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.

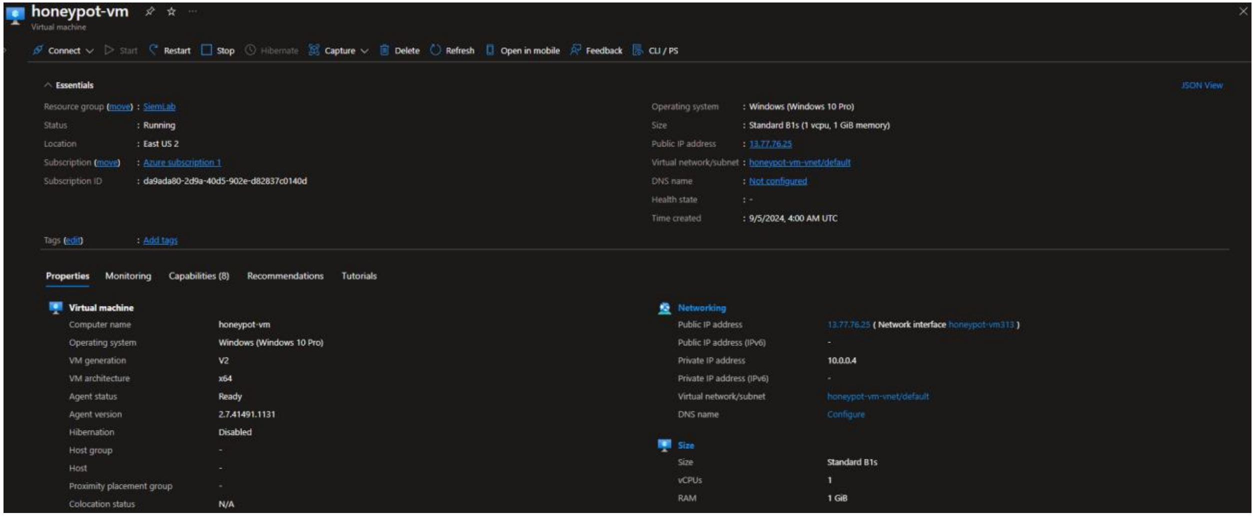
PROCEDURE

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.



honeypot-vm
Virtual machine

Essentials

Resource group (move) : SiemLab
Status : Running
Location : East US 2
Subscription (move) : Azure subscription 1
Subscription ID : da9ada80-2d9a-40d5-902e-d82837c0140d
Tags (edit) : Add tags

Operating system : Windows (Windows 10 Pro)
Size : Standard B1s (1 vcpu, 1 GiB memory)
Public IP address : 13.77.76.25
Virtual network/subnet : honeypot-vm-vnet/default
DNS name : Not configured
Health state : -
Time created : 9/5/2024, 4:00 AM UTC

Virtual machine

Computer name	honeypot-vm
Operating system	Windows (Windows 10 Pro)
VM generation	V2
VM architecture	x64
Agent status	Ready
Agent version	2.7.41491.1131
Hibernation	Disabled
Host group	-
Host	-
Proximity placement group	-
Colocation status	N/A

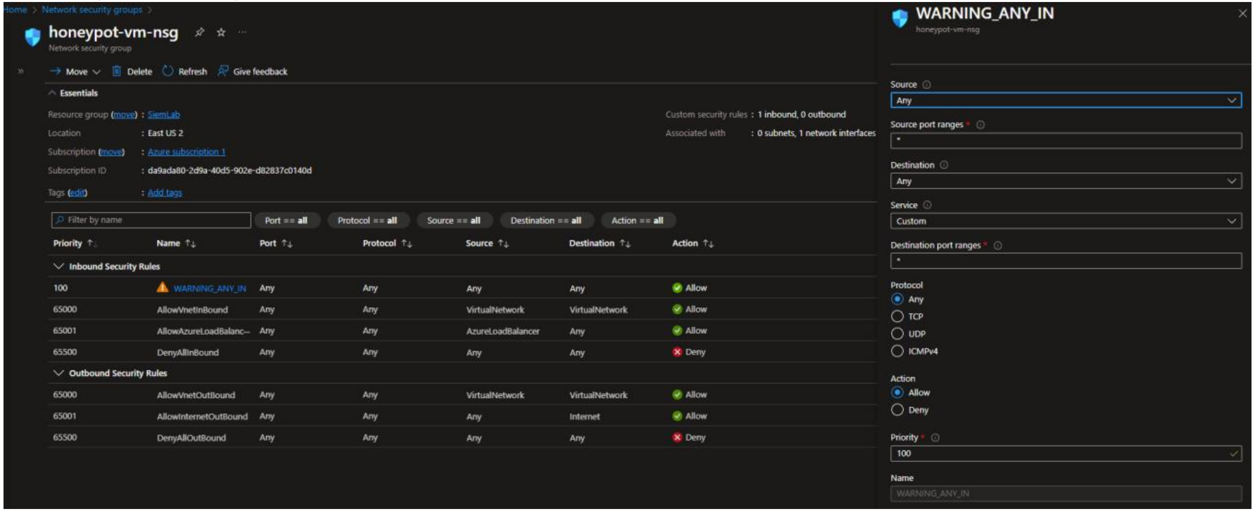
Networking

Public IP address	13.77.76.25 (Network interface honeypot-vm313)
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

Size

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.



honeypot-vm-nsg
Network security group

Essentials

Resource group (move) : SiemLab
Location : East US 2
Subscription (move) : Azure subscription 1
Subscription ID : da9ada80-2d9a-40d5-902e-d82837c0140d
Tags (edit) : Add tags

