Project Goal

The purpose of this project is to:

- Demonstrate a comprehensive understanding of network topology setup and Firewall configurations.
- Exhibit proficiency in implementing the Zero-Trust security concept.
- Configure various Next-Generation Firewall (NGFW) features and modules.

Tools Used

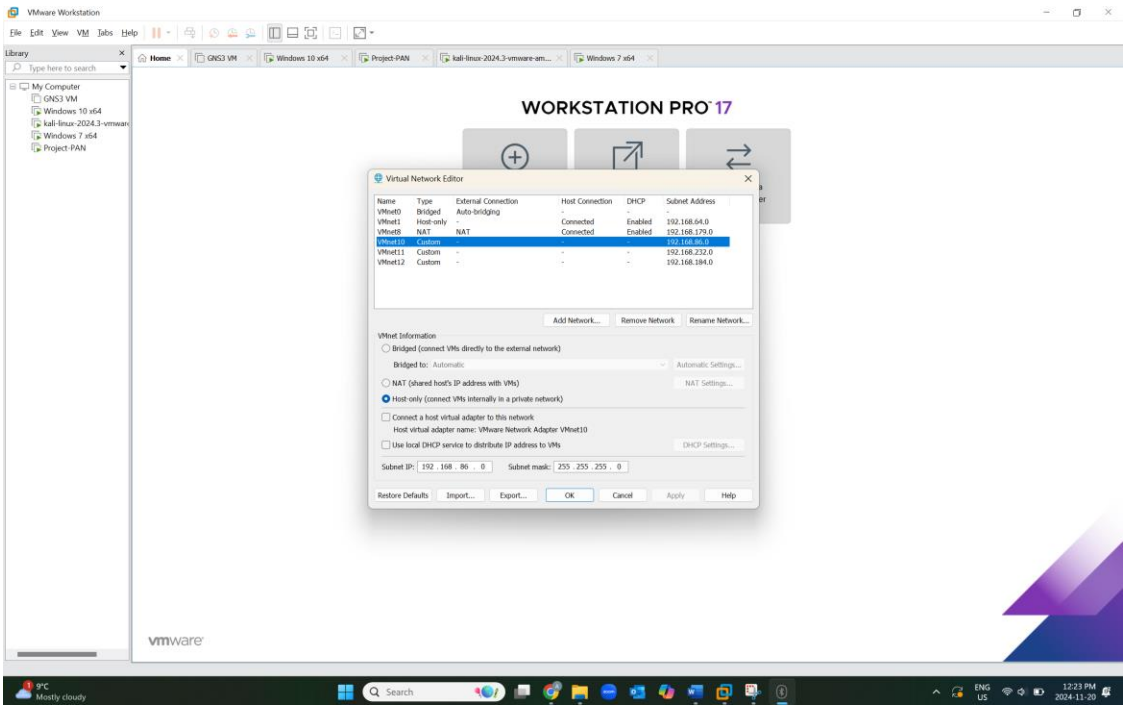
- VMware Workstation Pro (already installed during GNS3 Labs)
- Kali Linux VMware Virtual Machine
- Windows 7 Virtual Machine (DMZ-Host)
- Windows 10 Virtual Machine (Inside-Host)
- Palo Alto NGFW VM

Brief Description

- Implement Zero-Trust security by isolating hosts and blocking internet access.
- Configure and test firewall rules, application awareness, HTTPS inspection, URL filtering, and antivirus inspection.
- Simulate and mitigate network attacks, including exploiting and blocking the MS17-010 vulnerability using Metasploit and IPS modules.

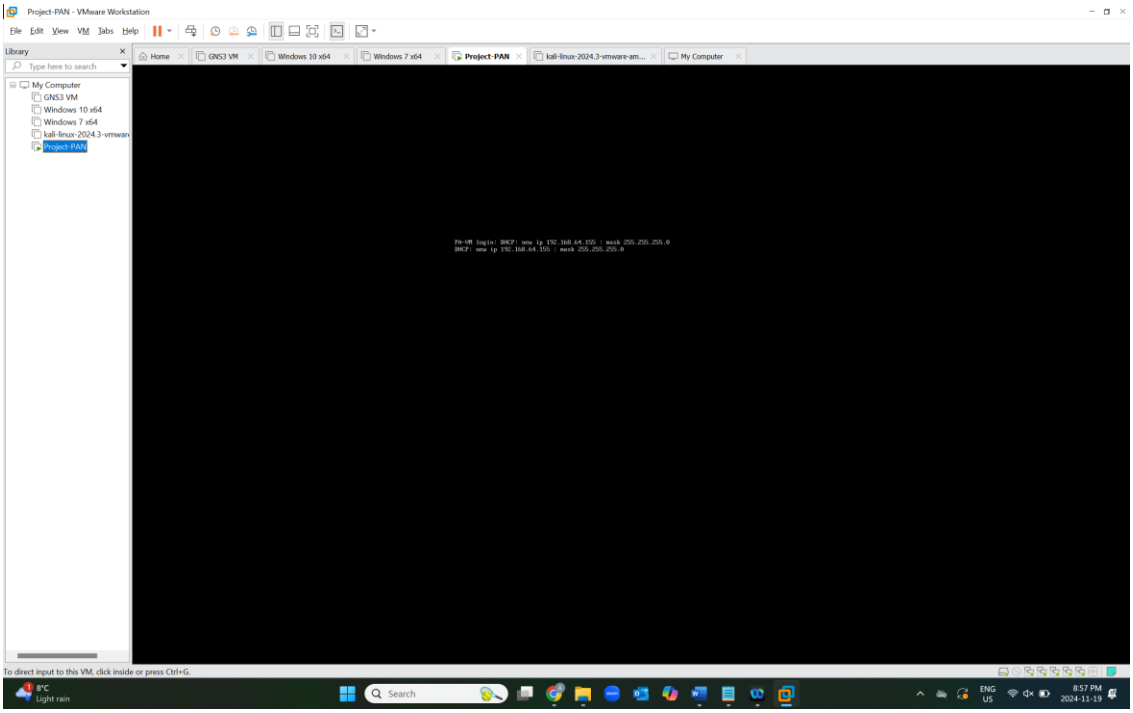
Topology, Steps, and Configuration

1. **Set Up Virtual Networks in VMware:**

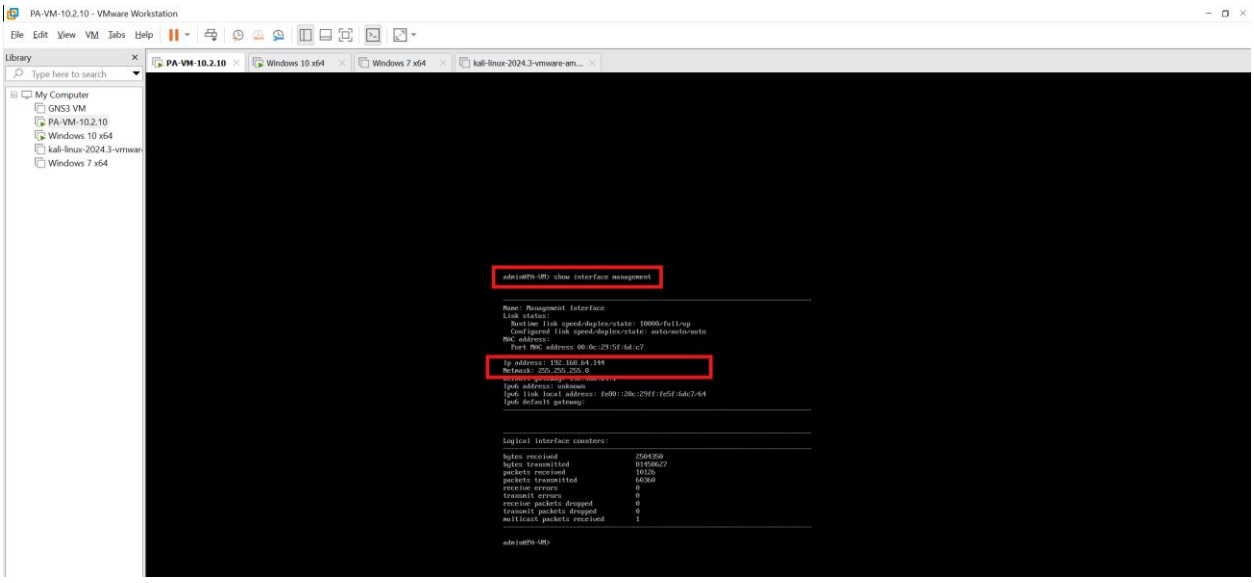


2. Palo Alto NGFW Setup:

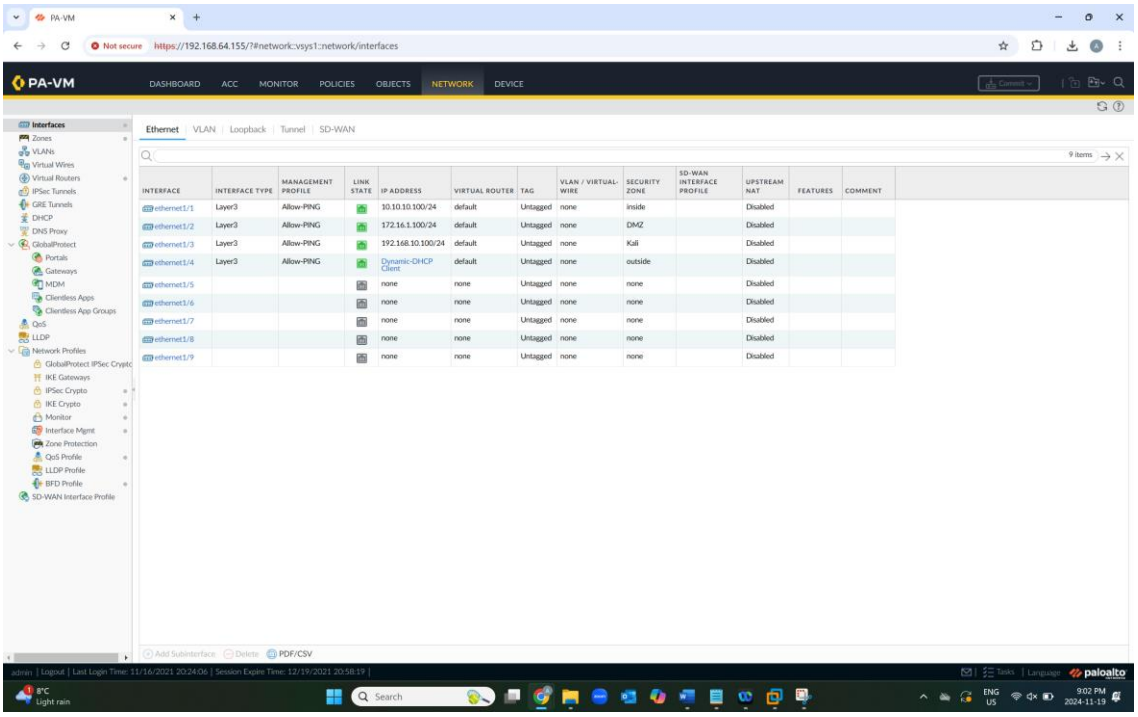
Below screenshot represents the Ip of Palo alto networks which is “192.168.64.155”.



My Ip for PA-VM changed the second time for the new PA-VM which was issued which is 192.168.64.144:

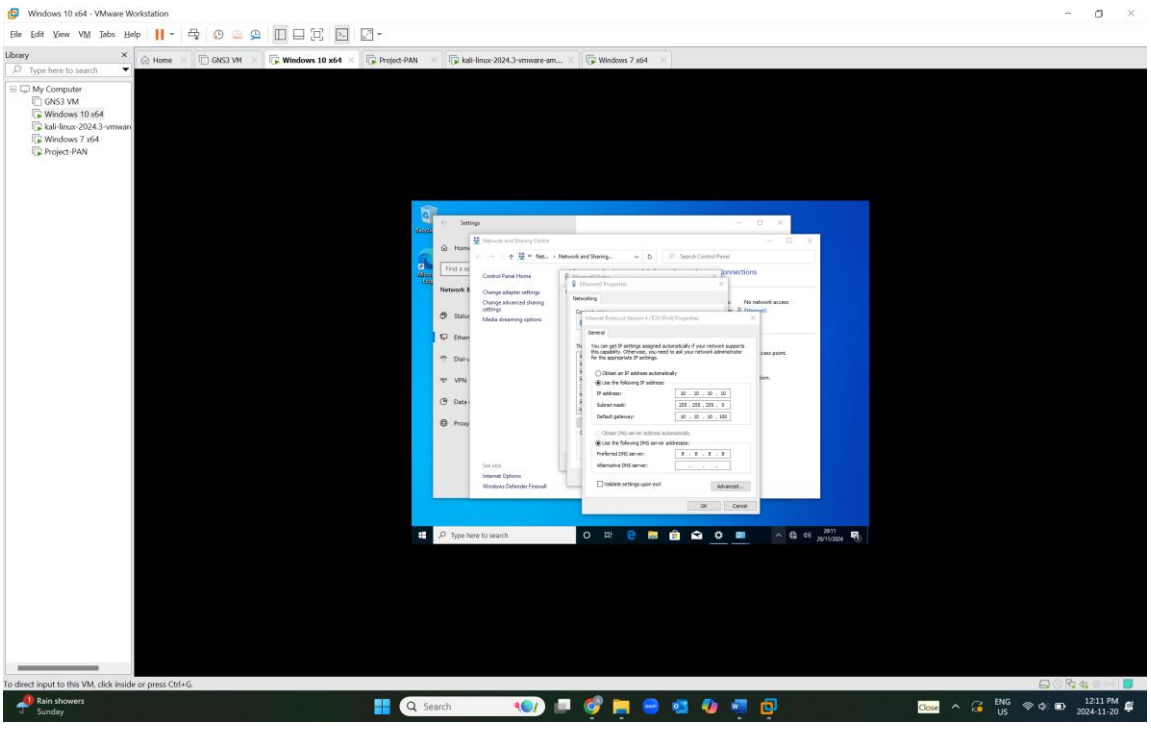


The bellow screenshot shows the correct FW Zones assignments and interfaces' IP addresses:

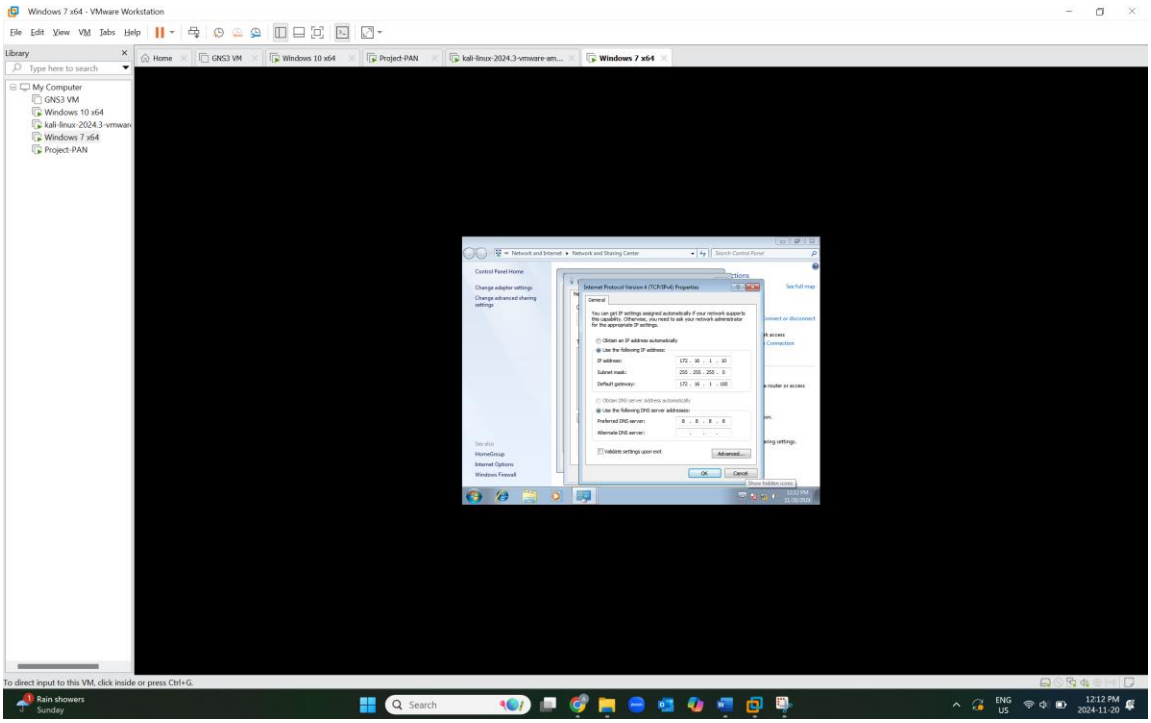


3. Host Configuration:

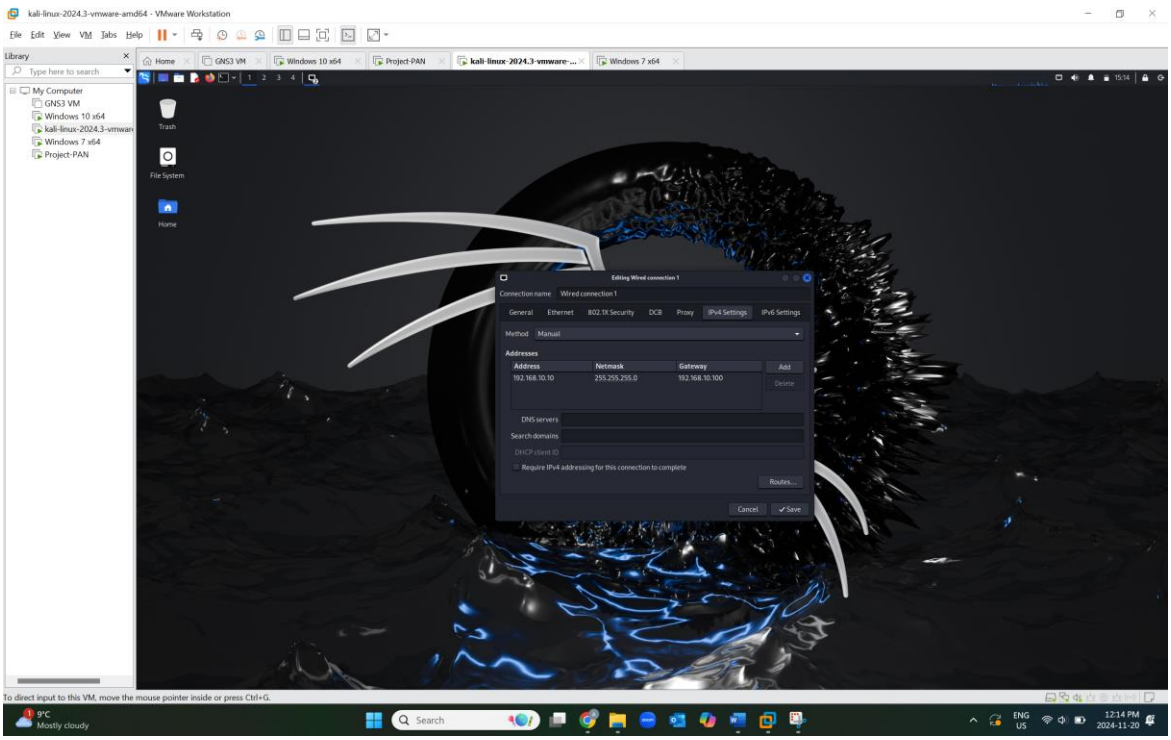
IP settings for windows 10 Inside-Host: 10.10.10.10 / 255.255.255.0 / GW: 10.10.10.100 / DNS: 8.8.8.8 as shown below:



IP settings for windows 10 DMZ-Host: 172.16.1.10 / 255.255.255.0 / GW: 172.16.1.100 / DNS: 8.8.8.8 as shown below:

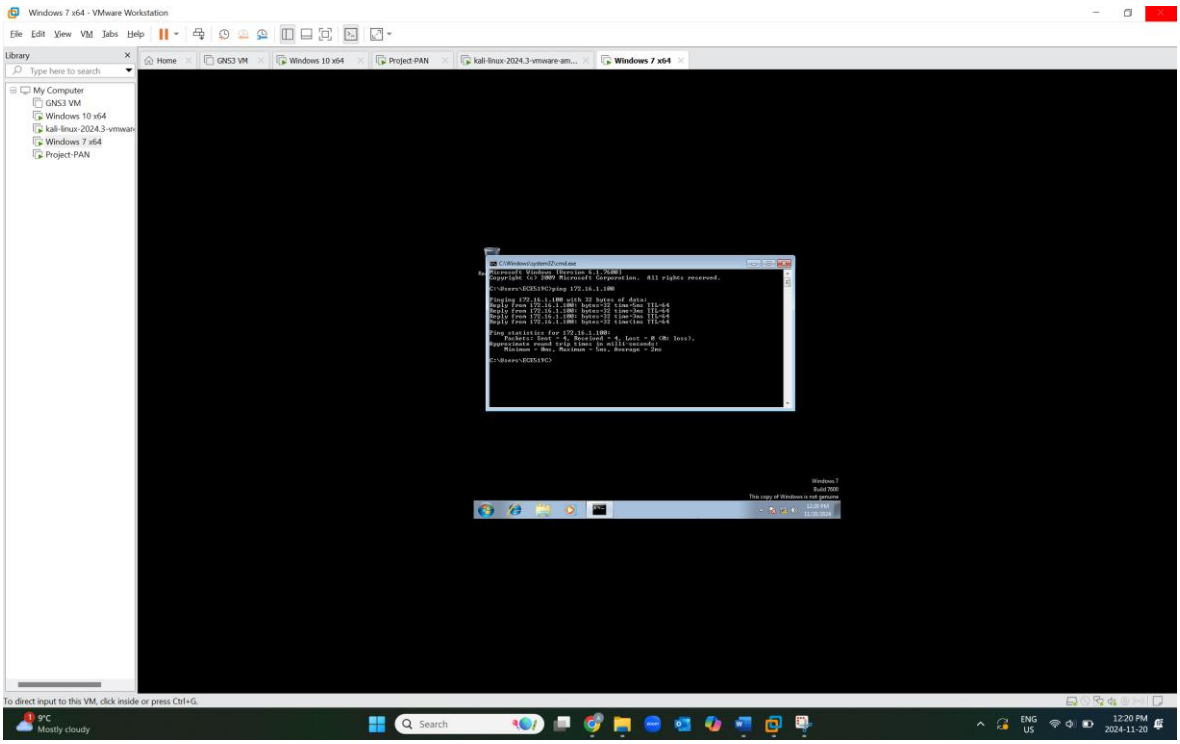


IP settings for Kali Linux: 192.168.10.10 / 255.255.255.0 / GW: 192.168.10.100 / DNS: 8.8.8.8 as shown below:

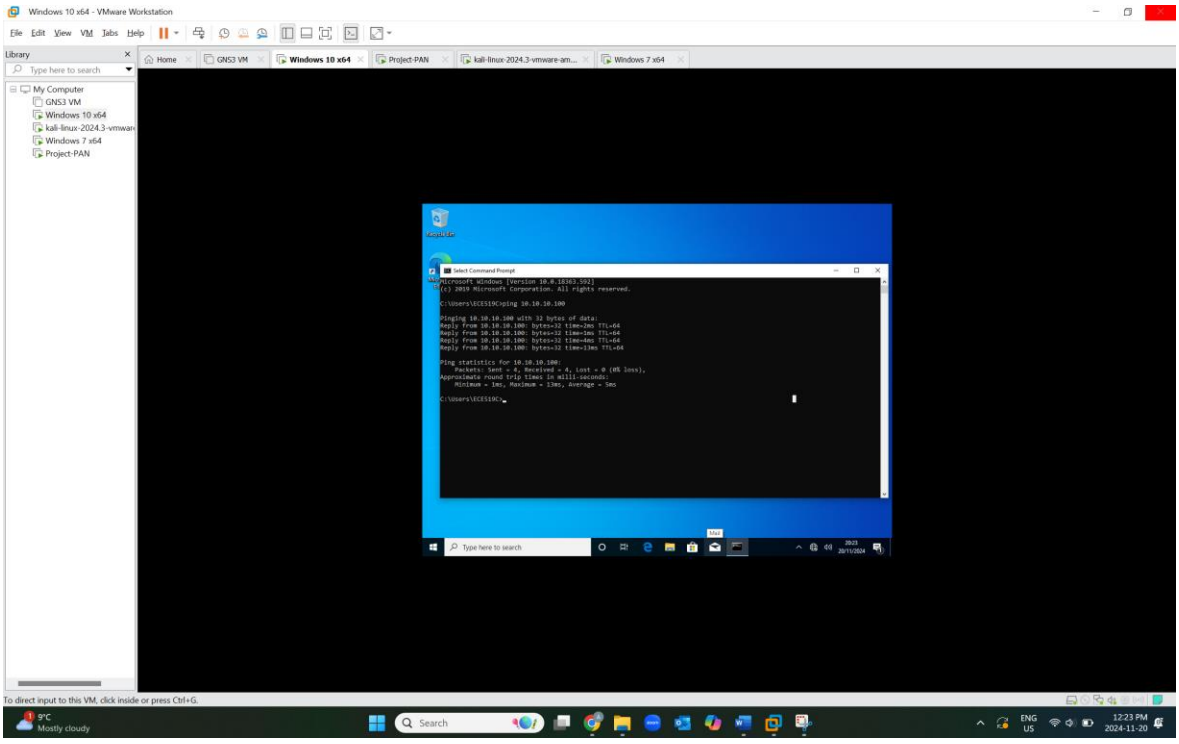


4. Test connectivity using the Ping tool for each host and its corresponding interface.

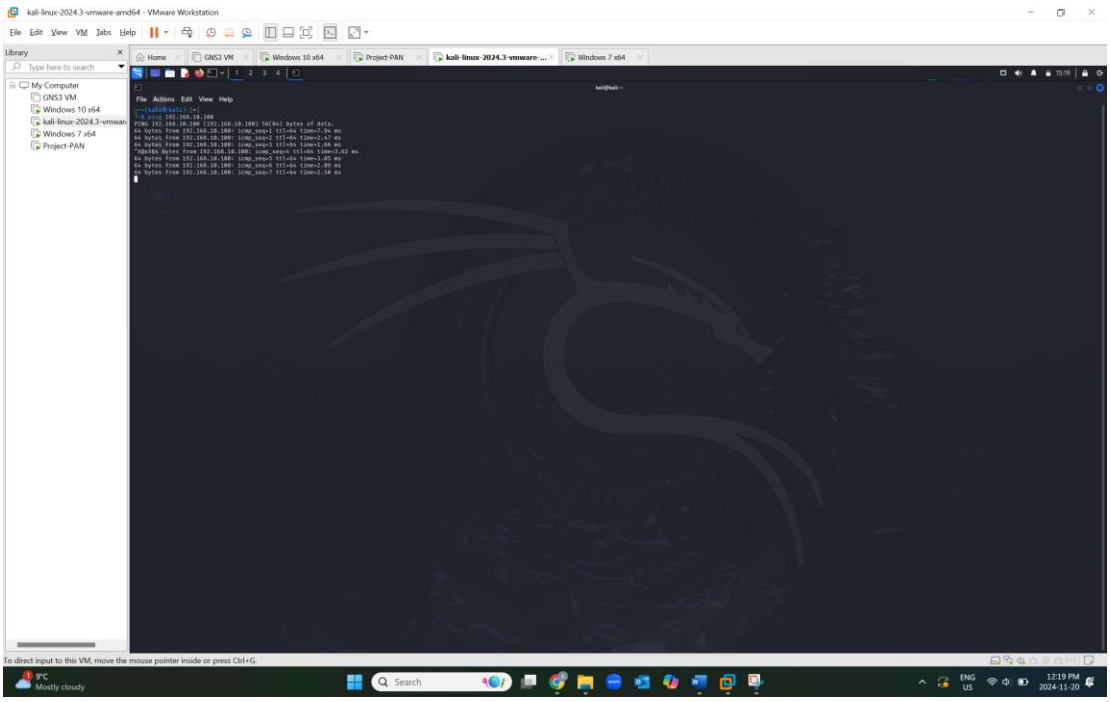
Ping for Windows 7 is working with the corresponding interface:



Ping for Windows 10 is working with the corresponding interface:



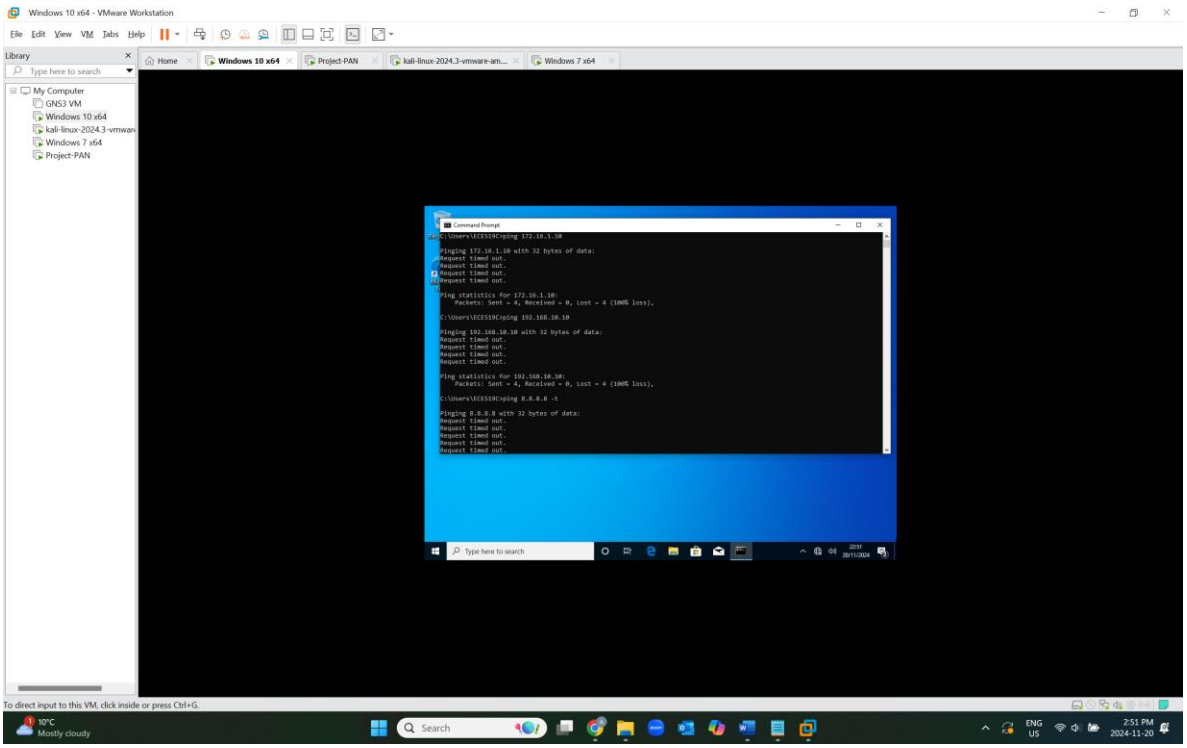
Ping for Kali is working with the corresponding interface:

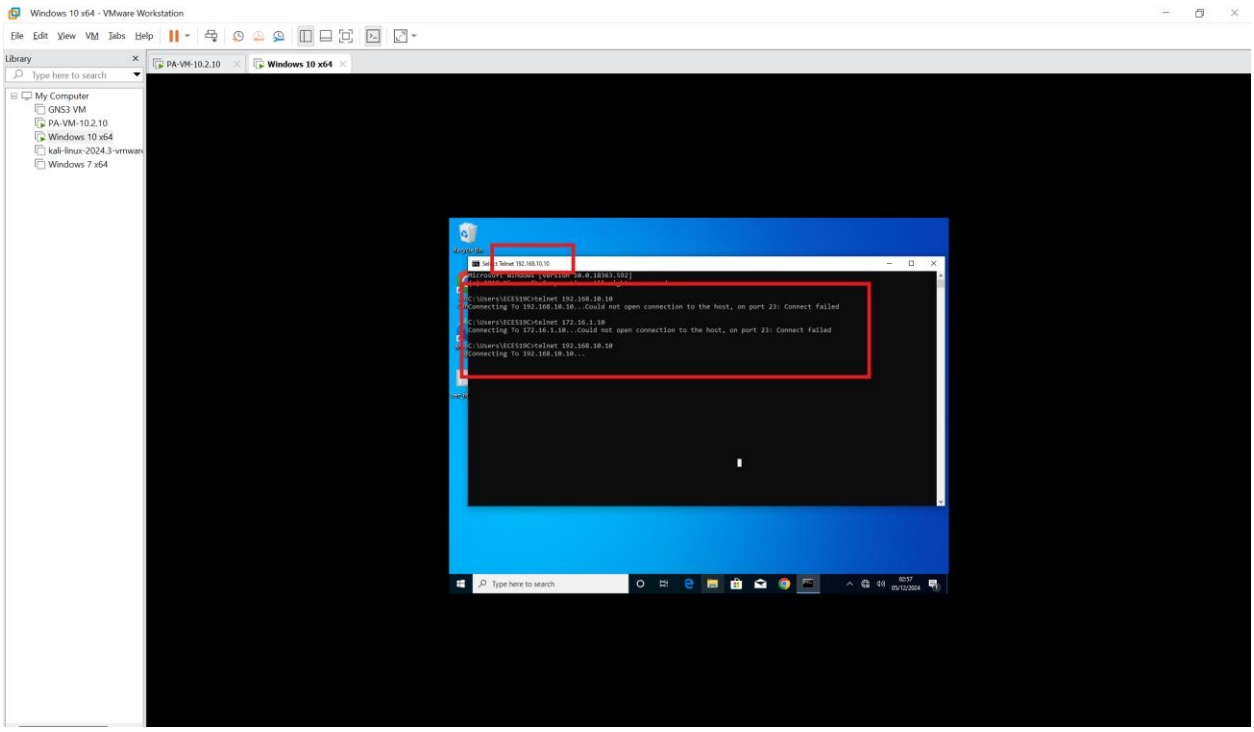


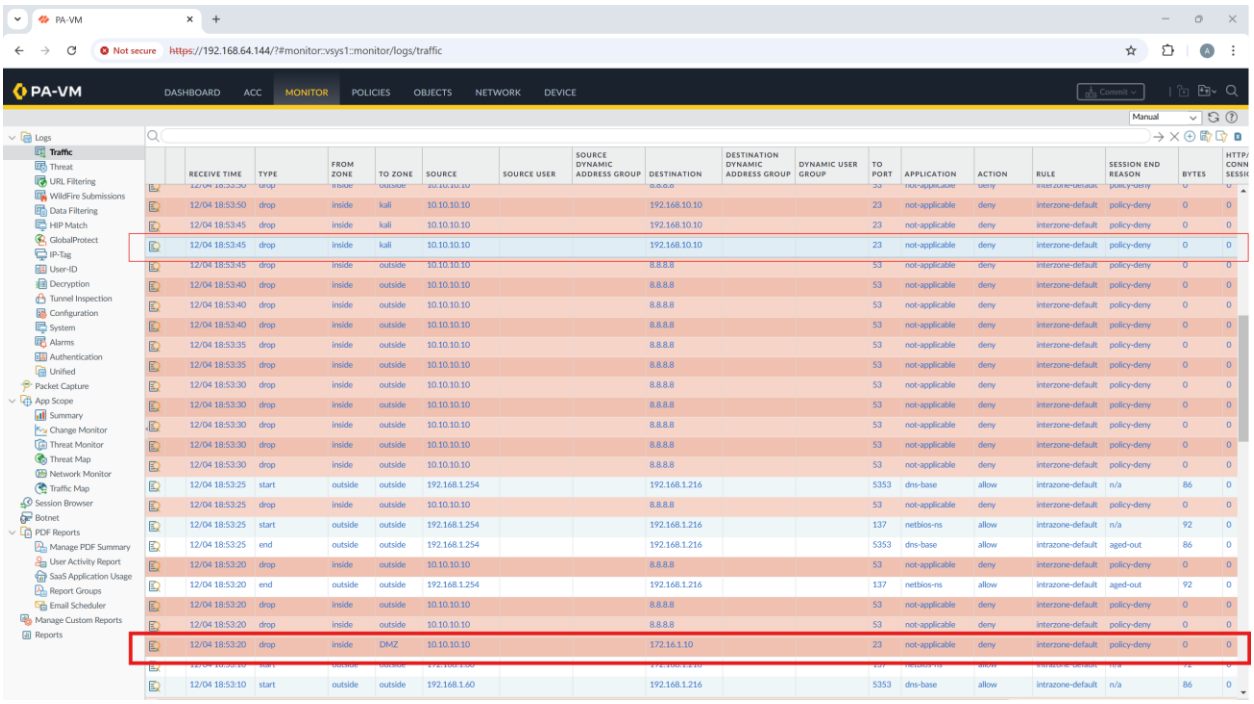
Project Tasks

Q1. Demonstrate that the Inside-Host, Kali-Linux, and DMZ-Host cannot access each other or the Internet (Zero-Trust concept). (Note: Some screenshots are from previous PA_VM Ip: 192.168.64.144 and some from new PA_VM Ip: 192.168.64.155)

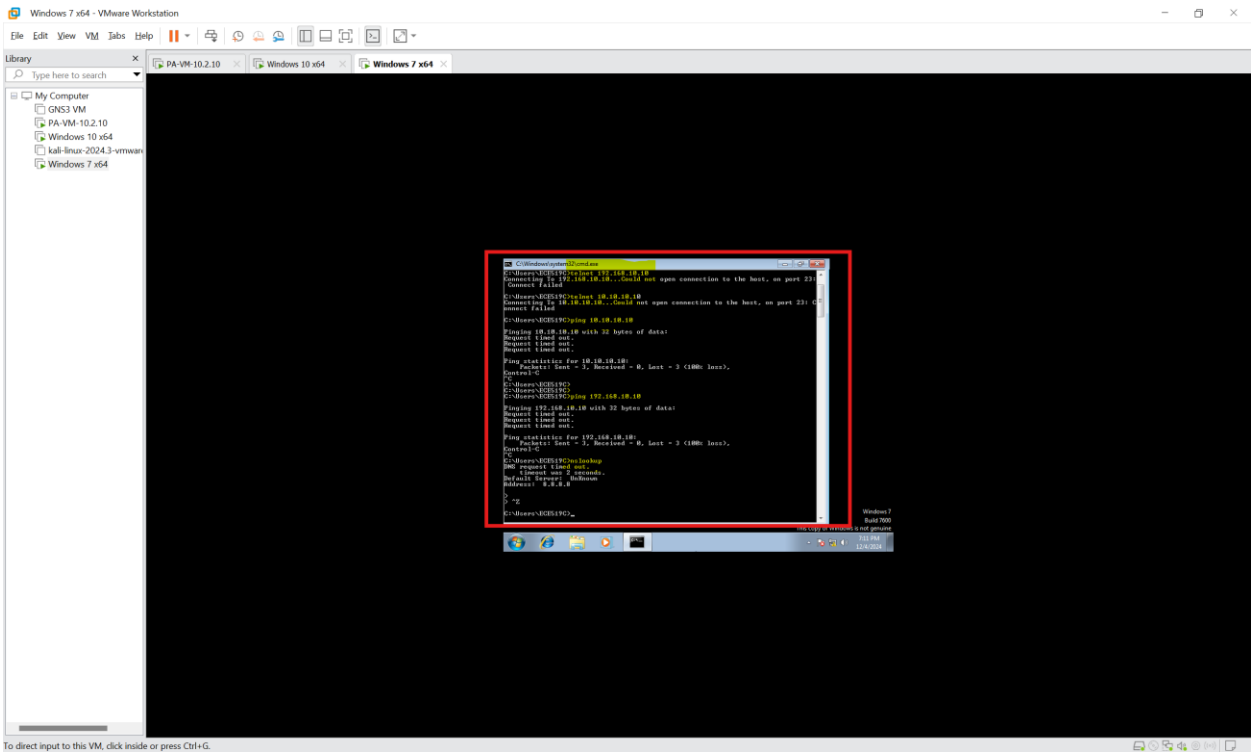
1. The below screenshot shows the deny request of ping from inside-host to other hosts and internet.



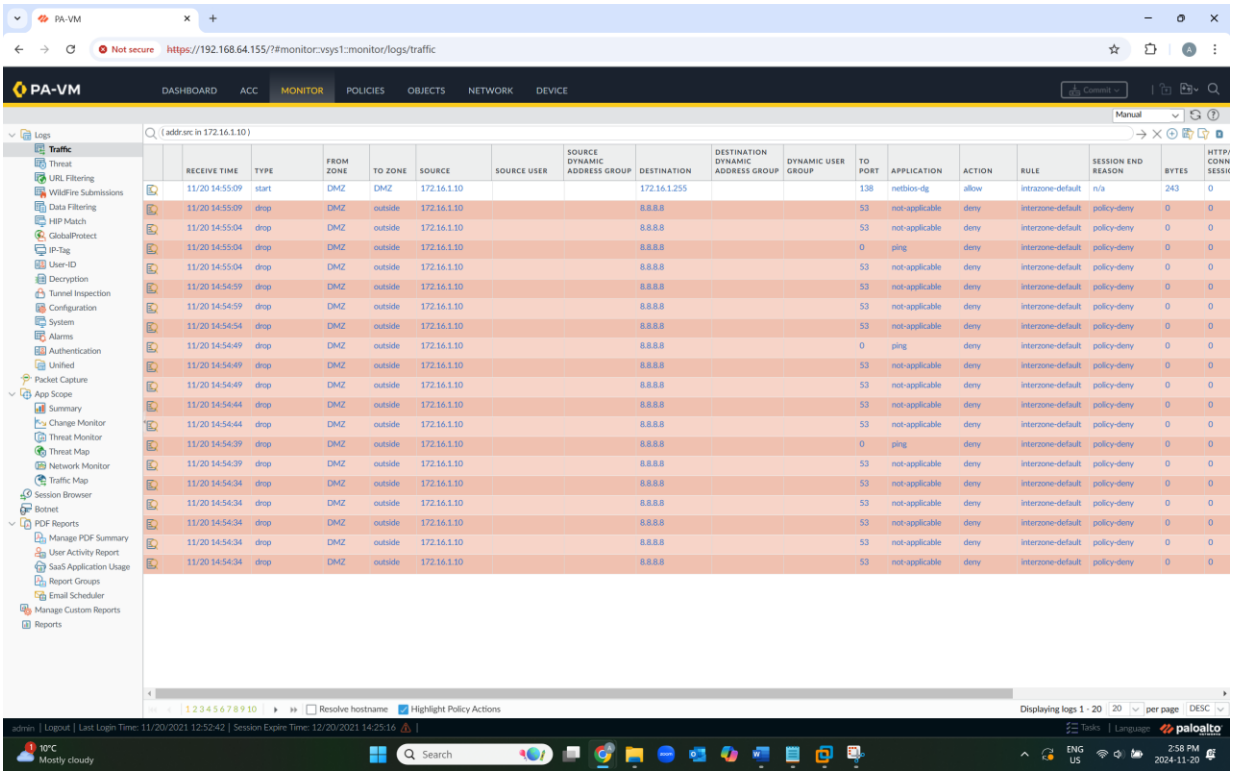


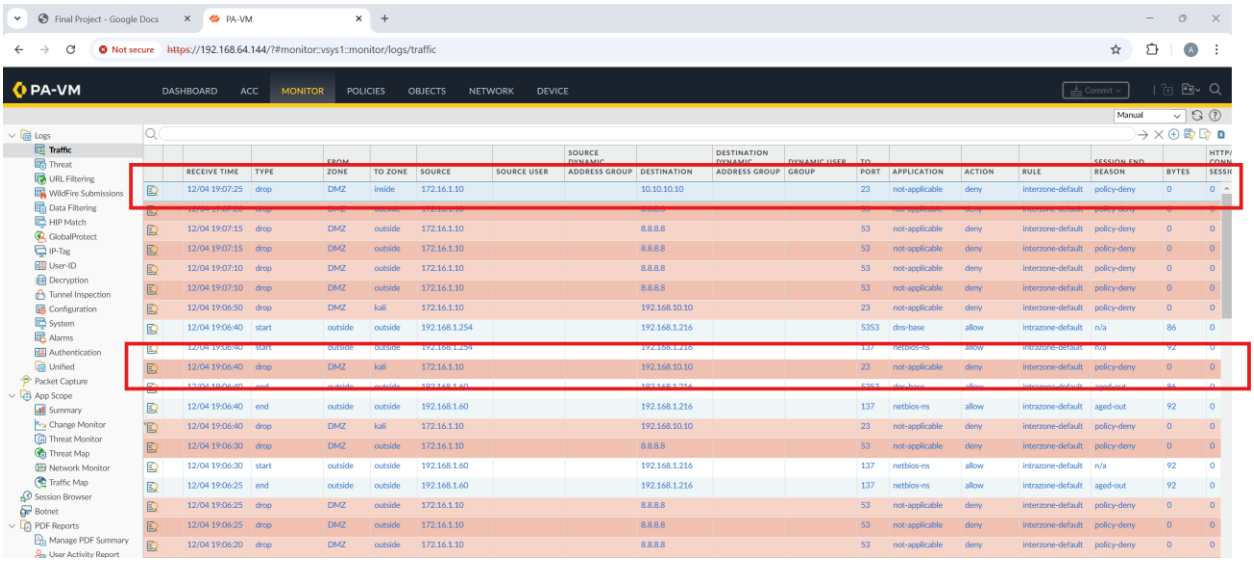


2. The below screenshot shows the deny request of ping from DMZ-host to other hosts and internet.

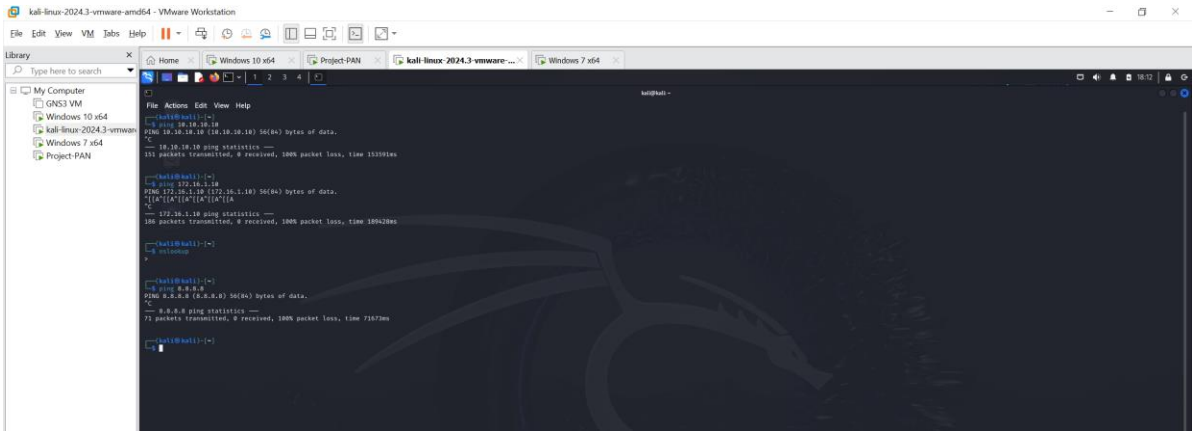


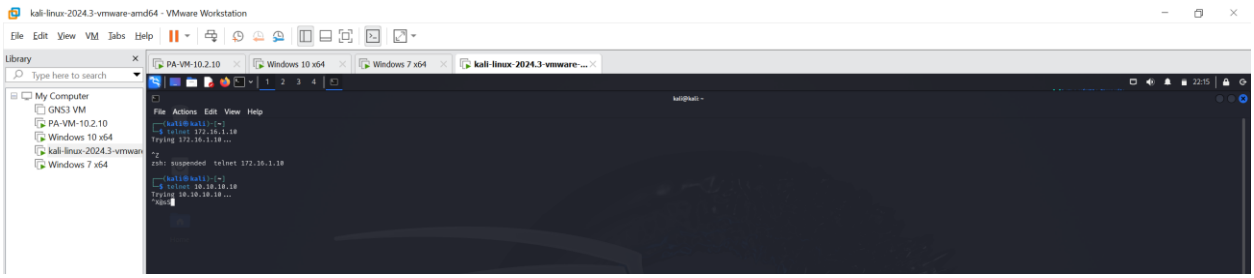
3. The below logs show the deny of the ping requests from DMZ-host to access the internet.



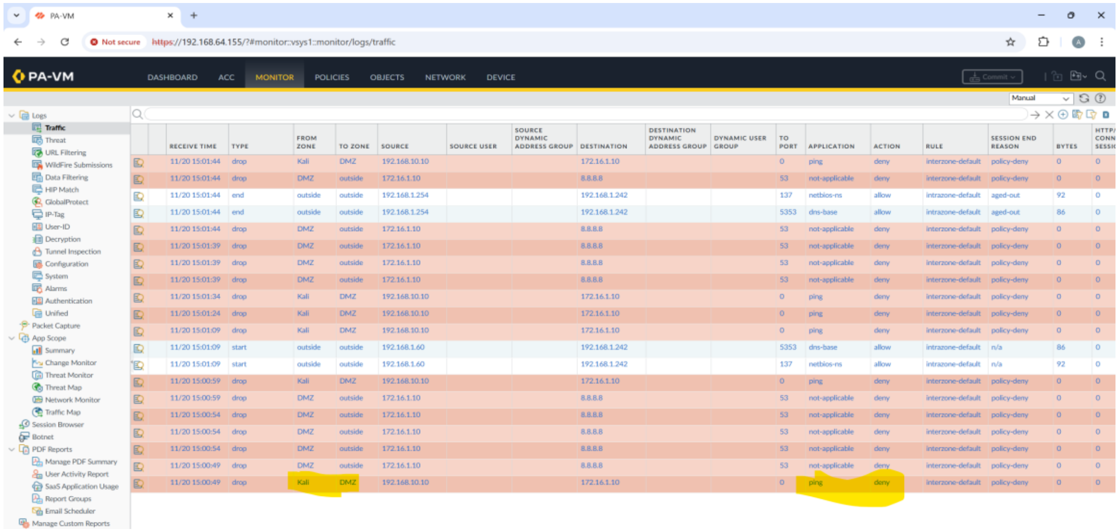


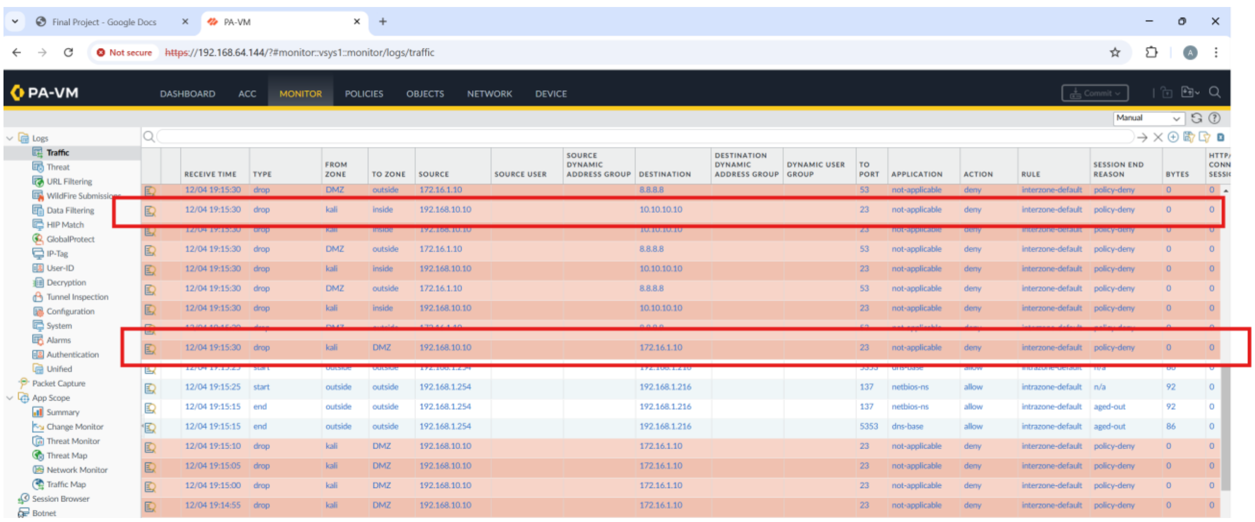
4. The below shows the deny of ping request from kali to other hosts and unable to access the internet.



