Fortifying the Virtual Perimeter: Honeypots and SIEM in Action

Justin Warren - Farah Costanza - Justin Chmura

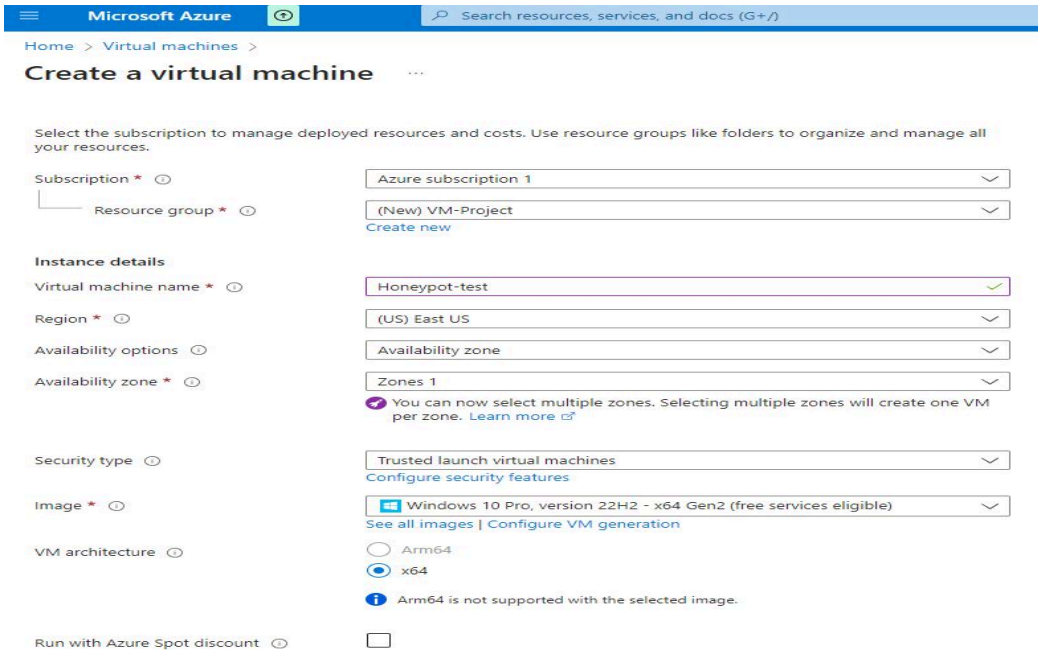
February 2024

* * *

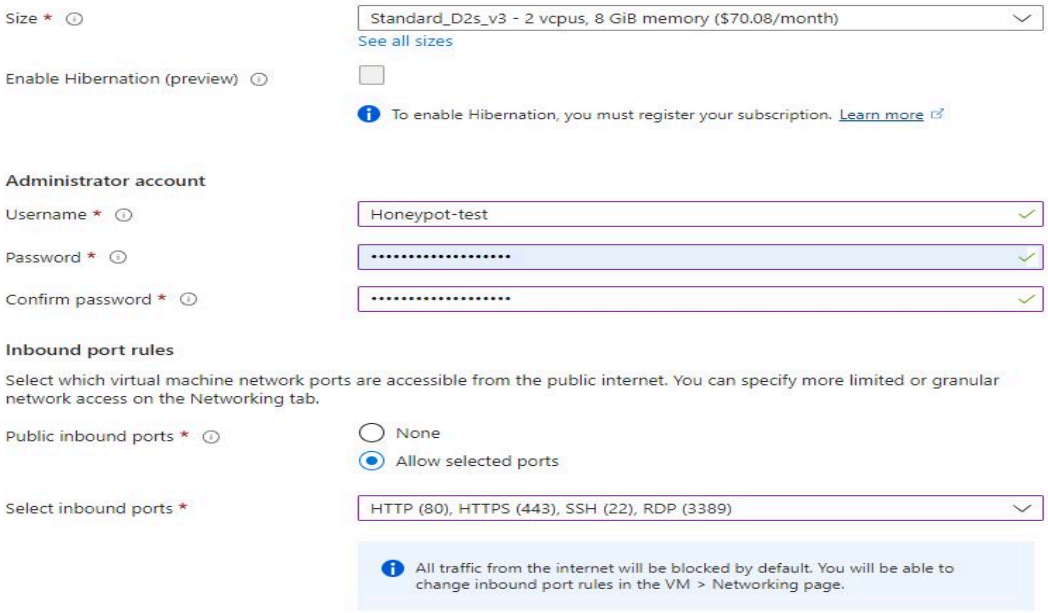
As technology evolves and interconnectivity deepens, fortifying the virtual perimeter becomes not just a defensive tactic but a proactive stance; essential for ensuring the confidentiality, integrity, and availability of digital assets in our increasingly evolving digital world. In this project, we go 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.

* * *

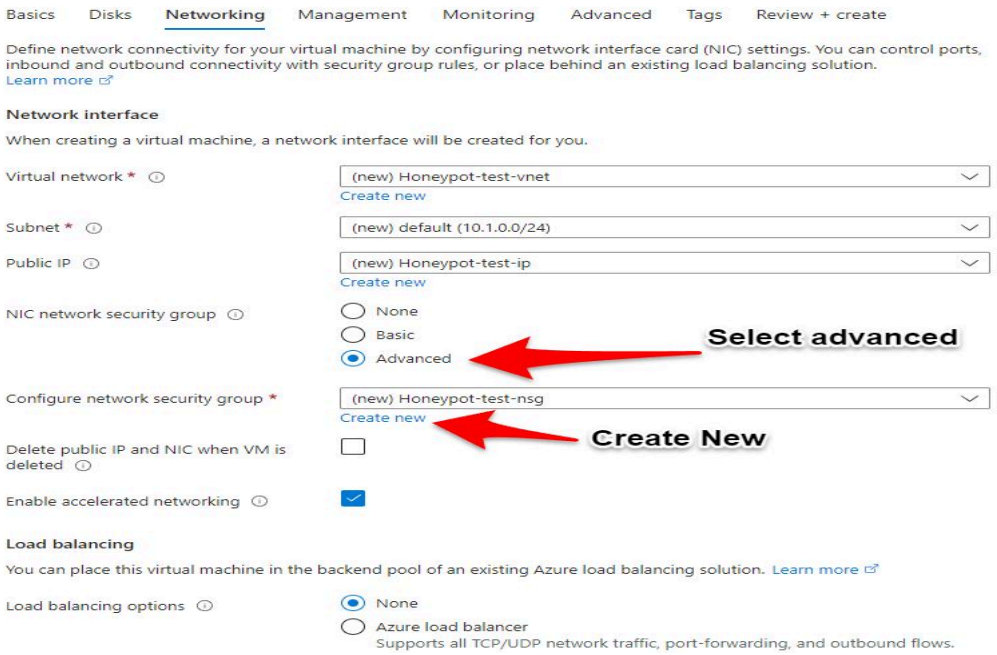
Getting started, we used Microsoft Azure, a cloud computing platform run by Microsoft to first set up the honeypot virtual machine. First, navigate to “Create a virtual machine”. Create a new resource group, in this case, we named it VM-Project, and named our VM “Honeypot-test”. The region, operating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



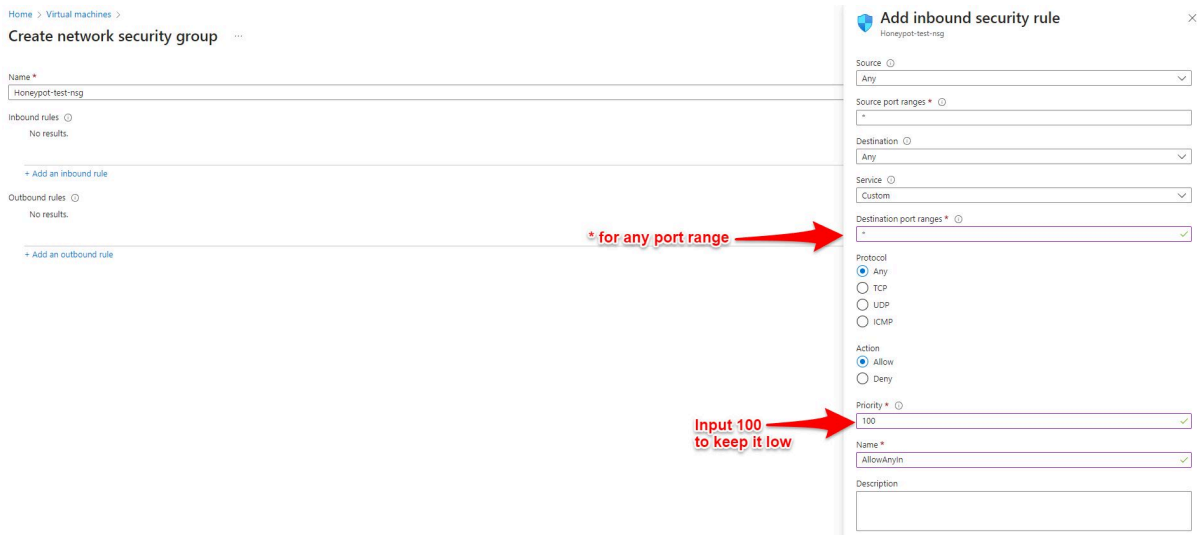
In the next step of creating the VM, we need to configure the size, and inbound ports and set our username and password. As shown below, we used the size “Standard_D2s_v3 - 2 vcpus, 8GiB memory, allowing the VM to efficiently run a Windows environment.



