Home SOC Lab: Threat Detection & SIEM with Microsoft Sentinel

Daniel Koztepe

Introduction

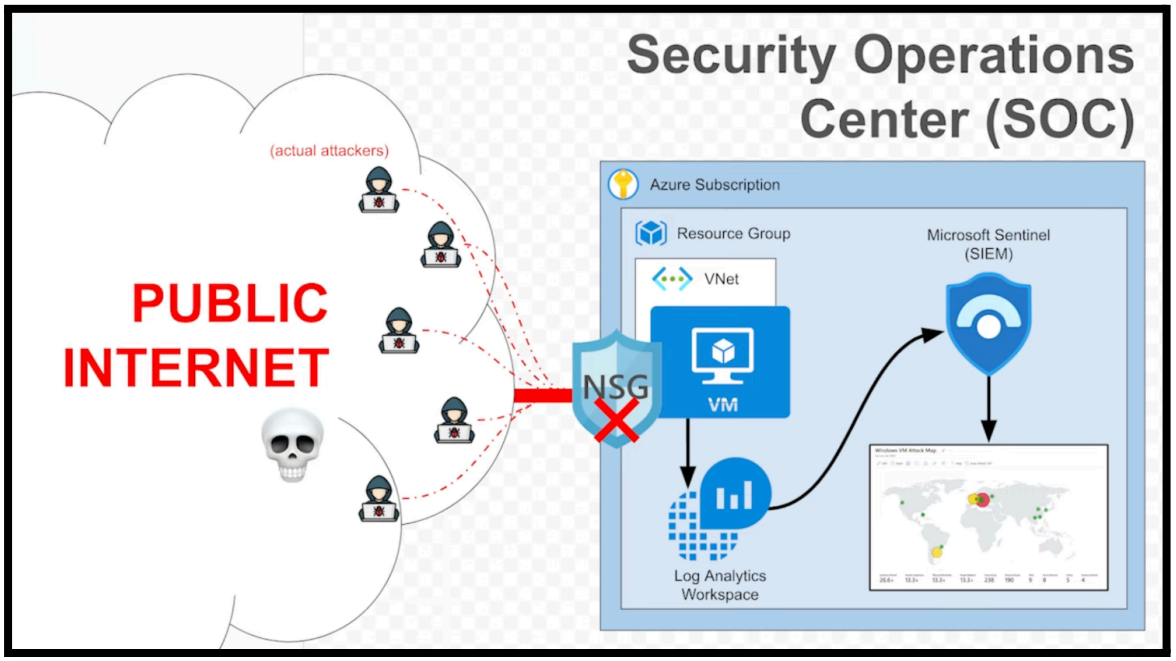
In this report, I will document the process of building a home **Security Operation Center** in **Microsoft Azure**. This project involves deploying a **honeypot Virtual Machine** to attract and capture **real-world cyber threats**. By collecting and analysing security logs with **Microsoft Sentinel (SIEM)**, I will demonstrate how to monitor failed login attempts, track attacker's geolocations, and visualise security threats.

Overview

I will walk you through the following key components of my SOC home lab:

1. **Create Free Azure Subscription:** Set up a free Azure account to access cloud resources needed for the lab.
2. **Create the Honey Pot (Azure Virtual Machine):** Deploy a Windows 10 VM, expose it to the internet, and configure it as a honeypot to attract attackers.
3. **Logging into the VM and inspecting logs:** Simulate failed login attempts, collect security event logs, and analyse unauthorised access attempts using Event Viewer.
4. **Log Forwarding and KQL:** Forward security logs to Log Analytics Workspace and use KQL (Kusto Query Language) to analyse login attempts.
5. **Log Enrichment and Finding Location Data:** Import geolocation into Microsoft Sentinel to map attacker IP addresses to their geographic locations.
6. **Attack Map Creation:** Use Sentinel Workbooks to visualise attacker activity on a map, providing insights into global threat sources.

Home SOC Architecture Diagram



Creating a Free Azure Subscription Account

What is Microsoft Azure?

Microsoft Azure is a flexible cloud platform that helps users create and manage digital solutions with ease and scalability, such as websites, virtual machines, and databases.

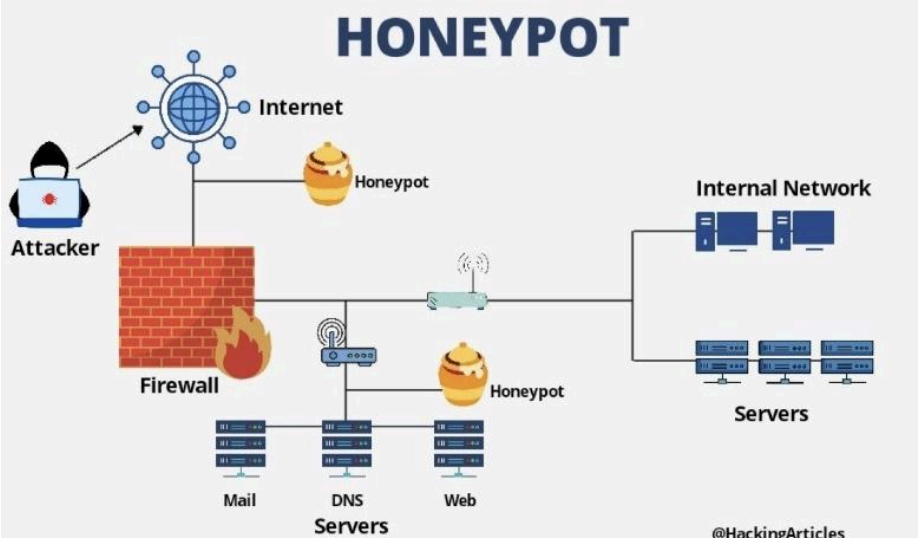
First, I'll have to create a free Azure account.

This video has shown me how to create an account for free!

Free Azure Account Setup

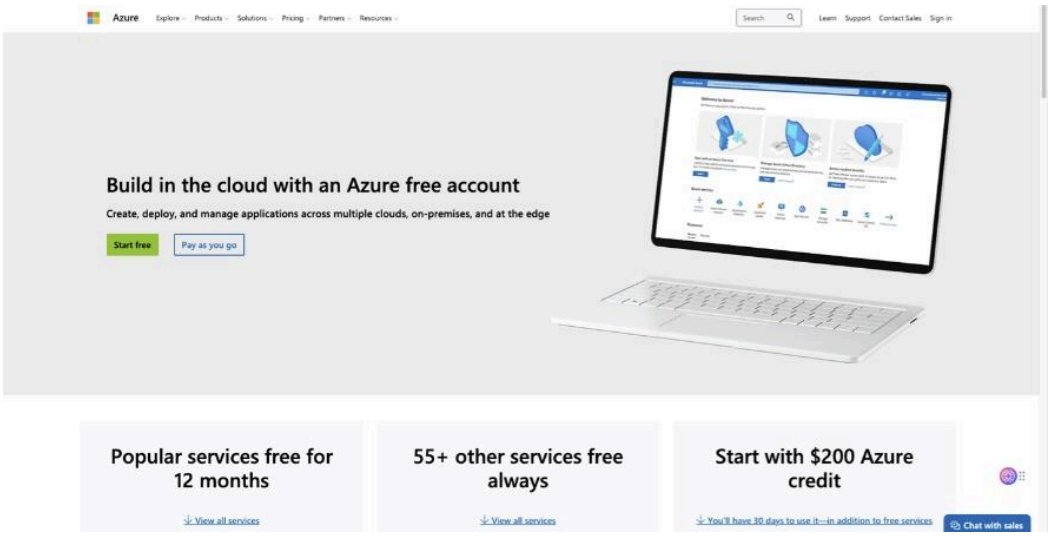
Creating the Honey Pot (Azure Virtual Machine)

What is Honey Pot?



A **honeypot** in cybersecurity is a decoy system or network intentionally designed to attract and trap attackers, such as hackers, malware, or bots. It mimics a legitimate target like a server, application, or database, but is isolated and monitored to study malicious behavior, gather intelligence, or divert threats away from real systems.

Go to: https://portal.azure.com and search for virtual machines



In this phase, I'll be creating a Windows 10 virtual machine that has a name that can attract attackers such as “CORP-NET-SOUTH-1”. If the PC name seems realistic then there’s a higher chance of getting intruded.