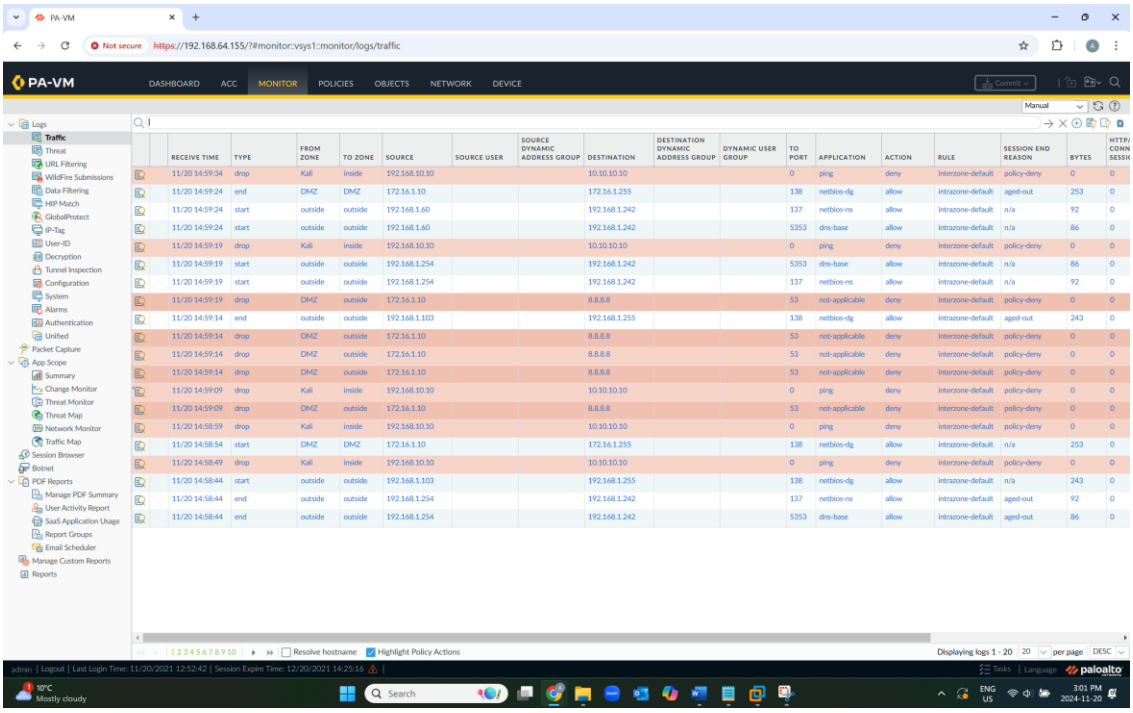


5. In the below logs showing the ping deny from Kali to DMZ.

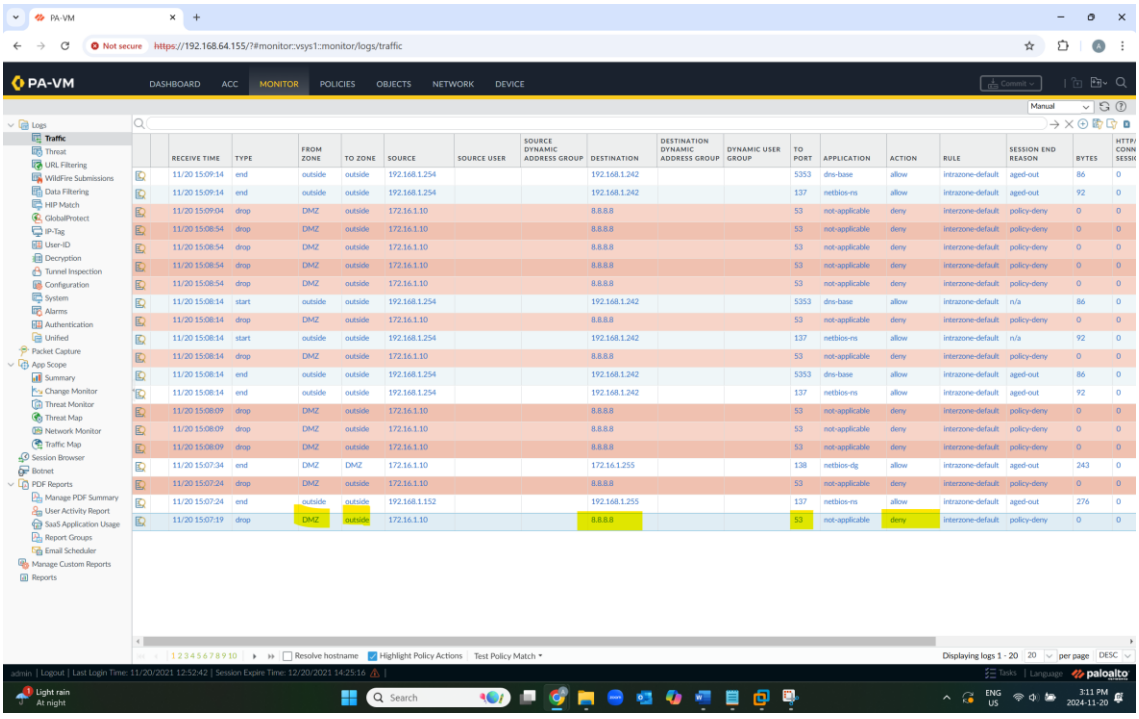




6. In the below logs showing the ping deny from Kali to inside-host.



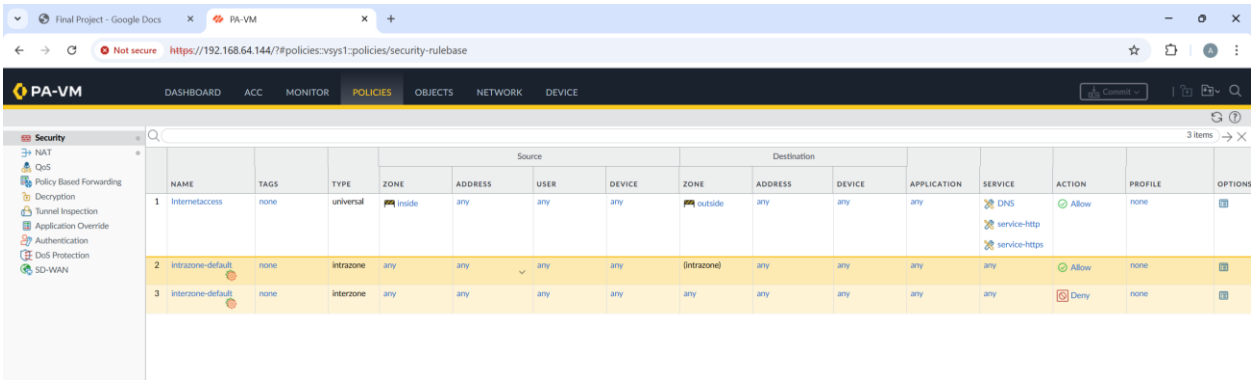
7. The below logs show the access to internet is denied from Kali when ping 8.8.8.8.



The above screenshots show the proof of **Zero-Trust concept** as they cannot access each other or the internet.

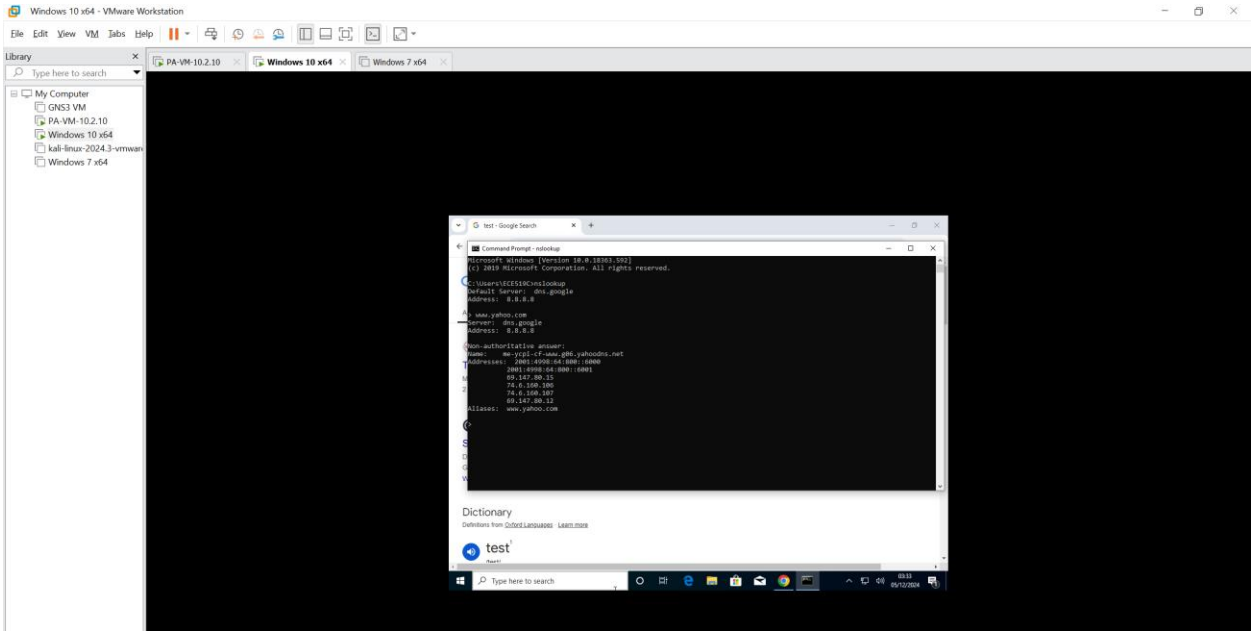
Q2. Configure firewall rules to allow Inside-Host Internet access (DNS, HTTP, HTTPS).

1. The below screenshot from PA-VM interface shows that a new rule named ***"internetaccess"*** has been configured from source zones as Inside-Host, and ***DNS on Port 53 with UDP protocol, service-http, service-https*** are configured in Service tab.

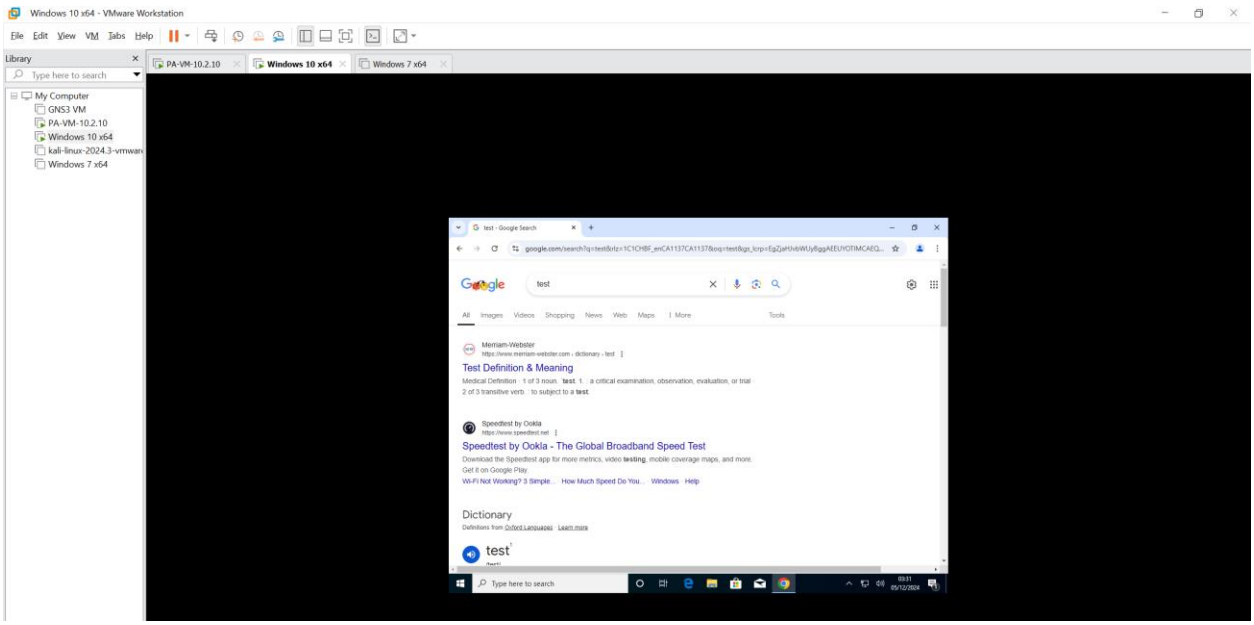


Q3. Demonstrate HTTP/HTTPS Internet access from Inside-Host with application awareness.

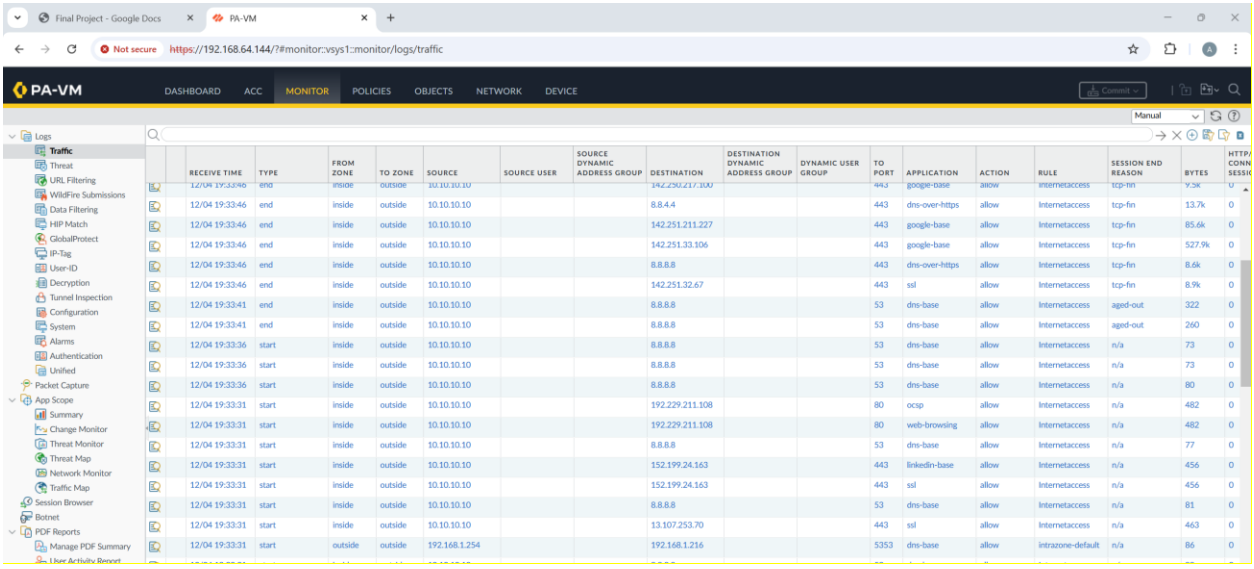
1. The below screenshot shows that I can access internet with **"nslookup"** in terminal which shows google server as we configured the rule in Question2.

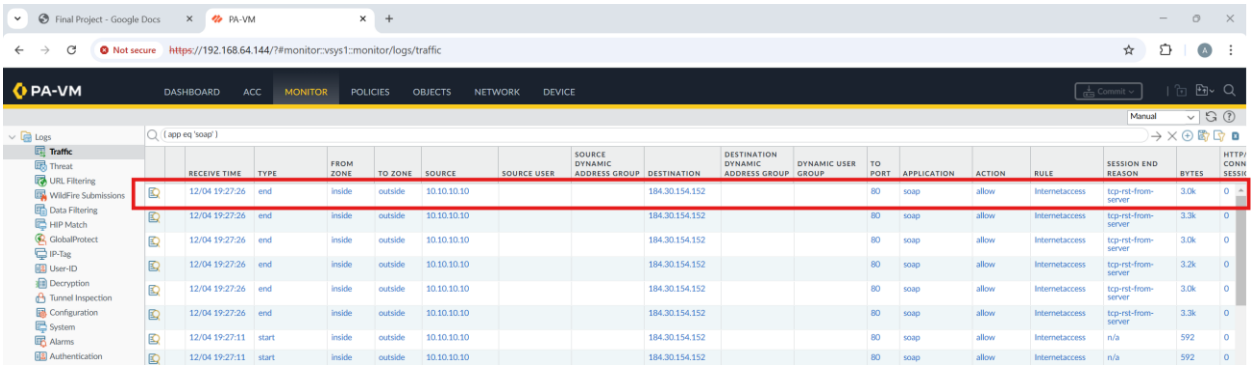


2. To test the above configured rule, we can browse for any information from the Inside-Host on the browser and it should be accessible as shown in below screen:



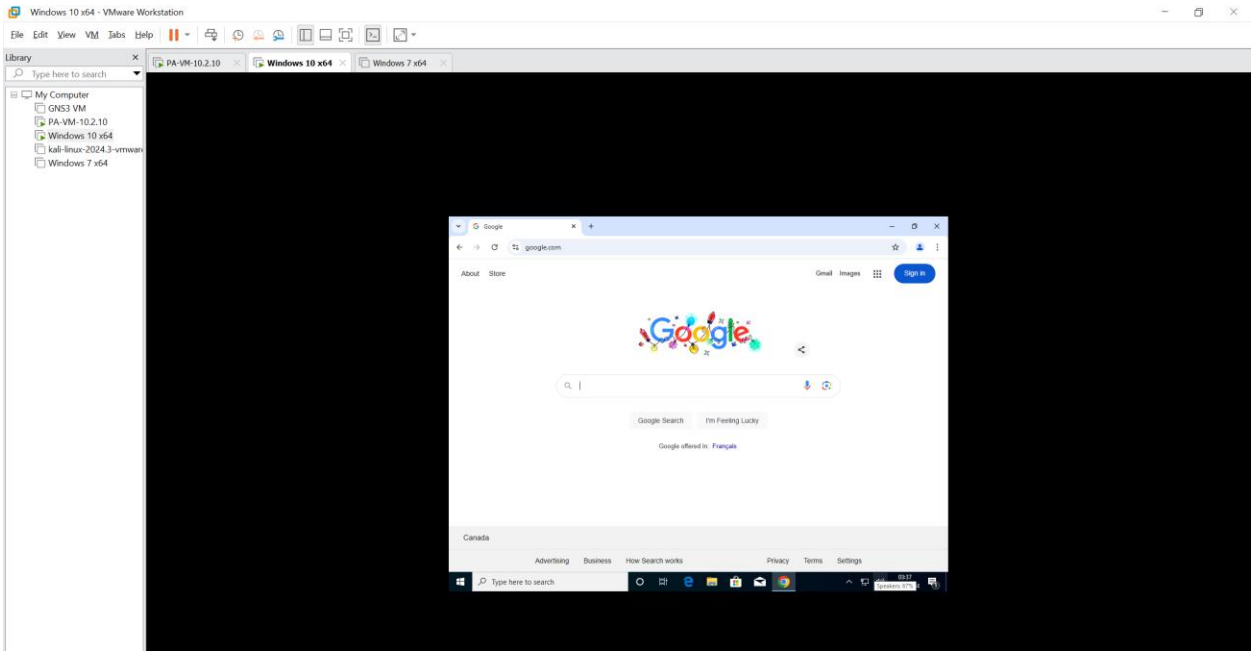
3. In the monitor tab shown below, we can see that the inside-host has internet access over HTTP (80)/HTTPS (443), and the PA-VM firewall.



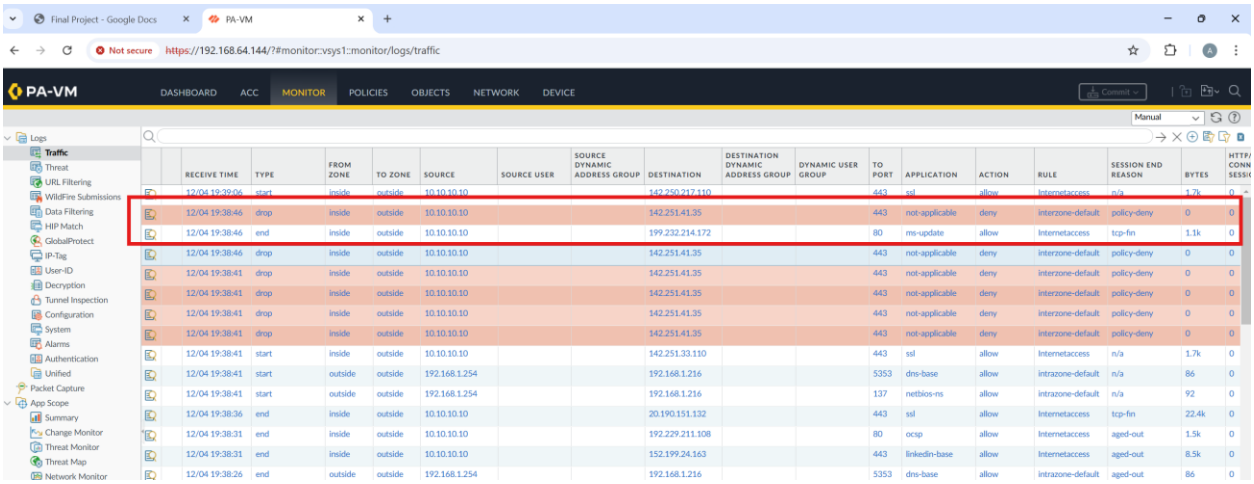


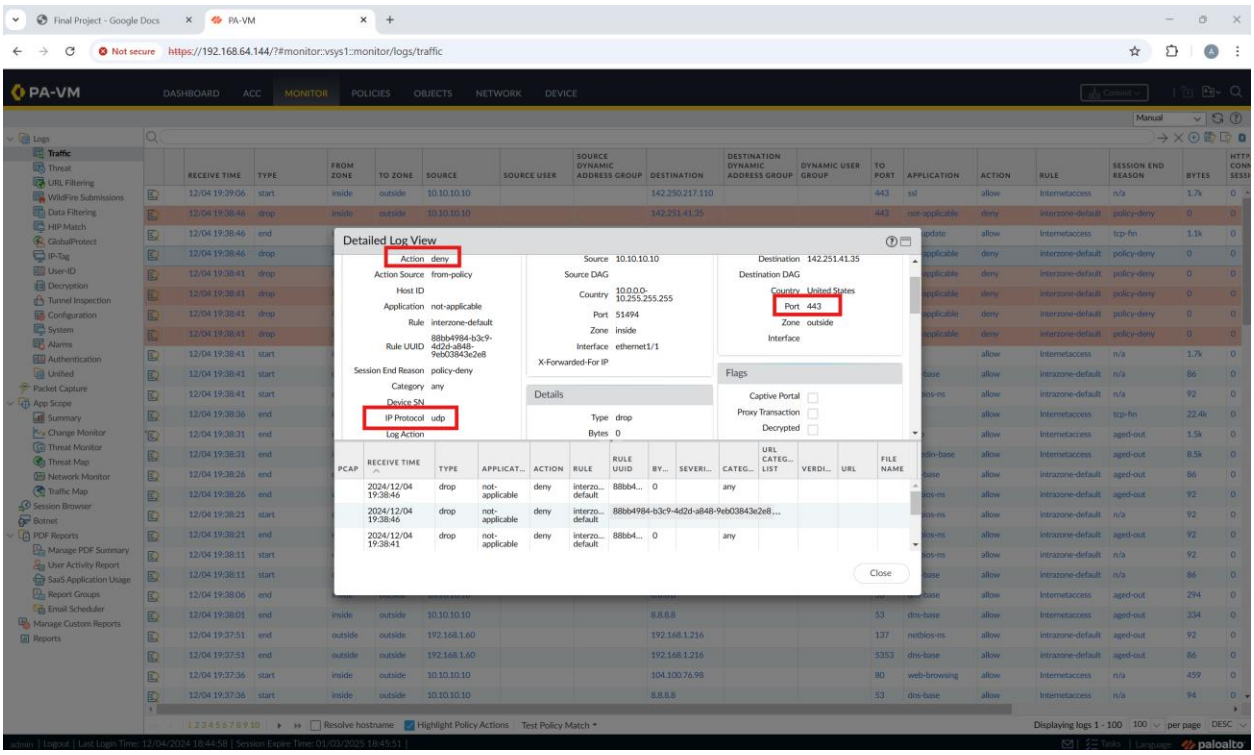
Q4. Download and use Google Chrome on Inside-Host to access https://www.google.com. Analyze denies in the Monitor tab.

1. The below screenshot shows that we download the chrome successfully and we accessed **https://www.google.com.**



2. When we observe the "Monitor" tab in PA-VM, we see some denies in the monitor tab initially, **because Quic protocol runs over 443 which used Ip protocol UDP.** The drop logs indicate the same as shown below:

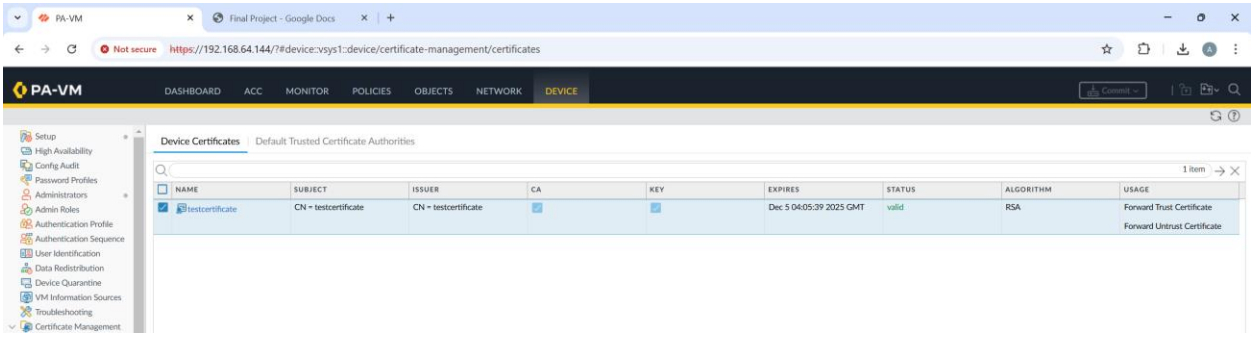




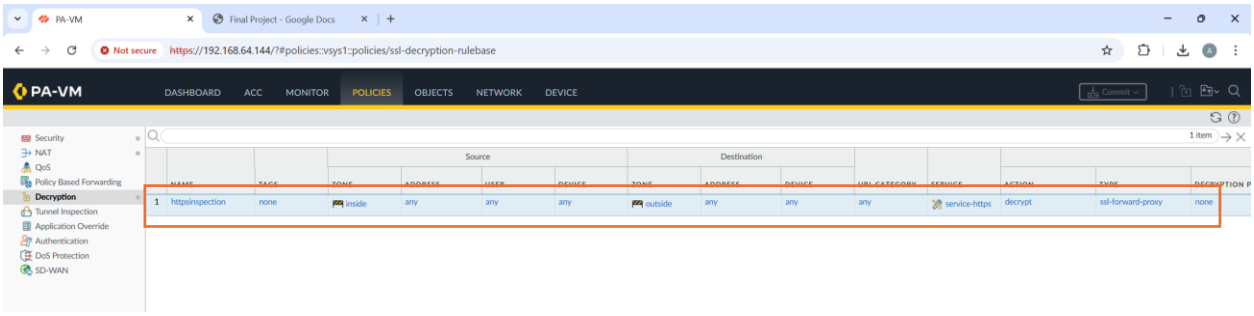
5. Apply HTTPS inspection for Inside-Host Internet traffic.

HTTPS inspection enables the firewall to decrypt and inspect HTTPS traffic, ensuring security policies are applied to encrypted communications.

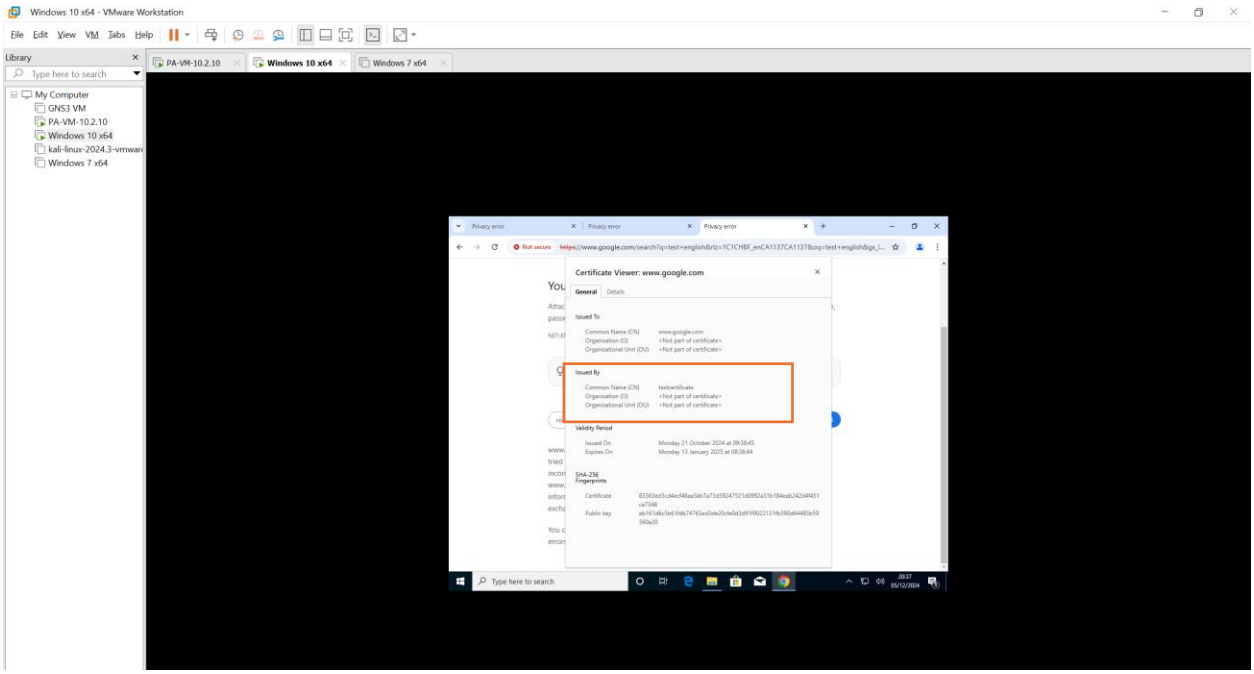
1. Generated a device certificate under devices→ certificate tab with name **“testcertificate”** with Forward Trusted Certificate and Forward Untrusted Certitifcate check.