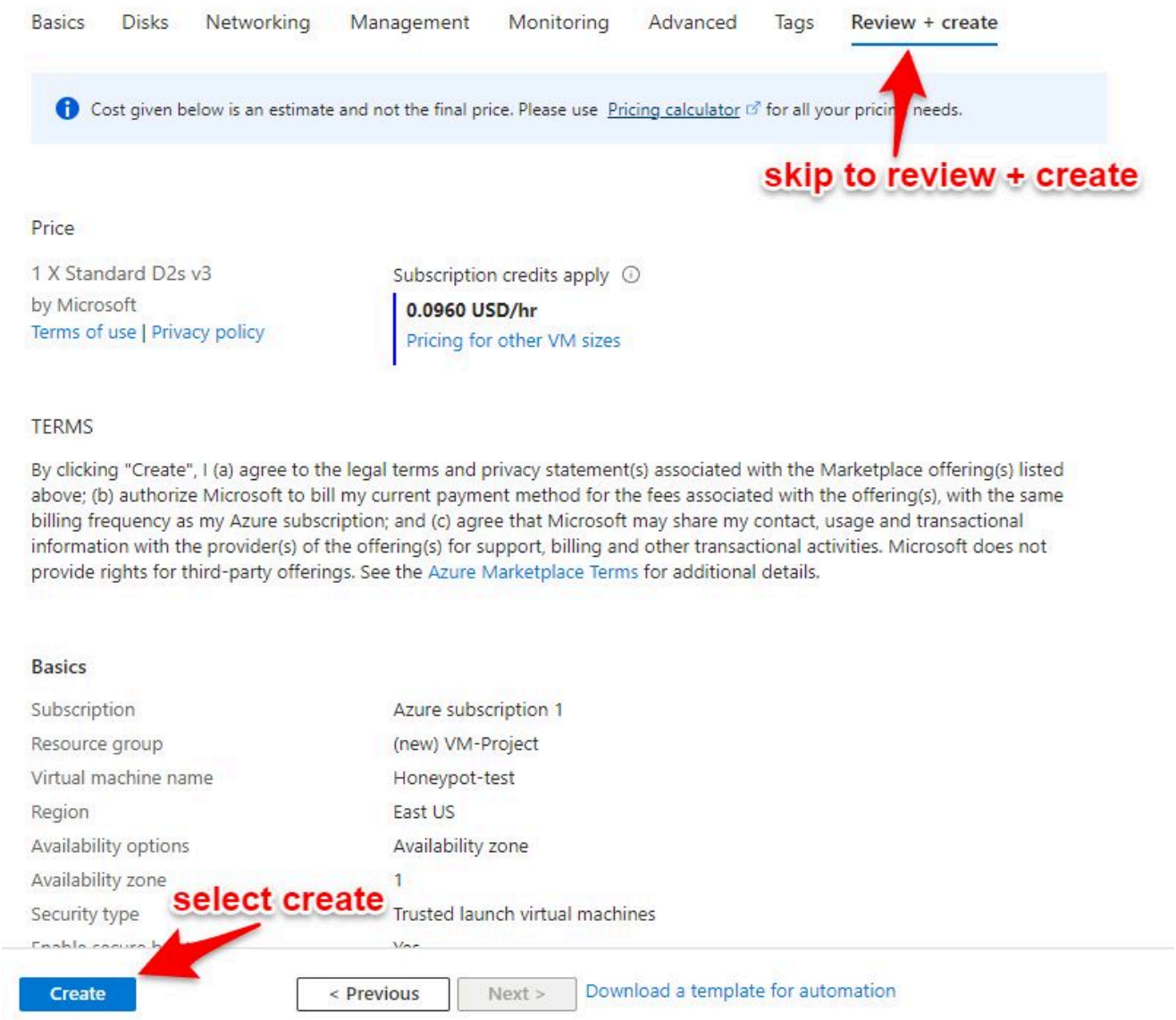
Continuing to the networking tab under (NIC networking security group), we select advanced and configure and create a new network security group.



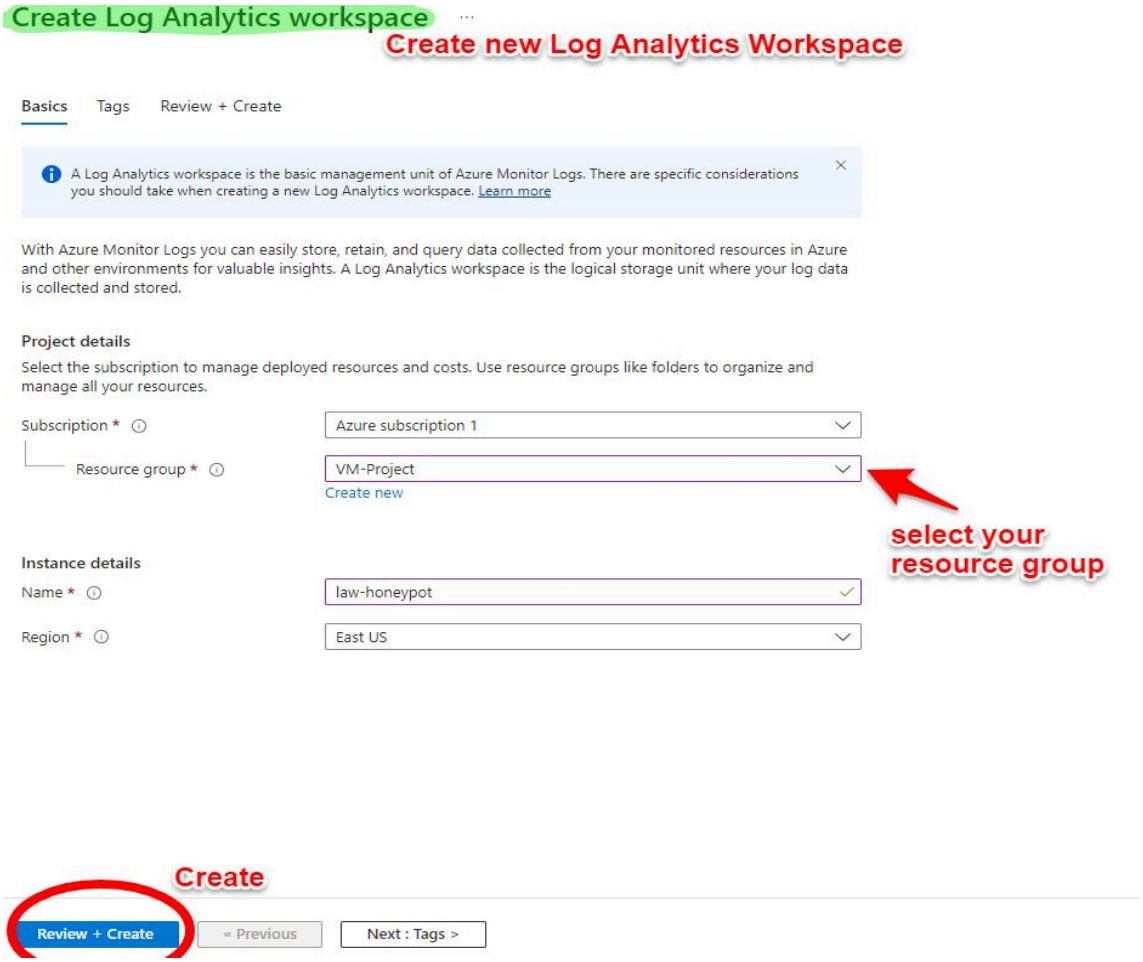
Once Create New is selected, delete the default inbound security rule and configure it as shown below. “Destination port ranges” will be changed to (*) for all. “Priority” should be changed to a low setting, in this case, we chose 100.



