1. Introduction

This project documents the design and implementation of a cloud-based Security Operations Center (SOC) lab using Microsoft Sentinel in Azure. The lab was built to capture real brute-force attack traffic, enrich logs with geolocation data, and simulate SOC workflows such as detection, visualization, and incident documentation

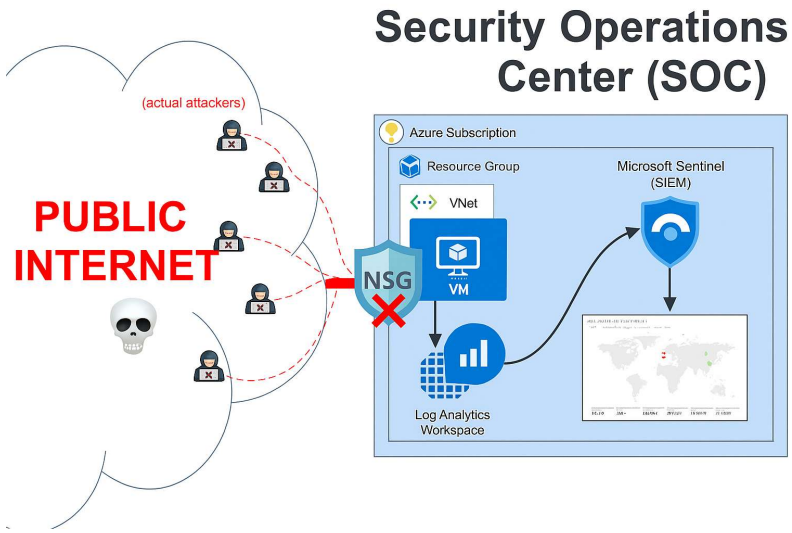

Figure 1: SOC lab architecture showing data flow from public internet to Azure Sentinel.

2. Project Objectives

- Capture real-world brute-force traffic against a honeypot VM
- Centralize logs for analysis using Microsoft Sentinel
- Enrich attacker data with geolocation context
- Visualize global attack patterns and simulate SOC workflow

3. Lab Setup

To simulate a real-world SOC environment, I deployed a honeypot virtual machine (VM) in Microsoft Azure and configured it to attract brute-force login attempts. The lab was designed to be low-cost, observable, and scalable, using native Azure services for monitoring and Microsoft Sentinel for centralized analysis.

3.1 Azure Subscription & Resource Group

A dedicated resource group named SOC-Lab was created within my Azure subscription to contain all lab components. This included the VM, its public IP, the network security group (NSG), the Log Analytics Workspace (LAW), and the Sentinel instance.

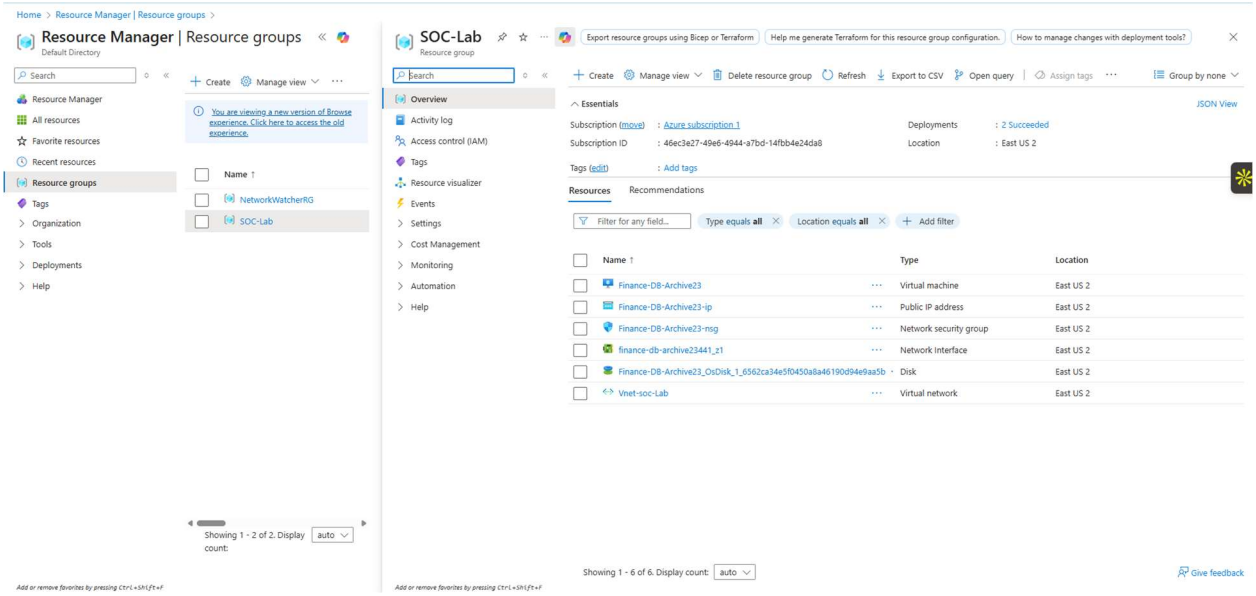

Figure 2: Resource Group overview showing VM, NSG, LAW, and VNet.

3.2 Virtual Machine Deployment

I deployed a **Windows 11 Enterprise, version 25H2 – x64 Gen2 VM** with the following configuration:

- Size: Standard_D2s_v3 (2 vCPU, 8 GB RAM)
- Region: East US 2 (Zone 1)
- Public IP: Enabled
- Disk: Standard SDD
- Inbound Ports: RDP (TCP 3389) exposed to the internet

The VM was intentionally configured with open RDP access to simulate a vulnerable endpoint and attract brute-force login attempts.

3.3 Network Security Group (NSG) Configuration

The NSG attached to the VM was configured with an **AllowAllTraffic** rule, permitting unrestricted inbound connections. This deliberate misconfiguration ensured the honeypot would be discoverable by attackers scanning the internet.

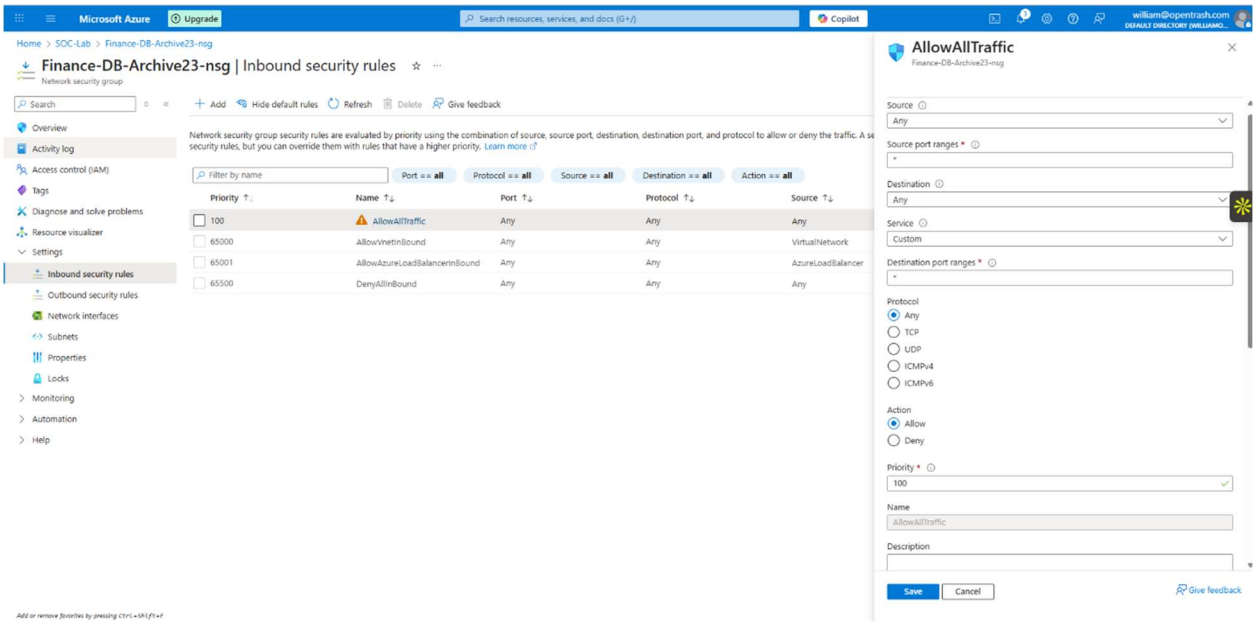

Figure 3: NSG inbound rules showing open RDP access.

3.4 Firewall Settings

During testing, the Windows Defender Firewall was toggled between enabled and disabled states to observe differences in attack behavior. This provided insight into how attackers respond to varying exposure levels.

