

Christian Sanchez

September 9, 2024

SIEM Lab – Live RDP Brute Force

ABSTRACT

This project aims to analyze and derive insights from live attack traffic to understand the risks associated with unsecured endpoints. The project employed Microsoft Azure to set up a honeypot virtual machine, establish a log analytics workspace, configure Microsoft Sentinel for security information and event management (SIEM), and visualize the collected data. PowerShell was utilized to extract data on failed Remote Desktop Protocol (RDP) logins identified in the Windows Event Viewer, which was then enriched with geolocation information via an API to pinpoint the origin of these login attempts. To maximize data collection, the virtual machine was configured with an inbound network rule to allow all traffic, and the Windows Firewall was disabled, increasing its visibility and accessibility to potential attackers. A custom log was generated and visualized in Sentinel, displaying attack attempts geographically by country, latitude, and longitude. The analysis of this data provided valuable insights into the vulnerabilities of inadequately secured endpoints and highlighted the significant risks of exposing such systems on insecure networks.

PROCEDURE

1. Set up Microsoft Azure subscription.
2. Create a virtual machine (VM) in Azure.
3. Configure a custom inbound network policy to allow all traffic.
4. Create a log repository in Log Analytics Workspace.
5. Connect the Log Analytics Workspace to the virtual machine.
6. Set up Microsoft Sentinel for security information and event management (SIEM).
7. Access the VM through Remote Desktop Protocol (RDP).
8. Disable the Windows Firewall on the VM.
9. Run a PowerShell script to extract failed RDP login data and retrieve geolocation information using ipgeolocation.io.
10. Create a custom log in Log Analytics Workspace using the script data.
11. Design a workbook in Microsoft Sentinel to visualize the failed RDP attempts on a map.
12. Analyze the visualized data to assess attack patterns.

DETAILS

- The first step for creating our lab environment is having an active subscription with Microsoft Azure. This can be done through a paid subscription with Azure, or in this case, I used the free trial offered by Microsoft.
- Once the subscription is in place, the first step is to create the virtual machine that will act as the honeypot for the exercise. Access the “Virtual Machines” service and click “Create”, selecting the “Azure virtual machine” option. In this section I named my virtual machine, selected its location (East US 2), chose an operating system image (Windows 10 Pro), and set the admin account username and password.

The screenshot displays the Azure portal interface for a virtual machine named 'honeypot-vm'. The 'Essentials' tab is active, showing the VM's status as 'Running' in the 'East US 2' region. Key details include the operating system 'Windows (Windows 10 Pro)', size 'Standard B1s (1 vCPU, 1 GiB memory)', and public IP address '13.77.76.23'. The 'Properties' tab is also visible, providing a detailed overview of the VM's configuration, including its computer name, operating system, VM generation, architecture, agent status, and network settings.

Property	Value
Computer name	honeypot-vm
Operating system	Windows (Windows 10 Pro)
VM generation	V2
VM architecture	x64
Agent status	Ready
Agent version	2.74149.1131
Hibernation	Disabled
Host group	-
Host	-
Proximity placement group	-
Colocation status	N/A

Property	Value
Public IP address	13.77.76.23 (Network interface honeypot-vm113)
Public IP address (IPv6)	-
Private IP address	10.0.0.4
Private IP address (IPv6)	-
Virtual network/subnet	honeypot-vm-vnet/default
DNS name	Configure

