Configuring Your Sentinel and Defender Environment

By Grant Knoetze

What is Microsoft Sentinel

- It is a Security Incident and Event Management Tool.
- It allows an organization to collect data (logs), analyse, and perform security operations on its computer systems, that can be hardware applications, applications, or both.

What is Microsoft Sentinel

- In its simplest form, a SIEM solution allows you to:
  - Collect and query logs.
  - Do correlation and anomaly detection.
  - Create alerts and incidents based on your findings.
- A SIEM solution can also:
  - Perform log management – the ability to collect, store, and query the log data from resources within your environment.
  - Alerting – A proactive look inside the log data for potential security incidents and anomalies.
  - Visualization – graphs and dashboards that provide visual insights into your log data.

What is Microsoft Sentinel

- Incident Management – The ability to create, update, assign, and investigate incidents that have been identified.
- Querying data – Using KQL, you can query and understand your data.

What is Microsoft Sentinel

- A cloud native SIEM system that a security operations team can use to:
  - Get security insights across the enterprise by collecting data from virtually any source.
  - Detect and investigate threats quickly by using built in machine learning and Microsoft threat intelligence.
  - Automate threat responses by using playbooks and by integrating Azure log apps.

What is Microsoft Sentinel

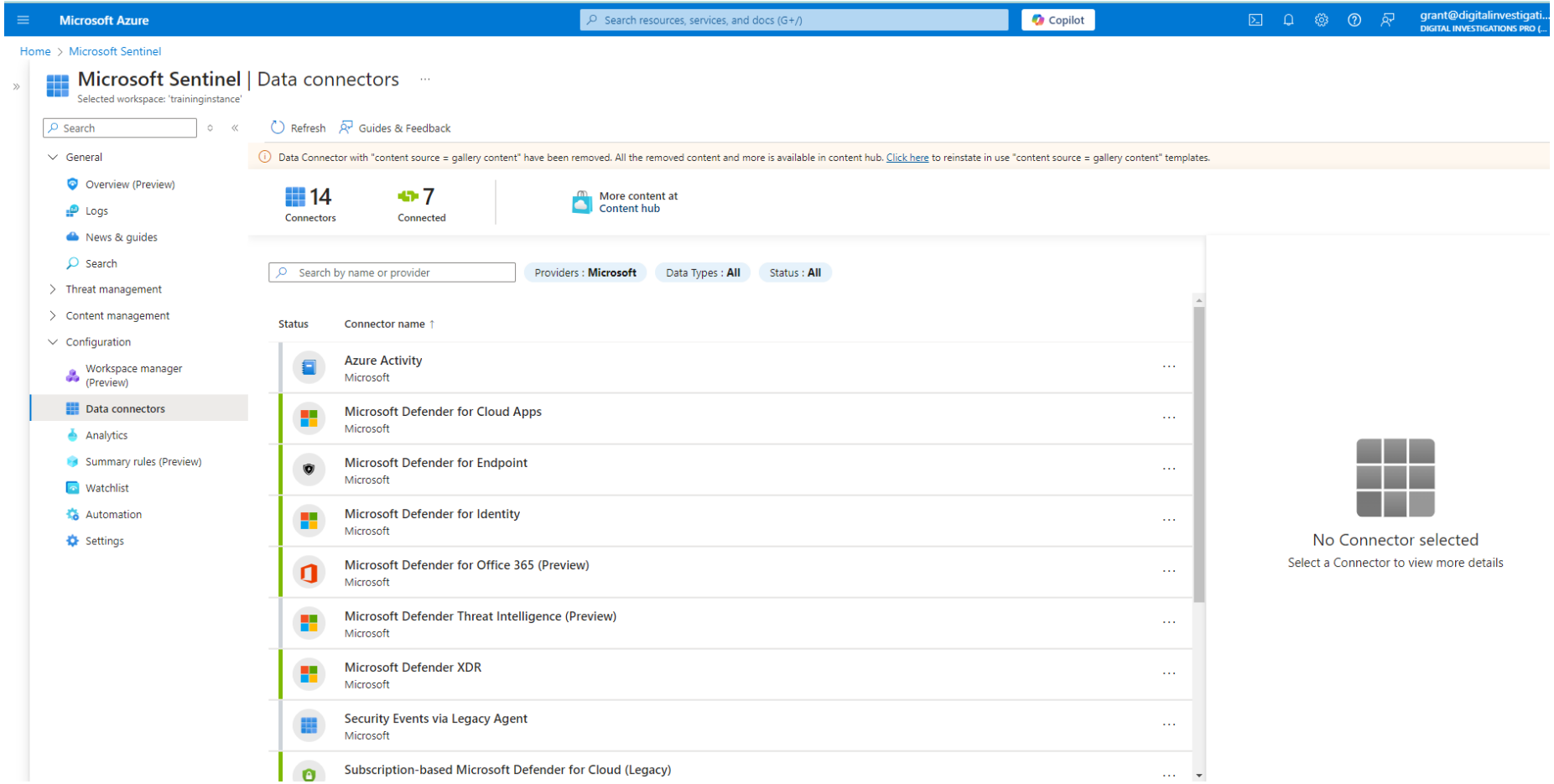
- Unlike with traditional SIEM solutions, you don't need to install any servers either on-premises or in the cloud to run Microsoft Sentinel. Microsoft Sentinel is a service that you deploy in Azure. You can get up and running with Sentinel in just a few minutes in the Azure portal.