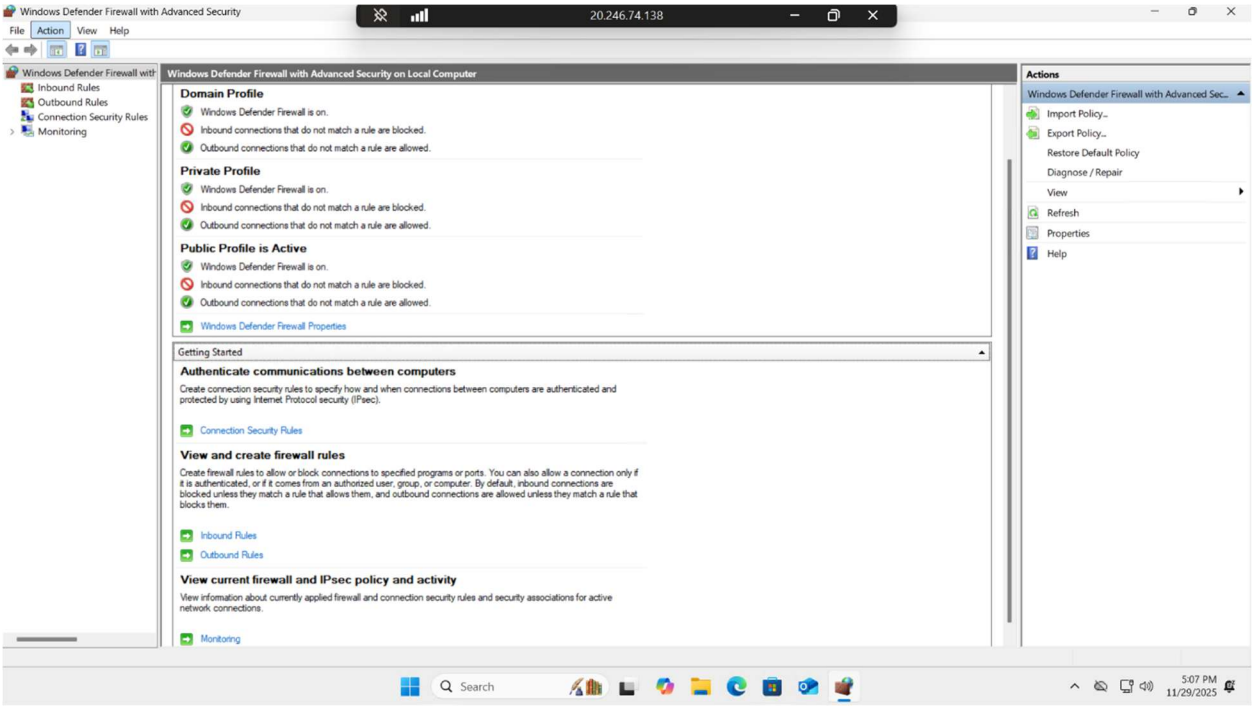

Figure 4: Firewall ON (default secure state).

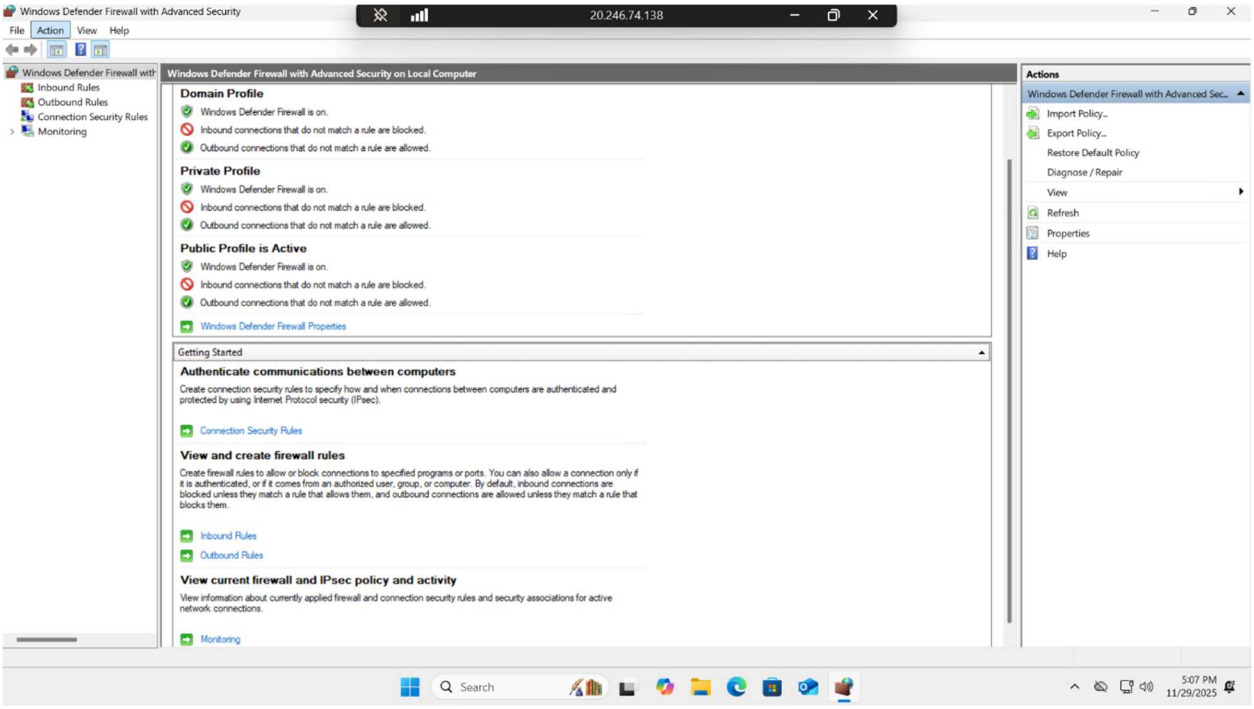

Figure 5: Firewall OFF (honeypot exposure state)

3.5 Log Analytics Workspace (LAW) Creation

A Log Analytics Workspace named **Log-Repository-SOC-lab-0000** was created and linked to the VM. This workspace served as the central repository for Windows Security Event logs, specifically Event ID 4625 (failed login attempts)

4. Log Collection

Once the honeypot VM was deployed and exposed to the internet, the next step was to capture and centralize its security logs. This ensured that all failed login attempts could be analyzed in Microsoft Sentinel using KQL queries.

4.1 Windows Security Event Logs

The honeypot VM generated **Windows Security Event Logs**, specifically **Event ID 4625** (failed login attempts). These logs provided visibility into brute-force activity, including attacker IP addresses, targeted accounts, and timestamps.

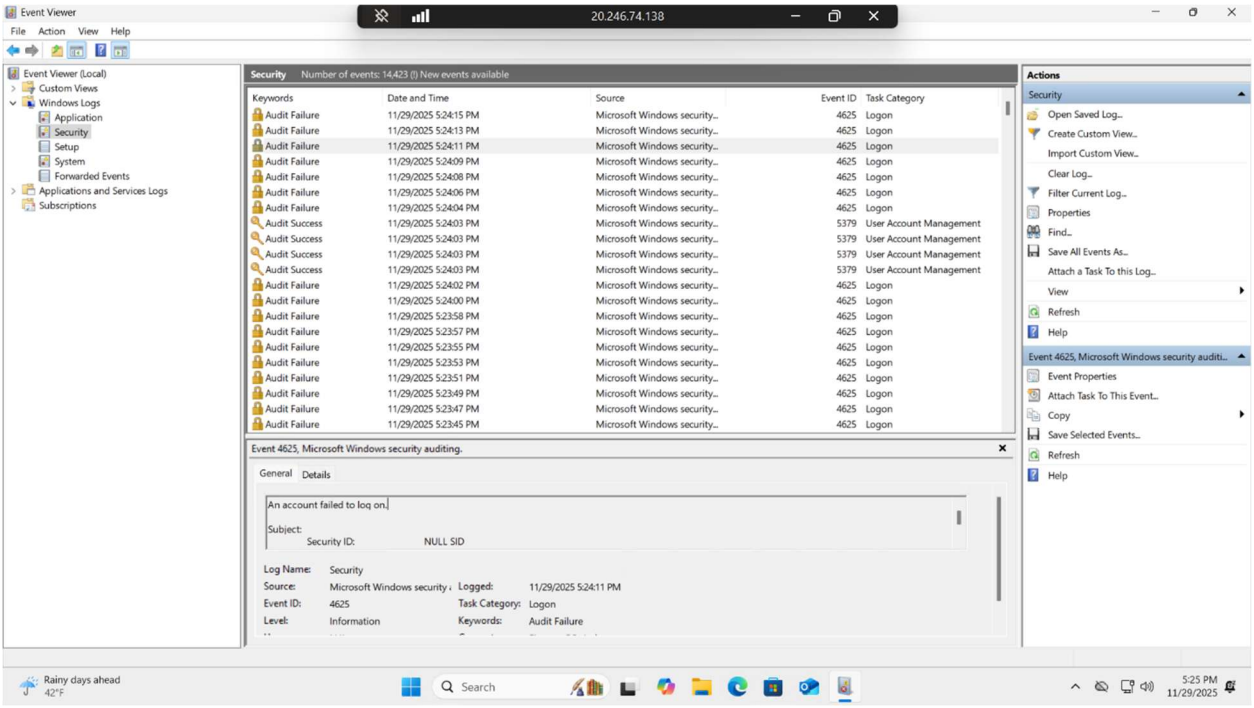

Figure 6: Event Viewer showing multiple Event ID 4625 entries

4.2 Log Analytics Workspace (LAW) Integration

A **Log Analytics Workspace (LAW)** named *Log-Repository-SOC-lab-0000* was created and linked to the VM. The **Azure Monitor Agent (AMA)** was installed on the VM to forward Windows Security Events directly into LAW.

