Sentinel Heatmap Lab Report

Case Study: Malicious Traffic Entering the Network Heatmap

Author: A. Brito

Date: September 2025

Table of Contents

Disclaimer

This document has been prepared for demonstration and educational purposes only. The assessment was performed in a cyber-range environment, which replicates real-world enterprise conditions but is designed solely for practice and learning. While the methodology, queries, and visualizations reflect professional security practices, no production systems or live organizational data were assessed. References to enterprise contexts are fictional and intended only to simulate a real-world scenario.

Objective

The objective of this lab is to demonstrate the practical use of **Kusto Query Language (KQL)** within Microsoft Sentinel by creating a heatmap visualization of malicious network traffic.

By leveraging KQL, I was able to:

- Correlate Azure Network Analytics logs with a GeoIP database.
- Identify geographic origins of malicious traffic in near real-time.
- Display results in a Sentinel Workbook using a heatmap for visual clarity.

This exercise highlights both my technical ability to craft KQL queries and my understanding of how to apply them in a SOC workflow for monitoring, detection, and analysis.

Real-World Benefits of the Heatmap

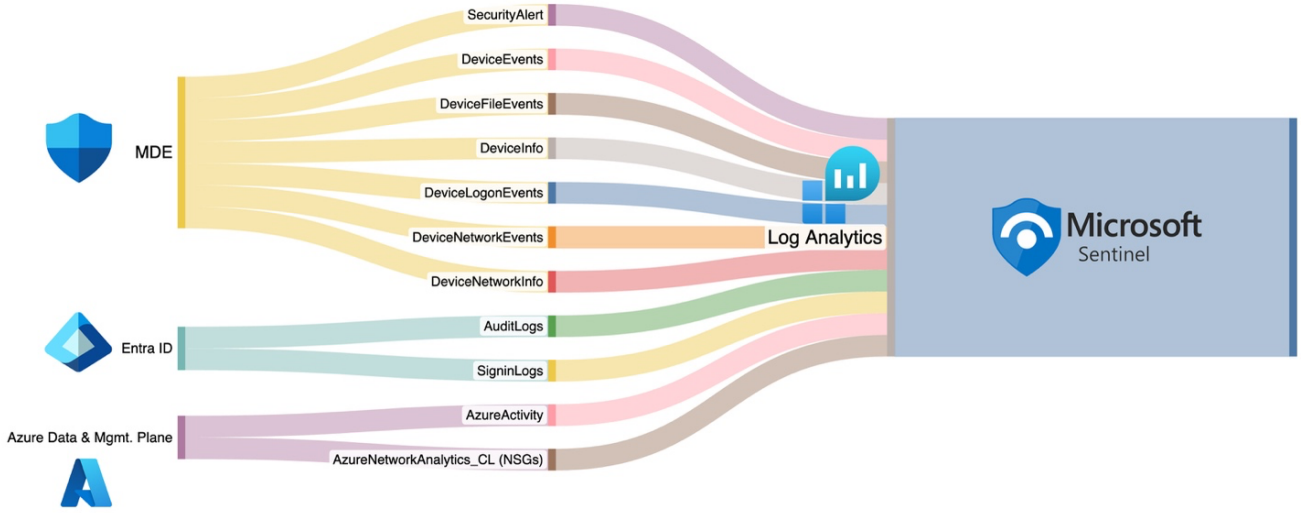
In a production environment, a heatmap of malicious traffic provides security teams with several key advantages:

- **Geographic Visibility** – Analysts can quickly identify where attacks originate, spotting suspicious regions or unexpected sources.
- **Threat Hunting Efficiency** – Visual correlation of data helps prioritize investigation efforts (e.g., focusing on repeat offenders or high-risk geographies).
- **Executive Reporting** – Heatmaps transform raw logs into intuitive visuals that management and stakeholders can understand at a glance.
- **Proactive Defense** – Trends in malicious traffic can inform firewall rules, geo-blocking policies, and threat intelligence enrichment. For example, a sustained spike from a specific country, like the one seen in the Eastern Europe cluster, directly informs the next steps to implement a **Temporary Geoblock Rule** at the network perimeter.