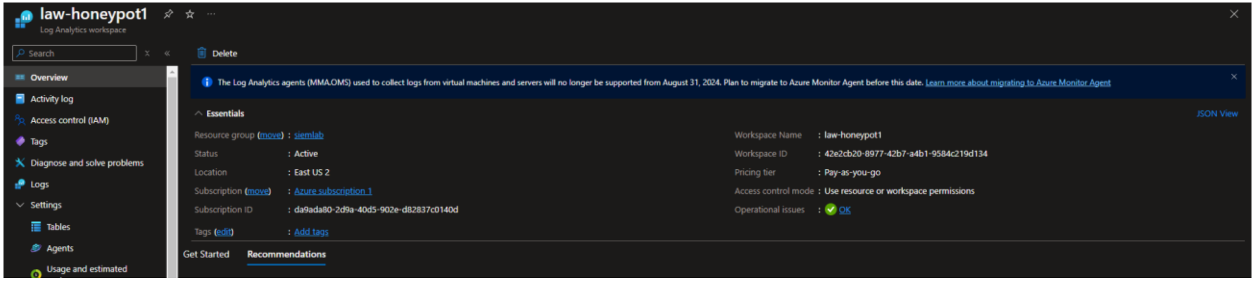
Custom security rules : 1 inbound, 0 outbound
Associated with : 0 subnets, 1 network interfaces

Priority ↑↓	Name ↑↓	Port ↑↓	Protocol ↑↓	Source ↑↓	Destination ↑↓	Action ↑↓
Inbound Security Rules						
100	⚠ WARNING_ANY_IN	Any	Any	Any	Any	✓ Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	✓ Allow
65001	AllowAzureLoadBalanc...	Any	Any	AzureLoadBalancer	Any	✓ Allow
65500	DenyAllInBound	Any	Any	Any	Any	✗ Deny
Outbound Security Rules						
65000	AllowVnetOutBound	Any	Any	VirtualNetwork	VirtualNetwork	✓ Allow
65001	AllowInternetOutBound	Any	Any	Any	Internet	✓ Allow
65500	DenyAllOutBound	Any	Any	Any	Any	✗ Deny

WARNING_ANY_IN
honeypot-vm-nsg

Source: Any
Source port ranges *: *
Destination: Any
Service: Custom
Destination port ranges *: *
Protocol
● Any
○ TCP
○ UDP
○ ICMPv4
Action
● Allow
○ Deny
Priority *: 100
Name: WARNING_ANY_IN

- 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.



law-honeypot1

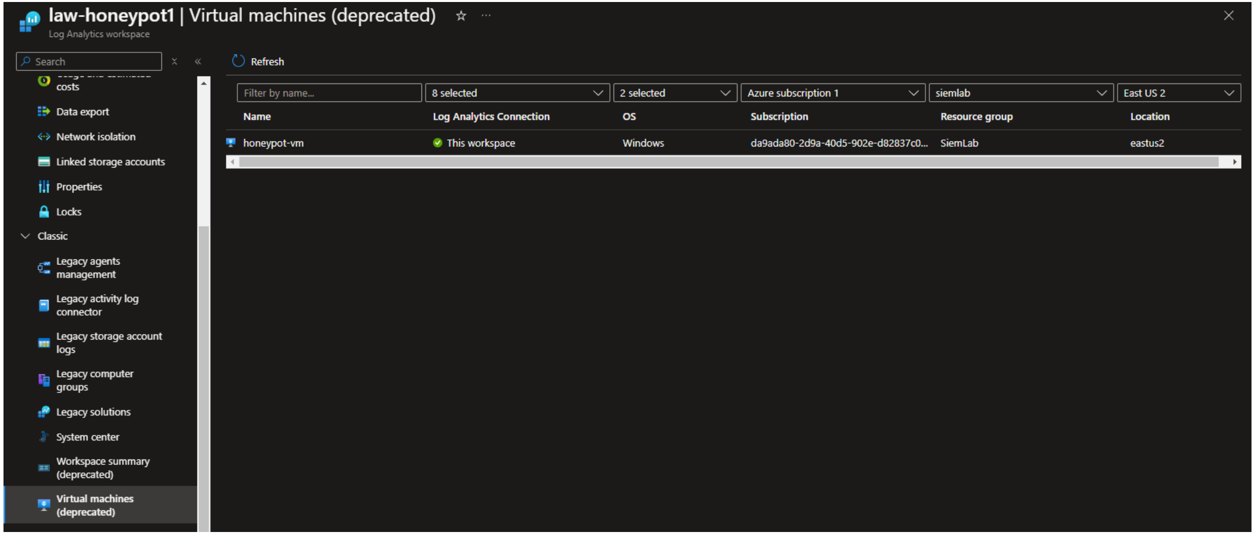
Log Analytics workspace

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)	: siemlab	Workspace Name	: law-honeypot1
Status	: Active	Workspace ID	: 42e2cb20-8977-42b7-a4b1-9584c219d134
Location	: East US 2	Pricing tier	: Pay-as-you-go
Subscription (move)	: Azure subscription 1	Access control mode	: Use resource or workspace permissions
Subscription ID	: da9ada80-2d9a-40d5-902e-d82837c0140d	Operational issues	: OK
Tags (edit)	: Add tags		

- 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.



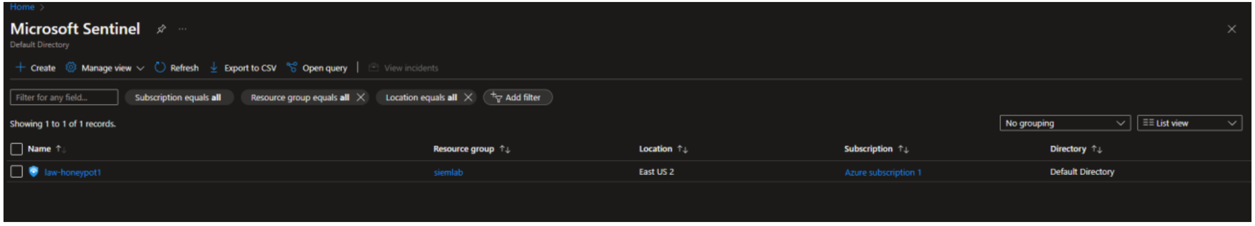
law-honeypot1 | Virtual machines (deprecated)

Log Analytics workspace

Filter by name... | 8 selected | 2 selected | Azure subscription 1 | siemlab | East US 2

Name	Log Analytics Connection	OS	Subscription	Resource group	Location
honeypot-vm	✓ This workspace	Windows	da9ada80-2d9a-40d5-902e-d82837c0...	SiemLab	eastus2

- 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.



