return only the fields you need

GPT Plugins

GPT Plugins can host skills expressed as templatized GPT prompts and harness the power of GPT4

- A skill to rewrite text following some particular guidelines → defang, summarize, structured reports
- A skill to turn a set of criteria expressed in natural language into various types of filters (Odata, NL2KQL, etc.)

GPT Type Plugin



```yaml
Descriptor:
  Name: SampleGPT
  DisplayName: My Sample GPT Skillset
  Description: Skills for defanging URLs

SkillGroups:
  - Format: GPT
    Skills:
      - Name: DefangUrls
        DisplayName: Defang URLs
        Description: Defangs URLs in the given text
        Inputs:
          - Name: text
            Description: The text containing URLs to be defanged
        Settings:
          ModelName: gpt-4-32k-v0613
          Template: |-
            To 'defang' a URL means to change the scheme to either hxxp or hxxps and replace '.' with '[.]' in the domain so that the URL is still easily readable by a human but doesn't automatically

            Some examples of defanging URLs:
            1. https://example.com --> hxxps://example[.]com
            2. http://subdomain.example.com/path.with.dots/ --> hxxp://subdomain[.]example[.]com/path.with.dots/

            Defang any URLs in the following text and return the new text:
            {{text}}
```