1. Log Flow to Sentinel

The diagram below illustrates how logs are currently ingested into Microsoft Sentinel within the lab environment. Sentinel collects and normalizes network flow data, which is then enriched through KQL queries and visualized in workbooks for easier analysis.

Log Pipeline Overview



2. JSON Code for data extraction

In Microsoft Sentinel, I created a new workbook by navigating to **Threat Management > Workbooks**. I then added a new query widget and inserted the JSON configuration shown below. This configuration defines both the data query and the visualization parameters, enabling the identification of malicious network traffic sources and their geographic locations on the heatmap.

```
{
  "type": 3,
  "content": {
    "version": "KqlItem/1.0",
    "query": "let GeoIPDB_FULL = _GetWatchlist(\"geoip\");\nlet MaliciousFlows = AzureNetworkAnalytics_CL \n| where FlowType_s == \"MaliciousFlow\"\n//| where SrcIP_s == \"10.0.0.5\" \n| order by
    "size": 3,
    "timeContext": {
      "durationMs": 2592000000
    },
    "queryType": 0,
    "resourceType": "microsoft.operationalinsights/workspaces",
    "visualization": "map",
    "mapSettings": {
      "locInfo": "LatLong",
      "locInfoColumn": "countryname",
      "latitude": "latitude",
      "longitude": "longitude",
      "sizeSettings": "city",
      "sizeAggregation": "Count",
      "opacity": 0.8,
      "labelSettings": "friendly_location",
      "legendMetric": "IpAddress",
      "legendAggregation": "Count",
      "itemColorSettings": {
        "nodeColorField": "city",
        "colorAggregation": "Count",
        "type": "heatmap",
        "heatmapPalette": "greenRed"
      }
    }
  },
  "name": "query - 0"
}
```

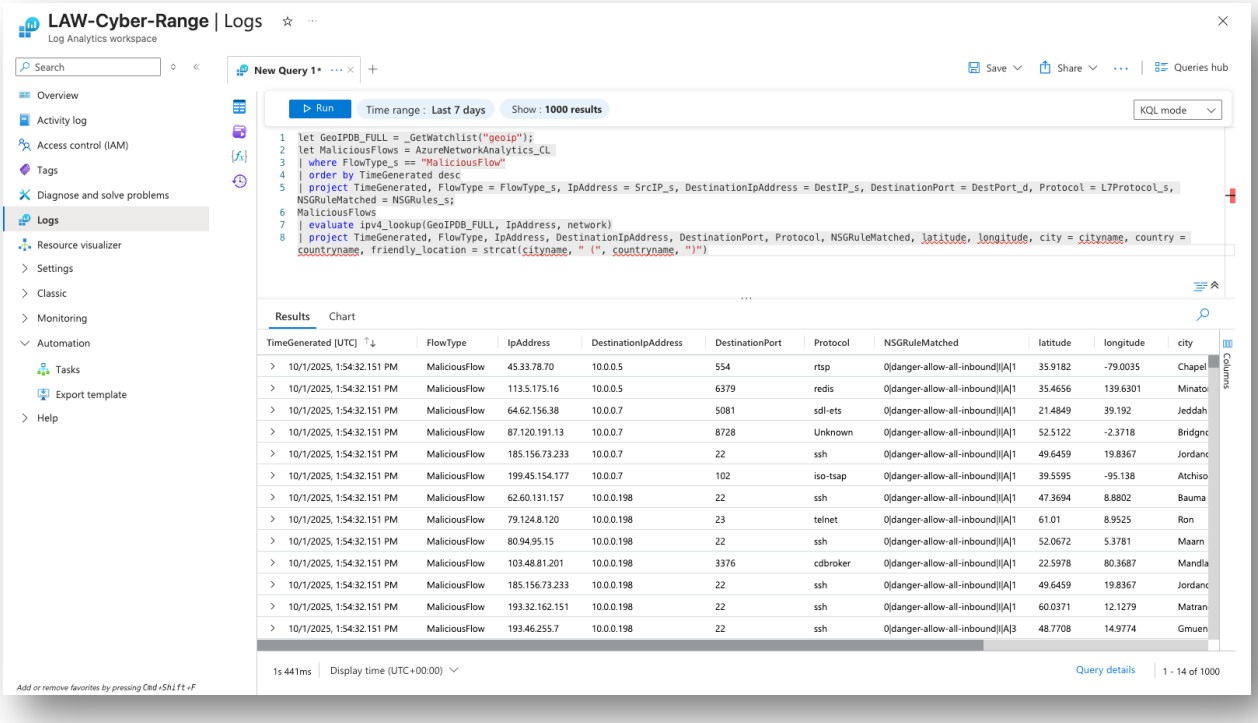
Core Function

The main purpose of this Json code is to:

