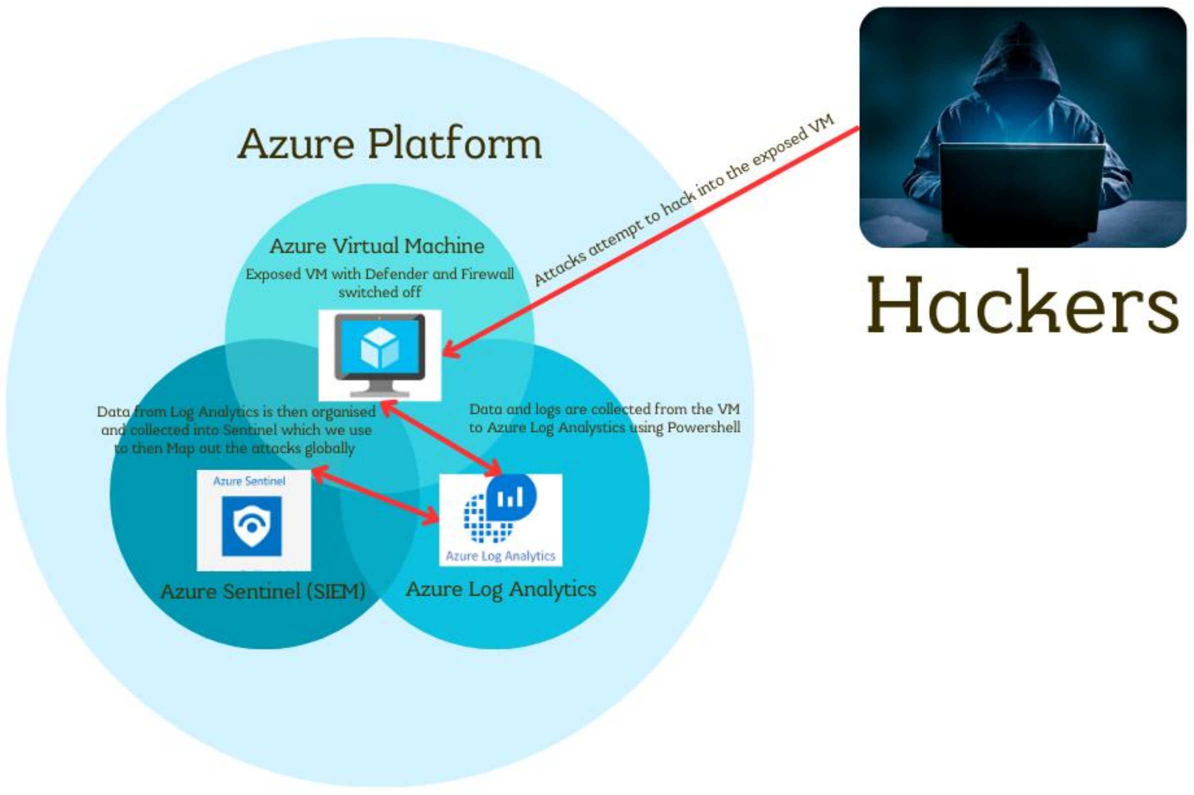
SIEM Tools Project - Azure Sentinel Global Map of Attacks

I've been eager to enhance my knowledge and experience in Azure Sentinel, given its increasing demand and adoption in the UK cybersecurity landscape. However unfortunately, Microsoft and most online IT educational platforms that I use offer very limited training resources or labs for Sentinel as Splunk seems to be the more dominant SIEM tool used within cybersecurity.

So, I therefore took advantage of Microsoft's free 1-month subscription offer and decided to configure Azure Sentinel to simulate real-world scenarios, I connected it to a live VM (virtual machine) functioning as a honeypot/decoy system. This setup allowed me to actively monitor and analyse live cyberattacks, specifically focusing on RDP brute force attacks originating from various global locations. To then enhance this further, I used a custom PowerShell script to fetch geolocation data about the attackers through their IP addresses and was able to visualise it on the Azure Sentinel Map, providing valuable threat intelligence.

This project was both intriguing and challenging as working with Azure proved to be particularly interesting but also frustrating. The Azure platform has undergone significant modifications this year alone, which meant that much of the information I found on forums and other online sources was outdated and irrelevant.

I have created a simple diagram explaining the scenario below and also outlined a step-by-step process for this project. For obvious reasons this isn't going to be very in depth.

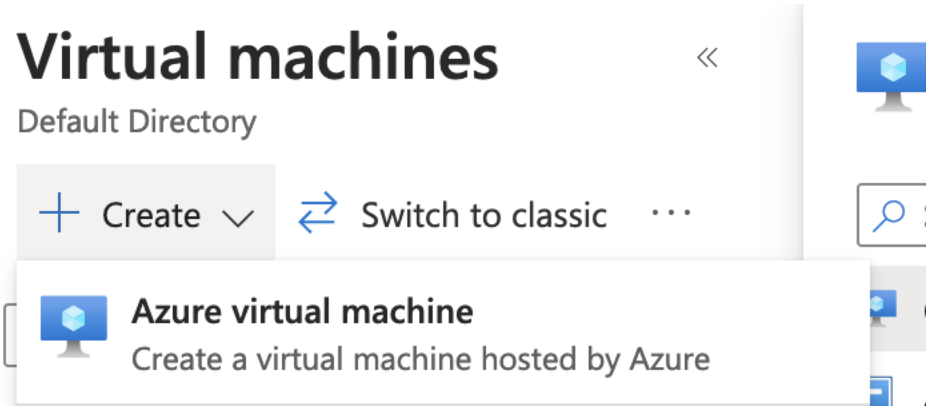

Azure Platform
Azure Virtual Machine
Exposed VM with Defender and Firewall switched off
Attacks attempt to hack into the exposed VM
Hackers
Data from Log Analytics is then organised and collected into Sentinel which we use to then Map out the attacks globally
Data and logs are collected from the VM to Azure Log Analystics using Powershell
Azure Sentinel
Azure Sentinel (SIEM)
Azure Log Analytics
Azure Log Analytics

Once I had signed up for the free month subscription, I was granted access to all of the software that Azure offers. For this project we only needed the following 3.

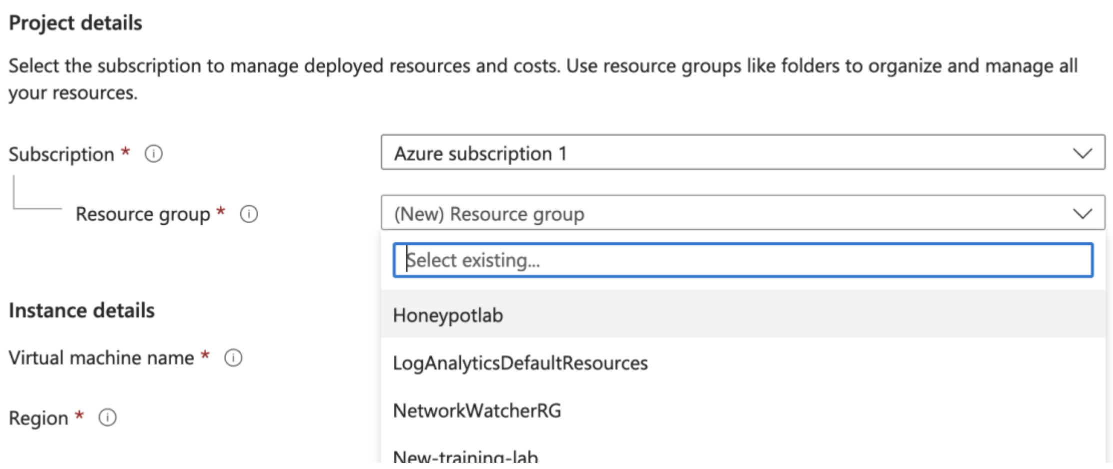
- Virtual Machine
- Sentinel
- Log Analytics

<u>Part 1 – Setting up the Virtual Machine (VM)</u>

After clicking on Virtual Machines within the Azure platform the next task was to create the new VM. Create > Azure virtual machine



From there we arrive to the following, where we need to choose the subscription type, name of the VM but also its location. As you can see, I named this "Honeypot"



Project details

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

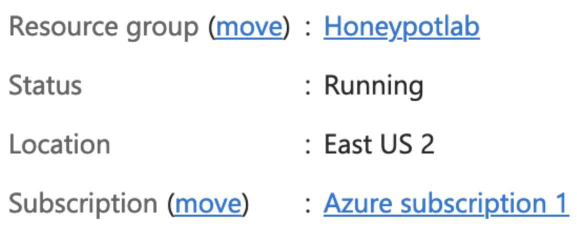
Subscription *	Azure subscription 1
Resource group *	(New) Resource group

Instance details

Virtual machine name *

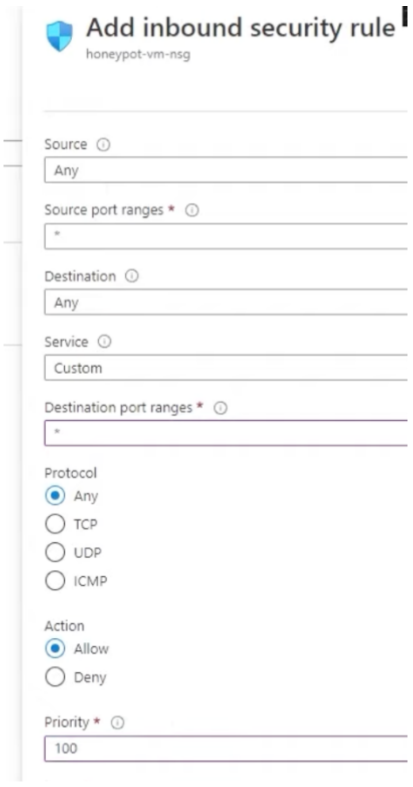
Region *

In regards with location I decided to use the "East US 2", the reason being is that although I'm in the UK this VM is going to be exposed to the whole world to entice attacks and my thinking process was that the US is the most attacked nation on the planet and therefore most likely to produce better results



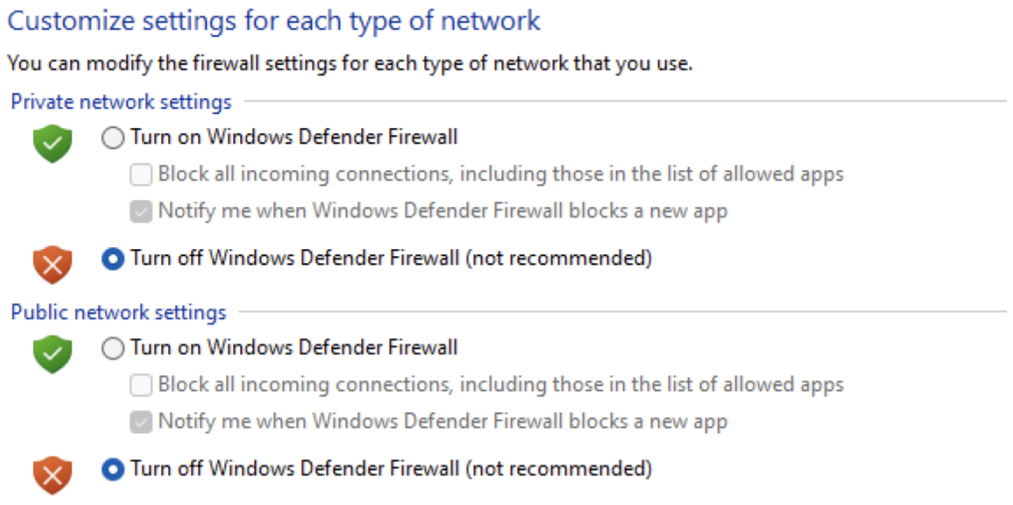
Resource group (move)	: Honeypotlab
Status	: Running
Location	: East US 2
Subscription (move)	: Azure subscription 1

For the security rules I wanted to allow all port ranges and also kept "priority" level down to a minimum


Add inbound security rule
honeypot-vm-nsg

Source: Any
Source port ranges *: *
Destination: Any
Service: Custom
Destination port ranges *: *
Protocol: [✓ Any] TCP UDP ICMP
Action: [✓ Allow] Deny
Priority *: 100

I also changed the Windows Defender Firewall settings as seen below, although I don't in anyway recommend doing this the objective of this project was to entice as many attacks as possible. Obviously normally the idea is to prevent attacks and therefore both these settings would be usually left "On".