Property	Value
Size	Standard B1s
Size	Standard B1s
vCPUs	1
RAM	1 GiB

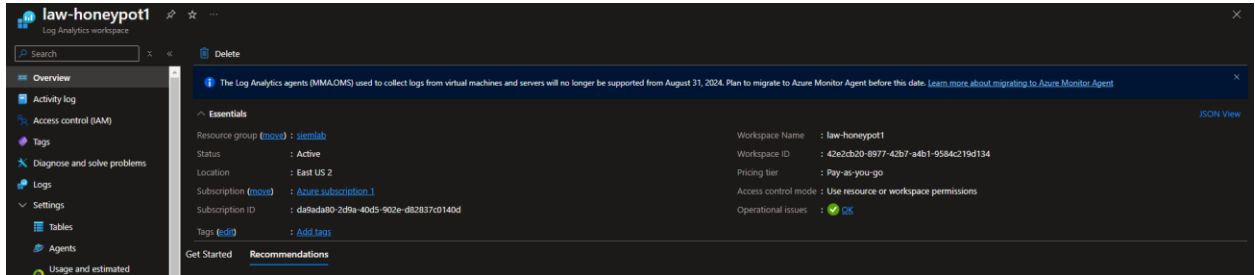
- Before actually creating the virtual machine, in the “Networking” tab of the VM creation process, I created a network security group with an inbound rule allowing traffic to any destination port using any protocol. This acts as an insecure firewall allowing any internet traffic into the virtual machine. After this step, you can select review and create, and Azure will begin creating the virtual machine.

The screenshot shows the Azure portal interface for a network security group named 'honeypot-vm-nsg'. The 'Essentials' tab is active, displaying the NSG's location in 'East US 2' and its association with '0 subnets, 1 network interfaces'. The 'Rules' tab is also visible, showing a list of inbound and outbound security rules. A specific rule named 'WARNING_ANY_IN' is highlighted, showing its configuration: priority 100, name 'WARNING_ANY_IN', port 'Any', protocol 'Any', source 'Any', destination 'Any', and action 'Allow'.

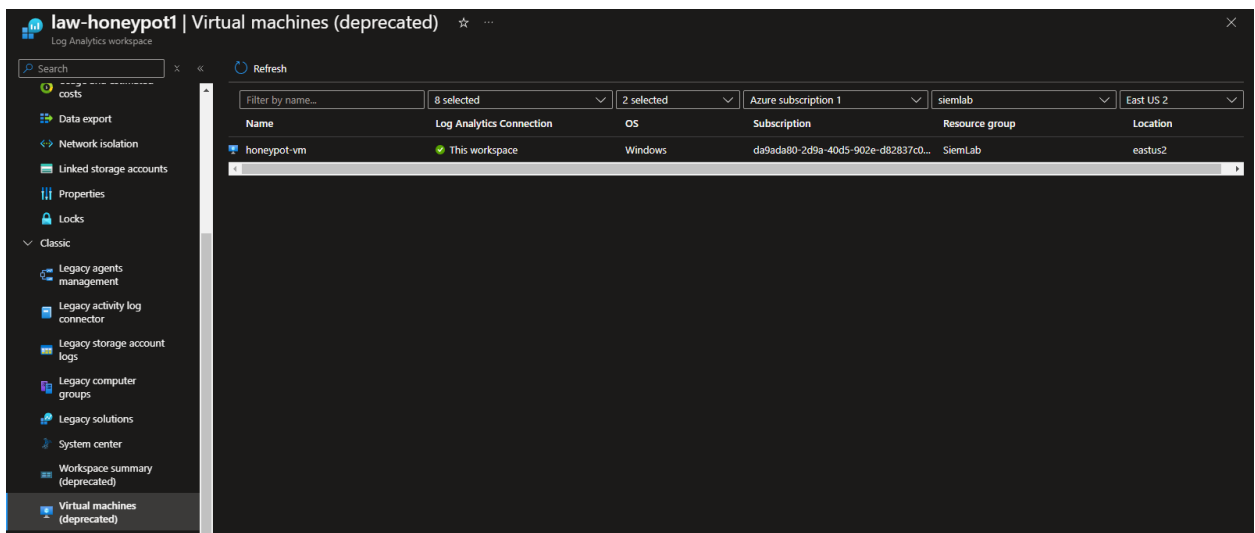
Priority	Name	Port	Protocol	Source	Destination	Action
100	WARNING_ANY_IN	Any	Any	Any	Any	Allow
65000	AllowVnetInbound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInbound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInbound	Any	Any	Any	Any	Deny

Priority	Name	Port	Protocol	Source	Destination	Action
65000	AllowVnetOutbound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowInternetOutbound	Any	Any	Any	Internet	Allow
65500	DenyAllOutbound	Any	Any	Any	Any	Deny