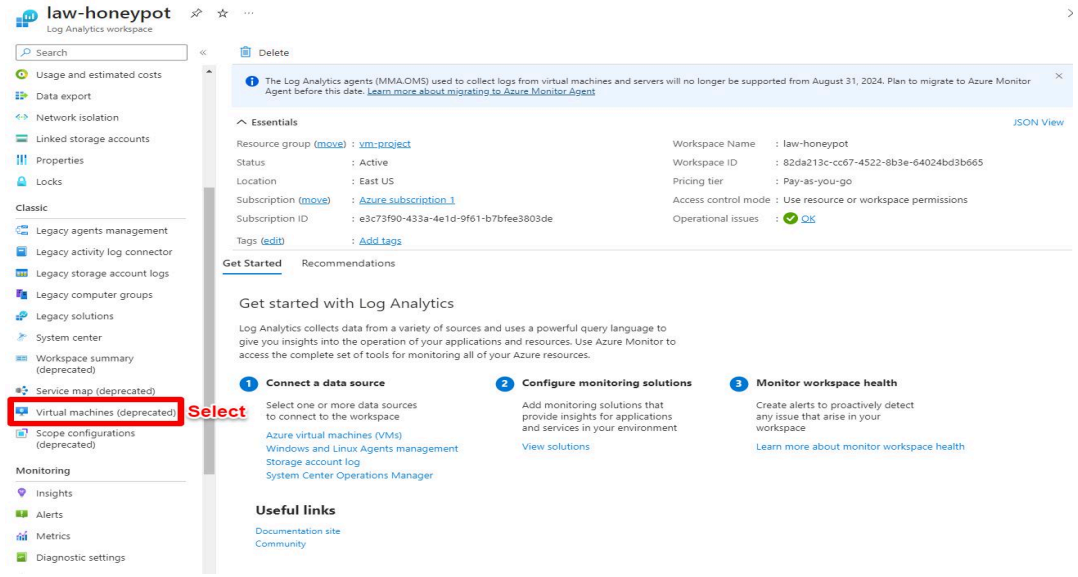
The final step in creating our honeypot is to select “Review + create” and allow the VM time to deploy.

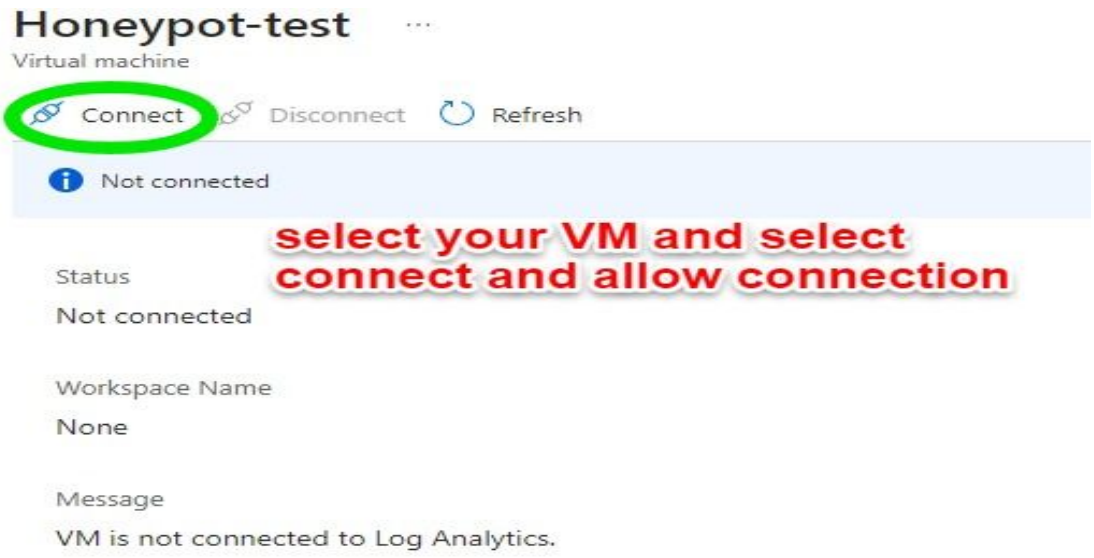


In the next step of this process we need to create a Log Analytics Workspace. Navigate to “Create Log Analytics Workspace” and select the resource group that we made in the previous step. Create the workspace and allow time to deploy.

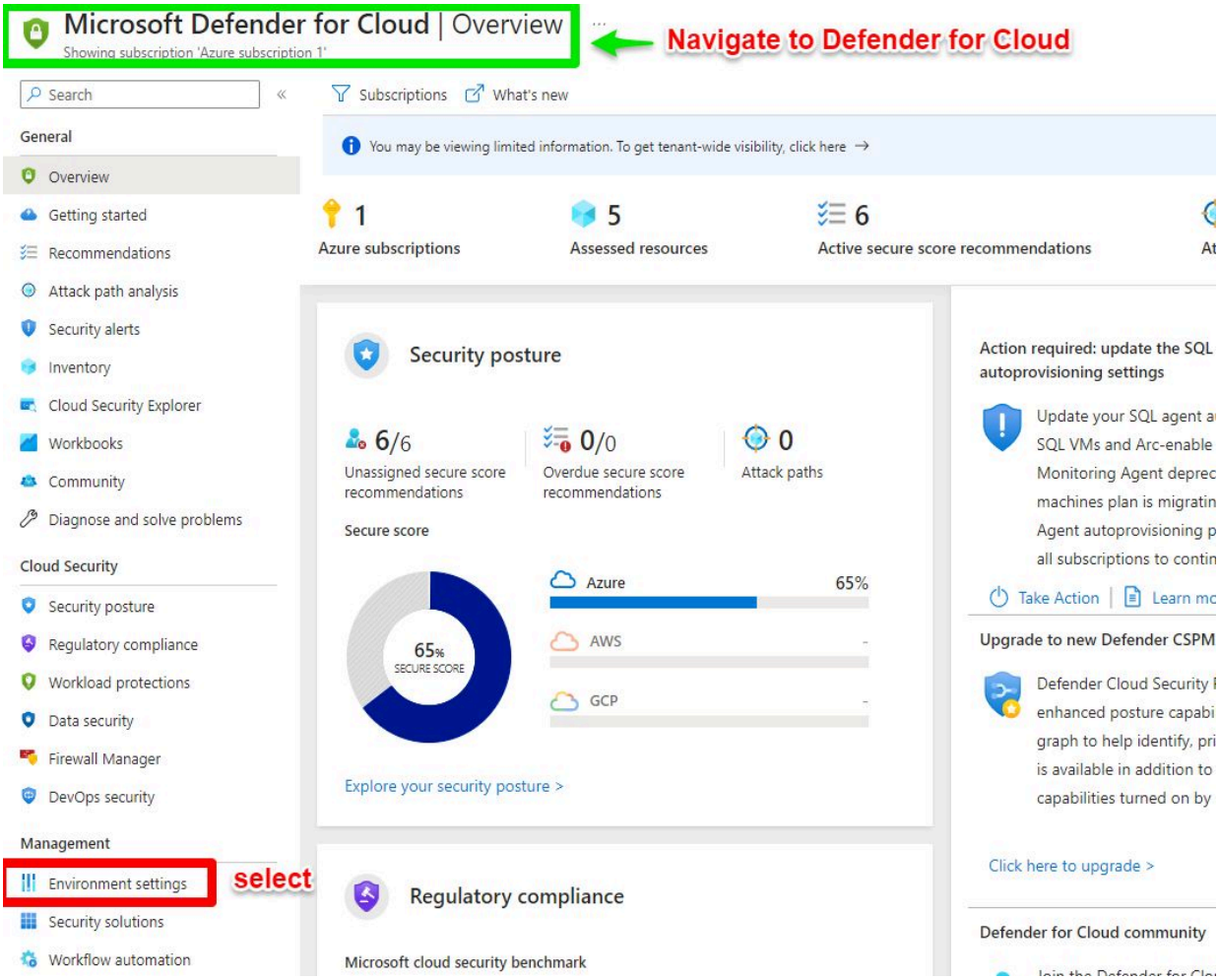


The next step is to connect our honeypot VM to our Log Analytics Workspace, to do so, in the newly created workspace navigate down the list of tabs to “Virtual Machine” as shown below.