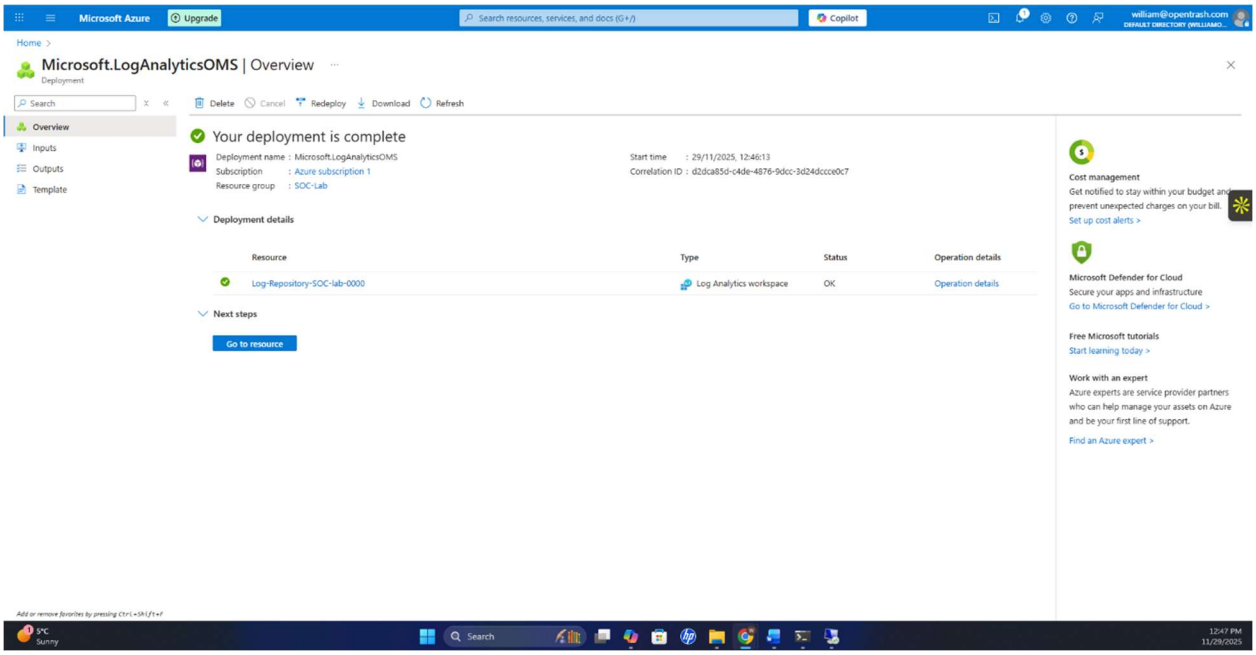

Figure 7: LAW deployment confirmation.

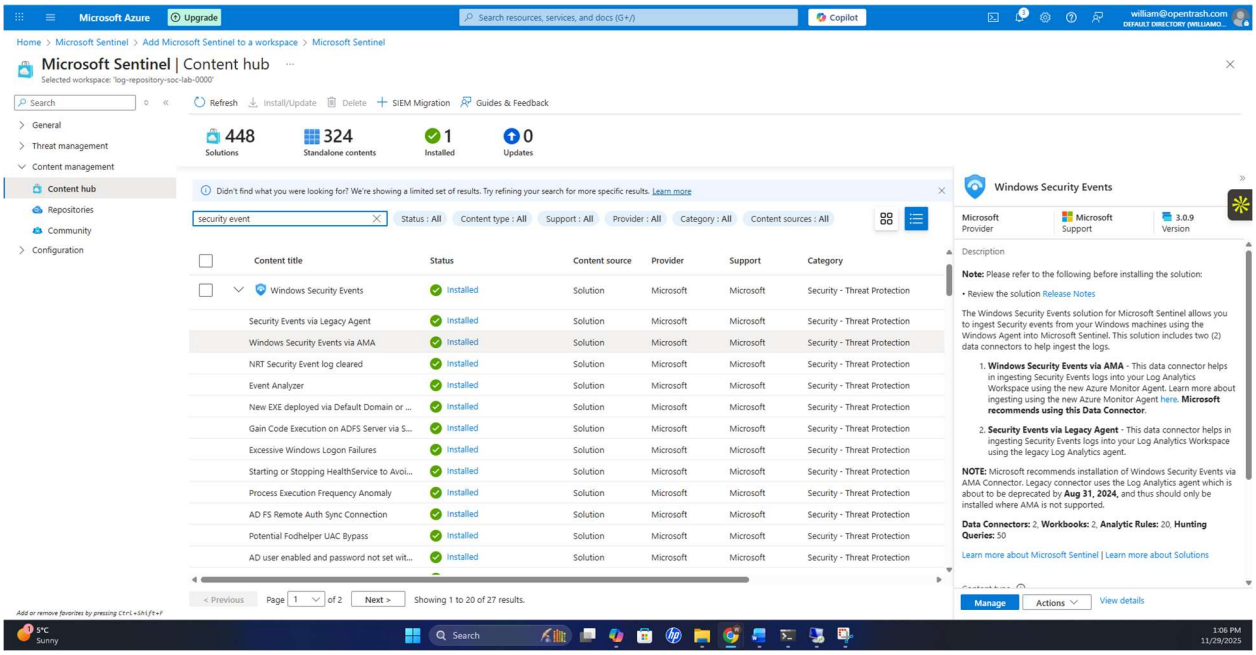

Figure 8: VM extensions showing Azure Monitor Agent installed.

4.3 Raw Log Ingestion

Once connected, the LAW began ingesting raw security events from the VM. These logs included account names, event sources, and timestamps, forming the foundation for later enrichment and visualization.

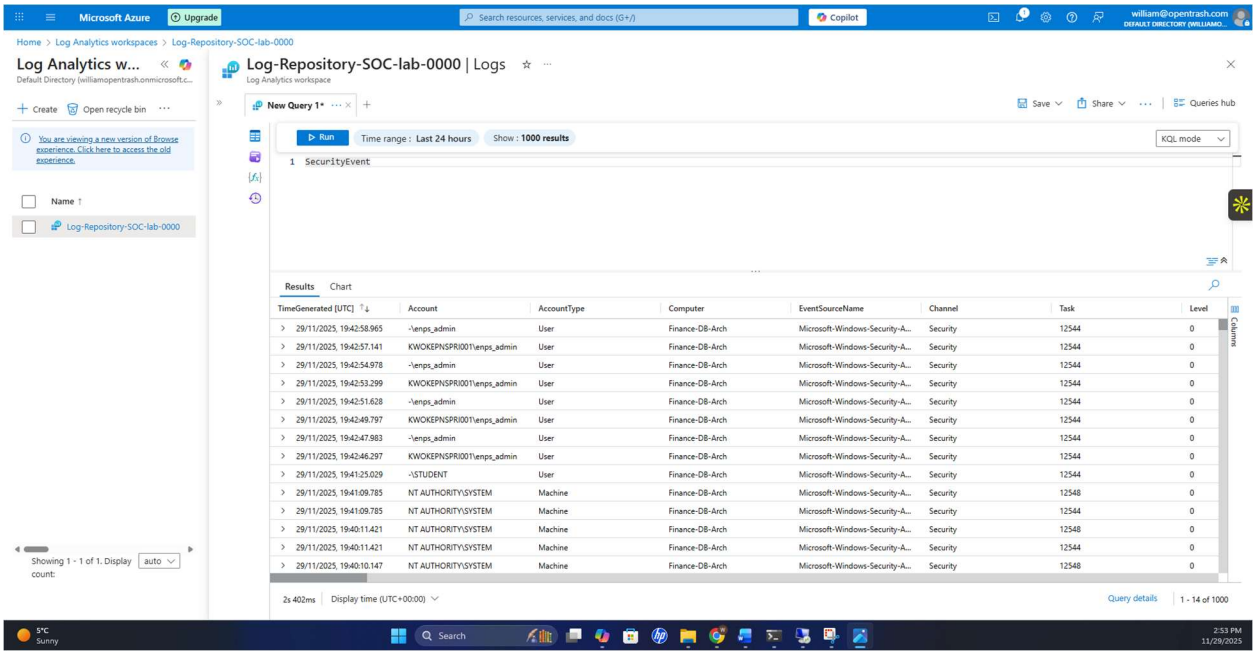

Figure 9: LAW query results showing raw SecurityEvent table entries.

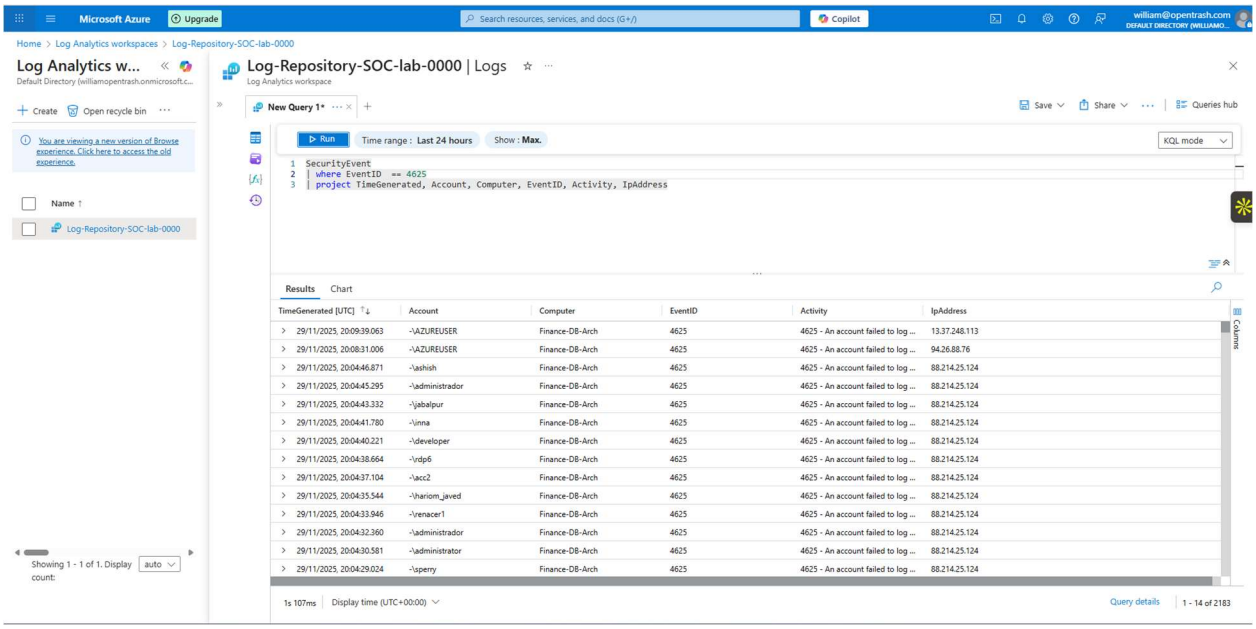

Figure 20: LAW query results with multiple failed login attempts from different accounts.

