KQL Mini Threat-Hunting Toolkit Write-Up

Overview

This lab involved designing a custom threat hunting toolkit using Kusto Query Language (KQL) within a simulated Microsoft Sentinel environment. The goal was to build a library of practical, real-world detection queries that a risk analyst could use to identify potential security incidents.

This hands-on project simulated how a risk professional would:

- Translate attack behaviors into KQL logic,
- Test detection queries against real data,
- Document queries for reproducibility and actionability.

Objectives

- Build a toolkit of reusable queries aligned with real-world threat scenarios.
- Gain practical fluency in KQL, including joins, filters, summarization, and aggregation.
- Practice hunting across SigninLogs_CL and SecurityEvent_CL data.
- Validate each query through testing and schema exploration.
- Deliver results in a clean, organized Markdown format to simulate SOC documentation standards.

Tools & Environment

- Microsoft Sentinel Content Hub
  - The lab environment was provisioned using the Microsoft Sentinel Training Lab from the Content Hub in Azure. This provided a full Sentinel workspace with sample log data for SigninLogs_CL, SecurityEvent_CL, and other relevant tables.
- KQL Explorer
  - Queries were developed and tested within the Logs section of Microsoft Sentinel. Extensive schema exploration was performed by expanding the table structure in the GUI to determine available fields and column types.
- Schema Validation
  - Each query was validated by inspecting the table schemas, identifying columns like UserPrincipalName_s, Location_s, Activity_s, and EventID_s, and adjusting queries to ensure compatibility with the dataset. This prevented common issues like querying non-existent fields.

Toolkit Summary

The toolkit includes 11 working and tested KQL queries, each with a defined detection objective:

1. Brute-Force Login Detection - Detects users with 3+ failed login attempts within one hour from the same IP.
2. Abnormal Login Times - Flags logins that occur outside the standard working window of 8:00 AM – 5:00 PM.
3. Potential File Exfiltration or Recon - Identifies users generating high volumes of sensitive access or process creation events.
4. Special Privilege Assignment - Detects accounts assigned elevated privileges via Event ID 4672.
5. Scheduled Task Creation - Displays creation of scheduled tasks (Event ID 4698), which may indicate persistence.
6. Account Lockout or Failed Login Summary - Broad view of failed logins (Event ID 4625) grouped by user and machine.
7. Audit Policy or Log Tampering - Detects suspicious events like log clearings or audit policy changes.
8. User Account Lifecycle Monitoring - Monitors for account creation, deletion, or enabling (Event IDs 4720, 4722, 4726).
9. High Volume Process Creation - Detects accounts creating large numbers of processes in a short period (Event ID 4688).
10. Suspicious Script Execution - Detects use of script interpreters such as PowerShell, CMD, or WMI across multiple sources.
11. Geographic Login Anomaly - Detects when a user's login location differs from their prior login location (from SigninLogs_CL).

Process Breakdown

1. Initial Research
   - Began by identifying high-risk attack techniques (e.g., brute-force, persistence, privilege escalation) using MITRE ATT&CK to evaluate detection coverage and potential organizational exposure.
2. Environment Setup
   - Used Microsoft Sentinel Content Hub to deploy the “Microsoft Sentinel Training Lab” solution into a new Azure workspace.
3. Schema Discovery

   - Explored the schema using the Microsoft Sentinel Logs UI to identify all available columns for SigninLogs_CL and SecurityEvent_CL. This ensured that queries only referenced valid fields and helped shape detection logic.

4. Query Testing
   - Each query was iteratively tested in the live training environment. Time windows (e.g., past 7 days or 10 years) were adjusted to ensure event data existed. Fields were fine-tuned based on testing outcomes.
5. Final Compilation
   - All validated queries were compiled into a Markdown (.md) file, including titles and formatting that simulate actual SOC tools and documentation practices.

Key Takeaways

- Hands-on KQL testing in a realistic environment revealed nuances like field suffixes (_s, _g) that would be missed without real schema access.
- Commented titles were used in queries to allow screenshot documentation without cutting off headings.
- False positives and sparse data were part of the learning process. Even queries with no results still modeled effective detection logic for real-world use.
- This lab highlights operational readiness to:
    - Detect threats using logs
    - Interpret alert context
    - Communicate detection of coverage gaps to risk stakeholders

References

Microsoft. (2024). Kusto Query Language (KQL) documentation. https://learn.microsoft.com/en-us/azure/data-explorer/kusto/query/

Microsoft. (2024). Hunt for threats with Microsoft Sentinel. https://learn.microsoft.com/en-us/azure/sentinel/hunting

MITRE ATT&CK®. (2024). Techniques and Tactics for Cyber Threat Detection. https://attack.mitre.org/