What is Microsoft Sentinel

- Microsoft Sentinel is tightly integrated with other cloud services. Not only can you quickly ingest logs, but you can also use other cloud services natively (for example, authorization and automation).
- Microsoft Sentinel helps you enable end-to-end security operations including collection, detection, investigation, and response:

What is Microsoft Sentinel

What is Microsoft Sentinel: Data Connectors



What is Microsoft Sentinel: Data Connectors

- The first thing to do is to have your data ingested into Microsoft Sentinel. Data connectors let you do just that. You connect Data connectors by first installing Content hub solutions. Once installed, you can add some services, such as Azure activity logs, just by selecting a button. Others, such as syslog, require more configuration. There are data connectors that cover all scenarios and sources, including but not limited to:

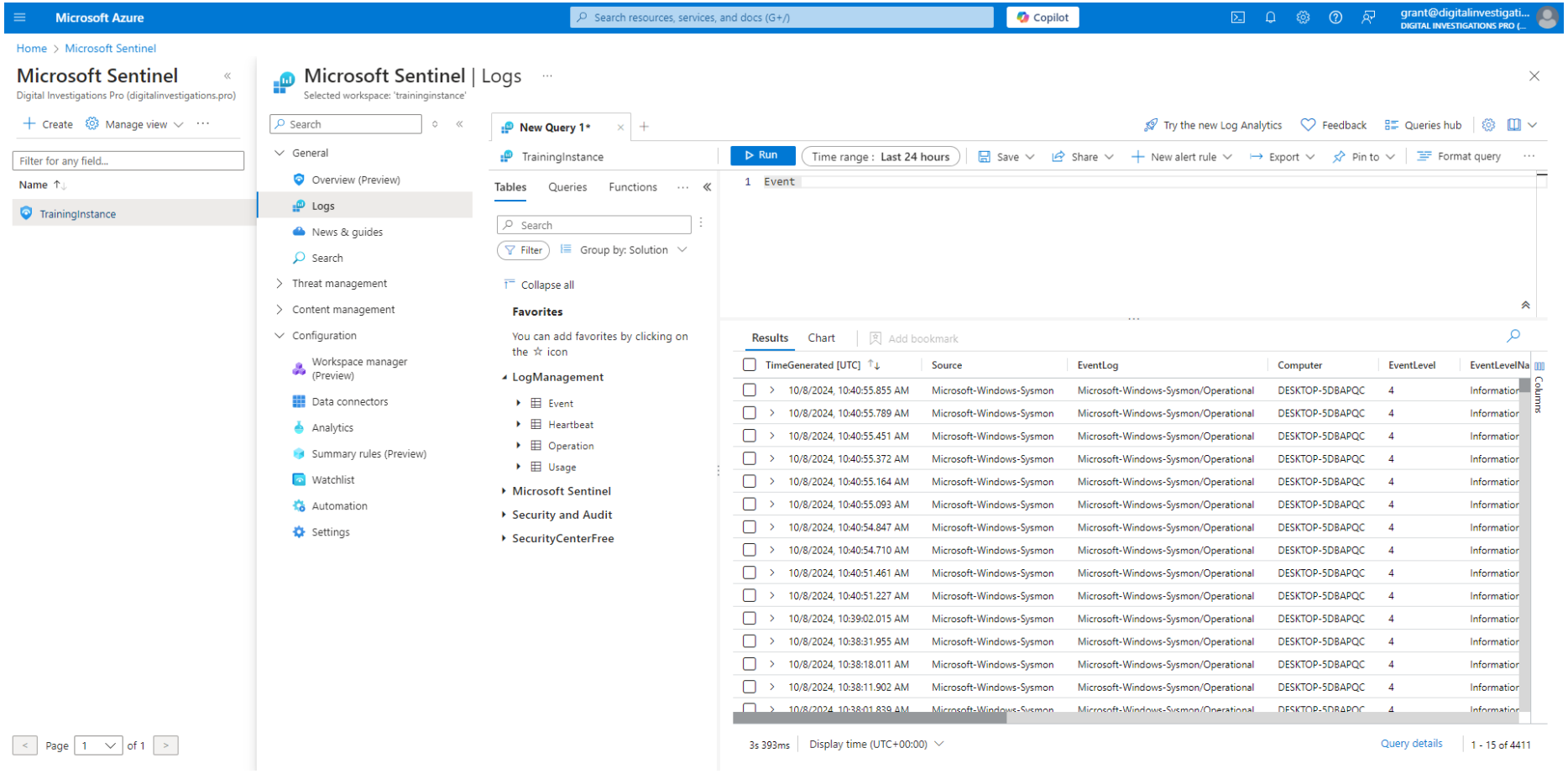
What is Microsoft Sentinel: Data Connectors