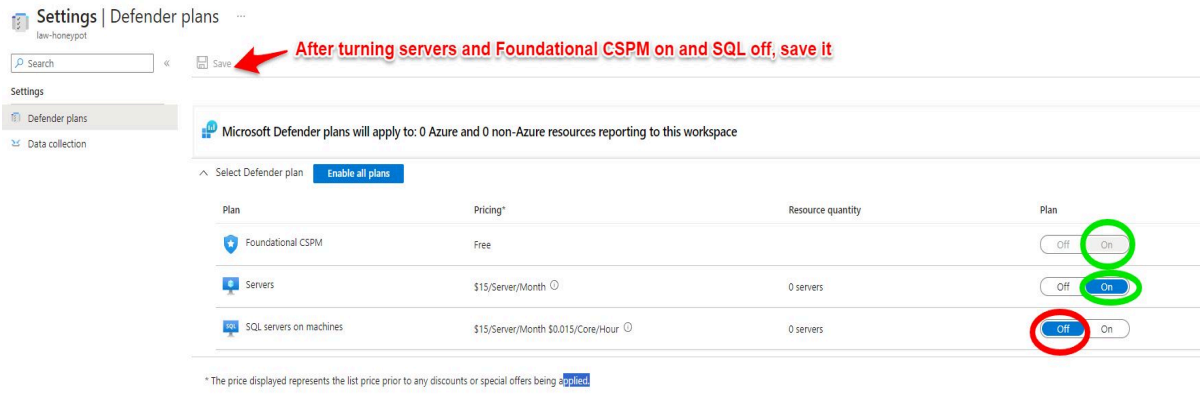




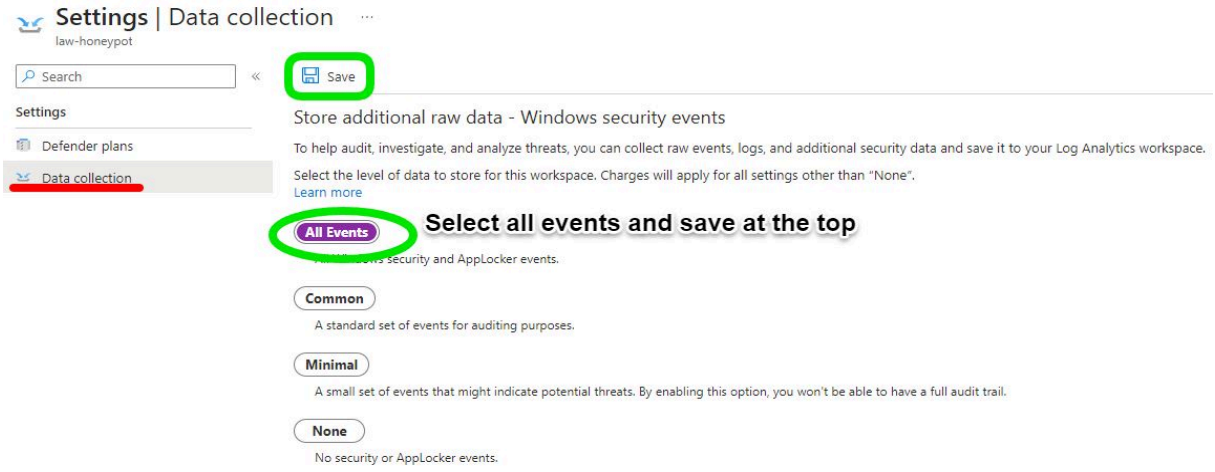
After we have successfully connected our VM to our Log Analytics Workspace, we then need to configure some settings in Microsoft Defender. As shown below, navigate to “Microsoft Defender for Cloud” and select the “Environment settings” tab.



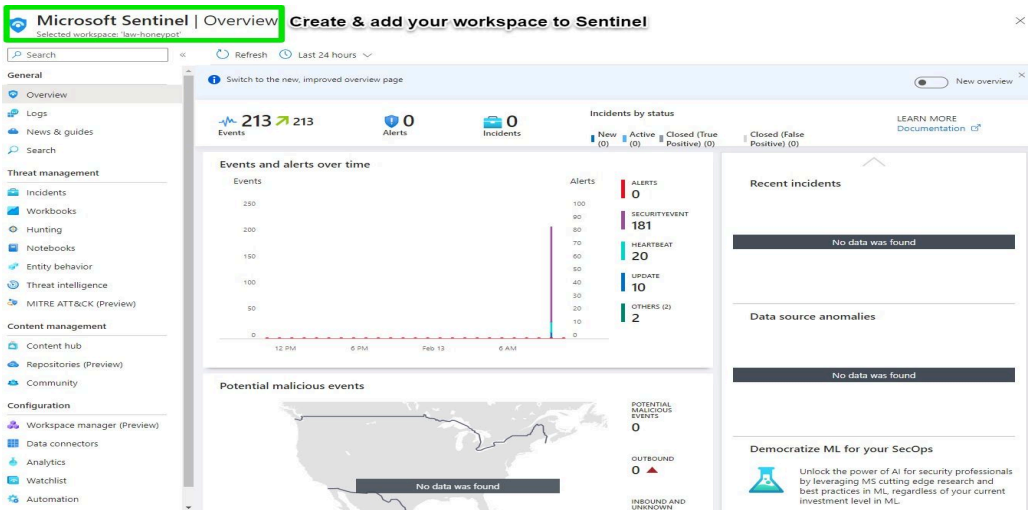
By selecting the lab that we have created, we can then configure our settings under “Defender plans”. Select the proper settings and save as pictured below.



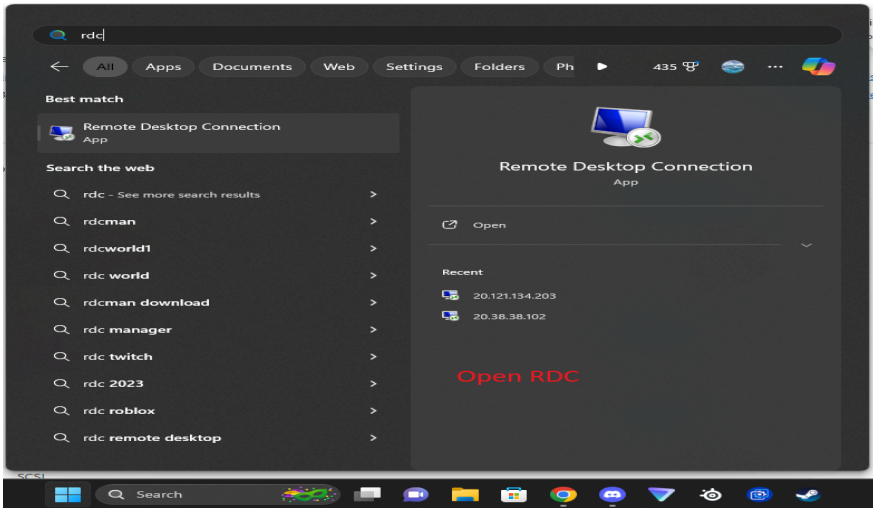
On the same page, navigate to the “Data collection” tab select all events and save.



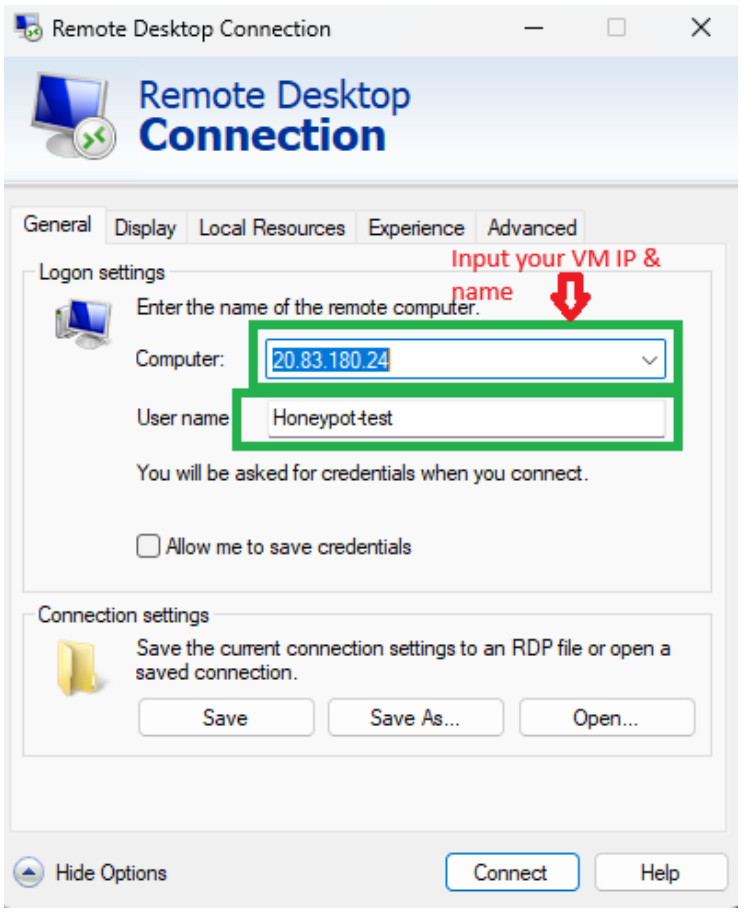
The next tool we are going to configure is Microsoft Sentinel, our SIEM environment. Simply navigate to “Microsoft Sentinel”, select create and add, and add the Log Analytics Workspace previously created. After doing so, it should look like the screenshot below.