Microsoft Sentinel

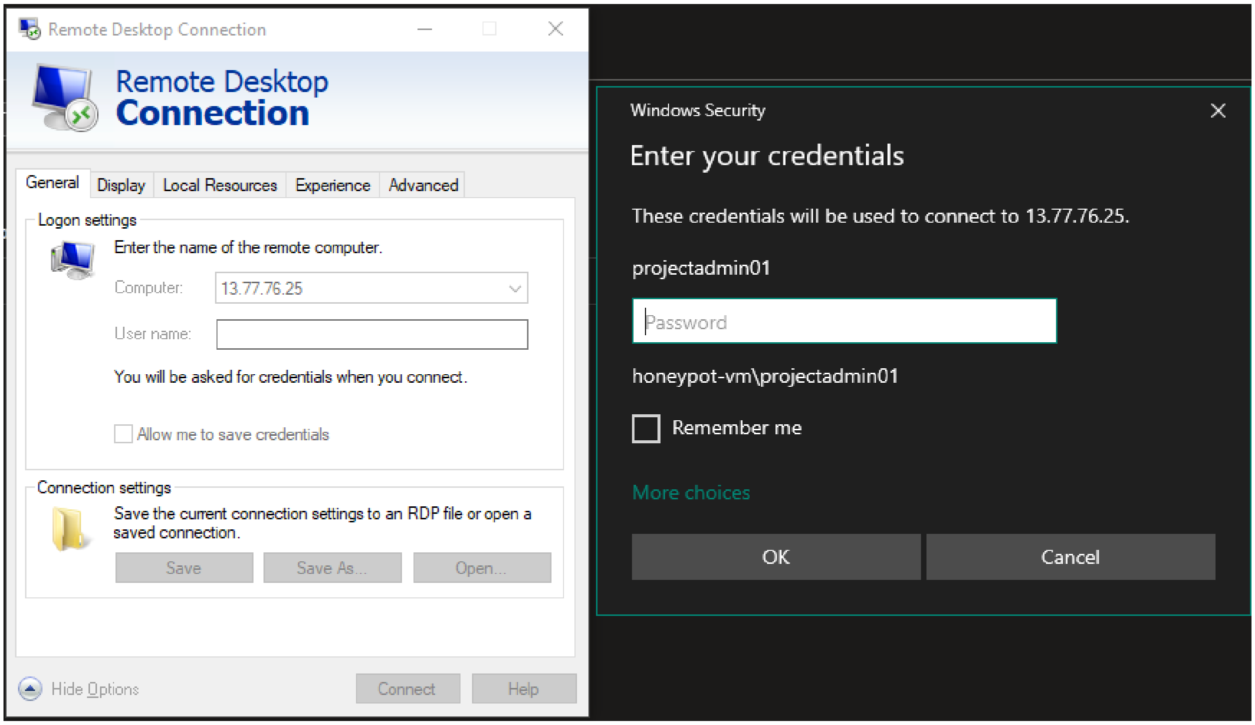
Default Directory

Filter for any field... Subscription equals all Resource group equals all Location equals all

Showing 1 to 1 of 1 records.

Name	Resource group	Location	Subscription	Directory
law-honeypot1	siemlab	East US 2	Azure subscription 1	Default Directory

- 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.



Enter the name of the remote computer.

Computer: 13.77.76.25

User name:

You will be asked for credentials when you connect.

□ Allow me to save credentials

Save the current connection settings to an RDP file or open a saved connection.

Enter your credentials

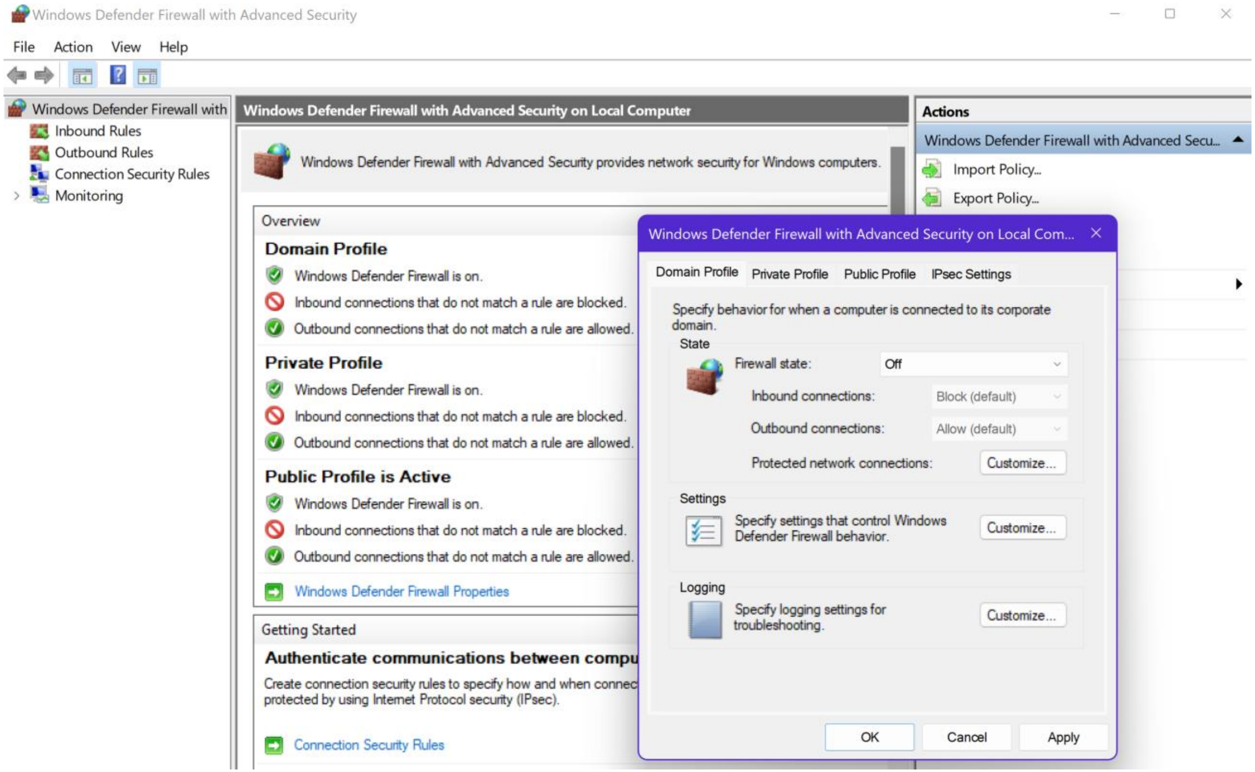
These credentials will be used to connect to 13.77.76.25.

projectadmin01

honeypot-vm\projectadmin01

□ Remember me

- 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.