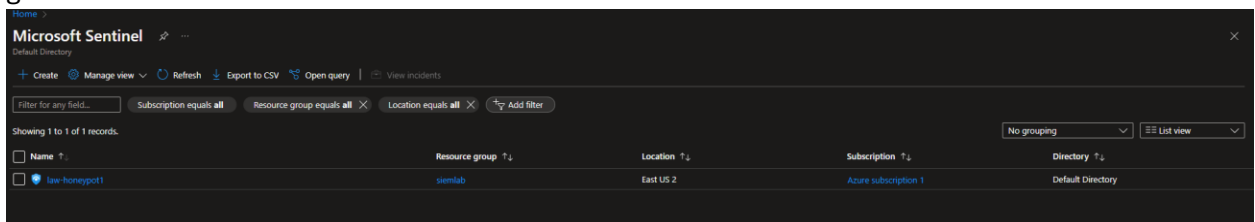
- Next, I created a log repository to ingest logs from the virtual machine. Using “Log Analytics Workspaces” in Azure, click “Create,” name your workspace, and add it to the same resource group as the virtual machine. I also added the workspace to the same location as the virtual machine (East US 2). Finally, review and create.



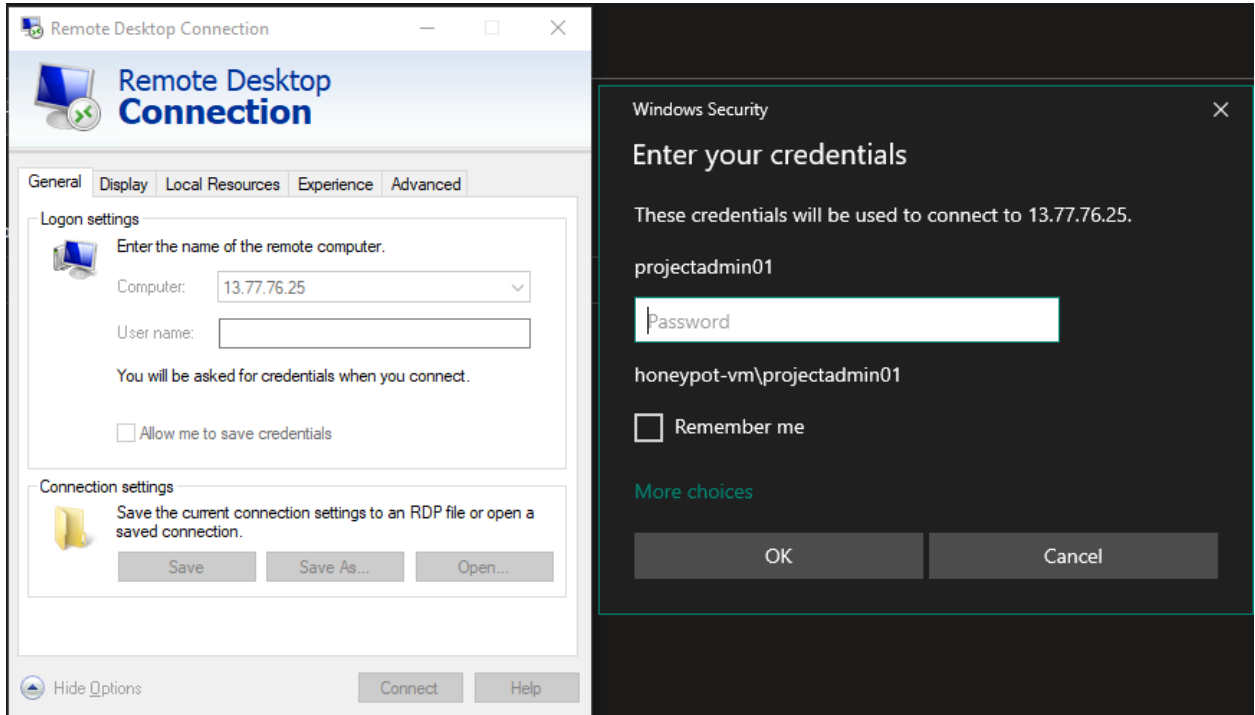
- After creating the log repository, I navigated to “Microsoft Defender for Cloud.” In the left panel under Management, I clicked “Environment Settings.” Here, I navigated to the log analytics workspace I created and turned off “SQL servers on machines.” Then, under Data Collection on the left panel, I selected “All Events” for collection. I then went back to “Log Analytics Workspaces,” selected the workspace I created, and in the left panel selected “Virtual Machines.” Here, I selected the honeypot VM and clicked “Connect” to link the log repository to the virtual machine.



