Security Copilot Preview

Featured Capability: Generate Hunting Queries

Introduction

Advanced Hunting in Security Copilot

Security Copilot is now enabled with purpose-built skills that leverage the generative AI capabilities of GPT-4 to reason over a natural-language prompt and generate a KQL query capable of retrieving data from Advanced Hunting tables in Microsoft Defender and top data tables in Microsoft Sentinel. See the full list of supported tables in Appendix.

This new functionality replaces an earlier implementation with a vastly improved algorithm and access to the latest Azure OpenAI models. Research is underway that would extend the functionality to additional data tables in Sentinel.

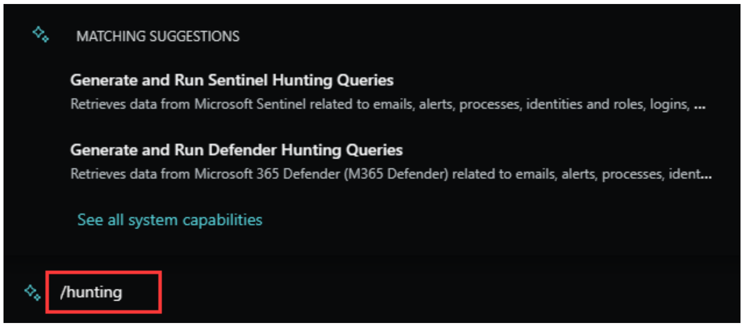
Getting Started

Prerequisites

1. Access to Microsoft Security Copilot
2. Access to Defender Advanced Hunting and Microsoft Sentinel with the same credentials used to access Security Copilot.
3. Access to the necessary features and licenses in Defender and Sentinel to activate all supported tables.

Activating the Skillset

- Enable “Natural language to Defender 365 KQL” and “Natural language to Sentinel KQL” in the admin console under “My connections”.
- In the prompt box, type “/hunting” to select the skill explicitly before entering a prompt



MATCHING SUGGESTIONS

Generate and Run Sentinel Hunting Queries
Retrieves data from Microsoft Sentinel related to emails, alerts, processes, identities and roles, logins, ...

Generate and Run Defender Hunting Queries
Retrieves data from Microsoft 365 Defender (M365 Defender) related to emails, alerts, processes, ident...

See all system capabilities

/hunting

- You can also try out the “EnableNL2KQLOrchestrator” feature flag to enable NL2KQL as a default skill.

Use Case Scenarios

You can start the investigation from an incident and ask follow up questions:

- Tell me about incident [*incidentID*]
- What are the recent logon events from the user in the alert?
- Give me all the recent AAD logins from the IP address
- Did the user login from outside the US lately?
- Did the user have any alerts?
- What are the other entities associated with this user?

You can also ask questions such as:

- Show me a list of the latest URL clicks
- Find all users who have clicked on a link in the past day
- Show me recent sign in events from [*IPaddress*]
- Give me the number of logons per user in the last 12 hrs by Location and sort it descending
- Is there a file in my organization with the name [*file name, e.g. notepad.exe*]
- Give me a list of 10 critical software vulnerabilities

Best Practices for Writing Prompts to Generate Hunting Queries

General guidelines when you write prompts to generate hunting queries:

1. **Be unambiguous:** try to ask questions with a clear subject. For example, "logins" could mean device logins or cloud logins.
2. **Ask one question at a time:** ask for a single task/type of information at a time when possible. Don't expect the model to perform several unrelated tasks at once. You can always ask follow up questions instead of combining unrelated asks into a single response.
3. **Be specific:** if you know anything about the data you are looking for, help provide that information in your question.
4. **Use other skills:** You can use Security Copilot's other skills to help gather information that might provide useful context while writing hunting queries

1. Be unambiguous: avoid ambiguous phrases or terms

Ambiguous	**Good**	**Explanation**
Have any users logged in from suspicious locations in the last week?	Which individual accounts had suspicious or impossible travel activities in the last week?	"Suspicious locations" is objective and might not give as consistent results.

Which users had the most failed login attempts in the last 72h?	Show me the 5 users with the most failed login attempts to local devices in the last 72h?	The better prompt specifies scope (5 users) and clarifies that we are looking for local device logins instead of cloud logins.
Network activity from suspicious Ips	Network activity from known proxy or tor-associated IP addresses	Defining what you consider suspicious is almost always better than letting the model decide.