Windows Defender Firewall with Advanced Security
File
Action
View
Help
Windows Defender Firewall with
Inbound Rules
Outbound Rules
Connection Security Rules
Monitoring
Actions
Windows Defender Firewall with Advanced Secu...
Import Policy...
Export Policy...

Windows Defender Firewall with Advanced Security on Local Computer

Windows Defender Firewall with Advanced Security provides network security for Windows computers.

Overview

Domain Profile

- Windows Defender Firewall is on.
- Inbound connections that do not match a rule are blocked.
- Outbound connections that do not match a rule are allowed.

Private Profile

- Windows Defender Firewall is on.
- Inbound connections that do not match a rule are blocked.
- Outbound connections that do not match a rule are allowed.

Public Profile is Active

- Windows Defender Firewall is on.
- Inbound connections that do not match a rule are blocked.
- Outbound connections that do not match a rule are allowed.

Windows Defender Firewall Properties

Getting Started

Authenticate communications between compu

Create connection security rules to specify how and when connec protected by using Internet Protocol security (IPsec).

Connection Security Rules

Windows Defender Firewall with Advanced Security on Local Com...

Specify behavior for when a computer is connected to its corporate domain.

State

Firewall state: Off

Inbound connections: Block (default)

Outbound connections: Allow (default)

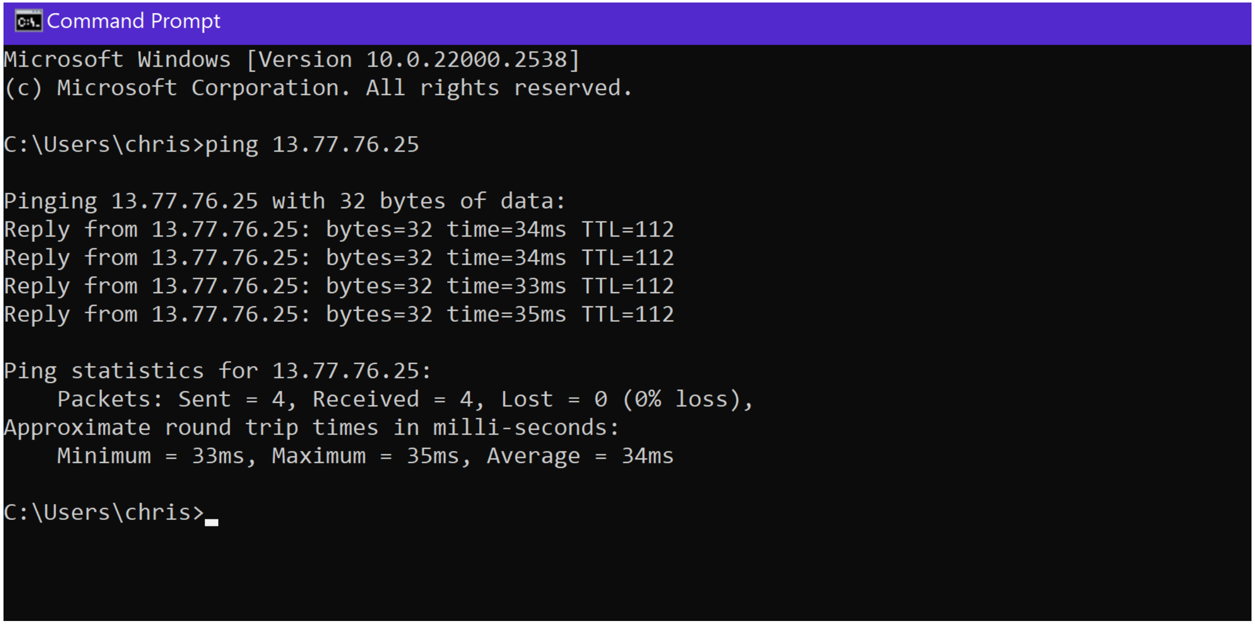
Protected network connections: Customize...

Settings

Specify settings that control Windows Defender Firewall behavior. Customize...

Logging

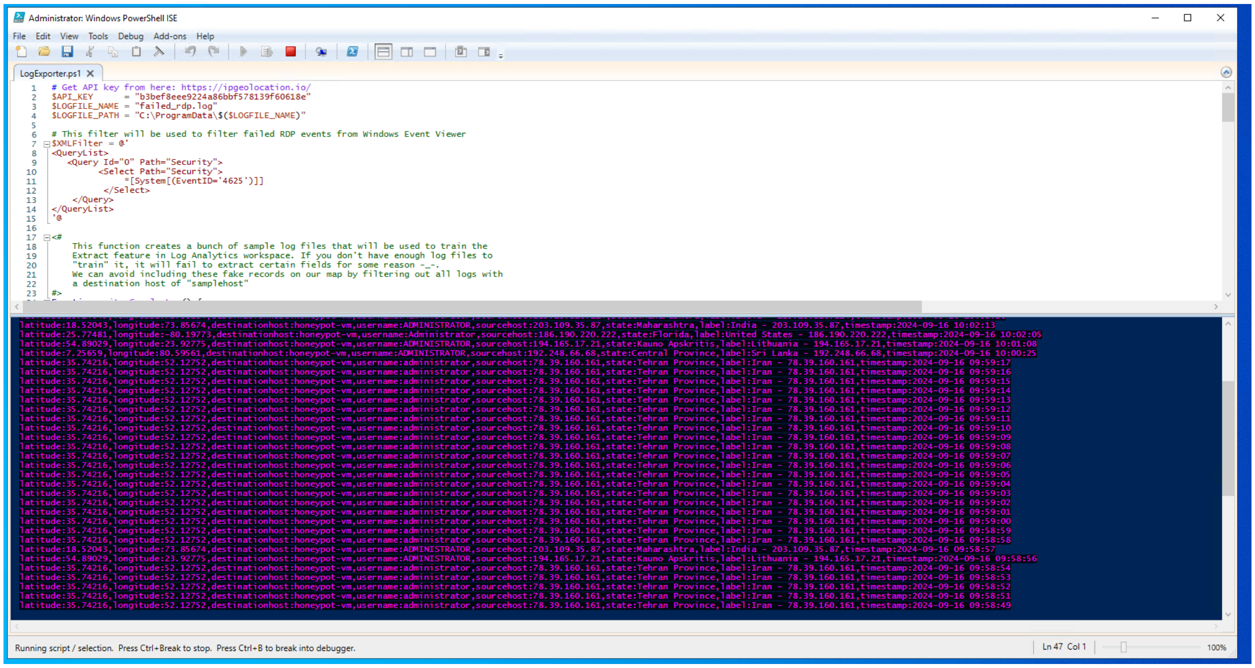
Specify logging settings for troubleshooting. Customize...



