Home Labs for Cybersecurity I:
Creating a Honeynet and SIEM System in Microsoft Azure

Adapted from Log(N) Pacific Cybersecurity Masterclass and Internship

By Michael P. Matusek, MNS

Updated September 2024

Acknowledgements: All of the work herein is a full writeup and technical manual aligned with the Leveld Masterclass authored by Mr. Josh Madakor. Josh is an expert in IT and an exceptional instructor. His teaching style is easy to follow, and he ensures students learn relevant, up-to-date skills highly sought after by technology firms. Through his company, Log (N) Pacific, Josh offers a comprehensive Cybersecurity Training and Internship program that provides hands-on experience and practical knowledge, preparing students to excel in real-world environments. He delivers incredible value to his students, making him a true asset for any employer looking for top-tier candidates.

This technical manual is meant to provide support and an easy to follow walkthrough for the Honeynet & SOC Lab provided via https://joshmadakor.tech/cyber/.

Please note that that this resource will undergo periodic review and updates every 4 – 6 months as there are updates in Microsoft Azure, or as the course is updated.

Update Notes: You will notice that a fair number of screenshots contain references to LotR themed resources, vnets, virtual machines, and more. These names are simply shown this way because during the update of this manual I got bored with the original names and wanted to do something that related the experience to the Lord of the Rings. The original resource group was called The West to make reference to the men of the west, and the attack resource group was obviously placed in the so-called 'Black Land' on the plains of Golgoroth.

Contents

Exercise 1: Resource Group – Windows and Linux VM Creation

We deploy a Honeynet Lab in Microsoft Azure which is equipped with a vulnerable resource group containing two workstations: (1) Windows 10 (windows-vm) and (2) Ubuntu 22.04 (linux-vm).

Our main resource group will be 'RG-Cyber-Lab' wherein we have the placed both workstations; our windows-vm and our linux-vm which are vulnerable to attacks from the resource group 'RG-Cyber-Lab-Attacker' which will play the part of our attack vm. When the lab goes live to the public internet we will update this topology and the attack vectors will come from the world wide web. Rogue agents will most likely attack via open port 3389 to attack the Windows Machine and via open port 22 to attack the Linux Machine.

Embedded within the windows-vm we will setup a simple SQL server to serve as another attack target for those outside of our network. So, our first step is to set up these virtual machines, which are shown below in their respective dashboards.

Setting Up The Resources and Virtual Machines:

1. With your new Azure Subscription search for "Virtual Machines" and click '+Create' and complete the following for the Windows-VM
    a. Select or create your tenant subscription:
        i. Name: Azure Subscription 1
    b. Create a new resource group (mine was 'RG-Cyber-Lab')
    c. Name your virtual machine instance (mine was 'Windows-vm')
    d. Choose a platform, for our purposes the Windows 10 instance is what we will use for this lab.
    e. Select the location. NOTE: Make sure you select this location for everything you create or you will be unable to connect different VMs to the same resources/networks, etc.
    f. Select an eligible disk size with at least 2 vcpus and not less than 8 Gib for processing.
    g. Set the username and password for the instance. For mine I chose something easy to remember: labuser, with password 'Cyberlab123!'
    h. Click the agreement indicating that you have a valid 'Windows 10' license
    i. Click Next: Disks
    j. Click Next: Network
    k. Create a new VNET and use whatever naming convention you want. I chose "LAB-VNET" do not change anything else, just click OK
    l. You should be ready to "Review + Create" the VM instance.
    m. Once everything is validated, click "Create" and wait for deployment.
    n. Once deployed, click on the VM and you should see something similar to this. If you do not have a public IP address, you probably need to edit settings and/or delete the instance and recreate it.

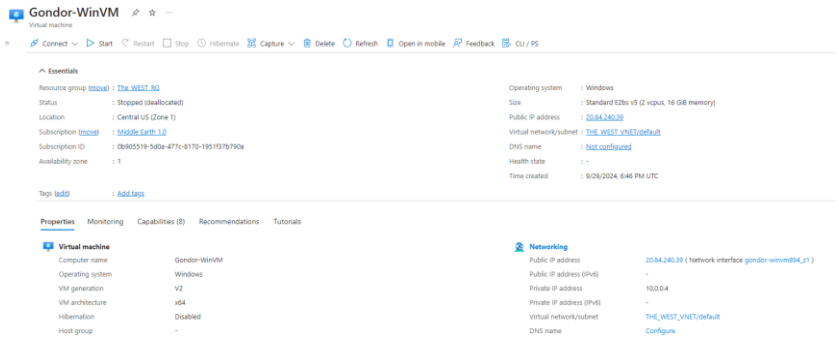


Figure 1-Dashboard for the Windows 10 Virtual Machine embedded in the 'The_WestRG'

2. Create a Linux Instance using the same setup as above with some minor changes.
    a. Select the same subscription, resource group, location, and Vnet as you did with the Windows instance.
    b. Choose Ubuntu 22.04 for your operating system.
    c. Make sure your new VM is in the same region and choose a similar disk size for this instance.
    d. For authentication, choose “password” and enter an appropriate username and password. Mine was ‘labuser’ and ‘Cyberlab123!’
    e. Make sure everything else looks good and click “REVIEW+CREATE”
    f. Once validation is complete click “Create” and wait for deployment.
    g. Click on the newly created VM and you should see something like this:

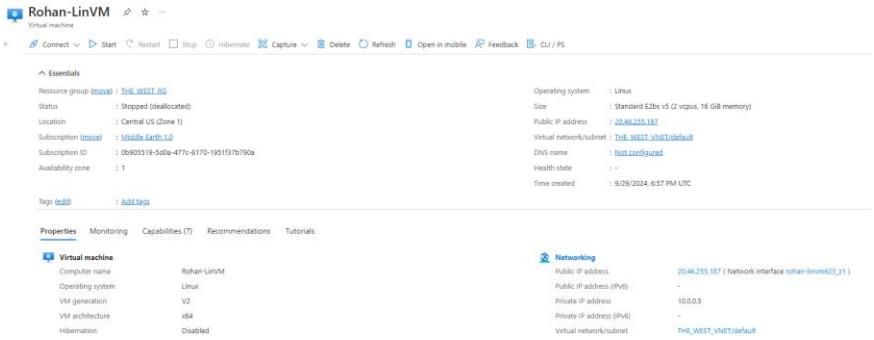


Figure 2 - Dashboard for the Ubuntu 22 Virtual Machine embedded in the 'RG-Cyber-Lab'

Exercise 2: Network Security Group – Inbound Rules

The next phase is to adjust the network security group to establish a baseline of security threats for a system that is completely open and vulnerable to the attacks from the public facing network (i.e. the world-wide web). Navigate to Network Security Groups within Azure and you should see both of your Network Security Groups for the two instances you created before. Click on one of the two and add a new inbound rule.

1. Click “Settings” and then “Inbound Rule”
2. Deleted inbound rule ‘RDP’ or ‘SSH’ depending on which instance you are working on at the moment.
3. For both instances we add a custom rule with
    a. Any source
    b. Any source port range
    c. Any destination
    d. Any service
    e. Any destination port
    f. Any protocol
    g. And choose ‘Allow’
    h. Priority set to 1.

As a side task, we also disabled all Windows Defender Firewall services within the VM within the Windows Machine itself. This will prevent the system from automatically blocking any connections automatically. We can later check the security and system logs with failed attempts from a third resource we will call Attack-vm once we have successfully setup and activated the SQL server in the Windows-vm.

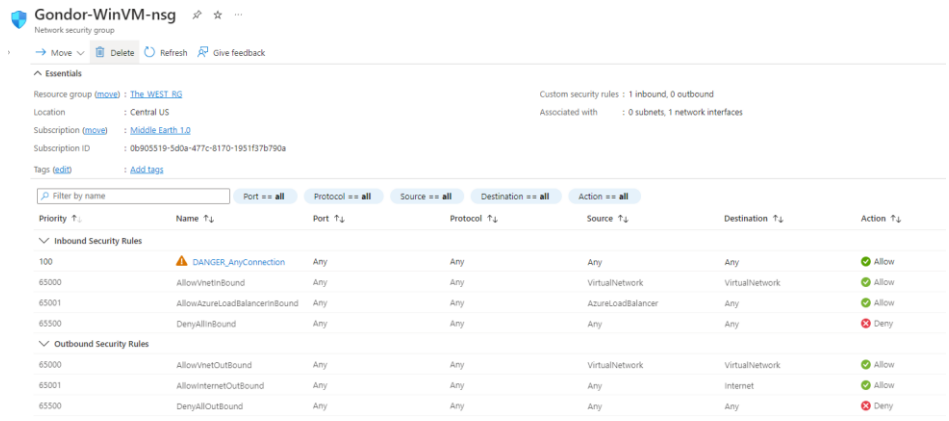


Figure 3 - NSG for Windows-vm

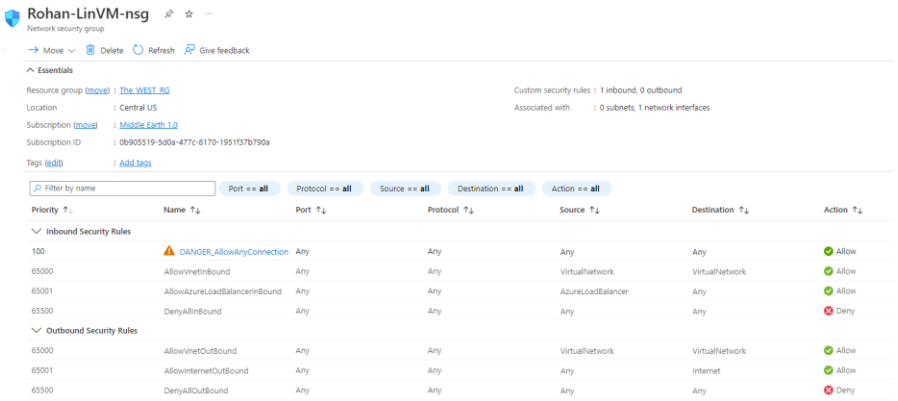


Figure 4 - NSG for Linux-vm

Exercise 3: SQL Server Creation and Logging For SQL Server

We now remote into the Windows-vm station to install the SQL Server Evaluation (located at https://www.microsoft.com/en-us/evalcenter/evaluate-sql-server-2019). We also need to manually install the SSMS (SQL Server Management Studio) in order to manage the SQL server (obtained at https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms). Once completed, we must enable the logging for SQL and port those logs into the Windows Event Viewer and verify that SQL logging is taking place successfully.

At this time we will also test connection to the Linux-vm to make sure it is accessible on the internet. If we want to fully test the vulnerabilities of these machines, we must make sure that they are full accessible. Eventually we will be forwarding logs into the Azure Log Analytics workspace to track attack vectors both geographically and via a Security Events and Incident Management system in Microsoft Sentinel. It is important to test everything before we get to that point, so we know it all functions the way we hope.

Installing SQL Server Evaluation

1. Connect and login to the Windows-vm by utilizing Microsoft Terminal Services Client (mstsc) using IP: <your_ip_address>
    a. Username: labuser
    b. Password: Cyberlab123!
2. If you haven't already done so, disable the Windows Defender Firewalls. Type wf.msc in the search bar and
3. Navigate to https://www.microsoft.com/en-us/evalcenter/evaluate-sql-server-2019 and download the installation agent. You will need to fill out a form for this download to take place.
    a. Open the downloaded file and click "Download Media" in the SQL Server 2019 Wizard.
    b. Choose to download an ISO to your desktop for easiest access.
    c. Navigate to the downloaded folder where the file has been downloaded.
    d. Right-click on the file and click 'mount'
    e. In the Installation Wizard, click "Installation"
    f. Click "New SQL Server stand-alone installation or add features to an existing installation"
    g. Follow the installation instructions
        i. Product Key → Evaluation "Next"
        ii. Click "I accept the license and terms"
        iii. Click Next on Microsoft Update
        iv. In "Feature Selection" click "Database Engine Services" and then NEXT
        v. In "Instance Configuration" click NEXT
        vi. In "Server Configuration" click NEXT
        vii. In "Database Engine Configuration" do the following:
            1. Click "Mixed Mode"
            2. Using default 'sa' account set the password as "Cyberlab123!" as we have done for the VM instance to make authentication easier for this portion of the lab.
            3. Click "Add Current User"

viii. Click “Install” in the “Ready to Install” page

4. Install SQL Server Management Studio
    a. Navigate to https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms and click “Download SQL Server Management Studio”
    b. Once downloaded, navigate to the file and open it
    c. Click “Install” and allow the installation process to complete.
    d. Click “Restart” in order for the installation to complete.
    e. Reconnect to the Windows-vm by using the Microsoft Terminal Services Client once again.

Enabling SQL Server Logging

1. If you want a resources to learn about this process you can find it here.
2. In the Windows Search Bar search for “Event Viewer”
3. Expand Windows Logs
    a. Click on Security
        i. Just take a moment to get familiar with this feature. We will need this later and it will help you once you start the logging process in Azure.
4. Right-click on the Windows Icon and search for regedit and hit “Run”
5. Browse to “HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog\Security”
6. Right-click on “Security” and click on “Permissions”
7. Click “Add” and type “NETWORK SERVICE” in the field “Enter the object names to select”
8. Click “Check Names” to verify and hit OK
9. Click the “Allow” choice for “Full Control” and then APPLY, then OK.

Configure the Audit Object Access in Windows

1. Scroll down to “Configure the Audit Object Access Setting in Windows using auditpol” in the same webpage for the previous step.
2. Copy the Windows Command Prompt
3. Open a Terminal with Administrator Privileges and paste the prompt and hit enter.

Testing and Enabling Auditing Configuration

1. Open the SQL Server Management Studio in the Windows-vm
2. Select SQL Server Authentication from the “Authentication” drop down menu.
3. Use the credentials we set up earlier for the SQL Server.
4. Click “Trust Server Certificate” before attempting to connect to avoid authentication issues.
5. Once connected, highlight the name of the server and right-click to select “properties” in the menu.
6. In the properties menu:
    a. Choose “Security”
    b. Under “Login Auditing” check “Both failed and successful logins”
    c. Click OK
7. Right-click the server and choose “Restart”
8. Click “Yes” in the prompt
9. Disconnect from the server and attempt to reconnect using a bad password a few times to generate some logs.

10. Navigate to the Event Viewer select “Application” under Windows Logs
11. The failed login attempts will be ID 18456; you should see failed logon IDs as you did and potentially others as well.

Exercise 4: Verify Connection to Linux VM

This is relatively straight-forward, we just want to verify that we can successfully login to the Linux-vm in order to generate syslog entries for any brute force attempts that we may encounter while the VM is active and open to the public facing network. Turn on your Linux-VM

To do this first, we test connectivity by choosing to PING the client from our local machine. We can do so by opening “terminal” in administrative mode and running the prompt: ping 20.46.255.187 and you will see the following results:

```
PS C:\Users\Mike> ping 20.46.255.187

Pinging 20.46.255.187 with 32 bytes of data:
Reply from 20.46.255.187: bytes=32 time=62ms TTL=44
Reply from 20.46.255.187: bytes=32 time=66ms TTL=44
Reply from 20.46.255.187: bytes=32 time=73ms TTL=44
Reply from 20.46.255.187: bytes=32 time=63ms TTL=44

Ping statistics for 20.46.255.187:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 62ms, Maximum = 73ms, Average = 66ms
```

Figure 5 - Ping results for Linux-vm

Next we want to connect directly to the instance, and we do so by entering the command:

ssh labuser@<your ip address>

After hitting enter you will be asked to trust the server, type ‘yes’ and hit enter. At this point you will be prompted to enter the password, which is Cyberlab123! You should see the following upon successful login:

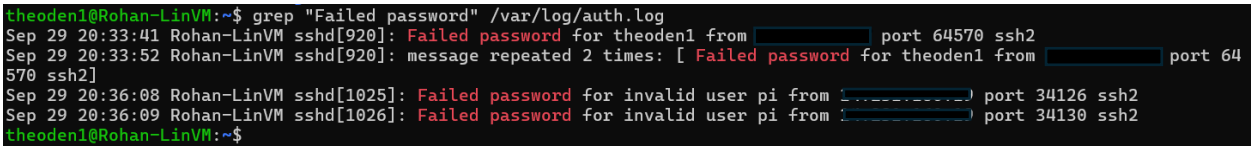
```
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.5.0-1025-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Sun Sep 29 20:28:38 UTC 2024

  System load:  0.0               Processes:             129
  Usage of /:   5.3% of 28.89GB   Users logged in:       0
  Memory usage: 2%                IPv4 address for eth0: 10.0.0.5
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Sun Sep 29 18:58:59 2024 from 97.86.197.9
To run a command as administrator (user "root"), use "sudo <command>"
See "man sudo_root" for details.
```

Figure 6 - Successful Linux VM connection

The logging files are located in the directory /var/log for Linux Machines. We will be collecting information from syslog, but we can also check the auth.log file in the /var/log directory for failed and successful login attempts. By entering ‘cat /var/log/lastlog’ we can see the last successful logins. Go ahead and exit and attempt to reconnect using three failed passwords. Then reconnect with a successful login attempt and check the logs by using the following command:

```
grep “Failed password” /var/log/auth.log
```

Here are my results from my failed attempts:



Clearly we have failed login attempts from labuser, which is the user we created, as well as from some user ‘pi’ which must be an attempt from a random attack on the public network.