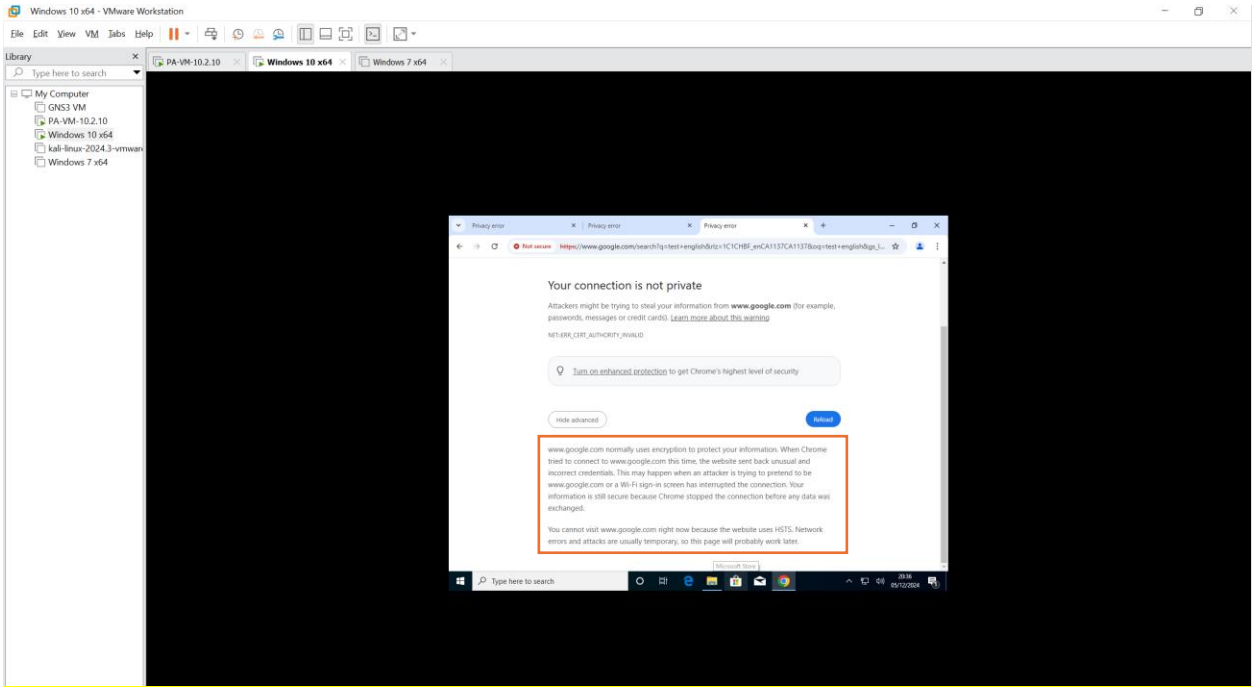


2. We add a decryption rule under the policies tab of PA-VM under Decryption tab as shown in the below screen, it will decrypt all the traffic from the Inside-Host:

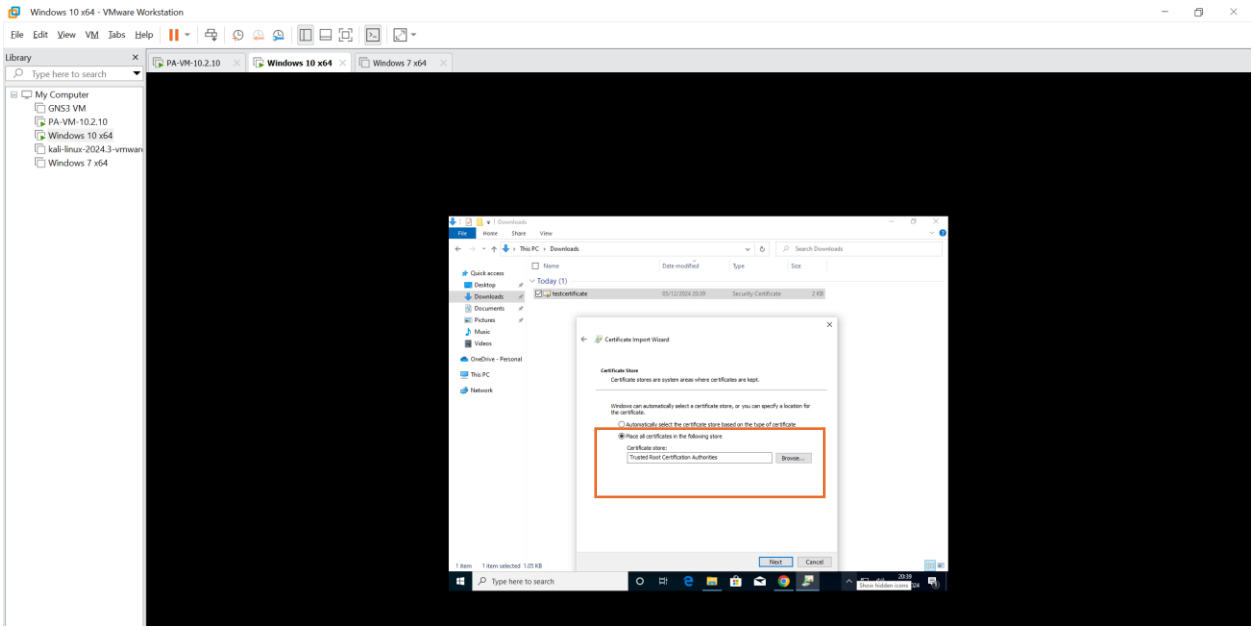


3. When we browse anything on chrome browser it is showing not secure as the certificate is not installed under trusted root, once installed it will not show this error as same demonstrated in below steps:

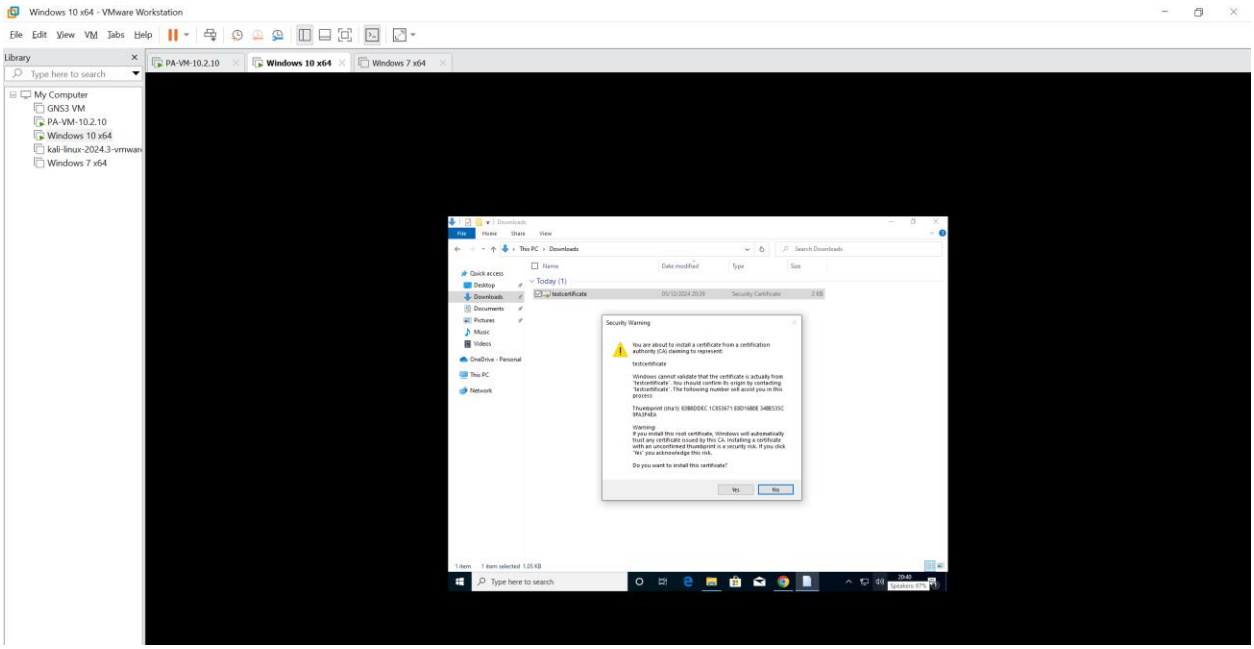




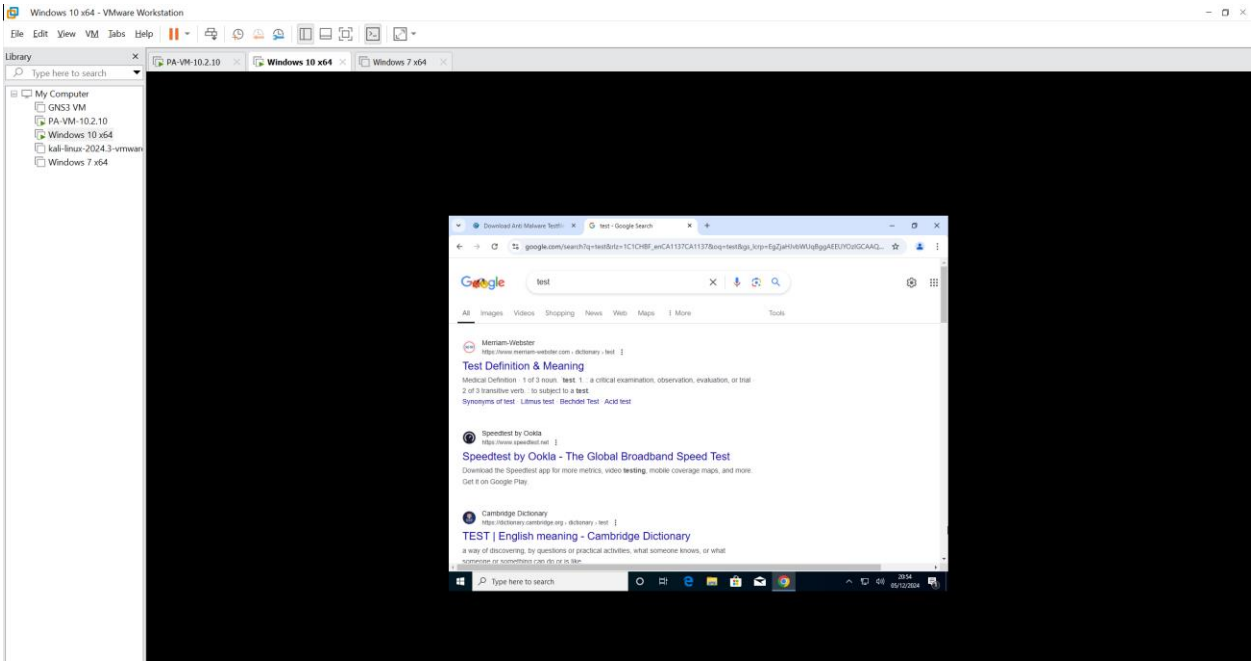
4. We imported the certificate under Trusted Root Authorities as shown below:



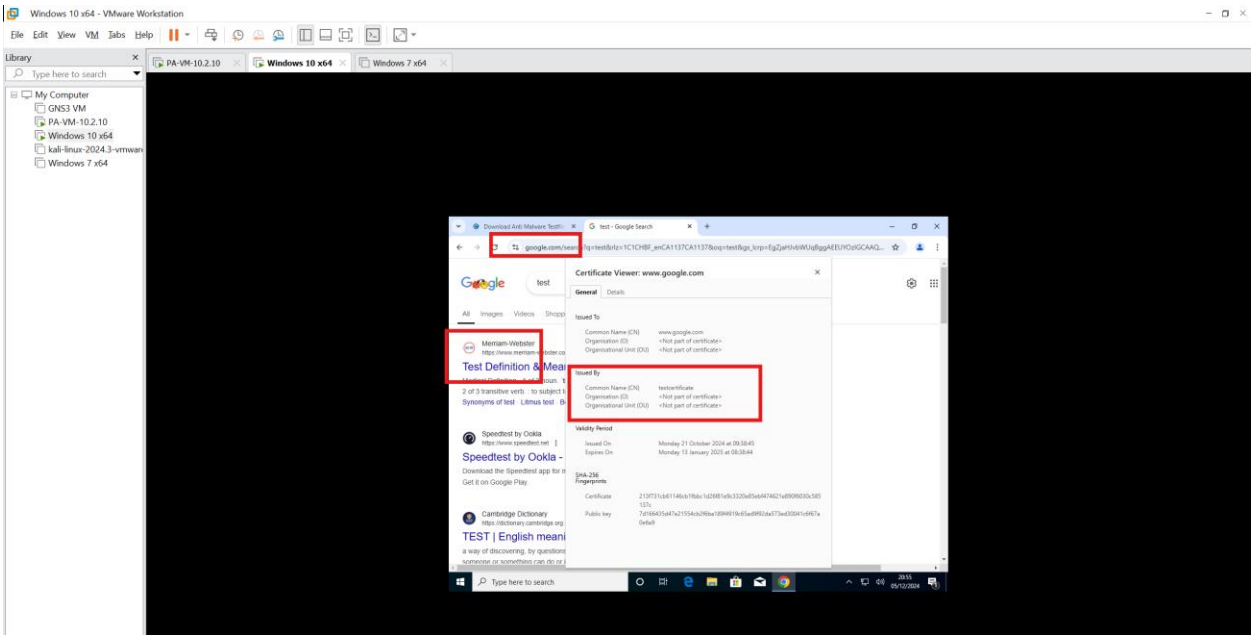
5. We get the warning with out Certificate name: **“Testcertificate”** which shows that it is properly generated.



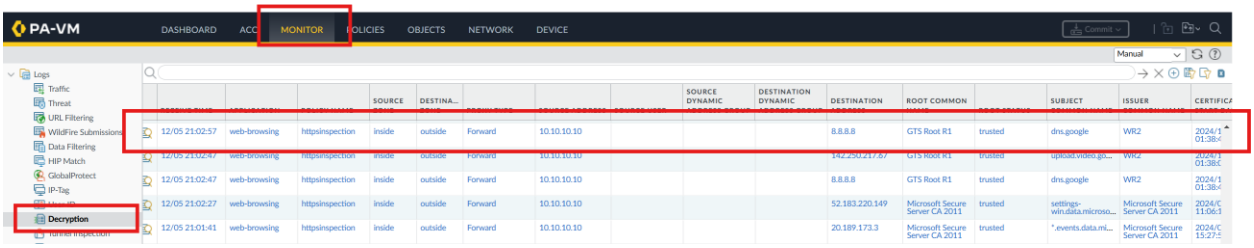
6. After importing the certificate, we can see below we are able to search in the browser and connection is secure:



7. The below screenshot shows that connection is secure, and it is showing the certificate name which is from PA-VM:

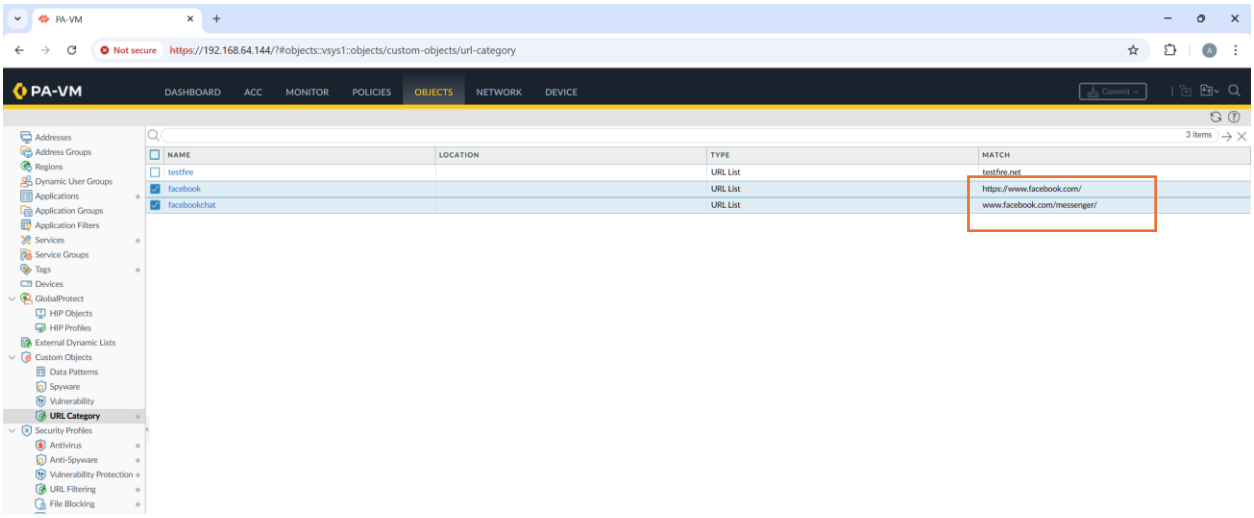


8. The below screen shows that the access has been allowed in PA-VM under Monitor Tab:

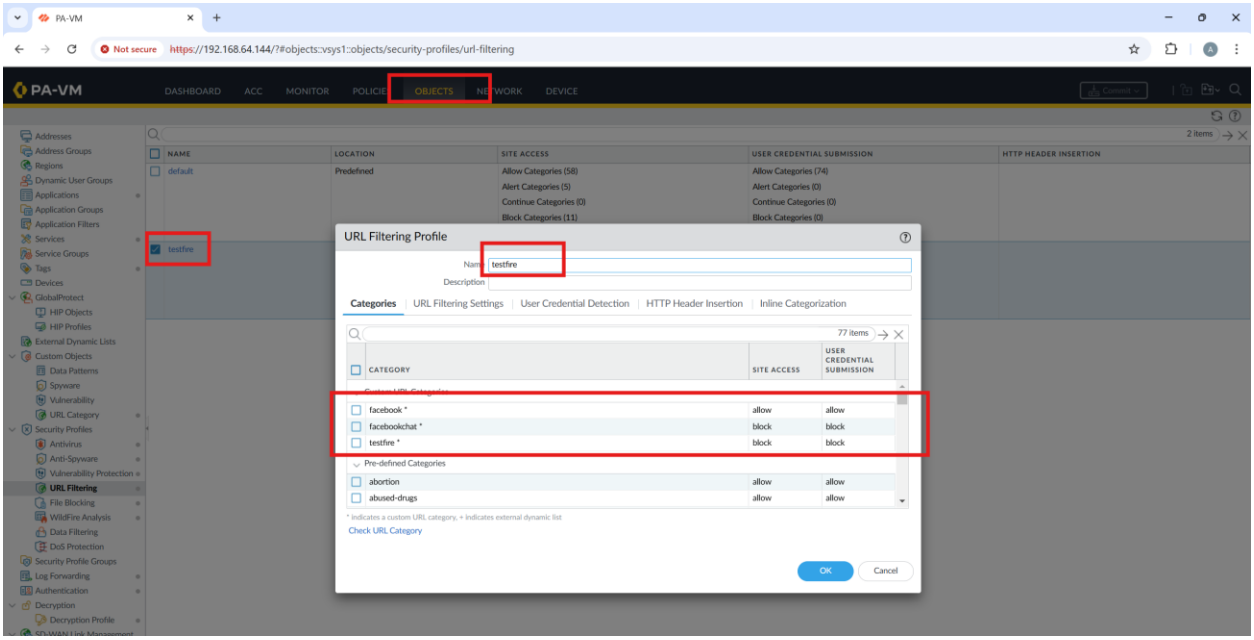


Q6. Configure URL filtering to allow Inside-Host access to Facebook but block Facebook Chat.

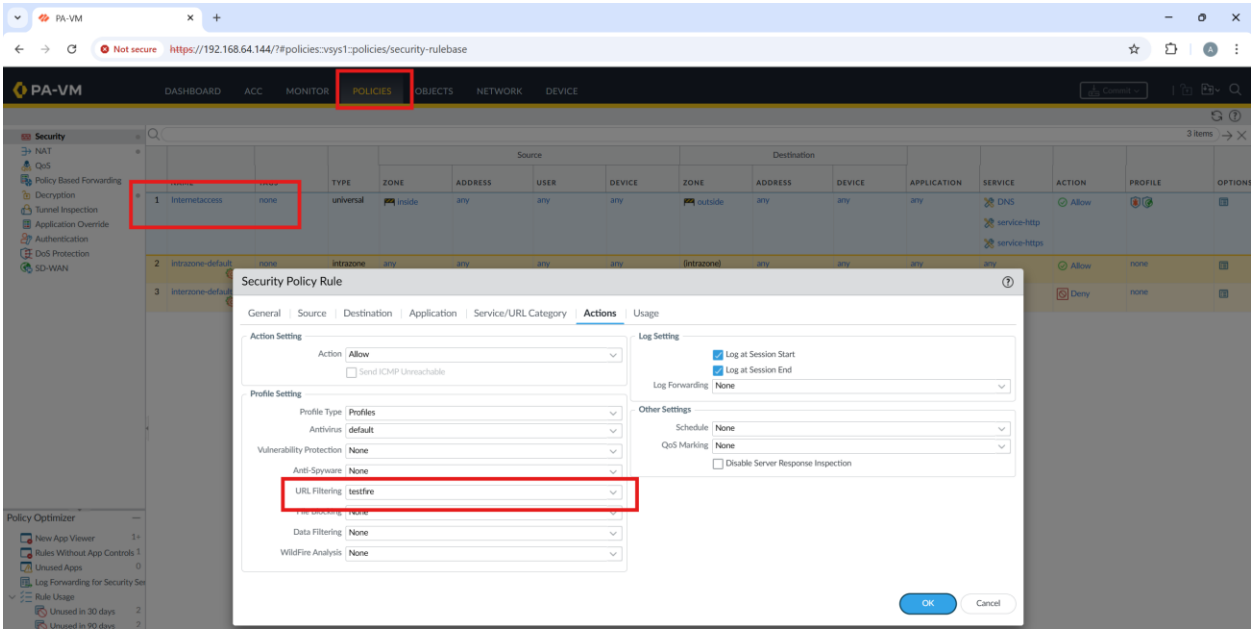
1. To Test the URL Filtering in Inside-host for Facebook and Facebook Chat we added a custom URL inside URL Category for Facebook as https://www.facebook.com and Facebook-Chat as www.facebook.com/messenger/ as shown below:



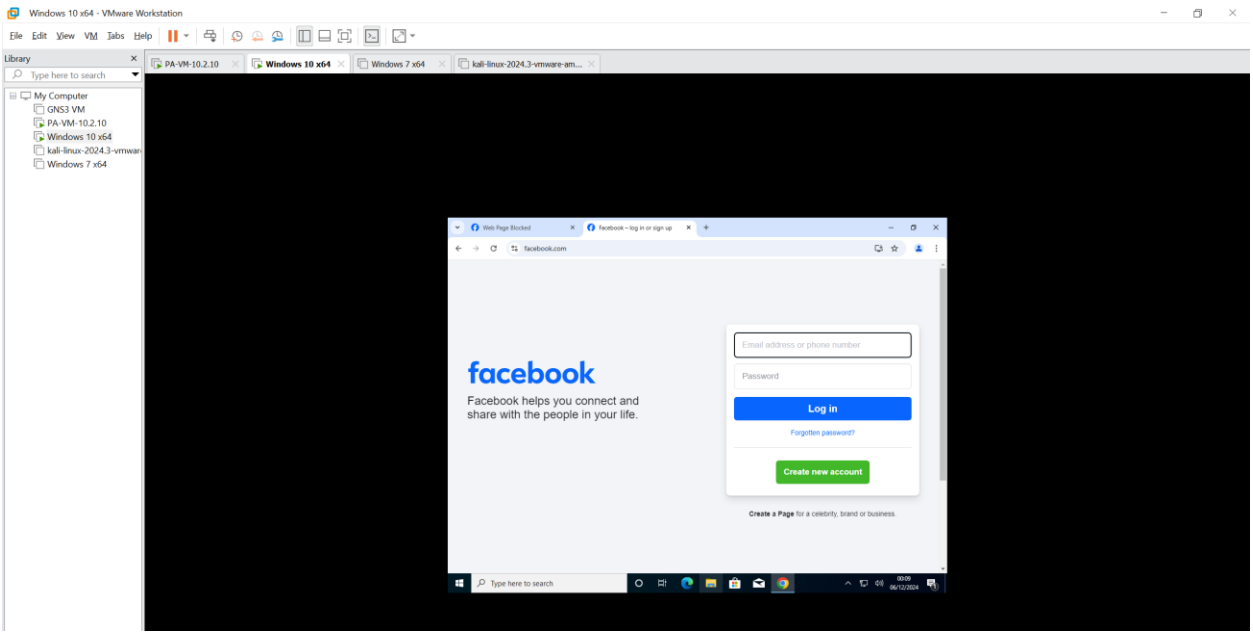
2. Under Objects → URL Filtering we add this rule under **"Testfire"** name: where we allowed the Facebook and blocked the Facebook Chat as shown in below snip.