Command Prompt

```
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\chris>ping 13.77.76.25

Pinging 13.77.76.25 with 32 bytes of data:
Reply from 13.77.76.25: bytes=32 time=34ms TTL=112
Reply from 13.77.76.25: bytes=32 time=34ms TTL=112
Reply from 13.77.76.25: bytes=32 time=33ms TTL=112
Reply from 13.77.76.25: bytes=32 time=35ms TTL=112

Ping statistics for 13.77.76.25:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 33ms, Maximum = 35ms, Average = 34ms

C:\Users\chris>_
```

- 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



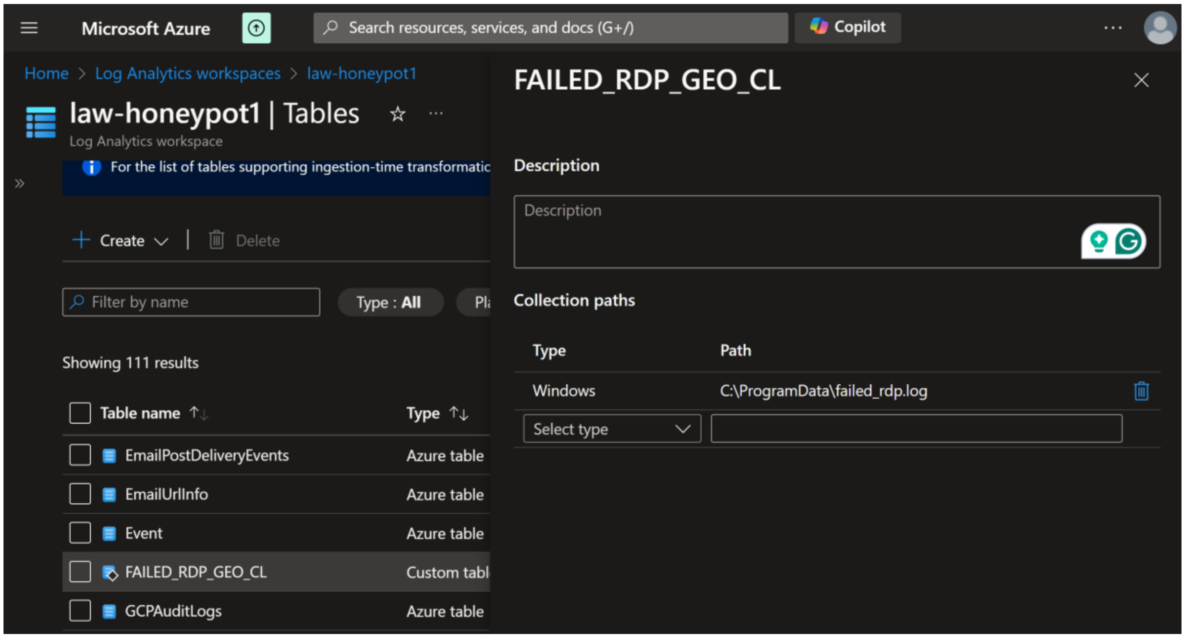
Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
LogExporter.ps1

```
# Get API key from here: https://ipgeolocation.io/
$API_KEY      = "b3bef8eee9224a86bbf578139f60618e"
$LOGFILE_NAME = "failed_rdp.log"
$LOGFILE_PATH = "C:\ProgramData\$($LOGFILE_NAME)"

# This filter will be used to filter failed RDP events from Windows Event Viewer
$XMLFilter = @'
<QueryList>
   <Query Id="0" Path="Security">
         <Select Path="Security">
              *[System[(EventID='4625')]]
          </Select>
    </Query>
</QueryList>
'@

<#
    This function creates a bunch of sample log files that will be used to train the
    Extract feature in Log Analytics workspace. If you don't have enough log files to
    "train" it, it will fail to extract certain fields for some reason -_-.
    We can avoid including these fake records on our map by filtering out all logs with
    a destination host of "samplehost"
#>
```

```
latitude:18.52043,longitude:73.85674,destinationhost:honeypot-vm,username:ADMINISTRATOR,sourcehost:203.109.35.87,state:Maharashtra,label:India - 203.109.35.87,timestamp:2024-09-16 10:02:13
latitude:25.77481,longitude:-80.19773,destinationhost:honeypot-vm,username:Administrator,sourcehost:186.190.220.222,state:Florida,label:United States - 186.190.220.222,timestamp:2024-09-16 10:02:05
latitude:54.89029,longitude:23.92775,destinationhost:honeypot-vm,username:ADMINISTRATOR,sourcehost:194.165.17.21,state:Kauno Apskritis,label:Lithuania - 194.165.17.21,timestamp:2024-09-16 10:01:08
latitude:7.25659,longitude:80.59561,destinationhost:honeypot-vm,username:ADMINISTRATOR,sourcehost:192.248.66.68,state:Central Province,label:Sri Lanka - 192.248.66.68,timestamp:2024-09-16 10:00:25
latitude:35.74216,longitude:52.12752,destinationhost:honeypot-vm,username:administrator,sourcehost:78.39.160.161,state:Tehran Province,label:Iran - 78.39.160.161,timestamp:2024-09-16 09:59:17
latitude:35.74216,longitude:52.12752,destinationhost:honeypot-vm,username:administrator,sourcehost:78.39.160.161,state:Tehran Province,label:Iran - 78.39.160.161,timestamp:2024-09-16 09:59:16
latitude:35.74216,longitude:52.12752,destinationhost:honeypot-vm,username:administrator,sourcehost:78.39.160.161,state:Tehran Province,label:Iran - 78.39.160.161,timestamp:2024-09-16 09:59:15
latitude:35.74216,longitude:52.12752,destinationhost:honeypot-vm,username:administrator,sourcehost:78.39.160.161,state:Tehran Province,label:Iran - 78.39.160.161,timestamp:2024-09-16 09:59:14
latitude:35.74216,longitude:52.12752,destinationhost:honeypot-vm,username:administrator,sourcehost:78.39.160.161,state:Tehran Province,label:Iran - 78.39.160.161,timestamp:2024-09-16 09:59:13
latitude:35.74216,longitude:52.12752,destinationhost:honeypot-vm,username:administrator,sourcehost:78.39.160.161,state:Tehran Province,label:Iran - 78.39.160.161,timestamp:2024-09-16 09:59:12
latitude:35.74216,longitude:52.12752,destinationhost:honeypot-vm,username:administrator,sourcehost:78.39.160.161,state:Tehran Province,label:Iran - 78.39.160.161,timestamp:2024-09-16 09:59:11
latitude:35.74216,longitude:52.12752,destinationhost:honeypot-vm,username:administrator,sourcehost:78.39.160.161,state:Tehran Province,label:Iran - 78.39.160.161,timestamp:2024-09-16 09:59:10