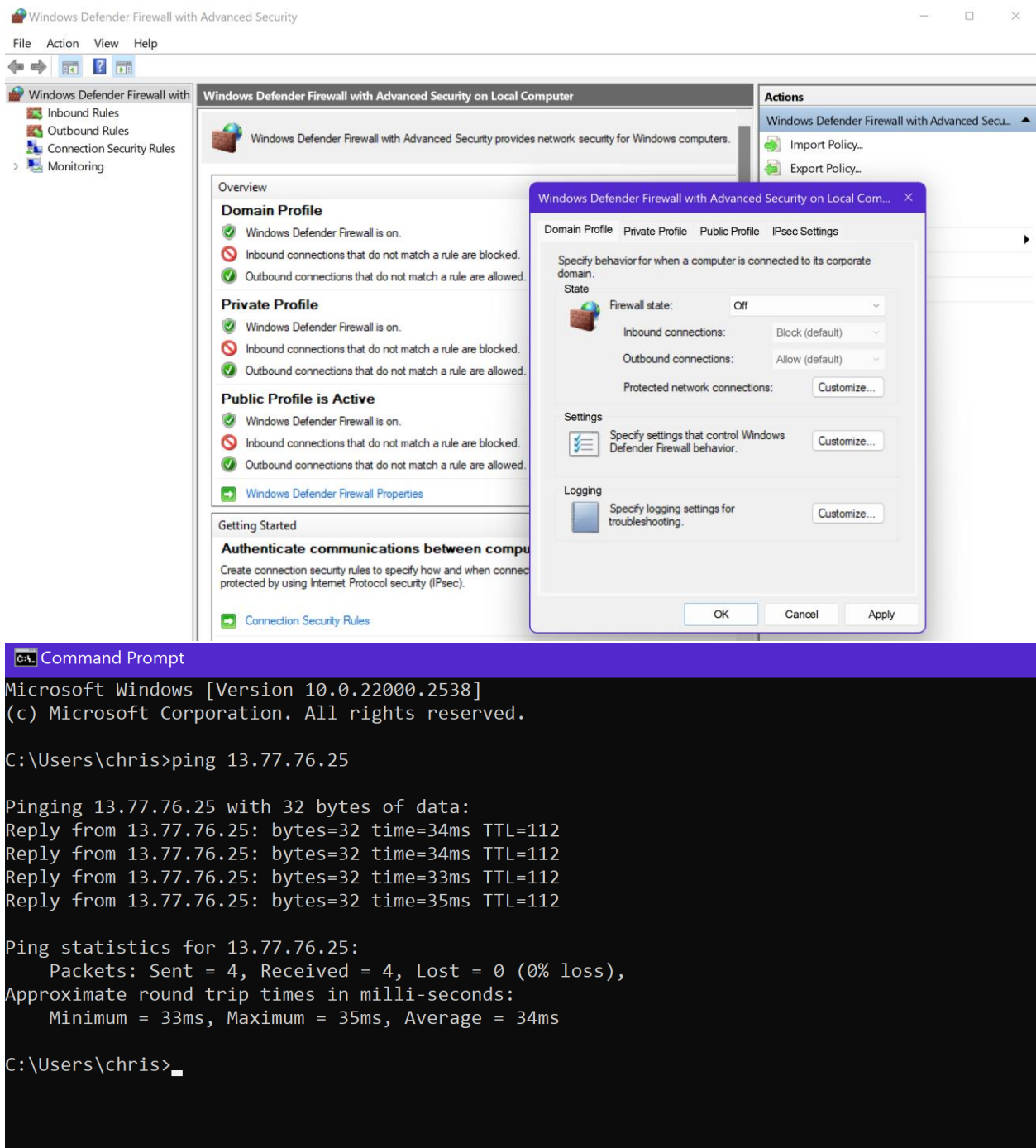
- The next step was to set up the Security Information and Event Management (SIEM) using Microsoft Sentinel. In Sentinel, I selected the log analytics workspace created and added it to Sentinel. The SIEM will be used later to analyze the attack data and create a map using geolocation data.