Customize settings for each type of network

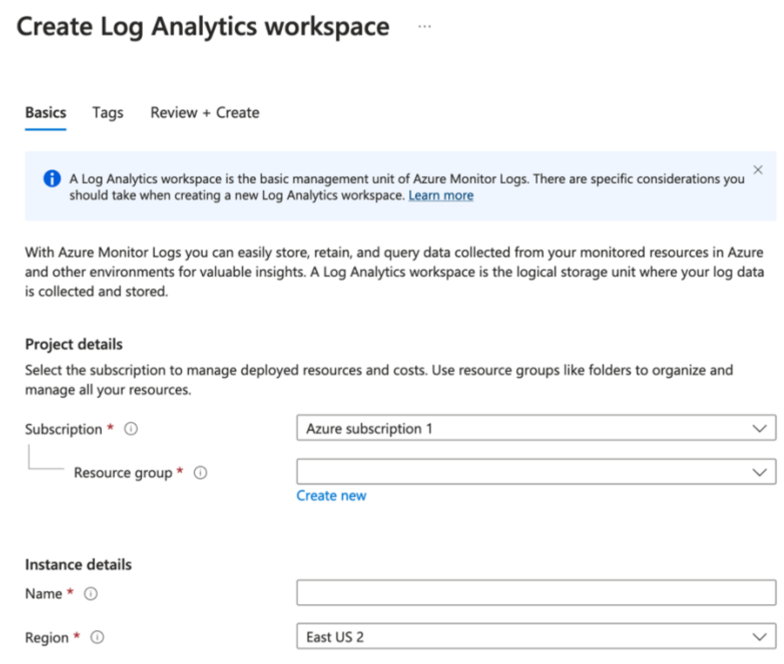
You can modify the firewall settings for each type of network that you use.

Private network settings
- Turn on Windows Defender Firewall
  - Block all incoming connections, including those in the list of allowed apps
  - Notify me when Windows Defender Firewall blocks a new app
- [✓ Turn off Windows Defender Firewall (not recommended)]

Public network settings
- Turn on Windows Defender Firewall
  - Block all incoming connections, including those in the list of allowed apps
  - Notify me when Windows Defender Firewall blocks a new app
- [✓ Turn off Windows Defender Firewall (not recommended)]

Part 2 – Creating a Log Analytics Workspace

Again, fairly straightforward, I used the same resource group as I did in the VM section. Provided a name and also kept the region/location the same.



Create 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.

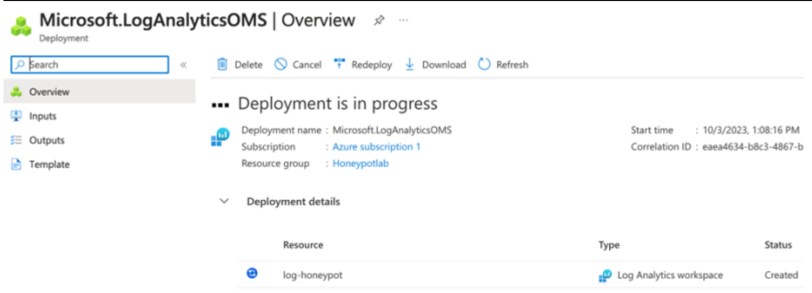
Project details

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

Subscription *	Azure subscription 1
Resource group *	
	Create new

Instance details

Name *	
Region *	East US 2



Microsoft.LogAnalyticsOMS | Overview

Deployment

Deployment is in progress

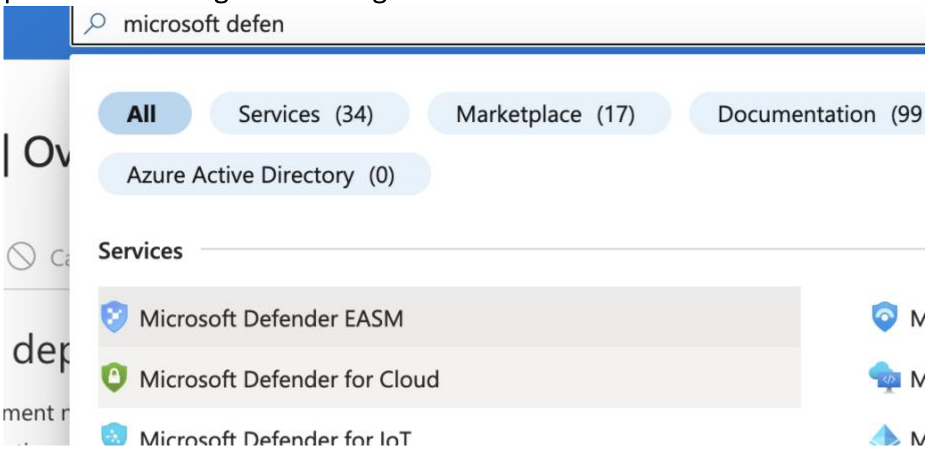
Deployment name : Microsoft.LogAnalyticsOMS
Subscription : Azure subscription 1
Resource group : Honeypotlab

Start time : 10/3/2023, 1:08:16 PM
Correlation ID : eaea4634-b8c3-4867-b

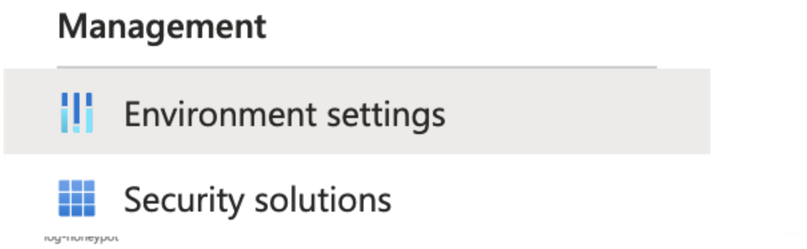
Deployment details

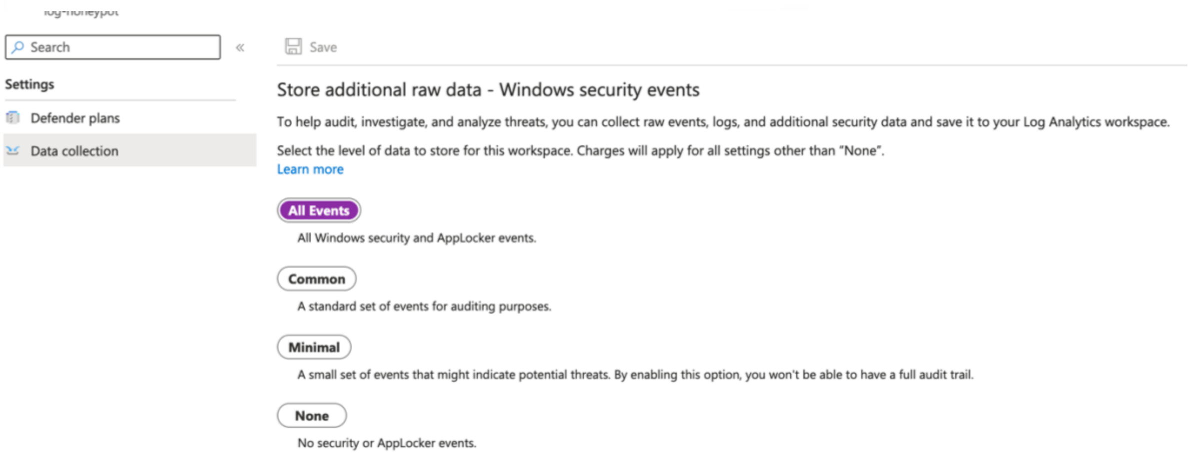
Resource	Type	Status
log-honeypot	Log Analytics workspace	Created

Once done we now have to go into "Microsoft Defender for Cloud". This is so we can enable the ability to gather logs from the VM into the Log Analytics Workspace. This used to be called "Security Center" but this was one of the recent changes which changed the whole process of linking the two together.



From there we go to “Environment settings > Data collection” and I made sure that “All Events” was enabled to make sure that all the raw data and logs were collected.





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".
Learn more

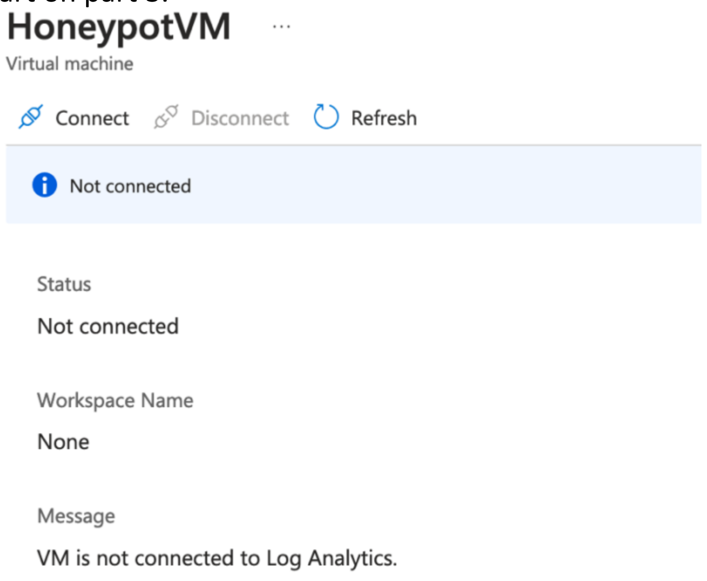
[✓ All Events]
All Windows security and AppLocker events.

Common
A standard set of events for auditing purposes.

Minimal
A small set of events that might indicate potential threats. By enabling this option, you won't be able to have a full audit trail.

None
No security or AppLocker events.

Now we can connect Log Analytics to the VM by clicking on the “Connect” button below. This usually takes a few minutes to go through. Whilst that’s going through, we can make a start on part 3.