latitude:35.74216,longitude:52.12752,destinationhost:honeypot-vm,username:administrator,sourcehost:78.39.160.161,state:Tehran Province,label:Iran - 78.39.160.161,timestamp:2024-09-16 09:59:09
latitude:35.74216,longitude:52.12752,destinationhost:honeypot-vm,username:administrator,sourcehost:78.39.160.161,state:Tehran Province,label:Iran - 78.39.160.161,timestamp:2024-09-16 09:59:08
latitude:35.74216,longitude:52.12752,destinationhost:honeypot-vm,username:administrator,sourcehost:78.39.160.161,state:Tehran Province,label:Iran - 78.39.160.161,timestamp:2024-09-16 09:59:07
latitude:35.74216,longitude:52.12752,destinationhost:honeypot-vm,username:administrator,sourcehost:78.39.160.161,state:Tehran Province,label:Iran - 78.39.160.161,timestamp:2024-09-16 09:59:06
latitude:35.74216,longitude:52.12752,destinationhost:honeypot-vm,username:administrator,sourcehost:78.39.160.161,state:Tehran Province,label:Iran - 78.39.160.161,timestamp:2024-09-16 09:59:05
latitude:35.74216,longitude:52.12752,destinationhost:honeypot-vm,username:administrator,sourcehost:78.39.160.161,state:Tehran Province,label:Iran - 78.39.160.161,timestamp:2024-09-16 09:59:04
latitude:35.74216,longitude:52.12752,destinationhost:honeypot-vm,username:administrator,sourcehost:78.39.160.161,state:Tehran Province,label:Iran - 78.39.160.161,timestamp:2024-09-16 09:59:03
latitude:35.74216,longitude:52.12752,destinationhost:honeypot-vm,username:administrator,sourcehost:78.39.160.161,state:Tehran Province,label:Iran - 78.39.160.161,timestamp:2024-09-16 09:59:02
latitude:35.74216,longitude:52.12752,destinationhost:honeypot-vm,username:administrator,sourcehost:78.39.160.161,state:Tehran Province,label:Iran - 78.39.160.161,timestamp:2024-09-16 09:59:01
latitude:35.74216,longitude:52.12752,destinationhost:honeypot-vm,username:administrator,sourcehost:78.39.160.161,state:Tehran Province,label:Iran - 78.39.160.161,timestamp:2024-09-16 09:59:00
latitude:35.74216,longitude:52.12752,destinationhost:honeypot-vm,username:administrator,sourcehost:78.39.160.161,state:Tehran Province,label:Iran - 78.39.160.161,timestamp:2024-09-16 09:58:59
latitude:35.74216,longitude:52.12752,destinationhost:honeypot-vm,username:administrator,sourcehost:78.39.160.161,state:Tehran Province,label:Iran - 78.39.160.161,timestamp:2024-09-16 09:58:58
latitude:18.52043,longitude:73.85674,destinationhost:honeypot-vm,username:ADMINISTRATOR,sourcehost:203.109.35.87,state:Maharashtra,label:India - 203.109.35.87,timestamp:2024-09-16 09:58:57
latitude:54.89029,longitude:23.92775,destinationhost:honeypot-vm,username:ADMINISTRATOR,sourcehost:194.165.17.21,state:Kauno Apskritis,label:Lithuania - 194.165.17.21,timestamp:2024-09-16 09:58:56
latitude:35.74216,longitude:52.12752,destinationhost:honeypot-vm,username:administrator,sourcehost:78.39.160.161,state:Tehran Province,label:Iran - 78.39.160.161,timestamp:2024-09-16 09:58:54
latitude:35.74216,longitude:52.12752,destinationhost:honeypot-vm,username:administrator,sourcehost:78.39.160.161,state:Tehran Province,label:Iran - 78.39.160.161,timestamp:2024-09-16 09:58:53
latitude:35.74216,longitude:52.12752,destinationhost:honeypot-vm,username:administrator,sourcehost:78.39.160.161,state:Tehran Province,label:Iran - 78.39.160.161,timestamp:2024-09-16 09:58:52
latitude:35.74216,longitude:52.12752,destinationhost:honeypot-vm,username:administrator,sourcehost:78.39.160.161,state:Tehran Province,label:Iran - 78.39.160.161,timestamp:2024-09-16 09:58:51
latitude:35.74216,longitude:52.12752,destinationhost:honeypot-vm,username:administrator,sourcehost:78.39.160.161,state:Tehran Province,label:Iran - 78.39.160.161,timestamp:2024-09-16 09:58:49
```

Running script / selection. Press Ctrl+Break to stop. Press Ctrl+B to break into debugger.