- syslog
- Common Event Format (CEF)
- Trusted Automated eXchange of Indicator Information (TAXII) (for threat intelligence)
- Azure Activity
- Microsoft Defender services
- Amazon Web Services (AWS) and Google Cloud Platform (GCP)

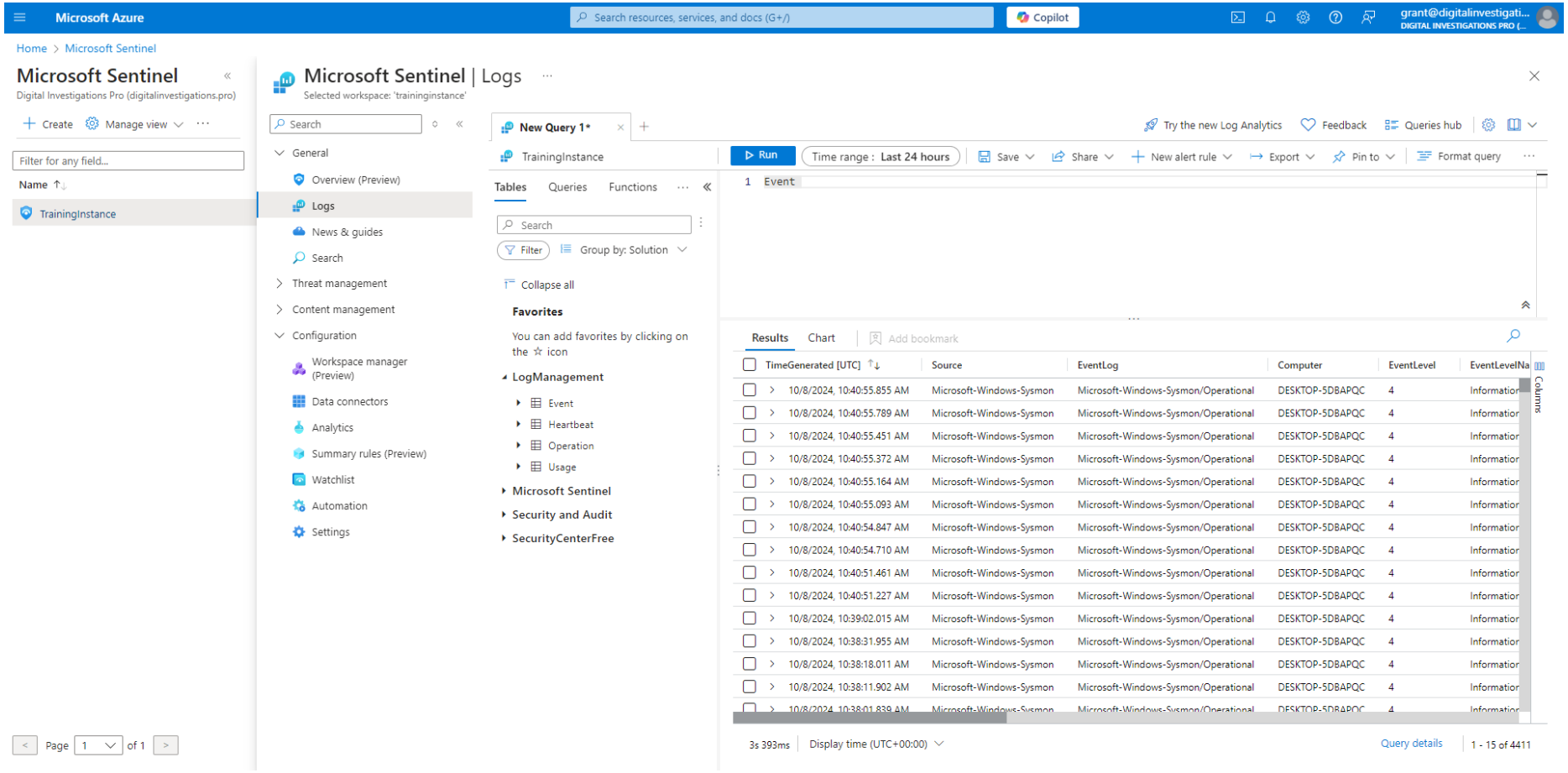
Log Retention

- After data is ingested into Microsoft Sentinel, the data is stored in the Log Analytics workspace. The benefits of using Log Analytics include the ability to use the Kusto Query Language (KQL) to query your data. KQL is a rich query language that gives you the power to dive into and gain insights from our data.