Exercise 5: Creating Attack Machine

We will create another Windows virtual instance in a region external to the US and this will be nestled in a new resource group we will call 'RG-Cyber-Lab-Attacker." The working virtual net will be named 'LAB-VNET-ATTACKER' and the associated virtual machine will be titled 'Attack-vm'. From this attack machine we will simulate brute force login attempts on windows-vm via port 3389, on linux-vm via port 22, and on the SQL server. We will then analyze the logs within these machines in order to confirm that attempted logins are occurring in connection with the attempts from the Attack-vm.

Setup the Attack-vm

1. Navigate to Virtual Machines within Azure.
2. Click 'create' to build a new virtual machine instance and choose 'Azure virtual machine.'
3. Leave the subscription as 'Azure Subscription 1'
4. Create a new resource group (ours is called 'RG-Cyber-Lab-Attacker'
5. Choose a random location outside of the US. We chose '(Asia Pacific) Australia East'
6. Select 'Windows 10 Pro, version 22H2 – x64 Gen2'
7. Select the same disk size as previously, if available. Otherwise make sure it is not a B-series and make sure it has at least 2 vcpus and enough memory.
8. The administrator account will have these credentials (for our purposes):
    a. Username: labuser
    b. Password: Cyberlab123!
9. In networking, create a new VNET (we titled ours 'LAB-VNET-ATTACKER')
10. Click 'Review+Create' and wait for validation.
11. Click 'Create' when the validation is completed.
12. Once deployment has completed, attempt logging to verify successful deployment. Below is the VM dashboard.

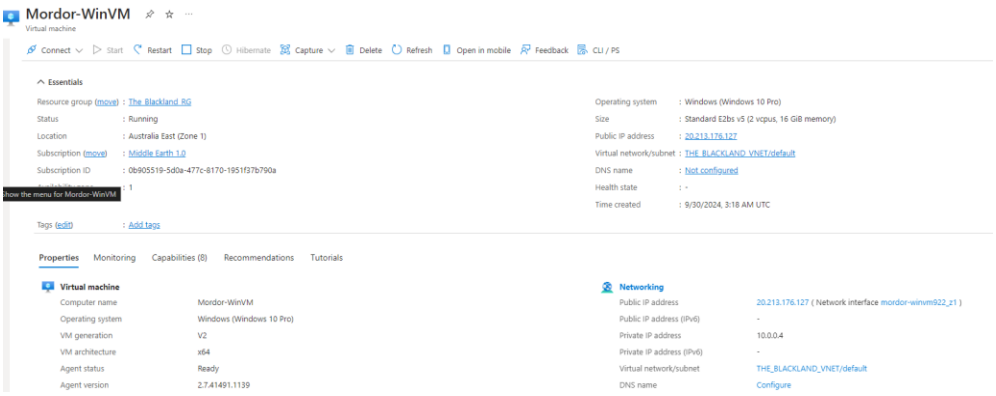


Figure 7 - The dashboard for the attacker instance titled 'Attack-vm'

Exercise 6: Generating Simulated Attacks

At this stage we are ready to use the attack virtual instance 'Attack-vm' to attack both 'Windows-vm' and 'Linux-vm' and then look at the associated logs. To do this, retrieve the IP addresses for the Windows-vm, Linux-vm, and recall that the SQL server is embedded in our Windows-vm instance.

Attack Mode – Generating Failed Login Attempts in Logs

IMPORTANT: Start all three Virtual Instances to complete this step.

Generate some failed RDP logs against the 'Windows-vm' by logging into 'Attack-vm' and attempt to use Microsoft Terminal Service Client to remote login to 'Windows-vm' with bad credentials. Repeat this step at least 3 to 4 times with the wrong username and password, to generate enough failed login attempts.

Generate some failed SSH logs against the 'Linux-vm' by logging into 'Attack-vm' and attempt to use PowerShell to remote login to 'Linux-vm' with bad credentials. Repeat this step at least 3 to 4 times with the wrong username and password, to generate enough failed login attempts.

Install SQL Server Management Studio on the 'Attack-vm' instance and attempt to login to the SQL Server from this instance using bad credentials. Do this a few times to create failed login attempts in the Application Logs within the 'Windows-vm' instance.

Logout of the 'Attack-vm' instance and shut it down for the remainder of this lab. You may continue using this instance as often as you would like to test different attack vectors on the 'RG-Cyber-Lab' and this should help create some recognition for when we start to ingest logs into the Azure Log Analytics Workspace and create incidents in Microsoft Sentinel.

Admin Mode – Analyze Failed Login Attempts in Windows, SQL, and Linux

Login to the 'Windows-vm' from your own workstation using Microsoft Terminal Service Client and navigate to the Event Viewer. Inspect the failures and successes from the Security Log for the RDP attempts and from the Application Log for the SQL Server. Here is a helpful list of EventIDs:

EventID	Log	Description
4625	Security	Failed login attempts for RDP
4624	Security	Successful login attempts in Windows
18456	Application	Failed login attempts for SQL
18454	Application	Successful login attempts for SQL

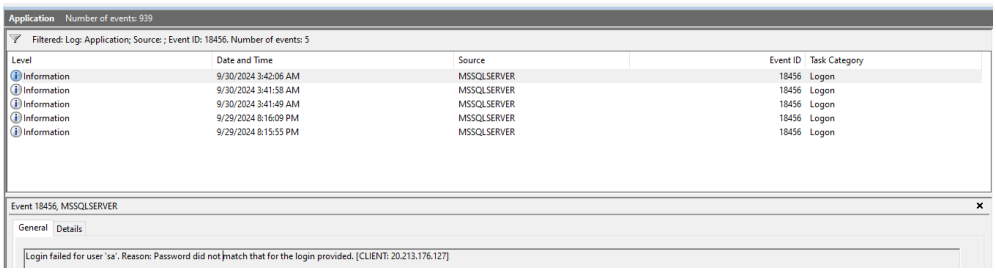


Figure 8 - Filtered Application Log for EventID 18456 which is a SQL failed login attempt. This shows that there are at least 5 failed attempts.

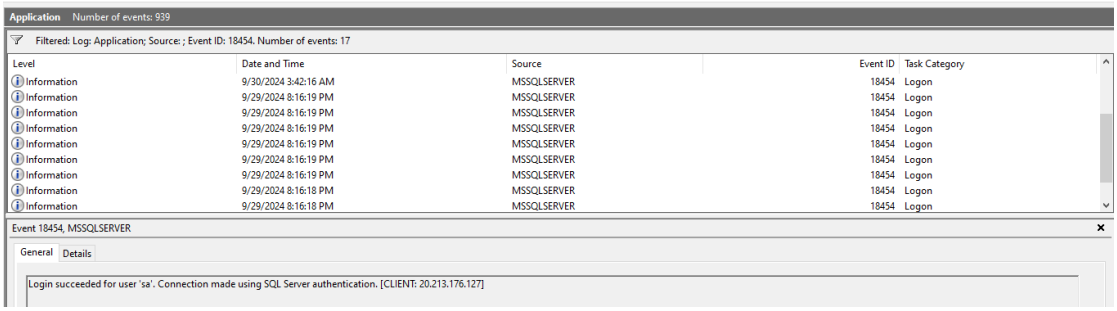


Figure 9 - Successful login attempts for the SQL server using EventID 18454

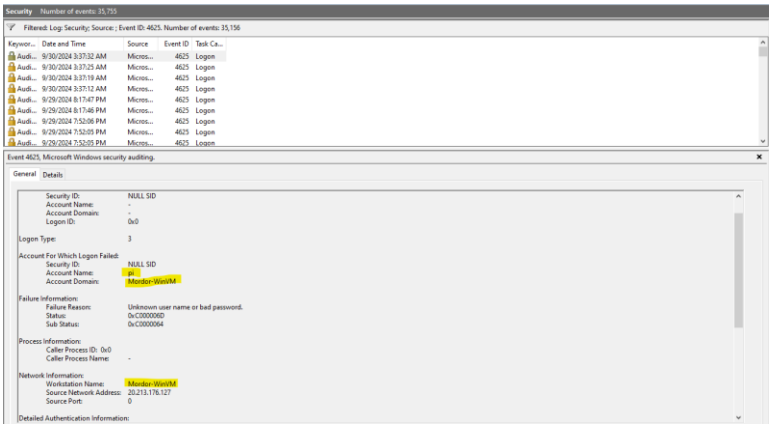


Figure 10 - Failed login attempts via port 3389 by filtering for EventID 4625. Notice the Account Domain is 'Attack-vm'

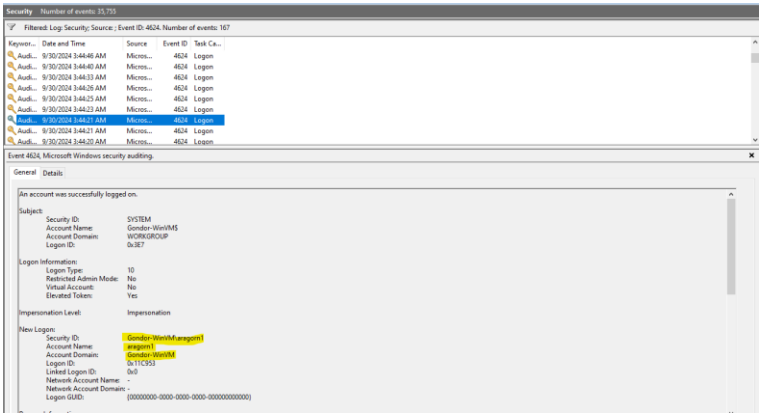


Figure 11 - Successful login attempts. The EventID 4624 generates multiple successful login attempts from both the System, Users, and potentially attackers. Notice the 'labuser' has a successful login.

Login to the 'Linux-vm' using SSH and check the auth.log file for failed and successful logs using these commands:

`cat /var/log/auth.log | grep password`

`cat /var/log/auth.log | grep Accepted`

When you do this do not be surprised to find that there may be far more failed login attempts from unrecognized IP addresses. These represent attempts from users on the public network using the bruteforce attack vector.

```
theoden1@Rohan-LinVM:~$ cat /var/log/auth.log | grep password
Sep 29 18:58:56 Rohan-LinVM sshd[1548]: Accepted password for theoden1 from 97.86.197.9 port 63845 ssh2
Sep 29 20:28:38 Rohan-LinVM sshd[786]: Accepted password for theoden1 from 97.86.197.9 port 64505 ssh2
Sep 29 20:33:41 Rohan-LinVM sshd[920]: Failed password for theoden1 from 97.86.197.9 port 64570 ssh2
Sep 29 20:33:52 Rohan-LinVM sshd[920]: message repeated 2 times: [ Failed password for theoden1 from 97.86.197.9 por
t 64570 ssh2]
Sep 29 20:34:05 Rohan-LinVM sshd[922]: Accepted password for theoden1 from 97.86.197.9 port 64667 ssh2
Sep 29 20:36:08 Rohan-LinVM sshd[1025]: Failed password for invalid user pi from 14.231.160.19 port 34126 ssh2
Sep 29 20:36:09 Rohan-LinVM sshd[1026]: Failed password for invalid user pi from 14.231.160.19 port 34130 ssh2
Sep 30 03:39:46 Rohan-LinVM sshd[787]: Failed password for invalid user pickle from 20.213.176.127 port 50545 ssh2
Sep 30 03:39:53 Rohan-LinVM sshd[787]: Failed password for invalid user pickle from 20.213.176.127 port 50545 ssh2
Sep 30 03:39:58 Rohan-LinVM sshd[787]: Failed password for invalid user pickle from 20.213.176.127 port 50545 ssh2
Sep 30 03:40:22 Rohan-LinVM sshd[789]: Failed password for invalid user dread from 20.213.176.127 port 50595 ssh2
Sep 30 03:40:29 Rohan-LinVM sshd[789]: Failed password for invalid user dread from 20.213.176.127 port 50595 ssh2
Sep 30 03:40:34 Rohan-LinVM sshd[789]: Failed password for invalid user dread from 20.213.176.127 port 50595 ssh2
Sep 30 03:48:11 Rohan-LinVM sshd[805]: Failed password for invalid user pi from 42.2.78.143 port 48440 ssh2
Sep 30 03:48:12 Rohan-LinVM sshd[806]: Failed password for invalid user pi from 42.2.78.143 port 48446 ssh2
Sep 30 04:02:44 Rohan-LinVM sshd[829]: Accepted password for theoden1 from 97.86.197.9 port 4477 ssh2
theoden1@Rohan-LinVM:~$
```

Figure 12 - All login attempts within the auth.log filed on our 'Linux-vm' instance. Note the failed attempts from 20.213.176.127 which is the IP for 'Attack-vm'

```
theoden1@Rohan-LinVM:~$ cat /var/log/auth.log | grep Accepted
Sep 29 18:58:56 Rohan-LinVM sshd[1548]: Accepted password for theoden1 from 97.86.197.9 port 63845 ssh2
Sep 29 20:28:38 Rohan-LinVM sshd[786]: Accepted password for theoden1 from 97.86.197.9 port 64505 ssh2
Sep 29 20:34:05 Rohan-LinVM sshd[922]: Accepted password for theoden1 from 97.86.197.9 port 64667 ssh2
Sep 30 04:02:44 Rohan-LinVM sshd[829]: Accepted password for theoden1 from 97.86.197.9 port 4477 ssh2
theoden1@Rohan-LinVM:~$
```

Figure 13 - Accepted password logins for 'Linux-vm' with user 'labuser' from the logs within auth.log

Exercise 7: Microsoft Entra ID – Assigning Roles and Triggering Activities

We will now use Microsoft Entra ID to create a new user for each of the following roles:

- Tenant-level global reader
- Observer Subscription Reader
- Resource Group Contributor

Configure and Observe Tenant-Level Global Reader

Create a user within Azure Entra ID **(username: globalreader <yourname>)**

- Assign Tenant-Level Global Reader to this new user
- In a new browser/incognito, log in as **globalreader<yourname>** and observe result of being a Tenant Level "Global Reader"
- Close browser/incognito when satisfied

Configure and Observer Subscription Reader

Back in main browser, create another user within Entra ID **(username: subreader<yourname>)**

- Assign Subscription-Level Reader to this new user
- In a new browser/incognito, log in as **subreader<yourname>** and observe result of being a Subscription Level "Global Reader"
- Close browser/incognito when satisfied

Configure and Observe Resource Group Contributor (like an admin)

Back in main browser, create another user within Entra ID **(username: rgcontributor<yourname>)**

- Create a new resource group called "Permissions-Tester"
- Assign Resource Group-level Contributor
- For our resource group (RG-Cyber-Lab), assign Contributor Permissions
- In a new browser/incognito, log in as **rgcontributor<yourname>_** and observe result of being a Subscription Level Reader
- Observe the result of being a Resource Group Level Contributor

When you are satisfied you are free to delete these users and groups as they serve no purpose for the remainder of this lab.