- 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.


Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
Home > Log Analytics workspaces > law-honeypot1
law-honeypot1 | Tables
Log Analytics workspace
For the list of tables supporting ingestion-time transformatic
Create
Delete
Filter by name
Type : All

Showing 111 results

Table name	Type
EmailPostDeliveryEvents	Azure table
EmailUrlInfo	Azure table
Event	Azure table
FAILED_RDP_GEO_CL	Custom tabl
GCPAuditLogs	Azure table

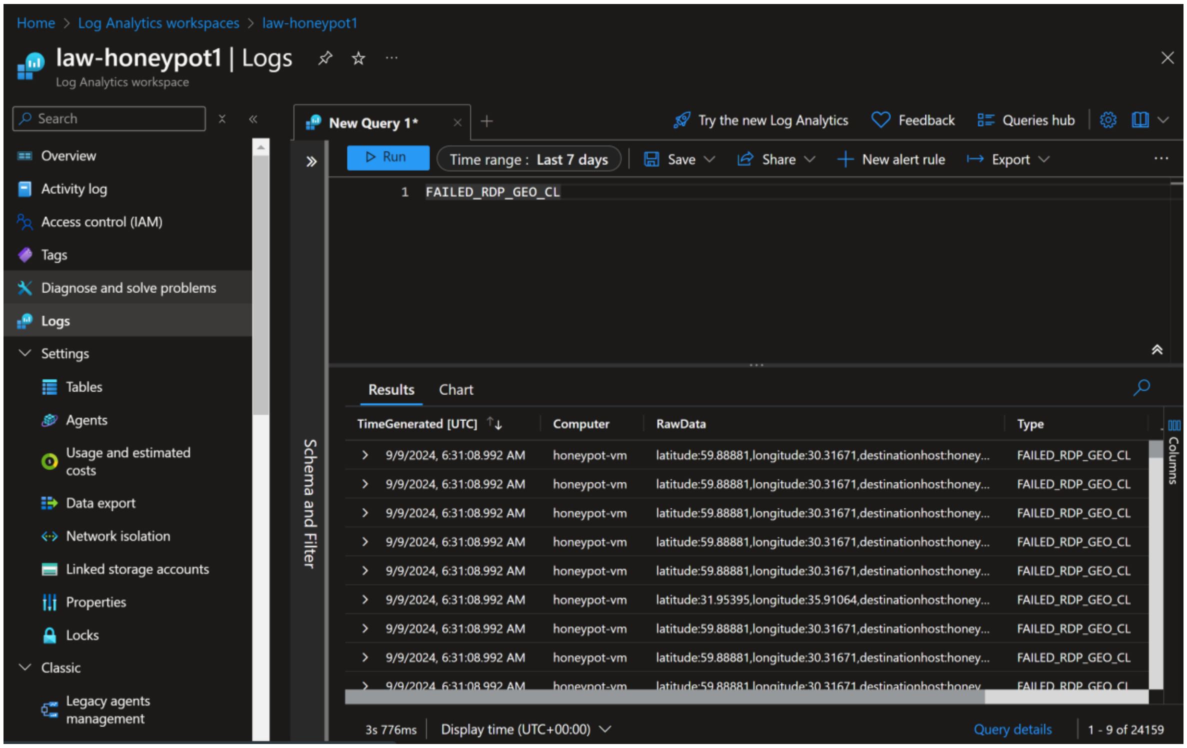
FAILED_RDP_GEO_CL

Description

Description

Collection paths

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

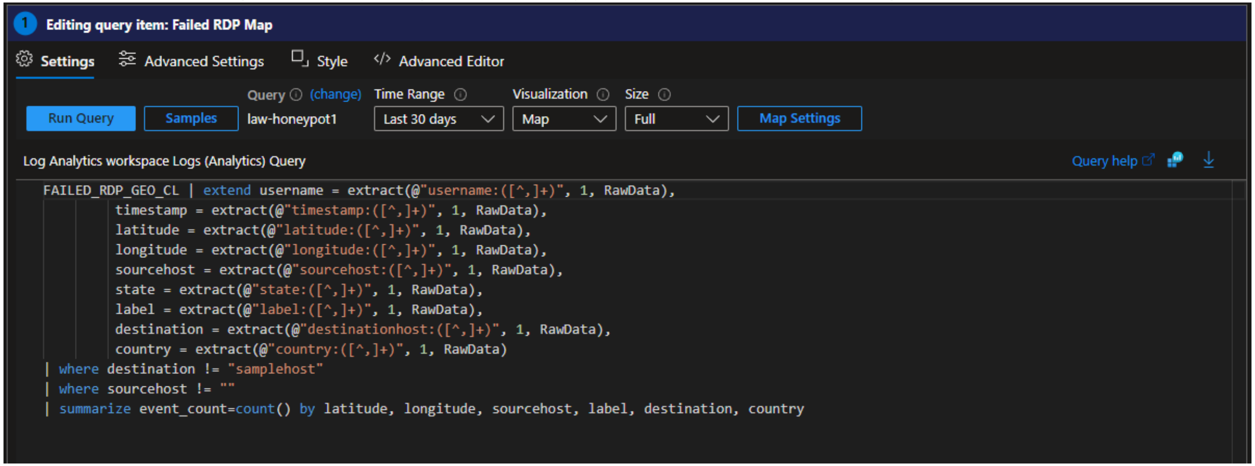


```
FAILED_RDP_GEO_CL
```

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

3s 776ms Display time (UTC+00:00) Query details 1 - 9 of 24159

- 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.