Not connected

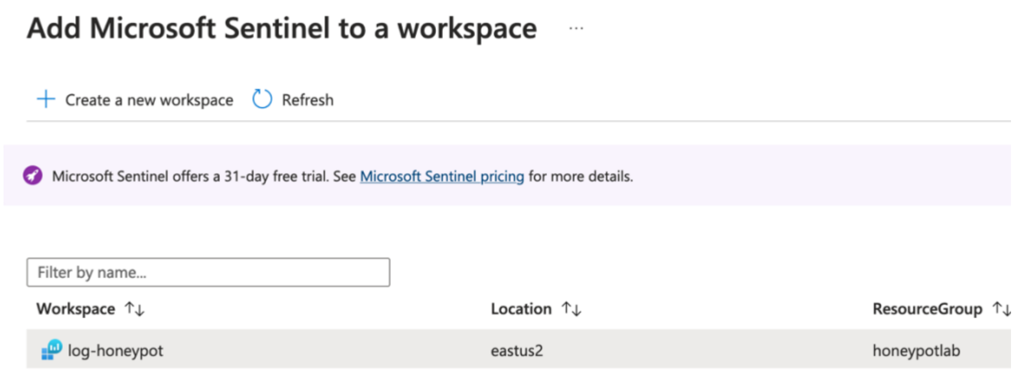
Status
Not connected

Workspace Name
None

Message
VM is not connected to Log Analytics.

Part 3 – Setting up Azure Sentinel

This section is very quick, once we find “Azure Sentinel” in the search bar above. Simply find the Log Workspace that we created earlier in the section below, click on it and then click “Add” to add that to the Sentinel platform



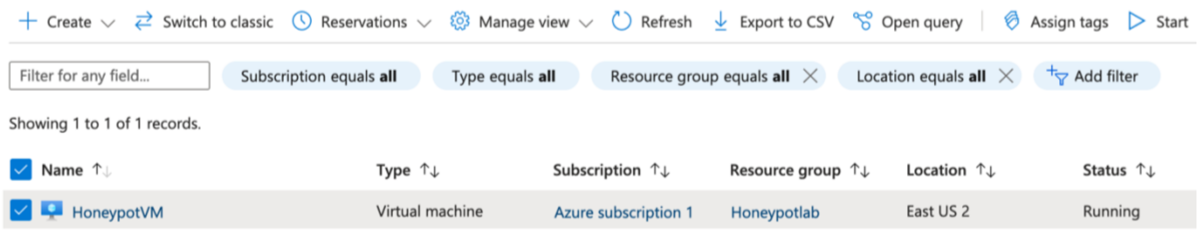
Add Microsoft Sentinel to a workspace

Microsoft Sentinel offers a 31-day free trial. See Microsoft Sentinel pricing for more details.

Workspace	Location	ResourceGroup
log-honeypot	eastus2	honeypotlab

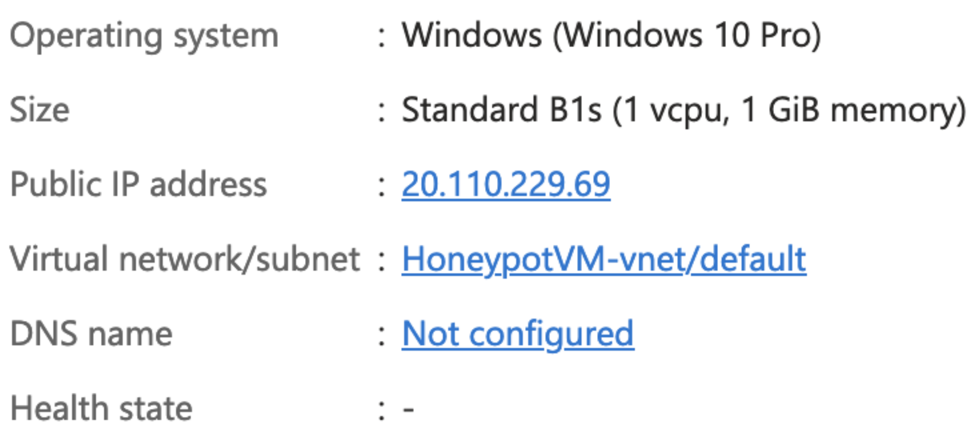
Part 4 – Logging into the new VM via RDP (remote desktop)

To log into the new VM I first needed the IP, so I went back to “Virtual Machines” within Azure and found the IP as seen below



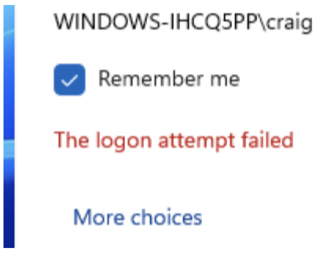
Showing 1 to 1 of 1 records.

Name	Type	Subscription	Resource group	Location	Status
HoneypotVM	Virtual machine	Azure subscription 1	Honeypotlab	East US 2	Running



Operating system	: Windows (Windows 10 Pro)
Size	: Standard B1s (1 vcpu, 1 GiB memory)
Public IP address	: 20.110.229.69
Virtual network/subnet	: HoneypotVM-vnet/default
DNS name	: Not configured
Health state	: -

Using this IP, I was able to log into the VM. I accidently used the wrong username first time round hence why the log in initially failed, however once In I was able to make my way to "Event Viewer"

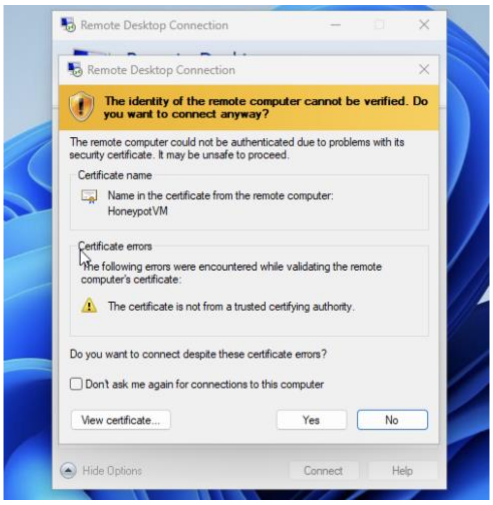


WINDOWS-IHCQ5PP\craig

Remember me

The logon attempt failed

More choices



Remote Desktop Connection

The identity of the remote computer cannot be verified. Do you want to connect anyway?

The remote computer could not be authenticated due to problems with its security certificate. It may be unsafe to proceed.

Certificate name

Name in the certificate from the remote computer:
HoneypotVM

Certificate errors

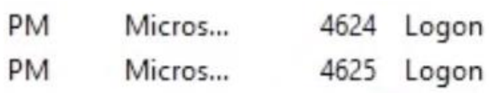
The following errors were encountered while validating the remote computer's certificate:

The certificate is not from a trusted certifying authority.

Do you want to connect despite these certificate errors?

Don't ask me again for connections to this computer

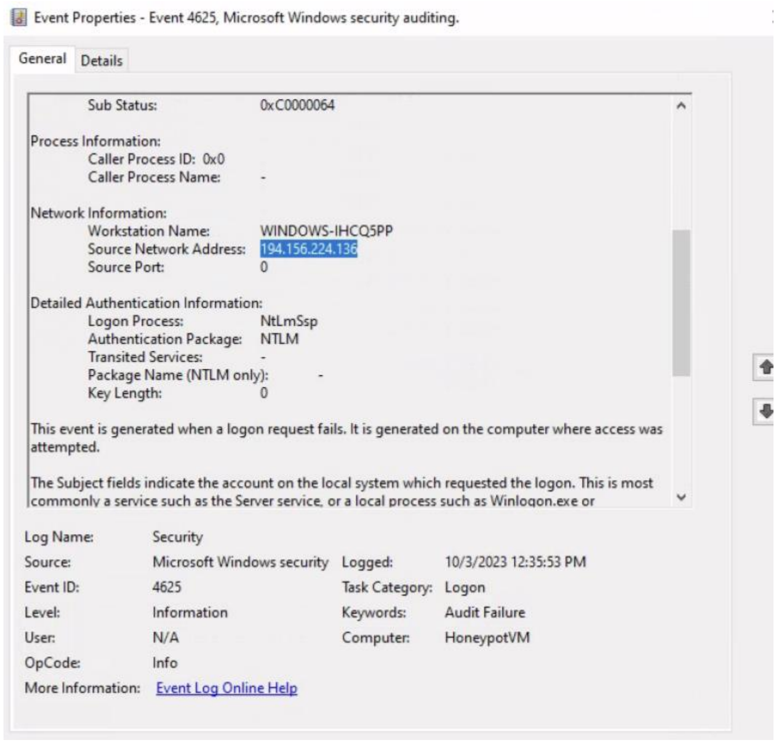
Now the objective of this project is to have our Virtual Machine attacked whilst registering this information and having it marked on our global map on Sentinel. In order to do this, we first need the EventID for "failed logins". We will be using this in the Powershell script to provide all the important information and data from each failed attempt. The Failed Login EventID is 4625



```
PM    Micros...    4624  Logon
PM    Micros...    4625  Logon
```

Here we can see my initial failed attempt registered within Event Viewer. The EventID is as shown above and below as 4625. Furthermore, you can also see some important data being collected, workstation name and IP address. The IP highlighted is the one I used in this failed login attempt. Now unfortunately Event Viewer doesn't provide longitude or latitude data of where the IP originated from however lots of third-party sites offer this type of service.

In this case I used https://app.ipgeolocation.io/



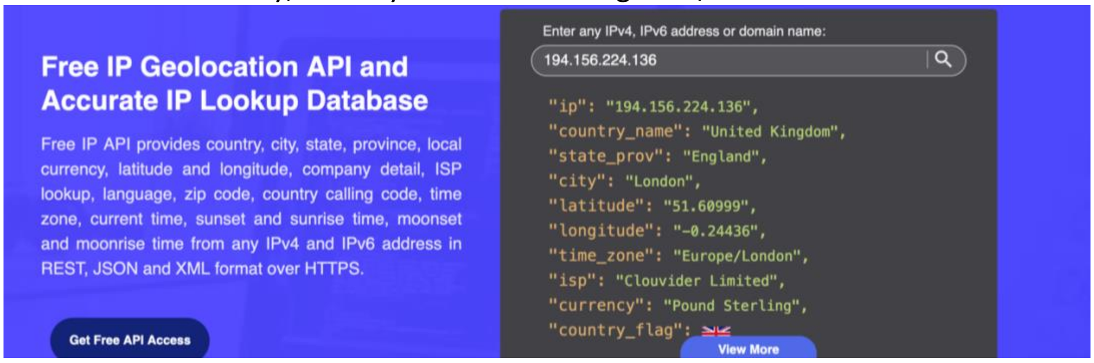
Event Properties - Event 4625, Microsoft Windows security auditing.

```
        Sub Status:                 0xC0000064

Process Information:
        Caller Process ID:  0x0
        Caller Process Name:        -

Network Information:
        Workstation Name:           WINDOWS-IHCQ5PP
        Source Network Address:     194.156.224.136
        Source Port:                0

Detailed Authentication Information:
        Logon Process:              NtLmSsp
        Authentication Package:     NTLM
        Transited Services:         -
        Package Name (NTLM only):       -
        Key Length:                 0

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
```

Log Name:	Security		
Source:	Microsoft Windows security	Logged:	10/3/2023 12:35:53 PM
Event ID:	4625	Task Category:	Logon
Level:	Information	Keywords:	Audit Failure
User:	N/A	Computer:	HoneypotVM
OpCode:	Info		
More Information:	Event Log Online Help		

I copied the IP above and inserted it into https://app.ipgeolocation.io/. You can see the information that was gathered just from that 1 IP address listed below

Here we have the city, country and also the longitude/latitude.