Log Retention



Log Retention

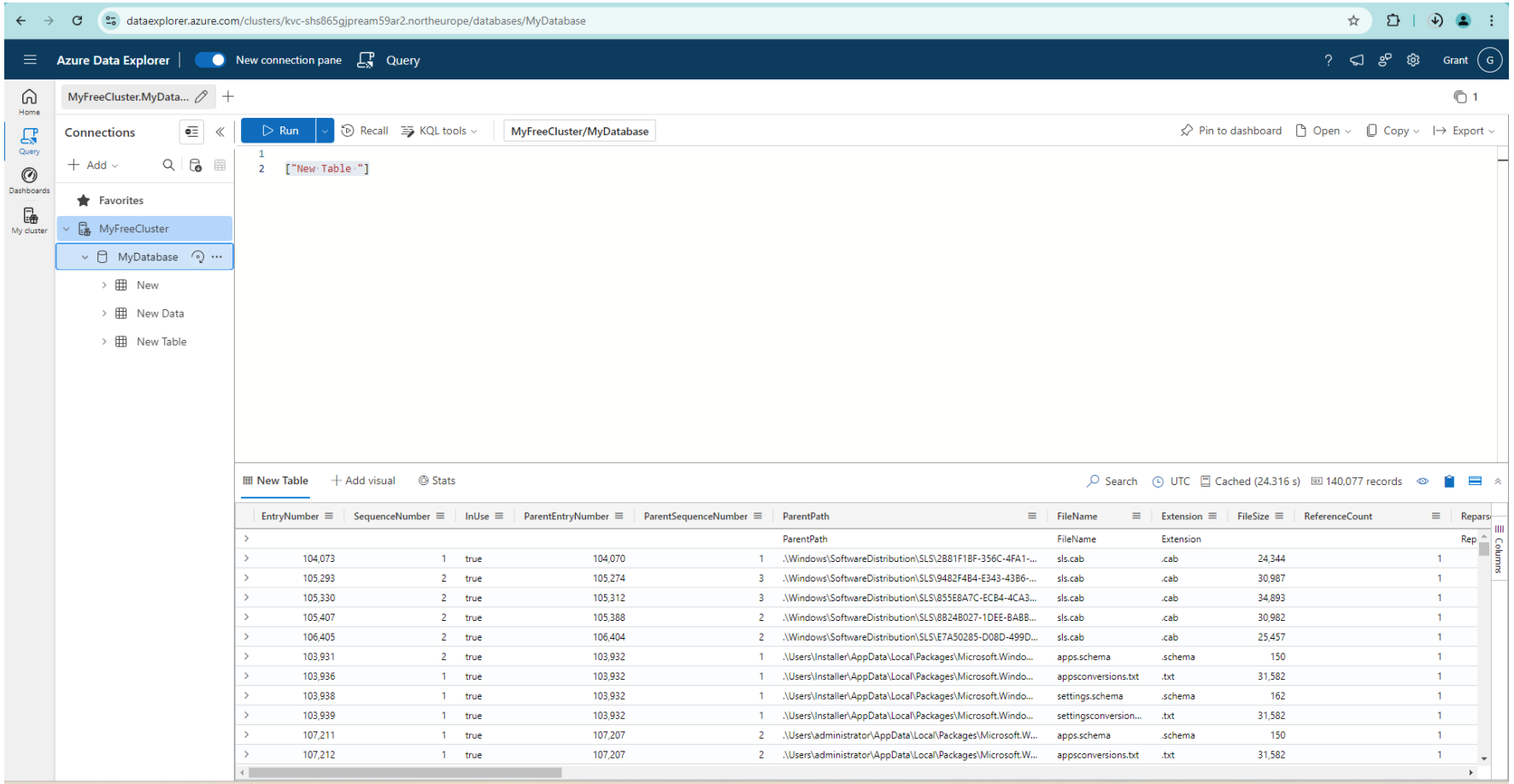


Azure Data Explorer

- Unless you have access to Azure, use the data explorer online free tool and upload the logs there:

https://dataexplorer.azure.com/clusters/kvc-shs865gjpream59ar2.northeurope/databases/MyDatabase

Azure Data Explorer



Onboarding in Defender for Advanced Hunting

• Device Discovery and Onboarding is the process of connecting devices to Microsoft Defender for Endpoint.

Go to:

Settings -> Endpoints -> Device Management -> Onboarding.

->Select the Operating System Type

->Select the Connectivity Type

->Select the Deployment Method

Search for Endpoints Communicating with a Specific Domain: Mitre Attack TTP: T1059.005

Malware communicates with command and control servers and malicious domains, use the following KQL query to detect what machines are communicating with a specific domain:

Search for Endpoints Communicating with a Specific Domain

- **Query:**

```
let Domain = “http://domain.com”;
DeviceNetworkEvents
| where Timestamp > ago(7d) and RemoteUrl contains Domain
| project Timestamp, DeviceName, RemotePort, RemoteUrl
| top 100 by Timestamp desc
```

Search for Created Scheduled Tasks

Scheduled Tasks are often used by attackers as a means of maintaining persistence. Use the following Query to search for all created scheduled tasks.

Search for Created Scheduled Tasks

```
Query

DeviceProcessEvents

| where FolderPath endswith "\\schtasks.exe" and

ProcessCommandLine has

"/create " and AccountName != "system"