Next, I will remove the RDP rule from the Inbound Security Rules to intentionally expose Remote Desktop Protocol (RDP) traffic to the internet, creating a vulnerability. Since RDP is commonly used for remote access, making it publicly accessible expands the attack surface, simplifying the detection of unauthorized access attempts in our security logs. Afterward, I’ll add a new rule to open all ports, further increasing the system’s exposure to potential attacks.

- Create a new resource group and give it a name (honeypot-lab)

A resource group is a container that helps organize and manage related cloud resources.

Instance Details

- Give your virtual machine a name (CORP-NET-SOUTH-1)
- Choose a recommended region: (UK-South)
- Availability options: No infrastructure redundancy required
- Security type: Standard
- Image: Windows 10 Pro, version 22H2 - x64 Gen2
- VM Architecture: x64
- Size: Default is fine (Standard_D2s_v3 – 2vcpus, 8 GiB memory)

Administrator account

- Set up a username and password for the virtual machine.

IMPORTANT: these identification details will be used to log into the virtual machine. (Make sure to keep them in mind)

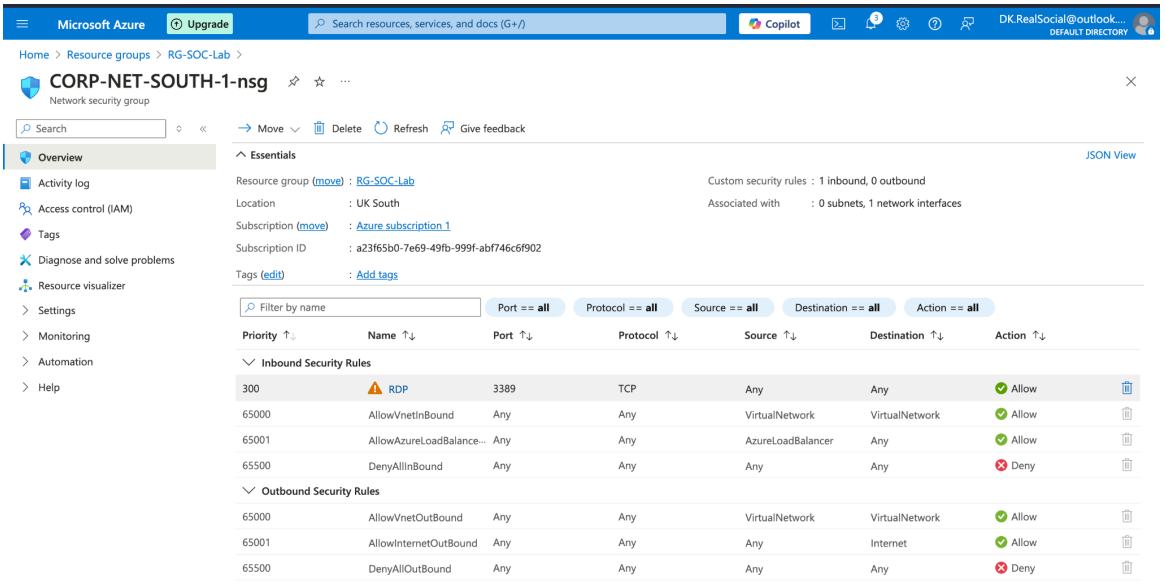
Inbount port rules

- Public inbound ports -> Allow selected ports: RDP (3389)

Licensing

- Confirm Licensing
- Select Next : Disks >

Before:



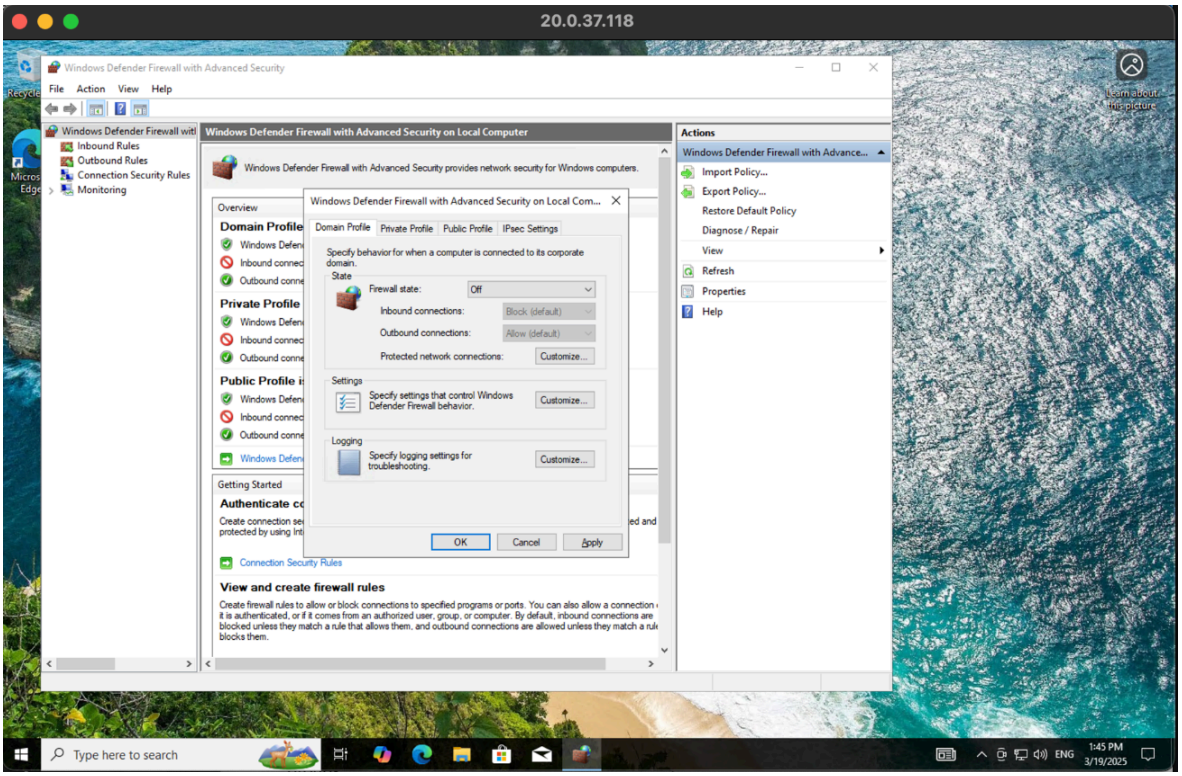
After:

Priority	Name	Port	Protocol	Source	Destination	Action
Inbound Security Rules						
100	DANGERAllowAnyCustomAnyInbound	Any	Any	Any	Any	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

In this stage, I've logged into my VM with the application “Windows App” where I can identify and test my VM later on in this document, however I've gone on and removed the Window Firewalls.

Here are the steps:

Start -> wf.msc -> properties -> all off



Lastly, I'll ping the VM to check if the machine is receiving bytes without any restrictions, and in this case the firewall is not preventing unauthorised devices from discovering the VM. The honeypot is now set and ready!

Before:

```
Last login: Mon Mar 24 14:00:09 on console
danielkoztepe@USERs-MacBook-Pro ~ % ping 20.0.37.118
PING 20.0.37.118 (20.0.37.118): 56 data bytes
Request timeout for icmp_seq 0
Request timeout for icmp_seq 1
Request timeout for icmp_seq 2
Request timeout for icmp_seq 3
Request timeout for icmp_seq 4
Request timeout for icmp_seq 5
^C
--- 20.0.37.118 ping statistics ---
7 packets transmitted, 0 packets received, 100.0% packet loss
```