Exercise 9: Logging and Monitoring: GeoIP Ingestion and Log Analytics Workspace

We have now reached the point where we want to configure logging and monitoring within our Honeynet. We will leverage the powerful tools within Azure to ingest all of our logs into a single location and from there we will be able to configure a SIEM (Security Incident and Event Management) using Microsoft Sentinel to generate automated Incidents and Alert for our environment.

Setup Log Analytics Workspace

1. Navigate to the github repository located at https://github.com/joshmadakor1/Cyber-Course-v2/blob/main/Sentinel-Maps(JSON)/geoip-summarized.csv
2. Download the raw file to your computer and make sure you note its directory location.
3. Navigate to Log Analytics Workspaces and click Create
4. Choose the standard resource group we have been working with (mine is RG-Cyber-Lab)
5. Name your workspace something unique (mine is LAW-Cyber-Lab-01)
6. Click review and create and await validation. Once validation is completed click 'Create'

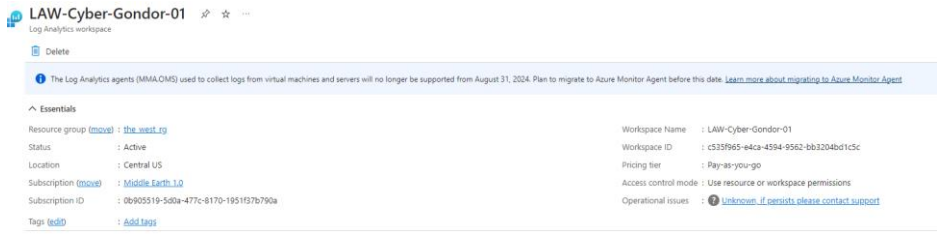


Figure 14 - Dashboard for newly created Log Analytics Workspace

Setup Microsoft Sentinel and Watchlists

1. Navigate to Microsoft Sentinel
2. Select the new Log Analytics Workspace you just created
3. Click 'Add'
4. After deployment navigate back to Microsoft Sentinel and click the Log Analytics Workspace you just added.
5. In the left menu select 'Configuration' followed by 'Watchlist'
6. Click new and be sure to enter the information exactly as seen here:
    a. General
        i. Name: geoip
        ii. Description: <none>
        iii. Alias: geoip
    b. Source
        i. Source type: Local File
        ii. File Type: CSV file with header (.csv)
        iii. Number of lines before row with headings: 0
        iv. Upload File: geoip-summarized.csv (from downloaded file)
        v. SearchKey: Network
    c. Validate that you've entered everything correctly and then click 'Create'

d. Wait for the watchlist to be generated. You can select the watchlist and a window will show status.

Figure 15 - Watchlist creation and progress for completion.

To verify that this has successfully been added to our environment navigate over to Log Analytics Workspace and select our new workspace we just created. In the menu on the left, select 'Logs' and close the 'Queries Hub' window. In the query window, enter `_GetWatchlist("geoip")` and await results.

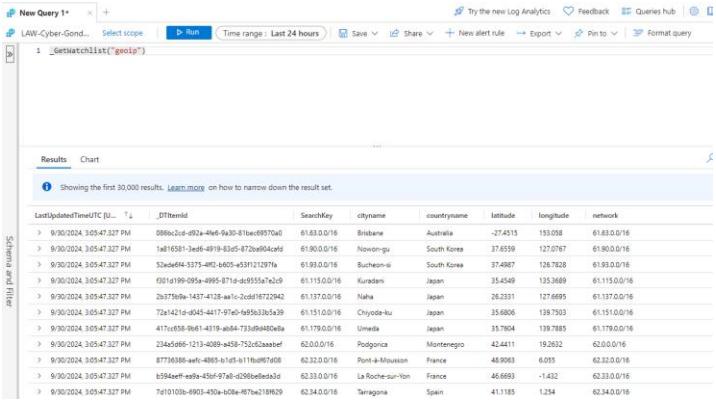


Figure 16 - First successful query in Log Analytics Workspace generating results from the newly created watchlist in Microsoft Sentinel

Exercise 10: Enabling Defender for Cloud, Log Collection for VMs and NSGs

We want to setup Microsoft Defender for Cloud which will automatically provision the Virtual Machines that will allow them to forward logs into our Log Analytics Workspace. ***For this part of the lab navigate to your virtual machines and make sure they are active.*** Navigate to Microsoft Defender for Cloud; you can find this by using the search bar in your Azure portal or you can navigate to 'Home>More Services' and under 'Hybrid+multicloud' choose 'Microsoft Defender for Cloud'

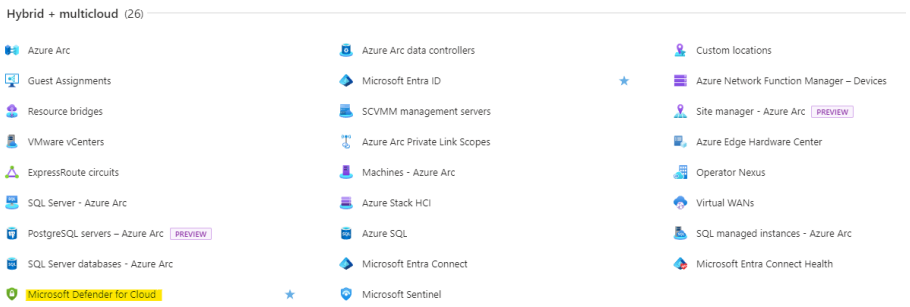
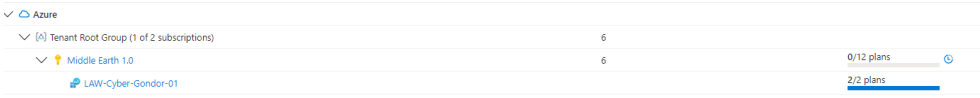


Figure 17 - Location for Microsoft Defender for Cloud under the All Services page

Enabling Microsoft Defender for Cloud for Log Analytics Workspace

After navigating to the main page for Microsoft Defender for Cloud, complete the following steps:

1. Select 'Management' from the left menu pane and the select 'Environment Settings'
2. On the list under 'Azure' expand both the Tenant, and the Subscription.
3. Click the 'Context Menu' for the Log Analytics Workspace (i.e. click the ••• menu) then edit settings.
4. Toggle both 'Servers' and 'SQL servers on machines' to the on position
5. Click save at the top
6. Navigate to 'Data Collection' in the left menu pane
7. Select 'All Events' and then 'save'
8. Return to 'Microsoft Defender for Cloud | Environment Settings'
9. Click refresh and you should see the new status of the Log Analytics Workspace (shown below)



Enable Microsoft Defender for Cloud for Azure Subscription

1. In 'Environment Settings' click the context menu and 'Edit Settings' for the Azure Subscription.
2. Under Cloud Workload Protection (CWP) turn on the following options:
    a. Servers
    b. Databases
    c. Storage Accounts
    d. Key Vaults

3. Click ‘Settings’ under Servers and select ‘Edit Configuration’ (NOTE: The Azure Monitor Agent is in deprecation, so that will not appear when editing this configuration)
4. Select ‘Custom Workspace’ and select your Log Analytic Workspace
5. Click ‘Apply’
6. Click ‘Continue’
7. Make sure you click ‘save’ when you reach the ‘settings | Defender Plans’ page.

Enable Microsoft Defender for Cloud Continuous Export

1. Click ‘Continuous Export’ from the left menu pane.
2. Select the ‘Log Analytics Workspace’ tab
    a. Ensure ‘Export Enabled’ is toggled in the on position.
    b. Under ‘Exported Data Types’ select the following options:
        i. Security Recommendations
            1. All recommendations selected
            2. Recommendations severity > Select All
            3. Include Security Findings > ‘Yes’
        ii. Secure Score
            1. Overall score, Control score
            2. Controls > All controls selected
        iii. Security Alerts > Low, Medium, High, Informational
        iv. Regulatory compliance > All Standards
        v. (**Updated**) Security attack paths > No selected risk levels (Note: You can enable this if you wish for your own personal explanation, but for the purpose of the lab you can leave this unselected.
    c. Under ‘Export Configuration’ select your resource group
    d. Under ‘Export Target’ select your Log Analytics Workspace
3. Click ‘Save’ when you are done
4. Navigate to ‘Log Analytics Workspace’ and delete any additional workspaces that may have been created by default, just to clean your environment a little.

Once this is deployed you will be able to collect any relevant logs within the log analytics workspace for further analysis.

Exercise 11: Enable Log Collection for Virtual Machines and Network Security Groups

Ensure that both of your virtual machines within your main resource group are active and running. Both of these need to be active to automatically (or manually) install the logging agent within the environments. Before we can do this, however, we need to create and enable an Azure storage account for our environment.

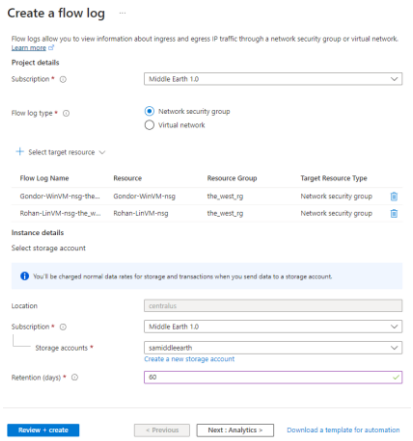
Setting up an Azure Storage Account

Navigate to 'All Services' and select 'Storage Accounts' from the 'Storage' menu which will lead you to the Storage Accounts dashboard. Click the '+ Create.' From this setup wizard do the following:

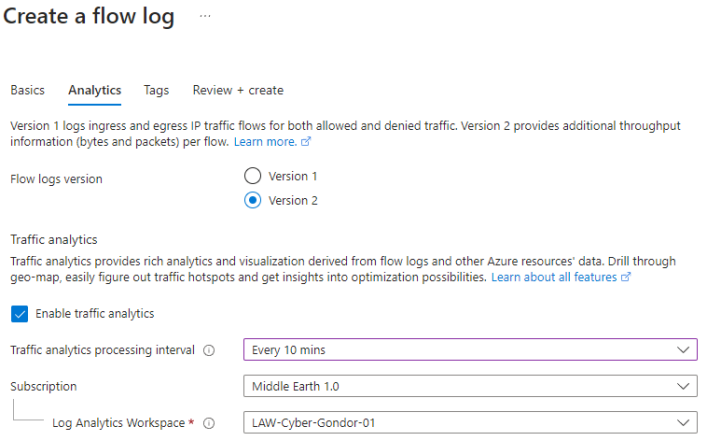
1. Select your appropriate Azure Subscription.
2. Select your appropriate Resource Group (use the same one you've used for the Windows and Linux virtual machines).
3. Choose a simple name for your storage account, something arbitrary (mine is samiddleearth).
4. Ensure that it is in the same location as all of your other resources.
5. Do not specify 'Primary Service' at this time.
6. Choose 'Standard' for performance.
7. Leave the default redundancy selected and ensure that the tickbox below is checked.
8. Select Review + Create and then wait for validation, once completed click 'Create'

Enable Flow Logs for Both Network Security Groups

1. Navigate to 'Network Security Groups' either from 'Home' or 'All Services' or using the search bar above.
2. Select either of the virtual machines to start and then click 'NSG Flow Logs' underneath 'monitoring' in the left menu pane.
3. Click 'Create Flow Log'
4. Ensure the proper subscription is selected and choose 'Network Security Group'
5. Click 'Select Target Resource' and choose both virtual machines then 'Confirm Selection'
6. Choose your storage account that you just created
7. For retention period, choose 60 days



8. Click ‘Next: Analytics’ and choose ‘version 2’ and enable traffic analysis
    a. Choose Every 10 minutes
    b. Ensure the proper subscription and Log Analytics Workspace are chosen



9. Finally, click ‘Review + Create’ and wait for validation.
10. Click ‘Create’ once validation completes.

What we have just enabled is the creation of a new table in our Log Analytics Workspace title ‘AzureNetworkAnalytics_CL, which is associated with our Network Security Group flow logs. Essentially, this allows us to monitor all incoming and outgoing traffic within our network security groups. It will then give us insights and the power to view malicious activities that may be penetrating our virtual networks and as a consequence infiltrating our devices.

Configure Data Collection Rules within Microsoft Sentinel

Navigate to Microsoft Sentinel and select the associated Log Analytics Workspace. From here, do the following:

1. Under ‘Content Management’ select ‘Content Hub’
2. Search for “Windows Security Events” from the search bar
    1. Select it and click install
    2. Navigate to your Windows virtual machine in a separate tab
    3. Under settings in the left menu pane click ‘Extensions + Applications’ you will notice the MicrosoftMonitoringAgent has automatically been provisioned

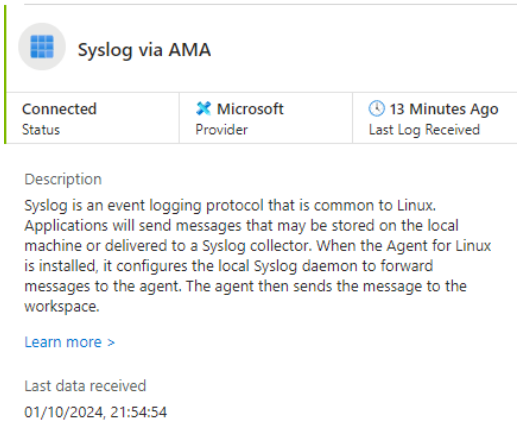
    4. Back in the ‘Content Hub’ click manage on the ‘Windows Security Events’ pane
    5. Select ‘Windows Security Events via AMA’ and then click ‘Open Connector Page’ in the menu pane
    6. Click ‘Create Data Collection Rule’
    7. Ensure the appropriate subscription and resource groups are selected
    8. Choose the appropriate Windows VM under your resource group (ignore the Attack VM)
    9. Choose ‘All Security Events' and then click ‘Review+Create’
    10. Wait for validation to complete and click ‘Create’
    11. If you go back to the ‘Extensions + Applications’ tab and click ‘Refresh’ you will see the creation of ‘AzureMonitorWindowsAgent’

Eventually the status will change from 'Transitioning' to 'Provisioning Succeeded'

3. Head back to the Sentinel Tab and refresh the page. You should see the following:

4. Navigate through Microsoft Sentinel once again to the Content Hub and search for 'Syslog'
    1. Navigate the results until you find 'Syslog' and select and install it
    2. Once installation is complete click 'Manage'
    3. Check 'Syslog via AMA' then click 'Open connector page'
    4. Click 'Create data collection rule'
        i. Select an appropriate name like 'DCR-Linux-vm'
        ii. Ensure the appropriate subscription and resource group are selected
        iii. Click 'Next: Resources' and choose the appropriate Linux VM
        iv. Choose 'LOG_DEBUG' for the 'LOG_AUTH' option only
        v. Review and create the rule
        NOTE: If you choose Ubuntu 24.04 this may fail, be sure that you have selected Ubuntu 22 to avoid this. You may need to recreate your Linux VM and then readdress your NSG flows and this step if this fails.
    5. Ensure that your AzureMonitorLinuxAgent has provisioned successfully in the 'Extensions+Applications' tab

    6. Ensure that your Syslog via AMA has status 'Connected'



At this point you should begin querying these logs within Log Analytics Workspace before moving on to the next lab. Navigate to your Log Analytics Workspace and enter your log query space. From there you can query the tables associated with our newly created Data Collection Rules. Namely, you can query the following tables in the query space:

- Syslog (Linux)
- SecurityEvent (Windows)
- AzureNetworkAnalytics_CL (NSG Flows)

You should end up with results similar to the figures below for each of the queries:

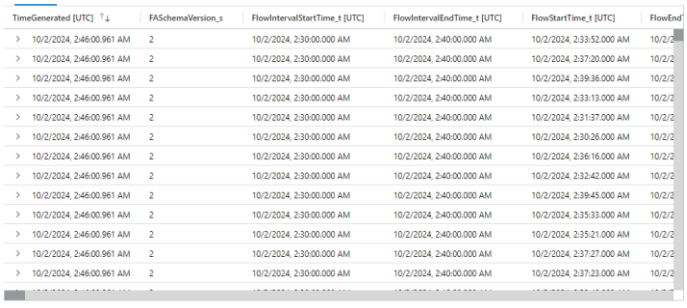


Figure 18 - Query for the NSG Flows using KQL command AzureNetworkAnalytics_CL

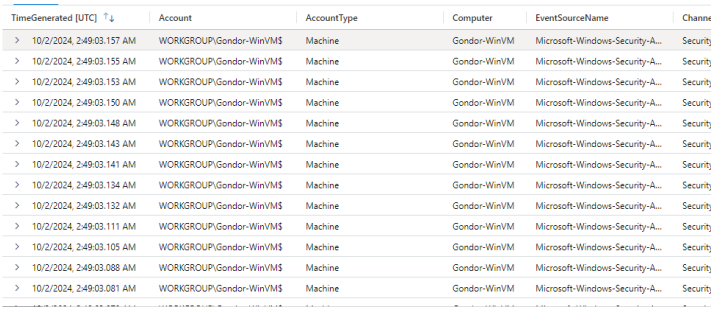


Figure 19 - Query for the Windows Security Events using KQL command SecurityEvent

You should also simulate a failed login attempt for both VMs and check query for those events within your Log Analytics Workspace. Your results should resemble these figures.