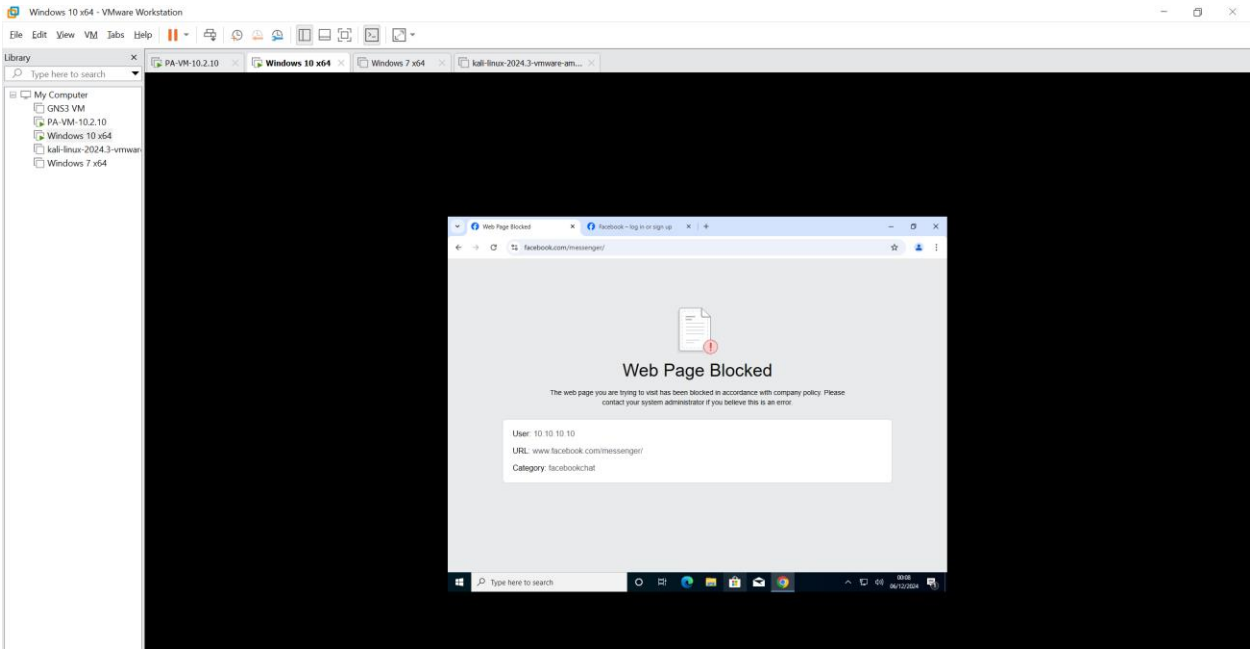
3. We allowed this under URL Filtering in **"Internetaccess"** rule from inside to outside under the **Profiles** tab in Actions and selected **"testfire"** rule which we created above as shown below screen:



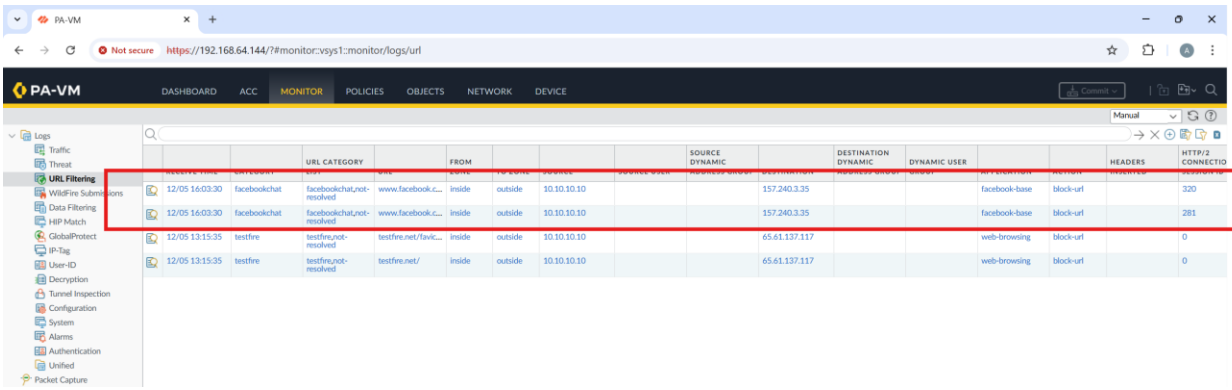
4. To test the above configured rules, we browsed for **facebook.com** on chrome browser on the Inside-Host machine and page gets loaded as shown below:



5. To test the above configured rules, we browsed for **Facebook-Chat** with **"facebook.com/messenger/"** URL on the Inside-Host machine and error is shown as **Web Page Blocked** as shown below:

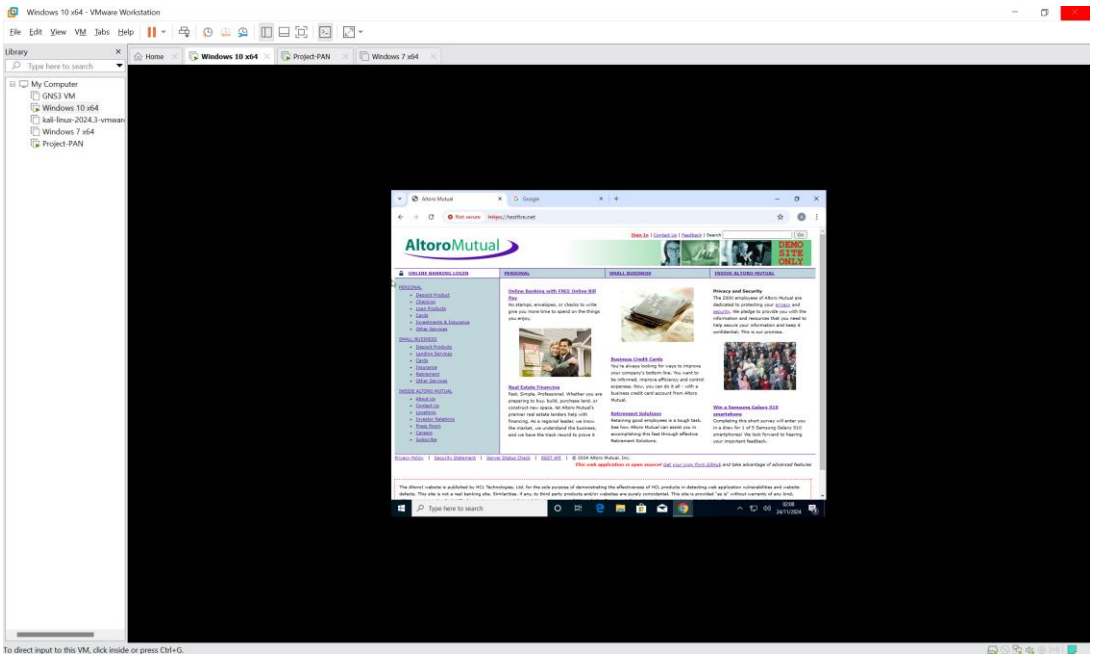


6. The above point can be noticed in the below screenshot under Monitor tab in **URL Filtering** tab as **Facebook-chat** gets denied. This is how we allowed the Facebook and blocked Facebook chat with the help of URL Filtering:

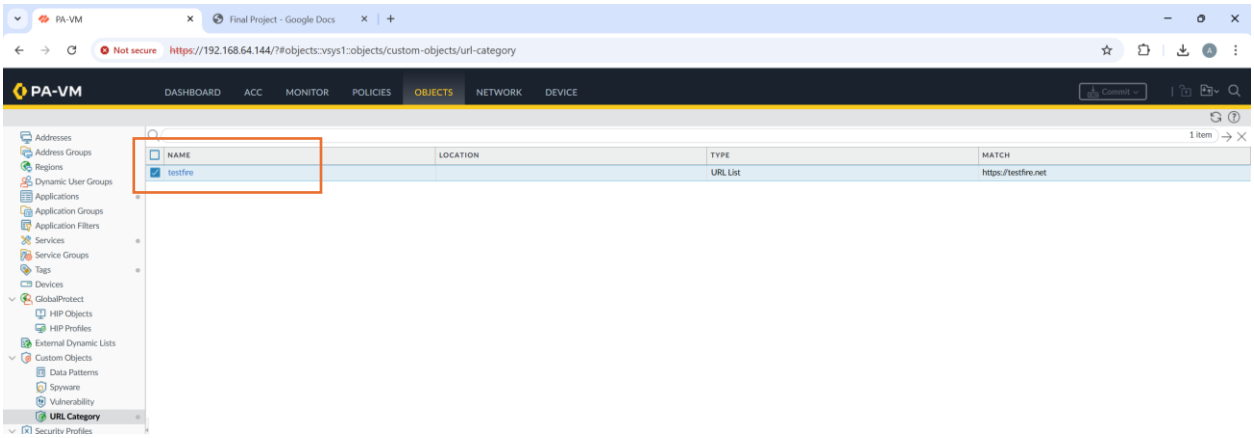


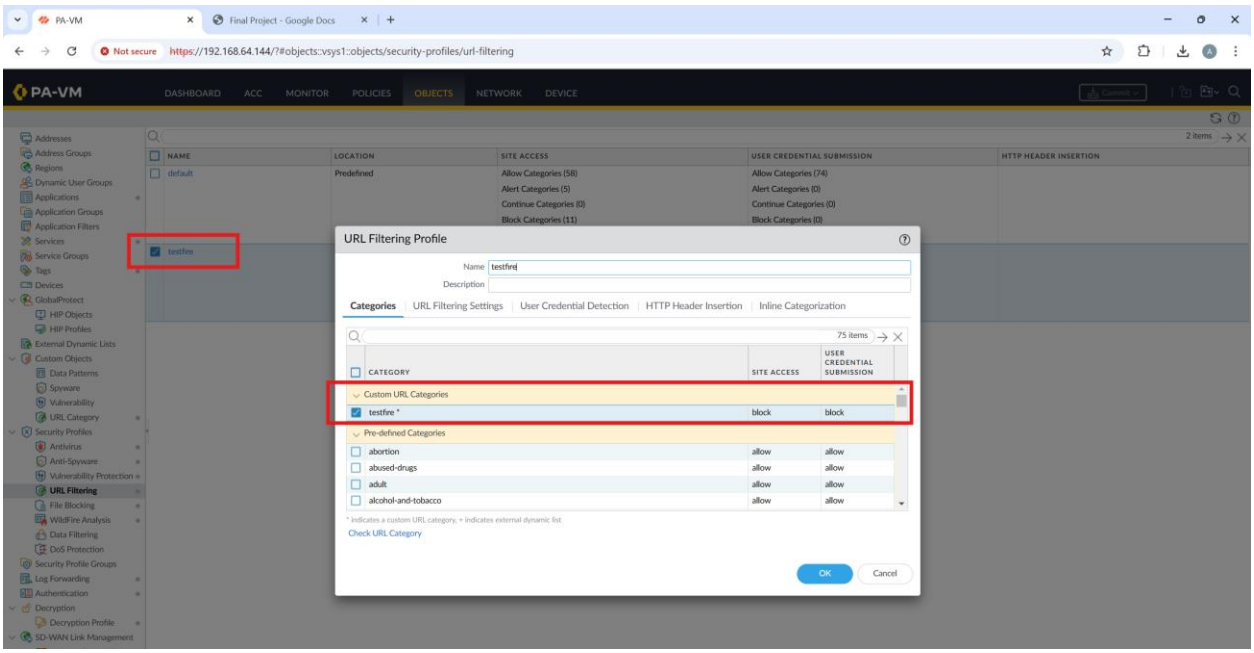
Q7. Use URL filtering to block Inside-Host access to testfire.net

1. If you search in browser just now without any rule the page is working when you access testfire.net.

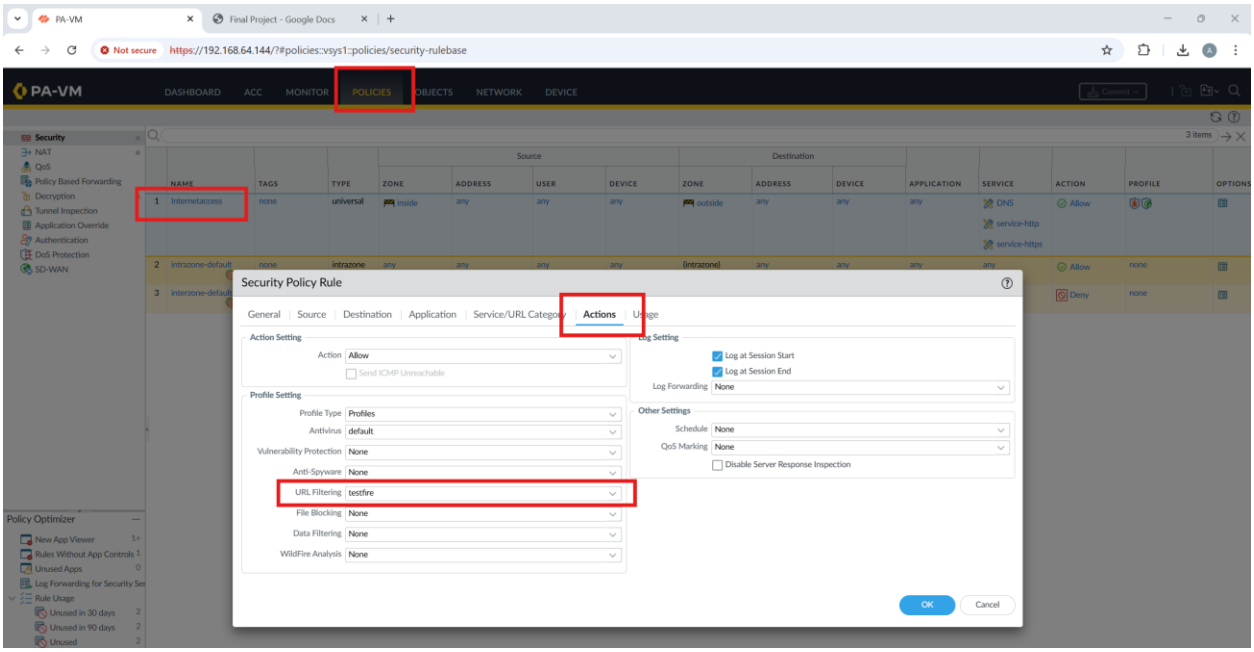


2. To block Inside-Host machine from accessing the specified URL, we first add the given URL (i.e. testfire.net) in the URL Category under the “Objects” tab of the PA-VM firewall:

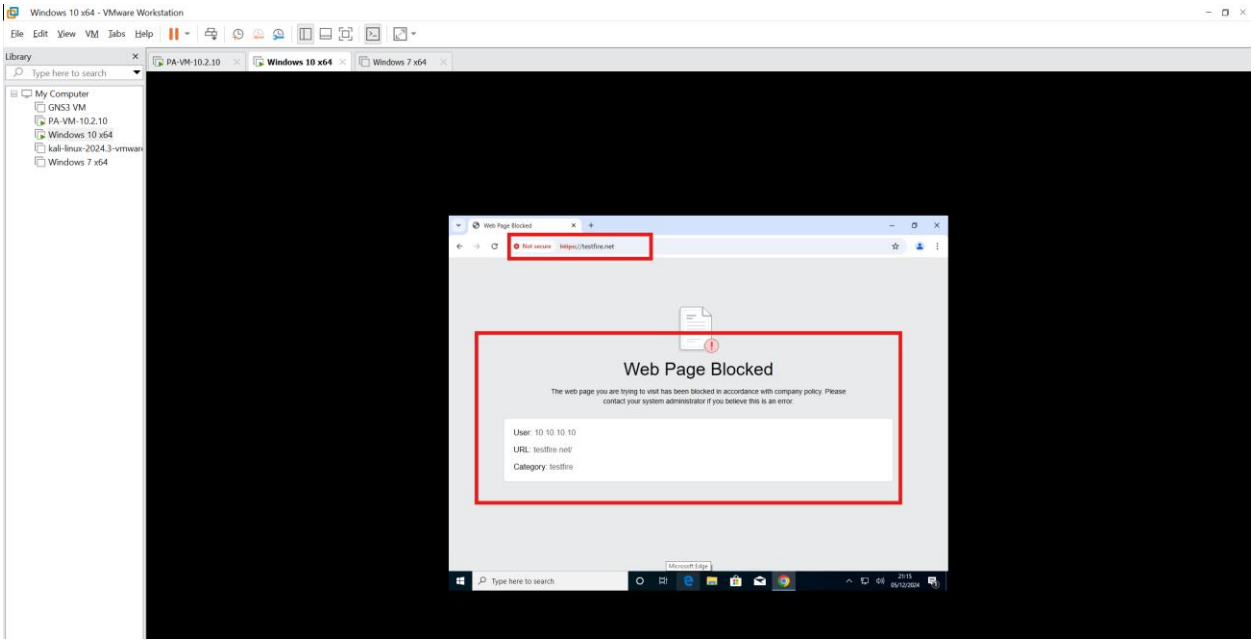




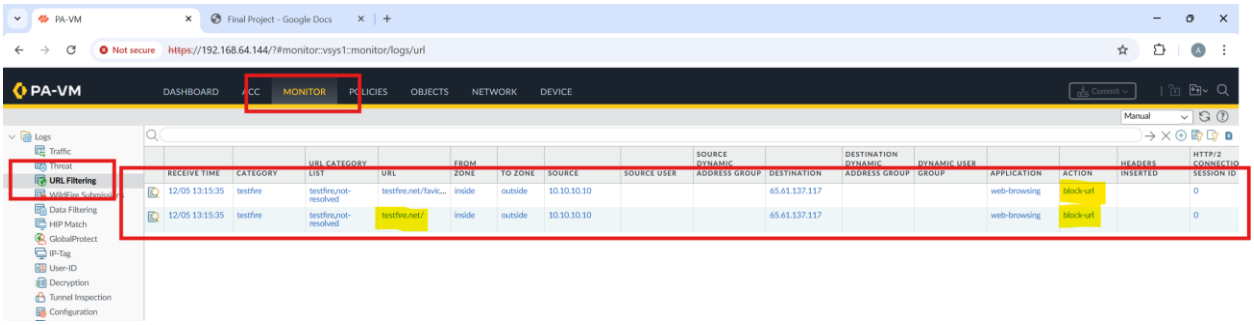
3. We configure a security rule under the policy tab of PA-VM interface with the name as "*Testfire*" as shown below:



4. To test the above configuration, browse for testfire.net from the Inside-Host machine, we see that **testfire.net** has been **blocked**, and we can see in the below screen it is showing **Web Page Blocked** and URL and Category and User: 10.10.10.10.

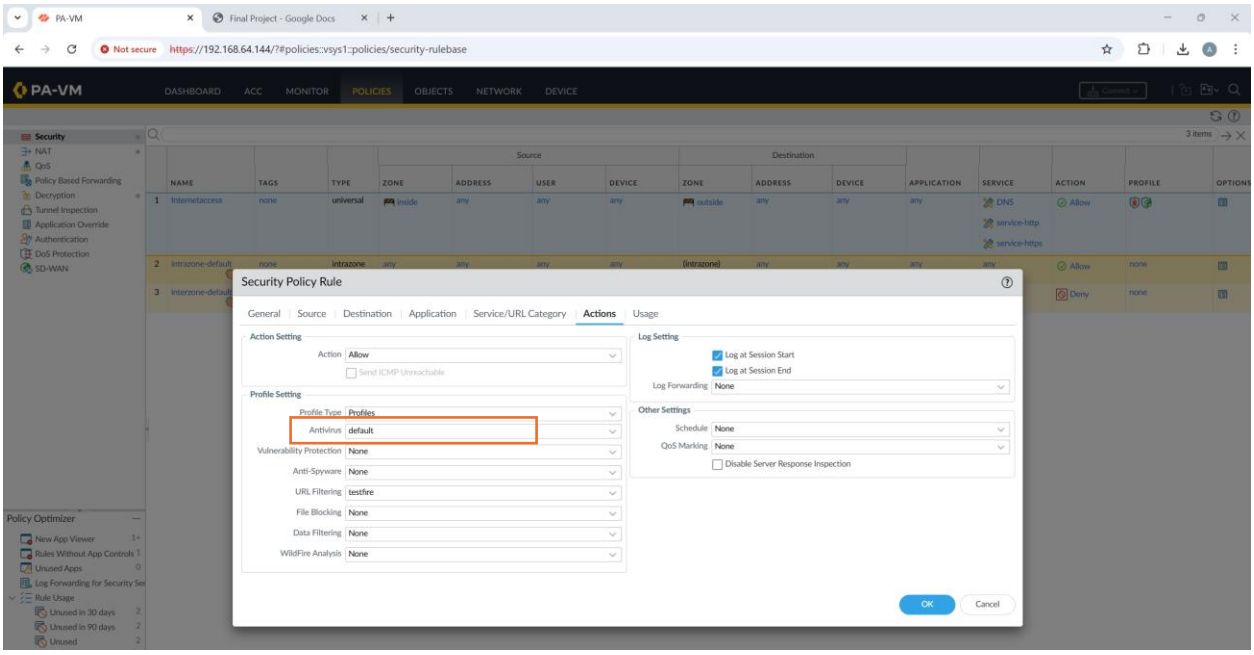


5. The monitor logs shown below indicate that the URL testfire.net has been blocked from the Inside-Host under Monitory inside URL Filtering in PA-VM:



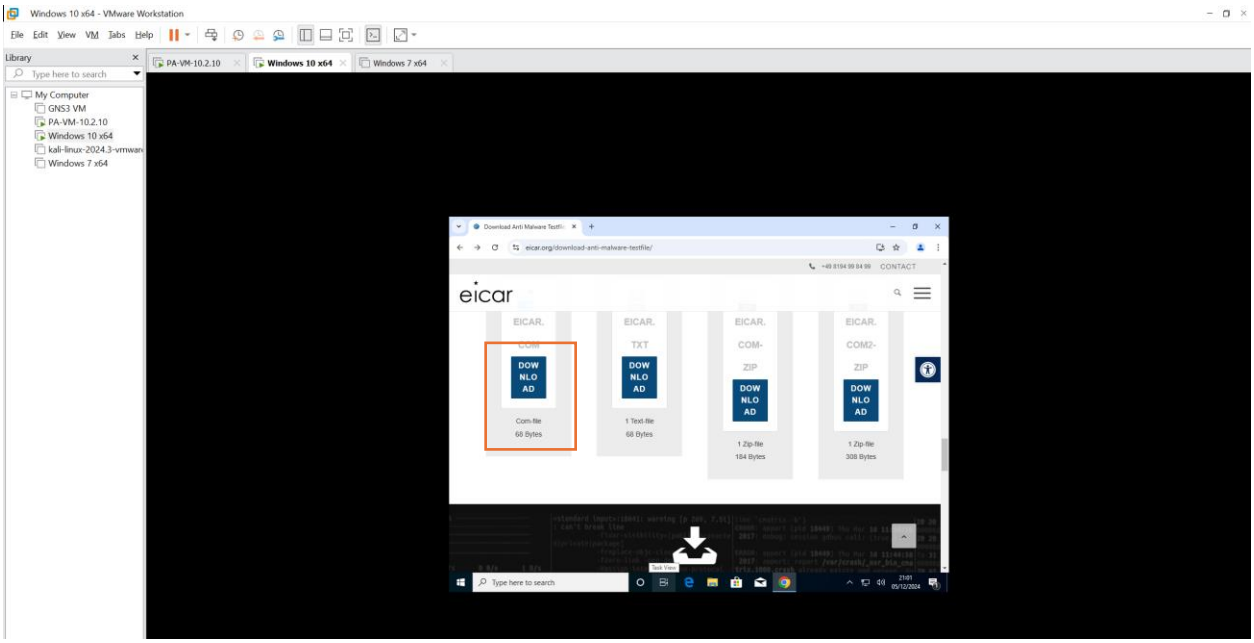
Q8. Apply antivirus inspection for Inside-Host Internet traffic.

1. To configure this rule, we allowed the Antivirus Inspection under profiles tab to Default in **"Internetaccess"** rule which we created earlier.

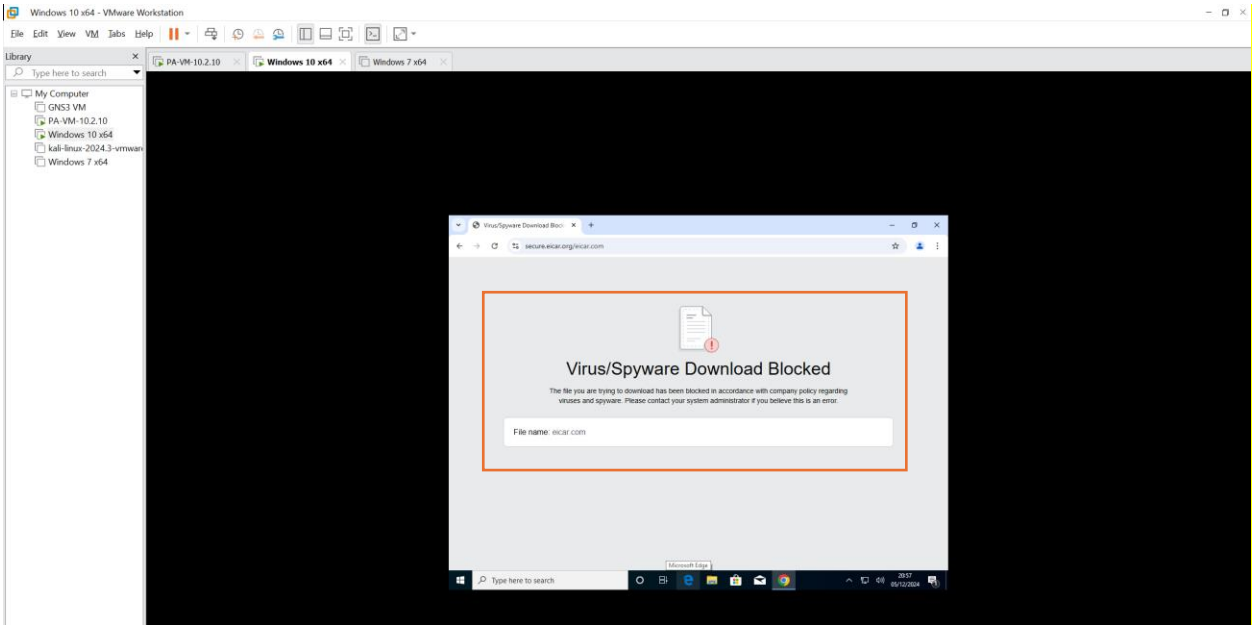


Q9. Attempt to download the eicar test virus from Inside-Host; illustrate the outcome.

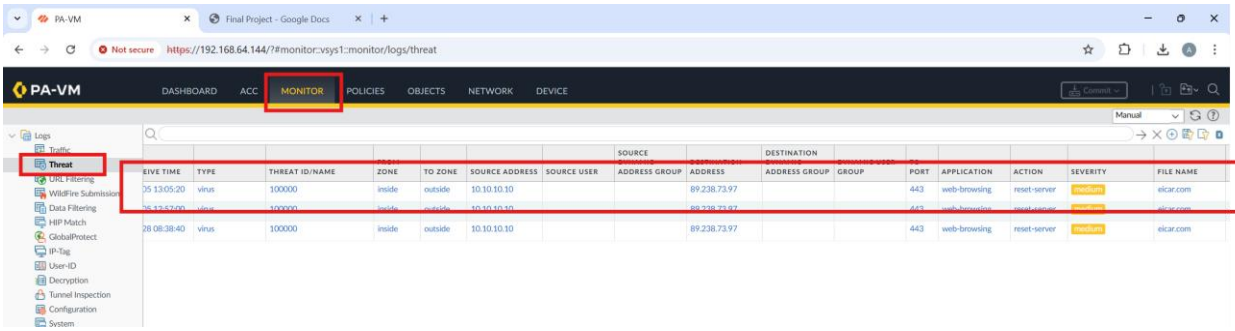
1. To test the download, we navigated to **"eicar.org"** website and performed the download for any of the files shown in below screen



2. We can see in below snip it shows the message **"Virus/Spyware Download Blocked"** as we allowed the Antivirus inspection earlier, we blocked the malicious download.