4.4 Custom KQL Queries

To filter relevant data, I wrote custom **KQL queries** targeting Event ID 4625. These queries extracted attacker IP addresses, accounts, and activity descriptions, allowing me to focus on brute-force attempts.

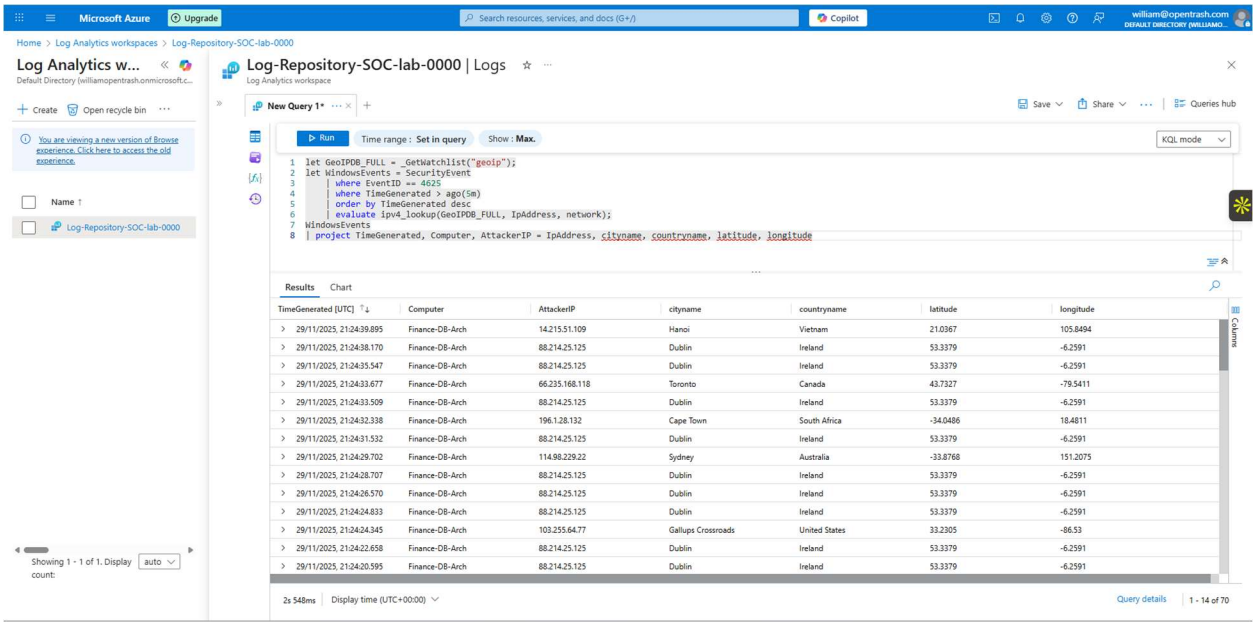

Figure 31: Query results showing attacker IPs and failed login attempts in the last 5 minutes.

4.5 Why This Matters

Centralizing logs in LAW provided a **single source of truth** for monitoring attacker activity. By filtering for failed login events, I was able to:

- Quantify the scale of brute-force attempts.
- Identify attacker IPs and targeted accounts.
- Prepare the data for enrichment with GeoIP information.

This step transformed raw Windows logs into structured data ready for SOC analysis in Sentinel.

4.6 Microsoft Sentinel Configuration

With logs flowing into the Log Analytics Workspace, the next step was to enable **Microsoft Sentinel** to provide SIEM capabilities. Sentinel allowed me to query, enrich, and visualize attacker activity while simulating SOC workflows

4.7 Onboarding Sentinel

Microsoft Sentinel was added to the existing Log Analytics Workspace (*Log-Repository-SOC-lab-0000*). This provided a centralized platform for security monitoring and analysis.

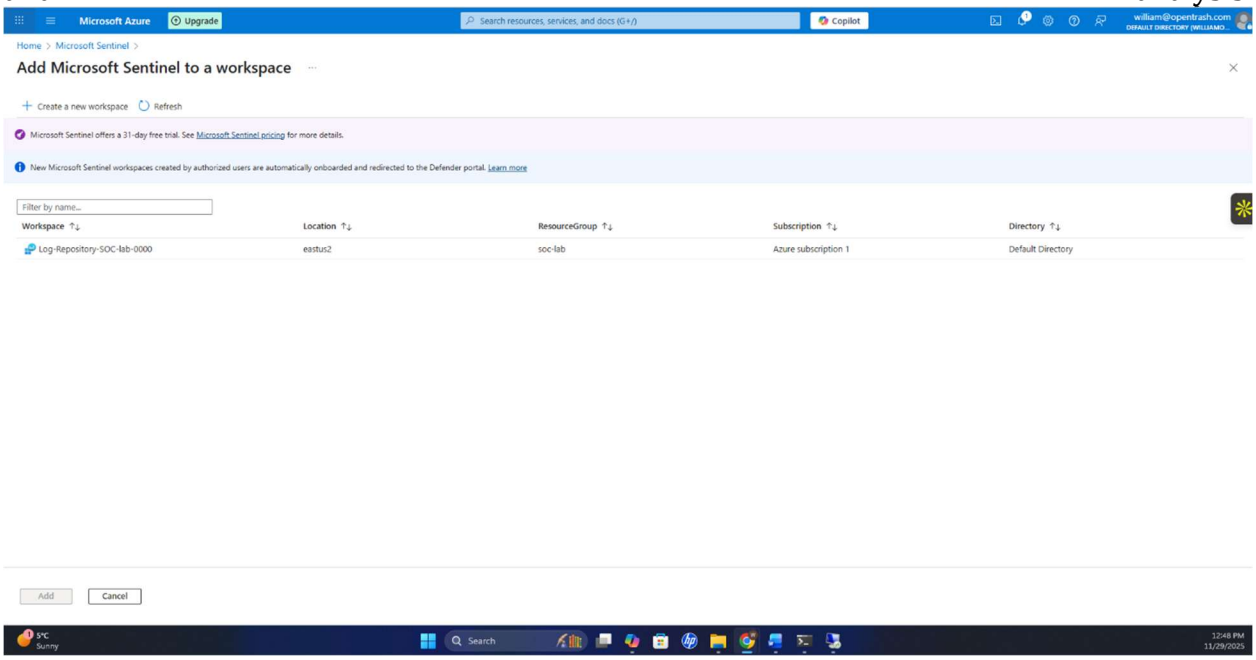

Figure 12: Sentinel onboarding interface showing workspace selection.

4.8 Content Hub & Data Connectors

To ensure proper ingestion of Windows Security Events, I installed the **Windows Security Events solution** from the Sentinel Content Hub. This included:

- **Data Connectors:** Windows Security Events via AMA (recommended) and Legacy Agent (deprecated).
- **Workbooks:** Prebuilt dashboards for log visualization.
- **Analytic Rules:** Templates for detecting excessive login failures.
- **Hunting Queries:** Prebuilt queries for threat hunting.

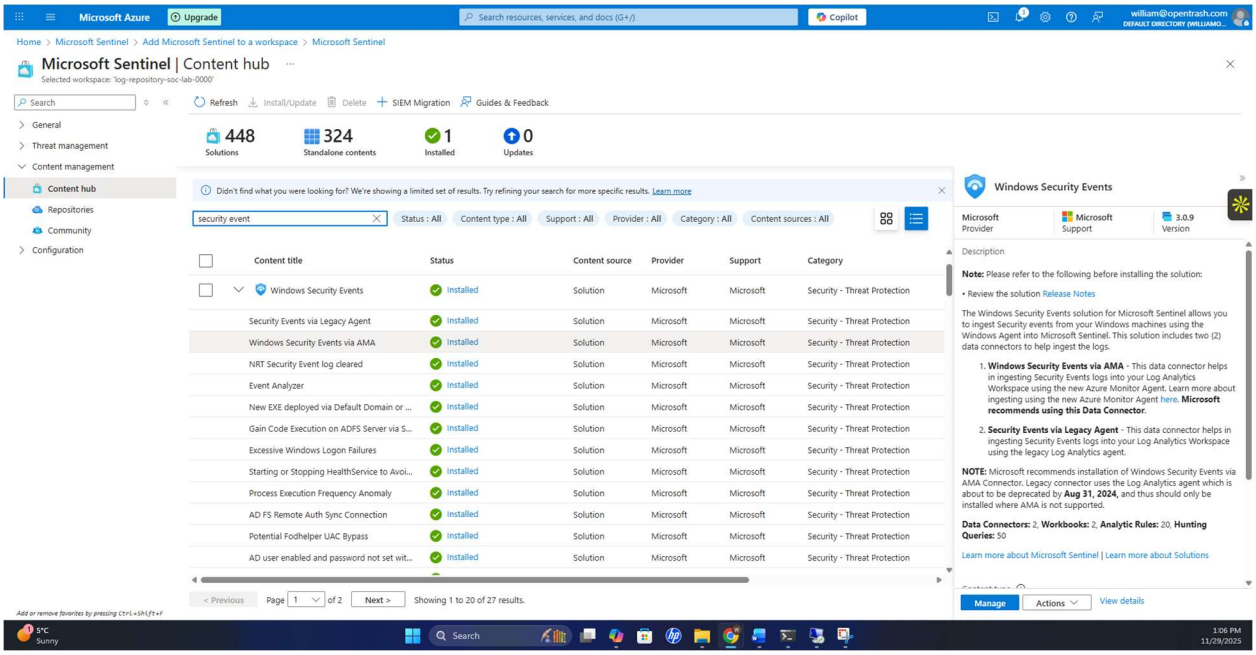

Figure 43: Sentinel Content Hub showing installed Windows Security Events solution.