Facility	HostName	SeverityLevel	SyslogMessage	ProcessID
auth	Rohan-LinVM	info	Connection reset by authentica...	10312
auth	Rohan-LinVM	info	Failed password for theoden1 f...	10312
auth	Rohan-LinVM	info	Failed password for theoden1 f...	10312
auth	Rohan-LinVM	info	Failed password for theoden1 f...	10312

Figure 20 - Failed login attemps on Linux Virtual Machines. Query was Syslog | where ProcessID == 10312

>	10/2/2024, 2:57:11.566 AM	Mikes_PC\aragorn1	User	Gondor-WinVM	Microsoft-Windows-S
>	10/2/2024, 2:57:11.566 AM	Mikes_PC\aragorn1	User	Gondor-WinVM	Microsoft-Windows-S
>	10/2/2024, 2:57:07.846 AM	Mikes_PC\aragorn1	User	Gondor-WinVM	Microsoft-Windows-S
>	10/2/2024, 2:57:07.846 AM	Mikes_PC\aragorn1	User	Gondor-WinVM	Microsoft-Windows-S
>	10/2/2024, 2:57:01.380 AM	Mikes_PC\aragorn1	User	Gondor-WinVM	Microsoft-Windows-S
>	10/2/2024, 2:57:01.380 AM	Mikes_PC\aragorn1	User	Gondor-WinVM	Microsoft-Windows-S

Figure 21 - Failed login attempts to the Windows VM. Query was SecurityEvent | order by TimeGenerated desc | where EventID == 4625

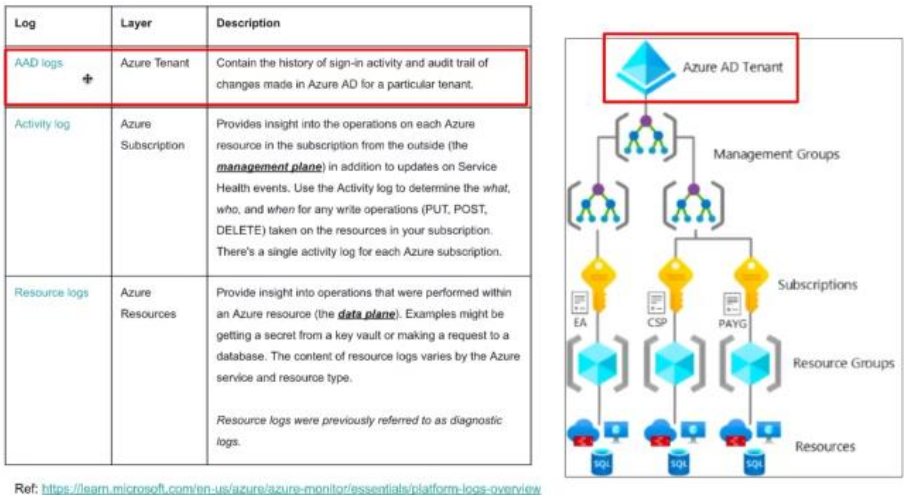
For a comprehensive overview and discussion of Kustos Query Language see appendix A. This resource is mentioned here as a subtle nod to Mr. Madakor's video in the masterclass 'KQL Deep Dive' and we will skip this section and move on to the next lab. Be sure to review the resources below as well for more training in Kustos Query Language (KQL).

- https://github.com/joshmadakor1/Cyber-Course/blob/main/KQL-Query-Cheat-Sheet.md
- https://learn.microsoft.com/en-us/training/modules/write-first-query-kusto-query-language/

Exercise 12: Tenant Level Logging

We will continue to ingest logs into our central logging system which is our Log Analytics Workspace in Microsoft Azure. Keep in mind that the whole point of this process is to centralize and automate the ingestion of logs for easy analysis and monitoring for our complete Honeynet environment. The goal is to eventually use the results of active threat intelligence through our monitoring efforts to harden the security of our system. We will use this pre- and post-hardening analysis to deepen our understanding of the cyber threat landscape and further expand our knowledge and skills as future cyber security analysts.

We have previously created logging rules for our network security groups, our virtual machines, and created associated geographic ip logging rules within Microsoft Sentinel. We now turn our attention to logging for our Tenant Subscription in Microsoft Azure itself.



Log	Layer	Description
AAD logs	Azure Tenant	Contain the history of sign-in activity and audit trail of changes made in Azure AD for a particular tenant.
Activity log	Azure Subscription	Provides insight into the operations on each Azure resource in the subscription from the outside (the ***management plane***) in addition to updates on Service Health events. Use the Activity log to determine the *what, who, and when* for any write operations (PUT, POST, DELETE) taken on the resources in your subscription. There's a single activity log for each Azure subscription.
Resource logs	Azure Resources	Provide insight into operations that were performed within an Azure resource (the ***data plane***). Examples might be getting a secret from a key vault or making a request to a database. The content of resource logs varies by the Azure service and resource type. *Resource logs were previously referred to as diagnostic logs.*

Ref: https://learn.microsoft.com/en-us/azure/azure-monitor/essentials/platform-logs-overview

Figure 22 - The hierarchy model for Logging within Microsoft Azure. We are currently at the Azure AD Tenant level. Note that Active Directory is now Entra ID within Azure.

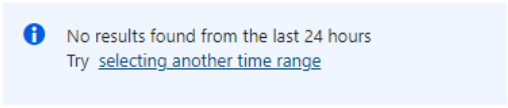
We will complete the following in this section of our lab:

1. Create Diagnostic Settings to Ingest Azure AD Logs
2. Create a dummy user in Azure Entra ID
3. Perform some actions and observe logging
    a. Mass sign-in failures (sign-in logs)
    b. Assignment of "Global Administrator" to user (audit logs)

Setup Logging for Azure Entra ID and Generate Some Logs

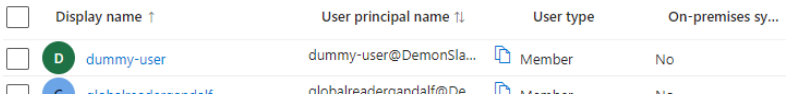
1. Navigate to Microsoft Entra ID
2. Scroll down to and select 'Diagnostic Settings' in the left menu pane
3. Click '+ Add diagnostic setting' and choose
    a. AuditLogs
    b. SigninLogs
4. Select 'Send to Log Analytics Workspace' and ensure the appropriate destinations are selected (i.e. your subscription and the correct resource group for this lab)

5. Navigate back to the Log Analytics Workspace and check that the tables have been created or you can also directly query them to see if there are results.
    a. Simply query with command 'AuditLogs' or 'SigninLogs'
    b. You should get this message if the table is successfully created

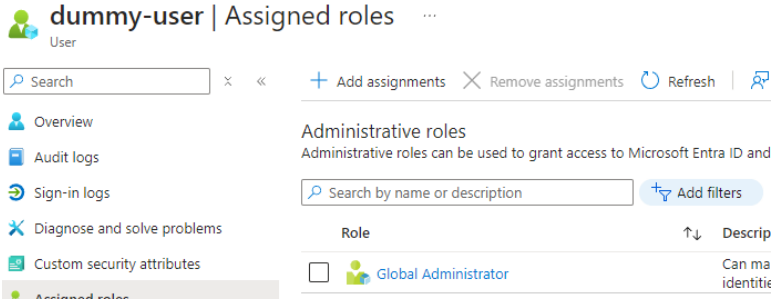


Creating a Dummy-User and Use it to Generate Audit and Sign-in Logs

1. Navigate back to Microsoft EntraID and perform the following steps to create a new 'dummy-user' with standard privileges:
    a. Click 'Users' from the left menu pane
    b. Click '+New User' in the user dashboard
    c. Under principal name type 'dummy-user'
    d. Under display name type 'dummy-user'
    e. The password is automatically created: <auto-generated>
    f. Ensure 'Account Enabled' is checked.
    g. Click 'Review+Create' and verify the information is correct, then 'Create'

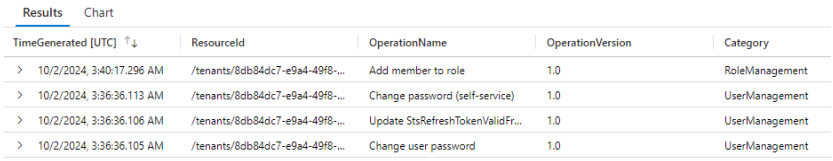


2. Open an incognito tab and navigate to portal.azure and login with the credentials given to you during user creation. Note that you will need to reset the password, so make sure you keep track of it. I chose 'CyberDummy123!' for the password for easy remembering.
3. Click on the new user 'dummy-user'
4. In the left menu pane select 'Assigned Roles'
5. Click '+ Add Assignment' and search for 'Global Administrator'
6. Check 'Global Administrator' and click 'Add'



You may need to click 'Refresh' to see the new assigned role.

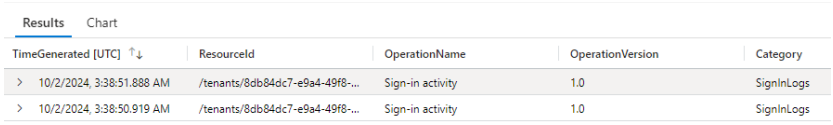
7. Finally, delete the dummy-user from EntraID (Note: The whole point of this exercise is to create logs and observe them)
    a. Navigate back to 'Default Directory | Users'
    b. Check 'dummy-user' and click delete
8. Navigate to the Log Analytics Workspace and query 'AuditLog' and 'SigninLog' to check for results



Results Chart

TimeGenerated [UTC]	ResourceId	OperationName	OperationVersion	Category
10/2/2024, 3:40:17.296 AM	/tenants/8db84dc7-e9a4-49f8-...	Add member to role	1.0	RoleManagement
10/2/2024, 3:36:36.113 AM	/tenants/8db84dc7-e9a4-49f8-...	Change password (self-service)	1.0	UserManagement
10/2/2024, 3:36:36.106 AM	/tenants/8db84dc7-e9a4-49f8-...	Update StsRefreshTokenValidFr...	1.0	UserManagement
10/2/2024, 3:36:36.105 AM	/tenants/8db84dc7-e9a4-49f8-...	Change user password	1.0	UserManagement

Figure 23 - AuditLog showing creation, role change, password change, and deletion of new user



Results Chart

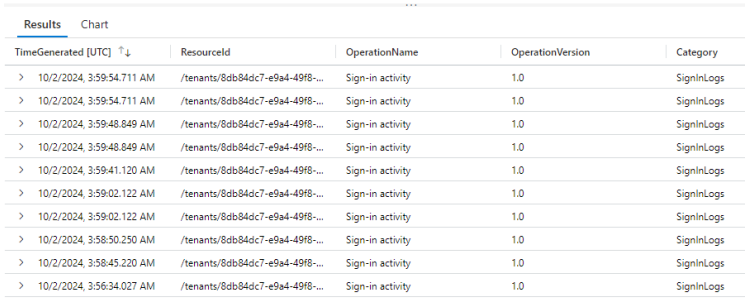
TimeGenerated [UTC]	ResourceId	OperationName	OperationVersion	Category
10/2/2024, 3:38:51.888 AM	/tenants/8db84dc7-e9a4-49f8-...	Sign-in activity	1.0	SignInLogs
10/2/2024, 3:38:50.919 AM	/tenants/8db84dc7-e9a4-49f8-...	Sign-in activity	1.0	SignInLogs

Figure 24 - SigninLog showing sign in activity for user dummy-user

Creating an Attacker User and Generating Brute Force Logs

1. Create a new user in EntraID titled ‘attack-user’ following the same instructions we used for the ‘dummy-user’ account.
2. Recall that a new password will need to be created for this user if you choose to login to Microsoft Azure with this account, choose something like ‘CyberAttack123!’ or ‘CyberLab123!’ for the password to easily remember it.
3. Navigate to an incognito Window in your browser and perform multiple brute force login attempts. Do this about 10 times to ensure logs are successfully created.
4. Head over to Log Analytics Workspace and run the SigninLogs query to check for the results. This takes time, so maybe take a break and come back.
   The following query gives the results in figure 25.

```
SigninLogs
| where ResultDescription startswith “Invalid”
```



Results Chart

TimeGenerated [UTC]	ResourceId	OperationName	OperationVersion	Category
10/2/2024, 3:59:54.711 AM	/tenants/8db84dc7-e9a4-49f8-...	Sign-in activity	1.0	SignInLogs
10/2/2024, 3:59:54.711 AM	/tenants/8db84dc7-e9a4-49f8-...	Sign-in activity	1.0	SignInLogs
10/2/2024, 3:59:48.849 AM	/tenants/8db84dc7-e9a4-49f8-...	Sign-in activity	1.0	SignInLogs
10/2/2024, 3:59:48.849 AM	/tenants/8db84dc7-e9a4-49f8-...	Sign-in activity	1.0	SignInLogs
10/2/2024, 3:59:41.120 AM	/tenants/8db84dc7-e9a4-49f8-...	Sign-in activity	1.0	SignInLogs
10/2/2024, 3:59:02.122 AM	/tenants/8db84dc7-e9a4-49f8-...	Sign-in activity	1.0	SignInLogs
10/2/2024, 3:59:02.122 AM	/tenants/8db84dc7-e9a4-49f8-...	Sign-in activity	1.0	SignInLogs
10/2/2024, 3:58:50.250 AM	/tenants/8db84dc7-e9a4-49f8-...	Sign-in activity	1.0	SignInLogs
10/2/2024, 3:58:45.220 AM	/tenants/8db84dc7-e9a4-49f8-...	Sign-in activity	1.0	SignInLogs
10/2/2024, 3:56:34.027 AM	/tenants/8db84dc7-e9a4-49f8-...	Sign-in activity	1.0	SignInLogs

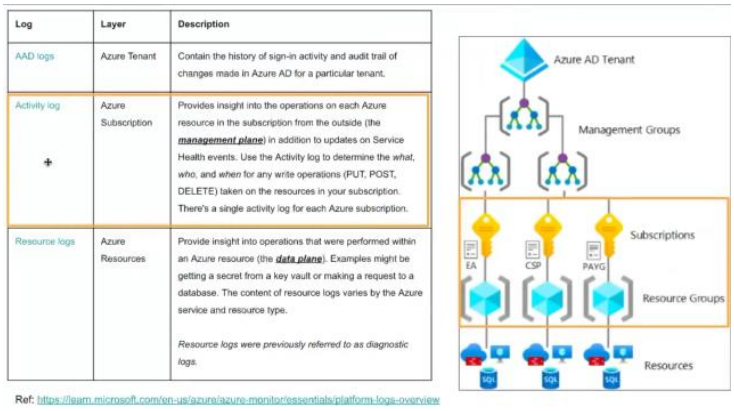
Figure 25

5. You may delete the user ‘attack-user’ as we will no longer need it going forward

At this stage you may wish to create a “Break Glass” account as a back-up Admin account for your Microsoft Azure environment. This is an important account to have within your IT systems in case something happens to your primary administrative account. If Microsoft Azure suspects malicious intent with your primary account, it may be deactivated and you will lose access to your Tenant. To set up this “Break-Glass” account simply navigate to Microsoft Entra ID and create a new user titled ‘break-glass’ and make sure to add a role in the ‘Assignments’ Tab by clicking ‘Add Role’ and searching for Global Administrator. I used the password “CyberLab123!” as usual.