- Now that there is a functional virtual machine, a repository to ingest the log data, and a SIEM for event management, it is time to remote login to the virtual machine using Remote Desktop Protocol (RDP). Keep in mind that failed remote desktop logins are used to plot the data later. Brute-forcing the login steps will generate the data used. Using the start menu, I searched for and opened the “Remote Desktop Connection” application. By entering the IP address of the VM that was created, I was prompted to enter a username and password, which are the credentials created during the VM setup. After logging in, I was met with the Windows 10 Pro interface on a new screen.



- Once logged into the virtual machine, it is time to shut off the Windows Firewall so the VM can respond to ICMP echo requests, making it more easily discoverable. Before shutting off the Windows Firewall, using the command line to ping the VM IP address resulted in a request timeout, indicating that the destination IP address could not communicate with other devices. To resolve this, I located wf.msc in the Windows start menu. Once open, I selected “Windows Defender Firewall Properties” and, for “Firewall State,” turned off the firewall for domain, private, and public profiles, then clicked “Apply.” Now that the Windows Firewall is off, using the same ping command to the VM as before, the destination IP now replies to the request.



- Next, I downloaded a log exporter PowerShell script from Josh Madakors page on GitHub that pulls IP addresses from the Event Viewer. In conjunction with an IP geolocation API, it retrieves geolocation data such as country, state, city, latitude, longitude, and time zone. To function properly, I requested a new API key from the website and replaced the default key in the PowerShell script, then saved it to my virtual machine. The free version of the service only allows for 1,000 API requests, so I used the paid version for more extensive results at the end of

the lab. After running the script, it created a log file and started recording any failed RDP logons (Event ID 4625) with the associated geolocation data.

- [PowerShell Script](#)
- [Geolocation API](#)

