

# Honeypot-Based Threat Detection: Real-World Attack Analysis with Azure Sentinel:

Prepared by: [Yasif Farook](#)  
Date of Creation: 10/03/2025

# Table of Contents

<b>1. Project Overview .....</b>	<b>2</b>
<b>2. Setting up Relevant Machines .....</b>	<b>3</b>
○ 2.1 Setting Up the Windows Server Virtual Machine .....	3
○ 2.2 Creating a Log Analytics Workspace under the VM .....	4
○ 2.3 Configuring Security Settings .....	6
○ 2.4. Connecting honeypot-vm to its Workspace .....	7
<b>3. Integrating Sentinel to the Workspace .....</b>	<b>9</b>
○ 3.1 Navigating to Microsoft Sentinel .....	9
○ 3.2 Adding Sentinel to the LAW-honeypot1 Workspace .....	9
<b>4. Starting the Virtual Machine .....</b>	<b>10</b>
○ 4.1 Navigating to Virtual Machines in Azure .....	10
○ 4.2 Connecting to the Virtual Machine .....	11
<b>5. Monitoring Inside the VM .....</b>	<b>13</b>
○ 5.1 Event Viewer .....	13
○ 5.2 Configuring the VM Firewall .....	17
○ 5.3 Generating Logs Inside the VM .....	19
<b>6. Upgrading Logs .....</b>	<b>23</b>
6.1 Creating a custom log in Azure .....	23
<b>7. Visualising Attack Patterns with Azure Sentinel Map .....</b>	<b>30</b>
○ 7.1 Integrating Microsoft Sentinel .....	30
○ 7.2 Latitude & Longitude-Based Map .....	32
○ 7.3 Country-Based Map .....	33
<b>8. Results and Analysis .....</b>	<b>34</b>
<b>9. Challenges and Solutions .....</b>	<b>35</b>
<b>10. Project Summary .....</b>	<b>35</b>

# 1.0 Project Overview

This project involves setting up a honeypot *virtual machine* (VM) to detect and analyse real Remote Desktop Protocol (RDP) attack attempts, including brute-force attacks, using Azure Sentinel. Unlike simulated attack scenarios, this honeypot was actively targeted by real-world attackers, providing genuine security event data for analysis. The project demonstrates the application of SOC (Security Operations Center) methodologies in monitoring, detecting, and responding to unauthorised login attempts.

## Key Activities:

- Deployment of a Windows Server VM as a honeypot to attract real attack attempts.
- Real-time monitoring of RDP brute-force attempts using Windows Event Viewer.
- Integration with Azure Log Analytics Workspace and Sentinel to aggregate, store, and analyse security event data.
- Threat investigation using Kusto Query Language (KQL) for querying security logs.
- Geolocation enrichment to map attack origins and identify high-risk regions.
- Visualisation of attack trends using Sentinel's Map feature for real-time monitoring.

## 1.1 Tools and Technologies

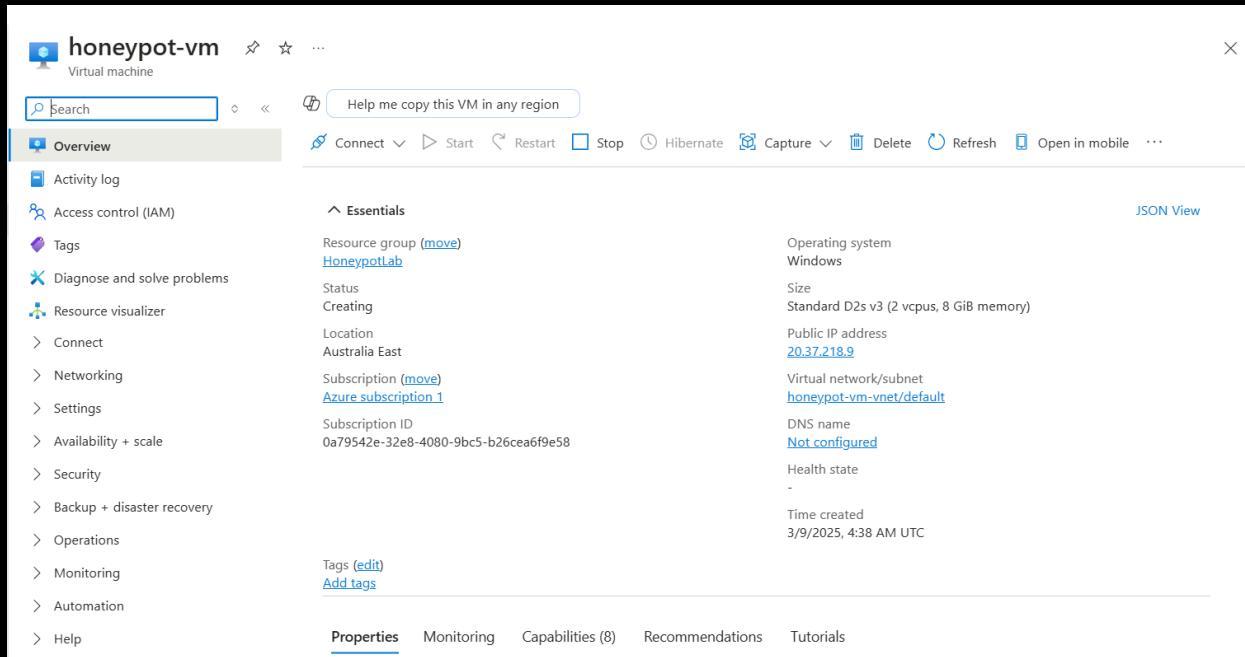
Tool	Purpose
Microsoft Azure	Hosts the honeypot VM and manages cloud-based security infrastructure.
Windows Server (Honeypot VM)	Acts as a decoy to attract and log unauthorised login attempts.
Event Viewer	Captures security event logs, including failed

	login attempts.
Azure Log Analytics	Collects, stores, and enables querying of event logs.
Azure Sentinel (SIEM)	Provides real-time threat detection, alerting, and visualization of attacks.
Kusto Query Language (KQL)	Enables detailed log analysis and correlation of security data.
IP Geolocation API	Enriches logs with location-based intelligence.

## 2.0 Setting Up Relevant Machines (VM, Log Analytics Workspace)

### 2.1 Setting Up the Windows Server Virtual Machine

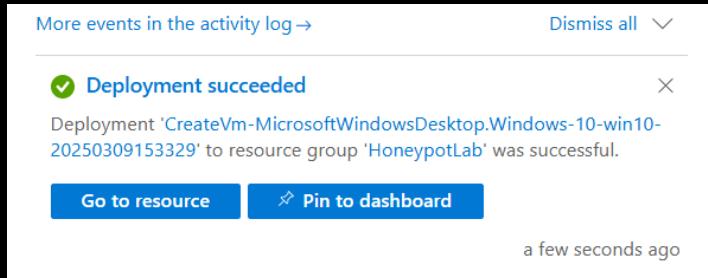
The VM is set up within the Azure environment. The purpose of this machine is to act as a decoy system, deliberately configured to attract RDP brute force attempts.



The screenshot shows the Azure portal interface for a virtual machine named 'honeypot-vm'. The left sidebar contains navigation links such as Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Resource visualizer, Connect, Networking, Settings, Availability + scale, Security, Backup + disaster recovery, Operations, Monitoring, Automation, and Help. The main content area displays the 'Overview' tab for the VM. Key details shown include:

- Resource group:** HoneypotLab
- Status:** Creating
- Location:** Australia East
- Subscription:** Azure subscription 1
- Subscription ID:** 0a79542e-32e8-4080-9bc5-b26cea6f9e58
- Operating system:** Windows
- Size:** Standard D2s v3 (2 vcpus, 8 GiB memory)
- Public IP address:** 20.37.218.9
- Virtual network/subnet:** honeypot-vm-vnet/default
- DNS name:** Not configured
- Health state:** -
- Time created:** 3/9/2025, 4:38 AM UTC

At the bottom of the page, there are tabs for Properties, Monitoring, Capabilities (8), Recommendations, and Tutorials. There is also a 'Tags' section with links to edit or add tags.



The Windows Server VM serves as a honeypot (hence the name "*honeypot-vm*"), to attract login attempts, which will then be logged and monitored under the **Log Analytics Workspace**, enabling real-time detection and analysis of activity. This could include failed login attempts or potential brute-force attacks. Unlike typical systems, the honeypot will be left intentionally vulnerable to lure real-world attackers into initiating login attempts, which would then be captured and analysed.

## 2.2 Creating a Log Analytics Workspace under the VM

The Log Analytics Workspace "*LAW-honeypot1*" was created in the Australia East region. The workspace is responsible for collecting and storing all logs generated by the honeypot VM, enabling effective monitoring and querying for security events. By associating this workspace with the honeypot VM, logs related to login attempts and security events are centrally stored for analysis.

**Create Log Analytics workspace** ...

**Info** A Log Analytics workspace is the basic management unit of Azure Monitor Logs. There are specific considerations you should take when creating a new Log Analytics workspace. [Learn more](#) X

With Azure Monitor Logs you can easily store, retain, and query data collected from your monitored resources in Azure and other environments for valuable insights. A Log Analytics workspace is the logical storage unit where your log data is collected and stored.

**Project details**

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ	Azure subscription 1
Resource group * ⓘ	HoneypotLab
	<a href="#">Create new</a>

**Instance details**

Name * ⓘ	LAW-honeypot1
Region * ⓘ	Australia East

Review + Create « Previous Next : Tags >

Workspace successfully created:

**Create Log Analytics workspace** ...

**Validation passed**

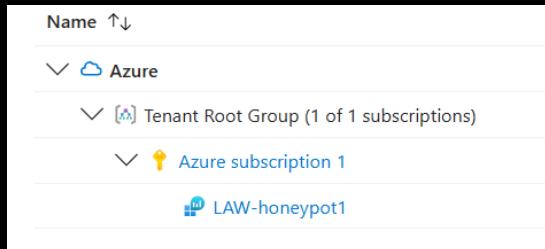
**Deployment succeeded**

Deployment 'Microsoft.LogAnalyticsOMS' to resource group 'HoneypotLab' was successful.

Go to resource ↗ Pin to dashboard

2 minutes ago

Full breakdown of the structure of the storage of the honeypot VM and its corresponding log analytics workspace, stored in the Azure cloud:



## 2.3 Configuring Security Settings

*Microsoft Defender* is enabled for **LAW-honeypot1** to ensure that all relevant security events, such as login attempts, are captured. The settings ensure that both successful and failed login attempts are logged in *Event Viewer* for further investigation.

This configuration is critical for accurately identifying attack patterns.

- Correctly configured *Microsoft Defender* settings for **LAW-honeypot1**

A screenshot of the Microsoft Defender plan configuration page. At the top, it says "Microsoft Defender plans will apply to: 0 Azure and 0 non-Azure resources reporting to this workspace". Below this, there's a section titled "Select Defender plan" with a button "Enable all plans". A table lists three plans:

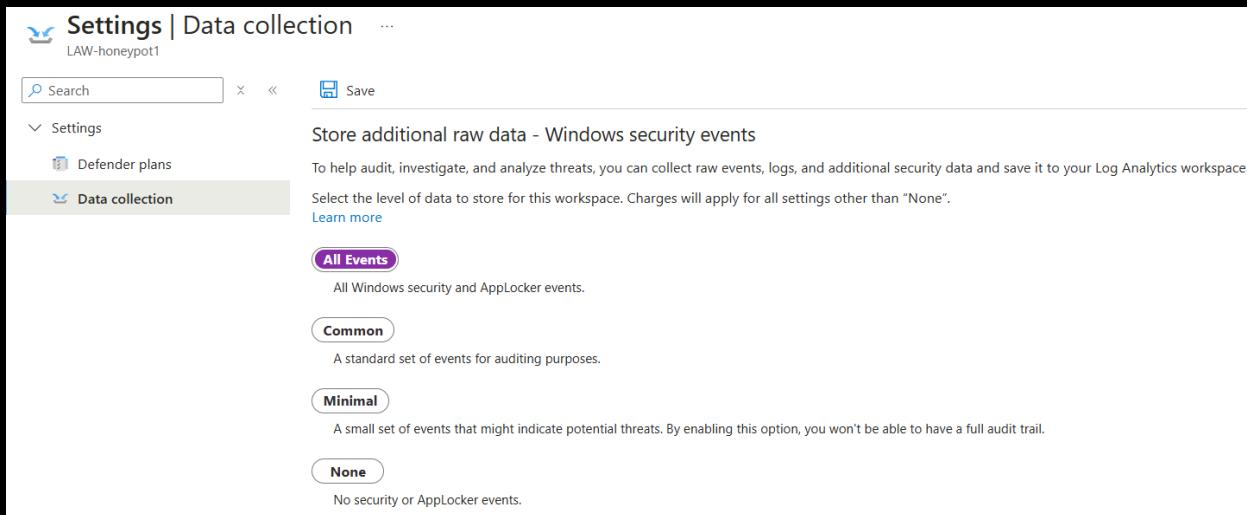
Plan	Pricing*	Resource quantity	Plan
Foundational CSPM	Free	0 servers	<input checked="" type="button"/> Off <input type="button"/> On
Servers	\$15/Server/Month ⓘ	0 servers	<input checked="" type="button"/> Off <input type="button"/> On
SQL servers on machines	\$15/Instance/Month \$0.015/Core/Hour ⓘ	0 servers	<input type="button"/> Off <input checked="" type="button"/> On

\* The price displayed represents the list price prior to any discounts or special offers being applied.