Exercise 13: Subscription Level Logging

We now move to subscription level logging which really helps us nail down what resources have been added or deleted from our subscription. This is also referred to as Activity Logging in Microsoft Azure. Essentially, everything you do to manipulate or change your resources will be logged at this level. Keep in mind that we want to be able to monitor everything that is happening in our environment, so monitoring change in our environment is very important. If changes occur in an organization without our consent, we want to be able to pull up our logs and determine when and how those resources were changed.



Log	Layer	Description
AAD logs	Azure Tenant	Contain the history of sign-in activity and audit trail of changes made in Azure AD for a particular tenant.
Activity log +	Azure Subscription	Provides insight into the operations on each Azure resource in the subscription from the outside (the ***management plane***) in addition to updates on Service Health events. Use the Activity log to determine the *what, who, and when* for any write operations (PUT, POST, DELETE) taken on the resources in your subscription. There's a single activity log for each Azure subscription.
Resource logs	Azure Resources	Provide insight into operations that were performed within an Azure resource (the ***data plane***). Examples might be getting a secret from a key vault or making a request to a database. The content of resource logs varies by the Azure service and resource type. *Resource logs were previously referred to as diagnostic logs.*

Ref: https://learn.microsoft.com/en-us/azure/azure-monitor/essentials/platform-logs-overview

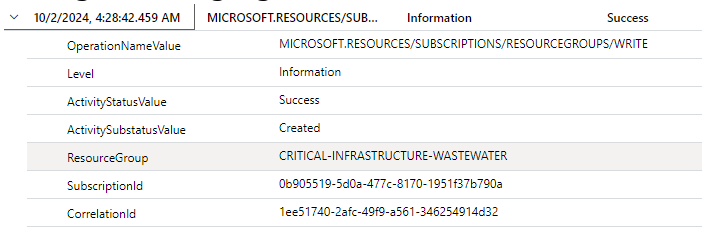
Figure 26 - We are now at the Subscription level of our hierarchy where we wish to log any changes to the resources within our subscription.

Export the Azure Activity Logs to Log Analytics Workspace

1. Navigate to Monitor and choose 'Activity Log' in the left menu pane.
2. Click "Export Activity Logs" at the top next to the icon
3. Click '+ Add diagnostic setting' and choose all log categories
4. Under 'Destination details' check 'Send to Log Analytics Workspace' and choose the appropriate workspace
5. Click 'Save'

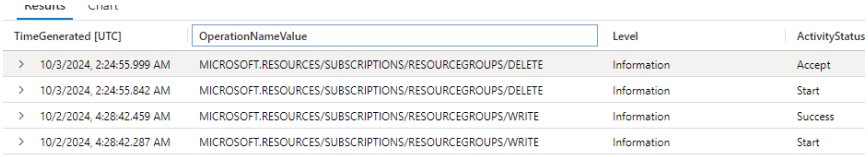
Create Logs by Adding New Dummy Resource Groups and then Delete Them

1. Navigate to resource groups and create two new resource groups.
    1. Their names can be completely arbitrary, just make sure you can identify each in your log queries.
2. Navigate to Log Analytics Workspace to query the activities you just completed to ensure the logs are being ingested.



10/2/2024, 4:28:42.459 AM	MICROSOFT.RESOURCES/SUB...	Information	Success
OperationNameValue	MICROSOFT.RESOURCES/SUBSCRIPTIONS/RESOURCEGROUPS/WRITE		
Level	Information		
ActivityStatusValue	Success		
ActivitySubstatusValue	Created		
ResourceGroup	CRITICAL-INFRASTRUCTURE-WASTEWATER		
SubscriptionId	0b905519-5d0a-477c-8170-1951f37b790a		
CorrelationId	1ee51740-2afc-49f9-a561-346254914d32		

3. You can also delete the newly created resource groups and verify in Log Analytics Workspace that this activity also shows



TimeGenerated [UTC]	OperationNameValue	Level	ActivityStatus
10/3/2024, 2:24:55.999 AM	MICROSOFT.RESOURCES/SUBSCRIPTIONS/RESOURCEGROUPS/DELETE	Information	Accept
10/3/2024, 2:24:55.842 AM	MICROSOFT.RESOURCES/SUBSCRIPTIONS/RESOURCEGROUPS/DELETE	Information	Start
10/2/2024, 4:28:42.459 AM	MICROSOFT.RESOURCES/SUBSCRIPTIONS/RESOURCEGROUPS/WRITE	Information	Success
10/2/2024, 4:28:42.287 AM	MICROSOFT.RESOURCES/SUBSCRIPTIONS/RESOURCEGROUPS/WRITE	Information	Start

4. Here are some useful queries for Azure Activity logs:
    1. Querying for the deletion of critical resource groups
       ```
       AzureActivity
       | where ResourceGroup startswith “critical-infrastructure-“
       | order by TimeGenerated
       ```
    2. Querying for changes to network security groups
       ```
       AzureActivity
       | where OperationNameValue ==
       “MICROSOFT.NETWORK/NETWORKSECURITYGROUPS/SECURITYRULES/WRITE”
       | order by TimeGenerated
       ```
    3. Deletion Activities with a certain timespan
       ```
       AzureActivity
       | where OperationNameValue endswith “DELETE”
       | where ActivityStatusValue == “Success”
       | where TimeGenerated > ago(30m)
       | order by TimeGenerated
       ```
    4. From Microsoft Defender for Cloud Security Events
       ```
       AzureActivity
       | where CategoryValue == “Security”
       ```
    5. Just stuff happening on the Management Plane
       ```
       AzureActivity
       | where CategoryValue != “Administrative”
       ```

Exercise 14: Enable Resource Level Logging

We will now enable logging for our storage account and creating a key vault. We will also logging for both of these resources to be ingested into our Log Analytics Workspace. At this level, we are ingesting logs for any and all of our resources so we can observe and monitor any changes or activity within our resources. We have previously done this for our virtual machines, and now we are configuring our storage account and key vaults.

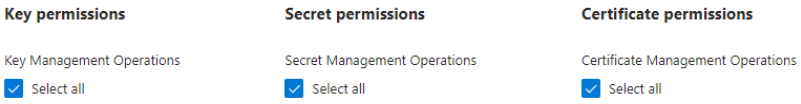
Enable Logging for Storage Accounts

1. Navigate to Storage Accounts in Microsoft Azure; you should have a storage account created from earlier.
2. Click on the storage account
3. Use the left menu pane to open 'Monitoring' and choose 'Diagnostic Settings'
    a. Click 'Blob' under your main storage account
    b. Click '+ Add diagnostic setting'
    c. Underneath 'Logs' choose audit
    d. Name it 'ds-storage-account' or something similar.
    e. Select 'Send to Log Analytics workspace' and make sure the appropriate options are chosen (Subscription and Log Analytics Workspace)
4. Navigate to your Storage Account and select 'Data Storage > Containers' and create a new container
    a. Name the container 'test'
    b. Click on 'test'
    c. Click 'upload' and drop any file you wish into the window
    d. Click 'Upload' to finish the process.

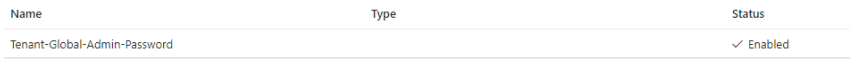
This should have generated a log for our storage account.

Create Azure Key Vault and Enable Logging

1. Navigate to the Key Vault dashboard in Microsoft Azure
2. Create key vault
    a. Choose the correct subscription and resource group
    b. Create a globally unique name for the key vault
    c. Place it in the same location as all of your other vital resources
3. Click 'Next'
4. Change the permission model to 'Vault Access Policy'
5. Click 'Review + Create'
6. Once validation is completed, click 'Create' and await deployment
7. Navigate to your newly created key vault and click 'Access Policies' in the left menu pane
    a. Check your name and click edit
    b. Verify that you have all the operational privileges you need



    c. You can close out once you've selected all (should be selected be default)
8. In the left menu pane, choose Objects > Secrets and create a new secret
    a. Name it 'Tenant-Global-Admin-Password'
    b. Set secret value to any password you please
    c. Choose 'Create'



9. Navigate to 'Monitoring > Diagnostic Settings' in the left menu pane
10. Click '+ Add diagnostic setting' and complete the following:
    a. Name it something like 'ds-azure-key-vault'
    b. Choose 'Audit' under Logs
    c. Choose 'Send to Log Analytics workspace' and choose the appropriate subscription and Log Analytics Workspace
    d. Click Save

You can add another secret if you wish, or navigate to your previously stored secret and view it. All of these activities should produce logs in our Log Analytics Workspace. Here are some helpful log queries for you once you've made some changes to the resources.

Storage Account Test Logs

```
// Authorization Error
StorageBlobLogs
| where MetricResponseType endswith "Error"
| where StatusText == "AuthorizationPermissionMismatch"
| order by TimeGenerated asc
```

Key Vault Test Logs

```
// List out Secrets
AzureDiagnostics
| where ResourceProvider == "MICROSOFT.KEYVAULT"
| where OperationName == "SecretList"

// Attempt to view passwords that don't exist
AzureDiagnostics
| where ResourceProvider == "MICROSOFT.KEYVAULT"
| where OperationName == "SecretGet"
| where ResultSignature == "Not Found"
```

For a more comprehensive list of potential queries see the appendices at the end of this manual.

Exercise 15: Utilizing Microsoft Sentinel as a Security Incident and Event Management system

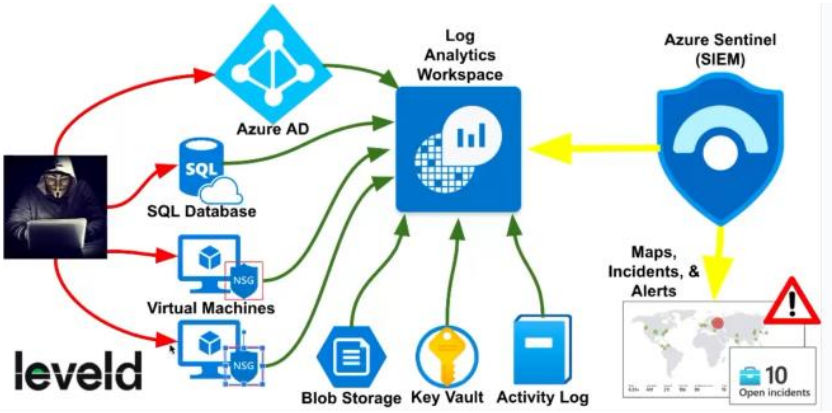
In this part of the lab we will use our geoip workbook in Sentinel to construct world maps to visualize the location of malicious events. We will also mock generate some of our own attack traffic in order to simulate and monitor how our system reacts. We will then create some manual alerts to trigger alerts and then, we will demonstrated some incident analysis in alignment with NIST 800-61.

Finally, we will open our Honeynet up to the public facing internet for a 24-hour period and record/manage incidents and activities for that period. This will trigger a period of hardening for our environment using Microsoft Defender for Cloud and configuring NIST 800-53 for regulatory compliance. We will then record/manage incidents and activities after another 24-hour period of the hardened environment.

Constructing the Sentinel Attack Maps

We will construct four maps that look at the following cases:

- Windows VM Attacks → Remote Desktop attacks; SMB attacks; General Authentication Failures
- Linux VM Attacks → SSH authentication failures
- Microsoft SQL Server (within Windows VM) → Authentication Failures
- Network Security Groups → Malicious Flows

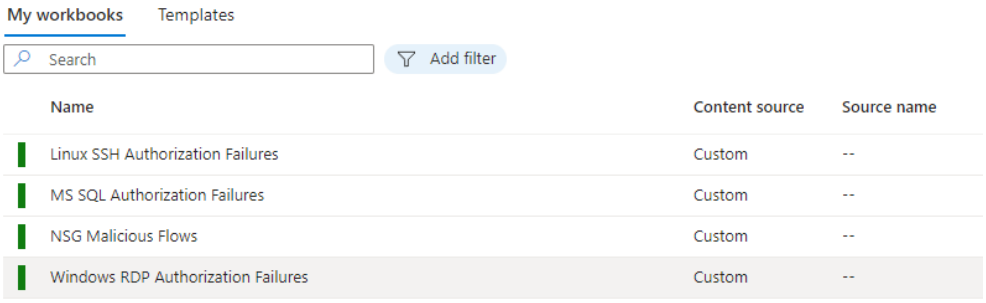


Navigate to the URL https://github.com/joshmadakor1/Cyber-Course-V2/tree/main/Sentinel-Maps(JSON) to obtain the mandatory JSON files for the World Maps. Download the following files to a local directory:

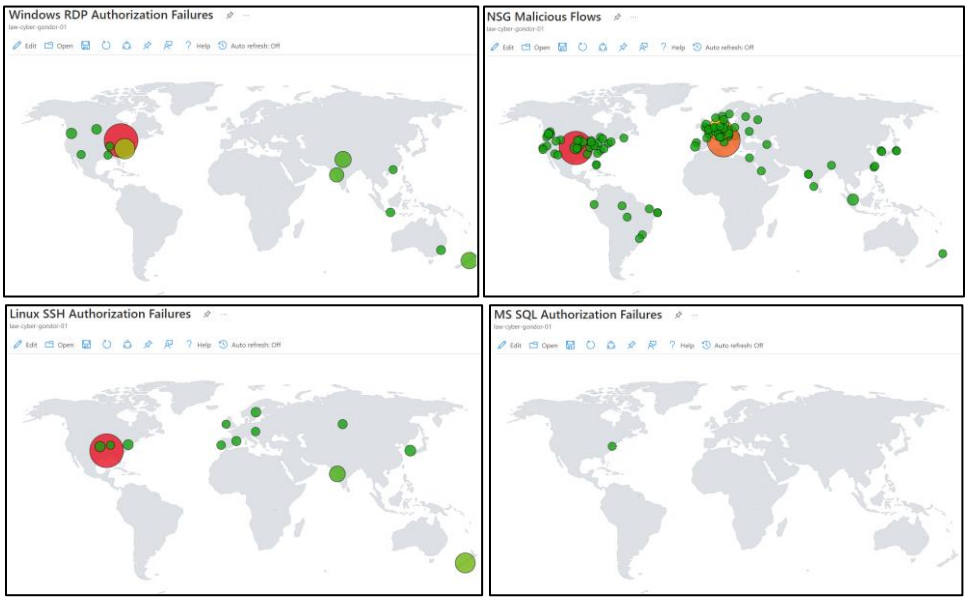
- linux-ssh-auth-fail.json
- mssql-auth-fail.json
- nsg-malicious-allowed-in.json
- windows-rdp-auth-fail.json

Alternatively you can open the four files and copy the json code within our workspace. For a detailed discussion of JSON files see the appendices at the end of this manual.

