

Fortifying the Virtual Perimeter: Honeypots and SIEM in Action

Justin Warren - Farah Costanza - Justin Chmura

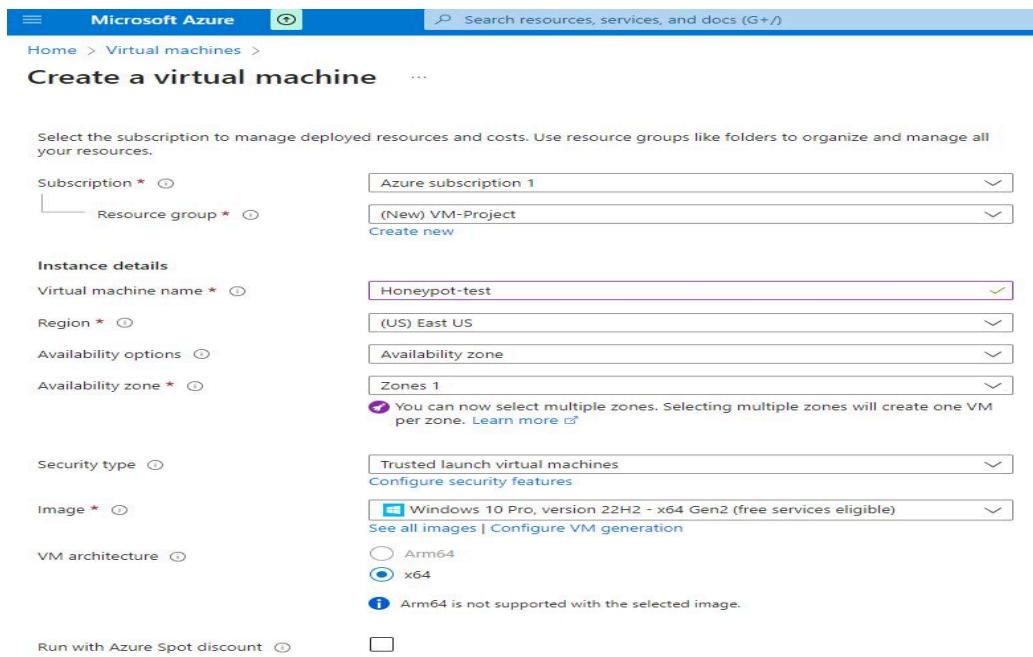
February 2024

* * *

As technology evolves and interconnectivity deepens, fortifying the virtual perimeter becomes not just a defensive tactic but a proactive stance; essential for ensuring the confidentiality, integrity, and availability of digital assets in our increasingly evolving digital world. In this project, we go over two very important parts of our virtual perimeter, a Honeypot and an SIEM tool. Our goal is to set up a SIEM framework around a vulnerable machine to display visually clear event data, including failed login attempts. Below will be a step-by-step process on how we accomplished our goal and what tools we used along the way.

* * *

Getting started, we used Microsoft Azure, a cloud computing platform run by Microsoft to first set up the honeypot virtual machine. First, navigate to “Create a virtual machine”. Create a new resource group, in this case, we named it VM-Project, and named our VM “Honeypot-test”. The region, operating system, and other settings are located in the screenshot below.



In the next step of creating the VM, we need to configure the size, and inbound ports and set our username and password. As shown below, we used the size “Standard_D2s_v3 - 2 vcpus, 8GiB memory, allowing the VM to efficiently run a Windows environment.

Size * ⓘ Standard_D2s_v3 - 2 vcpus, 8 GiB memory (\$70.08/month)

Enable Hibernation (preview) ⓘ ⓘ To enable Hibernation, you must register your subscription. [Learn more](#)

Administrator account

Username * ⓘ Honeypot-test

Password * ⓘ

Confirm password * ⓘ

Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports * ⓘ None Allow selected ports

Select inbound ports * HTTP (80), HTTPS (443), SSH (22), RDP (3389) ⓘ ⓘ All traffic from the internet will be blocked by default. You will be able to change inbound port rules in the VM > Networking page.

Continuing to the networking tab under (NIC networking security group), we select advanced and configure and create a new network security group.

Basics Disks **Networking** Management Monitoring Advanced Tags Review + create

Define network connectivity for your virtual machine by configuring network interface card (NIC) settings. You can control ports, inbound and outbound connectivity with security group rules, or place behind an existing load balancing solution. [Learn more](#)

Network interface

When creating a virtual machine, a network interface will be created for you.

Virtual network * ⓘ (new) Honeypot-test-vnet

Subnet * ⓘ (new) default (10.1.0.0/24)

Public IP ⓘ (new) Honeypot-test-ip

NIC network security group ⓘ None Basic Advanced **Select advanced** ↗

Configure network security group * ⓘ (new) Honeypot-test-nsg ↗ **Create New**

Delete public IP and NIC when VM is deleted ⓘ

Enable accelerated networking ⓘ

Load balancing

You can place this virtual machine in the backend pool of an existing Azure load balancing solution. [Learn more](#)

Load balancing options ⓘ None Azure load balancer Supports all TCP/UDP network traffic, port-forwarding, and outbound flows.

Once Create New is selected, delete the default inbound security rule and configure it as shown below. “Destination port ranges” will be changed to (*) for all. “Priority” should be changed to a low setting, in this case, we chose 100.