- *Microsoft Defender* Settings successfully saved:

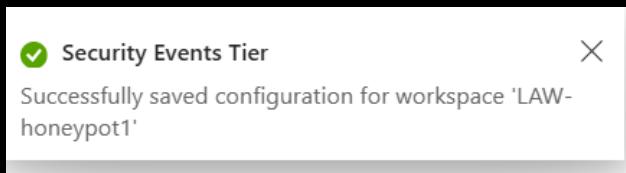
A screenshot of a success message in the activity log. It says "More events in the activity log →" and "Dismiss all" with a dropdown arrow. The message itself starts with a green checkmark icon and the text "Defender plans" followed by an "X" icon. Below that, it says "Microsoft Defender plan for workspace 'LAW-honeypot1' were saved successfully!" and "a few seconds ago".

By checking 'All Events', the VM's Event Viewer will display all log activity.



The screenshot shows the 'Settings | Data collection' page for a workspace named 'LAW-honeypot1'. The 'Data collection' tab is selected. A section titled 'Store additional raw data - Windows security events' explains that raw events, logs, and additional security data can be collected and saved to the Log Analytics workspace. It includes a note about charges for non-'None' levels. Below this, four options are listed: 'All Events' (selected), 'Common', 'Minimal', and 'None'. Each option has a brief description: 'All Events' collects All Windows security and AppLocker events; 'Common' collects a standard set for auditing; 'Minimal' collects a small set for potential threats; and 'None' collects no security or AppLocker events.

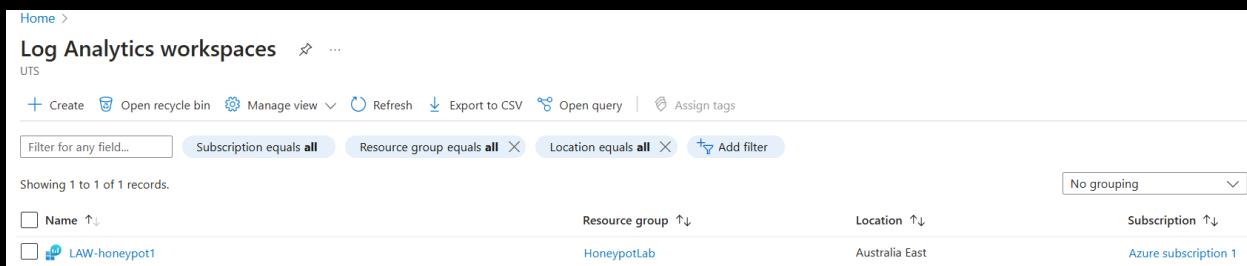
Successful configuration of *LAW-honeypot1*:



## 2.4 Connecting *honeypot-vm* to its Workspace

Once connected to the workspace, the honeypot VM will begin transmitting event logs in real-time. The connection between the VM and the workspace is crucial to ensure that all activity, including failed login attempts, is continuously monitored and captured for analysis in Azure Sentinel.

- Navigate to Log Analytics Workspaces:



The screenshot shows the 'Log Analytics workspaces' page. At the top, there are buttons for 'Create', 'Open recycle bin', 'Manage view', 'Refresh', 'Export to CSV', 'Open query', and 'Assign tags'. Below this is a search bar with filters: 'Subscription equals all', 'Resource group equals all', 'Location equals all', and 'Add filter'. A dropdown menu 'No grouping' is also present. The main table lists one record: 'LAW-honeypot1'. The columns are 'Name' (with an up-down arrow), 'Resource group' (sorted by 'HoneypotLab'), 'Location' (sorted by 'Australia East'), and 'Subscription' (sorted by 'Azure subscription 1').

- Select **LAW-honeypot1**

Home > Log Analytics workspaces >

## Log Analytics work...

UTS

+ Create ⚡ Open recycle bin ...

Filter for any field...

Name	...
LAW-honeypot1	...

LAW-honeypot1 Log Analytics workspace

Search

Delete

The Log Analytics agents (MMA.OMS) used to collect logs from virtual machines and servers will no longer be supported from August 31, 2024. Plan to migrate to Azure Monitor Agent before this date. [Learn more about migrating to Azure Monitor Agent](#)

**Essentials**

Resource group ([move](#)) **honeypotlab**

Status Active

Location Australia East

Subscription ([move](#)) **Azure subscription 1**

Subscription ID 0a79542e-32e8-4080-9bc5-b26cea6f9e58

Tags ([edit](#)) [Add tags](#)

Workspace Name LAW-honeypot1

Workspace ID eb424535-cb35-4c59-981b-8af59a612dce

Pricing tier Pay-as-you-go

Access control mode Use resource or workspace permissions

Operational issues **OK**

JSON View

Legacy agents management

Legacy activity log connector

Legacy storage account logs

Legacy computer groups

Legacy solutions

System center

Workspace summary (deprecated)

Virtual machines (deprecated)

Scope configurations (deprecated)

Get Started Recommendations

- Connection status displayed

Refresh

Filter by name... 8 selected 2 selected Azure subscription 1 honeypotlab Australia E...

Name	Log Analytics Connect...	OS	Subscription	Resource group	Location
honeypot-vm	Not connected	Windows	0a79542e-32e8-4080-9bc5-b26cea6f9e58	HoneypotLab	australiaeast

Connections may take a moment to finalise, but once connected, the ‘Connect’ option will grey out, and the option to disconnect the virtual machine will appear.

The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with 'Home >'. Below it, the title 'honeypot-vm' is displayed, followed by a 'Virtual machine' label. A horizontal toolbar contains three buttons: 'Connect' (with a gear icon), 'Disconnect' (with a gear icon), and 'Refresh' (with a circular arrow icon). Below the toolbar, there are several sections: 'Status' (with a link to 'This workspace'), 'Workspace Name' (showing 'LAW-honeypot1'), and 'Message' (with a link). The background of the main content area is white.

## 3.0 Integrating Sentinel to the Workspace

*Microsoft Sentinel* is the SIEM tool that will log all event data, as well as generate real-time alerts of activity relating to the honeypot VM. Integrating a cloud-based solution for a SIEM tool allows for greater ease of access, with its advanced capabilities in automated threat detection and alerting allow for immediate identification of suspicious activity within the honeypot environment.

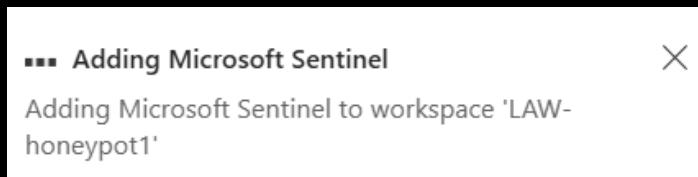
### 3.1 Navigating to Microsoft Sentinel

The screenshot shows the Microsoft Azure portal with the 'Microsoft Sentinel' service selected. The top navigation bar includes 'Copilot', a search bar, and user information ('Yasif Farook@student.ut... UTS STUDENTUTSEDUONMICR...'). Below the navigation, there's a toolbar with 'Create', 'Manage view', 'Refresh', 'Export to CSV', 'Open query', and 'View incidents'. A filter bar allows for filtering by 'Subscription equals all', 'Resource group equals all', 'Location equals all', and 'Add filter'. The main area displays a message 'Showing 0 to 0 of 0 records.' with sorting options for 'Name', 'Resource group', 'Location', 'Subscription', and 'Directory'. The background is light blue.

### 3.2 Adding Sentinel to the *LAW-honeypot1* Workspace

The screenshot shows the 'Add Microsoft Sentinel to a workspace' page. The top navigation bar includes 'Home > Microsoft Sentinel >'. Below it, there's a button to 'Create a new workspace' and a 'Refresh' button. A note states 'Microsoft Sentinel offers a 31-day free trial. See [Microsoft Sentinel pricing](#) for more details.' A filter bar at the top allows for 'Filter by name...'. The main table lists workspaces with columns: 'Workspace' (containing 'LAW-honeypot1'), 'Location' (containing 'australiaeast'), 'ResourceGroup' (containing 'honeypotlab'), 'Subscription' (containing 'Azure subscription 1'), and 'Directory' (containing 'UTS'). The row for 'LAW-honeypot1' has a red underline underneath it. The background is white.

The following message should appear once the system begins integrating Sentinel:



This step significantly enhances the security monitoring capability by leveraging Sentinel's detection rules and AI-driven insights.

## 4.0 Starting the Virtual Machine

### 4.1 Navigating to Virtual Machines in Azure

Before opening up the virtual machine, use the local host to navigate to the list of machines created on Azure.

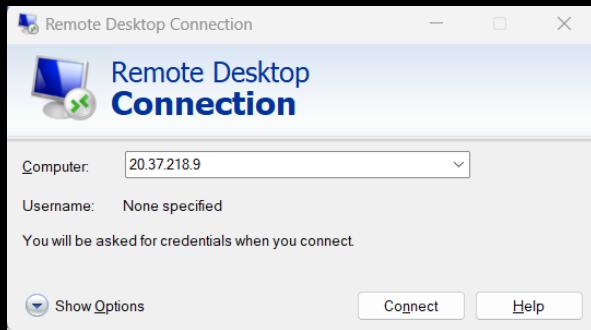
A screenshot of the Azure portal's "Virtual machines" page. The table shows one record: honeypot-vm. The details are: Name: honeypot-vm, Subscription: Azure subscription 1, Resource group: HoneypotLab, Location: Australia East, Status: Running, Operating system: Windows, Public IP address: 20.37.218.9, Size: Standard\_D2s\_v3.

Using the details listed on [honeypot-vm](#), a connection to the VM can be established. The public IP address below is listed as 20.37.218.9. This is the address that will be used to connect to the VM.

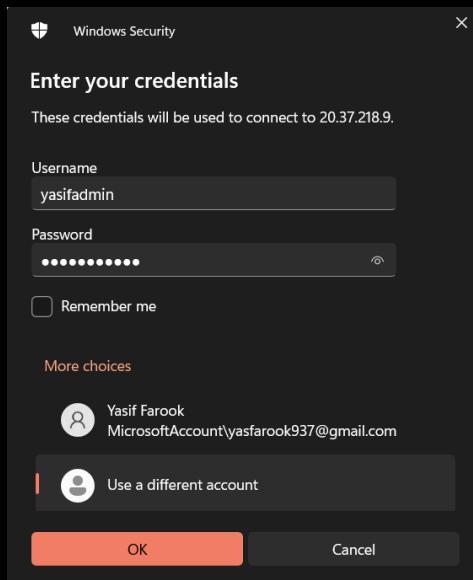
A screenshot of the Azure portal showing the details of the "honeypot-vm". In the "Essentials" section, the "Public IP address" field contains "20.37.218.9", which is underlined in red.