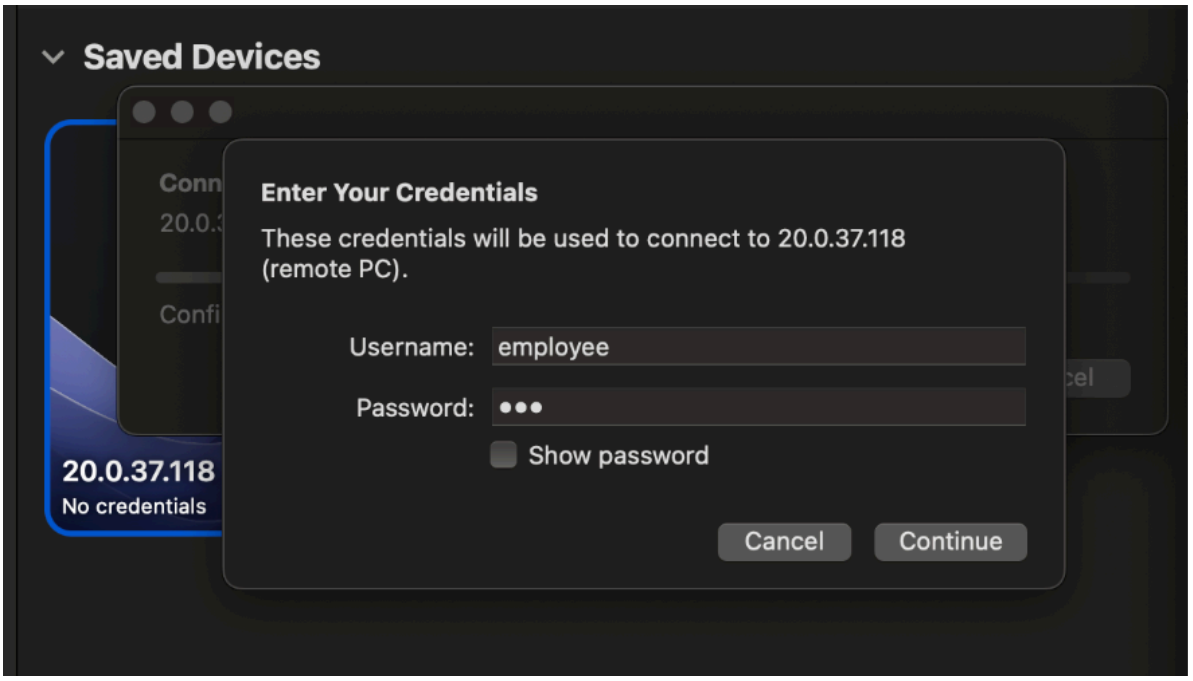
After:

```
danielkoztepe — -zsh — 88×25
~ — -zsh
Last login: Wed Mar 19 09:37:43 on ttys000
danielkoztepe@USERs-MacBook-Pro ~ % ping 20.0.37.118
PING 20.0.37.118 (20.0.37.118): 56 data bytes
64 bytes from 20.0.37.118: icmp_seq=0 ttl=250 time=8.362 ms
64 bytes from 20.0.37.118: icmp_seq=1 ttl=250 time=8.575 ms
64 bytes from 20.0.37.118: icmp_seq=2 ttl=250 time=8.541 ms
64 bytes from 20.0.37.118: icmp_seq=3 ttl=250 time=8.747 ms
64 bytes from 20.0.37.118: icmp_seq=4 ttl=250 time=8.910 ms
64 bytes from 20.0.37.118: icmp_seq=5 ttl=250 time=8.669 ms
^C
--- 20.0.37.118 ping statistics ---
6 packets transmitted, 6 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 8.362/8.634/8.910/0.171 ms
danielkoztepe@USERs-MacBook-Pro ~ %
```

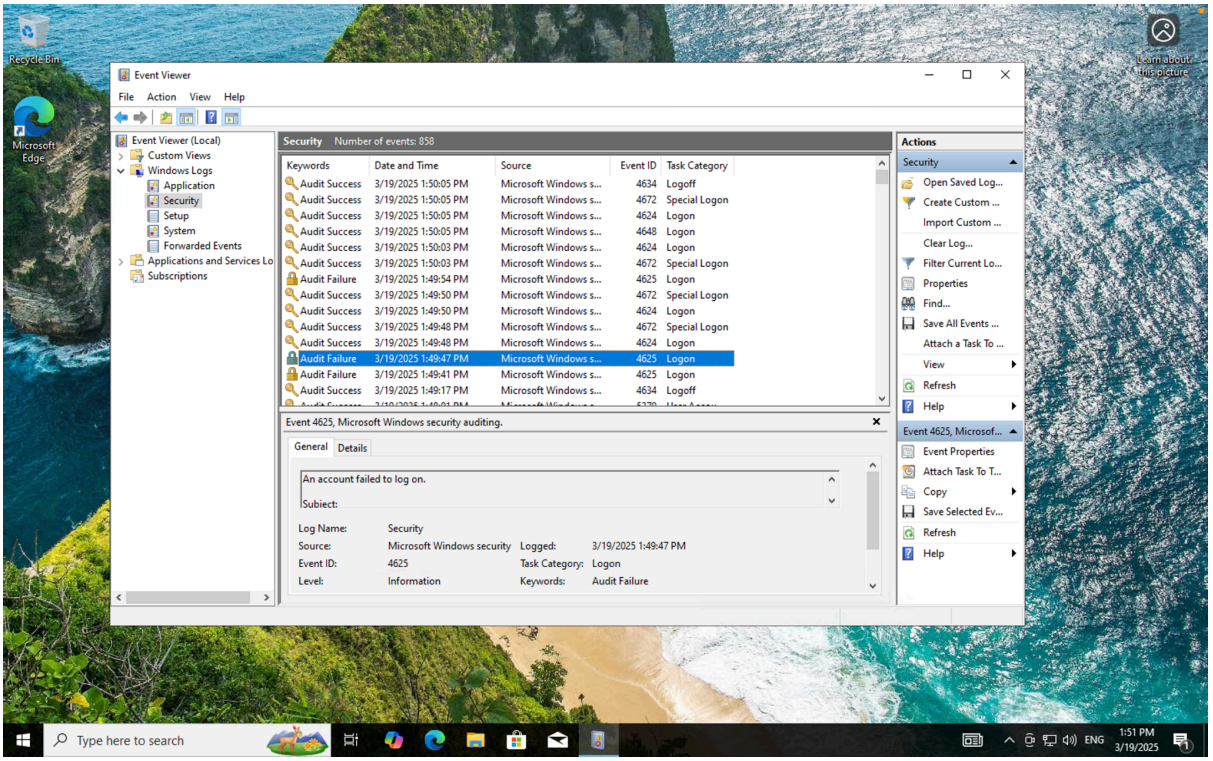
Logging into the VM and Inspecting Logs

Fail 3 logins as “employee” (or some other username):

To inspect some logs, I'll be doing a simple failed login into the VM a few times so I can generate some understanding on how authentication failures are recorded.



Opening up Event Viewer and Inspecting the Security Logs:

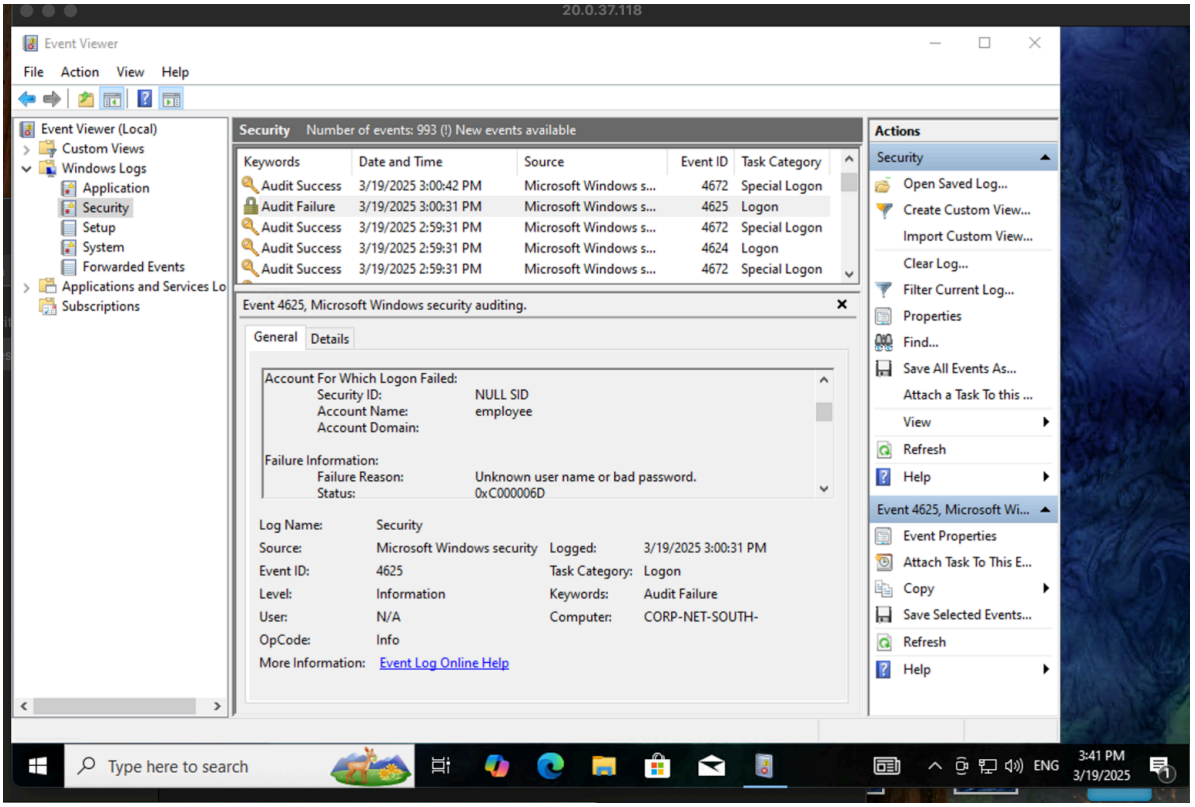


As we inspect the Windows Security logs in detail we can look into the Event ID 4625 which indicates a failed logon attempt. This will be crucial for me as I can view details such as their Source Network Address to identify where the intrusion originated.

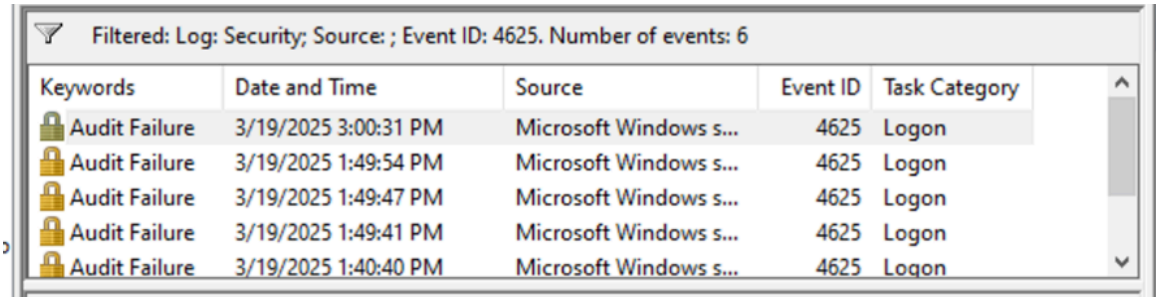
By combining the Event Viewer to the Microsoft Sentinel it’ll generate a well detailed log known as SIEM (Security Information and Event Management). These combinations will

allow an automated trigger alert which can speed up the process for incident reports in the real world.

Microsoft Sentinel also includes compliances such as the **GDPR** and **ISO 27001** which are crucial for organisations framework of their data protection.



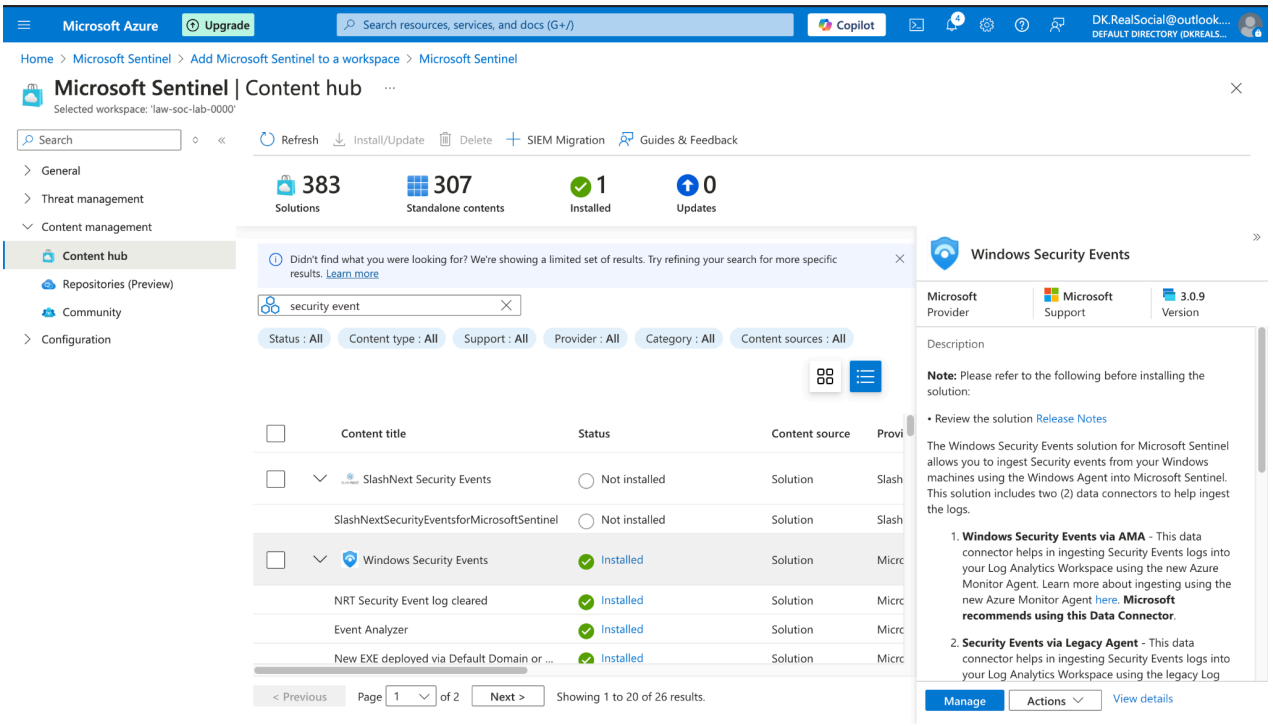
Here's a filtered version of the Event ID 4625:



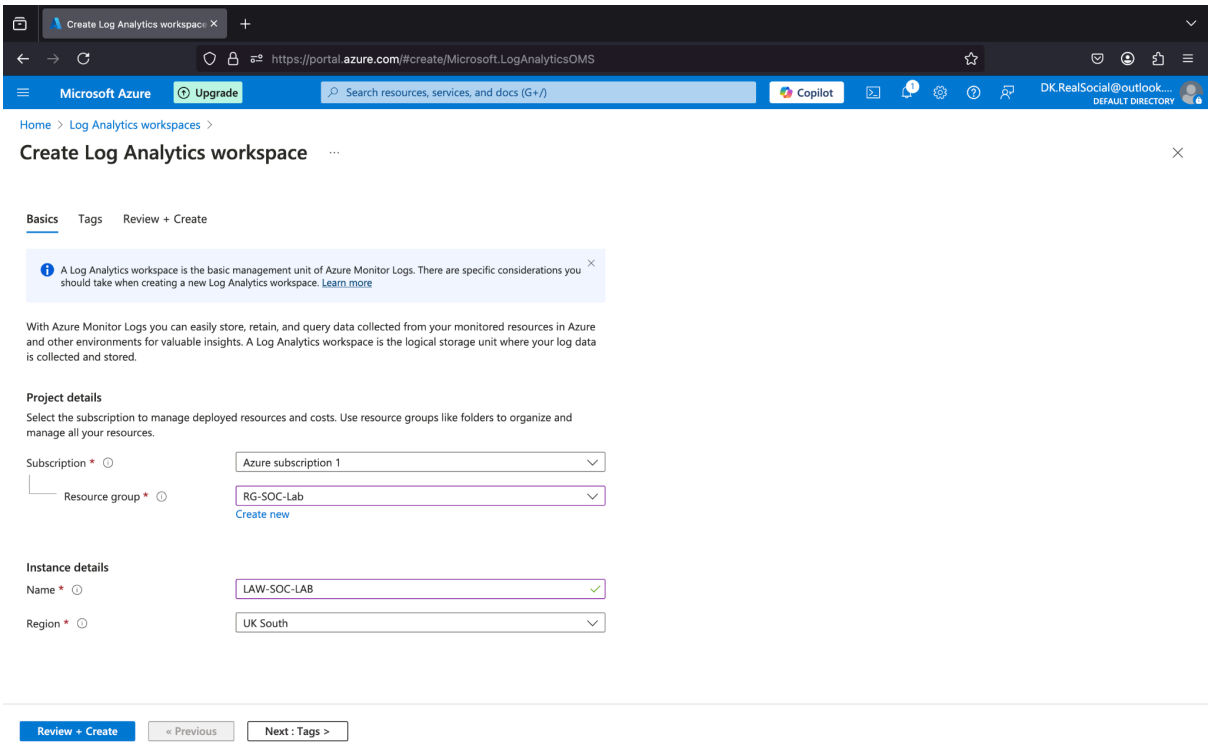
Log Forwarding and KQL (Kusto Query Language)

KQL (Kusto Query Language): KQL is a query language used to retrieve, analyze, and visualize large datasets, particularly for log and event data, within Azure Log Analytics and Azure Monitor.

