ECE 519C Final Task Report

By: Payton Murdoch, V00904677

Intro

After configuring all VMs to their corresponding VMnets and establishing the connections, the following figures show VMs pinging to their corresponding interface. Please note that some images will require zooming in to see the information.

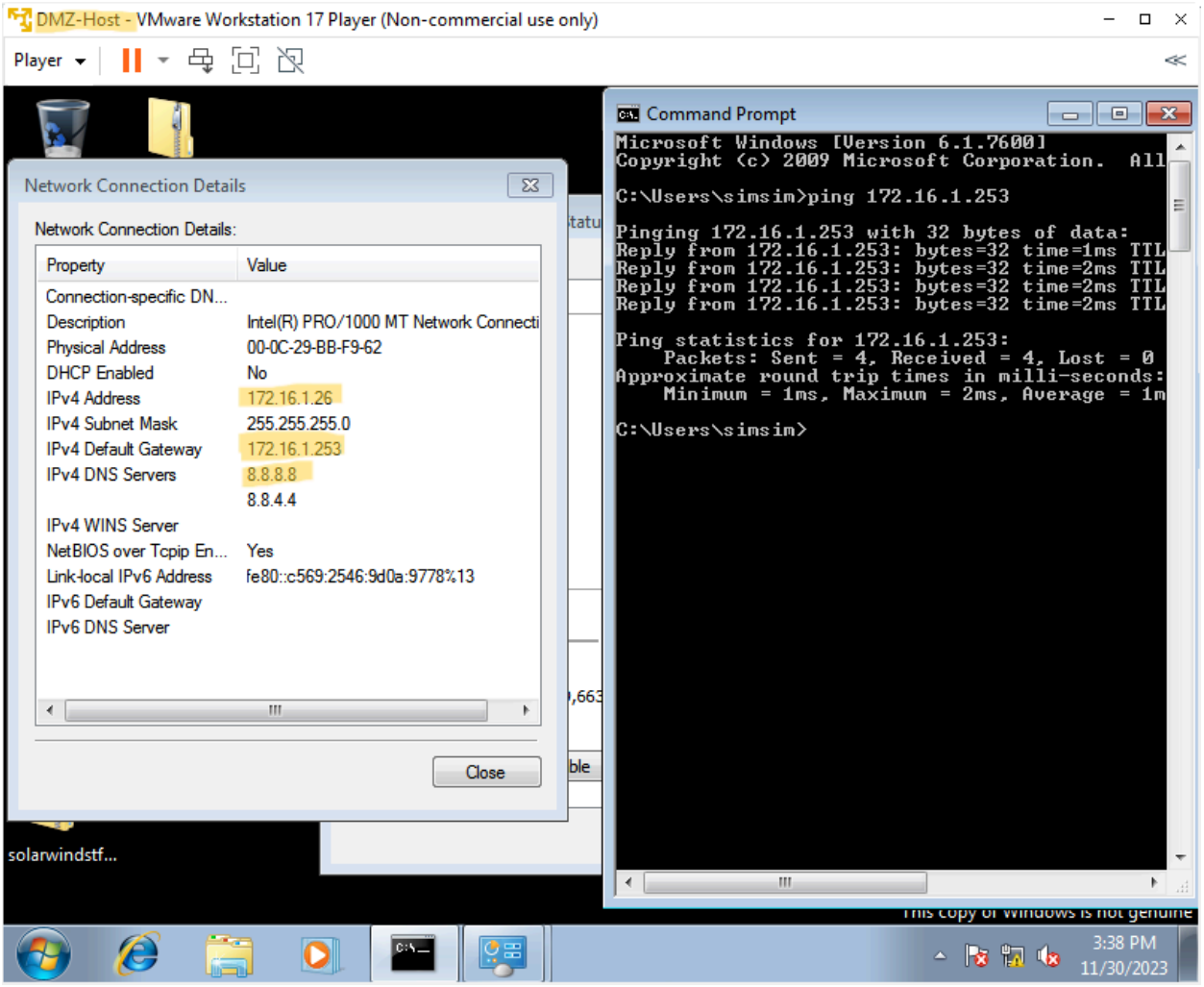

Figure 1. DMZ-Host IP and Ping to interface