## 4.2 Connecting to the Virtual Machine

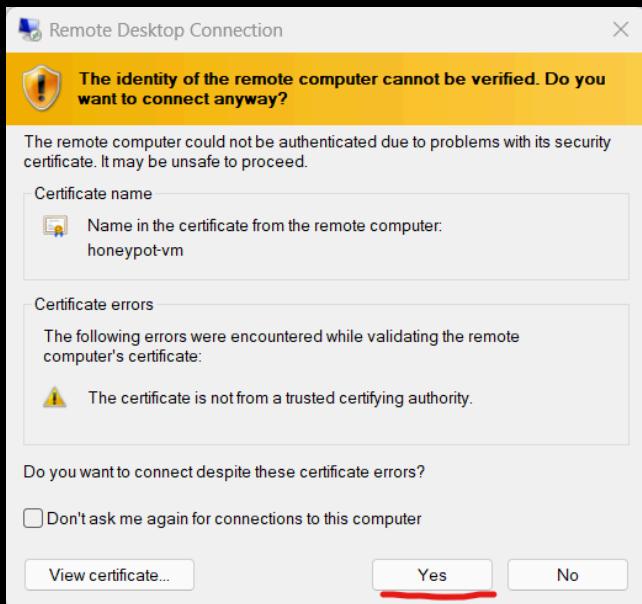
1. Open Remote Desktop Connection on local device, and connect using the Public IP address of the virtual machine:



2. Login using Azure credentials (same credentials used to create the VM and workspace):

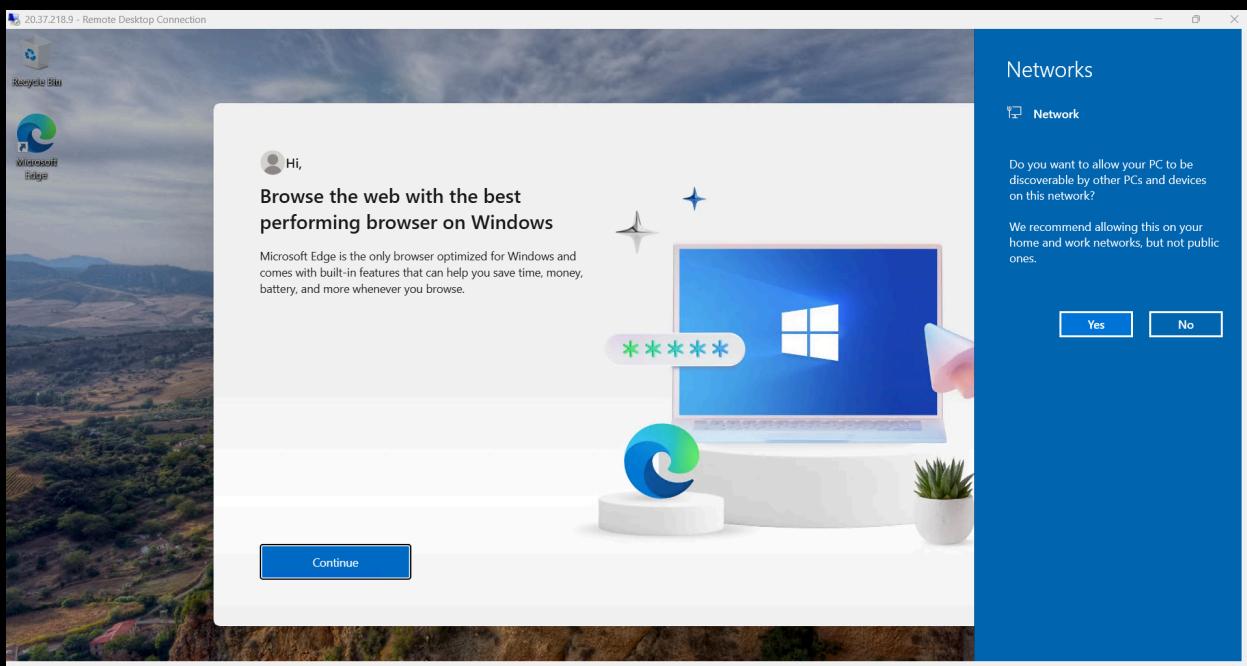


3. Accept Certificate to continue connecting:



4. Successfully connected:

Inside the Virtual Machine (IP Address displayed on the top left of the screen):

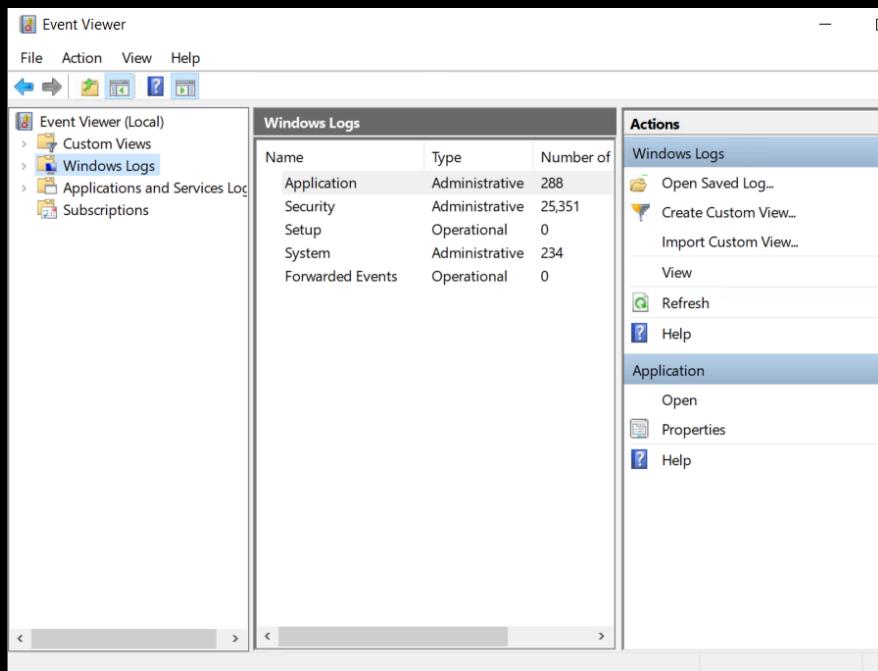


## 5.0 Monitoring Inside the VM

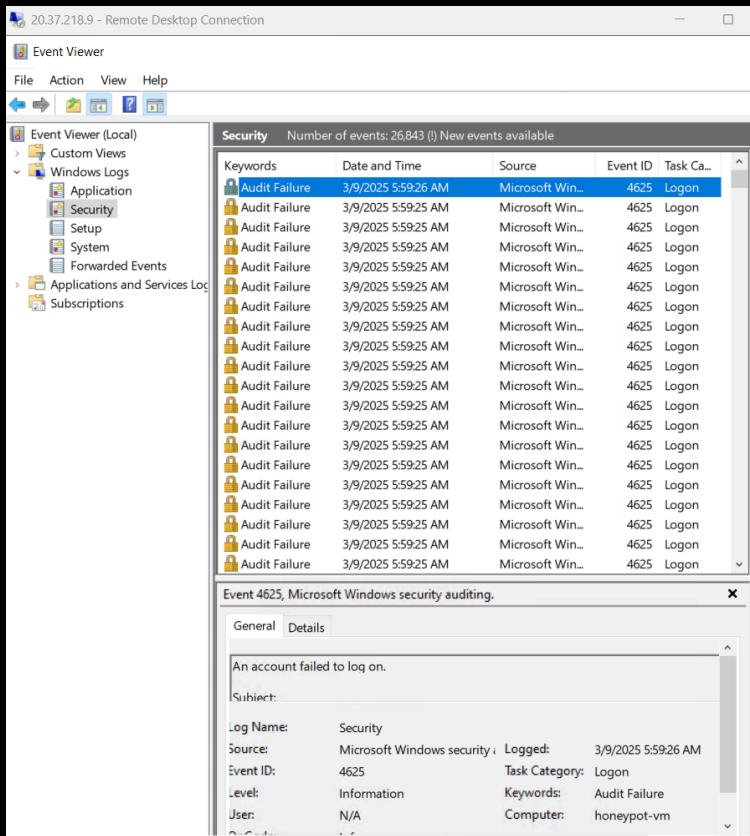
### 5.1 EventViewer

Inside the Virtual Machine for 20.37.218.9, open EventViewer.

EventViewer inside the VM:



Check security events on this VM via navigating to Windows Logs > Security:



### 5.1.1 Monitoring Security Events Inside EventViewer

- Inside Security Event ID 4625, occurring at 3/9/2025 5:59:26 AM:



The screenshot shows a Windows Event Viewer log entry. At the top, under 'Network Information', it lists: Workstation Name: -, Source Network Address: 185.243.96.107, and Source Port: 0. Below that, under 'Detailed Authentication Information', it shows Logon Process: NtLmSsp. A detailed description follows:

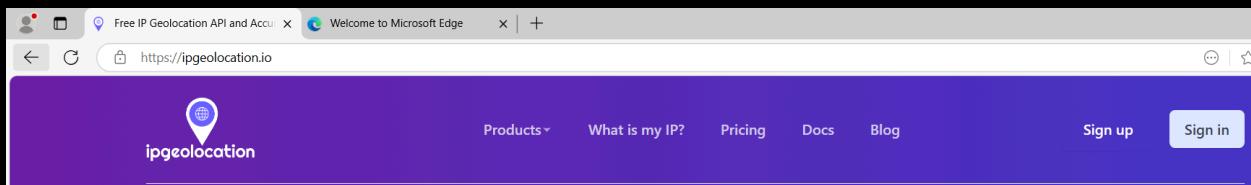
This event is generated when a logon request fails. It is generated on the computer where access was attempted.

The Subject fields indicate the account on the local system which requested the logon. This is most commonly a service such as the Server service, or a local process such as Winlogon.exe or Services.exe.

The alert suggests a failed login attempt. This is confirmed by tagging the event with the keyword "Audit Failure", and categorising the task as an attempted logon.

### 5.1.2 Integrating Geolocation API

More investigations behind the failed login attempt can be made. An [IP Geolocation API](#), can use the IP address behind the attempt, to reveal geological details such as country, state/province, city as well as coordinates. For example, for the user SANKHYA, from the failed logon above.



- Results returned inside the API:

The screenshot shows the search results for the IP address 185.243.96.107. The results are displayed as a JSON object:

```

{
  "ip": "185.243.96.107",
  "hostname": "185.243.96.107",
  "continent_code": "EU",
  "continent_name": "Europe",
  "country_code2": "UA",
  "country_code3": "UKR",
  "country_name": "Ukraine",
  "country_name_official": "Ukraine",
  "country_capital": "Kyiv",
  ...
}

```

Below the results, there are four buttons: Your IP, 49.12.212.42, 146.70.238.190, and Youtube.com.

```

185.243.96.107 Q

{
  "state_prov": "Dnipropetrovsk",
  "state_code": "UA-12",
  "district": "Novomoskovsk",
  "city": "Novomoskovsk",
  "zipcode": "51203",
  "latitude": "48.60888",
  "longitude": "35.17154",
  "is_eu": false,
  "calling_code": "+380",
  "country_tld": ".ua"
}

Your IP 49.12.212.42 146.70.238.190 Youtube.com

```

```

185.243.96.107 Q

{
  "isp": "Rices Privately owned enterprise",
  "connection_type": "",
  "organization": "Rices Privately owned enterprise",
  "asn": "AS48693",
  "geoname_id": "9532965",
  "country_emoji": "ua",
  "currency": Object {
    "name": "Hryvnia",
    "code": "UAH",
  }
}

Your IP 49.12.212.42 146.70.238.190 Youtube.com

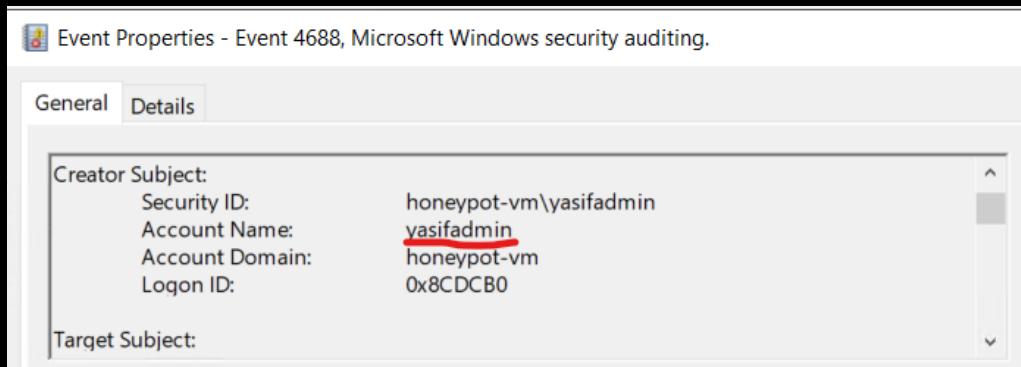
```

### 5.1.3 Monitoring More Events Inside EventViewer

- More details on Event ID 4688, occurring at 3/9/2025 5:59:02 AM.