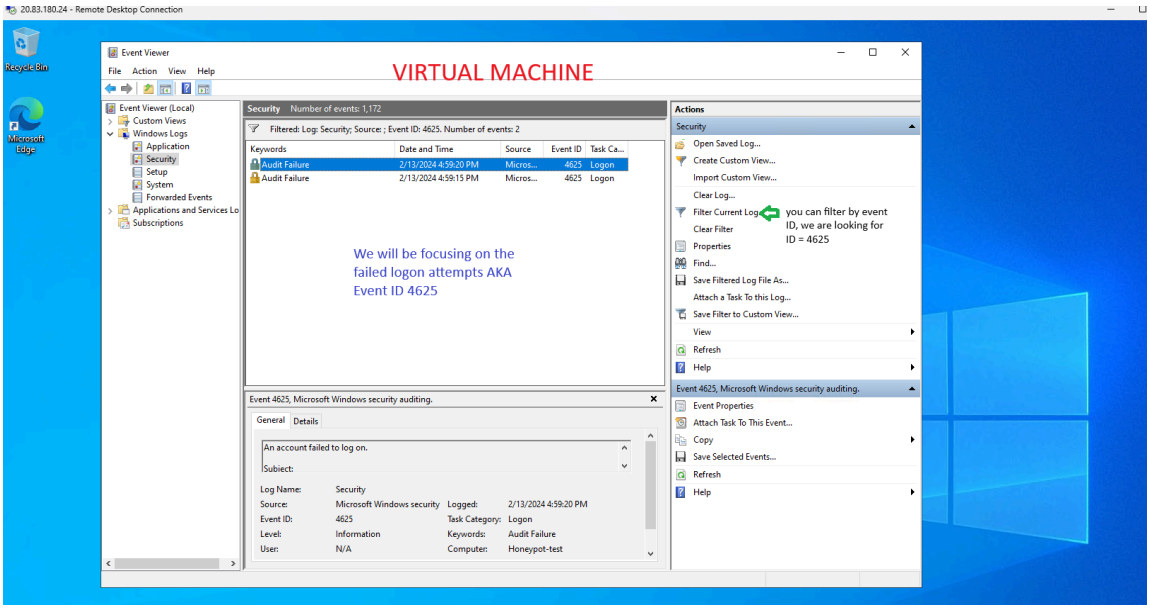
Next, we need to establish a remote desktop connection between our local and virtual machines. To do that, we need to navigate to RDC (Remote Desktop Connection) in our Windows search bar, as we are using a Windows VM.



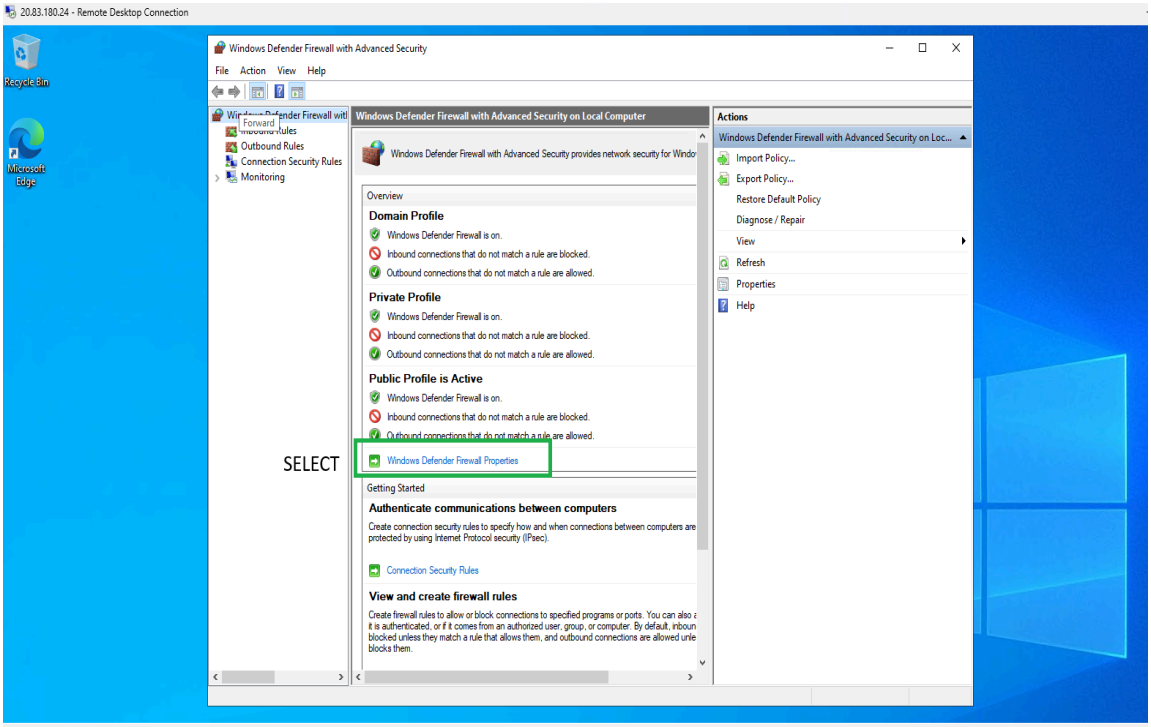
Once opened we input our virtual machine's public IPv4 address and Username we set earlier to establish a connection. You can purposefully input the wrong credentials on the login screen to see a failed attempt in the Event Viewer, we will talk about that shortly.



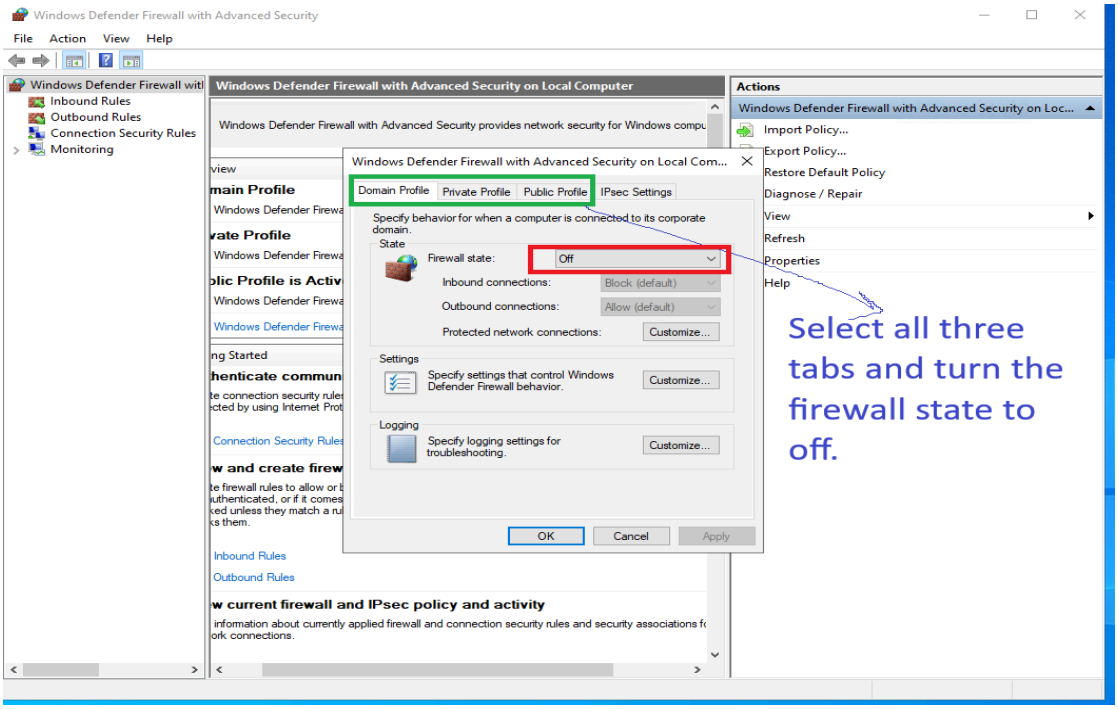
In our virtual machine, navigate to Event Viewer and filter the logs as shown below. We will focus on the Event ID (4625), which will show us all the failed login attempts.