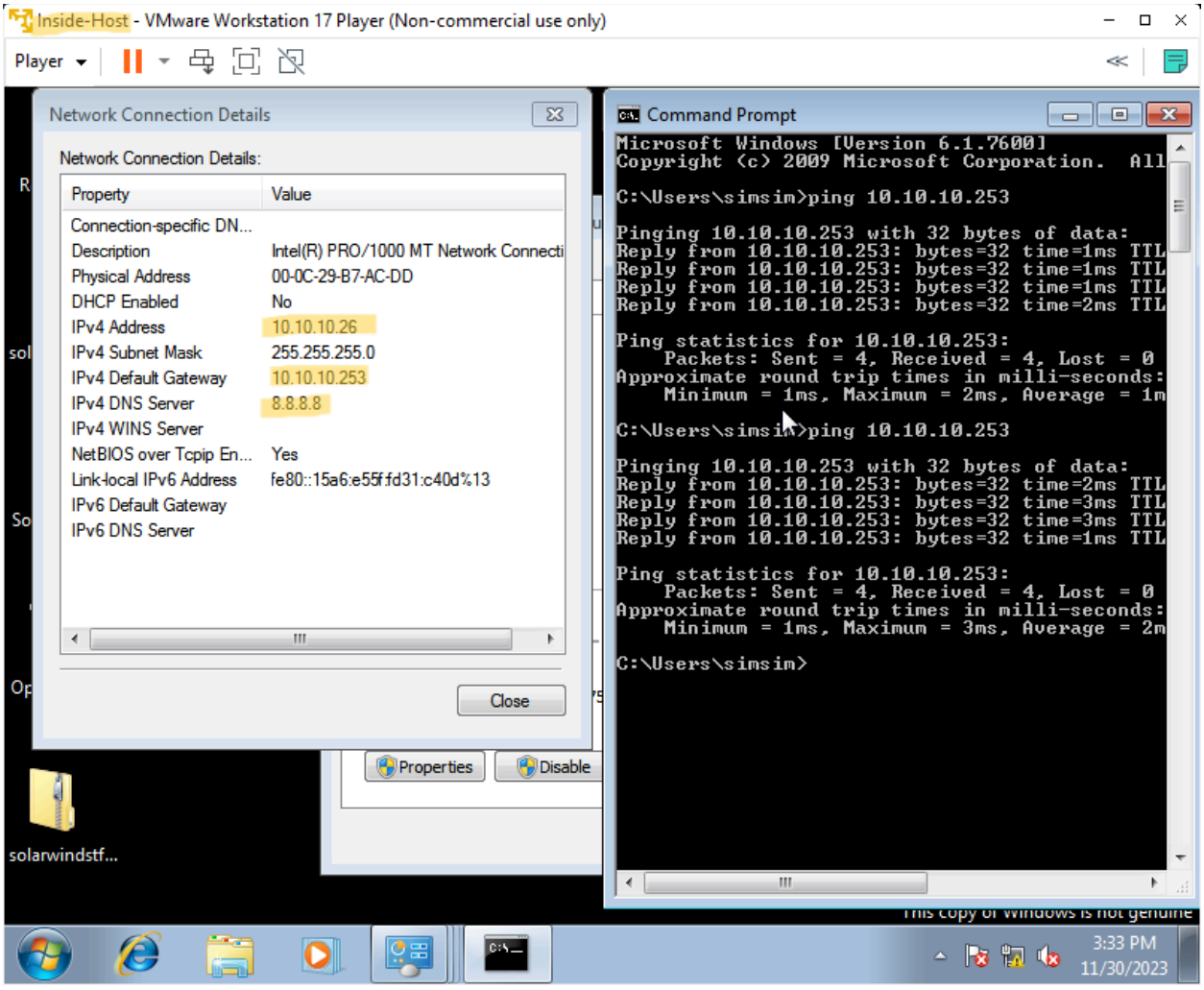

Figure 2. Inside-Host IP and Interface Ping

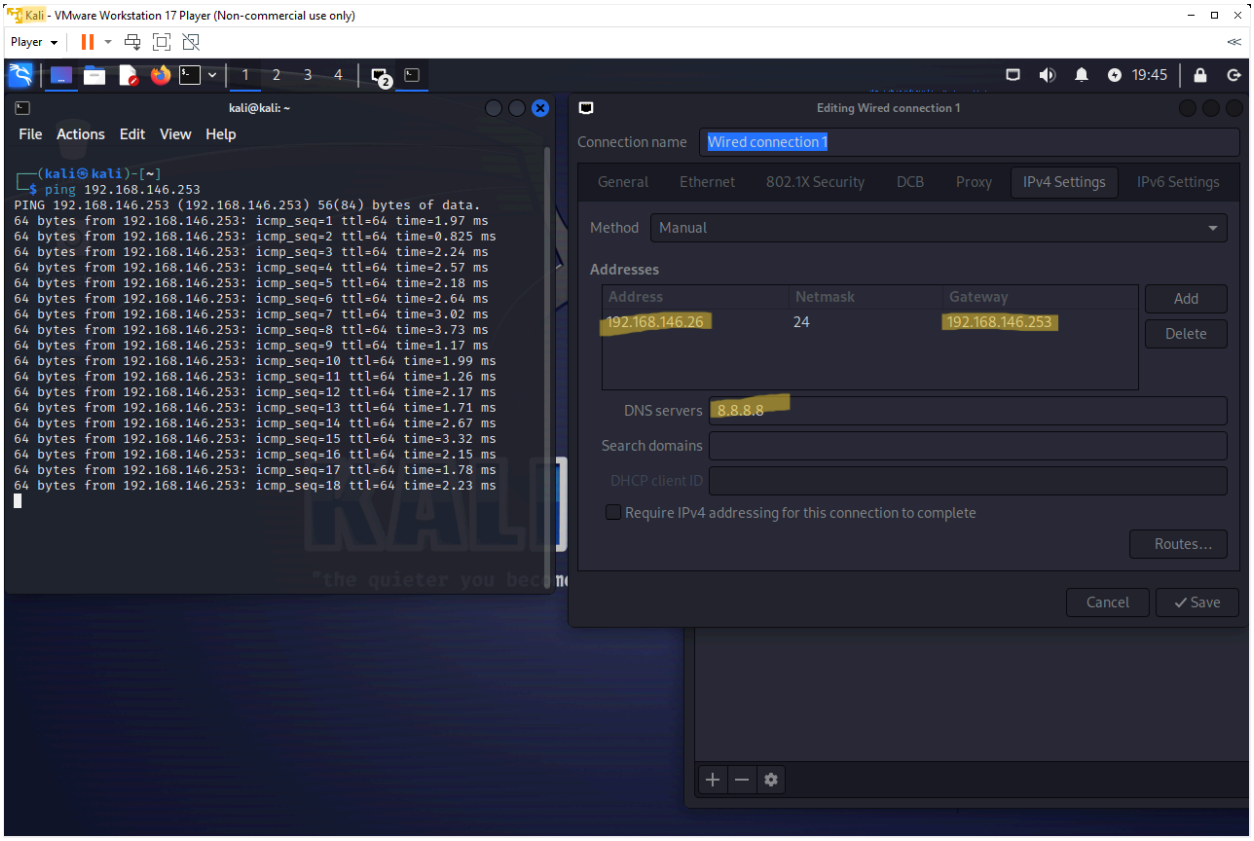

Figure 3. Kali IP and Interface Ping

1. Configure a PAT rule that allows the inside host and DMZ-Host to access the Internet by sharing the outside interface IP.

As we know in the Policies tab of the Palo Alto management interface, we can configure NAT policies to give the Inside host, and DMZ host a public IP by assigning them dynamic IPs through the ethernet ¼ line. This configuration of policies in Figure 4 does not allow internet access to the hosts; thus, they do not need logs. This is because the change to public IPs is not shown, and thus, the system change cannot be reflected in the monitor tab.