Security Number of events: 26,843 (I) New events available					
Keywords	Date and Time	Source	Event ID	Task Category	
Audit Failure		Microsoft Windows...	4625	Logon	
Audit Success	3/9/2025 5:59:02 AM	Microsoft Windows...	4688	Process Creation	
Audit Failure		Microsoft Windows...	4625	Logon	
Audit Failure		Microsoft Windows...	4625	Logon	

Alert provides contextual information behind the process, that verifies the user behind creating the process (myself):



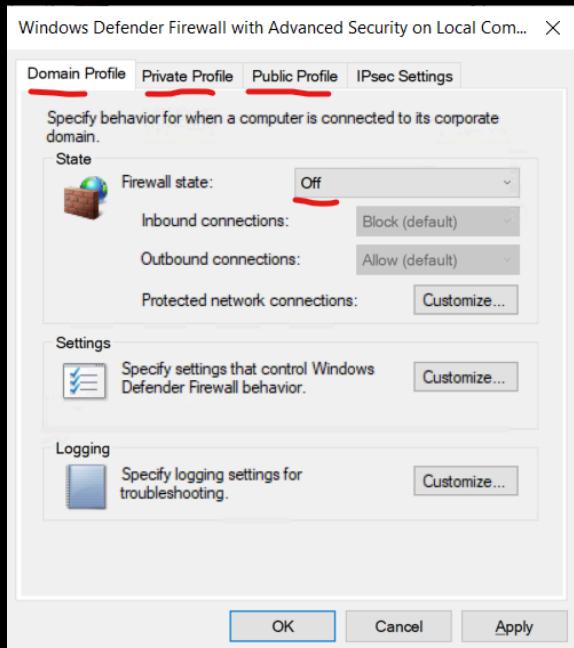
## 5.2 Configuring the VM Firewall

When attempting to ping the virtual machine from the local device, requests time out (due to the VM's firewall settings)

```
Command Prompt - ping 20.37.218.9 + - Microsoft Windows [Version 10.0.26100.3323]
(c) Microsoft Corporation. All rights reserved.

C:\Users\yasfa>ping 20.37.218.9 -t
Pinging 20.37.218.9 with 32 bytes of data:
Request timed out.
Request timed out.
```

Configure the firewall inside the VM to negate this. Start by entering 'wf.msc' on the start menu of the VM. On Firewall properties, turn the Firewall state to Off on Domain, Private and Public Profiles.



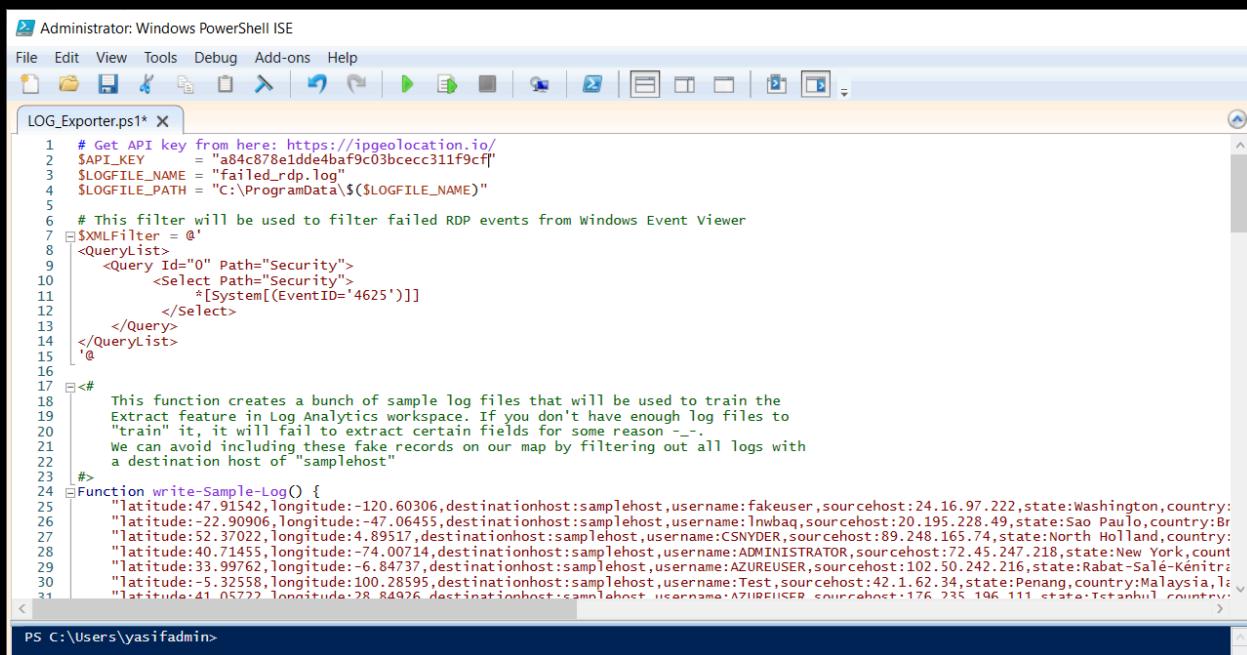
Pings are now successful:

```
Command Prompt - ping 20.37.218.9 + ^
```

```
Request timed out.  
Reply from 20.37.218.9: bytes=32 time=10ms TTL=113  
Reply from 20.37.218.9: bytes=32 time=10ms TTL=113  
Reply from 20.37.218.9: bytes=32 time=9ms TTL=113  
Reply from 20.37.218.9: bytes=32 time=13ms TTL=113  
Reply from 20.37.218.9: bytes=32 time=10ms TTL=113  
Reply from 20.37.218.9: bytes=32 time=8ms TTL=113  
Reply from 20.37.218.9: bytes=32 time=9ms TTL=113  
Reply from 20.37.218.9: bytes=32 time=11ms TTL=113
```

## 5.3 Generating Logs Inside the VM

Generate logs using open-source PowerShell Script.

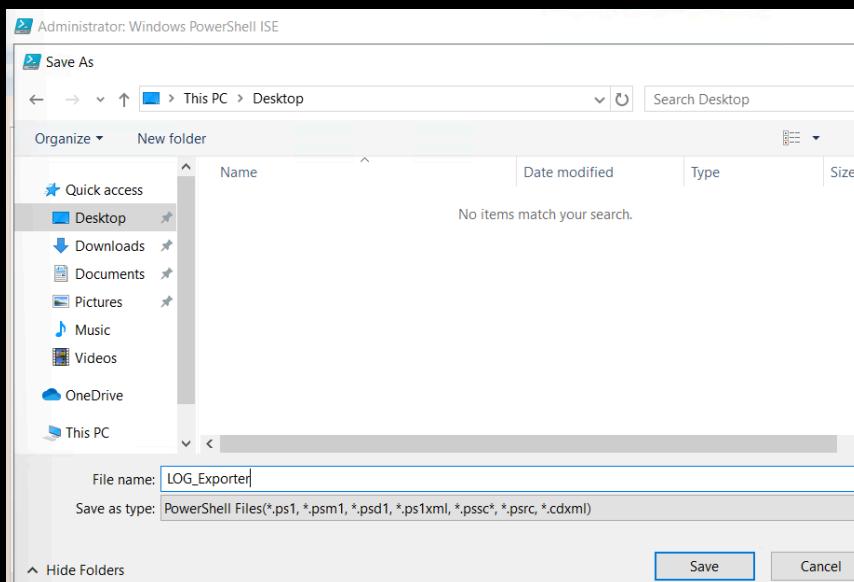


The screenshot shows the Windows PowerShell ISE interface with a script file named 'LOG\_Exporter.ps1' open. The code in the script is as follows:

```
1 # Get API key from here: https://ipgeolocation.io/
2 $API_KEY = "a84c878e1dde4ba9c03bcecc311f9cf"
3 $LOGFILE_NAME = "failed_rdp.log"
4 $LOGFILE_PATH = "C:\ProgramData\$($LOGFILE_NAME)"
5
6 # This filter will be used to filter failed RDP events from Windows Event Viewer
7 $XMLFilter = @'
8 <QueryList>
9   <Query Id="0" Path="Security">
10     <Select Path="Security">
11       *[System[(EventID='4625')]]
12     </Select>
13   </Query>
14 </QueryList>
15 '>
16
17 <#
18 This function creates a bunch of sample log files that will be used to train the
19 Extract feature in Log Analytics workspace. If you don't have enough log files to
20 "train" it, it will fail to extract certain fields for some reason --.
21 We can avoid including these fake records on our map by filtering out all logs with
22 a destination host of "samplehost"
23 #>
24 Function write-Sample-Log() {
25   "latitude:47.91542,longitude:-120.60306,destinationhost:samplehost,username:fakeuser,sourcehost:24.16.97.222,state:Washington,country:
26   "latitude:-22.90906,longitude:-47.06455,destinationhost:samplehost,username:lnwbaq,sourcehost:20.195.228.49,state:Sao Paulo,country:Br
27   "latitude:52.37022,longitude:4.89517,destinationhost:samplehost,username:CSNYDER,sourcehost:89.248.165.74,state:North Holland,country:
28   "latitude:40.71455,longitude:-74.00714,destinationhost:samplehost,username:ADMINISTRATOR,sourcehost:72.45.247.218,state:New York,countr
29   "latitude:33.99762,longitude:-6.84737,destinationhost:samplehost,username:AZUREUSER,sourcehost:102.50.242.216,state:Rabat-Sale-Kenitra
30   "latitude:-5.32558,longitude:100.28595,destinationhost:samplehost,username:Test,sourcehost:42.1.62.34,state:Penang,country:Malaysia,la
31   "latitude:-41.05722,longitude:-78.84926,destinationhost:samplehost,username:AZUREUSER,sourcehost:176.235.196.111,state:Tetanbul,country:
```

PS C:\Users\yasifadmin>

- Saving script:



The purpose of the script is to run a loop that initiates continuous event monitoring. This relates to any activity regarding the *honeypot-vm*, such as failed/successful login attempts, as well as connecting the VM to the Azure Cloud (as done in this project)

```

LOG_Exporter.ps1* X
1 # Get API key from here: https://ipgeolocation.io/
2 $API_KEY      = "a84c878e1dde4baF9c03bc ecc311f9cf"
3 $LOGFILE_NAME = "failed_rdp.log"
4 $LOGFILE_PATH = "C:\ProgramData\$($LOGFILE_NAME)"
5
6 # This filter will be used to filter failed RDP events from Windows Event Viewer
7 $XMLFilter = ...
8
9 <#...#>
10 Function write-Sample-Log() {...}
11
12 # This block of code will create the log file if it doesn't already exist
13 if ((Test-Path $LOGFILE_PATH) -eq $false) {...}
14
15 # Infinite Loop that keeps checking the Event Viewer logs.
16 while ($true)
17 {
18 }

