

Visualization of KQL Query Using a Workbook in Microsoft Sentinel

Explicit Credential Logons (Event ID 4648) – Workbook Dashboard Project

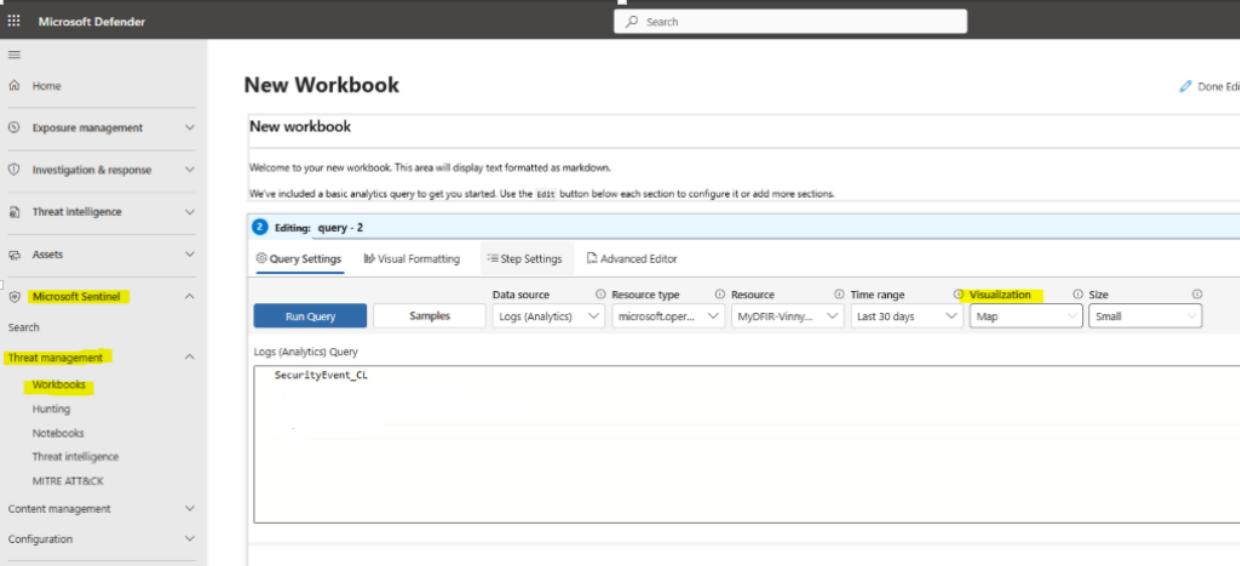
This project demonstrates how I used Microsoft Sentinel Workbooks to visualize suspicious authentication behavior using Kusto Query Language (KQL).

My goal was to build a custom dashboard inside the *Workbook* module to track Explicit Credential Logons (Event ID 4648) — an event commonly tied to lateral movement, credential abuse, and privilege escalation attempts.

How to Navigate to Workbooks in Sentinel

1. Go to security.microsoft.com
2. In the left navigation, select:
Microsoft Sentinel → Threat Management → Workbooks

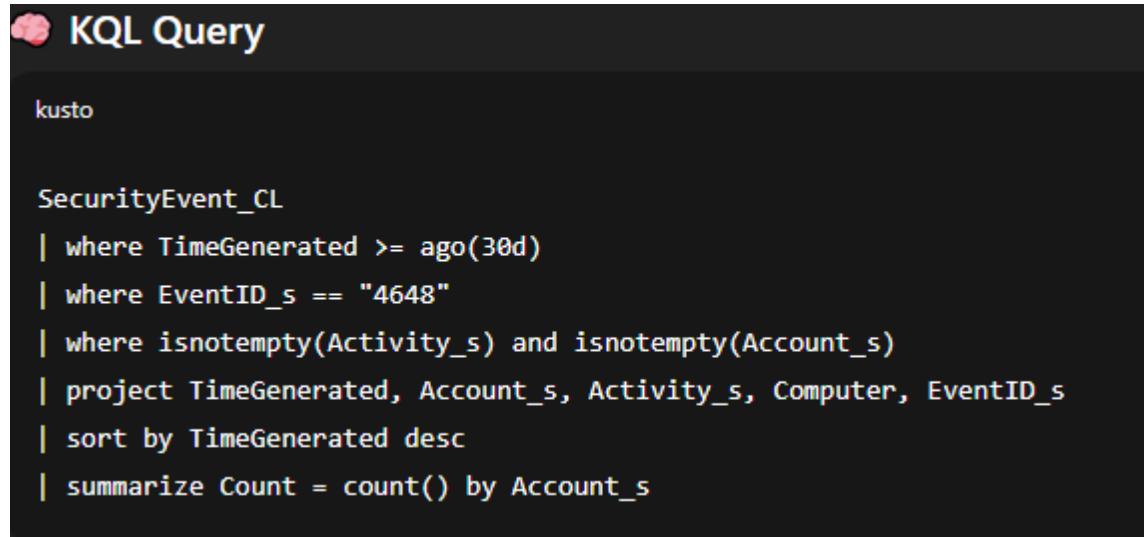
This is the area where you can build interactive dashboards from any KQL queries you run in Sentinel.