| where Timestamp > ago(7d)
```

Malicious Documents / Macros that tried to connect to a suspicious domain or website:

This query will identify any malicious documents that have been used to connect to a malicious domain.

Malicious Documents / Macros that tried to connect to a suspicious domain or website:

```
DeviceProcessEvents
| where ActionType == "ProcessCreated" and InitiatingProcessFileName endswith ".exe"
| where InitiatingProcessFileName in~ ("winword.exe", "excel.exe", "powerpnt.exe")
| where not (InitiatingProcessCommandLine contains ".microsoft.com" or InitiatingProcessCommandLine contains ".live.com" or InitiatingProcessCommandLine contains ".office.com")
| project InitiatingProcessFileName, InitiatingProcessCommandLine
```

Or

```
// Define the DeviceProcessEvents part of the query

let processEvents = DeviceProcessEvents

| where ActionType == "ProcessCreated"

| where InitiatingProcessFileName in~ ("winword.exe",

"excel.exe", "powerpnt.exe");
```

Or

```
// Define the DeviceNetworkEvents part of the query
let networkEvents = DeviceNetworkEvents
| where not (RemoteUrl has ".microsoft.com" or RemoteUrl has ".live.com" or RemoteUrl has ".office.com")
| project Timestamp, DeviceName, RemotePort, RemoteUrl, InitiatingProcessFileName, InitiatingProcessCommandLine;
```

Or

```
// Join the two parts

processEvents

| join kind=inner (networkEvents) on InitiatingProcessFileName

| project Timestamp, DeviceName, RemotePort, RemoteUrl,

InitiatingProcessFileName, InitiatingProcessCommandLine
```