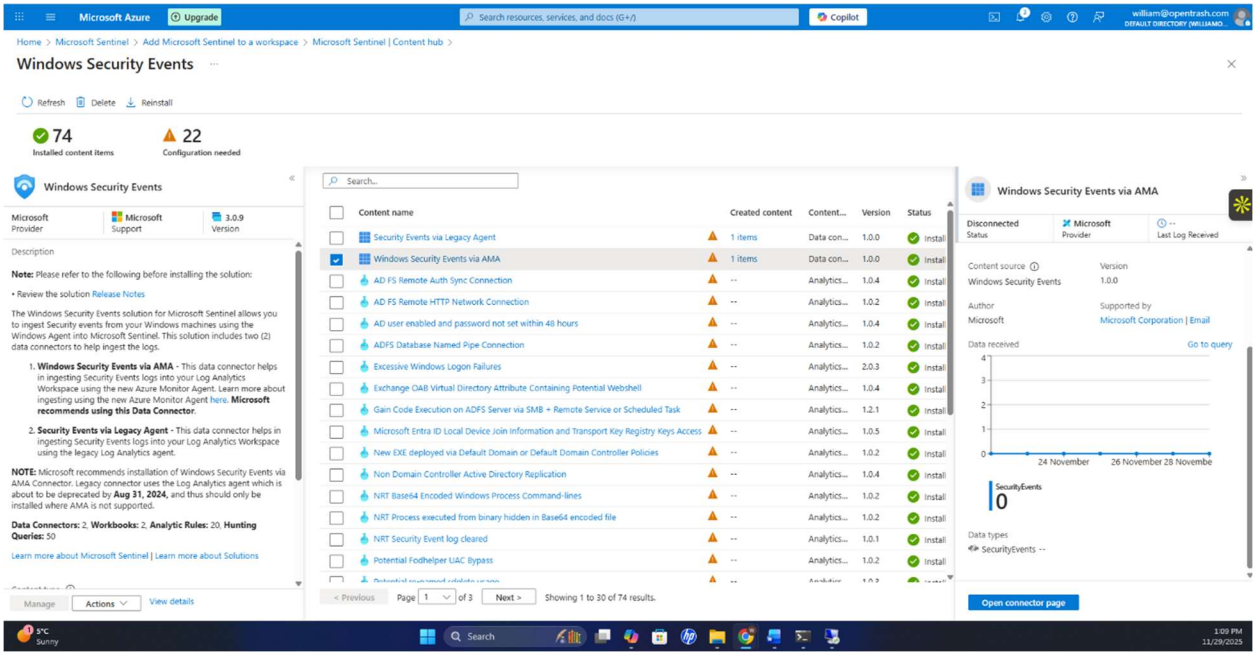

Figure 54: Sentinel connector status (AMA disconnected/connected)

4.9 KQL Query Development

Within Sentinel, I wrote custom **KQL queries** to analyze failed login events (Event ID 4625). These queries extracted attacker IPs, targeted accounts, and timestamps, enabling me to

correlate adversary behavior across datasets.

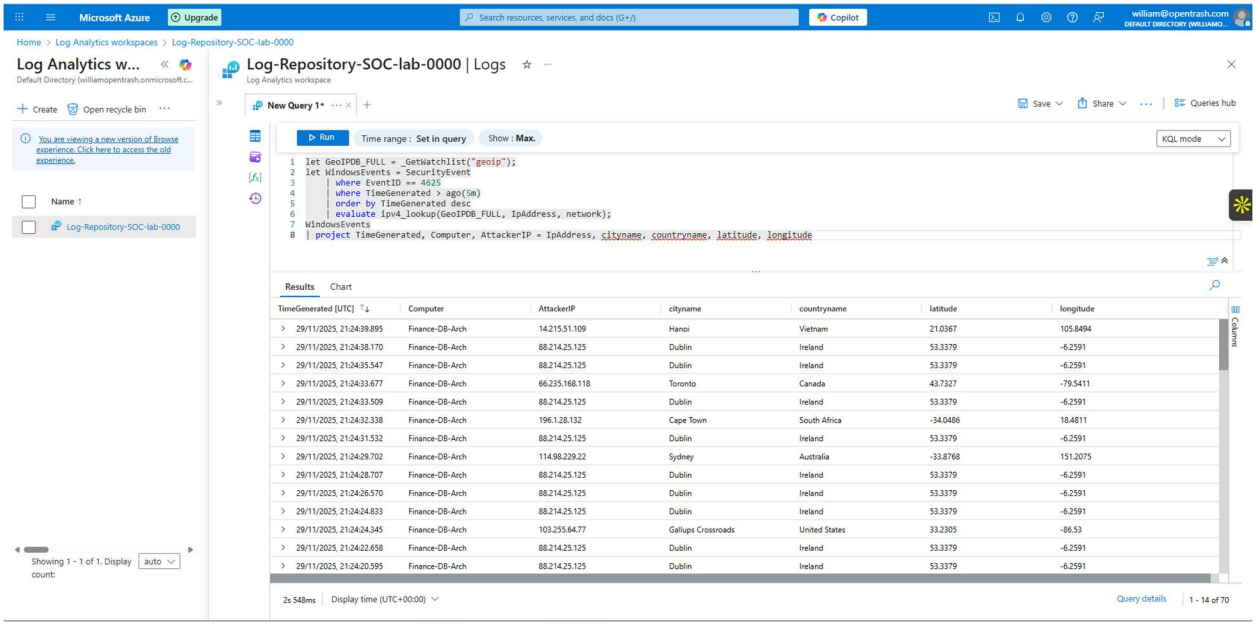

Figure 65: KQL query results inside Sentinel showing failed login attempts and attacker IPs.

6. Data Enrichment

To transform raw failed-login logs into actionable intelligence, I enriched attacker IP addresses with geolocation data. This step added critical context such as country, region, and city, enabling deeper analysis and global attack mapping.

6.1 GeoIP Watchlist Creation

I imported a 55,000-row GeoIP CSV dataset into Microsoft Sentinel as a Watchlist. This dataset contained IP ranges mapped to geographic locations. Once uploaded, Sentinel treated it as a reference table that could be joined with my SecurityEvent logs.

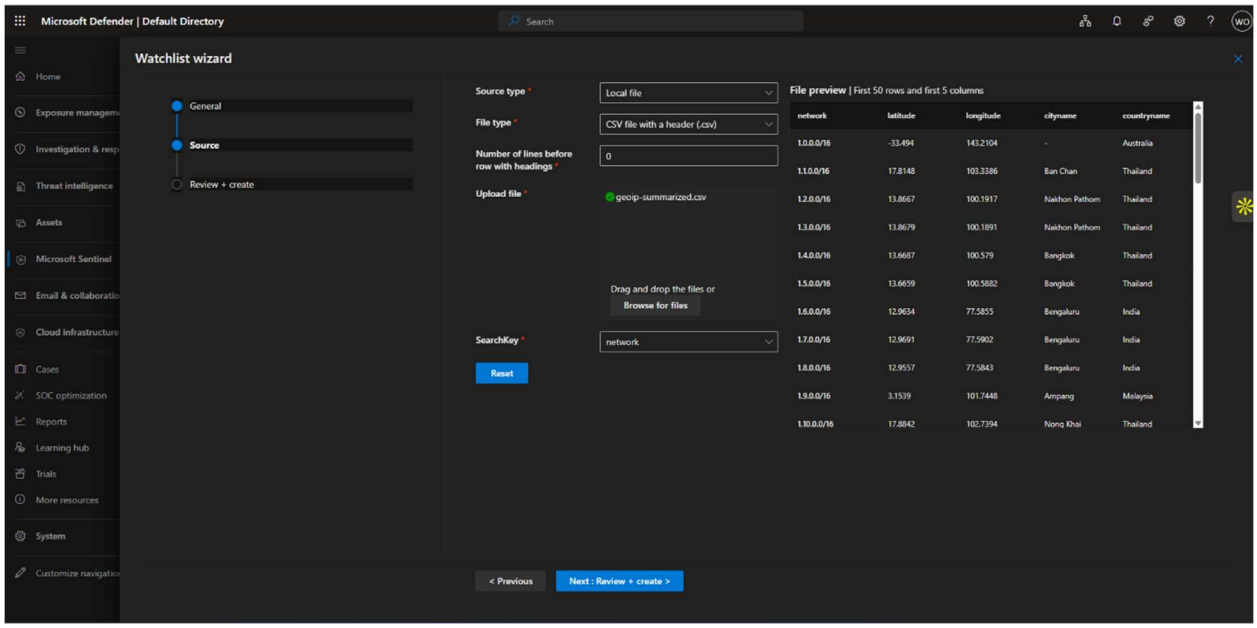

Figure 76: Watchlist wizard showing GeoIP CSV upload.