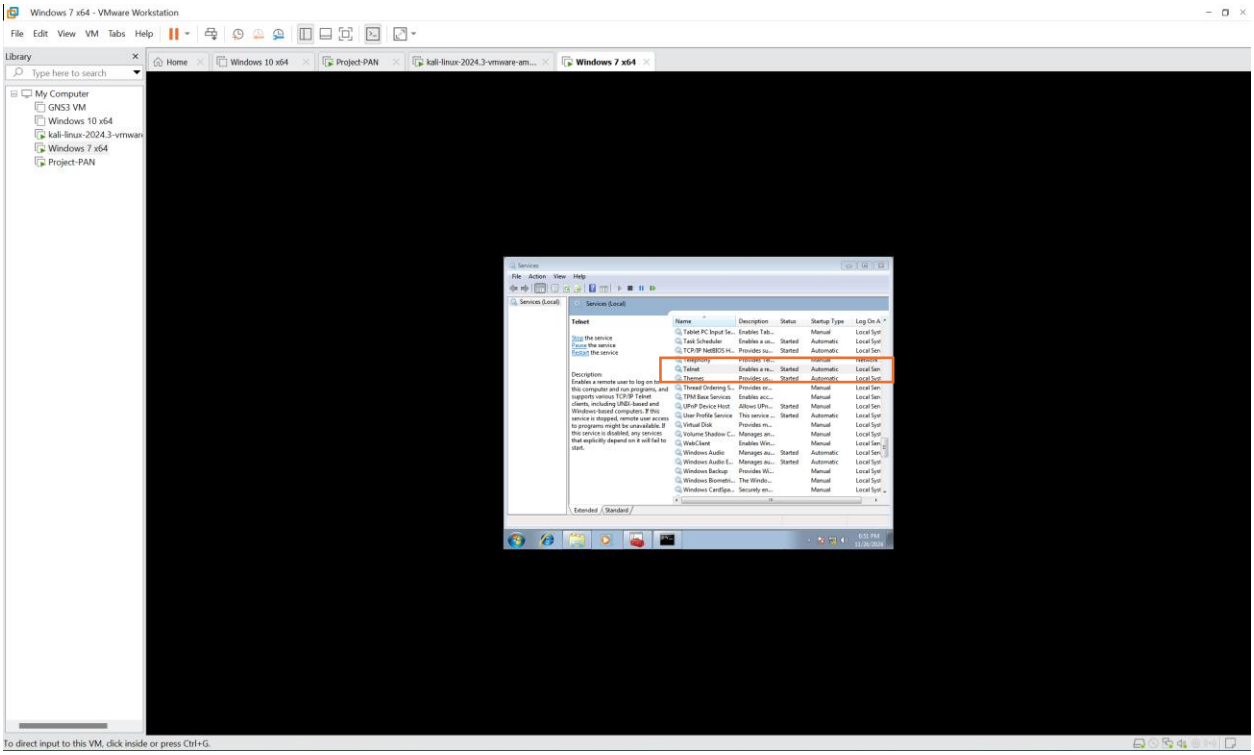


3. We can see the same in Monitor tab in Threat and type is virus and File name: eicar.com with Medium Severity.

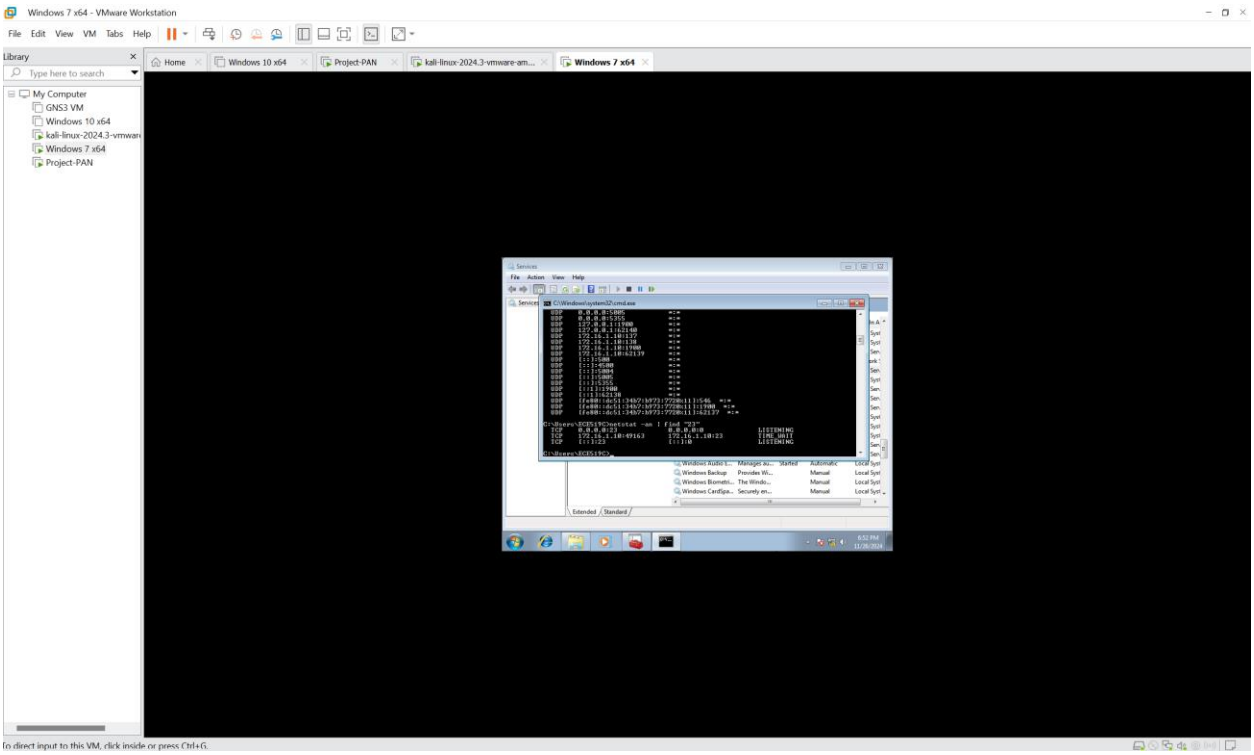


Q10. On DMZ-Host, ensure the Telnet Server is running.

1. We started the services in DMZ-host under **"services.msc"** tab to ensure that telnet server is running.



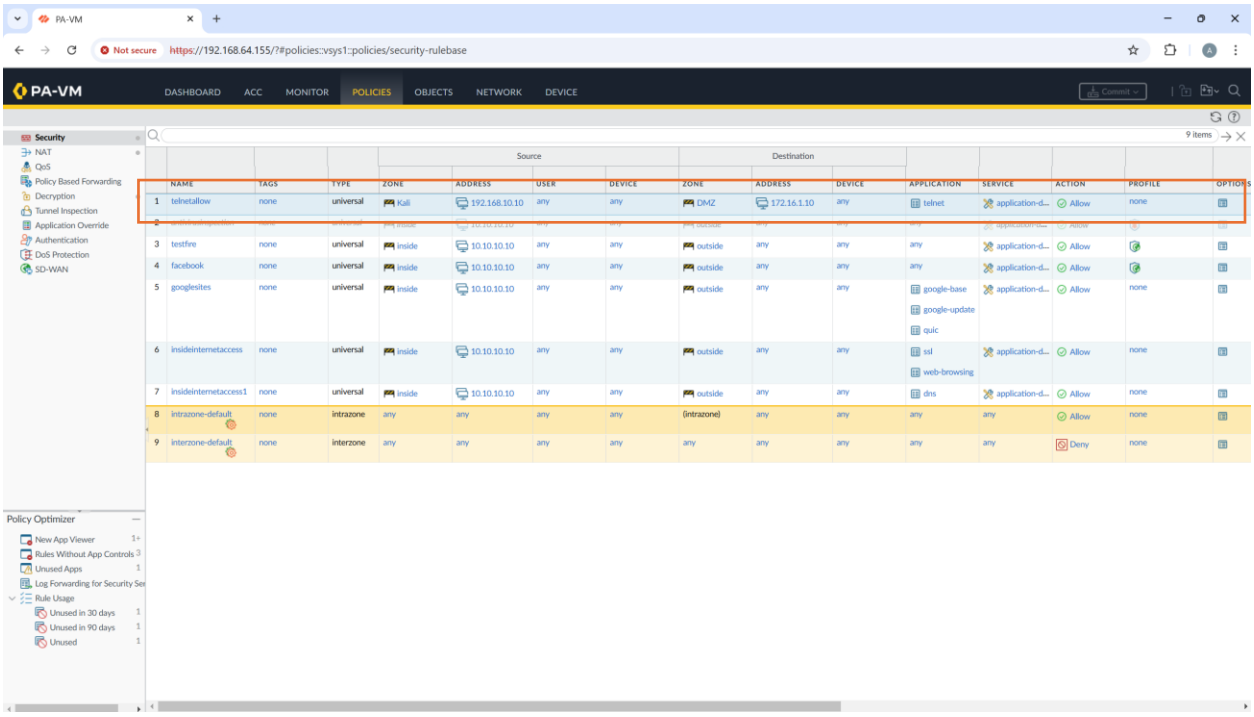
2. To check that Telnet services are running, typed the command in the windows terminal "netstat -an | find "23" as shown in below screen.



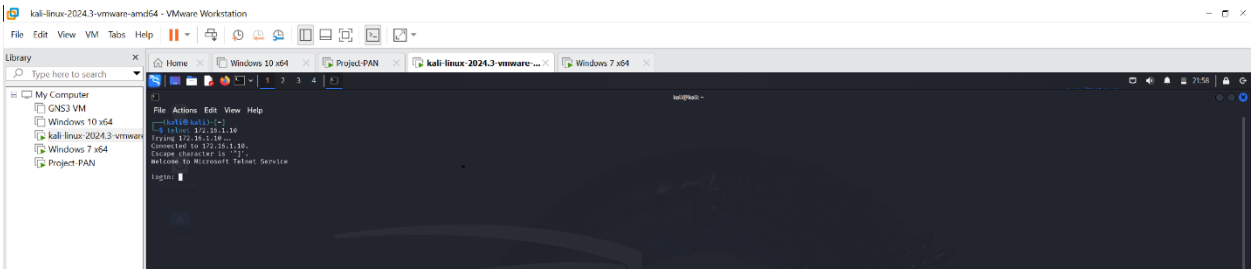
The above screenshots prove that the service is running on the DMZ-host machine on 172.16.1.10.

Q11. Allow Kali-Linux access to the DMZ Telnet Server using application awareness rather than port numbers. (Note: These screenshots are from my previous PA-VM Ip: 192.168.64.155 from Question 11 onwards)

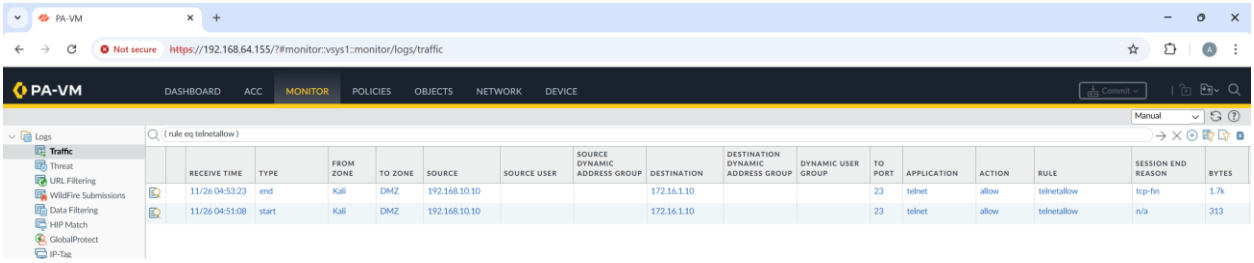
1. To test the application awareness of Telnet Server, we configure below rule in policies with name **"telnetallow"** for Kali-Linux to access the DMZ-host on application: **Telnet**.



2. Below screen shows that kali has access to DMZ-host on telnet application, command used: **telnet 172.16.1.10**.

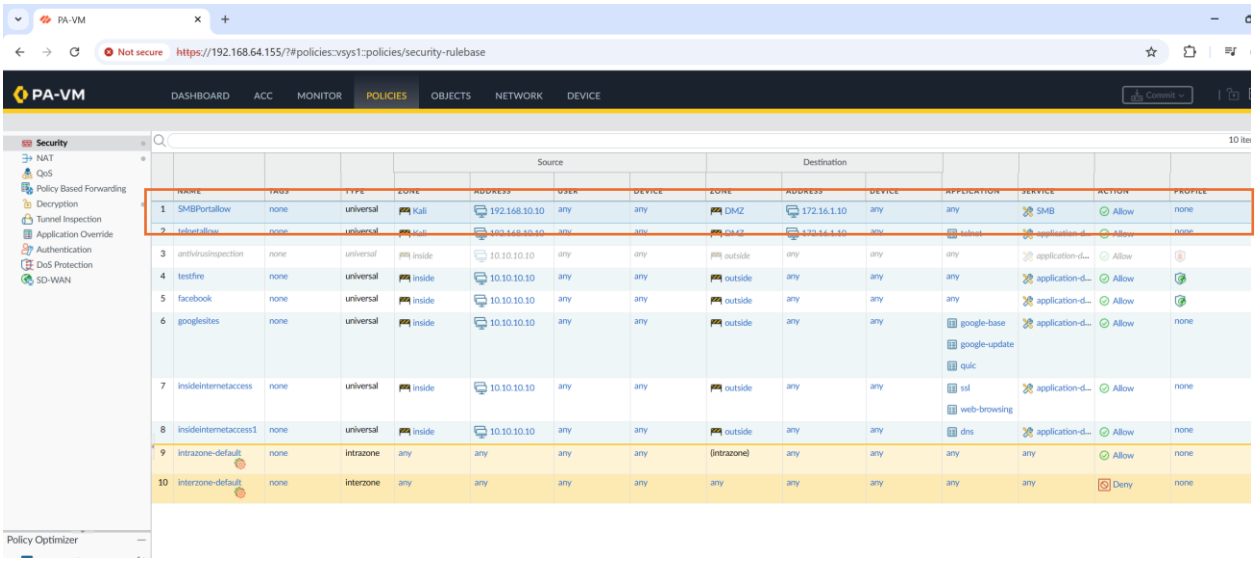


3. The below logs shows that the rule is hit which is **"telentallow"** under Monitor tab in Traffic:

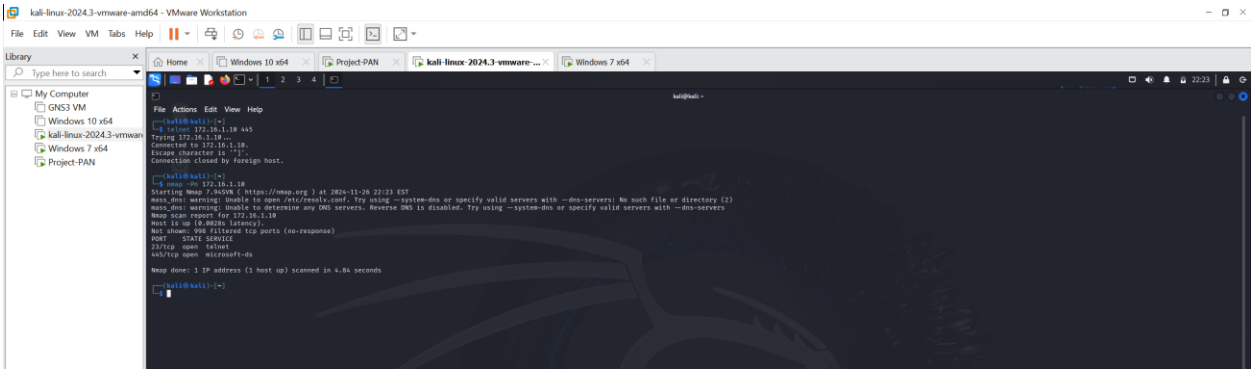


Q12. Allow Kali-Linux to access DMZ-Host over port 445.

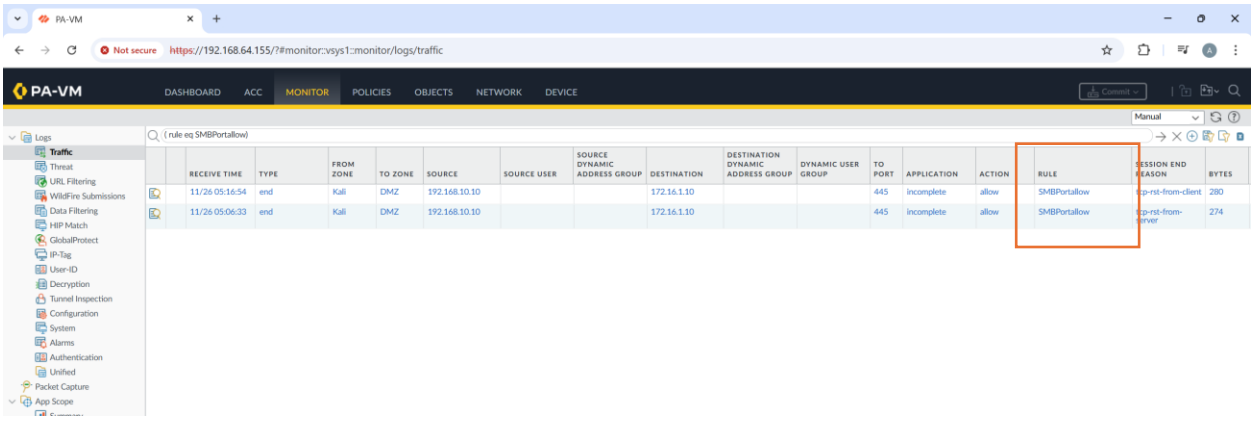
1. To test the scenario, we configure the firewall rule with name **"SMBPortallow"** from Kali to DMZ host with service SMB enable and action: **allow** under policies tab in Security.



2. We tested the rule and did **"nmap -Pn 172.16.1.10"** in terminal in Kali Linux machine it shows **445** port **open**.

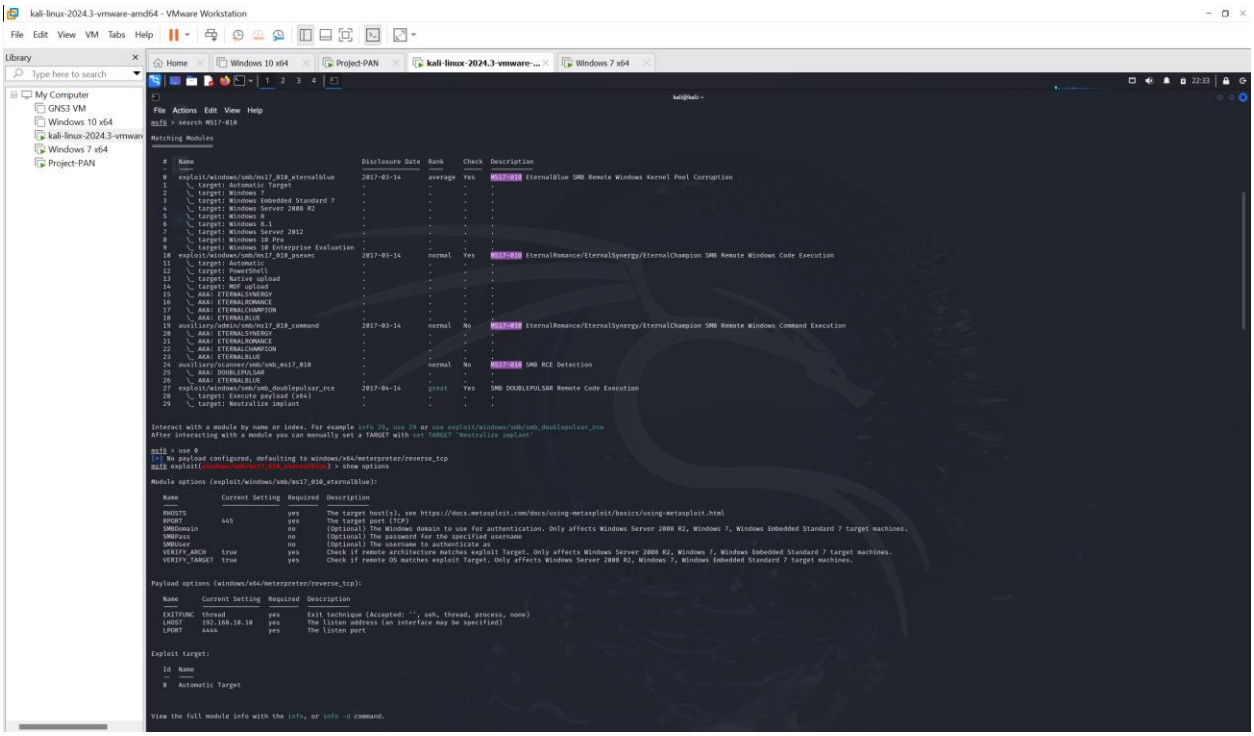


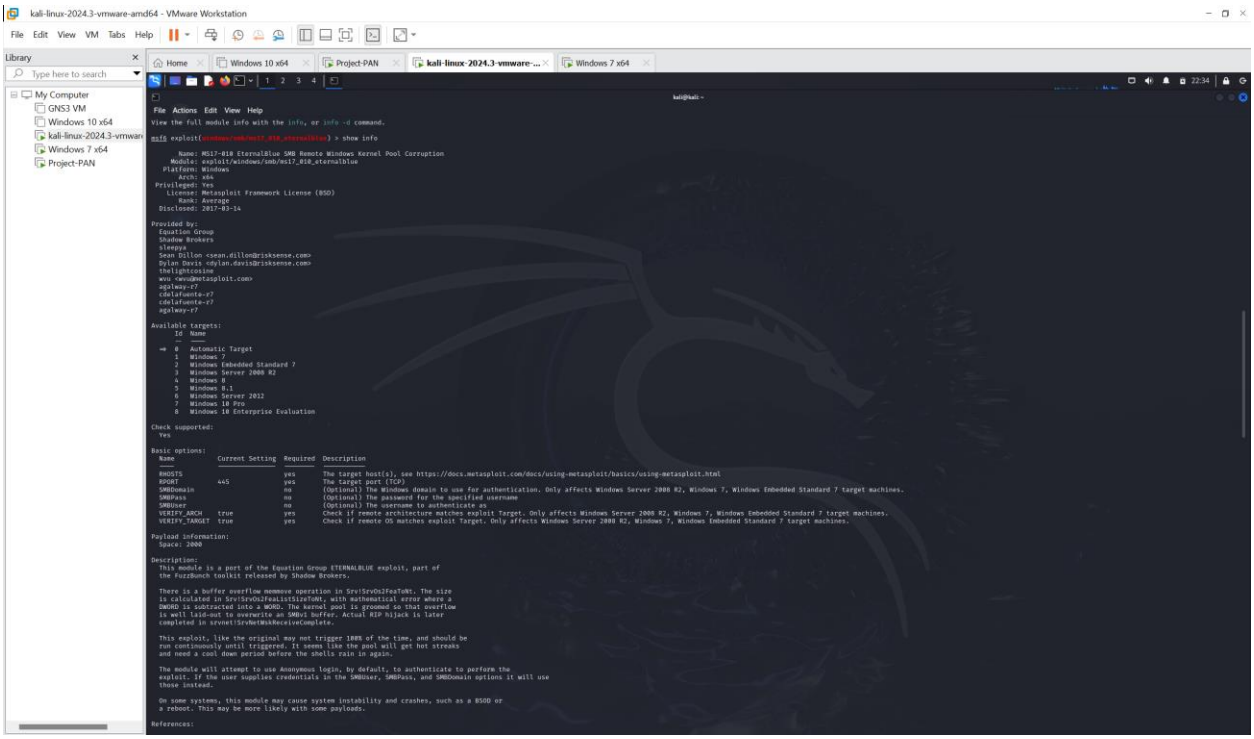
3. We can see the same in logs in Monitor Tab which shows the rule **"SMBPortallow"** to port **445**.



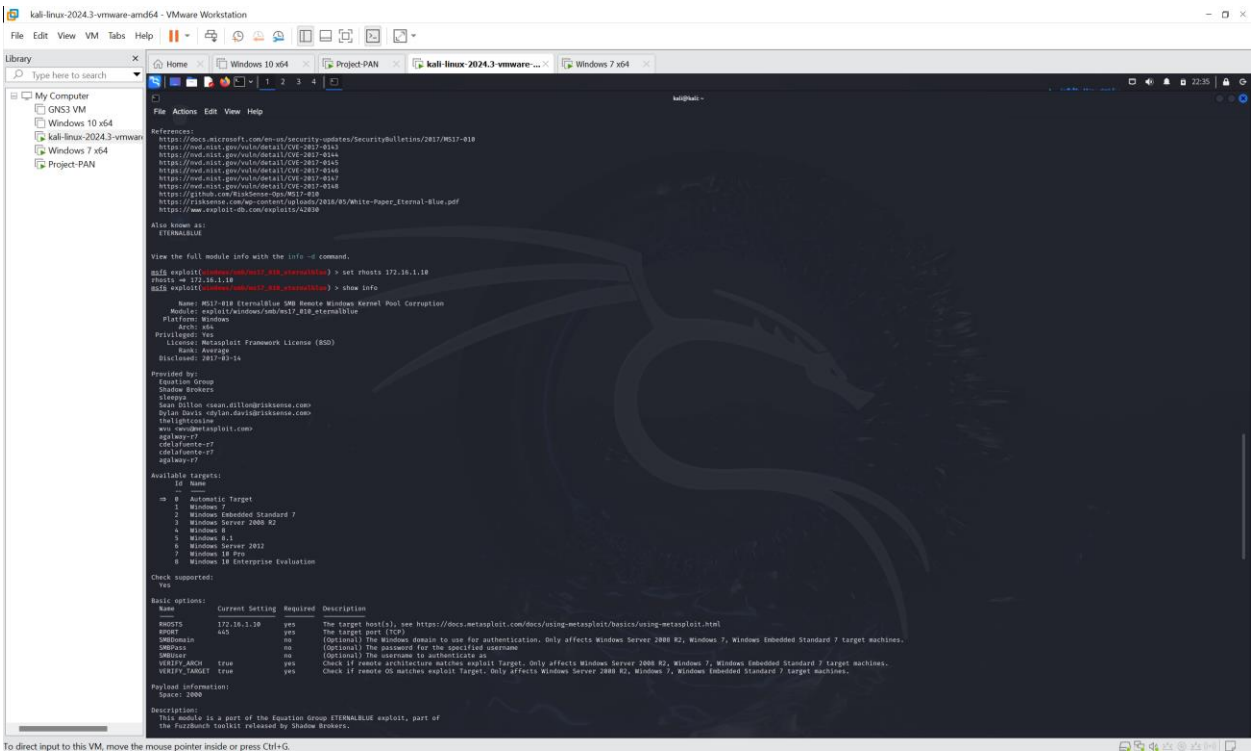
Q13. Use Metasploit on Kali Linux to exploit the MS17-010 vulnerability on DMZ-Host.