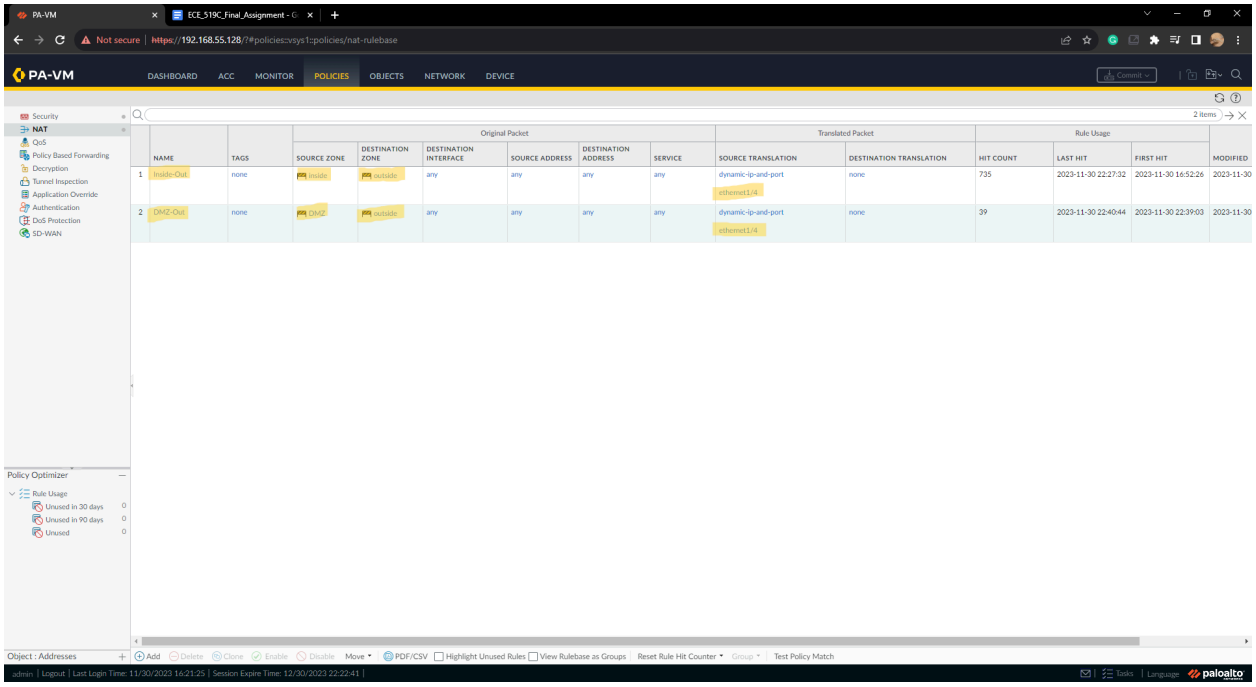

Figure 4: Configuration of PAT with Inside-Host and DMZ-Host to the outside network using Dynamic IPs through ethernet ¼.

2. Demonstrate that the Inside-Host, Kali-Linux, and DMZ-Host cannot access each other or the Internet (Zeo-Trust concept).