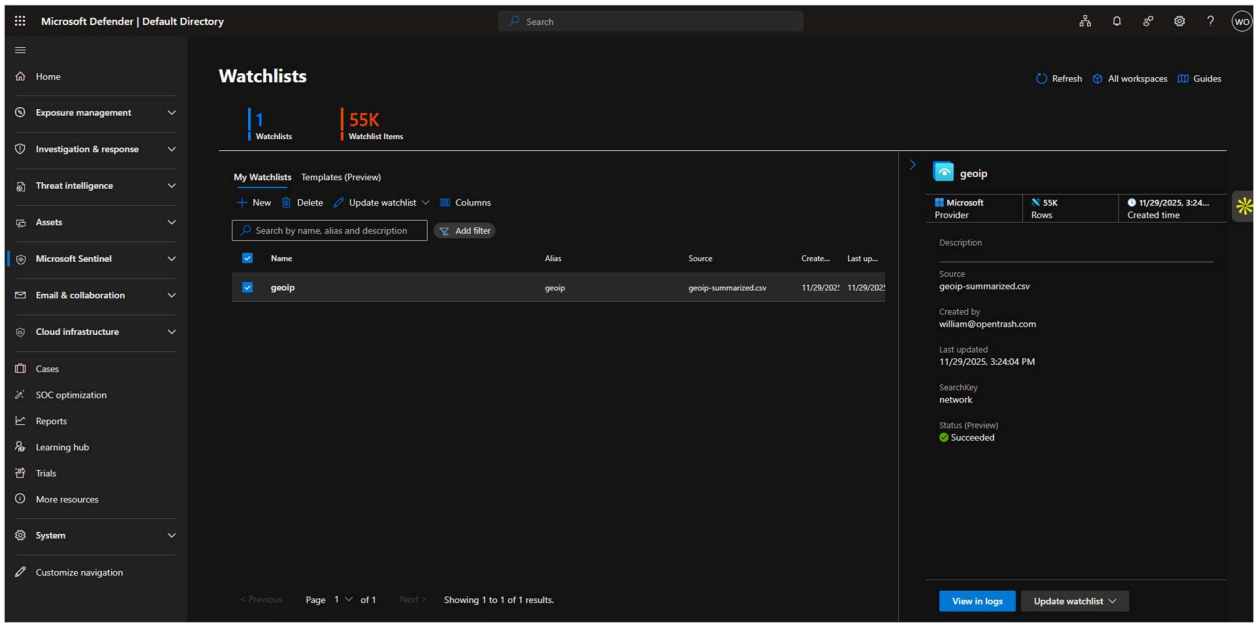

Figure 87: Watchlist summary confirming 55K entries.

6.2 KQL Enrichment Query

Using KQL, I joined attacker IPs from Event ID 4625 logs with the GeoIP watchlist. This produced enriched records containing:

- IP address
- Country

- Region
- City
- Latitude/Longitude
- Timestamp

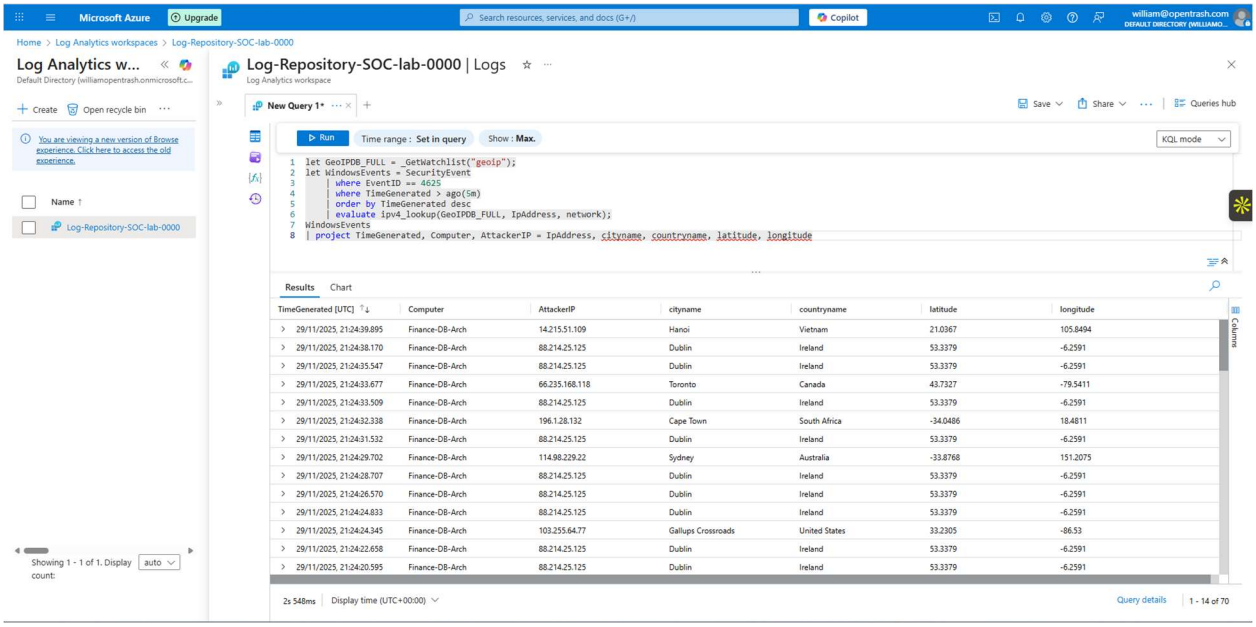

Figure 18: KQL query showing join between SecurityEvent and GeoIP watchlist.

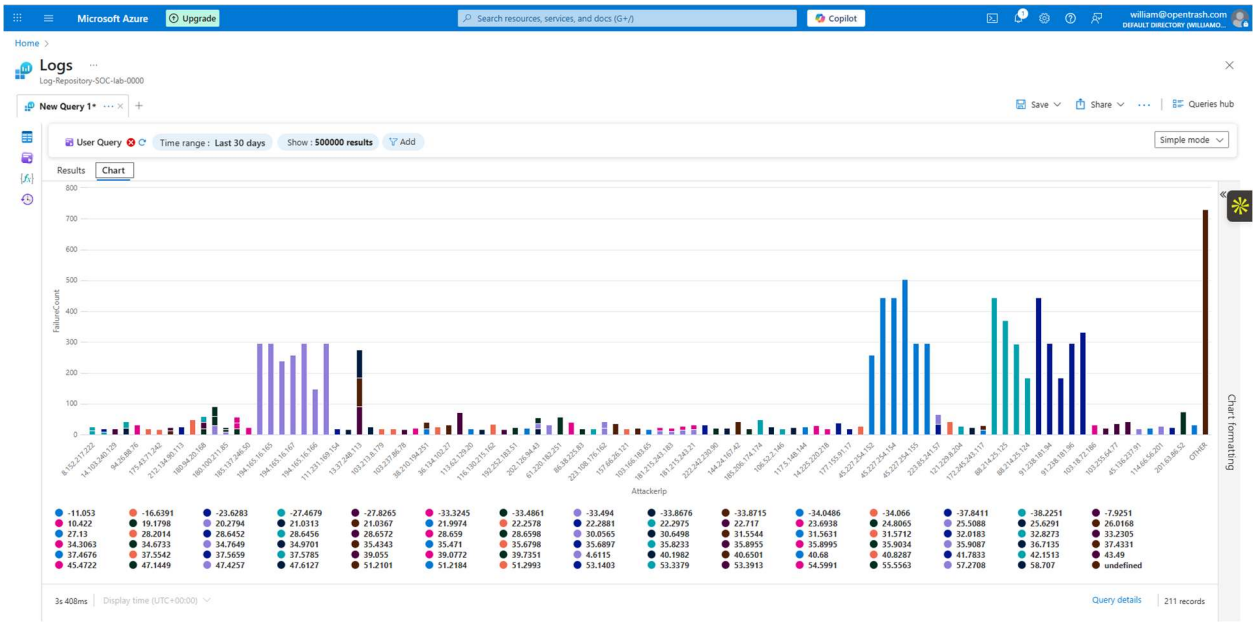

Figure 199: Query results showing attacker IPs mapped to cities and countries.

7. Visualization

With enriched data available, I built a Sentinel Workbook to visualize global attacker activity. This provided a real-time, interactive view of brute-force attempts across 37 countries.

7.1 Global Attack Map

Using Sentinel's map visualization component, I plotted attacker IPs by geolocation. Each marker represented a failed login attempt enriched with city and country data.