1. Navigate to Microsoft Sentinel and select your Log Analytics Workspace (do not choose the default listed)
2. Under 'Threat Management' in the left menu pane choose 'Workbooks'
3. Click '+Add Workbook' and select 'Edit'
    a. Remove the two automatically generated elements by selecting the ··· icon and choosing 'remove'
    b. Click '+Add' and choose 'Query'
    c. Click 'Advanced Editor' and paste the JSON raw file into the editor
    d. Click 'Done Editing'
    e. Click the save icon and name it something that indicates what kind of map it is.
4. Navigate back to 'Workbooks' and repeat step 3 to create the remaining maps.



To navigate back to any of your saved workbooks, go to Microsoft Sentinel | Workbooks and expand the right menu pane. Click on 'View saved workbook' to look at the workbooks. You should have something like these images for the Linux, Windows, and NSG world maps.



In general, there is a KQL query used to generate the world maps. The general form of the query which generates the maps looks similar to the following:

```
let GeoIPDB_FULL = _GetWatchlist("geoip");
let IpAddress_REGEX_PATTERN = @"\b\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}\b";
```

```
Syslog
| where Facility == "auth"
| where SyslogMessage startswith "Failed password for"
| order by TimeGenerated desc
| project TimeGenerated, SourceIP = extract(IpAddress_REGEX_PATTERN, 0, SyslogMessage), Destinati
onHostName = HostName, DestinationIP = HostIP, Facility, SyslogMessage, ProcessName, SeverityLevel,
 Type
| evaluate ipv4_lookup(GeoIPDB_FULL, SourceIP, network)
| project TimeGenerated, SourceIP, DestinationHostName, DestinationIP, Facility, SyslogMessage, Proces
sName, SeverityLevel, Type, latitude, longitude, city = cityname, country = countryname, friendly_locati
on = strcat(cityname, " (", countryname, ")");
```

Exercise 16: Manual and Automatic Alert Creation in Microsoft Sentinel

For this portion of the lab we will create a test alert for our SIEM in Microsoft Sentinel; the purpose behind this is to generate a simple rule that alerts us against any brute force logins for our Windows VM. We will create a manual alert named “TEST: Brute Force ATTEMPT – Windows” and then attempt to trigger this alert within our SIEM.

Creating a Manual Alert and Triggering it for Incident Generation

In Log Analytics Workspace query the following to see what kind of data is generated:

```
SecurityEvent
| where EventID == 4625
| where TimeGenerated > ago(60m)
| summarize FailureCount = count() by AttackerIP = IpAddress, EventID, Activity, DestinationHostName = Computer
| where FailureCount >= 10
```

Results | Chart

AttackerIP	EventID	Activity	DestinationHostName	FailureCount
8.142.83.116	4625	4625 - An account failed to log on.	Gondor-WinVM	32
4.184.136.102	4625	4625 - An account failed to log on.	Gondor-WinVM	48
97.86.197.9	4625	4625 - An account failed to log on.	Gondor-WinVM	20

As you can see there were a total of 3 different IP addresses that have attempted to brute force into our Windows VM. From these different originating IP addresses, there were a total of 100 failed login attempts (note that the failed login event ID # is 4625 as indicated earlier in the lab).

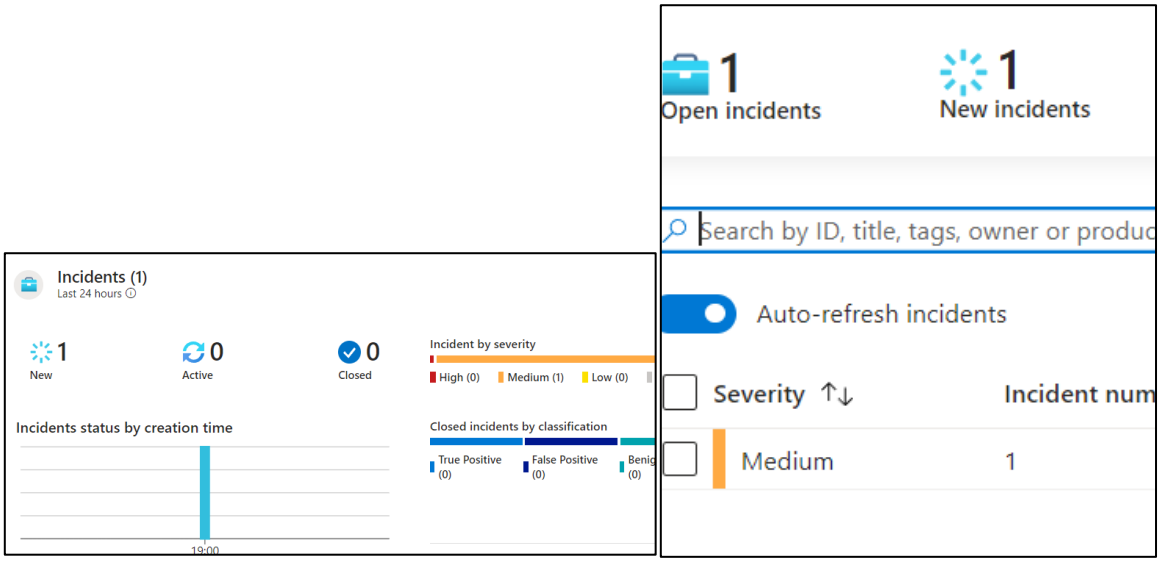
In Microsoft Sentinel, selecting your Log Analytics Workspace, navigate to Analytics under Configuration.

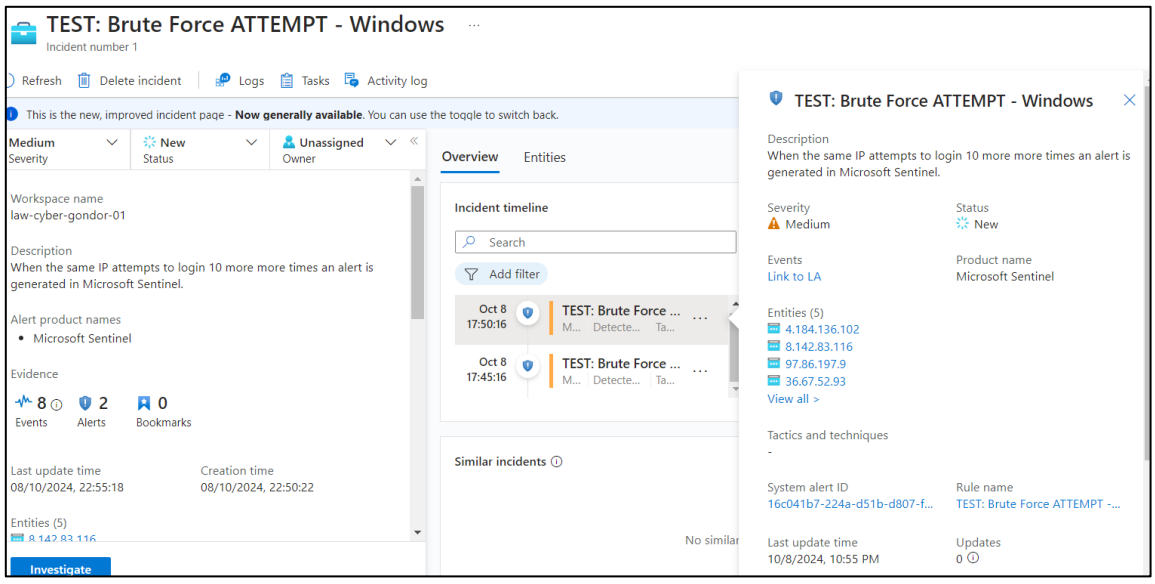
1. Select “+Create” and choose ‘Scheduled Query Rule’
2. Name it ‘TEST: Brute Force ATTEMPT – Windows’
3. Click ‘Set rule logic’ and paste the following Query into the window

   SecurityEvent
   | where EventID == 4625
   | where TimeGenerated > ago(60m)
   | summarize FailureCount = count() by AttackerIP = IpAddress, EventID, Activity, DestinationHostName = Computer
   | where FailureCount >= 10
4. Under ‘Entity Mapping’ and add the following new entities:
   a. IP > Address > AttackerIP
   b. Host > HostName > DestinationHostName
5. For Query Scheduling choose ‘Rune query every 5 minutes’
6. For Incident settings, leave ‘Create incidents from alerts triggered by this analytics rule’ enabled
7. Enable ‘Group related alerts, …’ and then click Next.
8. We will not set up any automation rules at this time.

9. Review the query schedule and save it.

Once completed, attempt to login to your Windows VM more than 10 times and then you will want to check to see if anything has been generated in Microsoft Sentinel. If you navigate to Incidents within Microsoft Sentinel, after your 10 failed login attempts you should observe that there has been an alert generated.





If you navigate to your Log Analytics Workspace and query the following, you will get more information about the specific alerts

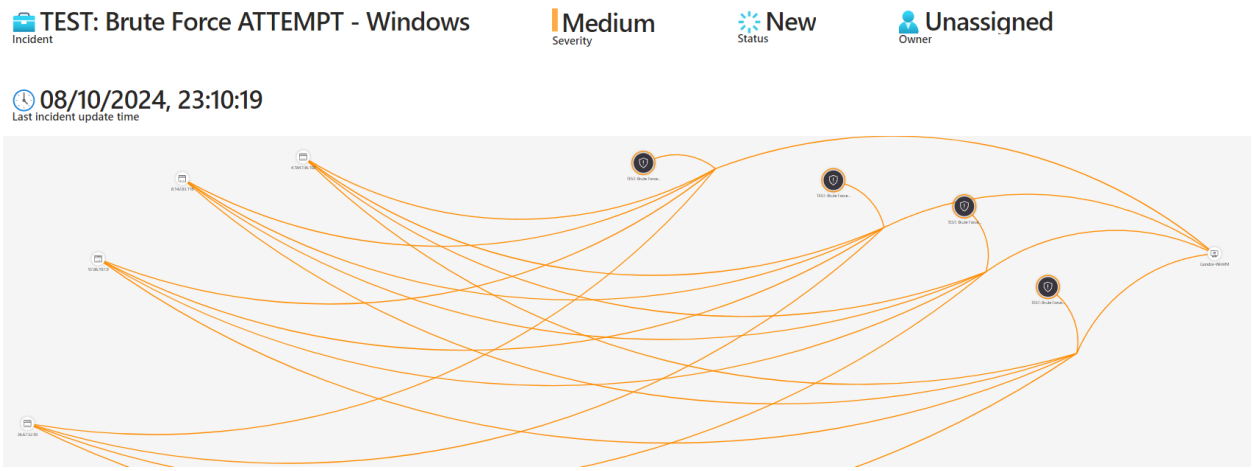
```
SecurityAlert
| summarize arg_max(TimeGenerated, *) by SystemAlertId
| where SystemAlertId in("2c98cb29-2a44-91c1-9fa9-f7d355d6c269", "16c041b7-224a-d51b-d807-fd63235c12b3")
```

```
SecurityAlert
| summarize arg_max(TimeGenerated, *) by SystemAlertId
| where SystemAlertId in("2c98cb29-2a44-91c1-9fa9-f7d355d6c269", "16c041b7-224a-d51b-d807-fd63235c12b3")
```

Results | Chart | Add bookmark

SystemAlertId	TimeGenerated [U...	DisplayName	AlertName	AlertSeverity	Description
16c041b7-224a-d51b-d807-fd63235c12b3	10/9/2024, 3:55:18.746 AM	TEST: Brute Force ATTEMPT - Windows	TEST: Brute Force ATTEMPT - Windows	Medium	When the sa
2c98cb29-2a44-91c1-9fa9-f7d355d6c269	10/9/2024, 3:50:22.158 AM	TEST: Brute Force ATTEMPT - Windows	TEST: Brute Force ATTEMPT - Windows	Medium	When the sa

Navigate back to Microsoft Sentinel | Incidents and select the generated alert. In the right menu pane click 'Action' and select Investigate



This gives you some insight into what is happening with your machines. On the left are the IP Attacker addresses, which are all targeting our Windows VM. The little black circles indicate that there is an alert generated on each of these attack vectors due to the multiplicity of brute force attempts on our Windows VM.

NOTE: You may delete this manual alert now as it will get in the way of the automatic alerts we will implement in the next section.

Creating Analytics Query Rules

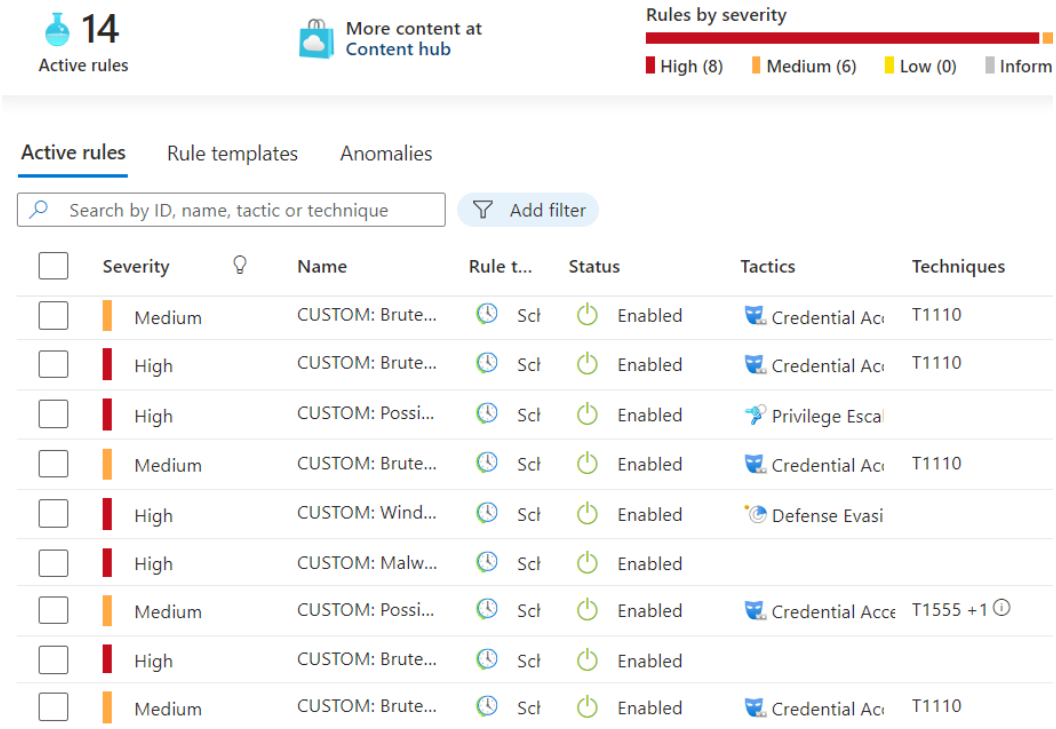
We will add the folllwing automatic custom Analytics Query Rules:

- Brute Force ATTEMPT – Linux Syslog
- Brute Force SUCCESS – Linux Syslog
- Brute Force ATTEMPT – Windows
- Brute Force SUCCESS – Windows
- Brute Force ATTEMPT – Microsoft Entra ID
- Brute Force SUCCESS – Microsoft Entra ID
- Brute Force ATTEMPT – MS SQL Server
- Brute Force ATTEMPT – Azure Key Vault
- Possible Privilege Escalation (Azure Key Vault Critical Credential View or Update)
- Possible Privilege Escalation (Global Admin Role Assignment)

- Possible Lateral Movement (Excessive Password Resets)
- Malware Detected

These alerts may or may not be generated in Microsoft Sentinel when we allow our Honeynet to roam naked in the public internet. But they will help us to direct our incident management efforts using NIST 800-61. To implement these custom Query Rules do the following:

1. Navigate the the GitHub repository located at https://github.com/joshmadakor1/Cyber-Course-V2/tree/main/Sentinel-Analytics-Rules and download the file to your local machine.
2. Navigate to Microsoft Sentinel | Analytics and click 'Import'
3. Find the JSON file you just downloaded (the file will automatically import all of the custom rules we outlined above).
4. Go ahead and investigate the custom rules and make sure you understand each of them, at least in a basic sense. Use AI to help you navigate through the query so you understand what it does.