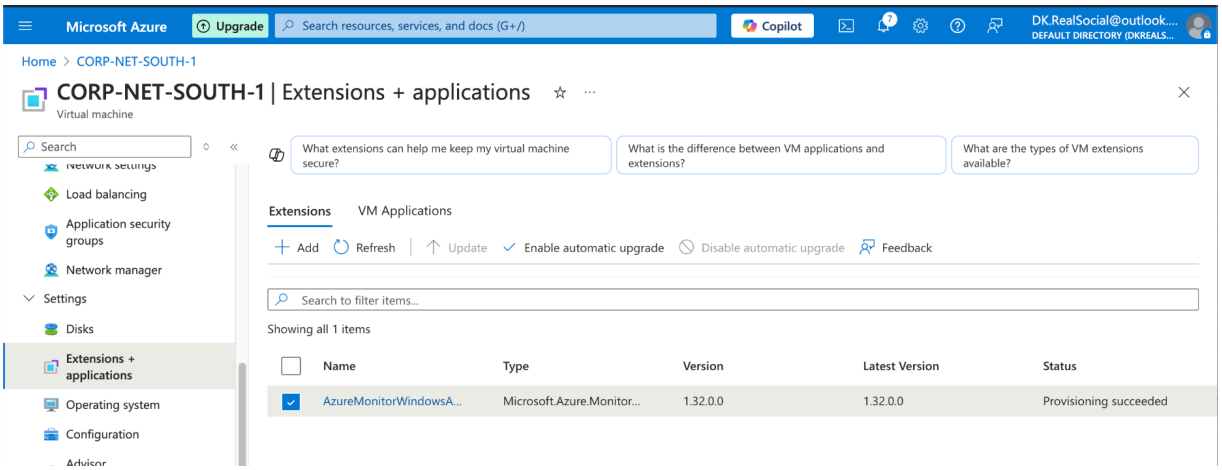
In this stage, I've activated the free trial from **Microsoft Sentinel** through the **Azure trial**, enabling me to configure the **Windows Security Event** in Sentinel. This configuration allows the continuous collection, analysis, and correlation of security logs from Windows-based systems to detect, alert, and respond to potential security incidents, such as failed login attempts.



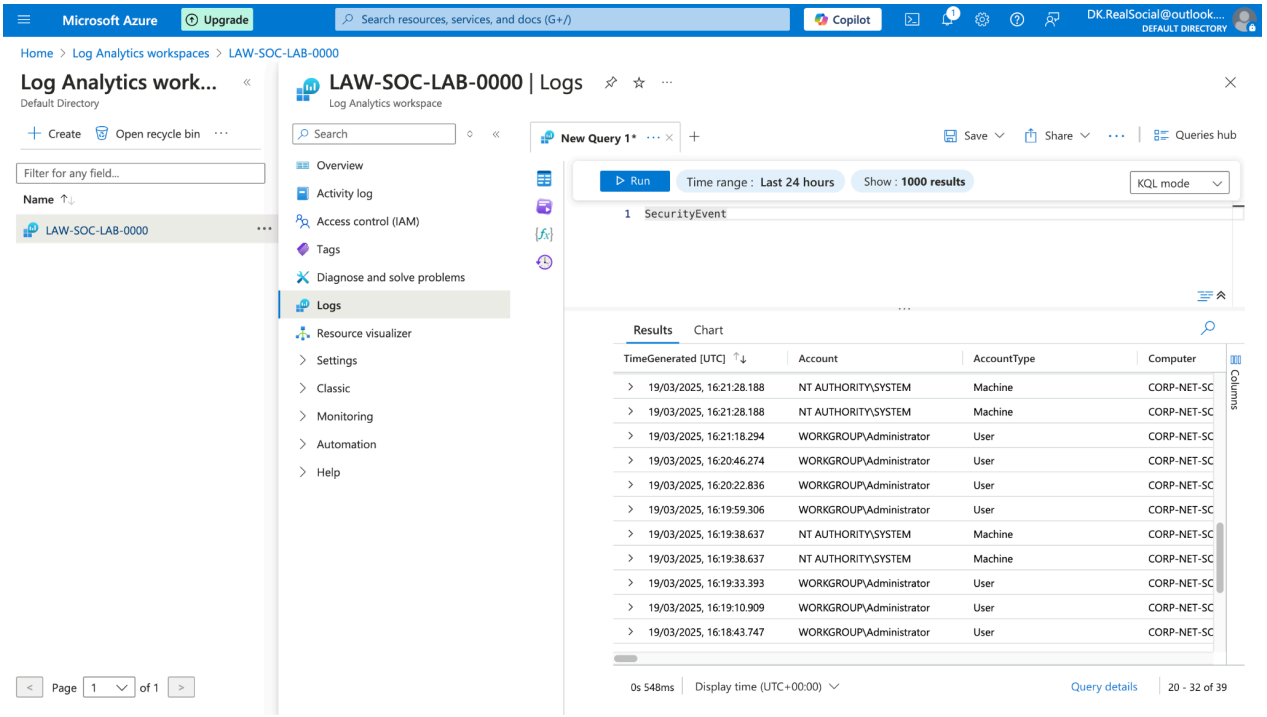
Next, I've created a **Central Log Repository** known as **LAW**. This will allow me to collect, analyze and query security logs from Azure's extension known as "AzureMonitorWindowsAgent". This tool helps with data collection in which it is then centralised to LAW where I can begin my security investigations.



The Data Collection Extension Tool:



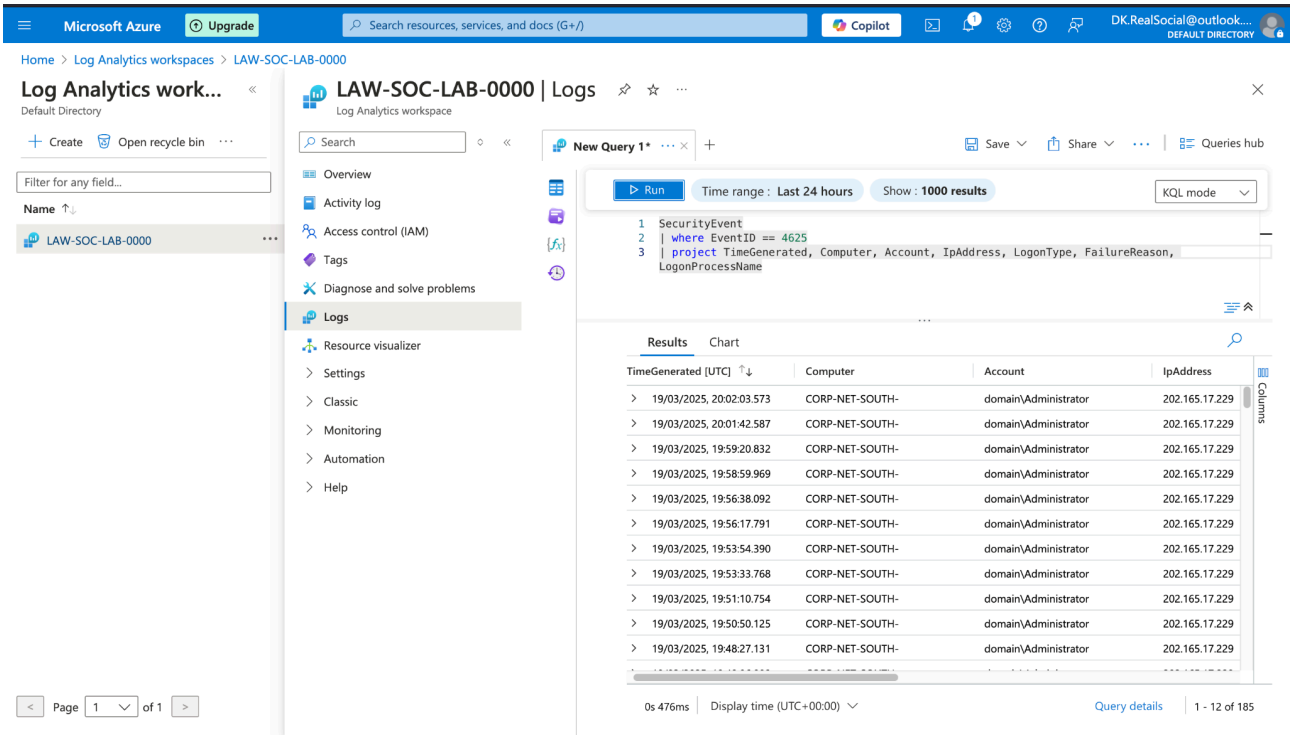
The Log Report of the LAW:



By querying for logs within the **Log Analytics Workspace (LAW)** using Kusto Query Language (KQL), I was able to identify multiple failed login attempts from the same IP address over an extended period.

The logs revealed a **brute-force attack** originating from an IP address located in **Malaysia**, with a high frequency of login attempts spanning from **4:18 PM to 8:02 PM**. This activity was consistent, suggesting an automated attack trying to gain unauthorized access.

View the IP Address



To save time and find specific logs, I can use KQL to allow me to filter through large amounts of data, quickly and precisely.

KQL Commands:

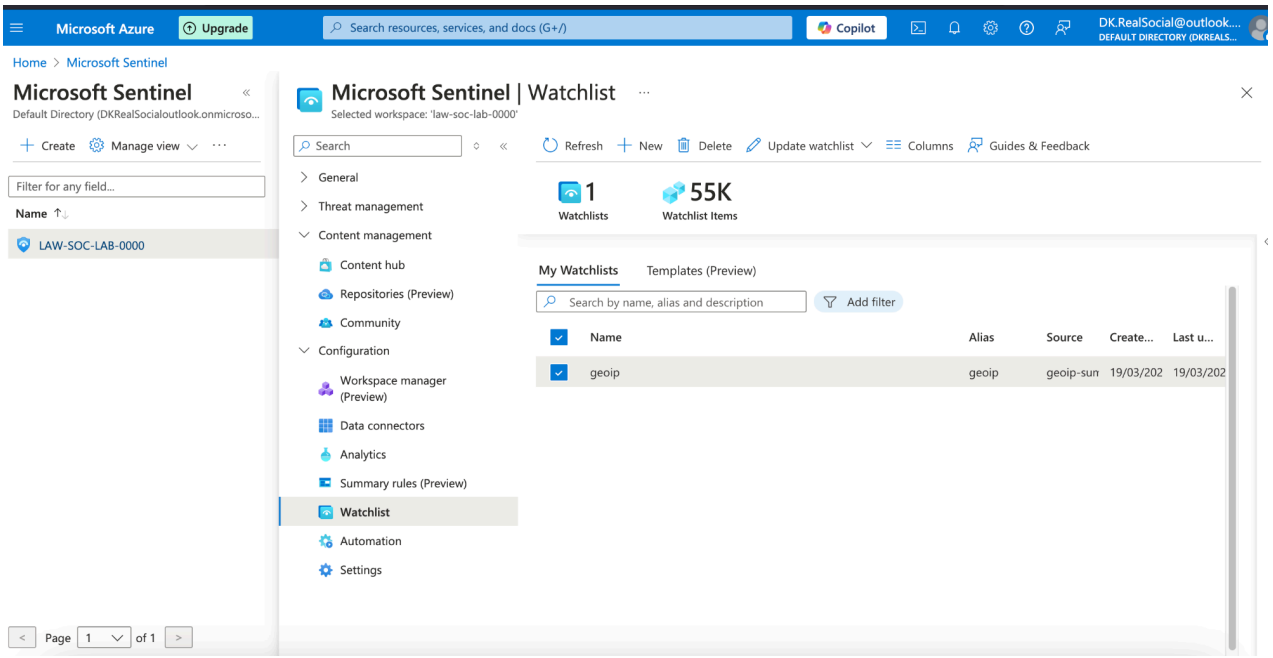
```
 SecurityEvent
| where EventID == 4625
| project TimeGenerated, Computer, Account, AccountType, IpAddress, LogonType,
FailureReason, LogonProcessName
```

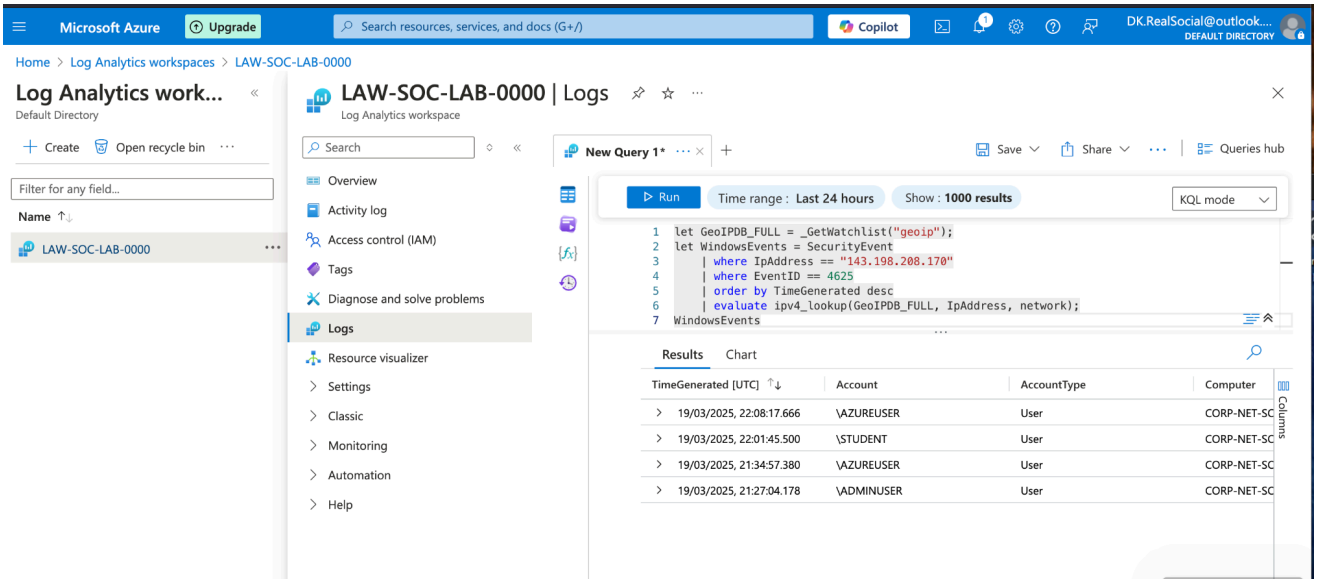
Log Enrichment and Finding Location Data

I'll be observing the SecurityEvent logs in the Log Analytics Workspace as there is no location data, only IP address, which we can use to derive the location data.

I'm going to import a spreadsheet known as a "Sentinel Watchlist" which contains geographic information for each block of IP addresses.

Spreadsheet File: geoip-summarised.csv





As you can see below, I was able to identify the attacker's location along with their attempted login attempts shown above. This was a clear sight that the attacker was trying to gain access to my Azure account. Better luck next time!