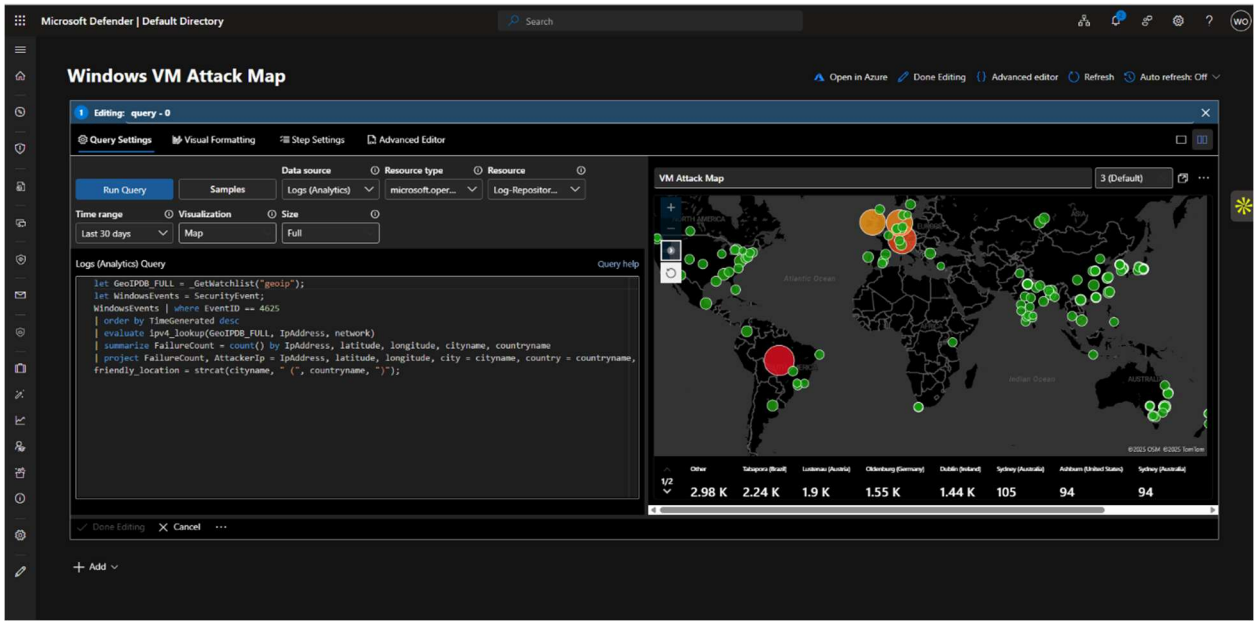

Figure 100: Global attack map with multiple markers.

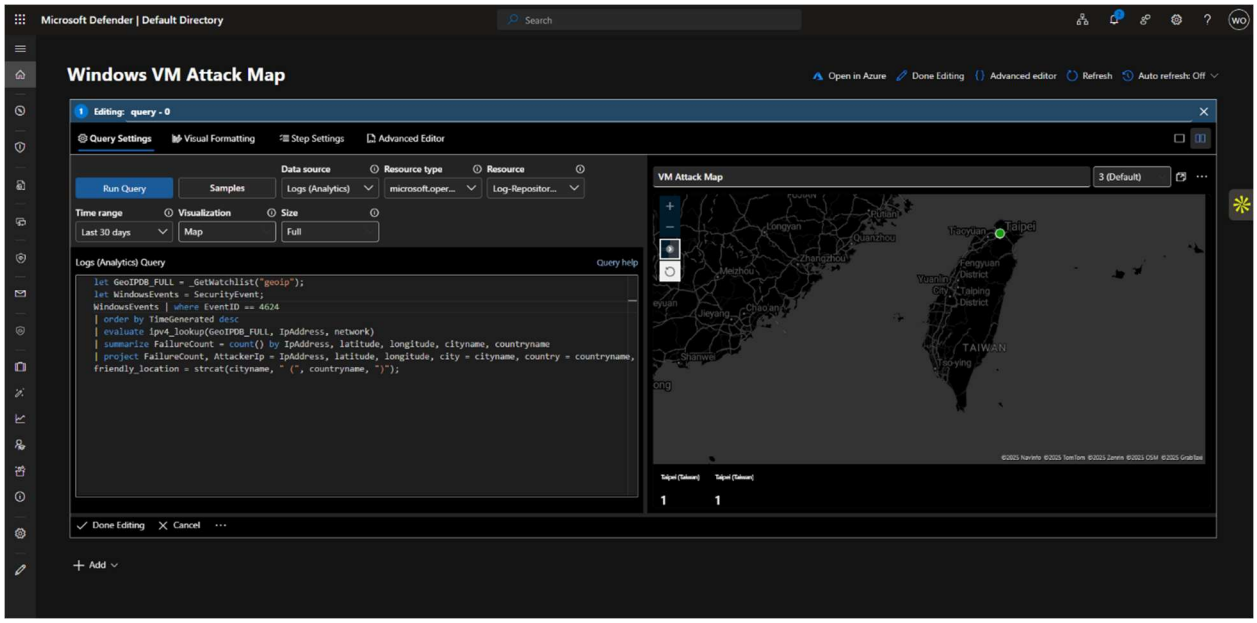

Figure 21: Attack map zoomed into Taipei, Taiwan.

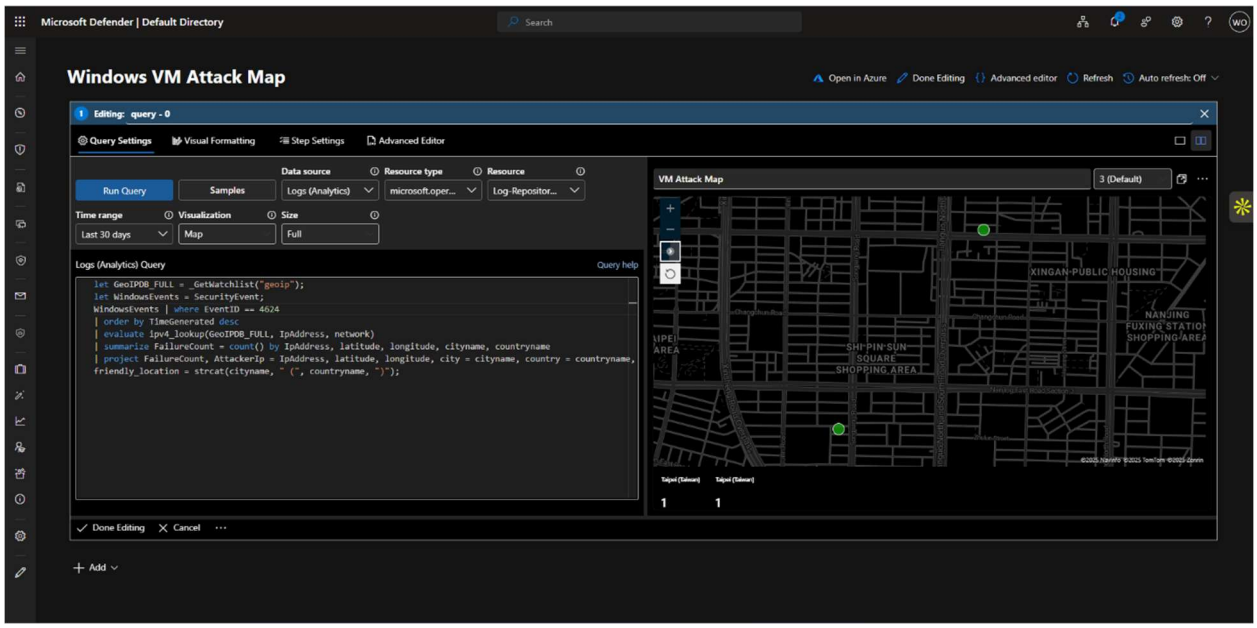

Figure 22: Map showing green markers for attacker IPs.

7.2 Additional Visualizations

To support the map, I added:

- Bar charts showing top attacker IPs
- Tables summarizing country-level attack counts
- Time charts showing attack frequency over time

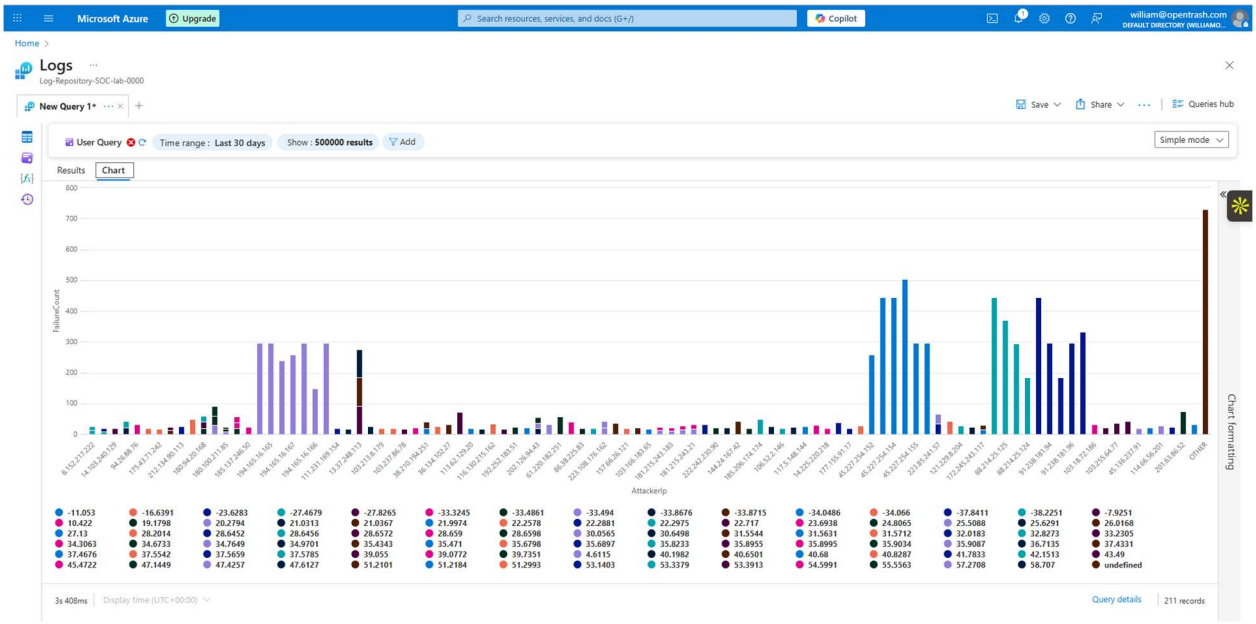

Figure 23: Bar chart of failed login attempts by IP.