Triggering Sentinel Alerts

Activate all of your virtual machines for this particular exercise

Trigger AAD Brute Force Success

(from within attack-vm) Simulate brute force success against Azure AD with your attacker account:

- In an incognito windows, open portal.azure.com and fail 10-11 logins in a row, followed by a successful login

- Bonus points if you want to try with PowerShell: AAD-Brute-Force-Success-Simulator.ps1

Trigger MSSQL Brute Force Attempt

(from within attack-vm) Open SSMS and simulate brute force attempt against your SQL Server by attempting to log into it 10-11 times

- Possible to do with PowerShell, not required SQL-Brute-Force-Simulator.ps1

Trigger Malware Outbreak

(from within windows-vm) Generate a Malware alert by using an EICAR file Malware-Generator-EICAR.ps1 **(Do from within Windows VM)**
(this can be done manually by creating a text file with the EICAR string in it)

Trigger Possible Privilege Escalation (AKV Critical Credential Retrieval or Update)

Manually Read Key Vault Secret "Tenant-Global-Admin-Password" in the portal and observe the incidents

Trigger Windows Host Firewall Tampering

Manually Enable and Disable the windows-vm Firewall and observe the incidents

Trigger Excessive Password Resets

Manually Trigger excessive password resets (KQL-Cheat-Sheet.md) and observe the incidents by resetting a users' password in the portal 10-11 times

Optional: attempt to trigger the rest of the custom rules to make sure they work

Exercise 17: Run Insecure Environment for 24 Hours and Capture Analytics

We will now activate the virtual machines and allow them to run for 24 hours without any activity from us. Before you let your environment run for 24 hours go into Logs and run the following queries to be sure we are collecting data for each of the tables:

- SecurityEvent
- Syslog
- SecurityAlert
- AzureNetworkAnalytics_CL

If you do not have data coming in from each of these queries you should go back and troubleshoot before allowing your environment to sit idle for the next 24 hours. Once we have sufficiently analyzed the analytics from this run we will attempt to manage incidents using Microsoft Sentinel in alignment with NIST 800-61 which is a compliance framework for incident management.

From there we will harden our environment using regulatory compliance standards and allow our hardened environment to run once again. We will compare the results from each of these 24 hour periods to determine how our security protocols and practices have enhanced our security posture.

>>> Activate both the Linux and the Windows VM and take a break for 24 hours! You deserve it, and congratulations for getting this far 😊

Capturing Analytics after Running Environment for 24 hours

At this time, it is safe to say your VMs have been running for long enough to capture enough data for analysis. You should probably shut down your virtual machines in order to avoid hefty expenses. We now need to capture some baseline data, so download the excel sheet located at the url below:

https://docs.google.com/spreadsheets/d/1NNejCYqNEM5x_uP9cqwFOOxE8-RC8YlngatREqv9jNw/edit#gid=0

Navigate to Log Analytics Workspace and record data for Security Events, Syslog, Security Alert, Security Incident, NSG Inbound Malicious Flows Allowed, NSG Inbound Malicious Flows Blocked:

Category	KQL Query in Log Analytics Workspace
Start Time Stop Time	range x from 1 to 1 step 1 | project StartTime = ago(24h), StopTime = now()
Security Event	SecurityEvent | where TimeGenerated >= ago(24h) | count
Syslog	Syslog | where TimeGenerated >= ago(24h) | count
Security Alert	SecurityAlert | where DisplayName !startswith "CUSTOM" and DisplayName !startswith "TEST" | where TimeGenerated >= ago(24h) | count

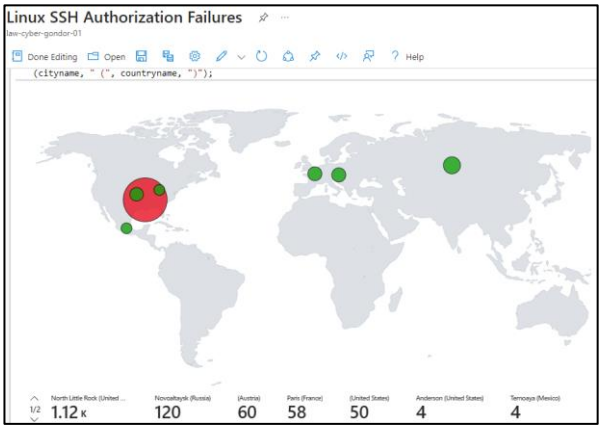
Security Incident	SecurityIncident | where TimeGenerated >= ago(24h) | count
NSG Inbound Malicious Flows Allowed	AzureNetworkAnalytics_CL | where FlowType_s == "MaliciousFlow" and AllowedInFlows_d > 0 | where TimeGenerated >= ago(24h) | count

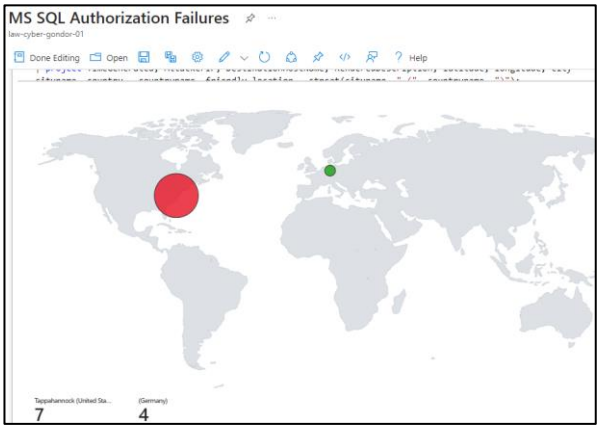
For each of the queries above copy the associated data. Use the provided Excel sheet to track the data, you should have something like this:

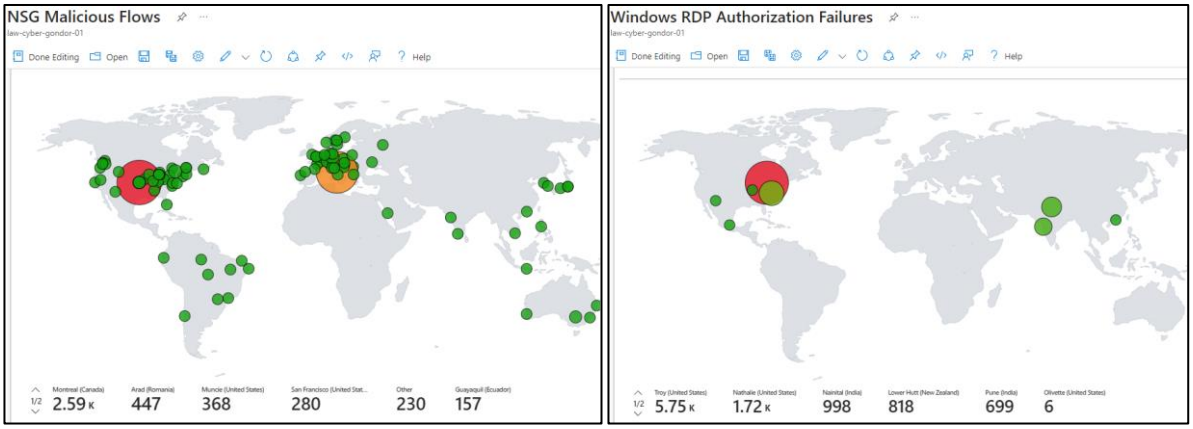
BEFORE SECURING ENVIRONMENT	
Start Time	2024-10-09T02:48:46.7896917Z
Stop Time	2024-10-10T02:48:46.7896917Z
Security Events (Windows VMs)	59326
Syslog (Linux VMs)	4154
SecurityAlert (Microsoft Defender for Cloud)	1
SecurityIncident (Sentinel Incidents)	118
NSG Inbound Malicious Flows Allowed	5486
AFTER SECURING ENVIRONMENT	
Start Time	
Stop Time	
Security Events (Windows VMs)	
Syslog (Linux VMs)	
SecurityAlert (Microsoft Defender for Cloud)	
SecurityIncident (Sentinel Incidents)	
NSG Inbound Malicious Flows Allowed	

RESULTS (will auto update, do not edit formulas)	
	Change after security environment
Security Events (Windows VMs)	-100.00%
Syslog (Linux VMs)	-100.00%
SecurityAlert (Microsoft Defender for Cloud)	-100.00%
Security Incident (Sentinel Incidents)	-100.00%
NSG Inbound Malicious Flows Allowed	-100.00%

Shown below is a comparison of our world maps after the 24-hour period.

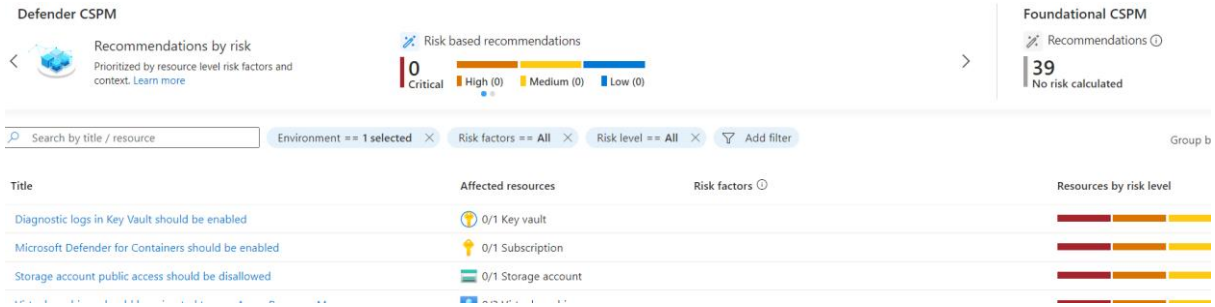






To learn about managing incidents please look at Appendix B at the end of this manual. We leave that to the individual lab participants. For the purposes of this lab, we wish to move onto securing the environment with regulatory compliance standard NIST 800-53. A big chunk of the time spent on this portion of the lab will be spent waiting for the assignment of NIST 800-53 regulatory compliance for security measures.

If you navigate to Microsoft Defender for Cloud | Overview and then > Security Posture > Recommendations, you will observe the recommendations to enhance security posture. Of the many things it recommends, among those are enabling Multi-Factor Authentication, Managing Ports, Applying System Updates, Encrypting Data in Transit, Managing AAA (Authorization, Access, and Accounting), and many more. Below is a screenshot of this dashboard.



One of the first things we should do to harden our system is to remove the insecure inbound rules from our network security groups. To do this, navigate to Network Security Groups and select one of the two VMs we have been working with. From there, navigate to 'Inbound security rules' and delete the 'DANGER…' rules we applied at the very beginning of the lab; do the same on the other Network Security Group.

Enabling NIST 800-53 For Regulatory Compliance (updated)

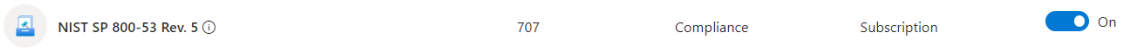
The process to enable this standard has completed and is much simpler but doesn't resemble the instructor video covering this topic. Follow these instructions instead:

1. Navigate to Microsoft Defender for Cloud.
2. From the left menu pane select 'Regulatory compliance'
3. Click the 'Manage compliance standards' tab at the top
4. Click on your 'subscription' to navigate to 'Settings | Defender plans'

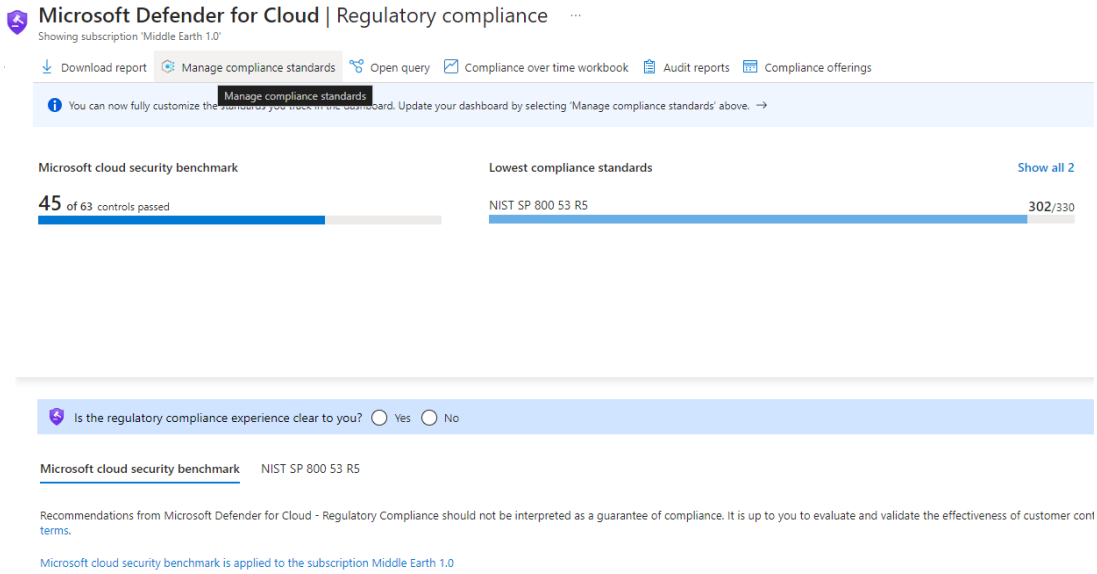
5. Select ‘Security policies’ from the left menu pane.

6. Simply click the toggle button to turn on ‘NIST SP 800-53 Rev. 5

   NOTE: It will say ‘Assigned Successfully’ in your notifications, but that doesn’t mean that the standard has been implemented. It takes upwards of 12 hours or more to implement in your environment. Maybe come back later to check it.



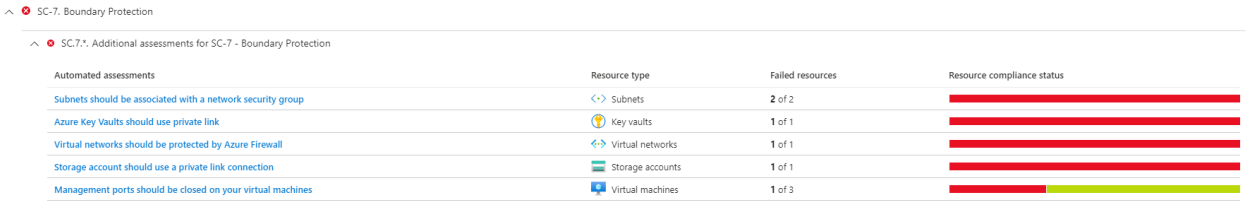
   Below is a screenshot of what you should observe once the NIST 800-53 policy standard has been correctly associated with your environment.



As a side, recall what NIST 800-53 does for us. Essentially it is a security policy standard that applied a family of controls to our environment. The different families include things like Access Control, Awareness and Training, Configuration Management, Incident Response, and so much more. For more information about NIST 800-53, please review the following https://csrc.nist.gov/pubs/sp/800/53/r5/upd1/final and be sure you understand this as well as the NIST 800-61 Incident Management playbook. These and many other standards are vitally important tools in the arsenal of cybersecurity professionals.

Exercise 18: Enabling Boundary Protection for Regulatory Compliance and Running Secure Environment

We will enable NIST 800-53 SC.7 (Boundary Protection). Shown below is a demonstration of the fact that our environment is out of compliance with this particular standard, so we will spend the bulk of this section bringing this into compliance.



It will be left to the participant to remediate any other categories that may be out of compliance with NIST 800-53. It is not mandatory that you do this, but for good practice and plenty of exposure to these compliance standards this would be a worthwhile activity to add to your GitHub portfolio or even add to your resume.

Configure Azure Private Link and Firewall for your Azure Key Vault Instance

Navigate to Key Vault in Microsoft Azure and select your key vault instance.