Home > Virtual machines >
Create network security group

Name *
Honeypot-test-nsg

Inbound rules (0)
No results.
+ Add an inbound rule

Outbound rules (0)
No results.
+ Add an outbound rule

Add inbound security rule

Honeypot-test-nsg

Source Any
Source port ranges * Any

Destination Any
Destination port ranges * Any

Protocol Any
TCP
UDP
ICMP

Action Allow

Priority * 100
Name * AllowAnyIn
Description

The final step in creating our honeypot is to select “Review + create” and allow the VM time to deploy.

Basics Disks Networking Management Monitoring Advanced Tags **Review + create**

skip to review + create

Cost given below is an estimate and not the final price. Please use [Pricing calculator](#) for all your pricing needs.

Price
1 X Standard D2s v3
by Microsoft
[Terms of use](#) | [Privacy policy](#)

Subscription credits apply ⓘ
0.0960 USD/hr
[Pricing for other VM sizes](#)

TERMS

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. See the [Azure Marketplace Terms](#) for additional details.

Basics

Subscription	Azure subscription 1
Resource group	(new) VM-Project
Virtual machine name	Honeypot-test
Region	East US
Availability options	Availability zone
Availability zone	1
Security type	Trusted launch virtual machines

select create

Enable secure boot

Create

< Previous

Next >

Download a template for automation

In the next step of this process we need to create a Log Analytics Workspace. Navigate to “Create Log Analytics Workspace” and select the resource group that we made in the previous step. Create the workspace and allow time to deploy.

Create Log Analytics workspace ... **Create new Log Analytics Workspace**

Basics Tags Review + Create

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](#)

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 * ⓘ

Resource group * ⓘ select your resource group

[Create new](#)

Instance details

Name * ⓘ

Region * ⓘ

Create

[Review + Create](#) [« Previous](#) [Next : Tags >](#)

The next step is to connect our honeypot VM to our Log Analytics Workspace, to do so, in the newly created workspace navigate down the list of tabs to “Virtual Machine” as shown below.

law-honeypot Log Analytics workspace

Search Delete

The Log Analytics agents (MMAOMS) 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 JSON View

Resource group (move) : vm-project	Workspace Name : law-honeypot
Status : Active	Workspace ID : 82da213c-cc67-4522-8b3e-64024bd3b665
Location : East US	Pricing tier : Pay-as-you-go
Subscription (move) : Azure subscription 1	Access control mode : Use resource or workspace permissions
Subscription ID : e3c73f90-433a-4e1d-9f61-b7bfeec303de	Operational issues : OK
Tags (edit) : Add tags	

[Get Started](#) Recommendations

Get started with Log Analytics

Log Analytics collects data from a variety of sources and uses a powerful query language to give you insights into the operation of your applications and resources. Use Azure Monitor to access the complete set of tools for monitoring all of your Azure resources.

1 Connect a data source **2 Configure monitoring solutions** **3 Monitor workspace health**

Select one or more data sources to connect to the workspace

Azure virtual machines (VMs)
Windows and Linux Agents management
Storage account log
System Center Operations Manager

[View solutions](#)

Useful links

Documentation site
Community

Honeypot-test

Virtual machine

Connect Disconnect Refresh

Not connected

select your VM and select connect and allow connection

Status
Not connected

Workspace Name
None

Message
VM is not connected to Log Analytics.

After we have successfully connected our VM to our Log Analytics Workspace, we then need to configure some settings in Microsoft Defender. As shown below, navigate to “Microsoft Defender for Cloud” and select the “Environment settings” tab.

Microsoft Defender for Cloud | Overview

Showing subscription 'Azure subscription 1'

Search Subscriptions What's new

You may be viewing limited information. To get tenant-wide visibility, click here →

General

- Overview
- Getting started
- Recommendations
- Attack path analysis
- Security alerts
- Inventory
- Cloud Security Explorer
- Workbooks
- Community
- Diagnose and solve problems

Cloud Security

- Security posture
- Regulatory compliance
- Workload protections
- Data security
- Firewall Manager
- DevOps security

Management

- Environment settings
- Security solutions
- Workflow automation

Navigate to Defender for Cloud

Security posture

1 Azure subscriptions 5 Assessed resources 6 Active secure score recommendations

6/6 Unassigned secure score recommendations 0/0 Overdue secure score recommendations 0 Attack paths

Secure score: 65% (Azure 65%, AWS, GCP)

Action required: update the SQL autoprovisioning settings

Update your SQL agent a SQL VMs and Arc-enable Monitoring Agent deprec machines plan is migratin Agent autoprovining p all subscriptions to contin

Take Action | Learn more

Upgrade to new Defender CSPM

Defender Cloud Security I enhanced posture capabili graph to help identify, pri is available in addition to capabilities turned on by

Click here to upgrade >

Defender for Cloud community

Join the Defender for Cloud

By selecting the lab that we have created, we can then configure our settings under “Defender plans”. Select the proper settings and save as pictured below.

After turning servers and Foundational CSPM on and SQL off, save it

Microsoft Defender plans will apply to: 0 Azure and 0 non-Azure resources reporting to this workspace

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

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

On the same page, navigate to the “Data collection” tab select all events and save.