```

### 5.3.1 Running Script

- Running the script, waiting for activity to be recorded and logged:

```

Directory: C:\ProgramData

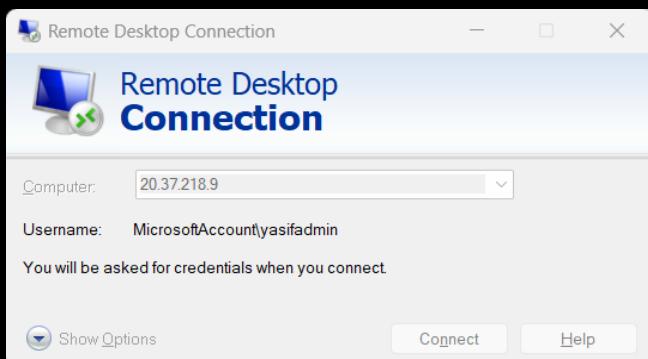
Mode                LastWriteTime         Length Name
----                -----              ---- -
-a---    3/9/2025   7:05 AM           0 failed_rdp.log

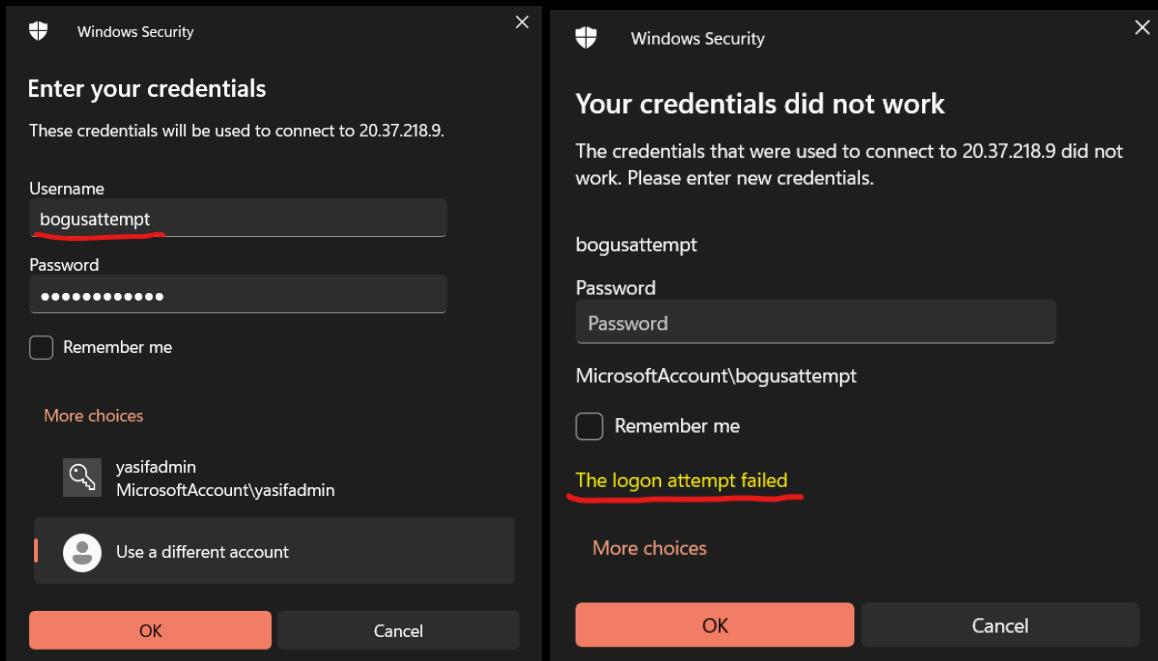
PS C:\Users\yasifadmin> C:\Users\yasifadmin\Desktop\LOG_Exporter.ps1

```

### 5.3.2 Verifying Event Monitoring

To verify activity is being correctly logged and script is running, attempt a failed login to the VM, using a bogus username and password from another instance of Remote Desktop Connection. This again, can be launched from the local host. See below for confirmation that the attempt was correctly monitored and logged.





### 5.3.3 Preparing the Log File

The Powershell script aims to generate a recurring *.log* file that prints out all activity logged on EventViewer, and the Powershell terminal.

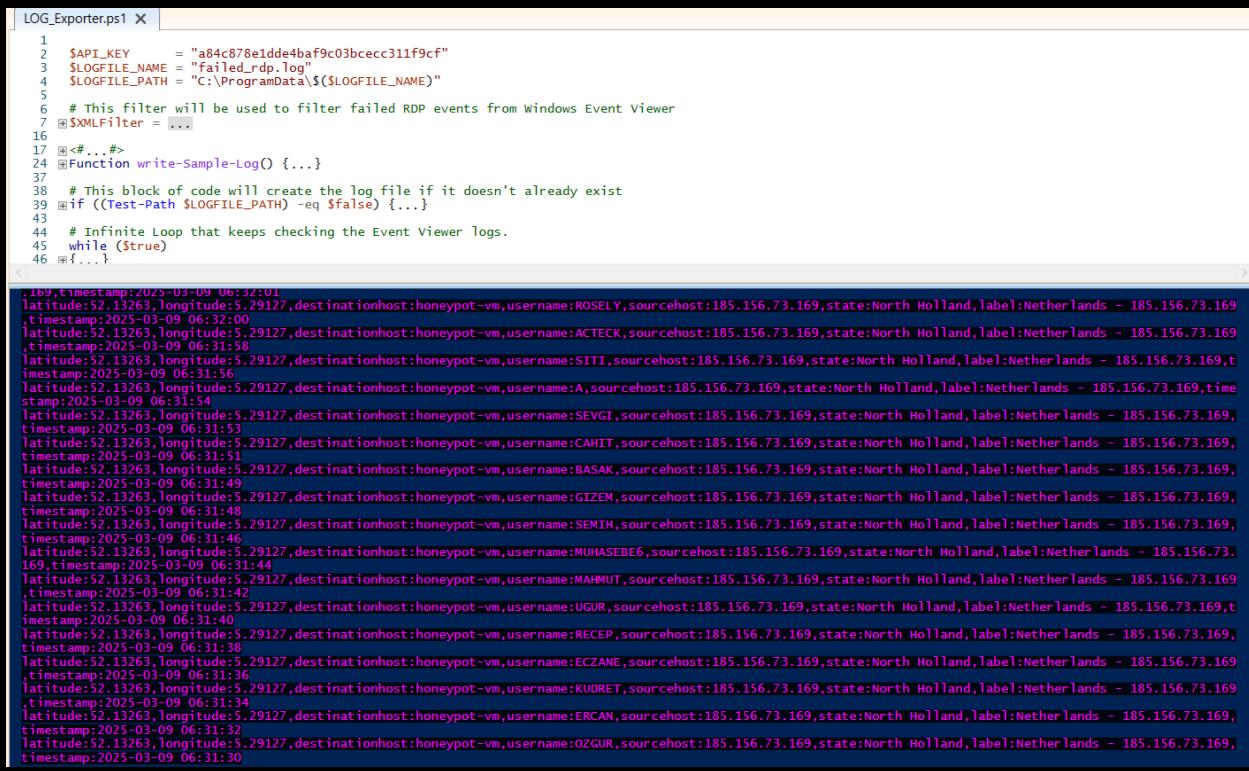
*Failed\_rdp.log* file created upon running the script. This file keeps track of all event activity regarding the VM:

	Name	Date modified	Type	Size
ss	Microsoft	3/9/2025 5:58 AM	File folder	
ds	Microsoft OneDrive	1/31/2025 1:05 PM	File folder	
nts	Packages	3/9/2025 6:11 AM	File folder	
	regid.1991-06.com.microsoft	3/9/2025 6:26 AM	File folder	
	SoftwareDistribution	12/7/2019 9:14 AM	File folder	
	ssh	1/31/2025 12:22 PM	File folder	
	USOPrivate	1/31/2025 12:59 PM	File folder	
	USOShared	12/7/2019 9:14 AM	File folder	
	WindowsHolographicDevices	12/7/2019 9:52 AM	File folder	
	failed_rdp	3/9/2025 7:13 AM	Text Document	9 KB
	ntuser.pol	3/9/2025 5:02 AM	POL File	4 KB

Upon inspecting the file, the intentional failed login attempt using bogus credentials can be verified:

```
latitude:52.37022,longitude:4.89517,destinationhost:honeypot-vm,username:CHUCK,sourcehost:169.245.98.167,state:  
latitude:52.37022,longitude:4.89517,destinationhost:honeypot-vm,username:bogusattempt,sourcehost:61.68.142.168,  
latitude:52.37022,longitude:4.89517,destinationhost:honeypot-vm,username:bogusattempt,sourcehost:61.68.142.168
```

Login attempts have appeared on the terminal:



The screenshot shows a terminal window with two main sections. The top section contains the content of a PowerShell script named LOG\_Exporter.ps1. The bottom section shows the execution of the script, displaying a list of failed RDP login attempts from the Windows Event Viewer log.

```
LOG_Exporter.ps1 X  
1 $API_KEY      = "a84c878e1dde4baf9c03bcecc311f9cf"  
2 $LOGFILE_NAME = "Failed_rdp.Log"  
3 $LOGFILE_PATH = "C:\ProgramData\$($LOGFILE_NAME)"  
4  
5 # This filter will be used to filter failed RDP events from Windows Event Viewer  
6 $XMLFilter = ...  
16  
17 <#...#>  
24 Function write-Sample-Log() {...}  
37  
38 # This block of code will create the log file if it doesn't already exist  
39 if ((Test-Path $LOGFILE_PATH) -eq $false) {...}  
43  
44 # Infinite Loop that keeps checking the Event Viewer logs.  
45 while ($true)  
46 { ... }  
  
.169,timestamp:2025-03-09 06:32:01  
latitude:52.13263,longitude:5.29127,destinationhost:honeypot-vm,username:ROSELY,sourcehost:185.156.73.169,state:North Holland,label:Netherlands - 185.156.73.169  
.timestamp:2025-03-09 06:32:00  
latitude:52.13263,longitude:5.29127,destinationhost:honeypot-vm,username:ACTECK,sourcehost:185.156.73.169,state:North Holland,label:Netherlands - 185.156.73.169  
.timestamp:2025-03-09 06:31:58  
latitude:52.13263,longitude:5.29127,destinationhost:honeypot-vm,username:SITI,sourcehost:185.156.73.169,state:North Holland,label:Netherlands - 185.156.73.169,t  
imestamp:2025-03-09 06:31:56  
latitude:52.13263,longitude:5.29127,destinationhost:honeypot-vm,username:A,sourcehost:185.156.73.169,state:North Holland,label:Netherlands - 185.156.73.169,t  
imestamp:2025-03-09 06:31:54  
latitude:52.13263,longitude:5.29127,destinationhost:honeypot-vm,username:SEVGI,sourcehost:185.156.73.169,state:North Holland,label:Netherlands - 185.156.73.169,  
t  
imestamp:2025-03-09 06:31:53  
latitude:52.13263,longitude:5.29127,destinationhost:honeypot-vm,username:CAHIT,sourcehost:185.156.73.169,state:North Holland,label:Netherlands - 185.156.73.169,  
t  
imestamp:2025-03-09 06:31:51  
latitude:52.13263,longitude:5.29127,destinationhost:honeypot-vm,username:BASAK,sourcehost:185.156.73.169,state:North Holland,label:Netherlands - 185.156.73.169,  
t  
imestamp:2025-03-09 06:31:49  
latitude:52.13263,longitude:5.29127,destinationhost:honeypot-vm,username:GIZEM,sourcehost:185.156.73.169,state:North Holland,label:Netherlands - 185.156.73.169,  
t  
imestamp:2025-03-09 06:31:48  
latitude:52.13263,longitude:5.29127,destinationhost:honeypot-vm,username:SEMIH,sourcehost:185.156.73.169,state:North Holland,label:Netherlands - 185.156.73.169,  
t  
imestamp:2025-03-09 06:31:46  
latitude:52.13263,longitude:5.29127,destinationhost:honeypot-vm,username:MIHASEREE,sourcehost:185.156.73.169,state:North Holland,label:Netherlands - 185.156.73.169  
.169,timestamp:2025-03-09 06:31:44  
latitude:52.13263,longitude:5.29127,destinationhost:honeypot-vm,username:MAHMUT,sourcehost:185.156.73.169,state:North Holland,label:Netherlands - 185.156.73.169  
.timestamp:2025-03-09 06:31:42  
latitude:52.13263,longitude:5.29127,destinationhost:honeypot-vm,username:UGUR,sourcehost:185.156.73.169,state:North Holland,label:Netherlands - 185.156.73.169,t  
imestamp:2025-03-09 06:31:40  
latitude:52.13263,longitude:5.29127,destinationhost:honeypot-vm,username:REcep,sourcehost:185.156.73.169,state:North Holland,label:Netherlands - 185.156.73.169,  
t  
imestamp:2025-03-09 06:31:38  
latitude:52.13263,longitude:5.29127,destinationhost:honeypot-vm,username:ECZANE,sourcehost:185.156.73.169,state:North Holland,label:Netherlands - 185.156.73.169  
.timestamp:2025-03-09 06:31:36  
latitude:52.13263,longitude:5.29127,destinationhost:honeypot-vm,username:KUDRET,sourcehost:185.156.73.169,state:North Holland,label:Netherlands - 185.156.73.169  
.timestamp:2025-03-09 06:31:34  
latitude:52.13263,longitude:5.29127,destinationhost:honeypot-vm,username:ERCAN,sourcehost:185.156.73.169,state:North Holland,label:Netherlands - 185.156.73.169,  
t  
imestamp:2025-03-09 06:31:32  
latitude:52.13263,longitude:5.29127,destinationhost:honeypot-vm,username:OZGUR,sourcehost:185.156.73.169,state:North Holland,label:Netherlands - 185.156.73.169,  
t  
imestamp:2025-03-09 06:31:30
```