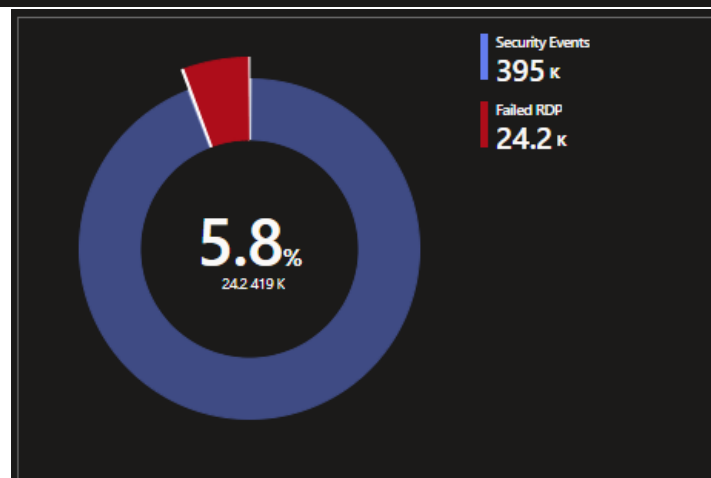
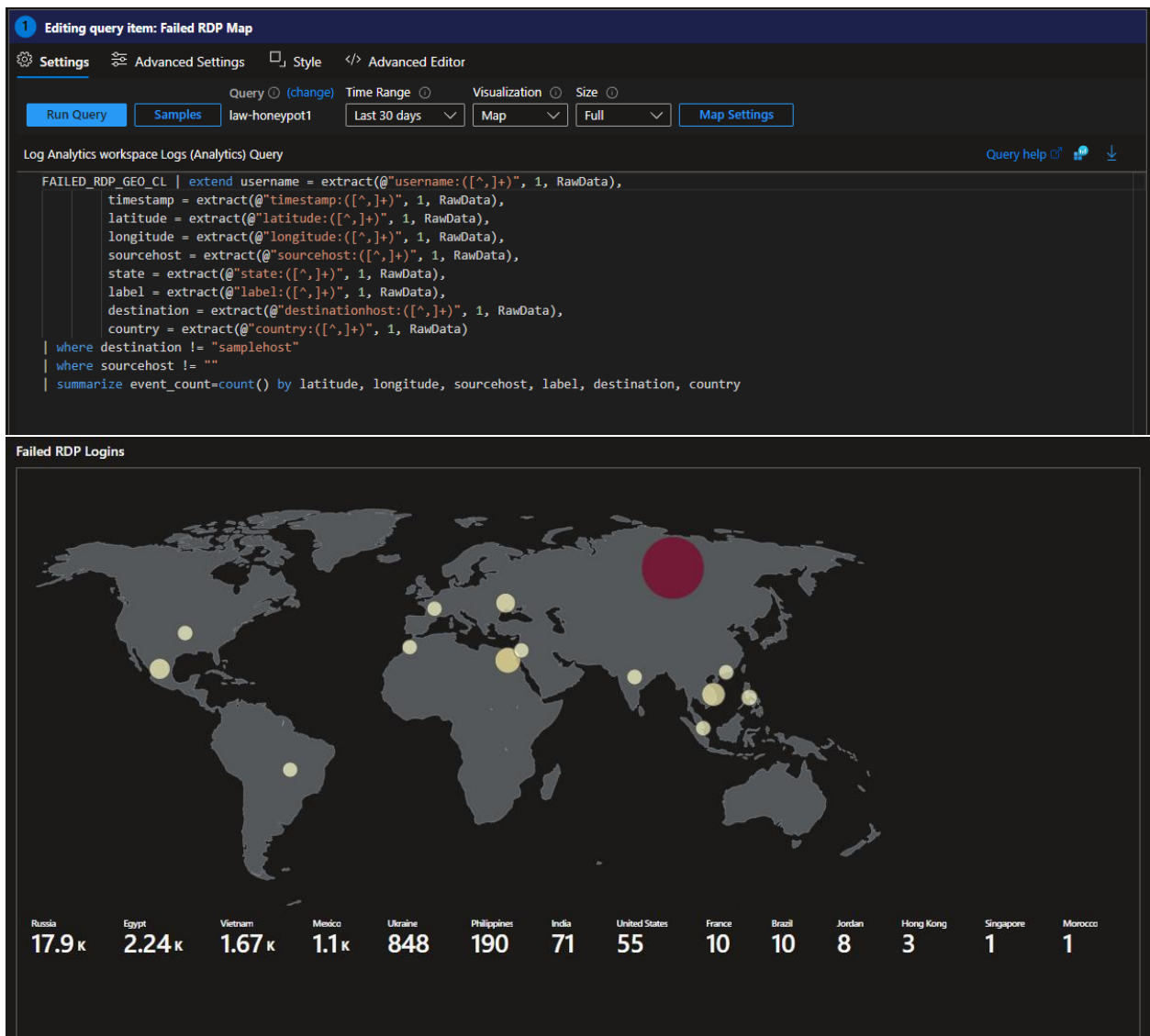
ADD PICTURE OF SCRIPT RUNNING

- Now that there is a log with the geodata, we can create a custom log in Log Analytics Workspace that allows us to interact with the data. In the left side panel in the Log Analytics Workspace under “Tables,” then “Create,” I created a new custom log. Here it asks for a sample log, which trains Log Analytics on what to look for in the data. I used the sample log that was created on the VM when I first ran the PowerShell script. Next, it asks for the collection path, which is where the log is located on the VM. Since it’s running Windows 10 Pro, I selected the type as Windows and entered the exact file path (e.g., C:\ProgramData). In “Details,” I named the custom log FAILED_RDP_GEO_CL and then finished creating the log.