Select all events and save at the top

All Events

Store additional raw data - Windows security events

To help audit, investigate, and analyze threats, you can collect raw events, logs, and additional security data and save it to your Log Analytics workspace.

Select the level of data to store for this workspace. Charges will apply for all settings other than “None”.

Common

Minimal

None

The next tool we are going to configure is Microsoft Sentinel, our SIEM environment. Simply navigate to “Microsoft Sentinel”, select create and add, and add the Log Analytics Workspace previously created. After doing so, it should look like the screenshot below.

Create & add your workspace to Sentinel

Selected workspace: law-honeypot

General

- Overview
- Logs
- News & guides
- Search

Threat management

- Incidents
- Workbooks
- Hunting
- Notebooks
- Entity behavior
- MITRE ATT&ACK (Preview)

Content management

- Content hub
- Repositories (Preview)
- Community

Configuration

- Workspace manager (Preview)
- Data connectors
- Analytics
- Watchlist
- Automation

Events and alerts over time

Incidents by status

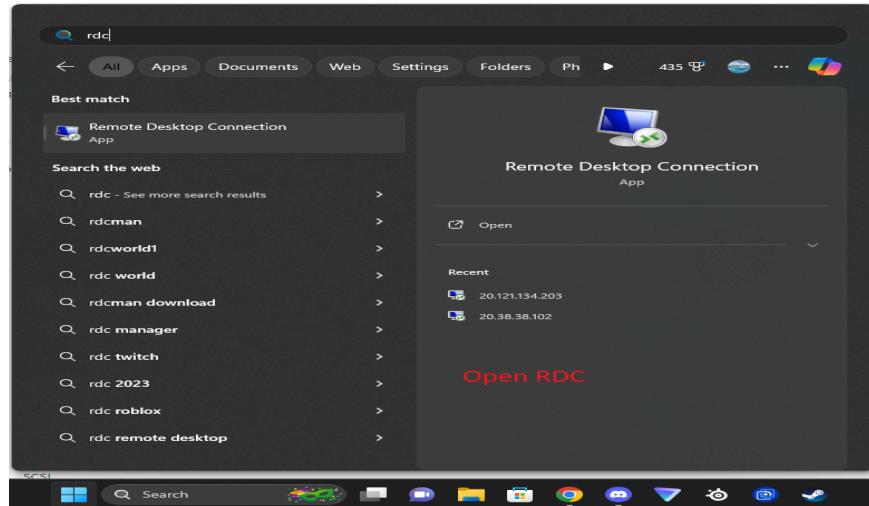
Potential malicious events

Recent incidents

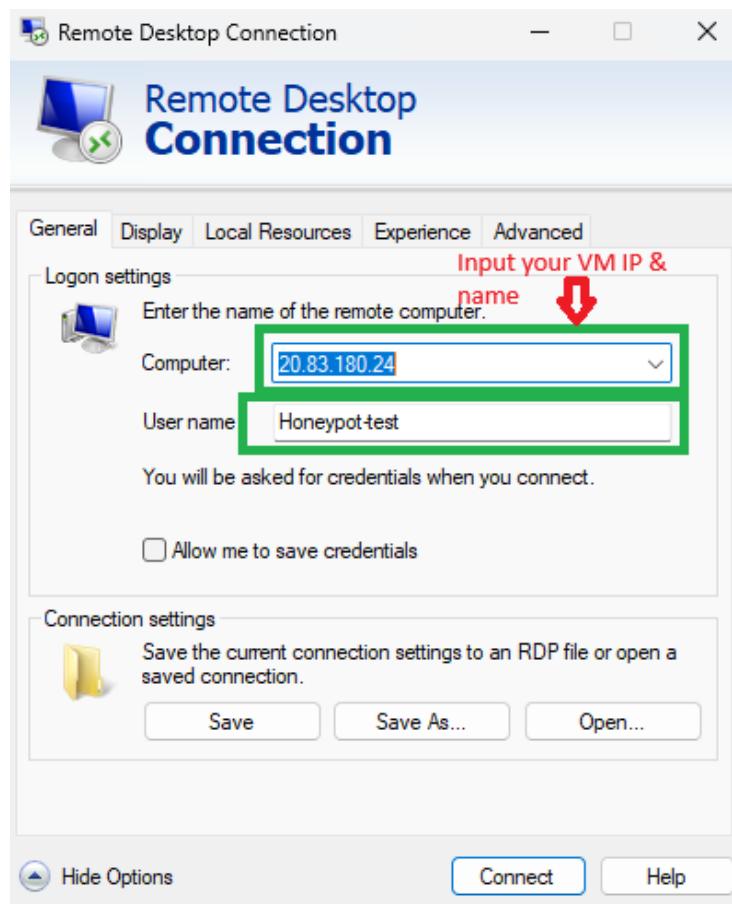
Data source anomalies

Democratize ML for your SecOps

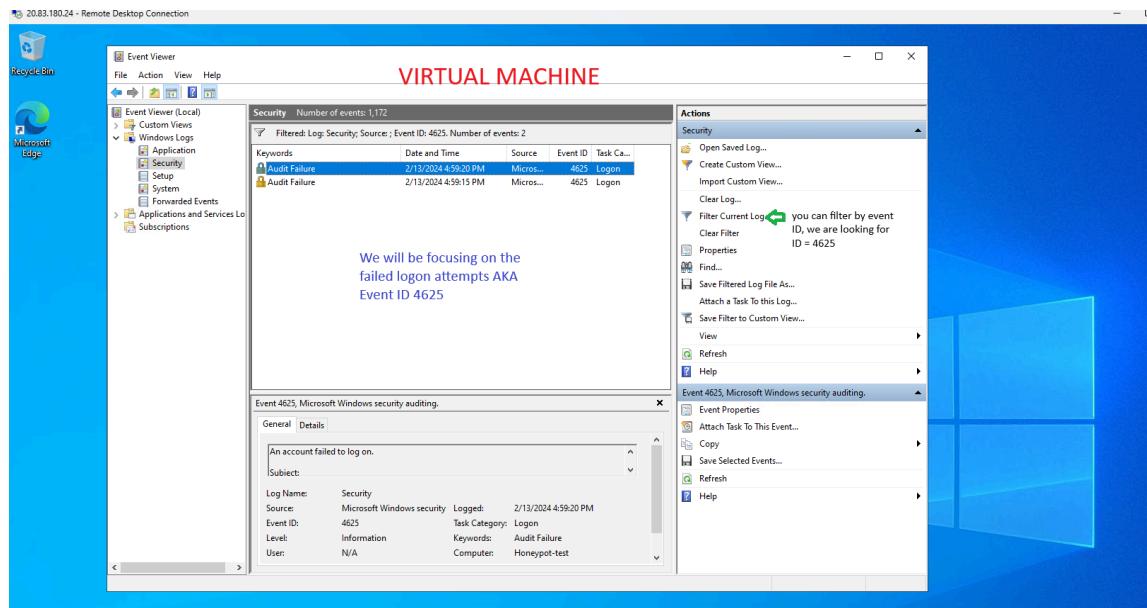
Next, we need to establish a remote desktop connection between our local and virtual machines. To do that, we need to navigate to RDC (Remote Desktop Connection) in our Windows search bar, as we are using a Windows VM.



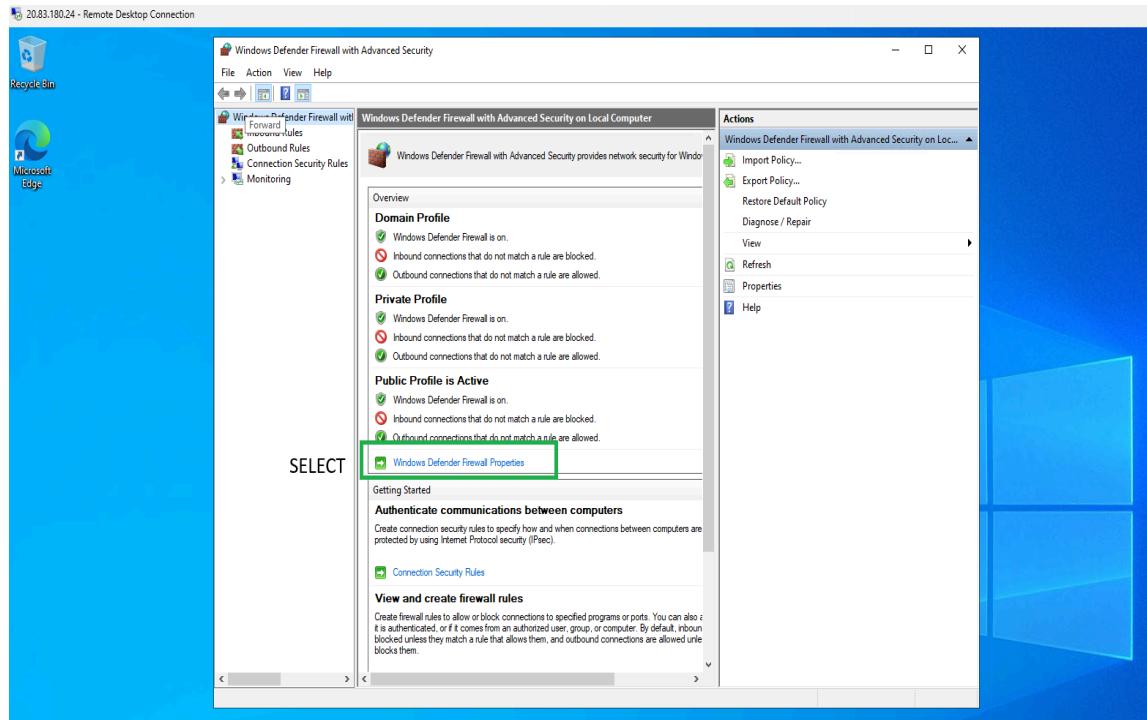
Once opened we input our virtual machine's public IPv4 address and Username we set earlier to establish a connection. You can purposefully input the wrong credentials on the login screen to see a failed attempt in the Event Viewer, we will talk about that shortly.