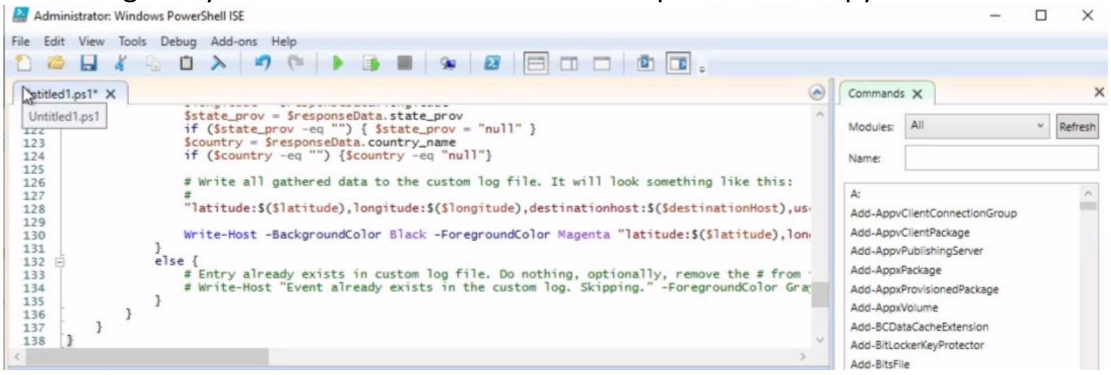
Free IP Geolocation API and Accurate IP Lookup Database

Free IP API provides country, city, state, province, local currency, latitude and longitude, company detail, ISP lookup, language, zip code, country calling code, time zone, current time, sunset and sunrise time, moonset and moonrise time from any IPv4 and IPv6 address in REST, JSON and XML format over HTTPS.

Enter any IPv4, IPv6 address or domain name:
194.156.224.136

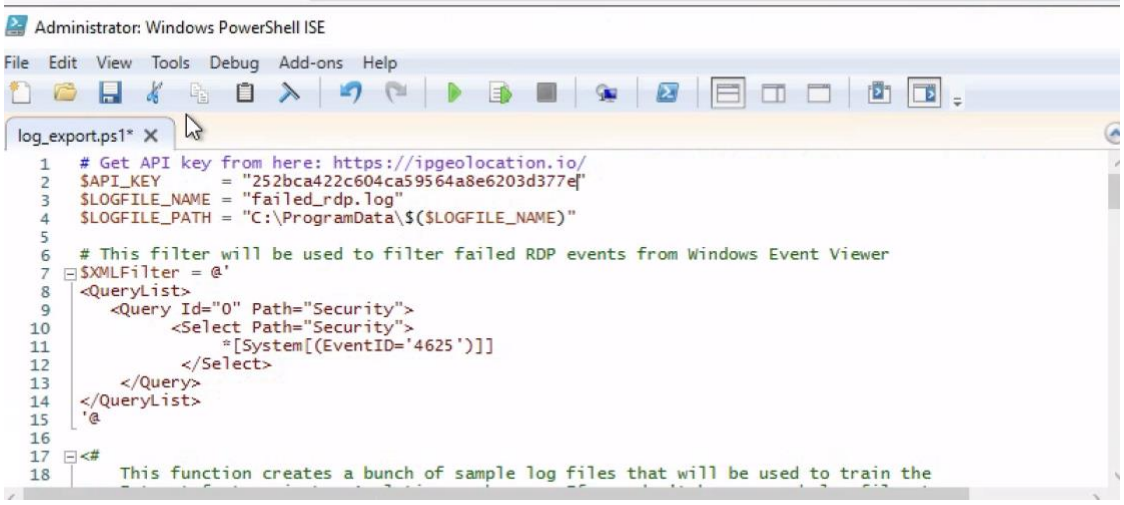
```
"ip": "194.156.224.136",
"country_name": "United Kingdom",
"state_prov": "England",
"city": "London",
"latitude": "51.60999",
"longitude": "-0.24436",
"time_zone": "Europe/London",
"isp": "Clouvider Limited",
"currency": "Pound Sterling",
"country_flag":
```

Now that we have the EventID and understand how the geolocation works, we can open up Powershell ISE. I found a script online which allows us gather this information and export this to Log Analytics. We need to create a new script and then copy it as I've done below.



```
            $state_prov = $responseData.state_prov
            if ($state_prov -eq "") { $state_prov = "null" }
            $country = $responseData.country_name
            if ($country -eq "") {$country -eq "null"}

            # Write all gathered data to the custom log file. It will look something like this:
            #
            "latitude:$($latitude),longitude:$($longitude),destinationhost:$($destinationHost),us

            Write-Host -BackgroundColor Black -ForegroundColor Magenta "latitude:$($latitude),lon
        }
        else {
            # Entry already exists in custom log file. Do nothing, optionally, remove the # from
            # Write-Host "Event already exists in the custom log. Skipping." -ForegroundColor Gra
        }
      }
  }
}
```

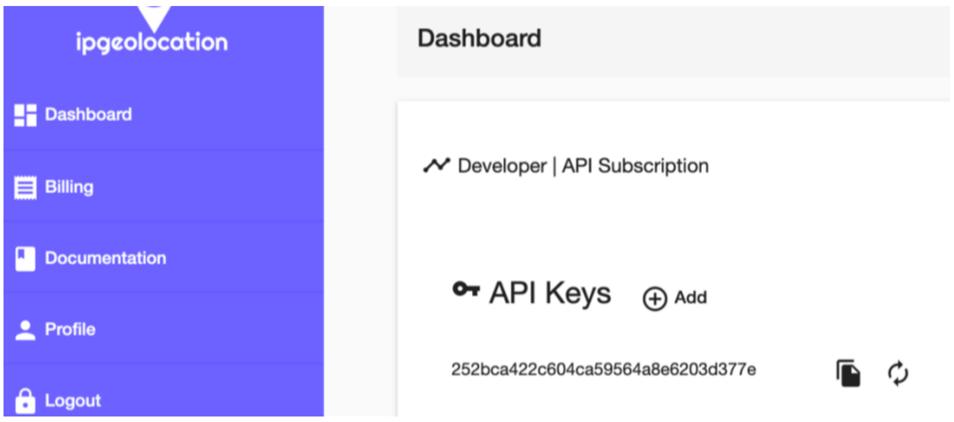
Once done I saved the script as log_export.ps1. Now we can clearly see from the script that a logfile named "failed_rdp.log" will be exported and filed in the directory "C:\ProgramData\" This is the text file that we will be using in the Log Analytics Workspace



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

```
# Get API key from here: https://ipgeolocation.io/
$API_KEY      = "252bca422c604ca59564a8e6203d377e"
$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
```

Furthermore, you can see my API Key listed directly above. This is where all the failed IP's will go through in order to grab that important geographical information which will later be used in Sentinel. In order for this to work you will need to provide your own API Key, you can acquire this by signing up to ipgeolocation and I believe they provide you with a free API for up to 30 days.



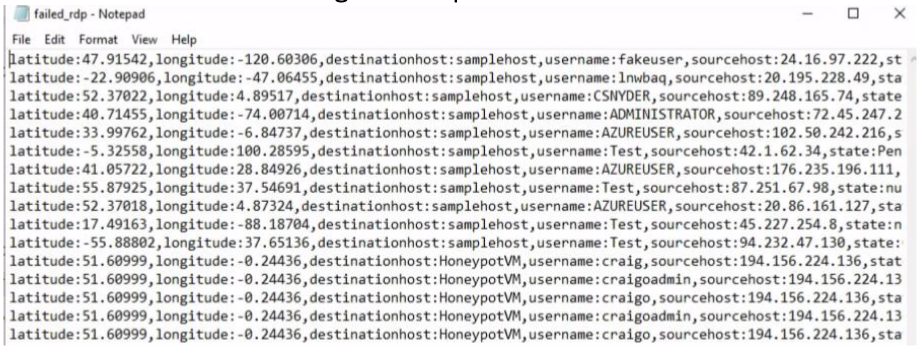
Dashboard

Developer | API Subscription

API Keys

252bca422c604ca59564a8e6203d377e

Heading over to the "failed_rdp.log" we can clearly see that I have already had a number of attempted RDP brute force attacks. We can also see my failed attempts at the bottom of the screen where I was testing the script.



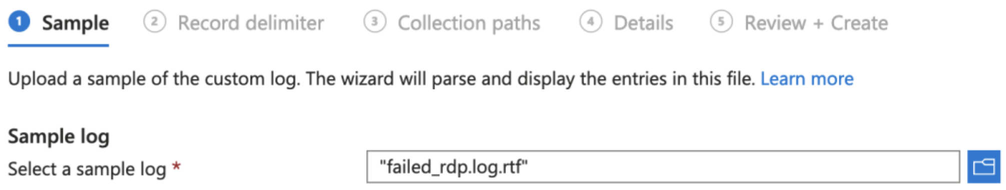
```
latitude:47.91542,longitude:-120.60306,destinationhost:samplehost,username:fakeuser,sourcehost:24.16.97.222,st
latitude:-22.90906,longitude:-47.06455,destinationhost:samplehost,username:lnwbaq,sourcehost:20.195.228.49,sta
latitude:52.37022,longitude:4.89517,destinationhost:samplehost,username:CSNYDER,sourcehost:89.248.165.74,state
latitude:40.71455,longitude:-74.00714,destinationhost:samplehost,username:ADMINISTRATOR,sourcehost:72.45.247.2
latitude:33.99762,longitude:-6.84737,destinationhost:samplehost,username:AZUREUSER,sourcehost:102.50.242.216,s
latitude:-5.32558,longitude:100.28595,destinationhost:samplehost,username:Test,sourcehost:42.1.62.34,state:Pen
latitude:41.05722,longitude:28.84926,destinationhost:samplehost,username:AZUREUSER,sourcehost:176.235.196.111,
latitude:55.87925,longitude:37.54691,destinationhost:samplehost,username:Test,sourcehost:87.251.67.98,state:nu
latitude:52.37018,longitude:4.87324,destinationhost:samplehost,username:AZUREUSER,sourcehost:20.86.161.127,sta
latitude:17.49163,longitude:-88.18704,destinationhost:samplehost,username:Test,sourcehost:45.227.254.8,state:n
latitude:-55.88802,longitude:37.65136,destinationhost:samplehost,username:Test,sourcehost:94.232.47.130,state:
latitude:51.60999,longitude:-0.24436,destinationhost:HoneypotVM,username:craig,sourcehost:194.156.224.136,stat
latitude:51.60999,longitude:-0.24436,destinationhost:HoneypotVM,username:craigoadmin,sourcehost:194.156.224.13
latitude:51.60999,longitude:-0.24436,destinationhost:HoneypotVM,username:craigo,sourcehost:194.156.224.136,sta
latitude:51.60999,longitude:-0.24436,destinationhost:HoneypotVM,username:craigoadmin,sourcehost:194.156.224.13
latitude:51.60999,longitude:-0.24436,destinationhost:HoneypotVM,username:craigo,sourcehost:194.156.224.136,sta
```

Once the script is complete, tested and saved we can then move back into our normal system and on the Azure platform, from there we head to Log Analytics.