The top screenshot shows the Microsoft Azure portal interface for the 'law-honeypot1' Log Analytics workspace. The 'Tables' view is active, displaying a list of tables. The table 'FAILED_RDP_GEO_CL' is highlighted, showing its description and collection paths. The bottom screenshot shows the 'Logs' view for the same workspace. A query is running, and the results are displayed in a table with columns: TimeGenerated [UTC], Computer, RawData, and Type. The results show multiple entries for 'FAILED_RDP_GEO_CL' with timestamps and computer names.

TimeGenerated [UTC]	Computer	RawData	Type
9/9/2024, 6:31:08.992 AM	honeypot-vm	latitude:59.88881,longitude:30.31671,destinationhost:honey...	FAILED_RDP_GEO_CL
9/9/2024, 6:31:08.992 AM	honeypot-vm	latitude:59.88881,longitude:30.31671,destinationhost:honey...	FAILED_RDP_GEO_CL
9/9/2024, 6:31:08.992 AM	honeypot-vm	latitude:59.88881,longitude:30.31671,destinationhost:honey...	FAILED_RDP_GEO_CL
9/9/2024, 6:31:08.992 AM	honeypot-vm	latitude:59.88881,longitude:30.31671,destinationhost:honey...	FAILED_RDP_GEO_CL
9/9/2024, 6:31:08.992 AM	honeypot-vm	latitude:59.88881,longitude:30.31671,destinationhost:honey...	FAILED_RDP_GEO_CL
9/9/2024, 6:31:08.992 AM	honeypot-vm	latitude:31.95395,longitude:35.91064,destinationhost:honey...	FAILED_RDP_GEO_CL
9/9/2024, 6:31:08.992 AM	honeypot-vm	latitude:59.88881,longitude:30.31671,destinationhost:honey...	FAILED_RDP_GEO_CL
9/9/2024, 6:31:08.992 AM	honeypot-vm	latitude:59.88881,longitude:30.31671,destinationhost:honey...	FAILED_RDP_GEO_CL
9/9/2024, 6:31:08.992 AM	honeypot-vm	latitude:59.88881,longitude:30.31671,destinationhost:honey...	FAILED_RDP_GEO_CL

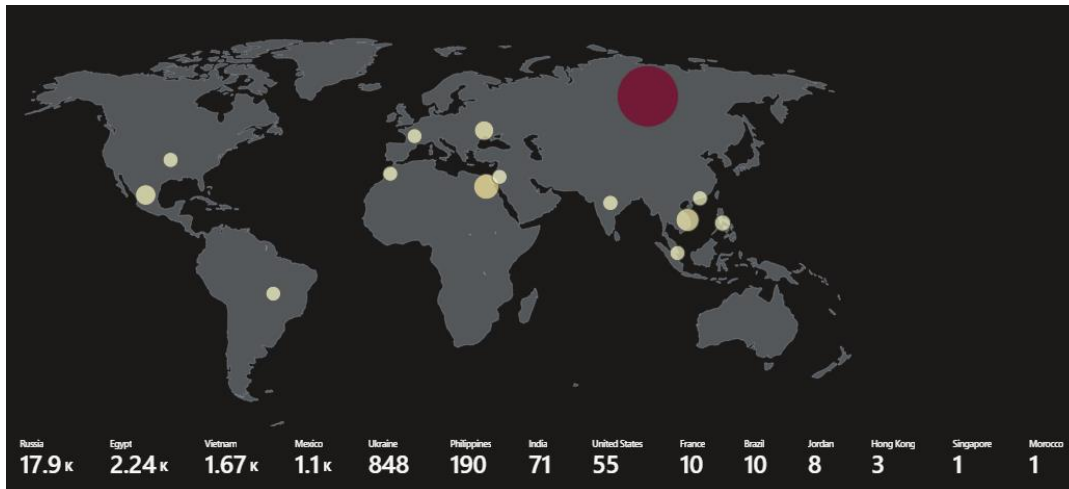
- In Microsoft Sentinel under Threat Management, I created a workbook called “Failed RDP Map.” I edited the workbook and ran a query (KQL) to extract fields from the raw data in the custom log, including timestamp, latitude, longitude, source host, state, label, destination, and country. In the visualization tab, I selected “Map.” In map settings, I could choose to plot the location info by latitude and longitude or by country. Latitude and longitude are much more precise on the map, while plotting by country provides a broader view. I also added a pie chart indicating the percentage of security events that are RDP login failures.