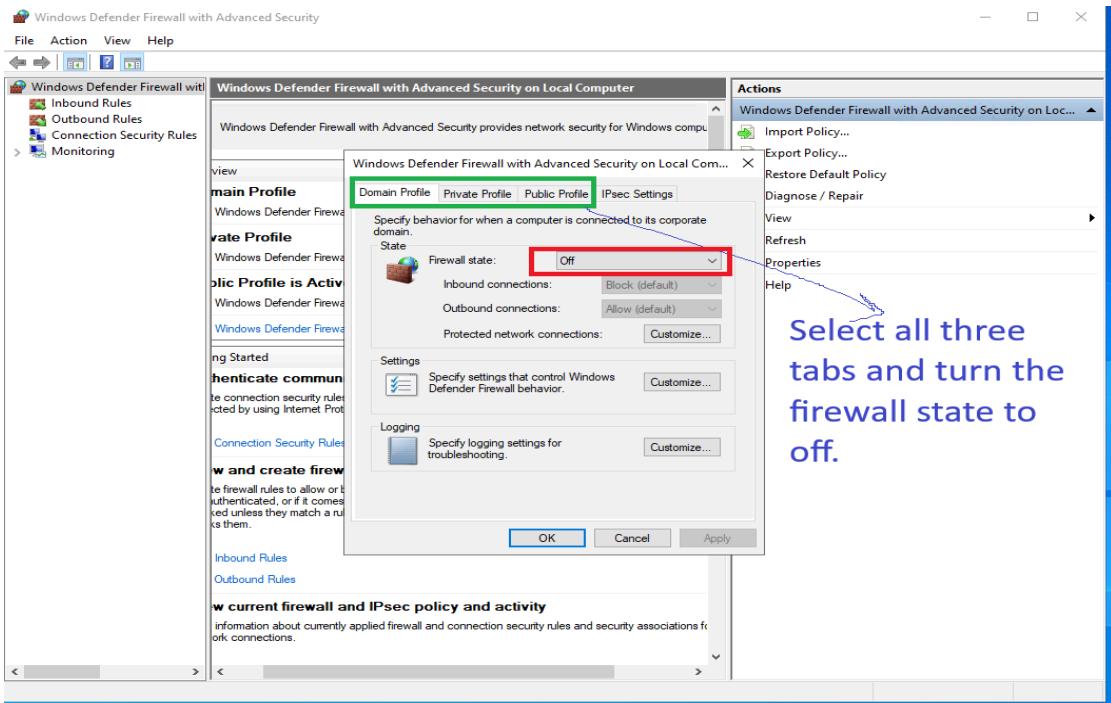
In our virtual machine, navigate to Event Viewer and filter the logs as shown below. We will focus on the Event ID (4625), which will show us all the failed login attempts.



The next step is to disable our VM's firewall to allow traffic through. In the VM, navigate to "WF-MSC" in the Windows search bar, and launch it. We then need to start disabling the firewall, we do so by entering the "Windows Defender Firewall Properties".



Select all three tabs, switch the Firewall state to off, and select ok. We then can proceed to PowerShell.



In the next step, we need to create a custom log, but before we do that we need to retrieve the script we will use to create our custom log, you can find that here:

https://github.com/AnastasiaCoskuner/Sentinel-Lab/blob/main/Custom_Security_Log_Exporter.ps1

```

# Get API key from here: https://ipgeolocation.io/
$API_KEY = "d468084efedf42b9828f5f155e41a457"
$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
$logFile_Folder = "Failed Logon"
$QueryList = 
<Query Id="0" Path="Security">
    <Select Path="Security">
        *System[EventID=4625]]
    </Select>
</Query>
</QueryList>
`#<br>

# 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"
#<br>
function write-SampleLog() {
    #Latitude:47.91542,Longitude:-120.09390,destinationHost:samplehost,username:fireuser,sourceHost:20.16.97.222,state:Mississippi,country:United States,label:United States - 20.16.97.222,timestamp:2021-10-26 09:46:20+00:00
    #Latitude:-22.98900,longitude:-47.48000,destinationHost:samplehost,username:linus,sourceHost:30.199.228.49,state:Paulista,country:Brazil,label:Brazil - 20.199.228.49,timestamp:2021-10-26 09:46:20+00:00
    #Latitude:33.99762,longitude:-77.03350,destinationHost:samplehost,username:CONTRABAND,sourceHost:180.205.155.74,state:Washington,country:United States,label:United States - 180.205.155.74,timestamp:2021-10-26 09:46:20+00:00
    #Latitude:48.71045,longitude:-76.00774,destinationHost:samplehost,username:ADMINISTRATOR,sourceHost:182.50.242.216,state:New York,country:United States,label:United States - 77.45.247.218,timestamp:2021-10-26 09:46:20+00:00
    #Latitude:33.99762,longitude:-6.84757,destinationHost:samplehost,username:AUROUSER,sourceHost:182.50.242.216,state:Rabat-Sale-Kenitra,country:Morocco,label:Morocco - 182.50.242.216,timestamp:2021-10-26 09:46:20+00:00
    #Latitude:5.32558,longitude:108.28595,destinationHost:samplehost,username:Test,sourceHost:42.1.62.34,state:Penang,country:Malaysia,label:Malaysia - 42.1.62.34,timestamp:2021-10-26 11:04:45+00:00
    #Latitude:41.05722,longitude:28.84936,destinationHost:samplehost,username:AUROUSER,sourceHost:176.235.196.111,state:Istanbul,country:Turkey,label:Turkey - 176.235.196.111,timestamp:2021-10-26 11:04:47+00:00
    #Latitude:55.87925,longitude:37.54691,destinationHost:samplehost,username:Tst,sourceHost:87.251.67.98,state:n/a,country:Russia,label:Russia - 87.251.67.98,timestamp:2021-10-26 12:13:45+00:00
}

```

Once you have the script needed, there is one thing we need before running that script and that is an API Key. We use the site <https://ipgeolocation.io/> to generate our key.