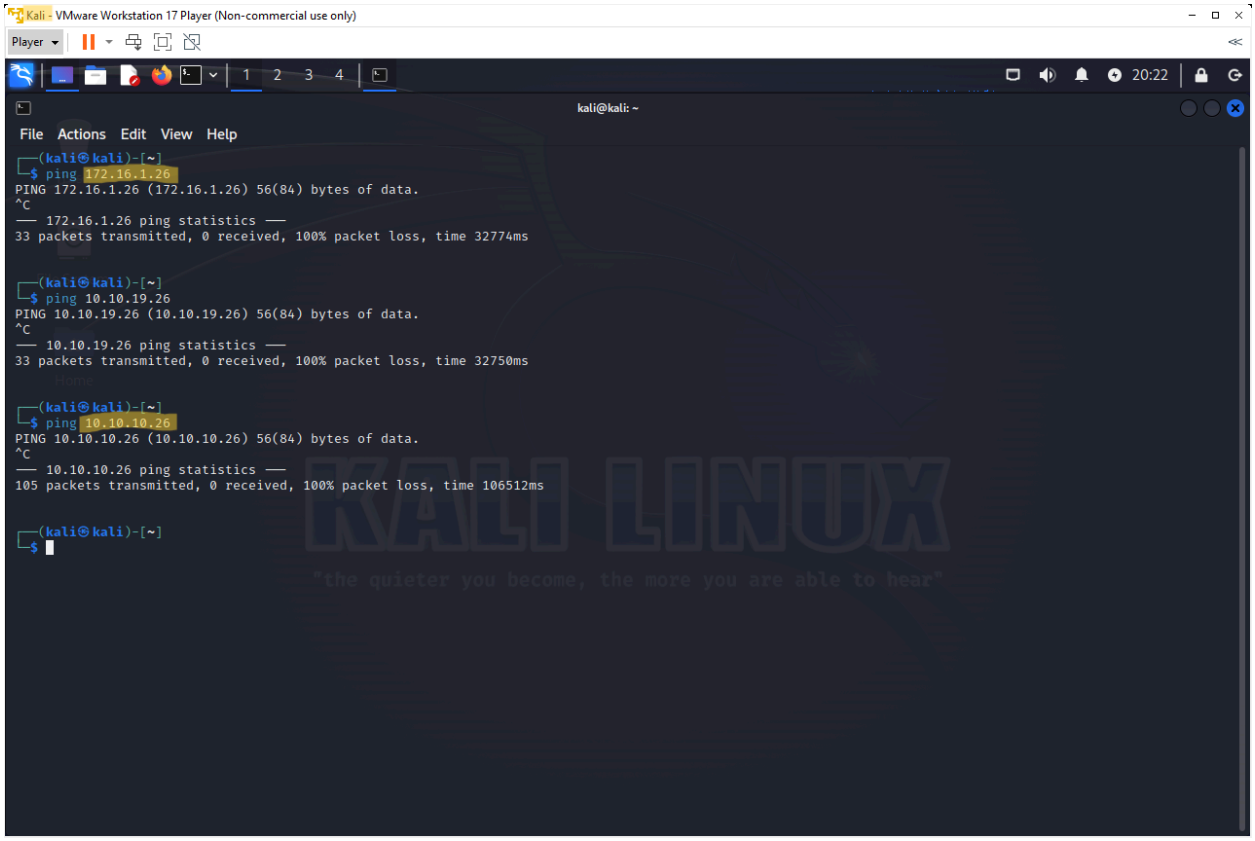

Figure 5: Kali Linux Ping Denied to Inside-Host and DMZ-Host

		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	ACTION	RULE	SESSION END REASON	BYTES	HTTP/2 CONNECTION SESSION ID	SDWAN SITE NAME	APP F COUN
		11/30 17:20:33	drop	DMZ	outside	172.16.1.26			8.8.8.8			53	not-applicable	deny	interzone-default	policy-deny	0	0		0
		11/30 17:20:33	drop	Kali	inside	192.168.146.26			10.10.10.26			0	ping	deny	interzone-default	policy-deny	0	0		0
		11/30 17:20:33	drop	DMZ	outside	172.16.1.26			8.8.8.8			53	not-applicable	deny	interzone-default	policy-deny	0	0		0
		11/30 17:20:28	drop	DMZ	outside	172.16.1.26			8.8.8.8			53	not-applicable	deny	interzone-default	policy-deny	0	0		0

Figure 6: Kali Linux Log For Denied Ping to Inside-Host

		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	ACTION	RULE	SESSION END REASON	BYTES	HTTP/2 CONNECTION SESSION ID	SDWAN SITE NAME	APP FLA COUNT
		11/30 17:23:03	drop	Kali	DMZ	192.168.146.26			172.16.1.26			0	ping	deny	interzone-default	policy-deny	0	0		0
		11/30 17:22:58	drop	DMZ	outside	172.16.1.26			8.8.8.8			53	not-applicable	deny	interzone-default	policy-deny	0	0		0
		11/30 17:22:53	drop	inside	outside	10.10.10.26			8.8.8.8			53	not-applicable	deny	interzone-default	policy-deny	0	0		0