1. In the left menu pane, select 'Settings > Networking'
2. Choose 'Disable public access' and under Exception, choose 'Allow trusted Microsoft services to bypass this firewall'
3. Click 'Apply' to deploy the changes to the key vault instance.
4. Select the 'Private endpoint connection' tab at the top of this same dashboard.
5. Click '+ Create' and choose the following options:
    a. Basics: Choose appropriate subscription and resource group, give it an appropriate name and select the same region we have been working in all along.
    b. Resource: Connection method should be 'Connect to an Azure resource in my directory' and choose 'Microsoft.KeyVault/vaults' and then select your appropriate key vault for resource.
    c. Networking: Choose the Vnet we've been using all along; for subnet choose 'default' and then make sure 'Private IP configuration' is set to 'Dynamically allocate IP address'
    d. DNS: Select 'Yes' for Integrate with private DNS zone.
    e. Navigate to 'Review + Create' and upon validation click 'Create'

Configure Azure Private Link and Firewall for your Storage Account

Navigate to Storage Accounts in Microsoft Azure and select your Storage Account.

1. Select 'Security + Networking > Networking' from the left menu pane.
2. For Firewalls and virtual networks, choose 'Disabled' for Public network access and leave Network Routing alone. Save the settings.
3. Click 'Private endpoint connections' and add a new 'Private endpoint'
4. As always select the appropriate subscription, resource group, name it something simple and ensure it is in the same region as always.

5. For resource just ensure that the 'Target sub-resource' is 'blob'
6. For virtual network choose the appropriate Vnet and ensure that Private IP configuration is set to 'Dynamically allocate IP address'
7. Choose 'Yes' to integrate with private DNS zone
8. Navigate to 'Review + Create' and click 'Create'

Reconfigure Inbound Rules for Virtual Machine NSGs

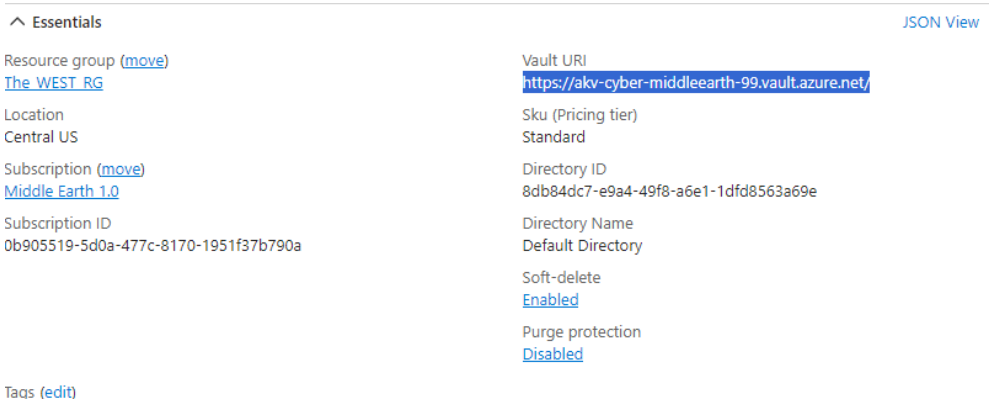
During this entire lab we have allowed anyone with any IPv4 or IPv6 to attempt to connect to our virtual machines on any port. Now we want to limit that exposure as much as possible, but we want to maintain our ability to connect to these virtual machines. So, we will remove the dangerous inbound rules for each instance and add a new inbound rule to allow only our public IP address to access the machines. *If you haven't already done this during the Incident Management segments please do this now.*

Navigate to Network Security Group and create the following inbound rule for each virtual machine.

1. Source: Choose My IP and enter your IPv4 address in CIDR notation.
2. Source port ranges: *
3. Destination: Any
4. Service: Custom
5. Destination port ranges: *
6. Protocol: Any
7. Action: Allow
8. Priority: 100
9. Name: AllowMyIPonly
10. Click Add

Repeat this for each virtual machine and be sure to delete the previous 'DANGER-AllowAnyConnection' rule that we had previously enabled. This will ideally prevent malicious flows and only allow you to connect to your own virtual machine instances.

To verify that the storage account and key vault instances have been configured properly with their Private Links and Firewall settings turn on your Windows VM and login using the Microsoft Terminal Service Client and open a Powershell Terminal. Grab the FQDN of the Azure Key Vault instance within Azure



To retrieve it, navigate to Azure Key Vaults and select your appropriate Key Vault instance. In over view you will see the Vault URI.

Back in your Windows-VM instance, within Powershell, type

nslookup <FQDN>

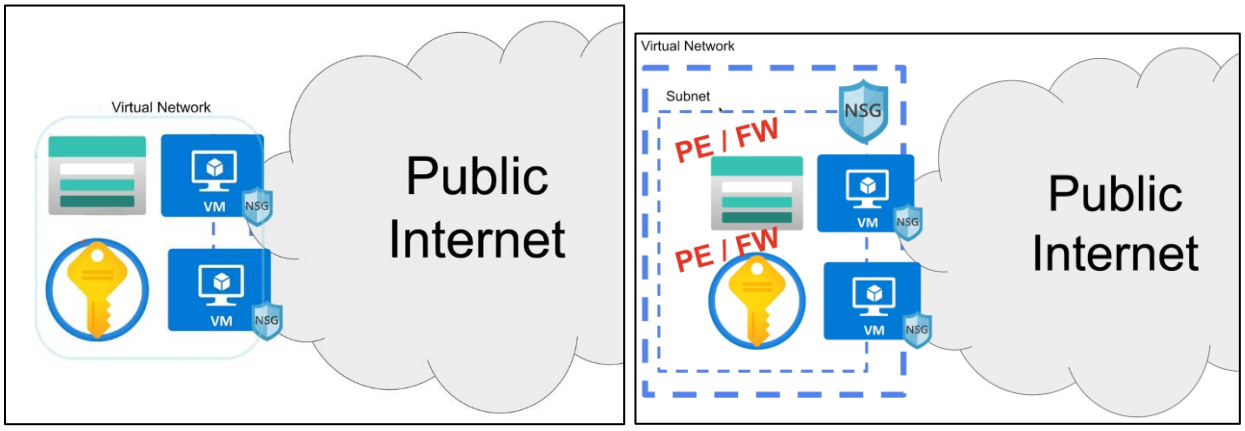
```
PS C:\Users\aragorn1> nslookup akv-cyber-middleearth-99.vault.azure.net
Server:  UnKnown
Address:  168.63.129.16

Non-authoritative answer:
Name:    akv-cyber-middleearth-99.privatelink.vaultcore.azure.net
Address:  10.0.0.6
Aliases:  akv-cyber-middleearth-99.vault.azure.net
```

Now navigate to Storage Accounts and select your appropriate storage account. In the left menu pane, navigate to 'Settings > Endpoints' and the grab the FQDN for the Blob Service primary endpoint

nslookup <FQDN>

```
PS C:\Users\aragorn1> nslookup samiddleearth.blob.core.windows.net
Server:  UnKnown
Address:  168.63.129.16

Non-authoritative answer:
Name:    samiddleearth.privatelink.blob.core.windows.net
Address:  10.0.0.7
Aliases:  samiddleearth.blob.core.windows.net
```

The remaining task is to place a network security group on the subnet we created for these resources. The end goal is to transition from an unprotected, open network like the one shown in the figure to the left, to a secured virtual network like the one shown in the figure on the right.



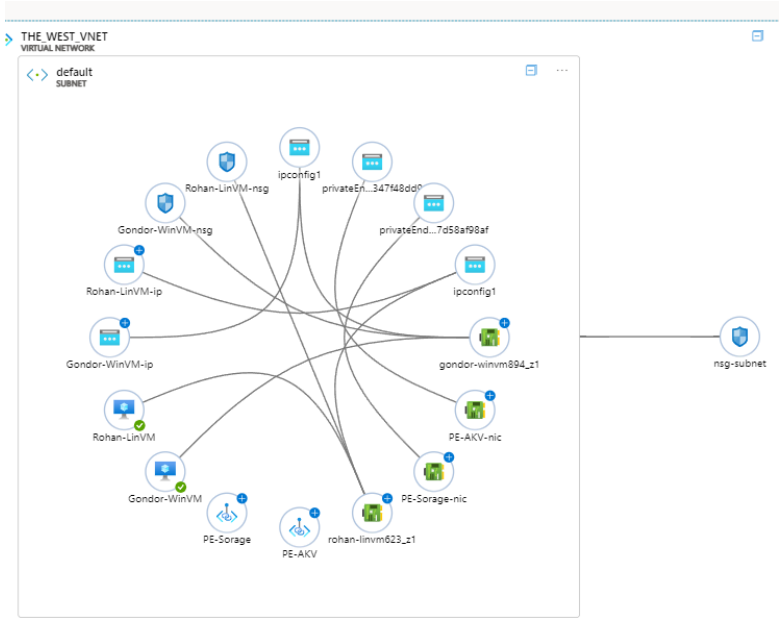
Navigate to Network Security Groups and create a new network security group in the same subscription and resource group as usual with the name 'nsg-subnet' and make sure it is located in the standard region where all other resources have been placed.

Navigate to Virtual Networks and click on the appropriate Virtual Network that we have been using for this lab. From there, be sure to select 'Settings > Subnets' from the left menu pane. Choose the default subnet, and under security use the drop down menu for 'Network Security Group' to select

nsg-subnet that you just created. Click save at the bottom and you have successfully assigned a network security group to your subnet which holds the Azure Key Vault and Storage Accounts. You are not in compliance with many of the components for NIST 800-53 S.C. 7.

Observe The Network Topology in Network Watcher

Finally, let us have a look at your network topology. To do so, navigate to Network Watcher in Microsoft Azure, and then click 'Monitoring > Topology' to get started. From there, change the filter so that you are only looking at the Resource Group associated with this lab, as well as the region where your resources are located. You should see something resembling the figure below



If you navigate back to Regulatory Compliance and check S.C. 7 under NIST 800-53, we can see that our environment is now in the process of recognizing the changes we made to the Boundary Protection.

SC.7.*. Additional assessments for SC-7 - Boundary Protection

Automated assessments	Resource type	Failed resources	Resource complianc...
Subnets should be associated w	Subnets	2 of 2	
Virtual networks should be prot	Virtual networks	1 of 1	
Management ports should be cl	Virtual machines	1 of 3	
Management ports of virtual ma	Virtual machines	1 of 3	
All network ports should be rest	Virtual machines	1 of 3	

>>> At this stage you are ready to run your environment for another 24-hour period and capture analytics. This is the last remaining section for the hands-on lab in this Cybersecurity Masterclass, congratulations on your achievements!

Post 24-hour Live Session

We have concluded the 24 hour period and now it is time to run our analysis on the environment once more. Using the same log queries that were used during Exercise 17 we should get a result like the one shown below in the Excel Sheet we used previously.

BEFORE SECURING ENVIRONMENT	
Start Time	2024-10-09T02:48:46.7896917Z
Stop Time	2024-10-10T02:48:46.7896917Z
Security Events (Windows VMs)	59326
Syslog (Linux VMs)	4154
SecurityAlert (Microsoft Defender for Cloud)	1
SecurityIncident (Sentinel Incidents)	118
NSG Inbound Malicious Flows Allowed	5486

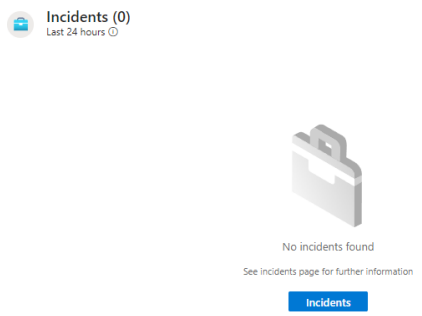
AFTER SECURING ENVIRONMENT	
Start Time	10/13/2024, 5:28:51 PM
Stop Time	10/14/2024, 5:28:51 PM
Security Events (Windows VMs)	17966
Syslog (Linux VMs)	2
SecurityAlert (Microsoft Defender for Cloud)	0
SecurityIncident (Sentinel Incidents)	0
NSG Inbound Malicious Flows Allowed	0

RESULTS (will auto update, do not edit formulas)	
	Change after security environment
Security Events (Windows VMs)	-69.72%
Syslog (Linux VMs)	-99.95%
SecurityAlert (Microsoft Defender for Cloud)	-100.00%
Security Incident (Sentinel Incidents)	-100.00%
NSG Inbound Malicious Flows Allowed	-100.00%

Note the tremendous results after lockdown. We have reduced Security Events by about 70% upon locking the system down from the public facing internet. In fact, if you dive into the logs, the vast majority (if not all) come from the system itself rather than outside agents.

The syslogs have been reduced by a staggering 99.95% from where they were before the security hardening. In fact, most of that comes from locking down the Network Security Group inbound rule changes.

Security Alerts, Security Incidents, and Inbound Malicious flows all reduced to zero.



In all cases, no results are generated for our world maps meaning we had zero brute force attempts since our environment was hidden from the public internet. This is exactly as expected.

Exercise 19: Wrapping up and Cleaning Your Environment

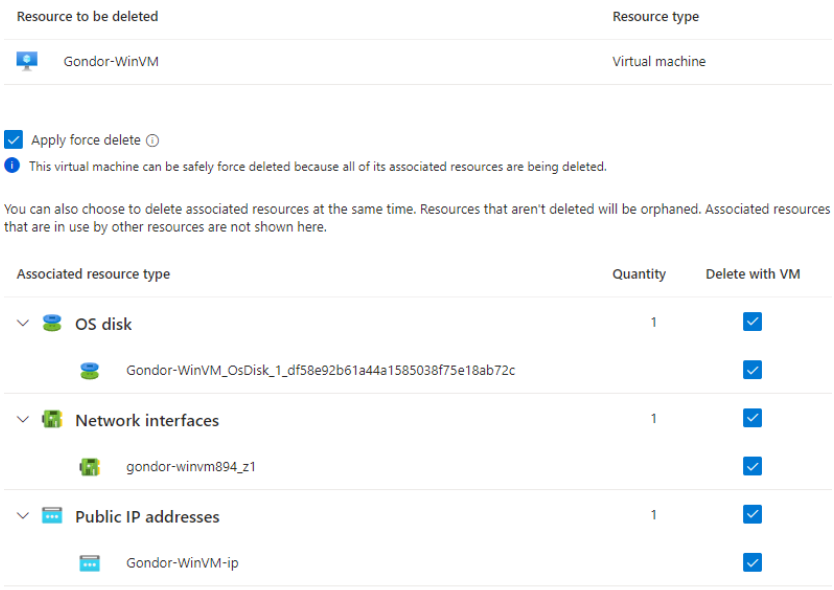
Beyond what we did in the lab, some of the recommended extra activities include brining your system into full regulatory compliance in accordance with NIST SP 800-53 rev 5. We will not go through that as it is a very tedious process to do so. Some other fun activities could include generating a variety of users for each system and assigning roles to those systems. You might simulate an inside threat actor that is exfiltrating large amounts of data and collect the Network Traffic that is generated in that scenario. You can also implement least privilege protocols and multi-factor authentication in your environment.

One other suggestion we have is to earn your AZ-900 Certificate which is the Microsoft Azure Fundamentals certification. Many lab participants have noted success in this endeavor after completing the lab and taking a course on LinkedIn learning or some other platform. Really, it is up to you.

Cleaning Your Environment

Now that the lab is finished and you're done playing around with the resources, you can go ahead and remove your resources and eventually the subscription associated with this lab if you wish. To do so follow these instructions.

1. Navigate to your home dashboard where all your resources are listed.
2. Click 'see all' at the bottom of the list to view all of your resources.
3. In order to delete your resource groups, you must first delete all of the items within the resource group first. Here is the hierarchal approach to this process:
    a. Delete Azure Key Vaults, Storage Accounts, and Private Endpoints
    b. Delete virtual machines; include any associated resource types



    c. Navigate to your resource group and verify that nothing remains, if there is anything else be sure to delete those items (Log Analytics Workspace, VNets, NSGs, etc.)
    d. Delete your resource groups at this point.

e. All that should remain is your subscription at this point and you may choose to remove that or keep it if you are going to mess around with the lab again.

NOTE: I would encourage you to keep your subscription if you plan on going through the lab once again; that will prevent you from having to create a new account/subscription.