- Retrieve a GeoIP database (a list of IP addresses mapped to physical locations).
- Filter Azure Network Analytics logs to find records marked as "**MaliciousFlow**".
- Use the GeoIP database to look up the source IP address for each malicious flow and find its corresponding latitude, longitude, city, and country.
- Visualize the results on a map, using a heatmap to show where the malicious traffic originates.

3. KQL script Analysis

This KQL script will help me identify the geographic source of malicious network traffic by combining network flow logs with an IP address geolocation database.




```
let GeoIPDB_FULL = _GetWatchlist("geoip");
let MaliciousFlows = AzureNetworkAnalytics_CL
| where FlowType_s == "MaliciousFlow"
| order by TimeGenerated desc
| project TimeGenerated, FlowType = FlowType_s, IpAddress = SrcIP_s,
DestinationIpAddress = DestIP_s, DestinationPort = DestPort_d, Protocol =
L7Protocol_s, NSGRuleMatched = NSGRules_s;
MaliciousFlows
| evaluate ipv4_lookup(GeoIPDB_FULL, IpAddress, network)
| project TimeGenerated, FlowType, IpAddress, DestinationIpAddress,
DestinationPort, Protocol, NSGRuleMatched, latitude, longitude, city = cityname,
country = countryname, friendly_location = strcat(cityname, " (", countryname, ")")
```

KQL Breakdown

The query executes in three main stages:

1. Retrieve the GeoIP Database

```
let GeoIPDB_FULL = _GetWatchlist("geoip");
```

- Loads a pre-configured watchlist mapping public IPs to their geographic locations (latitude, longitude, city, etc)
- Stores the result in the variable `GeoIPDB_FULL`

2. Identify and Prepare Malicious Network Flows

```
let MaliciousFlows = AzureNetworkAnalytics_CL
| where FlowType_s == "MaliciousFlow"
| order by TimeGenerated desc
| project TimeGenerated, FlowType = FlowType_s, IpAddress = SrcIP_s,
DestinationIpAddress = DestIP_s, DestinationPort = DestPort_d, Protocol =
L7Protocol_s, NSGRuleMatched = NSGRules_s;
```

- Queries the `AzureNetworkAnalytics_CL` table (Azure Network Watcher flow logs).
- Filters to only flows classified as *MaliciousFlow*.
- Orders events chronologically (newest first).
- Projects clean, renamed columns: source IP, destination IP/port, protocol, and NSG rule.