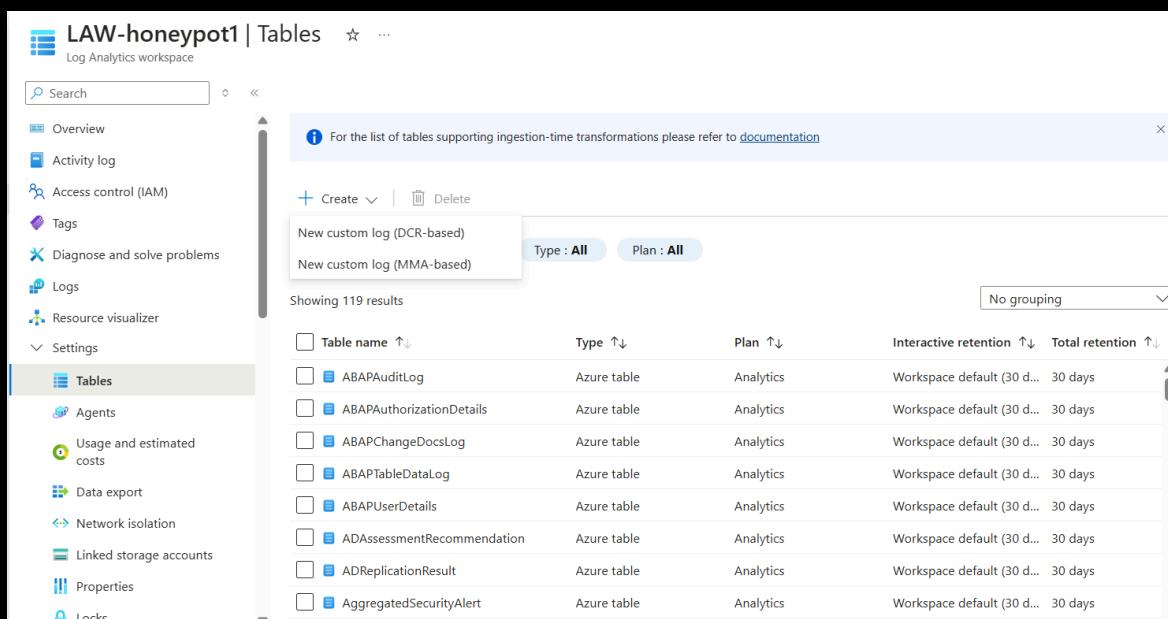
## 6.0 Upgrading Logs

With the use of Azure, the logs generated by the Powershell script can be optimised to allow for greater efficiency, as well as more in-depth analysis and viewing.

### 6.1 Creating a custom log in Azure

The following steps are completed on the local machine:

1. Navigate to **Tables** in Azure:



LAW-honeypot1 | Tables

Log Analytics workspace

Search

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Logs

Resource visualizer

Settings

Tables

Agents

Usage and estimated costs

Data export

Network isolation

Linked storage accounts

Properties

Locks

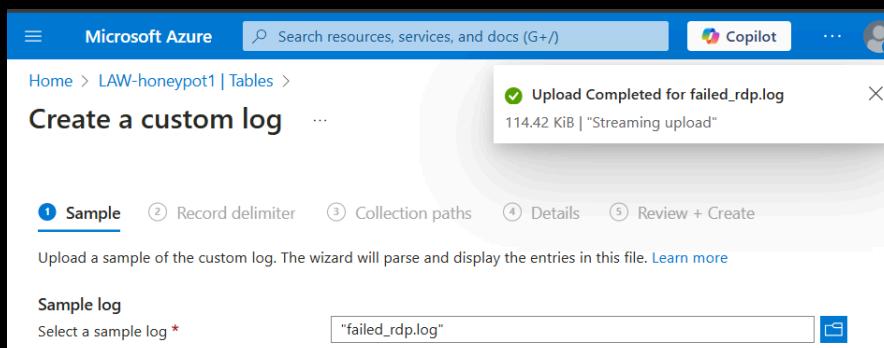
+ Create | Delete

Type : All Plan : All

Showing 119 results

Table name ↑	Type ↑↓	Plan ↑↓	Interactive retention ↑↓	Total retention ↑↓
ABAAuditLog	Azure table	Analytics	Workspace default (30 d... 30 days	30 days
ABAPAuthorizationDetails	Azure table	Analytics	Workspace default (30 d... 30 days	30 days
ABAPChangeDocsLog	Azure table	Analytics	Workspace default (30 d... 30 days	30 days
ABAPTableDataLog	Azure table	Analytics	Workspace default (30 d... 30 days	30 days
ABAPUserDetails	Azure table	Analytics	Workspace default (30 d... 30 days	30 days
ADAessmentRecommendation	Azure table	Analytics	Workspace default (30 d... 30 days	30 days
ADReplicationResult	Azure table	Analytics	Workspace default (30 d... 30 days	30 days
AggregatedSecurityAlert	Azure table	Analytics	Workspace default (30 d... 30 days	30 days

2. Navigate to **Create a custom log**:



Microsoft Azure

Search resources, services, and docs (G+/-)

Copilot

Home > LAW-honeypot1 | Tables >

Create a custom log

Upload Completed for failed\_rdp.log

114.42 KiB | "Streaming upload"

Sample

Record delimiter

Collection paths

Details

Review + Create

Upload a sample of the custom log. The wizard will parse and display the entries in this file. [Learn more](#)

Sample log

Select a sample log \*

"failed\_rdp.log"

3. Set the correct path to the log file that the Powershell script generates. This path can be found on the file library, back in the VM.

Home > LAW-honeypot1 | Tables >

## Create a custom log

... >

✓ Sample ✓ Record delimiter ③ Collection paths ④ Details ⑤ Review + Create

Define one or more paths on the agent where it can locate the custom log. [Learn more](#)

**Collection paths**

Type	Path
Windows	C:\ProgramData\failed_rdp.log
Select type	

Microsoft Azure Search resources, services, and docs (G+/) Copilot

Home > LAW-honeypot1 | Tables >

## Create a custom log

... >

✓ Sample ✓ Record delimiter ✓ Collection paths ✓ Details ⑤ Review + Create

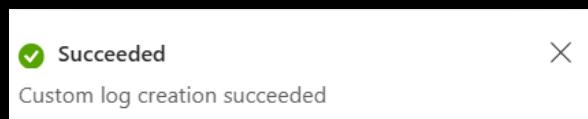
**Sample**  
Sample log failed\_rdp.log

**Record delimiter**  
Record delimiter New line

**Collection paths**  
Windows C:\ProgramData\failed\_rdp.log

**Details**  
Custom log name FAILED\_RDP\_WITH\_GEO\_CL  
Description

« Previous Create



Custom logs can be found in the **Logs** section:

LAW-honeypot1 | Logs

New Query 1\*

Search

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Logs

Resource visualizer

Settings

Classic

1 FAILED\_...

FailedIngestion

StorageMoveCopyLogsFailed

\_ASim\_FileEvent\_AzureFileStorageV01([disabled])

\_ASim\_WebSession\_F5ASMV01([disabled])

\_ASim\_FileEvent\_AzureTableStorageV01([disabled])

\_ASim\_FileEvent\_NativeV01([disabled])

\_ASim\_Dns\_AzureFirewallV03([disabled])

\_ASim\_FileEvent\_AzureBlobStorageV01([disabled])

\_ASim\_FileEvent\_AzureQueueStorageV01([disabled])

\_ASim\_FileEvent\_Microsoft365DV02([disabled])

\_ASim\_Dns\_FortinetFortigateV01([disabled])

Query history

#### 4. Run the selected log:

New Query 1\*

LAW-honeypot1

Select scope

Run

Time range : Last 24 hours

Try the new Log An... Feedback Queries hub

Save Share New alert rule Export Pin to

1 FAILED\_RDP\_WITH\_GEO\_CL

2

Results

TimeGenerated [UTC]	Computer	RawData	Type	_ResourceId
> 3/9/2025, 12:42:40.122 PM	honeypot-vm	latitude:52.37022,longitude:4.89517,destinationhost:honeyp...	FAILED_RDP_WITH_GEO_CL	/subscriptions/0a79542e-32e8-4080-9bc5-b26cea6f9e58
> 3/9/2025, 8:20:41.872 AM	honeypot-vm	latitude:52.37022,longitude:4.89517,destinationhost:honeyp...	FAILED_RDP_WITH_GEO_CL	/subscriptions/0a79542e-32e8-4080-9bc5-b26cea6f9e58
> 3/9/2025, 8:20:41.872 AM	honeypot-vm	latitude:52.37022,longitude:4.89517,destinationhost:honeyp...	FAILED_RDP_WITH_GEO_CL	/subscriptions/0a79542e-32e8-4080-9bc5-b26cea6f9e58
> 3/9/2025, 8:20:41.872 AM	honeypot-vm	latitude:52.37022,longitude:4.89517,destinationhost:honeyp...	FAILED_RDP_WITH_GEO_CL	/subscriptions/0a79542e-32e8-4080-9bc5-b26cea6f9e58
> 3/9/2025, 8:20:41.872 AM	honeypot-vm	latitude:52.37022,longitude:4.89517,destinationhost:honeyp...	FAILED_RDP_WITH_GEO_CL	/subscriptions/0a79542e-32e8-4080-9bc5-b26cea6f9e58
> 3/9/2025, 8:20:41.872 AM	honeypot-vm	latitude:52.37022,longitude:4.89517,destinationhost:honeyp...	FAILED_RDP_WITH_GEO_CL	/subscriptions/0a79542e-32e8-4080-9bc5-b26cea6f9e58
> 3/9/2025, 8:20:41.872 AM	honeypot-vm	latitude:52.37022,longitude:4.89517,destinationhost:honeyp...	FAILED_RDP_WITH_GEO_CL	/subscriptions/0a79542e-32e8-4080-9bc5-b26cea6f9e58

### 6.1.1 Testing Other Queries

- Run **SecurityEvent**. This brings the Windows Event Viewer into the Log Analytics Workspace.

The screenshot shows the Log Analytics workspace interface. The top navigation bar includes 'New Query 1\*', 'Select scope', 'Run' (button), 'Time range: Last 24 hours', 'Save', 'Share', 'New alert rule', 'Export', 'Pin to', and more. The main area displays a query editor with the following code:

```
1 SecurityEvent
```

The results pane shows the first 30,000 results of the query. The schema and filter sidebar on the left lists columns: TimeGenerated [UTC], Account, AccountType, Computer, EventSourceName, and Channel. The results table contains several rows of data, such as:

TimeGenerated [UTC]	Account	AccountType	Computer	EventSourceName	Channel
3/9/2025, 1:10:19.712 PM	WORKGROUP\honeypot-vm\$	Machine	honeypot-vm	Microsoft-Windows-Security-A...	Security
3/9/2025, 1:10:19.712 PM	WORKGROUP\honeypot-vm\$	Machine	honeypot-vm	Microsoft-Windows-Security-A...	Security
3/9/2025, 1:10:19.712 PM			honeypot-vm	Microsoft-Windows-Security-A...	Security
3/9/2025, 1:10:19.711 PM			honeypot-vm	Microsoft-Windows-Security-A...	Security

- Reviewing all failed RDP logs (events with EventId 4625):

The screenshot shows the Log Analytics workspace interface. The top navigation bar includes 'New Query 1\*', 'Select scope', 'Run' (button), 'Time range: Last 24 hours', 'Save', 'Share', 'New alert rule', 'Export', 'Pin to', and more. The main area displays a query editor with the following code:

```
1 SecurityEvent | where EventID == 4625
```

The results pane shows the first 30,000 results of the query. The schema and filter sidebar on the left lists columns: TimeGenerated [UTC], Account, AccountType, Computer, EventSourceName, and Channel. The results table contains several rows of data, such as:

TimeGenerated [UTC]	Account	AccountType	Computer	EventSourceName	Channel
3/9/2025, 1:05:16.425 PM	honeypot-vm\yasifadmin	User	honeypot-vm	Microsoft-Windows-Security-A...	Security
3/9/2025, 1:04:44.996 PM	honeypot-vm\yasifadmin	User	honeypot-vm	Microsoft-Windows-Security-A...	Security
3/9/2025, 1:04:39.907 PM	honeypot-vm\yasifadmin	User	honeypot-vm	Microsoft-Windows-Security-A...	Security
3/9/2025, 6:02:00.697 AM	\KIRAN	User	honeypot-vm	Microsoft-Windows-Security-A...	Security
3/9/2025, 6:02:00.680 AM	\OFFICEUSER	User	honeypot-vm	Microsoft-Windows-Security-A...	Security
3/9/2025, 6:02:00.643 AM	\TRANSDATA	User	honeypot-vm	Microsoft-Windows-Security-A...	Security
3/9/2025, 6:02:00.569 AM	\KAR	User	honeypot-vm	Microsoft-Windows-Security-A...	Security
3/9/2025, 6:02:00.553 AM	\SOLOMON	User	honeypot-vm	Microsoft-Windows-Security-A...	Security

The screenshot shows the Microsoft Azure Log Analytics interface. A query is running in a workspace named "LAW-honeypot1". The query is:

```
1 SecurityEvent | where EventID == 4625
2
```

The results table displays the following data:

EventID	Activity	AuthenticationPackageName	FailureReason	IpAddress	IpPort
4625	4625 - An account failed to log ...	Negotiate	%2313	61.68.142.168	0
4625	4625 - An account failed to log ...	Negotiate	%2313	61.68.142.168	0
4625	4625 - An account failed to log ...	Negotiate	%2313	61.68.142.168	0
4625	4625 - An account failed to log ...	NTLM	%2313	92.63.197.9	0
4625	4625 - An account failed to log ...	NTLM	%2313	94.102.52.73	0
4625	4625 - An account failed to log ...	NTLM	%2313	185.243.96.107	0
4625	4625 - An account failed to log ...	NTLM	%2313	92.63.197.9	0
4625	4625 - An account failed to log ...	NTLM	%2313	185.243.96.107	0

The 3 most recent events (at 1:05 PM), that occurred 7 hours after the previous recorded event (at 6:02 AM), are related to logon attempts made from the admin user themselves (*honeypot-vm\yasifadmin*), who attempted to remember their password without logging into their Google Password Manager. This is further supported by the fact that the logged times align with the admin user attempting to login after not using their laptop (which contained the local host and VM) for 7 hours.