Results Chart

LastUpdatedTimeUTC [UTC]	SearchKey	cityname	countryname
19/03/2025, 21:35:32.009	143.198.0.0/16	Miami	United States
19/03/2025, 21:35:32.009	143.198.0.0/16	Miami	United States
19/03/2025, 21:35:32.009	143.198.0.0/16	Miami	United States
19/03/2025, 21:35:32.009	143.198.0.0/16	Miami	United States

KQL Command to observe the logs that have geographic information, so you can see where the attacks are coming from:

```
let GeoIPDB_FULL = _GetWatchlist("geoip");
let WindowsEvents = SecurityEvent
    | where IpAddress == <"IP Address">
    | where EventID == 4625
    | order by TimeGenerated desc
    | evaluate ipv4_lookup(GeoIPDB_FULL, IpAddress, network);
WindowsEvents
```

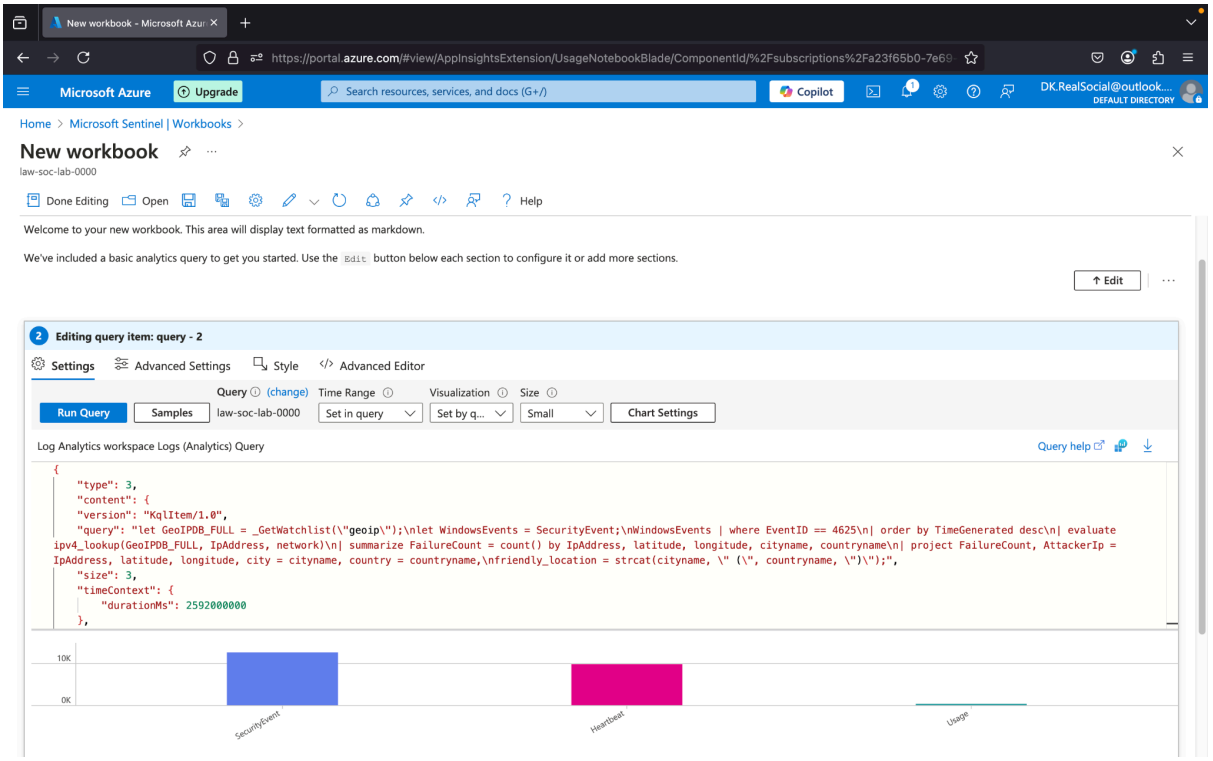
```
SecurityEvent
| where EventID == "4625"
```

I'll be using this red command beforehand to find the IP address that I want to analyze.

Attack Map Creation

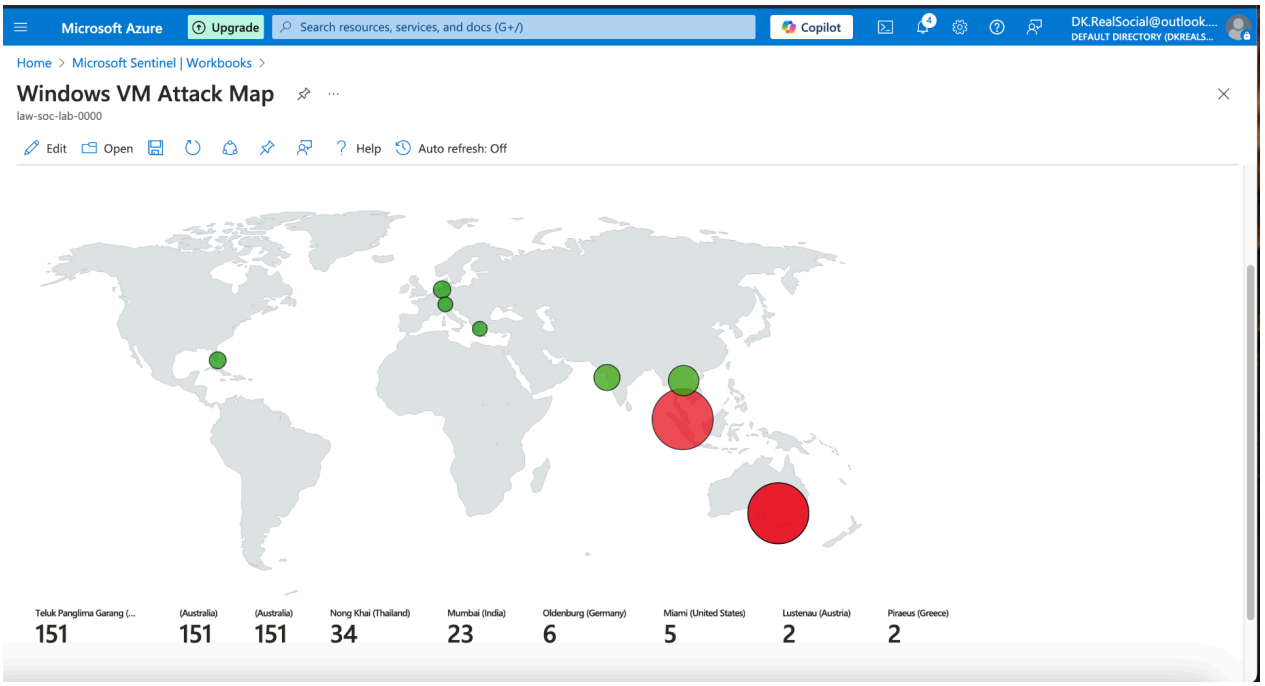
For the last stage, I'll be generating a visual geographic map of all the attacks that were happening towards my VM. This provides insights into threat sources and attack trends.

The following step is to create a new Workbook

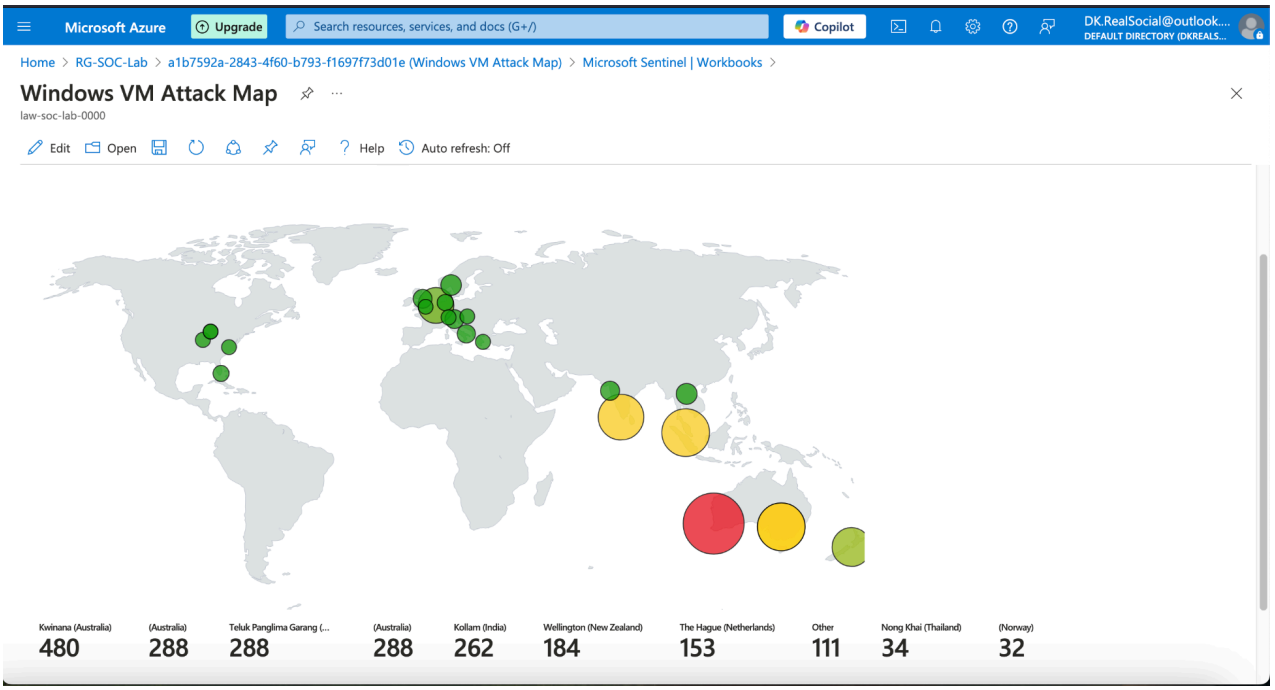


This part will also simulate real-world cyber threats, enabling better understanding of security incidents, logging mechanisms, and threat analysis techniques.