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".

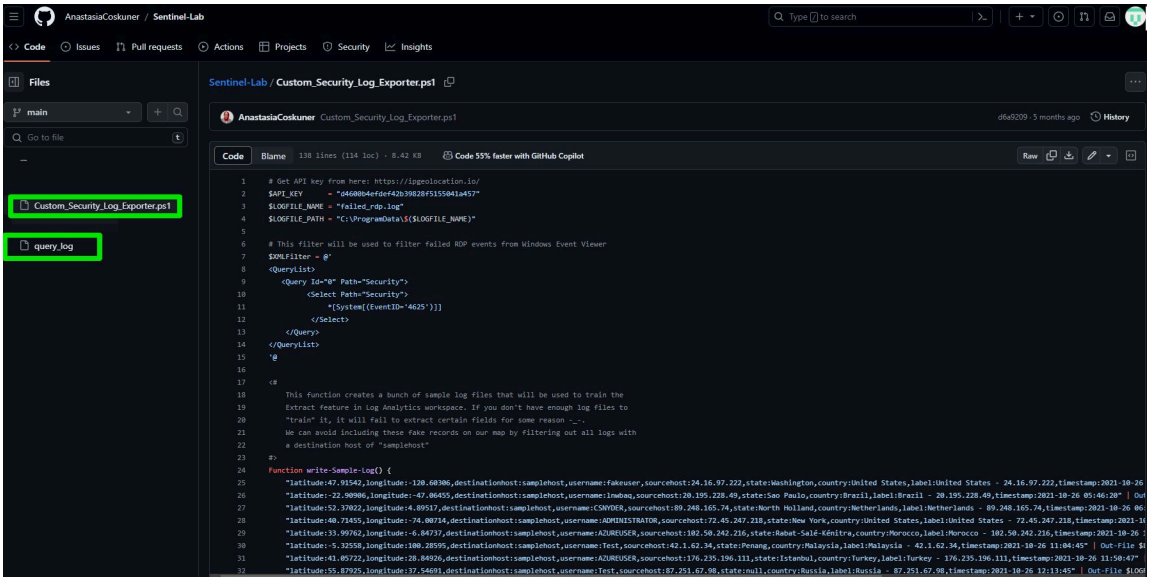


Select all three tabs, switch the Firewall state to off, and select ok. We then can proceed to PowerShell.

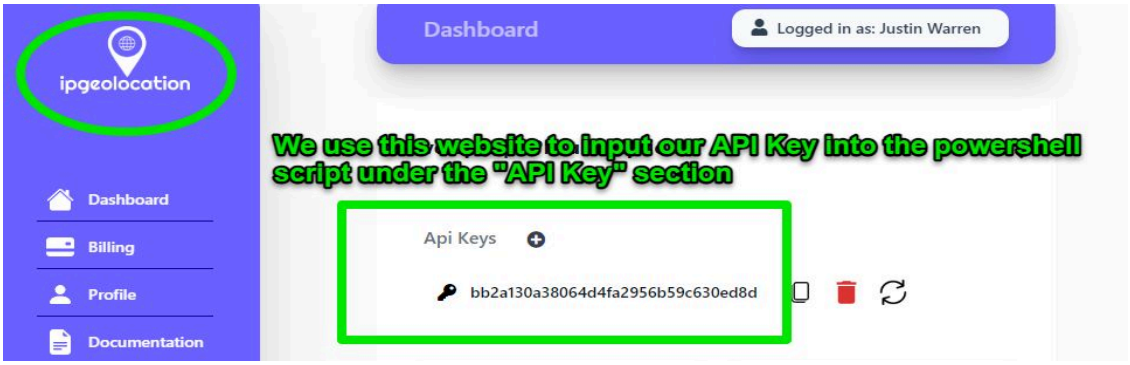


In the next step, we need to create a custom log, but before we do that we need to retrieve the script we will use to create our custom log, you can find that here: https://github.com/AnastasiaCoskuner/Sentinel-Lab/blob/main/Custom_Security_Log_Exporter.ps1

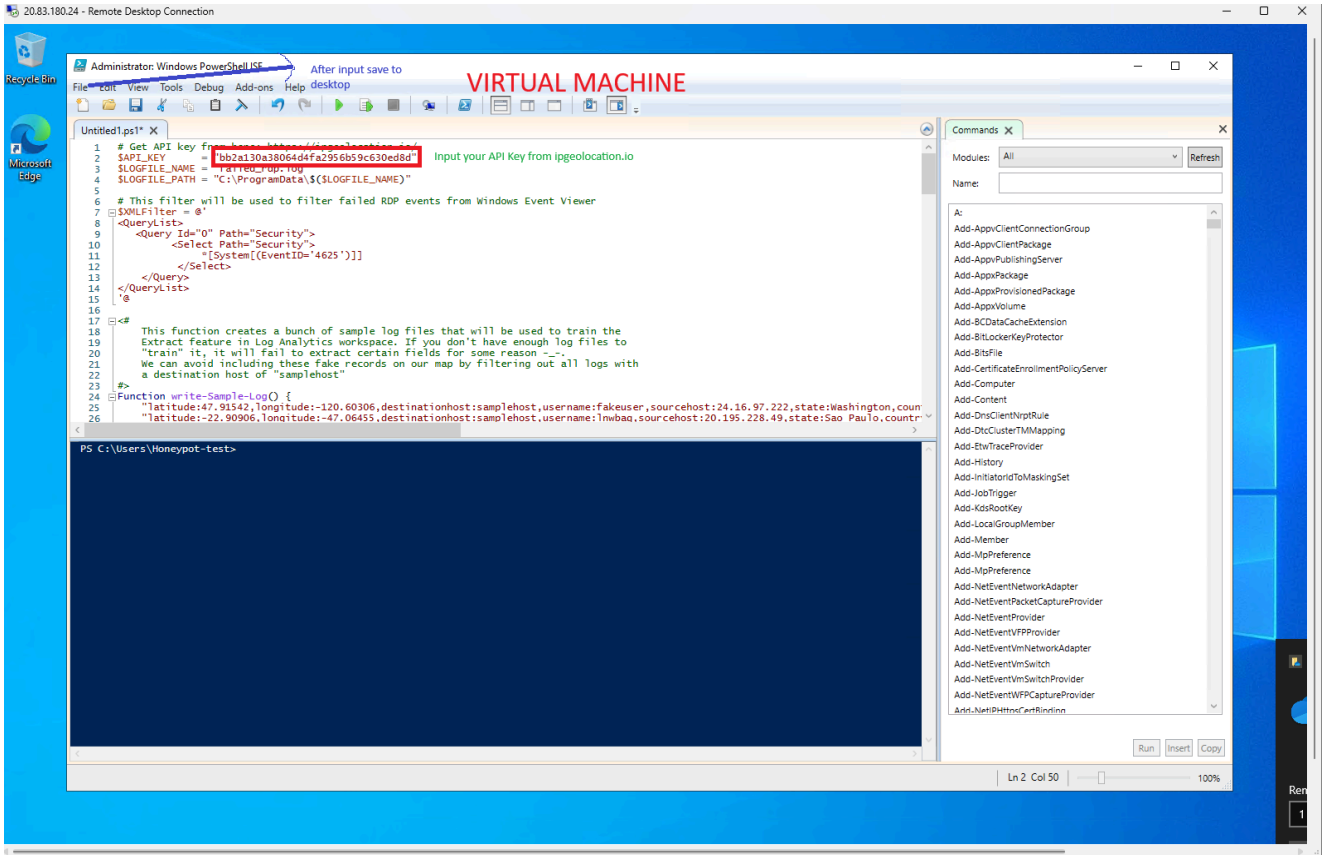
.