1. After configuring the rule, we exploited the **MS17-010** vulnerability on DMZ-Host as shown below screen

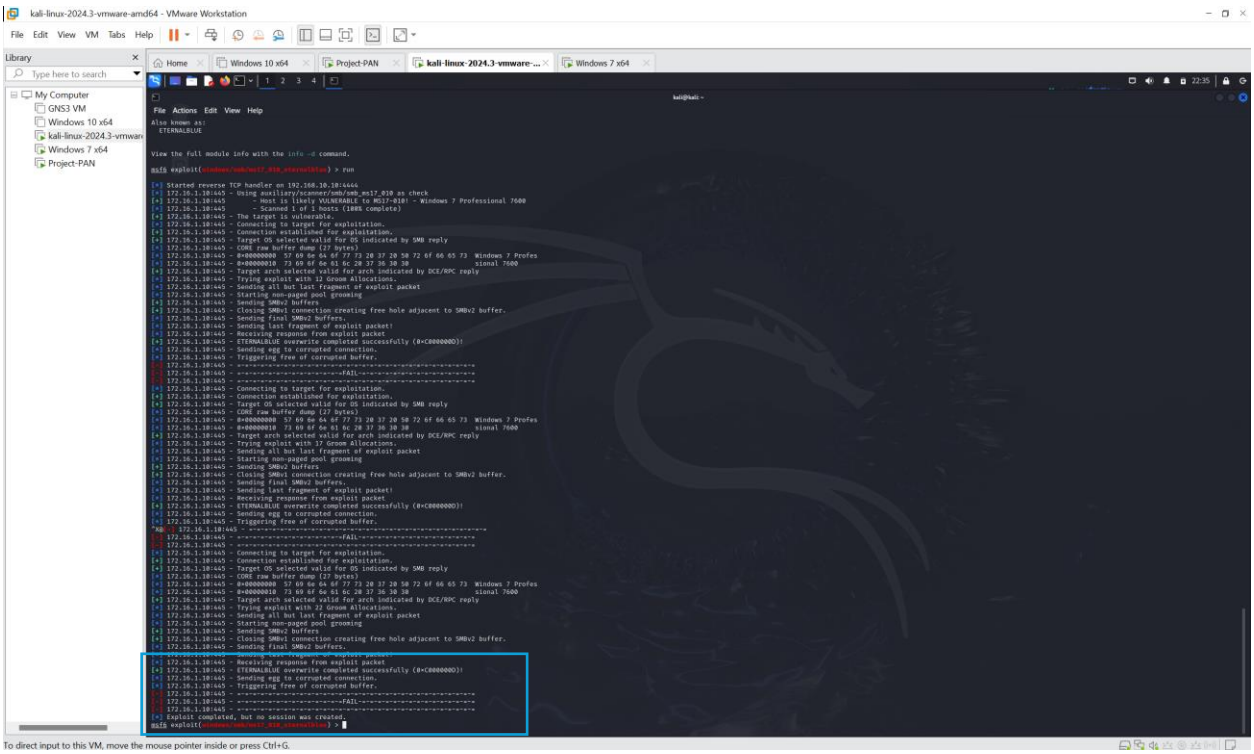




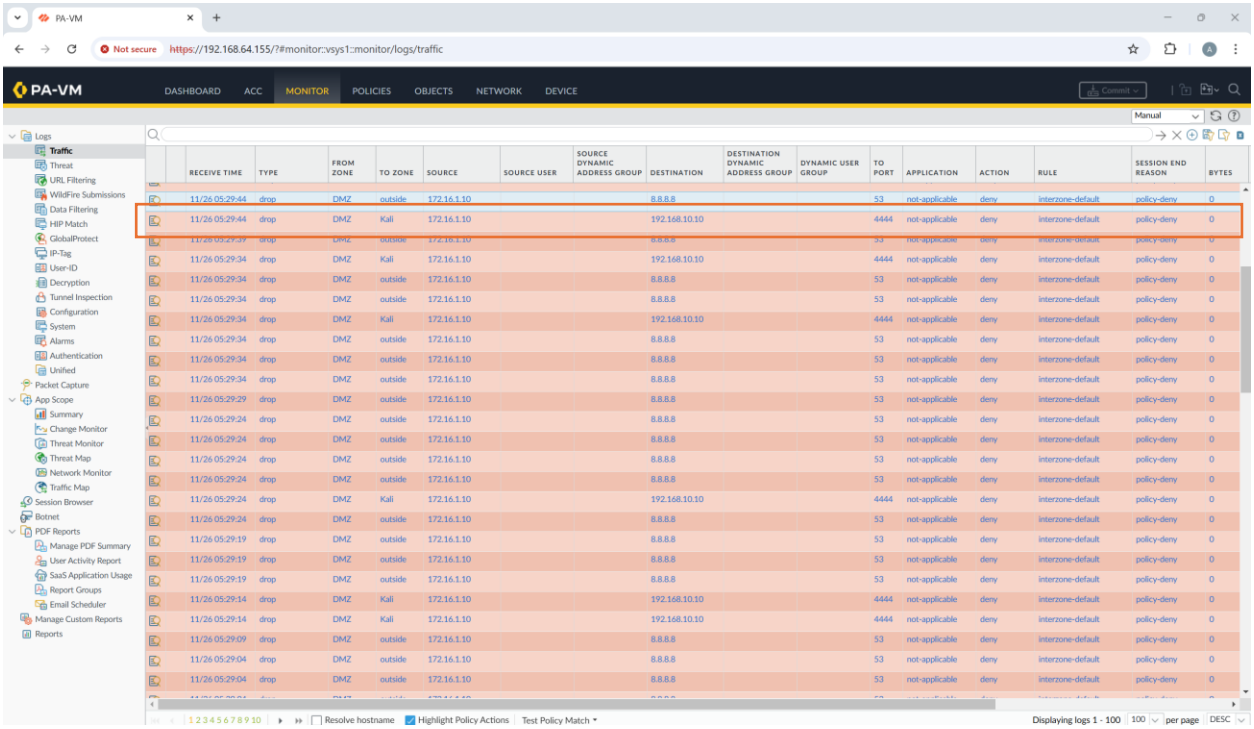
2. We configured the **RHOSTS** in show info to **172.16.1.10** and **RPORT** was set already as **445**.



3. We attempt to exploit the attack **MS17-010** vulnerability, and it got failed and gave a message **"exploit was completed, but no session was created"**. The attack was not successful because we don't have the reverse connection configured (i.e. the reverse connection from DMZ-host will be rejected by the firewall).

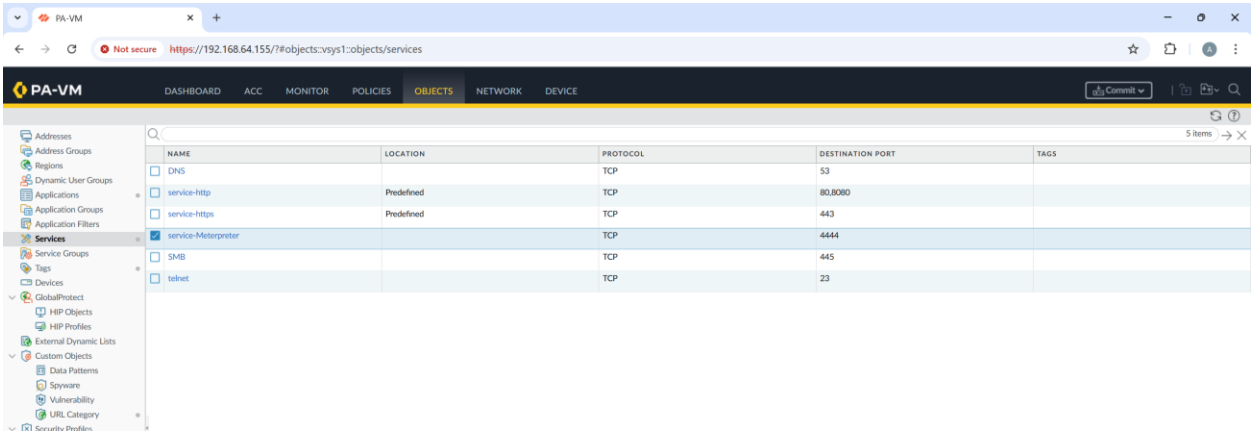


4. We can see the same in logs in Monitor tab where it gets rejected on the port 4444 from interzone-default policy.

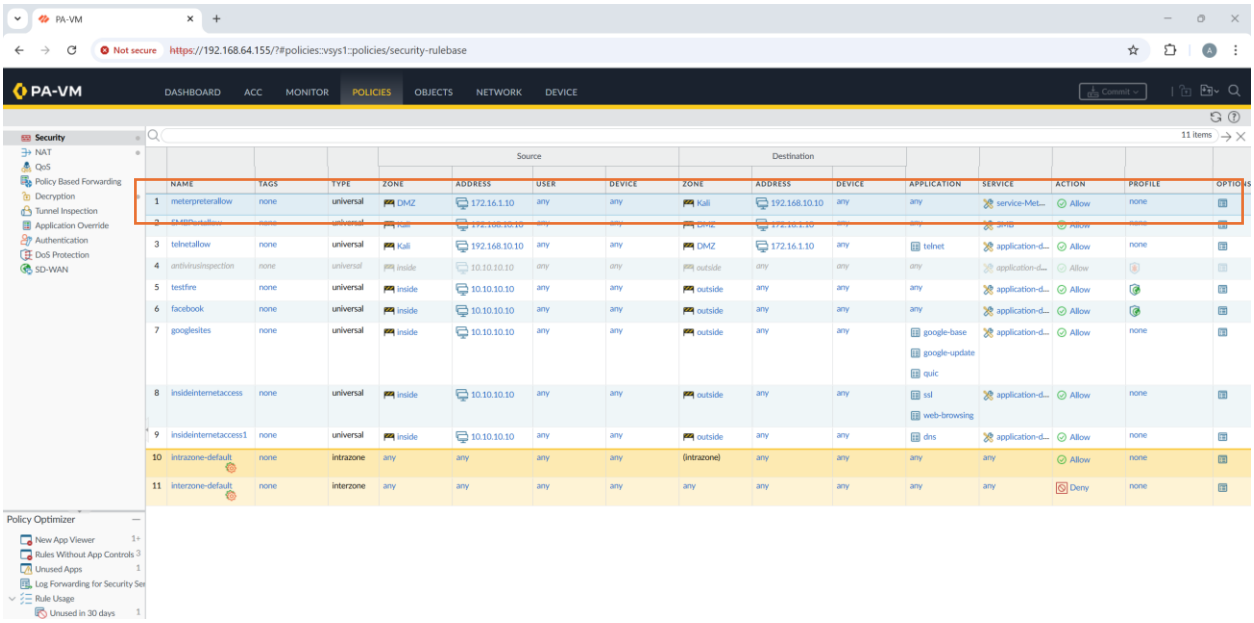


Q14. Assess the success of the attack and apply any required steps to achieve success.

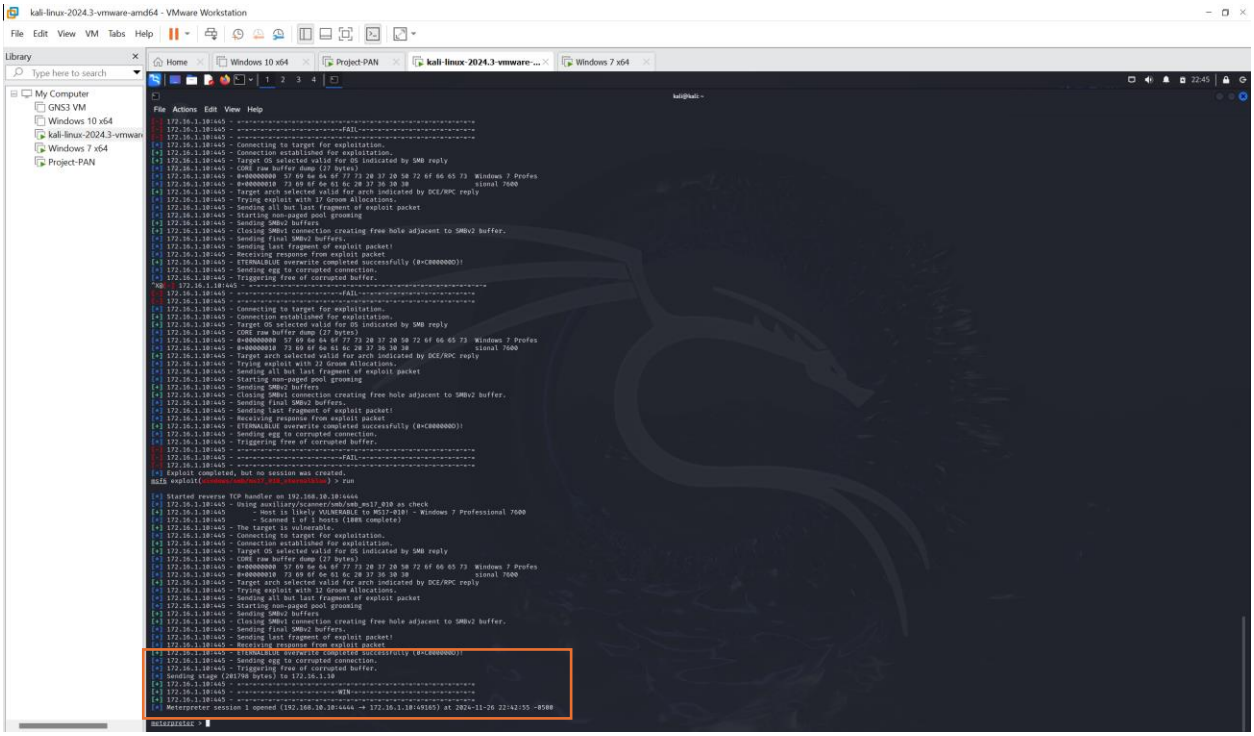
1. We need to configure the services under the Object tab with name "**service-Meterpreter"** and port **4444** as shown in below screenshot.



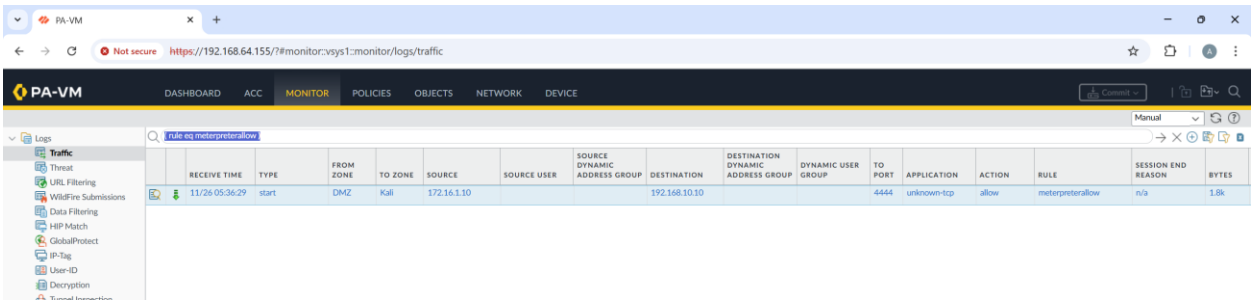
2. Then, we configured the **Service-Meterpreter** under the policies tab of PA-VM firewall from DMZ to Kali with service enable as shown below:



3. To test the above configuration, we exploited the vulnerability again DMZ-host from Kali-Linux machine and it got successfully as demonstrated in below illustration (***the win message is shown***).

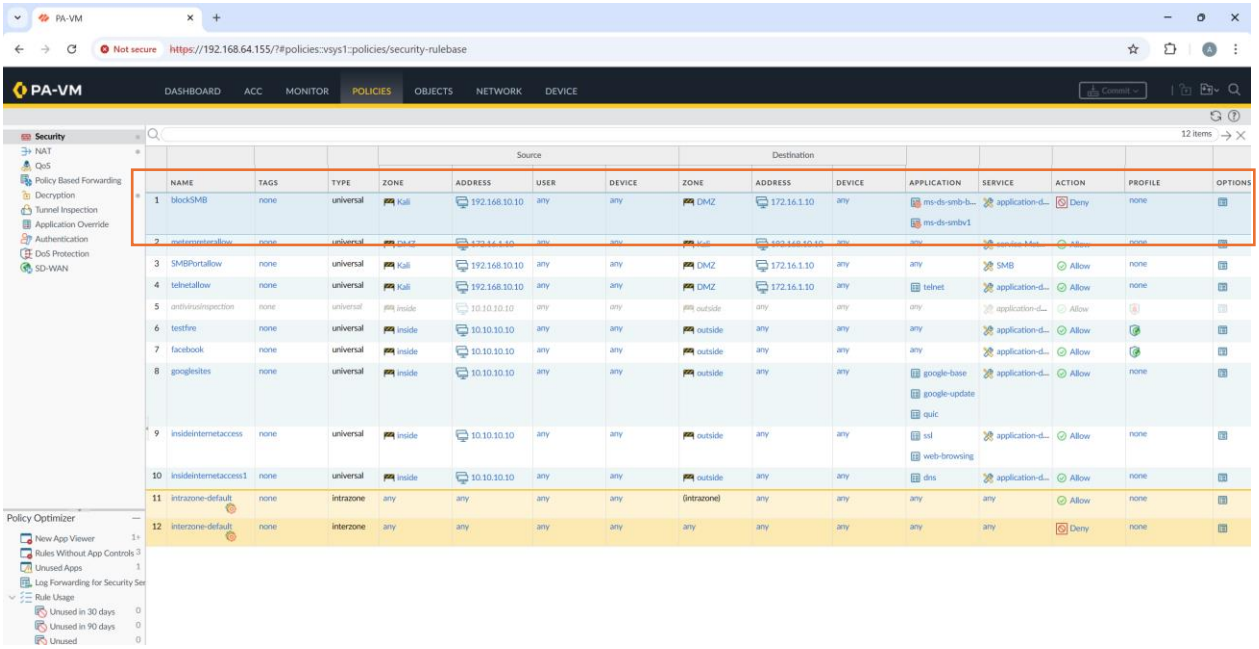


4. We can see the same in logs in Monitor tab in PA-VM interface that the rule was allowed which we configured for port **4444** to allow the successful attack of MS17-010 vulnerability.

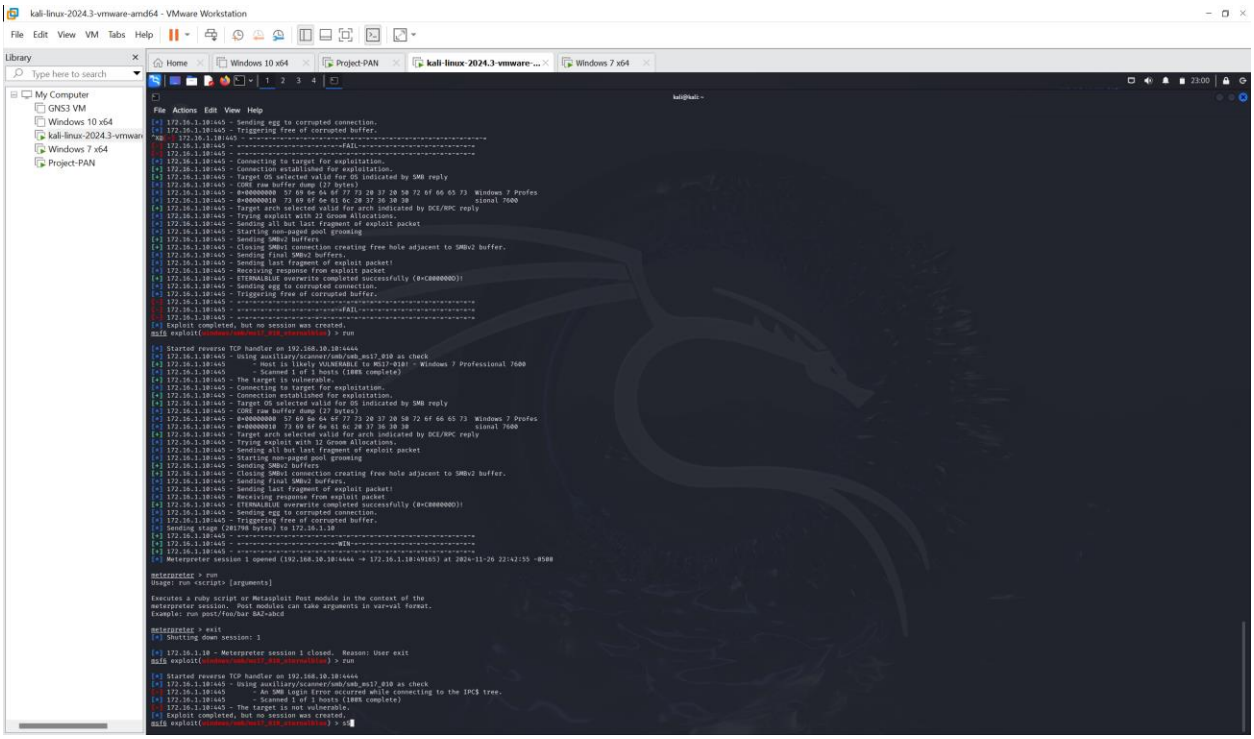


15. Block the application used in the attack and demonstrate that port 445 remains open, but the attack is prevented.

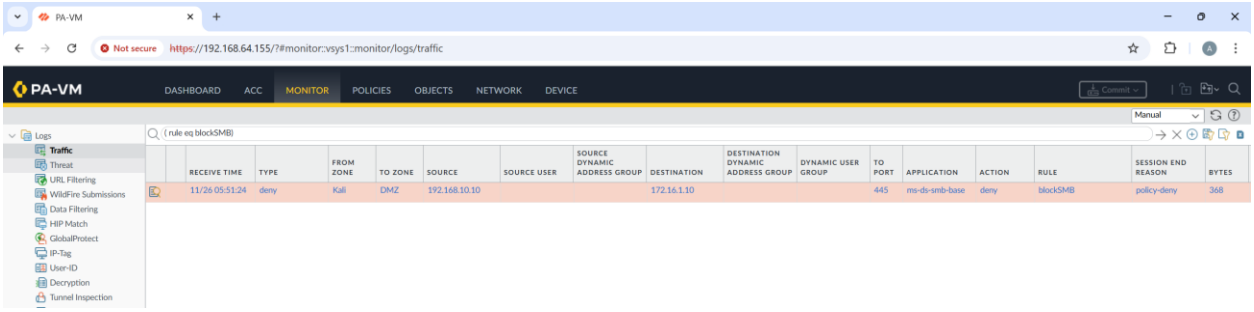
1. To test this scenario, we configured the rule with name: **"blockSMB"** from Kali 192.168.1.10 to DMZ 172.16.1.10 with the application SMB and smbv1 and the action: **Deny**.



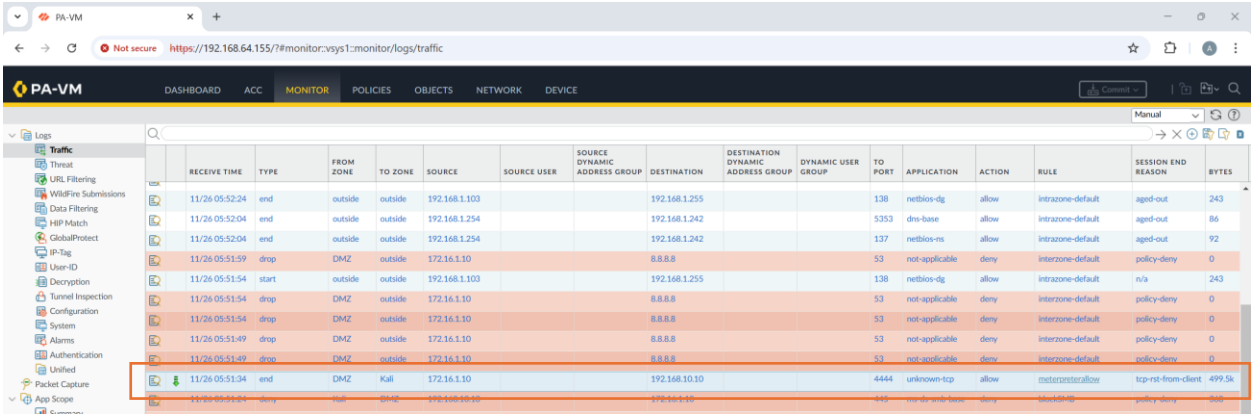
2. To test the above rule, exploited the **MS17-010** and the attack is blocked as shown in below screen.



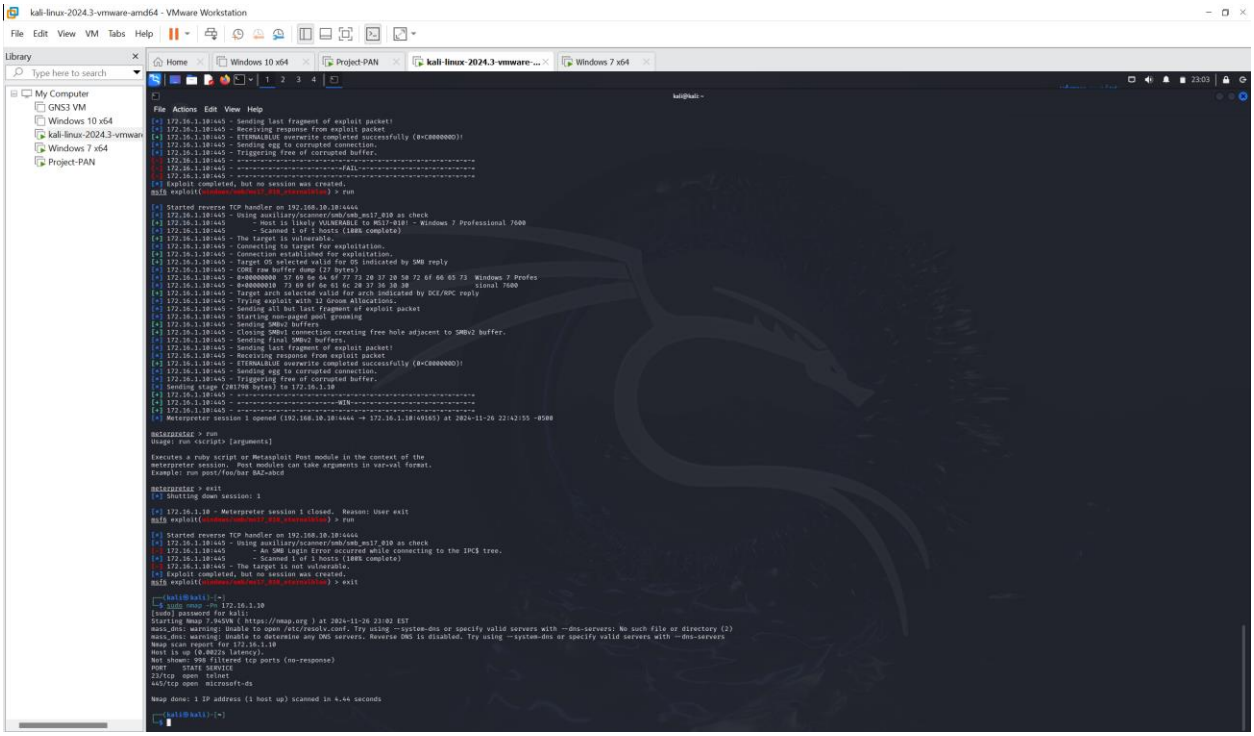
3. In Monitor tab, we can see the rule **"blockSMB"** has been denied to the port 445.