Attack Map 19/3/2025 at 11pm:



Attack Map 20/3/2025 at 11.30am:



By analyzing the attack maps, I observed that most attacks on my Azure VM originated from **Australia**. Over **12 hours**, the number of attempted connections ranged from **151 to 480**, indicating a significant volume of scanning activities. This seems likely to be caused by automated scanners or a targeted attack effort.

Conclusion

In this project, I successfully built a home Security Operation Center (SOC) in Microsoft Azure to monitor and analyse real-world cyber threats. By deploying a honeypot Virtual Machine and integrating it with Microsoft Sentinel, I was able to collect, analyse, and visualise security logs. This process demonstrated how attackers attempt to compromise systems, allowing for deeper insights into their techniques and geolocations.

Through log forwarding, KQL analysis, and visualisation tools like Sentinel Workbooks, I effectively tracked failed login attempts and identified potential threats. This hands-on experience reinforced the importance of proactive security monitoring and the power of cloud-based SIEM solutions. Overall, this project highlights how organisations can leverage Azure's security tools to enhance their threat detection and response capabilities.

Reference

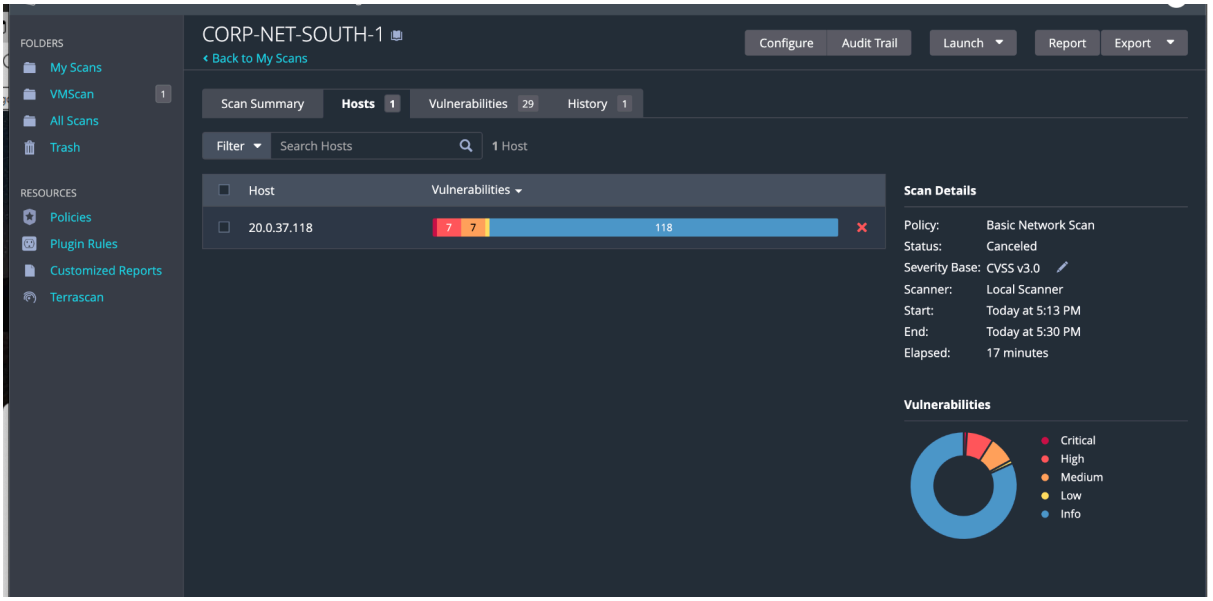
https://youtu.be/g5JL2RIbThM?si=jIhiGjGkPeJ4KstL

Scan & Fix Vulnerabilities

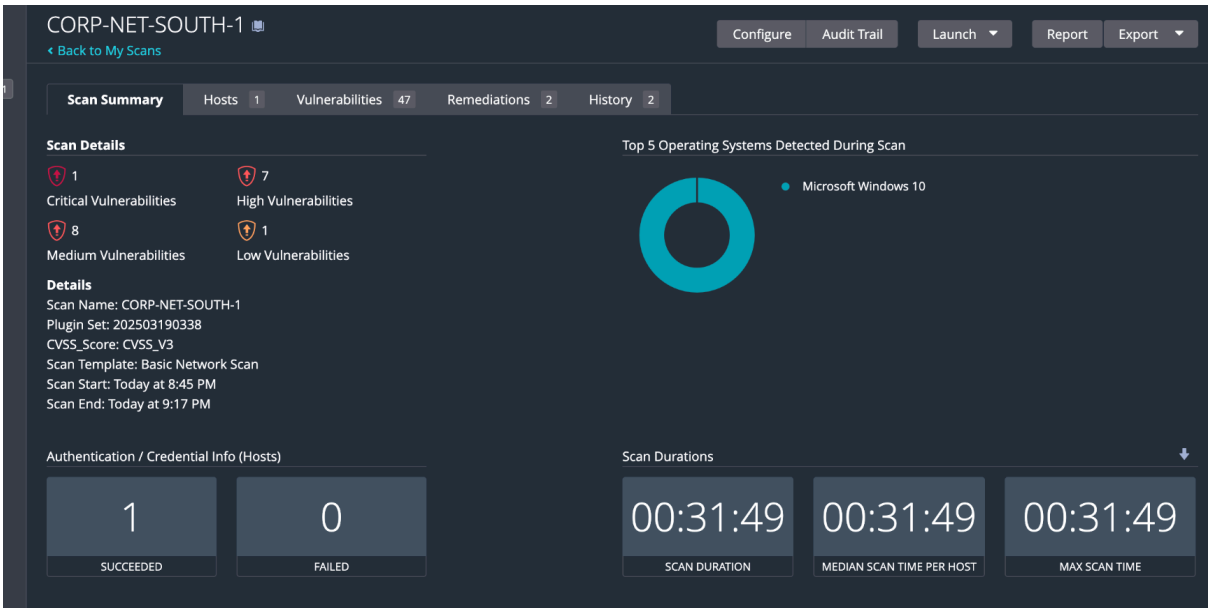
For the last part of my documentation, i'll be identifying, assessing and remediating vulnerabilities in a secure environment with the essential tools of Tenable and also continuing with Microsoft Azure

Tenable (Nessus Professional) 7 Day Trial

First Scan:



Last Scan:


CORP-NET-SOUTH-1
Back to My Scans
Configure
Audit Trail
Launch
Report
Export
Scan Summary
Hosts 1
Vulnerabilities 47
Remediations 2
History 2
Scan Details
1
Critical Vulnerabilities
7
High Vulnerabilities
8
Medium Vulnerabilities
1
Low Vulnerabilities
Details
Scan Name: CORP-NET-SOUTH-1
Plugin Set: 202503190338
CVSS_Score: CVSS_V3
Scan Template: Basic Network Scan
Scan Start: Today at 8:45 PM
Scan End: Today at 9:17 PM
Top 5 Operating Systems Detected During Scan
Microsoft Windows 10
Authentication / Credential Info (Hosts)
1
SUCCEEDED
0
FAILED
Scan Durations
00:31:49
SCAN DURATION
00:31:49
MEDIAN SCAN TIME PER HOST
00:31:49
MAX SCAN TIME