<u>Part 5 – Creating Custom Logs</u>

This was by far the most stressful stage of the whole project, initially the process has always been very simple from what I made out however with some of the huge changes that Microsoft made with the recent "update", something ever so simple now is invariably more complicated.

From Log Analytics we go to "Create a custom log" and it is here where we insert the file name of the log that our Powershell script exports our data to within our VM. Scroll up to Part 4 for confirmation.

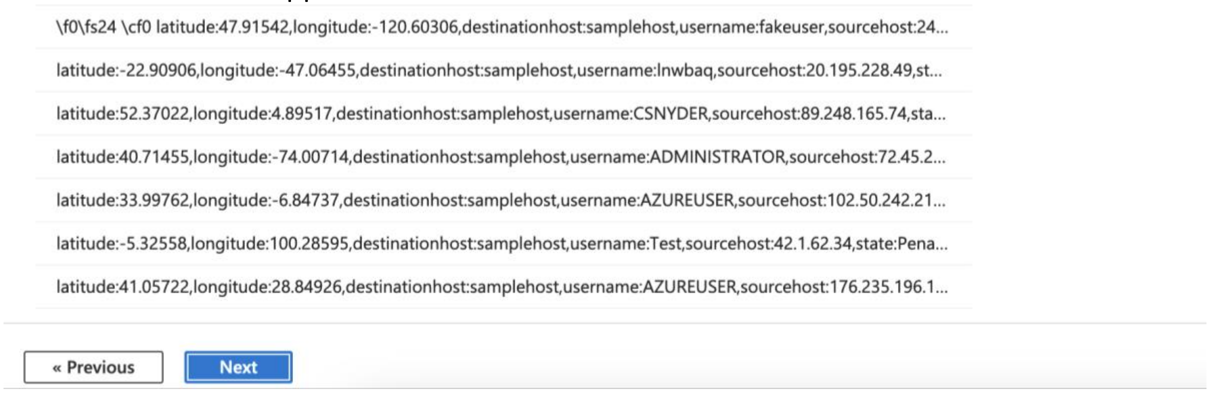


Upload a sample of the custom log. The wizard will parse and display the entries in this file. Learn more

Sample log

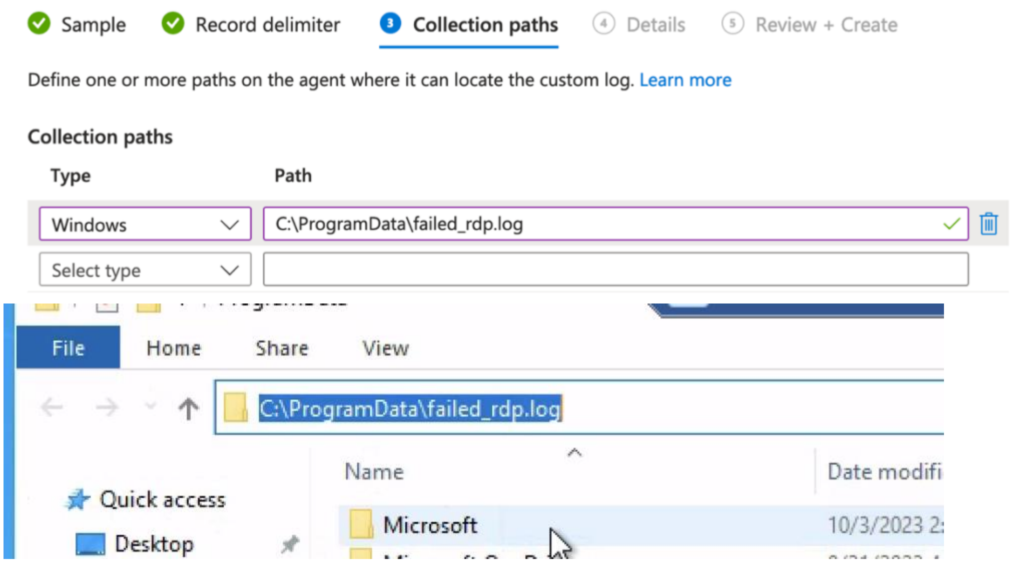
Select a sample log * "failed_rdp.log.rtf"

This is how our data appears as seen below. Click next



```
\f0\fs24 \cf0 latitude:47.91542,longitude:-120.60306,destinationhost:samplehost,username:fakeuser,sourcehost:24...
latitude:-22.90906,longitude:-47.06455,destinationhost:samplehost,username:lnwbaq,sourcehost:20.195.228.49,st...
latitude:52.37022,longitude:4.89517,destinationhost:samplehost,username:CSNYDER,sourcehost:89.248.165.74,sta...
latitude:40.71455,longitude:-74.00714,destinationhost:samplehost,username:ADMINISTRATOR,sourcehost:72.45.2...
latitude:33.99762,longitude:-6.84737,destinationhost:samplehost,username:AZUREUSER,sourcehost:102.50.242.21...
latitude:-5.32558,longitude:100.28595,destinationhost:samplehost,username:Test,sourcehost:42.1.62.34,state:Pena...
latitude:41.05722,longitude:28.84926,destinationhost:samplehost,username:AZUREUSER,sourcehost:176.235.196.1...
```

Make sure the directory path to the file is correct. Again see Part 4. Click next

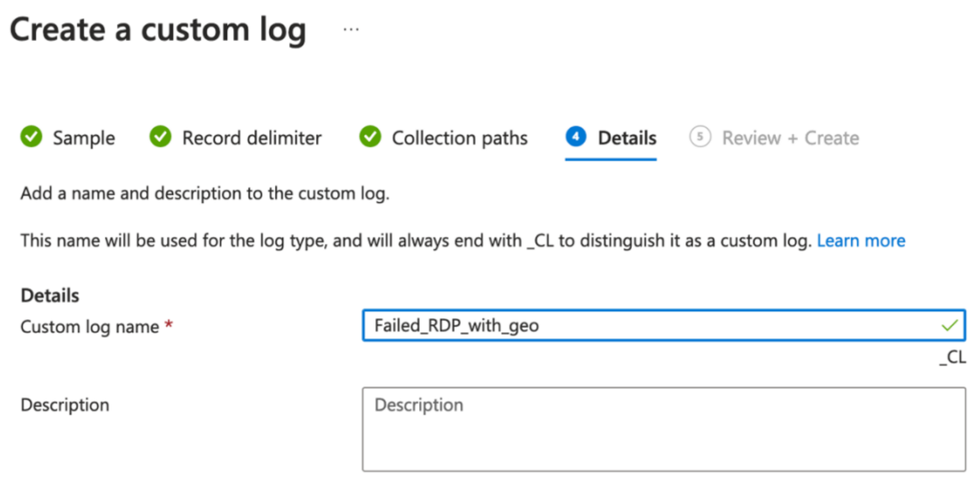


Define one or more paths on the agent where it can locate the custom log. Learn more

Collection paths

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

Provide a name for the Custom log as I have done below and then click “Review + Create”



Create a custom log

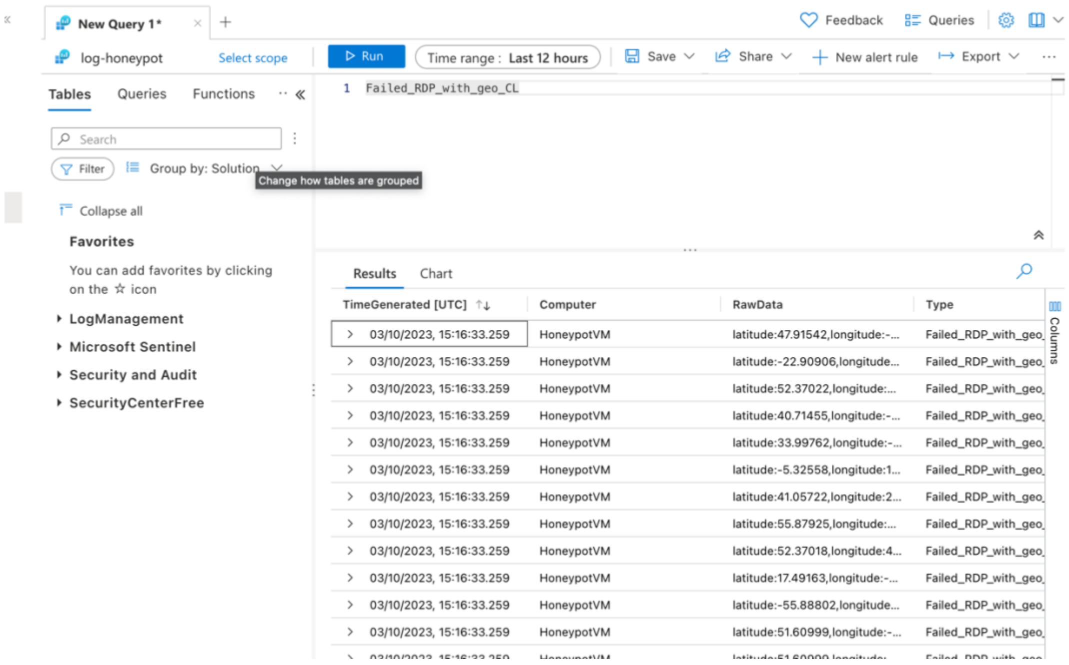
Add a name and description to the custom log.

This name will be used for the log type, and will always end with _CL to distinguish it as a custom log. Learn more

Details

Custom log name *	Failed_RDP_with_geo _CL
Description	Description

We will have to wait for a few minutes for the custom log to finalise. Once done if we head down to logs and run the query below for a custom log then we should start seeing our results being pulled from that log file on our VM.



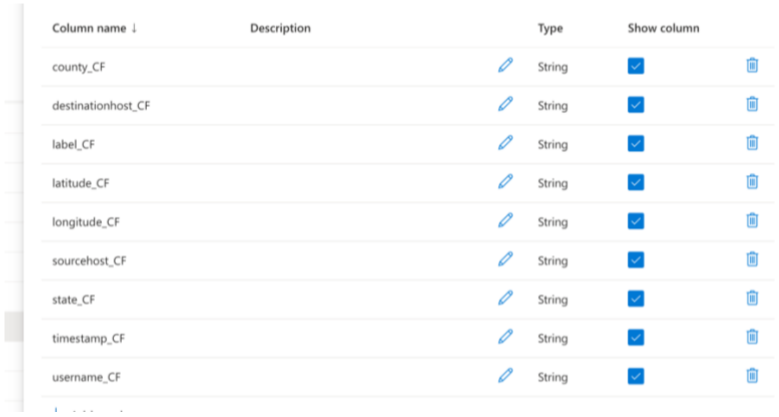
```
Failed_RDP_with_geo_CL
```

TimeGenerated [UTC]	Computer	RawData	Type
03/10/2023, 15:16:33.259	HoneypotVM	latitude:47.91542,longitude:-...	Failed_RDP_with_geo_
03/10/2023, 15:16:33.259	HoneypotVM	latitude:-22.90906,longitude...	Failed_RDP_with_geo_
03/10/2023, 15:16:33.259	HoneypotVM	latitude:52.37022,longitude:...	Failed_RDP_with_geo_
03/10/2023, 15:16:33.259	HoneypotVM	latitude:40.71455,longitude:-...	Failed_RDP_with_geo_
03/10/2023, 15:16:33.259	HoneypotVM	latitude:33.99762,longitude:-...	Failed_RDP_with_geo_
03/10/2023, 15:16:33.259	HoneypotVM	latitude:-5.32558,longitude:1...	Failed_RDP_with_geo_
03/10/2023, 15:16:33.259	HoneypotVM	latitude:41.05722,longitude:2...	Failed_RDP_with_geo_
03/10/2023, 15:16:33.259	HoneypotVM	latitude:55.87925,longitude:...	Failed_RDP_with_geo_
03/10/2023, 15:16:33.259	HoneypotVM	latitude:52.37018,longitude:4...	Failed_RDP_with_geo_
03/10/2023, 15:16:33.259	HoneypotVM	latitude:17.49163,longitude:-...	Failed_RDP_with_geo_
03/10/2023, 15:16:33.259	HoneypotVM	latitude:-55.88802,longitude...	Failed_RDP_with_geo_
03/10/2023, 15:16:33.259	HoneypotVM	latitude:51.60999,longitude:-...	Failed_RDP_with_geo_

Now originally we used to be able to click on the arrow in the box highlighted above and from there manually select the custom columns that we wish to use but because of all these recent changes/updates we are now no longer able to do it that way. As you can see from the image above, its the "RawData" column that we want to create custom fields from. Below we can see that within this "RawData" we have the fields, latitude, longitude, destinationhost, username, sourcehost and also country and state fields. These are the additional fields that we need to add in our custom log so that Sentinel can grab that GeoData and plot that on our Map. Otherwise its just 1 long string of random information.