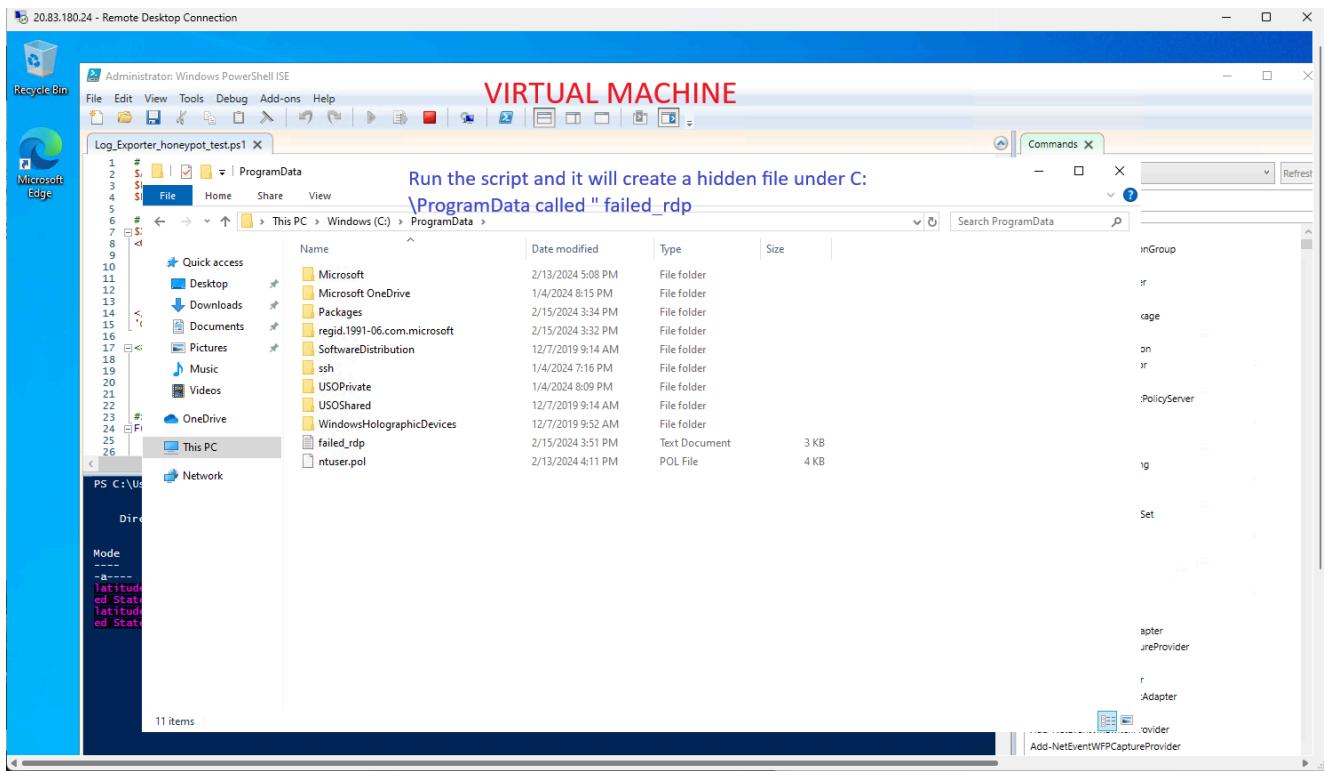
On the VM, open PowerShell ISE and copy and paste the script acquired previously. Under the "\$API_KEY" section, input your API Key from the ipgeolocation site, and select “run” to verify everything is working properly. Be sure to save this to your desktop.

A screenshot of a Windows PowerShell ISE window titled "Untitled1.ps1". The code in the editor is as follows:

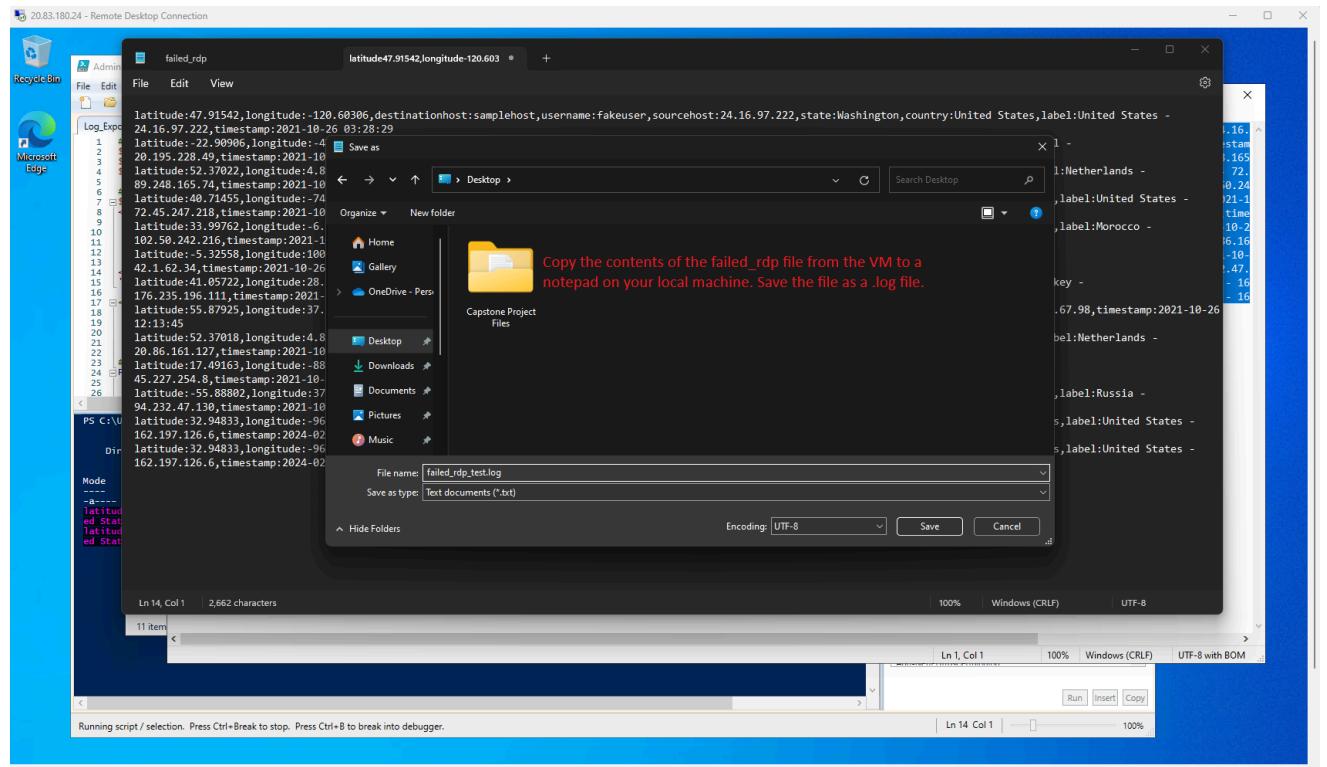
```
1 # Get API key from https://ipgeolocation.io/
2 $API_KEY = "bb2a130a38064d4fa2956b59c630ed8d"
3 $LOGFILE_NAME = "C:\Temp\Op.log"
4 $LOGFILE_PATH = "C:\ProgramData\$LOGFILE_NAME"
5
6 # This filter will be used to filter failed RDP events from Windows Event Viewer
7 $XMLFilter = @'
8 <QueryList>
9   <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,coun
26 }
```

The line "Input your API Key from ipgeolocation.io" is highlighted in red. The PowerShell ISE interface includes a toolbar, a file menu, and a command palette on the right side.

After running the script in PowerShell ISE, we then need to locate the hidden file it creates. This file is located at C:\ProgramData\failed_rdp. We will need this to create our custom log.



The next step is to copy the “failed_rdp” file from our VM to our local machine. We copied the file's contents from our VM and pasted it into a notepad file on our local machine, saving it as a .log file named “failed_rdp.log”.



Back on our local machine, we need to create our custom log. In Log Analytics Workspace, navigate to Tables, select MMA, and upload the .log file we just created, and set the full path to the original file in the path section. Review and create and allow time for custom log to generate.

Microsoft Azure Upgrade

Search resources, services, and docs (G+)

Dashboard > Log Analytics workspaces > law-honeypot | Tables >

Create a custom log

Sample Record delimiter Collection paths Details 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	

Back in Log Analytics workspace under "Tables" select MMA based and input the failed_rdp.log. Under collection path put the full path to the file stated on our VM.

After our custom log is integrated, we then want to test it with a custom query and make sure the geographical location of failed login attempts into our honeypot or “attacks” are recorded. We can find that custom query here:

https://github.com/AnastasiaCoskuner/Sentinel-Lab/blob/main/query_log. Be sure to start query line #1 with the correct name of the custom log created earlier. Select “Run” to test results.

Home > Log Analytics workspaces > Honeypot-logs

Honeypot-logs | Logs ...

New Query 1+ +

Input custom query under the cuture log we created.

```

1 FAILED_RDP_WITH_GEO_CL
2 |extend username = extract(@"username:([^,]+)", 1, RawData),
3 |extend timestamp = extract(@"timestamp:([^,]+)", 1, RawData),
4 |extend latitude = extract(@"latitude:([^,]+)", 1, RawData),
5 |extend longitude = extract(@"longitude:([^,]+)", 1, RawData),
6 |extend sourcehost = extract(@"sourcehost:([^,]+)", 1, RawData),
7 |extend state = extract(@"state:([^,]+)", 1, RawData),
8 |extend label = extract(@"label:([^,]+)", 1, RawData),
9 |extend destination = extract(@"destinationhost:([^,]+)", 1, RawData),
10 |extend country = extract(@"country:([^,]+)", 1, RawData)
11 |where destination != "samplehost"