The screenshot shows the Microsoft Sentinel 'New Workbook' interface. The left sidebar is collapsed. The main area has a title 'New Workbook' and a sub-section 'New workbook'. It includes a welcome message and a note about basic analytics queries. Below this is a query editor with tabs for 'Editing: query - 2', 'Query Settings', 'Visual Formatting', 'Step Settings', and 'Advanced Editor'. The 'Editing' tab is selected. The 'Run Query' button is highlighted. The 'Logs (Analytics) Query' section contains the KQL query: 'SecurityEvent_CL'. The 'Visualization' tab is selected in the top bar, and the visualization settings show 'Map' and 'Small'.

KQL Threat Hunting Example: Investigating Explicit Credential Logons (Event ID 4648)

As part of my Microsoft Sentinel detective work, I use KQL to hunt for suspicious authentication activity—especially events tied to credential misuse and lateral movement. The query below focuses on **Event ID 4648**, which indicates that a process attempted to log on using **explicit credentials** (often associated with attacker techniques such as pass-the-hash or privilege escalation).

A screenshot of a Kusto Query Editor window titled "KQL Query". The code area contains a Kusto query. The first line is "kusto". The main query starts with "SecurityEvent_CL" and includes filters for TimeGenerated (last 30 days), EventID_s (4648), and non-empty Activity_s and Account_s fields. It then projects TimeGenerated, Account_s, Activity_s, Computer, and EventID_s, sorts by TimeGenerated in descending order, and summarizes the count of events per account.

```
kusto

SecurityEvent_CL
| where TimeGenerated >= ago(30d)
| where EventID_s == "4648"
| where isnotempty(Activity_s) and isnotempty(Account_s)
| project TimeGenerated, Account_s, Activity_s, Computer, EventID_s
| sort by TimeGenerated desc
| summarize Count = count() by Account_s
```

KQL Query Breakdown (Simplified Bullet Points)

- **Filters Security Logs** Queries the SecurityEvent_CL table where Windows Security Events are stored.
- **Last 30 Days Only** | where TimeGenerated >= ago(30d) Focuses the investigation on recent activity.
- **Looks for Event ID 4648** | where EventID_s == "4648" This event shows when a process uses **explicit credentials** — a common sign of lateral movement, credential theft, or privilege escalation.
- **Removes Empty/Noisy Logs** | where isnotempty(Activity_s) and isnotempty(Account_s) Ensures only clean, useful records are analyzed.

Shows Only Important Fields | project TimeGenerated, Account_s, Activity_s, Computer, EventID_s Displays the key details for investigation:

- Time
- Account used
- Activity performed

- Computer
- Event ID

Sorts by Newest Activity | sort by TimeGenerated desc Makes it easier to spot suspicious events in order.

Counts Events per Account | summarize Count = count() by Account_s Helps identify:
Accounts authenticating more than usual

- Abnormal service account behavior
- Credential testing / brute-force attempts
- Unexpected movement across systems

Below is an visualization of that KQL in a PIE CHART layout:

```
SecurityEvent_CL
| where TimeGenerated >= ago(30d)
| where EventID_s == "4648"
| where isnotempty(Activity_s) and isnotempty(Account_s)
| project TimeGenerated, Account_s, Activity_s, Computer, EventID_s
| sort by TimeGenerated desc
| summarize Count = count() by Account_s
```



-