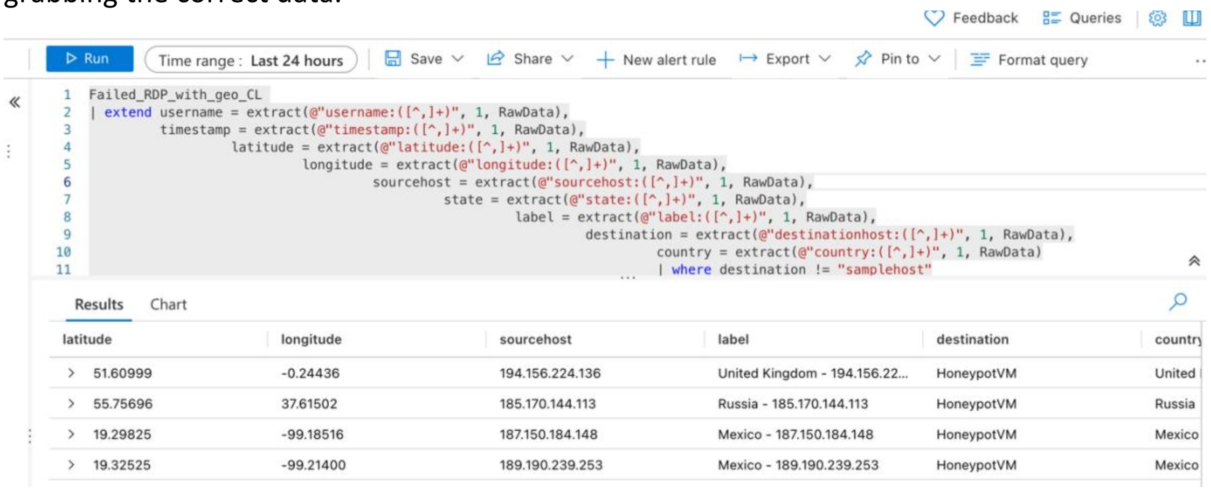
```
latitude:41.05722,longitude:28.84926,destinationhost:samplehost,username:AZUREUSER,sourcehost:176.235.196.1...
```

To get around this I tried manually adding the custom fields via "Edit Schema" on the logs section however I was unsuccessful as the query although allowed me to add the fields, was unable to grab the necessary data from the section above and organise it in a way that Sentinel could understand.



Column name ↓	Description	Type	Show column
county_CF		String	✓
destinationhost_CF		String	✓
label_CF		String	✓
latitude_CF		String	✓
longitude_CF		String	✓
sourcehost_CF		String	✓
state_CF		String	✓
timestamp_CF		String	✓
username_CF		String	✓

So I scrapped that and decided to go straight from the query. I added the query below which you can clearly see extracts the key fields I noted above from the “RawData” column. You can clearly see 9 new custom fields being taken just from the “RawData” section I showed above.
And now when I click “Run” you can see the longitude, latitude, country etc fields below grabbing the correct data.



```
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)
                                                                        | where destination != "samplehost"
```

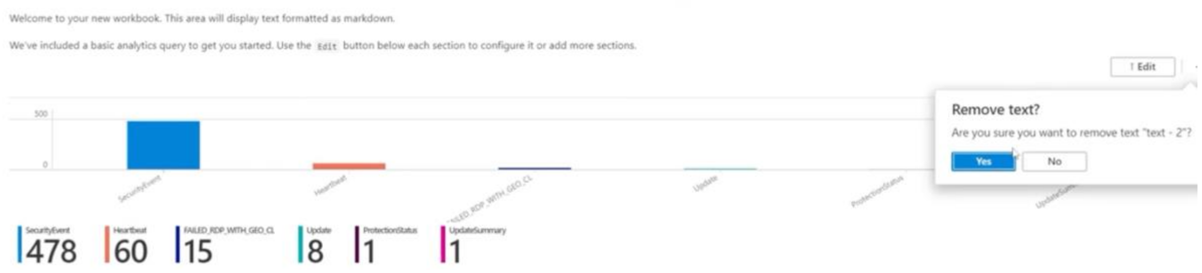
latitude	longitude	sourcehost	label	destination	country
51.60999	-0.24436	194.156.224.136	United Kingdom - 194.156.22...	HoneypotVM	United
55.75696	37.61502	185.170.144.113	Russia - 185.170.144.113	HoneypotVM	Russia
19.29825	-99.18516	187.150.184.148	Mexico - 187.150.184.148	HoneypotVM	Mexico
19.32525	-99.21400	189.190.239.253	Mexico - 189.190.239.253	HoneypotVM	Mexico

Part – 6 Testing the extracts from the query and setting up the workbook for Sentinel Mapping.

Now just to check that all is working as it should be, we can go back and purposely attempt a few failed logins to our VM to see if it is registering correctly in our query.

Once confirmed we can now head over to Sentinel > Workbooks > Create New Workbook

Now automatically Sentinel provides you with a few widgets to display your data, we want to remove these and then go to Add > Add a query

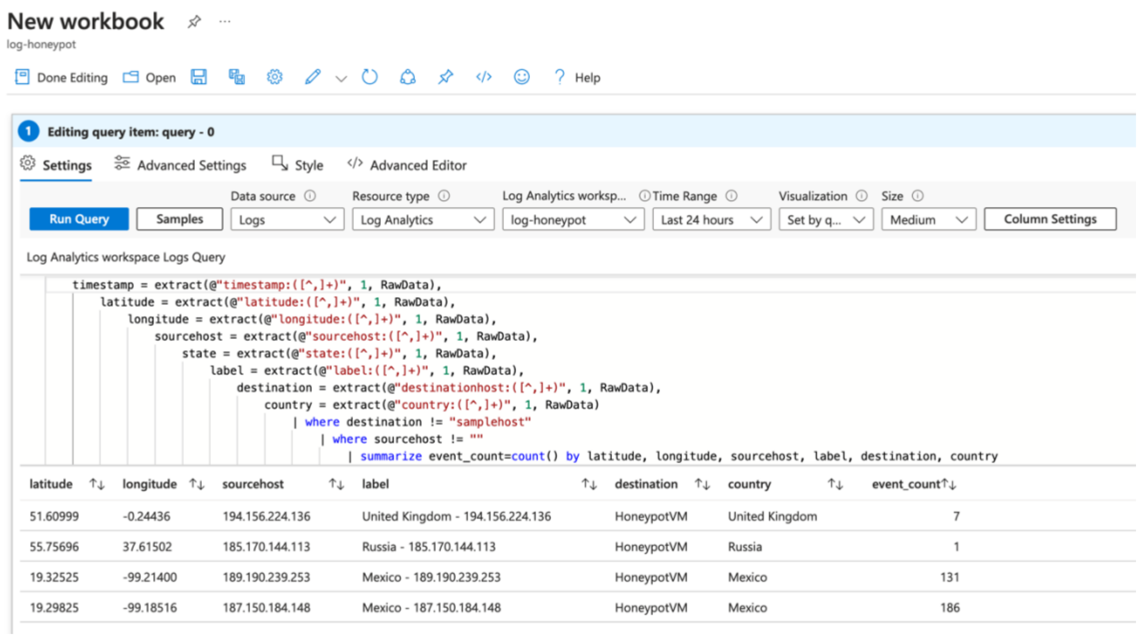

Welcome to your new workbook. This area will display text formatted as markdown.

We've included a basic analytics query to get you started. Use the `Edit` button below each section to configure it or add more sections.

Remove text?

Are you sure you want to remove text "text - 2"?

Here we want to paste I have done below the same query as I used in the log query earlier above. Run the query to make sure that the data is coming in correctly as seen below with the correct fields shown.



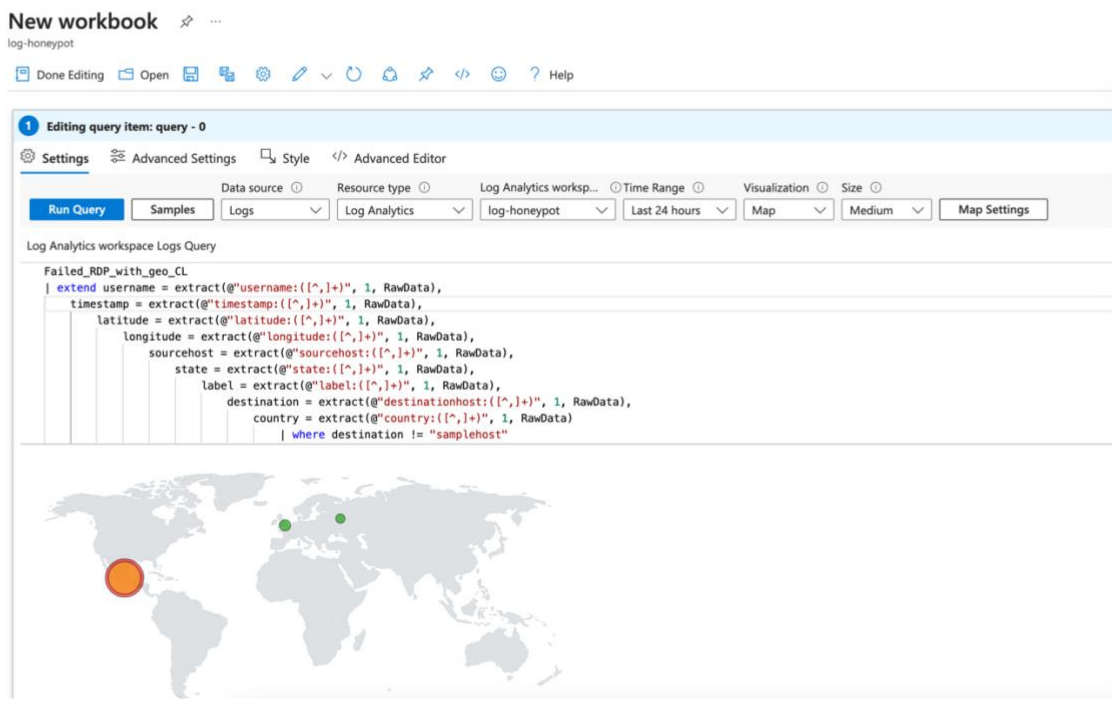
Data source: Logs
Resource type: Log Analytics
Log Analytics worksp...: log-honeypot
Time Range: Last 24 hours
Visualization: Set by q...
Size: Medium

Log Analytics workspace Logs Query

```
    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
```

latitude	longitude	sourcehost	label	destination	country	event_count
51.60999	-0.24436	194.156.224.136	United Kingdom - 194.156.224.136	HoneypotVM	United Kingdom	7
55.75696	37.61502	185.170.144.113	Russia - 185.170.144.113	HoneypotVM	Russia	1
19.32525	-99.21400	189.190.239.253	Mexico - 189.190.239.253	HoneypotVM	Mexico	131
19.29825	-99.18516	187.150.184.148	Mexico - 187.150.184.148	HoneypotVM	Mexico	186

I then changed the visualization section above to “Map” as we want this Geodata to be presented to us on a global scale so we can pinpoint where these attacks are coming from.