Query (change) law-honeypot1

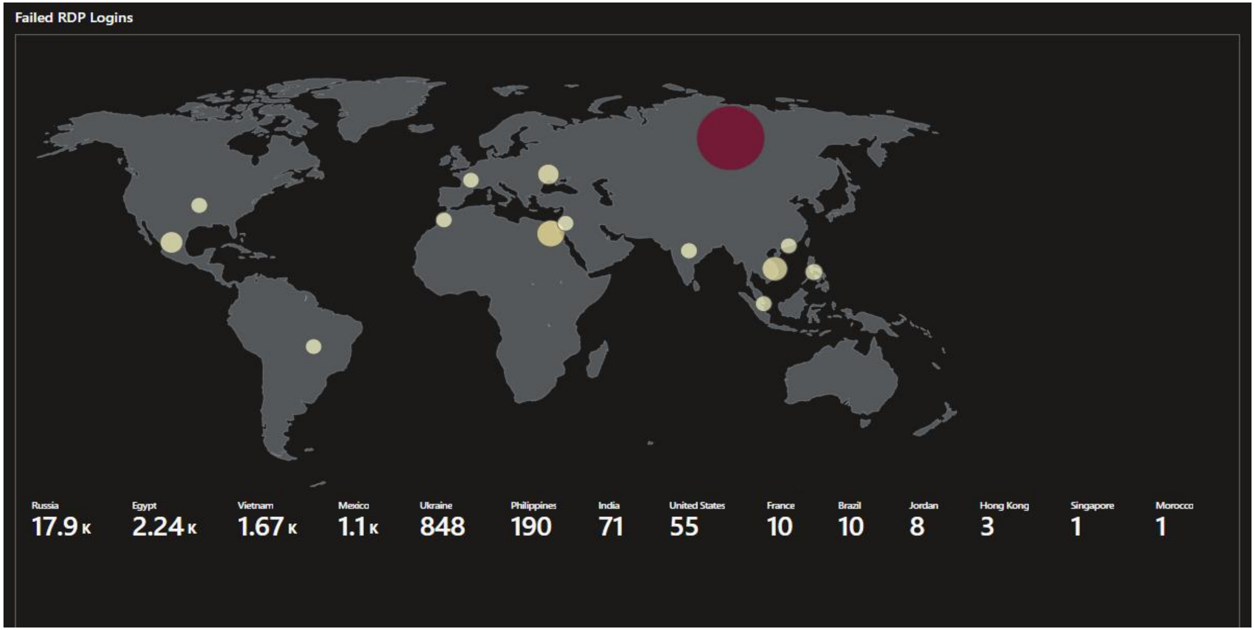
Time Range: Last 30 days

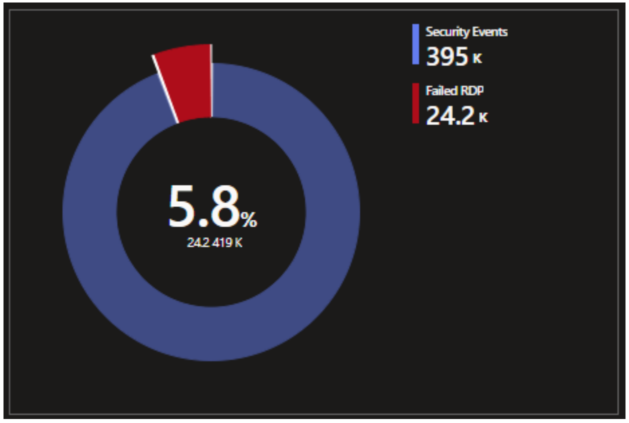
Visualization: Map

Size: Full

Log Analytics workspace Logs (Analytics) Query

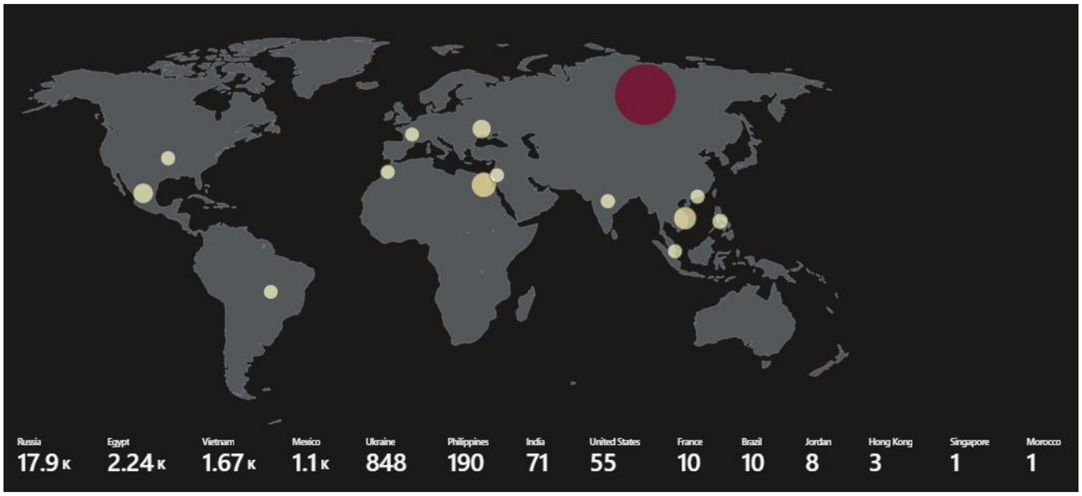
```
FAILED_RDP_GEO_CL | extend username = extract(@"username:([^,]+)", 1, RawData),
         timestamp = extract(@"timestamp:([^,]+)", 1, RawData),
         latitude = extract(@"latitude:([^,]+)", 1, RawData),
         longitude = extract(@"longitude:([^,]+)", 1, RawData),
         sourcehost = extract(@"sourcehost:([^,]+)", 1, RawData),
         state = extract(@"state:([^,]+)", 1, RawData),
         label = extract(@"label:([^,]+)", 1, RawData),
         destination = extract(@"destinationhost:([^,]+)", 1, RawData),
         country = extract(@"country:([^,]+)", 1, RawData)
| where destination != "samplehost"
| where sourcehost != ""
| summarize event_count=count() by latitude, longitude, sourcehost, label, destination, country
```





- 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).