4. In below screen, the logs shows that the meterpreter was allowed in the port 4444 to execute the attack but it got blocked to the configured rule.

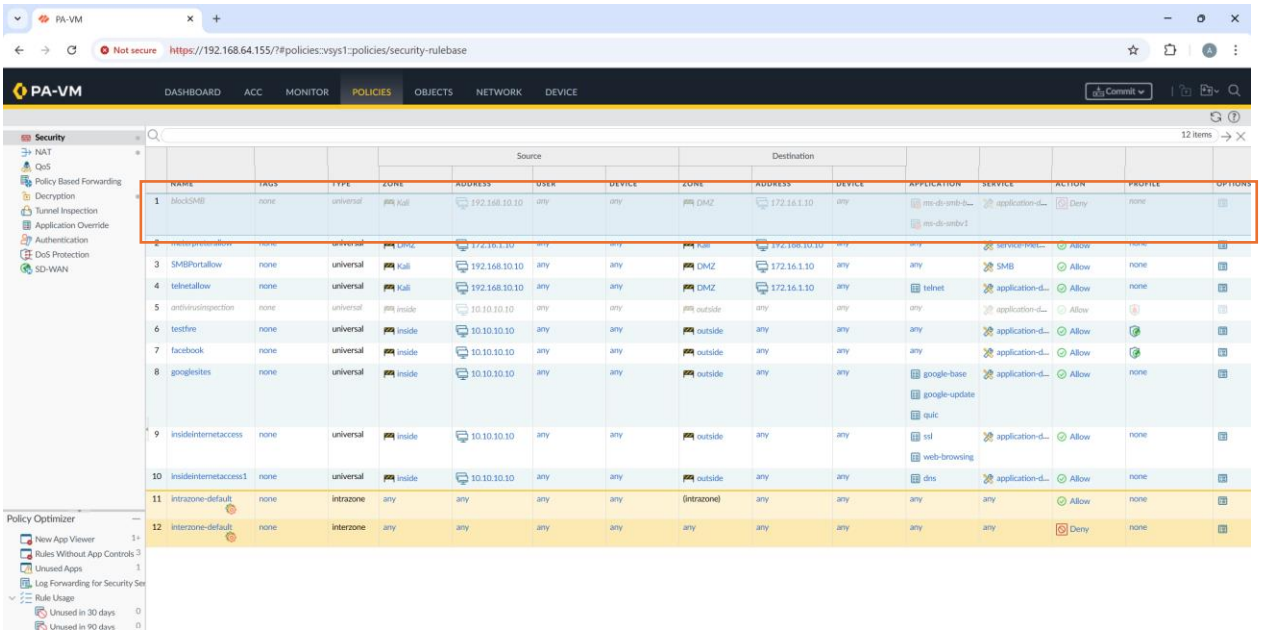


5. The below snip from terminal in Kali Linux machine shows that the port 445 is still open but the attack was prevented from Firewall rule which configured **“blockSMB”**.

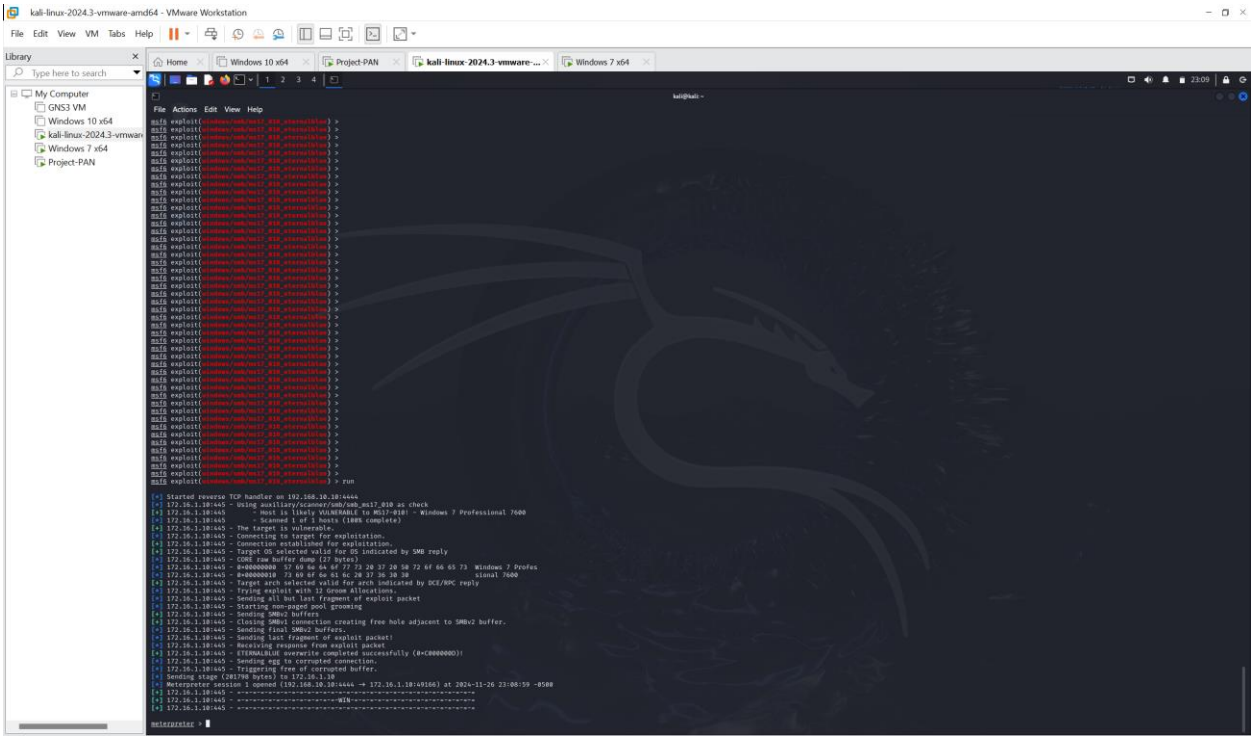


16. Undo Step 15.

1. In the below screen we can see that the rule name: **“blockSMB”** is disabled which was allowed in step15.



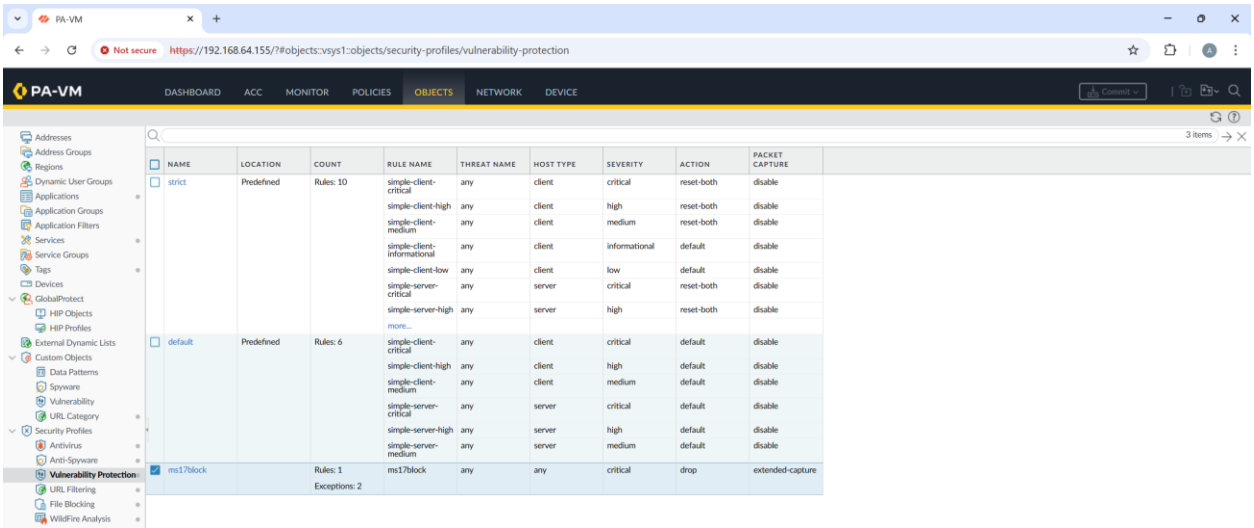
6. After disabling the policy, performed the attack again in Metasploit framework it was working, and session was created.



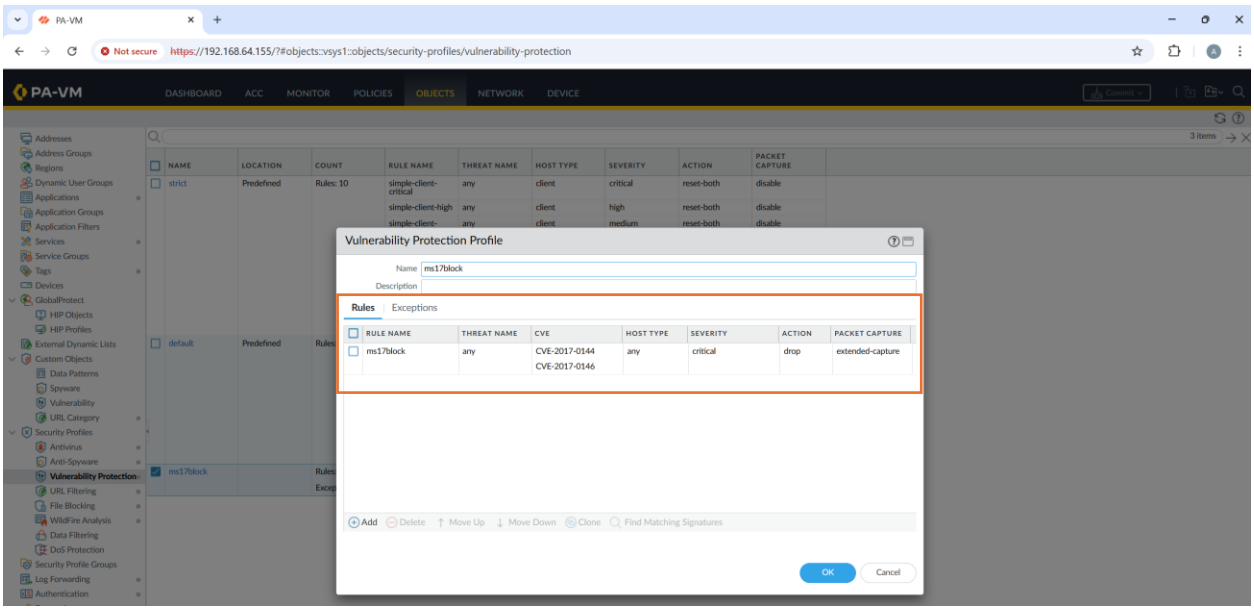
17. Use the PAN-OS IPS module to inspect attacker track and block the attack.

PAN-OS Intrusion Prevention System (IPS) to detect and block the MS17-010 exploit attempt while keeping port 445 open.

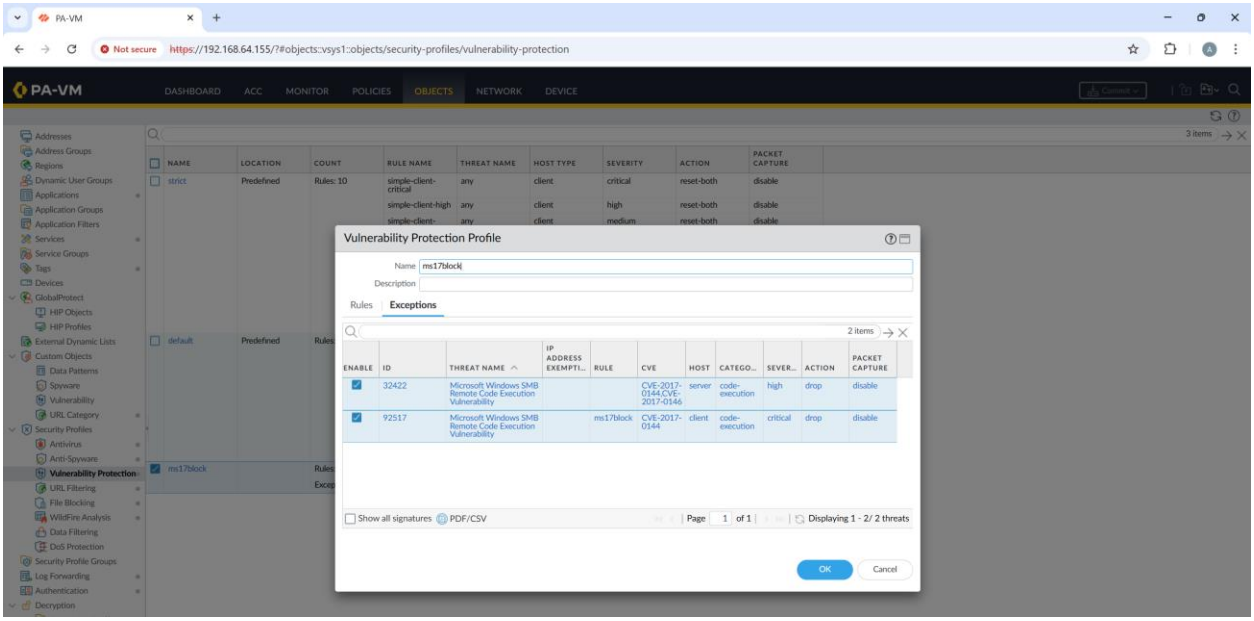
1. We configured the rule under Object→ Vulnerability Protection tab to block **MS17-010** exploit as shown in below screen.



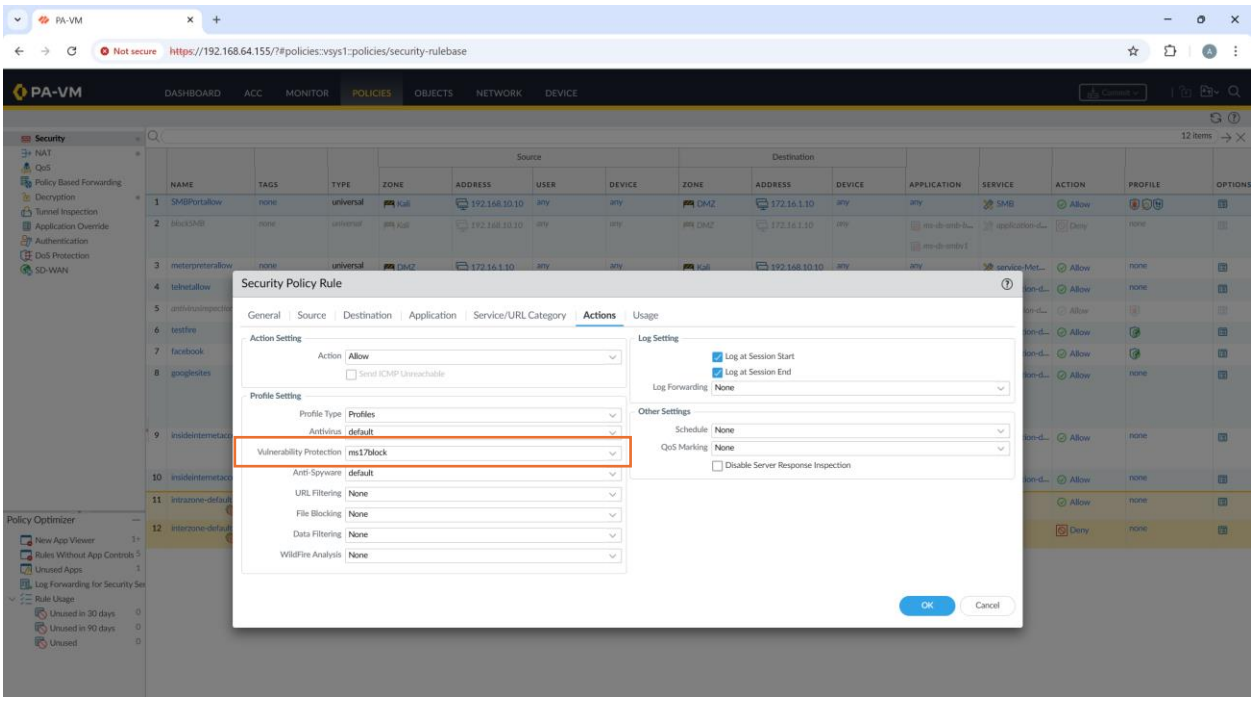
7. We first added the rule with the CV number which were used in this attack the **CVE-2017-0144** and the action “drop” with severity “critical” and the rule name: **“ms17block”**.



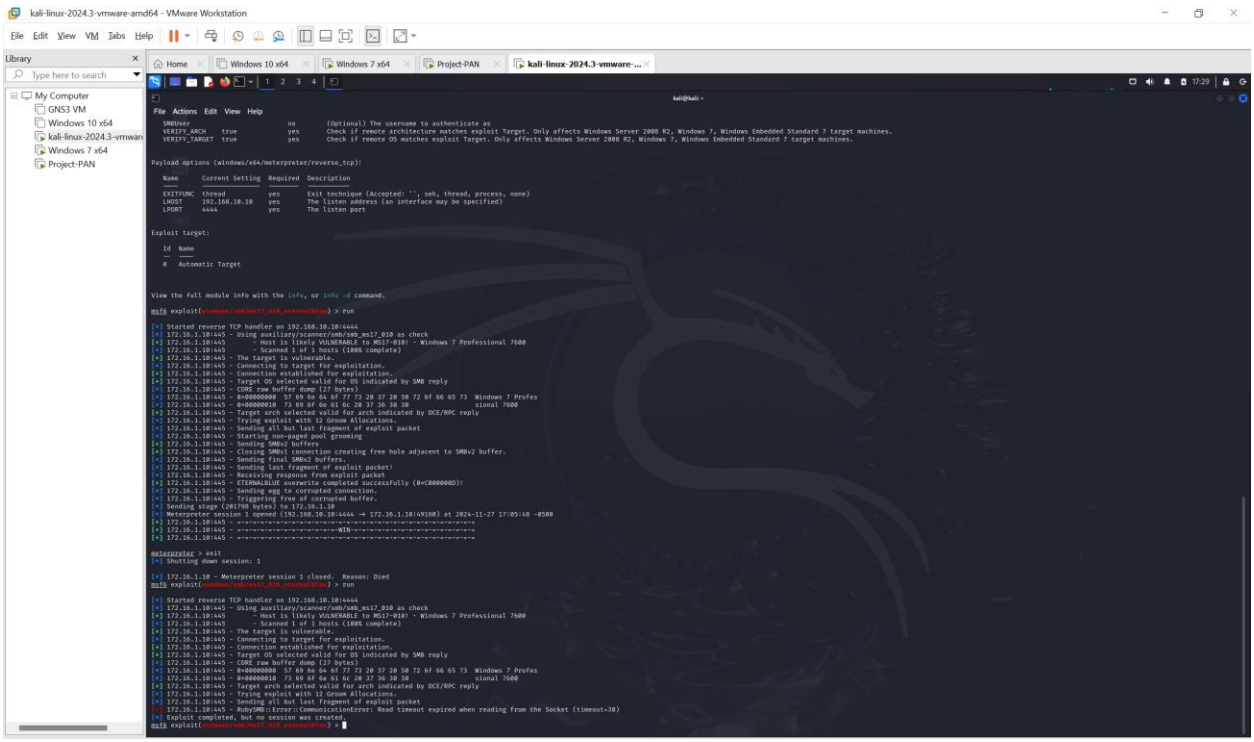
8. In the Vulnerability Protection rule add the exceptions related to code-execution and mark the action as "**drop**" as shown in below screen.



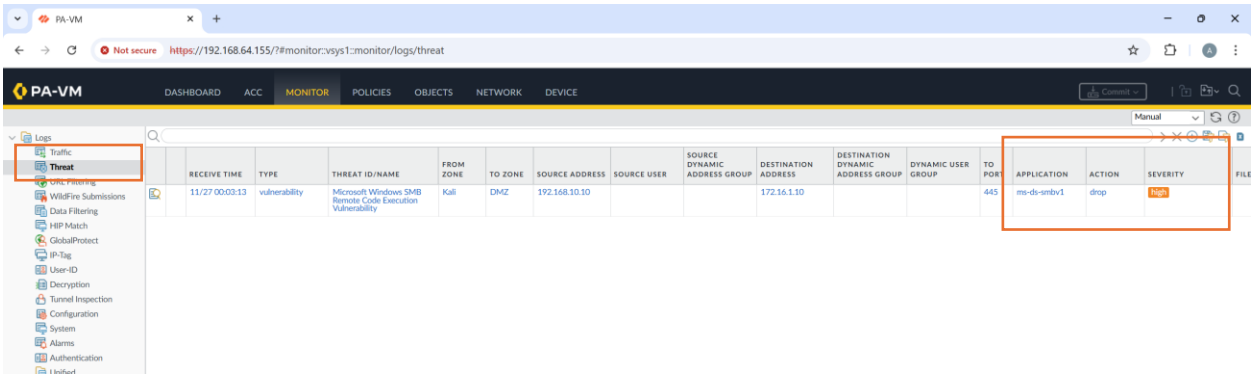
9. We configure the security policy rule and we enable vulenrability protection under profile settings with **"ms17block"** name.



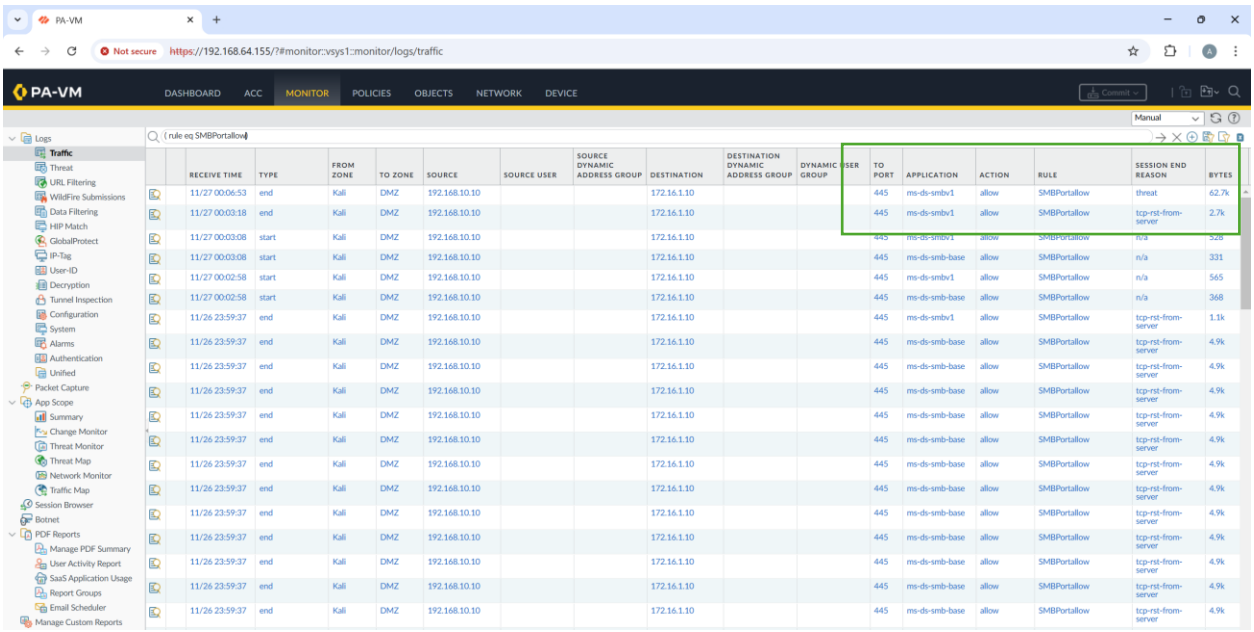
10. To verify the above rules created, we run the exploit again in **metasploit** framework, but no session was created and exploit got completed as shown in below screen.



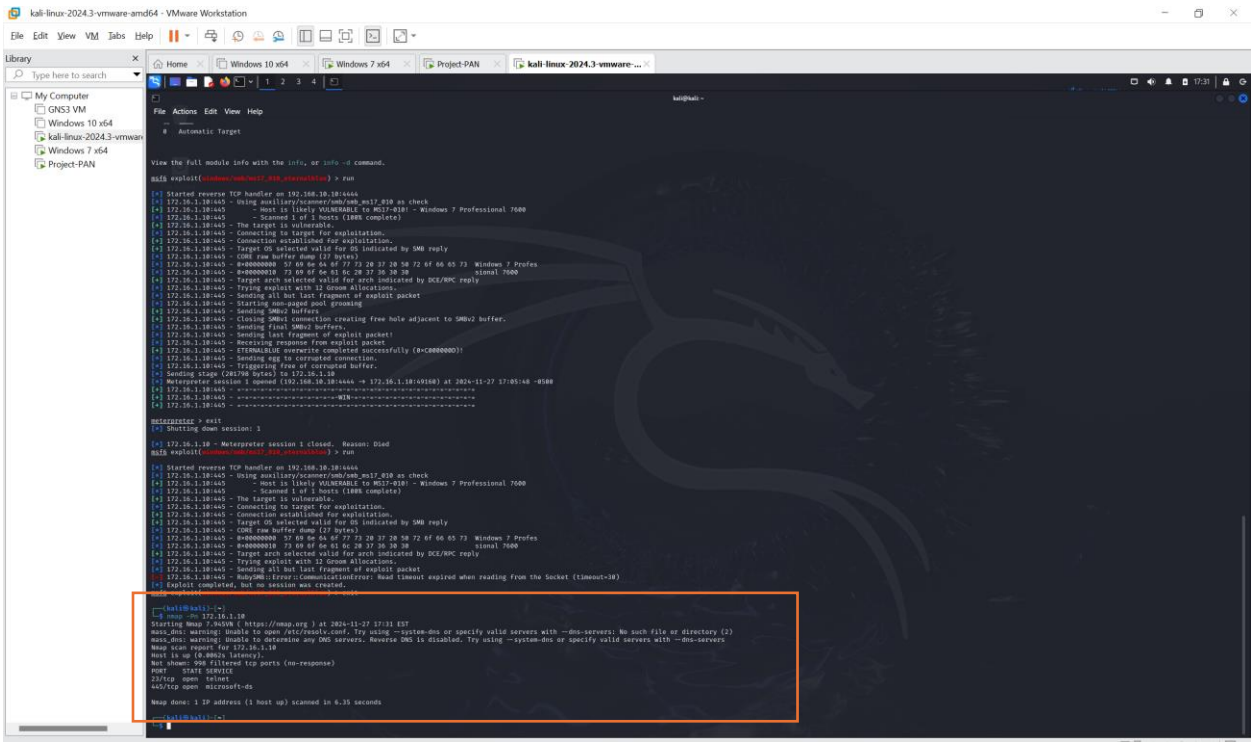
11. We can see the same in our logs like in **Monitor → Threat** from Kali to DMZ and type: **vulnerability** which got drop and severity is **high**.



12. We can see the same in monitor tab in the logs when we inspect the traffic that the policy we configured the **"SMBPortallow"** session was ended due to threat.



13. The below snip from terminal in Kali Linux machine shows that the port **445** is still **open** but the attack was **prevented** from **PAN-OS IPS module**.



Conclusion:

This project showcases a detailed understanding of network topology and Next-Generation Firewall (NGFW) configurations. It shows real world scenarios like internet access restriction, blocking the URLS which should not be allowed.