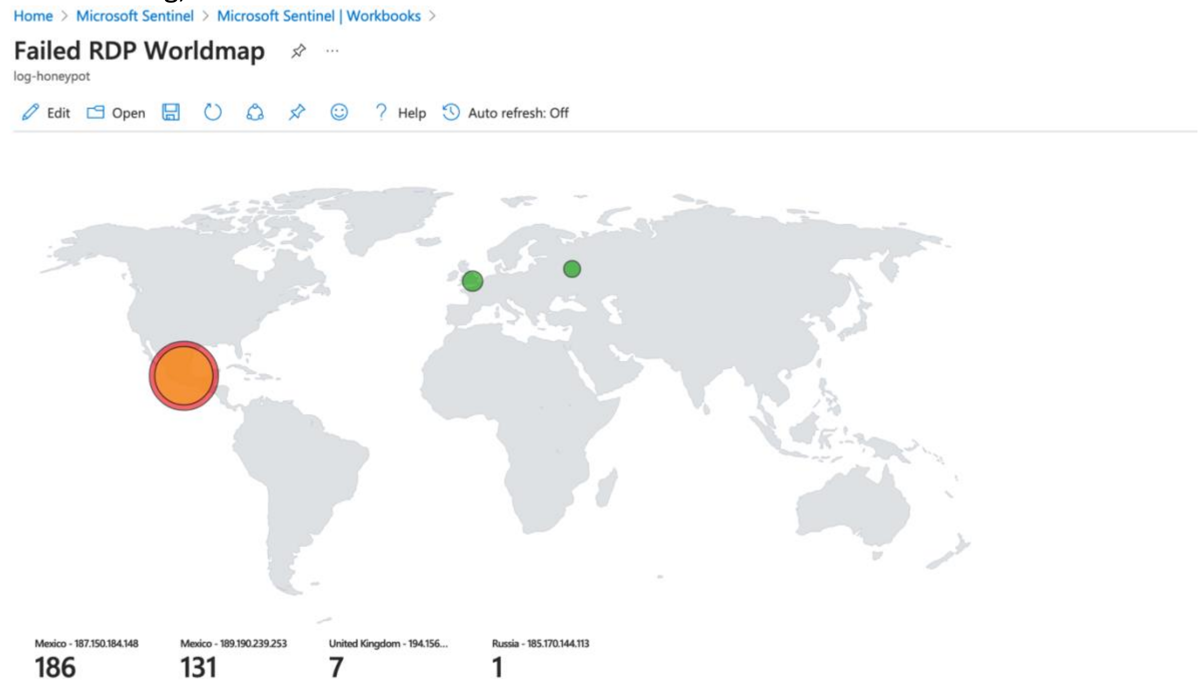
Data source: Logs
Resource type: Log Analytics
Log Analytics worksp...: log-honeypot
Time Range: Last 24 hours
Visualization: Map
Size: Medium

Log Analytics workspace Logs 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)
                                    | where destination != "samplehost"
```

There are a couple of adjustments that need to be made in “Map Settings” to make sure that we have selected the right data to be pulled from the correct fields and once done that should be it.

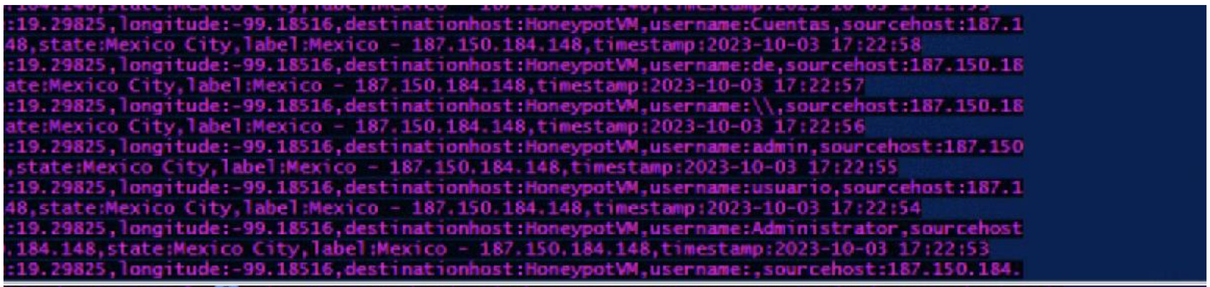
As you can see already we have some attacks being registered and plotted on our Map. If we see further down we can note the number of attacks coming through and from which country and IP. This screen shot below was taken immediately after I finished setting this up in the evening, 19:30.


Home > Microsoft Sentinel > Microsoft Sentinel | Workbooks >
Failed RDP Worldmap
log-honeypot
Edit
Open
Help
Auto refresh: Off

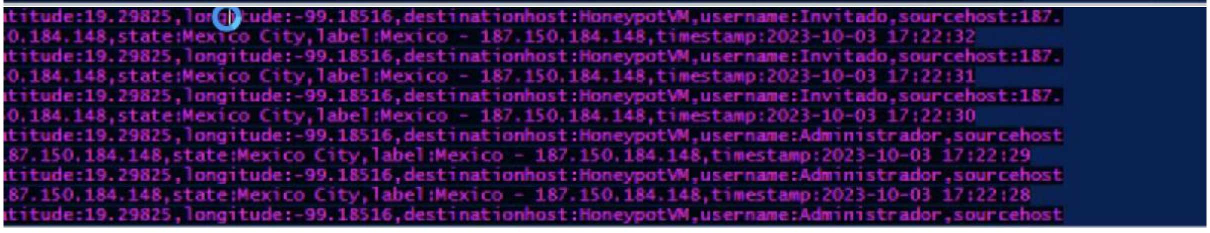
Mexico - 187.150.184.148	Mexico - 189.190.239.253	United Kingdom - 194.156...	Russia - 185.170.144.113
186	131	7	1

Here via Powershell on the VM we can clearly see the attempted usernames being used by our attackers. A lot of these are spanish which makes sense as the majority of attacks I received have so far come from Mexico within the first hour.

Username: Cuentas
Username: \\
Username: admin
Username: usuario
Username: Administrator
Username: de
Username: Invitado

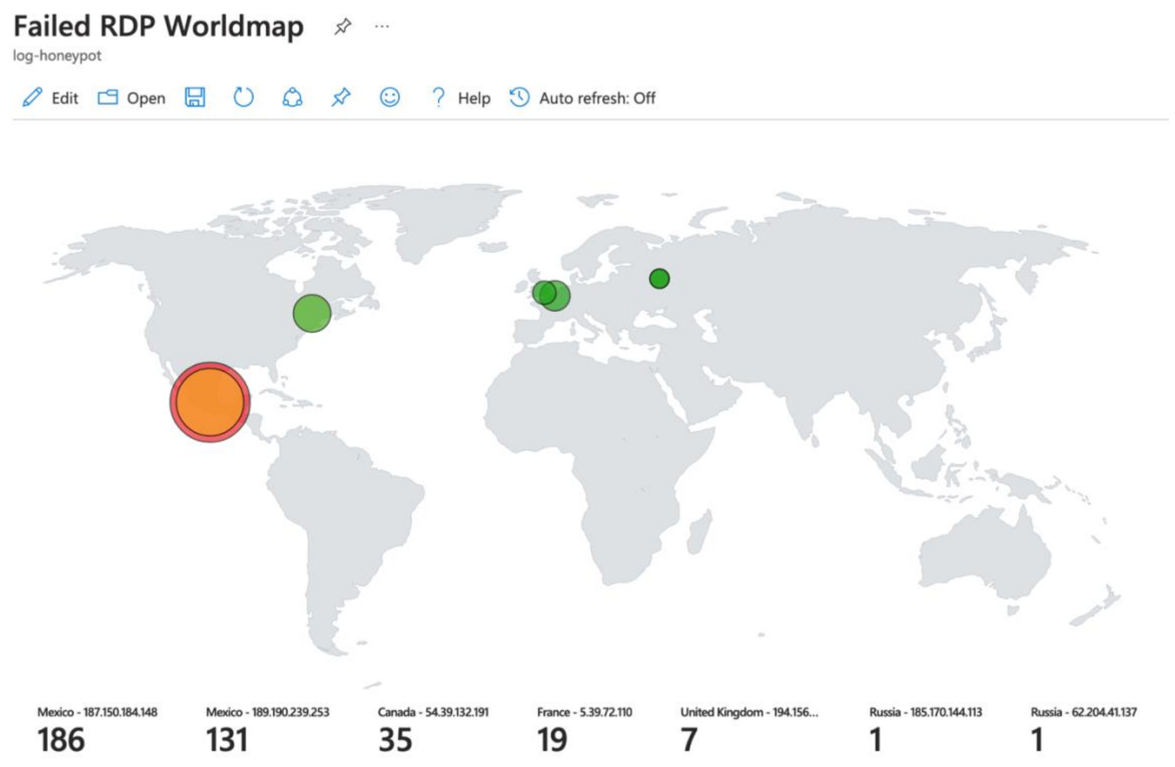


```
:19.29825,longitude:-99.18516,destinationhost:HoneypotVM,username:Cuentas,sourcehost:187.1
48,state:Mexico City,label:Mexico - 187.150.184.148,timestamp:2023-10-03 17:22:58
:19.29825,longitude:-99.18516,destinationhost:HoneypotVM,username:de,sourcehost:187.150.18
ate:Mexico City,label:Mexico - 187.150.184.148,timestamp:2023-10-03 17:22:57
:19.29825,longitude:-99.18516,destinationhost:HoneypotVM,username:\\,sourcehost:187.150.18
ate:Mexico City,label:Mexico - 187.150.184.148,timestamp:2023-10-03 17:22:56
:19.29825,longitude:-99.18516,destinationhost:HoneypotVM,username:admin,sourcehost:187.150
,state:Mexico City,label:Mexico - 187.150.184.148,timestamp:2023-10-03 17:22:55
:19.29825,longitude:-99.18516,destinationhost:HoneypotVM,username:usuario,sourcehost:187.1
48,state:Mexico City,label:Mexico - 187.150.184.148,timestamp:2023-10-03 17:22:54
:19.29825,longitude:-99.18516,destinationhost:HoneypotVM,username:Administrator,sourcehost
.184.148,state:Mexico City,label:Mexico - 187.150.184.148,timestamp:2023-10-03 17:22:53
:19.29825,longitude:-99.18516,destinationhost:HoneypotVM,username:,sourcehost:187.150.184.
```