- The final results from the data collection show diversity in the attack locations and different levels of attack intensity from each place. Thousands of events plotted on the map help

demonstrate the severity of the threat from each country, and the pie chart shows how common just one avenue of attack can be on a network.

FINAL RESULTS



- After a few days of monitoring the traffic, I collected over 397,000 security events on the virtual machine. From these events, 24,200 came from failed RDP logins to the machine. Over the course of the experiment, attacks were recorded from 14 different countries around the world. The biggest threat by far was Russia, with an incredible 17,900 failed logins. Russia was eight times more active than the next biggest threat, Egypt, which had 2,240 failed attempts. Russia accounted for nearly 75% of the total RDP failures. Vietnam, Mexico, the Philippines, and Ukraine were also very active, with clear signs of brute force activity. These countries all logged heavy, simultaneous traffic, indicating that the attackers were persistent and likely more sophisticated than those from countries with fewer attempts. The attackers from the US, France, Brazil, Jordan, Hong Kong, Singapore, and Morocco were not very persistent; it appears that these attackers would try a few passwords before giving up.

LESSONS LEARNED

- Importance of Password Security
 - I realized very quickly while performing this lab that password complexity is crucial for protecting an account. On my first attempt at this lab, I used a relatively weak username and password for ease of access. Within the first 30 minutes of monitoring the results, I was logged out of the virtual machine because another connection was established. This occurred several times due to a common username and an easily guessable password. It is paramount that end users practice proper password hygiene.

At a minimum, users should have a password with more than 10 characters, featuring numbers, uppercase and lowercase letters, and symbols. A unique username should also be used. When reviewing the attempted usernames from each login attempt, the following appeared very frequently: Administrator, Admin, User, Username, Client, and Default.

- Discoverability
 - Anything with an internet connection is not safe. I was shocked by how quickly the IP address of the virtual machine was discovered and even more surprised at how rapidly it could be brute-forced and accessed by an unauthorized user. Within minutes, the IP address was discovered by several countries, and once discovered, the attempts to gain access were relentless.
- Threat Landscape
 - The results of this experiment show that threats can originate from anywhere in the world when exposed to the internet. I expected to see more traffic from areas like the US or neighboring regions, but the majority of the traffic came from distant locations such as Russia and Egypt. I also noted that some threats were more persistent than others. For instance, once Russia discovered the IP address, the login requests did not stop, a pattern also observed with Egypt, Vietnam, and Ukraine. Other threats were less concerning, such as those from Morocco, France, the United States, and Singapore. These were less alarming because there was significantly less volume compared to countries like Russia, and the attackers would stop after a few attempts. Furthermore, the login attempts from these countries were spaced out, unlike the continuous requests from more active countries. I would also like to add that the threat landscape is significantly broader than what was covered in this experiment. I only collected data from failed logins using Remote Desktop, although there are certainly other attack vectors since I allowed all traffic to any destination port. For example, Secure Shell (SSH) on port 22 or Server Message Block (SMB) on port 445 were likely heavily targeted. This means that a significant number of attacks went undetected, which could be an interesting topic for a future project.
- Network Policy and Firewall
 - Setting up inbound network rules and keeping your Windows Firewall secure are the most effective ways to prevent this type of attack. Before shutting off the Windows Firewall, external communication to the computer was not possible, which alone prevents outsiders from attempting to access your computer. The inbound network rule used for this project left the computer completely vulnerable to any incoming network traffic. Ideally, risky ports like 3389 (RDP) should not be exposed to the internet because they are common targets. The same applies to other frequently targeted ports like 21 (FTP), 22 (SSH), 80 (HTTP), and 445 (SMB).