This table lists logon events ordered by TimeGenerated [UTC]. The last three entries, which occurred at 1:05 PM, are highlighted with a red box.

TimeGenerated [UTC] ↑	Account
> 3/9/2025, 1:05:16.425 PM	honeypot-vm\yasifadmin
> 3/9/2025, 1:04:44.996 PM	honeypot-vm\yasifadmin
> 3/9/2025, 1:04:39.907 PM	honeypot-vm\yasifadmin
> 3/9/2025, 6:02:00.697 AM	\KIRAN

This table lists logon events. The last three entries, which occurred at 1:05 PM, are highlighted with a red box.

TargetAccount	TargetDomainName	TargetUserName	TargetUserSid	TransmittedServices	WorkstationName
honeypot-vm\yasifadmin	honeypot-vm	yasifadmin	S-1-0-0	-	honeypot-vm
honeypot-vm\yasifadmin	honeypot-vm	yasifadmin	S-1-0-0	-	honeypot-vm
honeypot-vm\yasifadmin	honeypot-vm	yasifadmin	S-1-0-0	-	honeypot-vm
\KIRAN		KIRAN	S-1-0-0	-	-

## 6.1.2 Adding More Fields to Logs for Greater Details

The query below aims to output the log data with greater structure, with finer details such as State, Country and longitude & latitude, in separate fields. This can allow for even greater log analysis and filtering, as the previous query (based on the initial Powershell script), displayed the newly filtered data in a long string format under *EventSourceName*

```

1 FAILED_RDP_WITH_GEO_CL
2 | extend Timestamp = todatetime(TimeGenerated),
3     Latitude = tostring(extract("latitude:(\[-9.0-9.\]+)", 1, RawData)),
4     Longitude = tostring(extract("longitude:(\[-9.0-9.\]+)", 1, RawData)),
5     DestinationHost = tostring(extract("destinationhost:(\w+)", 1, RawData)),
6     Username = tostring(extract("username:(\w+)", 1, RawData)),
7     SourceHost = tostring(extract("sourcehost:(\w+)", 1, RawData)),
8     State = tostring(extract("state:(\w+\s+)", 1, RawData)),
9     Country = tostring(extract("country:(\w+\s+)", 1, RawData)),
10    Label = tostring(extract("label:(\w+\s+)", 1, RawData))
11 | project Timestamp, Username, SourceHost, DestinationHost, State, Country, Latitude, Longitude, Label

```

Timestamp [UTC]	Username	SourceHost	DestinationHost	State	Country
> 3/9/2025, 1:16:10.115 PM	yasifadmin	61.68.142.168	honeypot-vm	North Holland	Netherlands
> 3/9/2025, 1:16:10.115 PM	yasifadmin	61.68.142.168	honeypot-vm	North Holland	Netherlands
> 3/9/2025, 1:16:10.115 PM	yasifadmin	61.68.142.168	honeypot-vm	North Holland	Netherlands
> 3/9/2025, 12:42:40.122 PM	ADMINISTRATOR	102.129.138.212	honeypot-vm	North Holland	Netherlands
> 3/9/2025, 8:20:41.872 AM	bogusattempt	61.68.142.168	honeypot-vm	North Holland	Netherlands
> 3/9/2025, 8:20:41.872 AM	bogusattempt	61.68.142.168	honeypot-vm	North Holland	Netherlands

Host	State	Country	Latitude	Longitude	Label
/vm	North Holland	Netherlands	52.37022	4.89517	Netherlands - 61.68.142.168
/vm	North Holland	Netherlands	52.37022	4.89517	Netherlands - 61.68.142.168
/vm	North Holland	Netherlands	52.37022	4.89517	Netherlands - 61.68.142.168
/vm	North Holland	Netherlands	52.37022	4.89517	Netherlands - 102.129.138.212
/vm	North Holland	Netherlands	52.37022	4.89517	Netherlands - 61.68.142.168
/vm	North Holland	Netherlands	52.37022	4.89517	Netherlands - 61.68.142.168

### 6.1.3 Breakdown of Updated Fields:

Added elements to the query to extract specific information ( Latitude, Longitude, Username, etc.) from the *RawData* field, which stores everything as a long string in a key-value format.

- **extend Clause:** The extend keyword is used to create new columns from the existing data. In this case, it creates new fields by extracting values from the RawData string.
- **Timestamp:** Converted the TimeGenerated field into a readable date and time format using todatetime().
- For each field (Latitude, Longitude, etc.), used **extract()** to extract values from RawData based on specific patterns (like latitude:, longitude:, etc.).
- **extract() Function:** This function helps pull out values from the RawData string. The general format is: **extract("pattern", group\_number, RawData)**

In this structure:

**Pattern:** A regular expression (regex) that describes the structure of the data that is to be extracted.

**Group Number:** Specifies which part of the match to return. In this case, 1, as the actual value after the colon (e.g., after latitude:) should be returned.

**RawData:** The field data is being extracted from.

- **project** Clause: Once the required fields have been extracted (Latitude, Longitude, Username, etc.), use the project clause to display only the fields needed.

```
| project Timestamp, Username, SourceHost, DestinationHost, State, Country, Latitude, Longitude, Label
```

- **project**: Tells the query to display only the columns specified.
- **Timestamp**: The time of the event.
- **Username**: The username associated with the event.
- **SourceHost**: The source IP address (or hostname).
- **DestinationHost**: The target host (where the login attempt was made).
- **State and Country**: The location of the source.
- **Latitude and Longitude**: The geographical coordinates of the source.
- **Label**: A label summarising the event.

This update query with separated fields makes analysis and filtering more effective and easier. Specific fields such as *longitude* can be pulled out of the log.

**Full query:**

```
FAILED_RDP_WITH_GEO_CL
| extend Timestamp = todatetime(TimeGenerated),
    Latitude = tostring(extract("latitude:([0-9.-]+)", 1, RawData)),
    Longitude = tostring(extract("longitude:([0-9.-]+)", 1, RawData)),
    DestinationHost = tostring(extract("destinationhost:(\w.-)+", 1, RawData)),
    Username = tostring(extract("username:(\w.-)+", 1, RawData)),
    SourceHost = tostring(extract("sourcehost:(\d.+)", 1, RawData)),
    State = tostring(extract("state:(\w\s)+", 1, RawData)),
    Country = tostring(extract("country:(\w\s)+", 1, RawData)),
    Label = tostring(extract("label:(\w\s.-)+", 1, RawData))
| project Timestamp, Username, SourceHost, DestinationHost, State, Country, Latitude, Longitude, Label
```

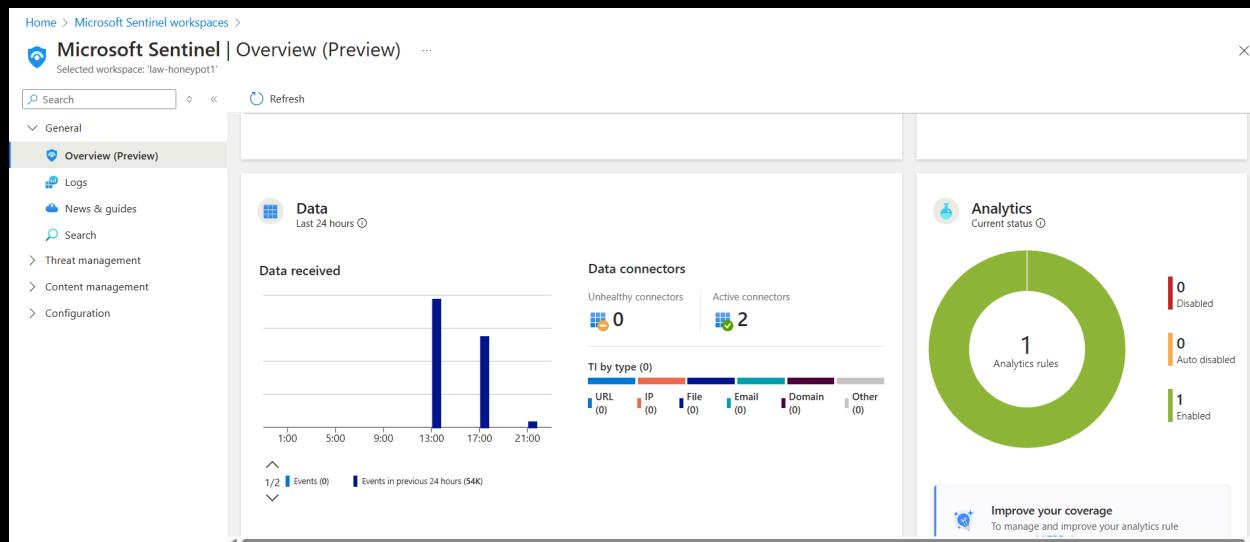
## 7.0 Visualising Attack Patterns with Azure Sentinel Map

Following on from the previous steps, a map representing log data can be generated.

This map will enable real-time monitoring of events, with geographic location-based alerts. The mapping of events can be set by either country, as well as longitude & latitude.