8. Results

The honeypot VM attracted significant real-world attacker activity within hours of deployment. The enriched and visualized data revealed clear patterns in global brute-force behavior.

8.1 Attack Volume

Within the first 10 hours, the VM recorded:

- 12,000+ failed login attempts
- Attempts targeting multiple default Windows accounts
- Repeated attempts from persistent attacker IPs

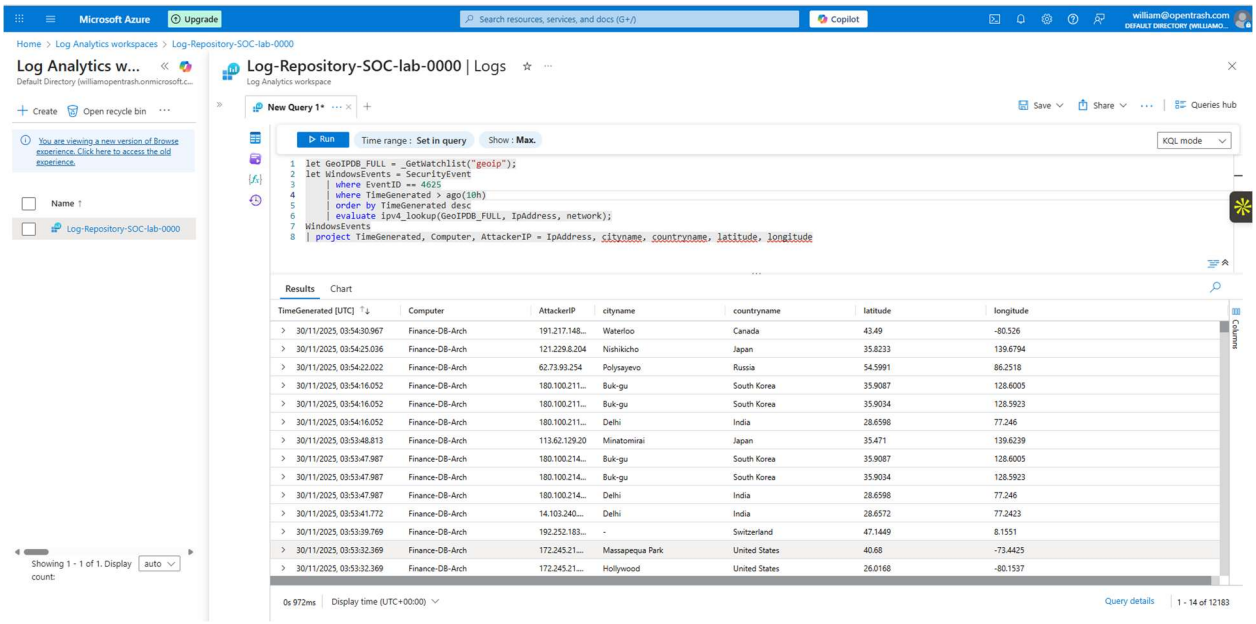

Figure 11: KQL results showing high-volume failed login attempts.

8.2 Geographic Distribution

Enriched logs showed attackers originating from 37 countries, including:

- Taiwan
- China
- Russia
- Brazil
- United States
- India

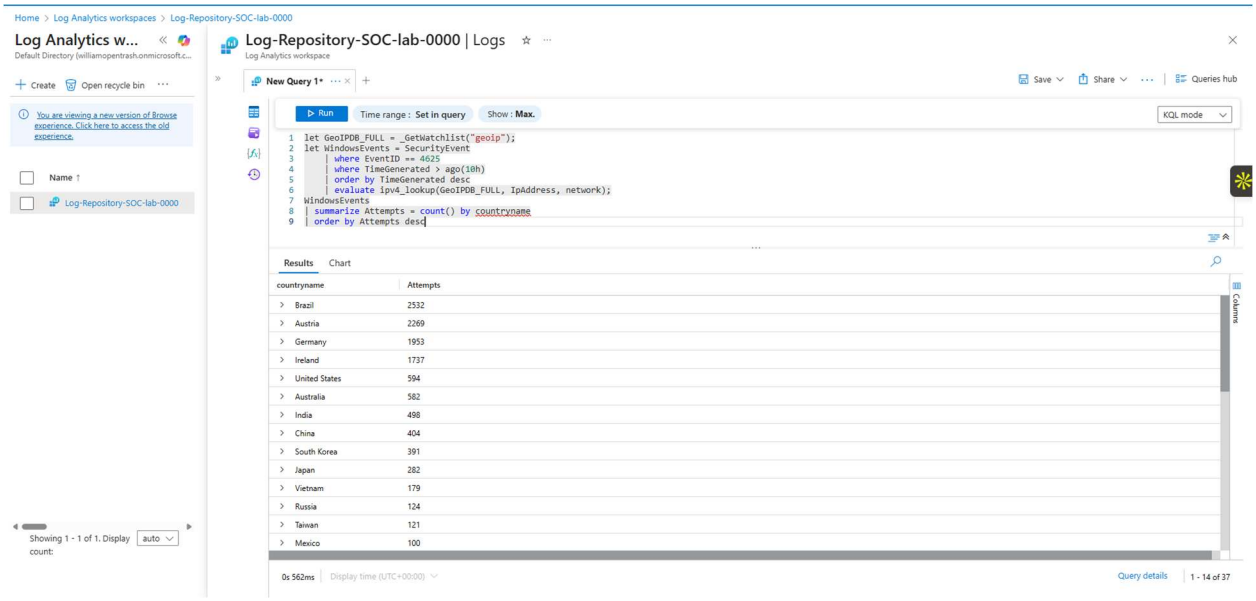

Figure 125: Country breakdown table.

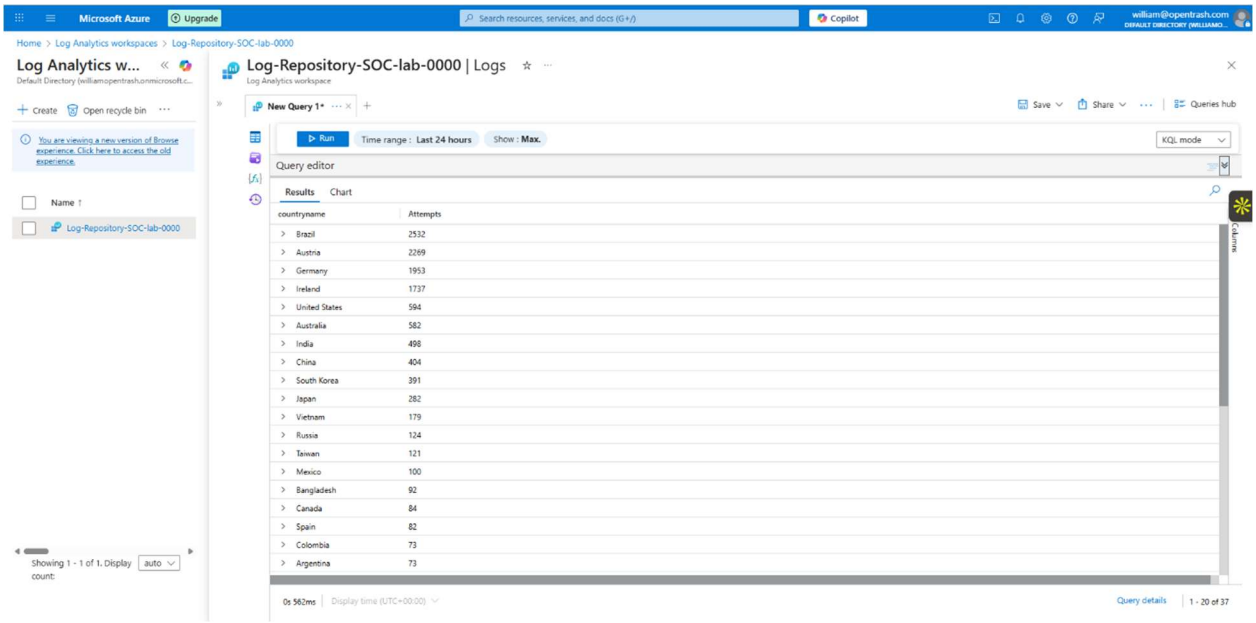

Figure 26: KQL summary of login attempts by country.

8.3 Behavioral Patterns

Analysis revealed:

- Attack bursts occurring in waves
- Multiple IPs from the same region targeting the same account
- Consistent use of common usernames (e.g., “administrator,” “admin”)

9. Key Learnings

This project provided hands-on experience with cloud security monitoring, SIEM configuration, and adversary analysis. Key takeaways include:

Technical Learnings

- Centralized logging is essential for visibility and correlation.
- KQL is a powerful language for threat hunting and log analysis.
- Data enrichment dramatically improves the quality of security insights.
- Sentinel workbooks enable clear, real-time visualization of attack patterns.

Operational Learnings

- Exposing a single RDP port is enough to attract global attackers within minutes.
- Attackers often reuse the same IPs and target common usernames.
- Even a small honeypot can generate enterprise-scale telemetry.

Professional Learnings

- Documenting SOC workflows improves communication with technical and non-technical audiences.
- Building a cloud SOC lab demonstrates initiative and practical security skills valued by employers.