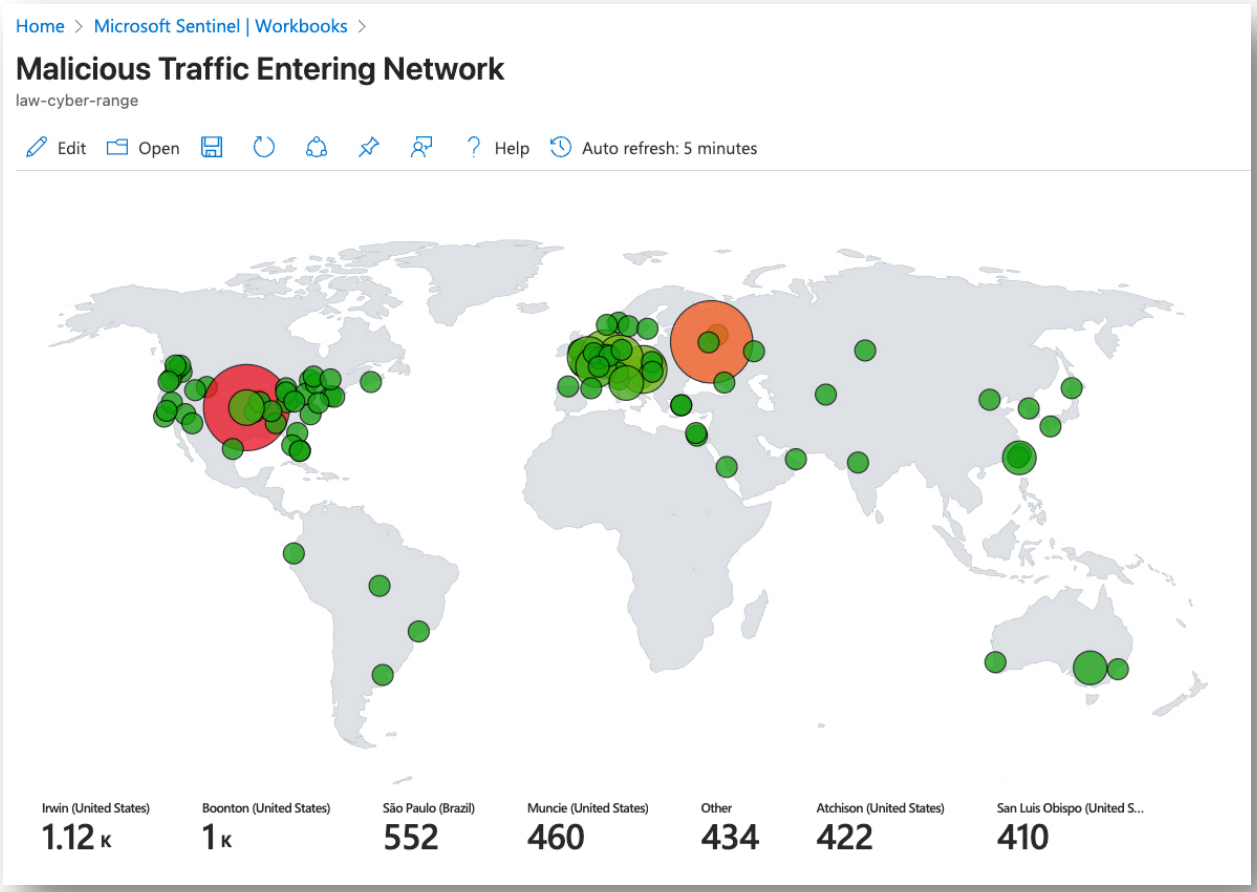
3. Geocode and Finalize Data

```
MaliciousFlows
| evaluate ipv4_lookup(GeoIPDB_FULL, IpAddress, network)
| project TimeGenerated, FlowType, IpAddress, DestinationIpAddress,
DestinationPort, Protocol, NSGRuleMatched, latitude, longitude, city = cityname,
country = countryname, friendly_location = strcat(cityname, " (", countryname, ")")
```

- Joins malicious flow data with the GeoIP watchlist to enrich each record with latitude, longitude, city, and country.
- Projects the final set of fields, including a `friendly_location` column (e.g., *"London (United Kingdom)"*) for easy visualization on the heatmap.

4. Heatmap Result/Findings

The visualization clearly shows two primary hotspots for malicious activity: the central United States (Irwin, Boonton, Muncie, Atchison) and a cluster in Central/Eastern Europe. The high count from '**Other**' indicates a need for further investigation into unclassified or unknown IP ranges. This immediate visual prioritization allows the SOC team to focus geo-blocking efforts on the most active regions.



Adaptability of the Map

Lastly, this map can be modified and adapted depending on the type of data being analyzed. The same KQL and Sentinel Workbook methodology can be applied to a variety of scenarios, providing tailored insights for different aspects of enterprise security. Examples include:

- **Azure Entra ID Authentication Success** Heatmap – visualize global successful login activity.
- **Azure Entra ID Authentication Failures** Heatmap – identify geographic sources of failed login attempts that may indicate brute-force or credential stuffing attacks.
- **Azure Resource Creation** Heatmap – track where new resources are being provisioned to detect potential misuse or misconfigurations.
- **VM Authentication Failures** Heatmap – pinpoint login attempts against virtual machines across geographies to identify suspicious or unauthorized access activity.

By customizing the underlying queries and data sources, this visualization approach becomes a reusable, flexible tool for strengthening situational awareness and supporting proactive threat hunting.