```
titude:19.29825,longitude:-99.18516,destinationhost:HoneypotVM,username:Invitado,sourcehost:187.
0.184.148,state:Mexico City,label:Mexico - 187.150.184.148,timestamp:2023-10-03 17:22:32
titude:19.29825,longitude:-99.18516,destinationhost:HoneypotVM,username:Invitado,sourcehost:187.
0.184.148,state:Mexico City,label:Mexico - 187.150.184.148,timestamp:2023-10-03 17:22:31
titude:19.29825,longitude:-99.18516,destinationhost:HoneypotVM,username:Invitado,sourcehost:187.
0.184.148,state:Mexico City,label:Mexico - 187.150.184.148,timestamp:2023-10-03 17:22:30
titude:19.29825,longitude:-99.18516,destinationhost:HoneypotVM,username:Administrador,sourcehost
87.150.184.148,state:Mexico City,label:Mexico - 187.150.184.148,timestamp:2023-10-03 17:22:29
titude:19.29825,longitude:-99.18516,destinationhost:HoneypotVM,username:Administrador,sourcehost
87.150.184.148,state:Mexico City,label:Mexico - 187.150.184.148,timestamp:2023-10-03 17:22:28
titude:19.29825,longitude:-99.18516,destinationhost:HoneypotVM,username:Administrador,sourcehost
```

4 Hours later

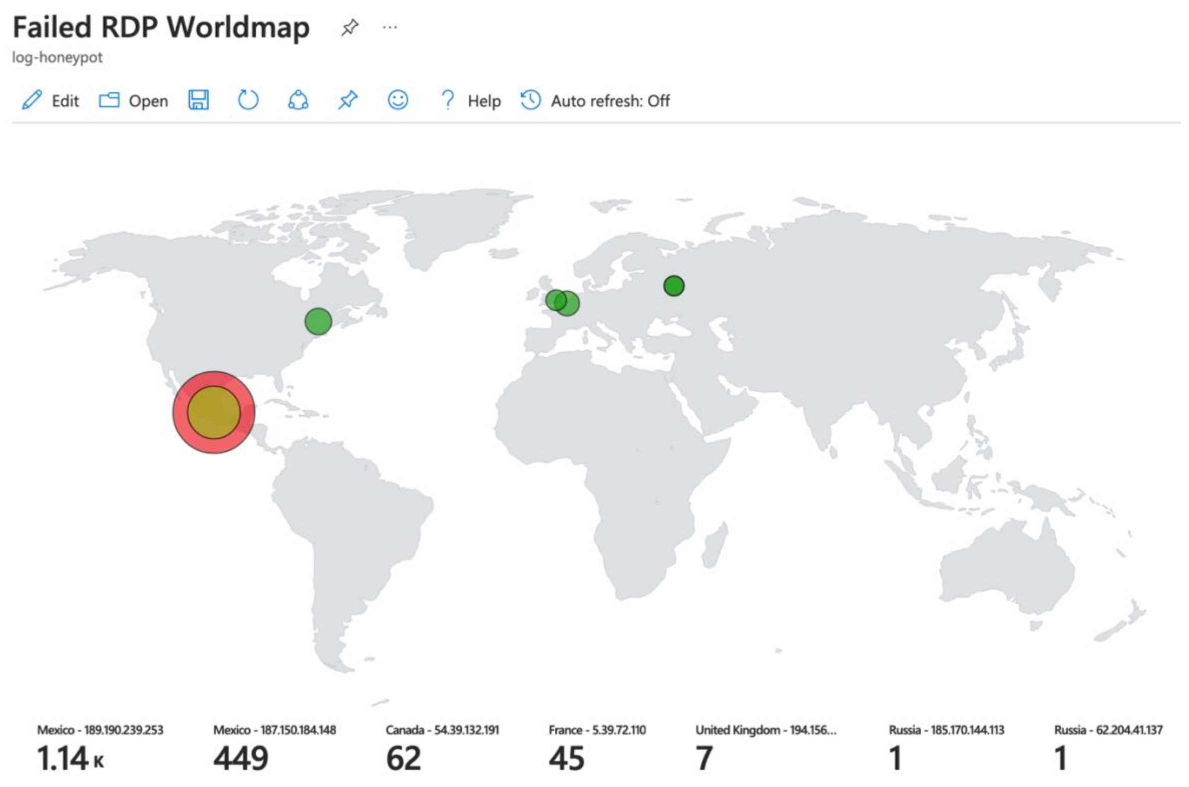


Failed RDP Worldmap

log-honeypot

Mexico - 187.150.184.148	Mexico - 189.190.239.253	Canada - 54.39.132.191	France - 5.39.72.110	United Kingdom - 194.156...	Russia - 185.170.144.113	Russia - 62.204.41.137
186	131	35	19	7	1	1

13 hours later

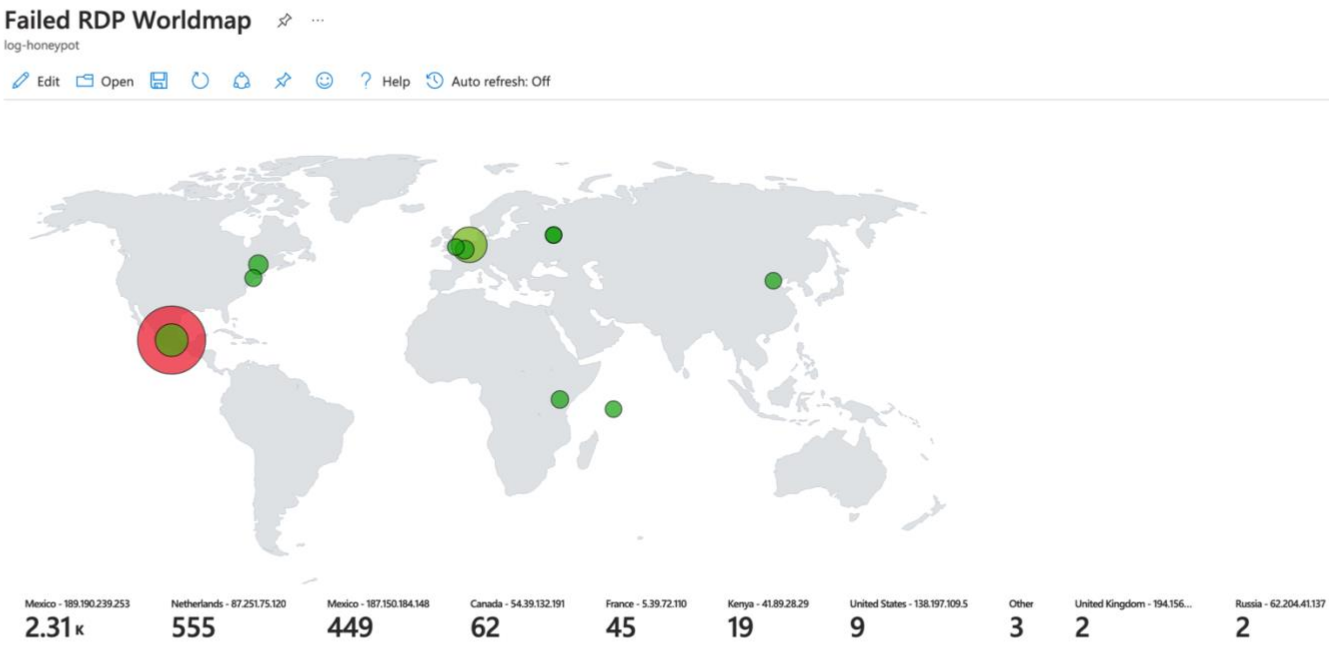


Failed RDP Worldmap

log-honeypot

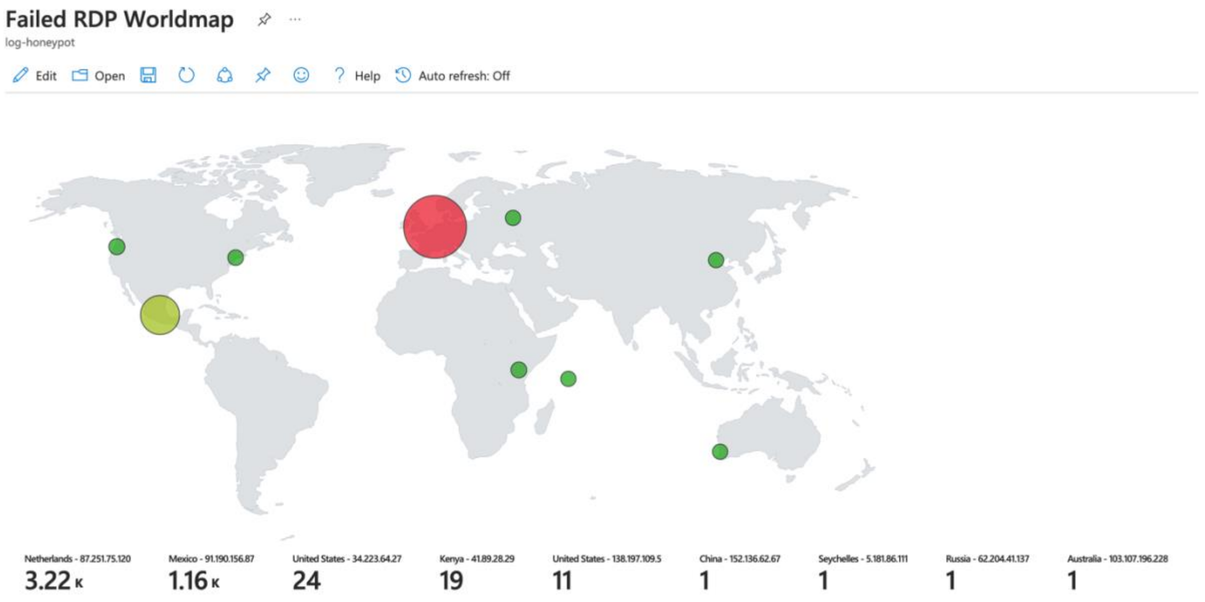
Mexico - 189.190.239.253	Mexico - 187.150.184.148	Canada - 54.39.132.191	France - 5.39.72.110	United Kingdom - 194.156...	Russia - 185.170.144.113	Russia - 62.204.41.137
1.14 K	449	62	45	7	1	1

22 hours later



Mexico - 189.190.239.253	Netherlands - 87.251.75.120	Mexico - 187.150.184.148	Canada - 54.39.132.191	France - 5.39.72.110	Kenya - 41.89.28.29	United States - 138.197.109.5	Other	United Kingdom - 194.156...	Russia - 62.204.41.137
2.31 K	555	449	62	45	19	9	3	2	2

34 hours later



Netherlands - 87.251.75.120	Mexico - 91.190.156.87	United States - 34.223.64.27	Kenya - 41.89.28.29	United States - 138.197.109.5	China - 152.136.62.67	Seychelles - 5.181.86.111	Russia - 62.204.41.137	Australia - 103.107.196.228
3.22 K	1.16 K	24	19	11	1	1	1	1