### 7.1 Integrating Microsoft Sentinel

Navigate to *Microsoft Sentinel* (found via the search bar):



- Adding previous detailed query to new Workbook:

The screenshot shows a 'New workbook' page in Microsoft Sentinel. The top navigation bar includes 'Done Editing', 'Open', 'Samples', 'Logs (Analytics)', 'Log Analytics', 'law-honeypot1', 'Last 24 hours', 'Set by q...', and 'Medium'. The main area is titled 'Editing query item: query - 0' and contains tabs for 'Settings', 'Advanced Settings', 'Style', and 'Advanced Editor'. The 'Run Query' tab is selected. The 'Log Analytics workspace Logs (Analytics) Query' section displays the following PowerShell-like query script:  
```  
FAILED\_RDP\_WITH\_GEO\_CL  
| extend Timestamp = todatetime(TimeGenerated),  
| Latitude = tostring(extract("latitude:([0-9.-]+)", 1, RawData)),  
Longitude = tostring(extract("longitude:([0-9.-]+)", 1, RawData)),  
DestinationHost = tostring(extract("destinationhost:([\w.-]+)", 1, RawData)),  
Username = tostring(extract("username:([\w.-]+)", 1, RawData)),  
SourceHost = tostring(extract("sourcehost:([\d.]+)", 1, RawData)),  
State = tostring(extract("state:([\w\s]+)", 1, RawData)),  
Country = tostring(extract("country:([\w\s]+)", 1, RawData)),  
Label = tostring(extract("label:([\w\s.-]+)", 1, RawData))  
| projecton Timestamp, Username, SourceHost, DestinationHost, State, Country, Latitude, Longitude, Label  
```

- Running Query:

New workbook aw-honeypot1

```
Latitude = tostring(extract("latitude:([-9-9.-]+)", 1, RawData)),
Longitude = tostring(extract("longitude:([-9-9.-]+)", 1, RawData)),
DestinationHost = tostring(extract("destinationhost:(\w+)", 1, RawData)),
Username = tostring(extract("username:(\w+)", 1, RawData)),
SourceHost = tostring(extract("sourcehost:(\w+)", 1, RawData)),
State = tostring(extract("state:(\w+)", 1, RawData)),
Country = tostring(extract("country:(\w+)", 1, RawData)),
Label = tostring(extract("label:(\w+)", 1, RawData))
| project Timestamp, Username, SourceHost, DestinationHost, State, Country, Latitude, Longitude, Label
```

Timestamp	Username	SourceHost	DestinationHost	State	Country	Latitude	Longitude	Label
3/9/2025, 7:06:55.680 PM	ACC1	185.243.96.107	honeypot-vm	North Holland	Netherlands	52.37022	4.89517	Netherlands - 185.243.96.107
3/9/2025, 7:06:55.680 PM	WIN9876	94.102.52.73	honeypot-vm	North Holland	Netherlands	52.37022	4.89517	Netherlands - 94.102.52.73
3/9/2025, 7:06:55.680 PM	JACK	92.63.197.9	honeypot-vm	North Holland	Netherlands	52.37022	4.89517	Netherlands - 92.63.197.9
3/9/2025, 7:06:55.680 PM	TEMPORAL	185.243.96.107	honeypot-vm	North Holland	Netherlands	52.37022	4.89517	Netherlands - 185.243.96.107
3/9/2025, 7:06:55.680 PM	TERMINAL03	185.243.96.107	honeypot-vm	North Holland	Netherlands	52.37022	4.89517	Netherlands - 185.243.96.107
3/9/2025, 7:06:55.680 PM	TEMP	94.102.52.73	honeypot-vm	North Holland	Netherlands	52.37022	4.89517	Netherlands - 94.102.52.73
3/9/2025, 7:06:55.680 PM	MAIL	92.63.197.9	honeypot-vm	North Holland	Netherlands	52.37022	4.89517	Netherlands - 92.63.197.9
3/9/2025, 7:06:55.680 PM	1C	94.102.52.73	honeypot-vm	North Holland	Netherlands	52.37022	4.89517	Netherlands - 94.102.52.73
3/9/2025, 7:06:55.680 PM	CARE	94.102.52.73	honeypot-vm	North Holland	Netherlands	52.37022	4.89517	Netherlands - 94.102.52.73

To view the output as a map instead of grid table, switch the **Visualisation** to **Map**.

Failed RDP World Map aw-honeypot1

1 Editing query item: query - 0

Settings Advanced Settings Style Advanced Editor

Query (change) Time Range Visualization Size  
  law-honeypot1 Last 24 hours  Medium

Log Analytics workspace Logs (Analytics) Query  
 FAILED RDP WITH GEO CL

## 7.2 Latitude & Longitude-Based Map

- Adjusting Map settings to accurately account for latitude and longitude as coordinates:

The screenshot shows the Microsoft Sentinel Workbooks interface with a world map visualization titled "Failed RDP World Map". The map displays various locations with colored dots representing event counts. Below the map, a summary table provides details for specific locations:

Latitude	Longitude	EventCount
52.37022	1.35 k	800
52.1263	3	Other
47.91542	1	1
-22.90006	1	1
40.7455	1	1
33.99762	1	1
-5.32558	1	1
41.05722	1	1
55.67925	1	1

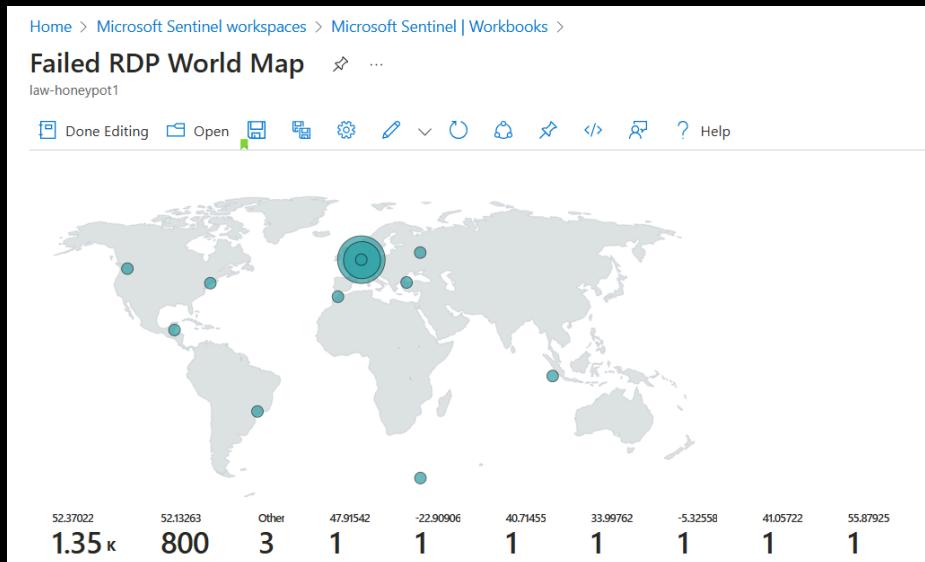
On the left, the "Map Settings" pane is open, showing the following configuration:

- Layout Settings**:
  - Location Info using: Latitude/Longitude
  - Latitude: 70
  - Longitude: 10
  - Size by: EventCount
  - Aggregation for location: Sum of values
  - Minimum region size: 20
  - Maximum region size: 70
  - Default region size: 10
  - Minimum value: (auto)
  - Maximum value: (auto)
  - Opacity of items on Map: 0.7
- Color Settings**:
  - Coloring Type: None
- Metric Settings**:
  - Metric Label: Latitude
  - Metric Value: EventCount
  - Create 'Others' group after: 10
  - Aggregate 'Others' metrics by: Sum of values
  - Custom number formatting

On the right, another "Map Settings" pane is shown with the following configuration:

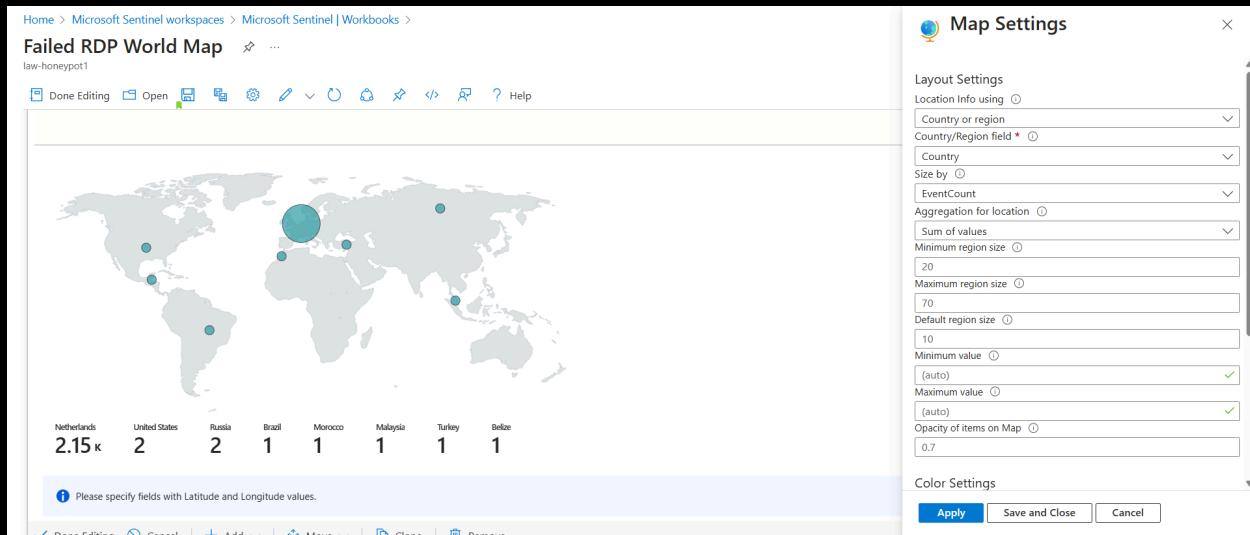
- 70
- 10
- Minimum value: (auto)
- Maximum value: (auto)
- Opacity of items on Map: 0.7
- Color Settings**:
  - Coloring Type: None
- Metric Settings**:
  - Metric Label: Latitude
  - Metric Value: EventCount
  - Create 'Others' group after: 10
  - Aggregate 'Others' metrics by: Sum of values

**Global Map View**, displaying all log activity, and ready to register future event activity (filtering by Latitude):

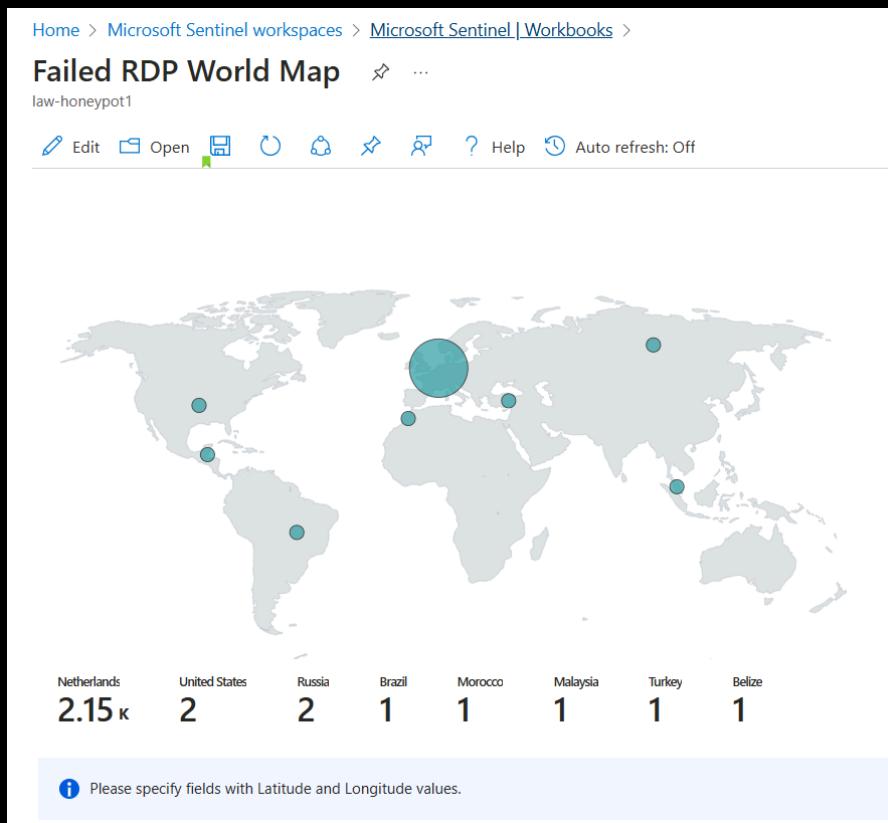


## 7.3 Country-Based Map

- Global Map settings to filter by Country:



**Global Map View**, displaying all log activity, and ready to register future event activity (filtering by country):



## 8.0 Results and Analysis

### Observed Patterns:

- The map visualisation highlights clusters of failed logins from specific locations.
- Attack Trends: A large number of failed attempts from certain countries may indicate a brute-force campaign.

The integration of Sentinel with the honeypot VM provided real-time alerts and detailed analysis of attack patterns. The visual map confirmed the presence of multiple failed login attempts originating from various global regions, signaling potential automated attack campaigns. The project highlighted the effectiveness of honeypots in attracting real-world attackers and demonstrated the ability to use Sentinel to detect and analyse real attack data in a production-like environment.

## 9.0 Challenges and Solutions

Challenge	Solution
Incomplete logs in Event Viewer	Configured advanced logging policies to capture all security events.
No geolocation data for attacks	Integrated IP Geolocation API to enrich logs with country and coordinates.
Sentinel map not displaying data	Adjusted metric settings and verified correct latitude/longitude formatting.

## 10.0 Project Summary

This project successfully demonstrated how a honeypot can be leveraged to detect and analyse real-world RDP brute-force attacks. By integrating Azure Sentinel, custom KQL queries, and geolocation enrichment, the system effectively tracked and visualized attack trends. The insights gained highlight the importance of proactive threat detection and SOC monitoring in defending against cyber threats.

### Future Improvements:

- Automate attack response by integrating Sentinel playbooks.
- Correlate logs across multiple honeypots for enhanced threat intelligence.
- Expand monitoring to additional attack vectors beyond RDP brute force.