2. Ask one question at a time

Ask for a single task/type of information at a time when possible. Don't expect the model to perform several unrelated tasks at once. You can always ask follow up questions instead of combining unrelated asks into a single response.

Bad	**Good**	**Explanation**
How many devices are not compliant and what are the most common security vulnerabilities on them?	How many devices are not compliant?	Breaking the question up allows Copilot to focus on each task independently and deliver a better, more focused response.
	What are the most common security vulnerabilities on non-compliant devices?	

3. Be specific

If you know anything about the data you are looking for, help provide that information in your question.

Bad	**Good**	**Explanation**
List devices where Teams is set to open on startup.	Search the registry for devices where Teams is set to open on startup.	By telling the model to consider the registry, we can "jumpstart" its search for relevant data and help it identify the most useful tables available.

4. Use other skills

You can use Security Copilot's other skills to help gather information that might provide useful context while writing hunting queries.

Bad	Good
What is the CVE for "XYZ" and which devices are affected.	Use the "FindThreatIntelligence" skill and follow up by asking for a query for affected devices
Show me users involved in incident 1234	Use the "GetIncidentEntities" skill and follow up by asking for a query for other information about those users or alerts.

Feedback

To help us improve, it is important that you share feedback with us. Please use the feedback buttons in the product to provide feedback so we can understand whether the model generated the right queries for you.

- If the query is correct and the results are as expected, please select “**Confirm**”.
- If the LookupDataFromDefender365Hunting skill is not being used properly, please use “**Off-target**” as described above.
- If the skill is run as expected but the query is incorrect, please use “**Off-target**”. It would be great if you provide details on why the query is incorrect and an example of the correct query.
  For example:
  - NL prompt: Get devices with high exposure level
  - Query generated by the model:

    ```
    DeviceRegistryEvents
    | where ActionType == "Create"
    | where InitiatingProcessIntegrityLevel == "High"
    | summarize count() by DeviceId, DeviceName
    ```

  When you share feedback, please let us know it’s off-target and share an example of the correct query as well as the failure category.

  KQL ground truth

  ```
  DeviceInfo
  | summarize arg_max(Timestamp, *) by DeviceId
  | where ExposureLevel == "High"
  ```

  Failure category: Wrong table

  Here are some common failure categories you can use while sharing feedback:

Failure categories
Wrong table
Wrong column, right table
Correct schema, but column values are incorrect.
No query, model returns error
Too much details (for example redundant aggregation/rendering/project at the end)
Close enough, minor edit
Wrong intent

Appendix

Supported Defender table list

All Defender Advanced Hunting tables are supported by NL2KQL. Please note that the AdditionalFields columns are not supported.

Supported Sentinel table list

The skill might still generate KQL queries for the Sentinel tables not included in the list below. The KQL queries generated for these tables are a best-effort attempt and may come with a lower level of confidence.

Updated 11/10

AADManagedIdentitySignInLogs
AADNonInteractiveUserSignInLogs
AADProvisioningLogs
AADRiskyUsers
AADServicePrincipalSignInLogs
AADUserRiskEvents
ABAPAuditLog_CL
Anomalies
AppDependencies
AppTraces
AuditLogs
AWSCloudTrail
AWSGuardDuty
AzureActivity
AzureDevOpsAuditing
AzureDiagnostics
AzureMetrics
BehaviorAnalytics
CommonSecurityLog
ContainerInventory
ContainerLog
DnsEvents
Dynamics365Activity
Event
Heartbeat
IdentityInfo
InsightsMetrics
IntuneAuditLogs

IntuneDevices
LAQueryLogs
MicrosoftAzureBastionAuditLogs
MicrosoftPurviewInformationProtection
OfficeActivity
Perf
PowerBIActivity
ProtectionStatus
SecurityAlert
SecurityEvent
SecurityIncident
SecurityRecommendation
SigninLogs
SqlAtpStatus
StorageBlobLogs
StorageFileLogs
Syslog
ThreatIntelligenceIndicator
Update
UrlClickEvents
Usage
UserAccessAnalytics
UserPeerAnalytics
VMBoundPort
VMComputer
VMConnection
VMProcess
W3CIISLog
WindowsEvent
WindowsFirewall