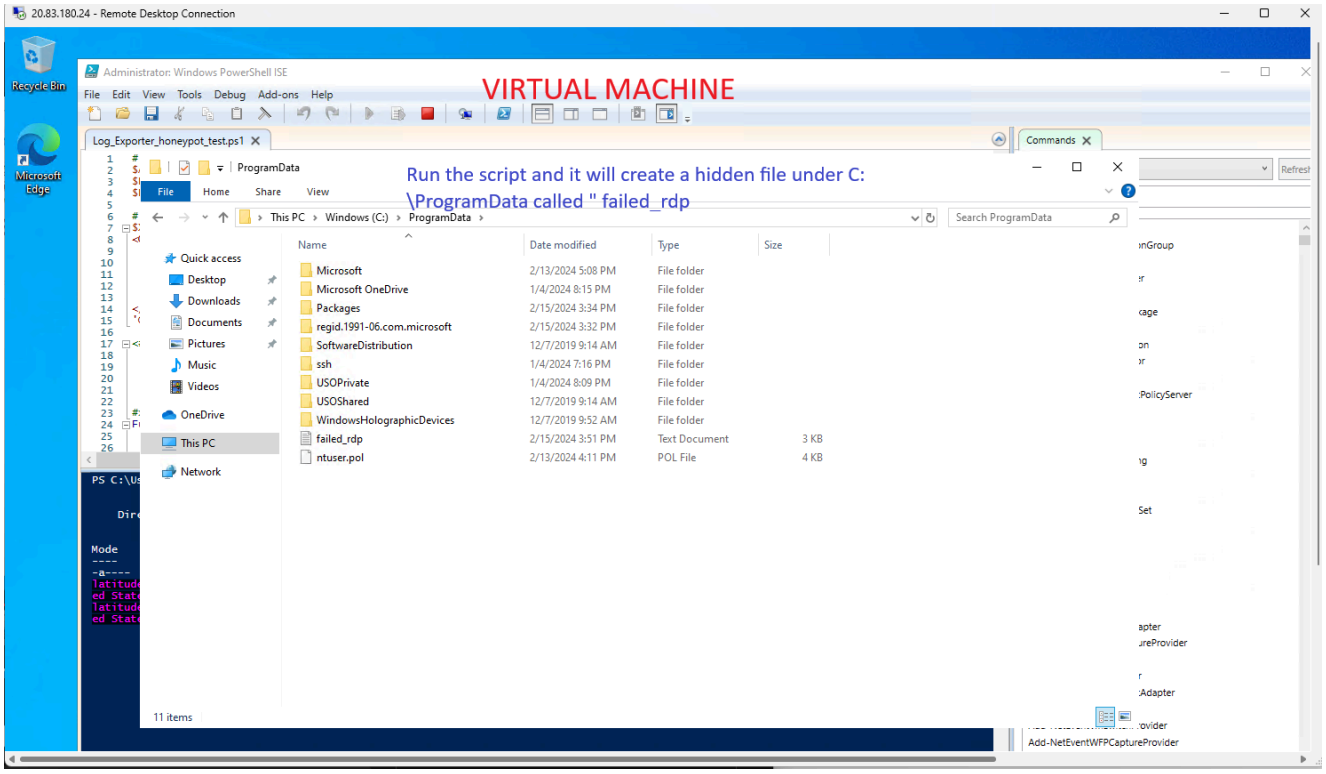
Once you have the script needed, there is one thing we need before running that script and that is an API Key. We use the site https://ipgeolocation.io/ to generate our key.



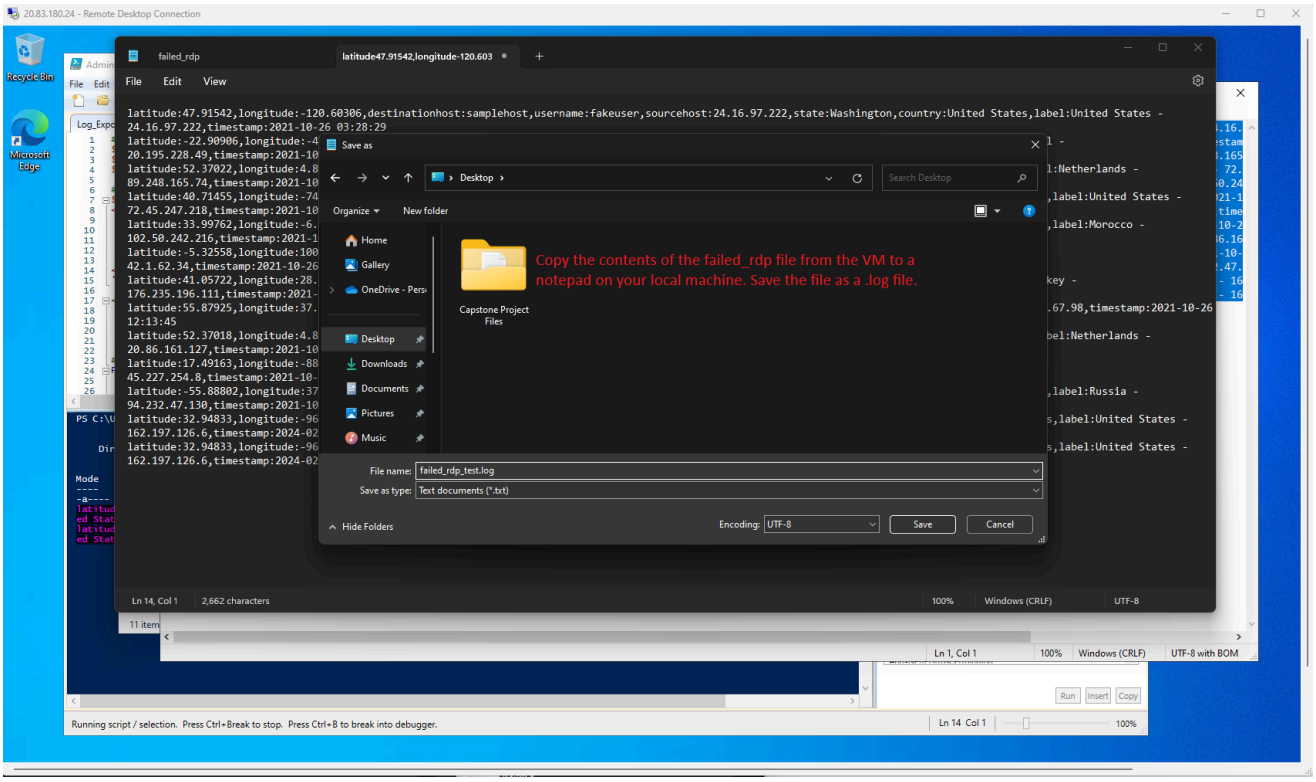
On the VM, open PowerShell ISE and copy and paste the script acquired previously. Under the ”$API_KEY” section, input your API Key from the ipgeolocation site, and select “run” to verify everything is working properly. Be sure to save this to your desktop.



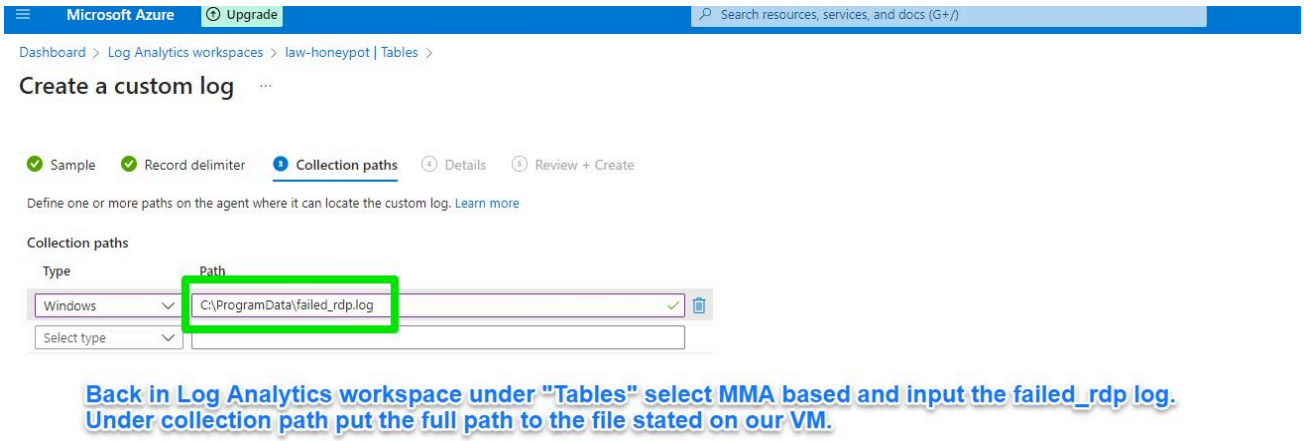
After running the script in PowerShell ISE, we then need to locate the hidden file it creates. This file is located at C:\ProgramData\failed_rdp. We will need this to create our custom log.