```

Run Time range : Last 7 days Save Share New alert rule Export Pin to Format query

Results Chart

timestamp	label	country	state	sourcehost	username	destination	longitude	latitude
> 2024-02-08 19:07:07	Morocco - 160.178.177.107	Morocco	Rabat-Sal��-K��nitra	160.178.177.107	ADMINISTRATOR	honeypot-vm	-6.84793	33.99774
> 2024-02-08 18:02:19	United States - 162.197.126.6	United States	Texas	162.197.126.6	honeypot-vm	honeypot-vm	-96.72985	32.94833
> 2024-02-08 18:01:50	United States - 162.197.126.6	United States	Texas	162.197.126.6	honeypot-vm	honeypot-vm	-96.72985	32.94833
> 2024-02-08 18:01:40	United States - 162.197.126.6	United States	Texas	162.197.126.6	honeypot-vm	honeypot-vm	-96.72985	32.94833
> 2024-02-08 18:43:45	Morocco - 160.178.177.107	Morocco	Rabat-Sal��-K��nitra	160.178.177.107	AZUREUSER	honeypot-vm	-6.84793	33.99774
> 2024-02-08 18:47:27	Morocco - 160.178.177.107	Morocco	Rabat-Sal��-K��nitra	160.178.177.107	STUDENT	honeypot-vm	-6.84793	33.99774
> 2024-02-09 01:02:51	Germany - 80.75.212.43	Germany	Bavaria	80.75.212.43	ADMINISTRATOR	honeypot-vm	10.46410	50.30288
> 2024-02-09 01:40:24	Germany - 80.75.212.43	Germany	Bavaria	80.75.212.43	ADMINISTRATOR	honeypot-vm	10.46410	50.30288
> 2024-02-09 02:33:06	Germany - 80.75.212.43	Germany	Bavaria	80.75.212.43	ADMINISTRATOR	honeypot-vm	10.46410	50.30288
> 2024-02-09 02:55:59	Pakistan - 202.163.105.1	Pakistan	Sindh	202.163.105.1	admin	honeypot-vm	67.02601	24.85488
> 2024-02-09 02:55:59	Pakistan - 202.163.105.1	Pakistan	Sindh	202.163.105.1	admin	honeypot-vm	67.02601	24.85488
> 2024-02-09 02:55:57	Pakistan - 202.163.105.1	Pakistan	Sindh	202.163.105.1	admin	honeypot-vm	67.02601	24.85488
> 2024-02-09 02:55:56	Pakistan - 202.163.105.1	Pakistan	Sindh	202.163.105.1	admin	honeypot-vm	67.02601	24.85488
> 2024-02-09 02:55:55	Pakistan - 202.163.105.1	Pakistan	Sindh	202.163.105.1	admin	honeypot-vm	67.02601	24.85488
> 2024-02-09 02:55:54	Pakistan - 202.163.105.1	Pakistan	Sindh	202.163.105.1	admin	honeypot-vm	67.02601	24.85488

0s 545ms Display time (UTC+00:00) Query details 1 - 15 of 1201

Lastly, as stated in our goal above, we want to display a clear visual of our custom logs' data to show the geographical location of the failed login events. Navigate back to Microsoft Sentinel, open the “Workbooks” tab, and select “Add Workbook”.

The screenshot shows the Microsoft Sentinel interface with the "Workbooks" tab selected in the sidebar. A single workbook titled "Failed-logins" is listed under "My workbooks". The interface includes navigation links like "Search", "Refresh", "Add Workbook", "Guides & Feedback", and sections for "General", "Logs", "Threat management", "Content management", and "More content at Content hub".

In our new workbook, the first thing is to delete the Default Queries, replace them with our custom query, and run that query. Under the visualization tab, select the Map option. Select your desired time range and map size, the bigger the better. Finally, we can now see the geolocations of some possible threats that our logs have captured on a visual map.

The screenshot shows the "New workbook" configuration screen. The top bar has the URL "Home > Microsoft Sentinel | Workbooks > honeypot-logs" and a note: "Delete default queries and add our custom query with Map selected.". The main area shows a "Run Query" button, a dropdown for "Data source" set to "Logs", "Resource type" set to "Log Analytics", "Log Analytics workspace" set to "honeypot-logs", "Time Range" set to "Last 7 days", "Visualization" set to "Map", and "Size" set to "Large". Below this is the "Log Analytics workspace Logs Query" editor containing the following custom query:

```
FAILED_RDP_WITH_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)
```

Below the query is a world map titled "Location of possible threats" with several green dots representing threat locations. One specific dot in Europe is highlighted with a large red circle and a green arrow pointing to it.

In conclusion, throughout this project, we explored two crucial parts of fortifying the virtual perimeter: the implementation of a Honeypot and the utilization of a Security Information and Event Management System (SIEM) tool. Our primary objective was to establish an effective SIEM framework around a vulnerable machine, with a focus on visualizing event data such as failed login attempts. By detailing the step-by-step process in this report, we have not only achieved our goal but also gained valuable insights into the details of deploying and configuring these essential components of our virtual perimeter. As technology grows, and our dependence on digital systems deepens, guarding our virtual perimeter is no longer just a defensive security measure but a proactive necessity.