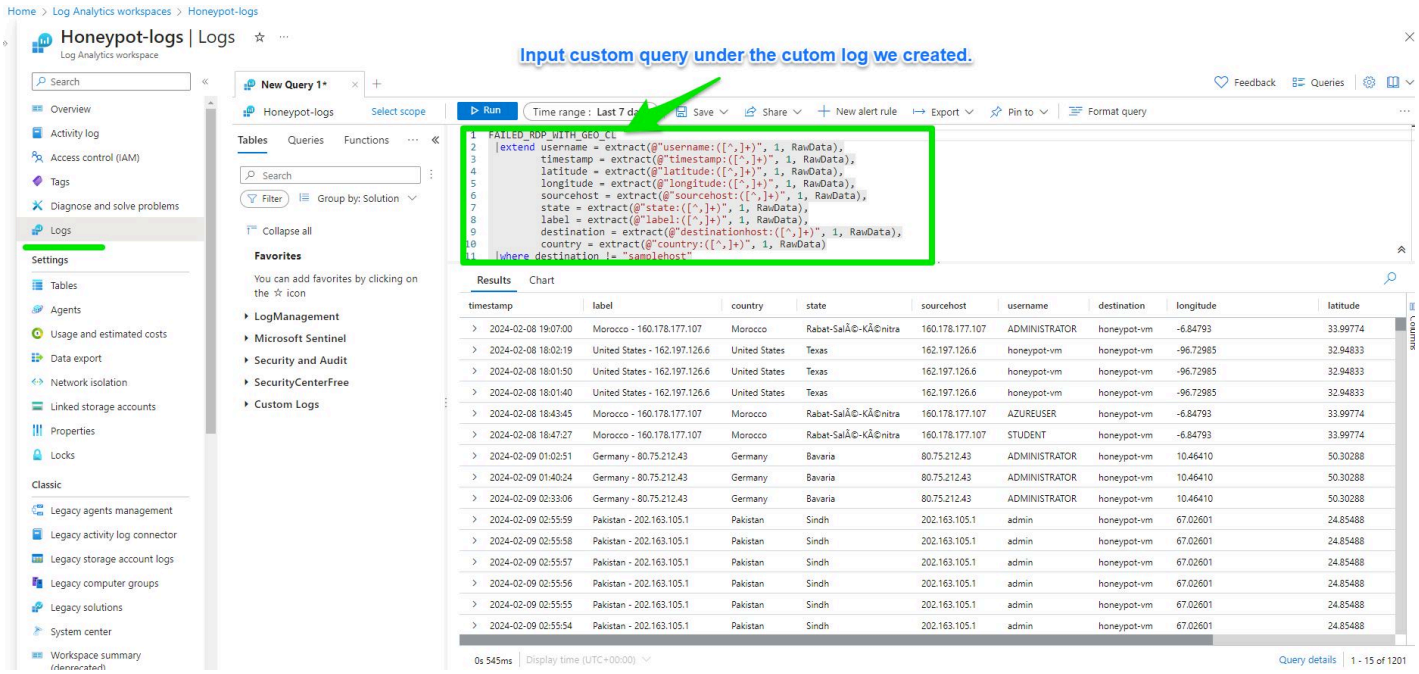
The next step is to copy the “failed_rdp” file from our VM to our local machine. We copied the file's contents from our VM and pasted it into a notepad file on our local machine, saving it as a .log file named “failed_rdp.log”.



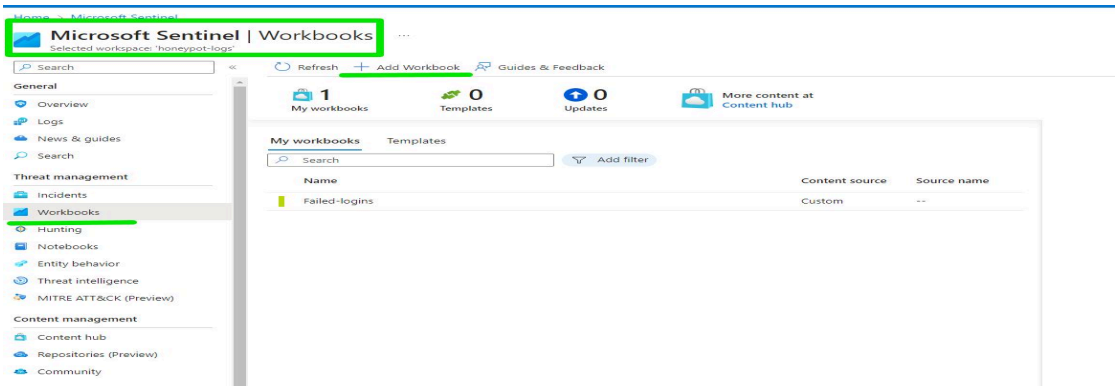
Back on our local machine, we need to create our custom log. In Log Analytics Workspace, navigate to Tables, select MMA, and upload the .log file we just created, and set the full path to the original file in the path section. Review and create and allow time for custom log to generate.



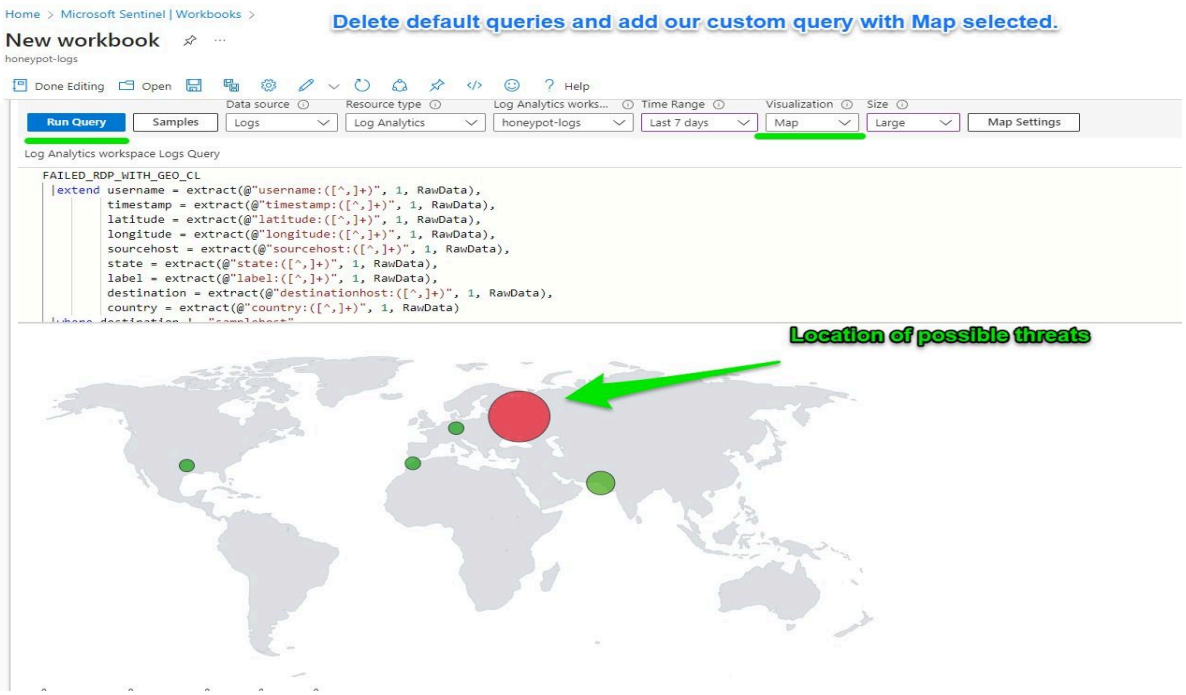
After our custom log is integrated, we then want to test it with a custom query and make sure the geographical location of failed login attempts into our honeypot or “attacks” are recorded. We can find that custom query here: https://github.com/AnastasiaCoskuner/Sentinel-Lab/blob/main/query_log. Be sure to start query line #1 with the correct name of the custom log created earlier. Select “Run” to test results.



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".



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.



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.