Figure 7: Kali Linux Log For Denied Ping to DMZ-Host

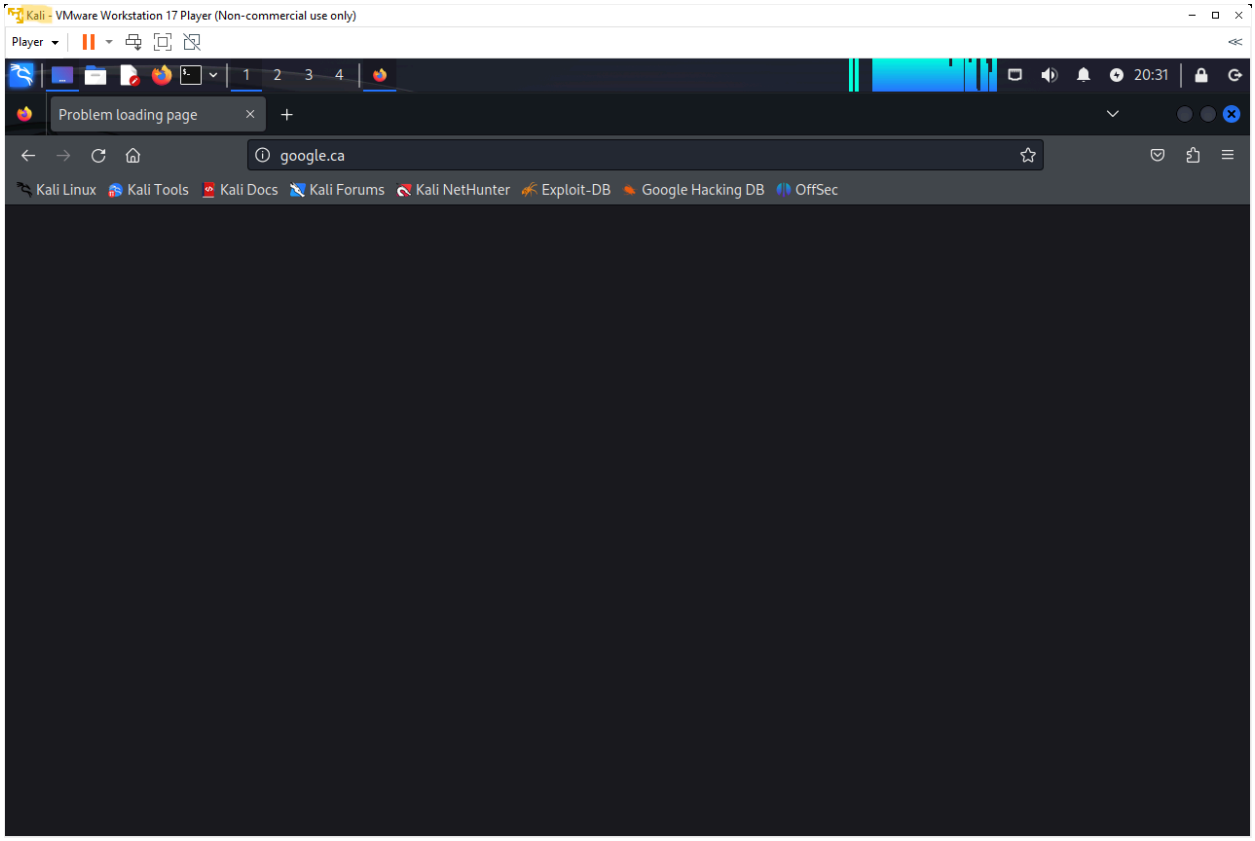

Figure 8: Kali Linux is unable to load Google.

		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	ACTION	RULE	SESSION END REASON	BYTES	HTTP/2 CONNECTION SESSION ID	SDWAN SITE NAME	APP FLA COUNT
		11/30 17:27:24	drop	Kali	outside	192.168.146.26			8.8.8.8			53	not-applicable	deny	interzone-default	policy-deny	0	0		0
		11/30 17:27:24	drop	Kali	outside	192.168.146.26			8.8.8.8			53	not-applicable	deny	interzone-default	policy-deny	0	0		0
		11/30 17:27:24	drop	Kali	outside	192.168.146.26			8.8.8.8			53	not-applicable	deny	interzone-default	policy-deny	0	0		0
		11/30 17:27:24	drop	Kali	outside	192.168.146.26			8.8.8.8			53	not-applicable	deny	interzone-default	policy-deny	0	0		0
		11/30 17:27:24	drop	Kali	outside	192.168.146.26			8.8.8.8			53	not-applicable	deny	interzone-default	policy-deny	0	0		0
		11/30 17:27:24	drop	Kali	outside	192.168.146.26			8.8.8.8			53	not-applicable	deny	interzone-default	policy-deny	0	0		0
		11/30 17:27:24	drop	Kali	outside	192.168.146.26			8.8.8.8			53	not-applicable	deny	interzone-default	policy-deny	0	0		0
		11/30 17:27:24	drop	Kali	outside	192.168.146.26			8.8.8.8			53	not-applicable	deny	interzone-default	policy-deny	0	0		0
		11/30 17:27:24	drop	Kali	outside	192.168.146.26			8.8.8.8			53	not-applicable	deny	interzone-default	policy-deny	0	0		0
		11/30 17:27:24	drop	Kali	outside	192.168.146.26			8.8.8.8			53	not-applicable	deny	interzone-default	policy-deny	0	0		0
		11/30 17:27:24	drop	Kali	outside	192.168.146.26			8.8.8.8			53	not-applicable	deny	interzone-default	policy-deny	0	0		0
		11/30 17:27:19	drop	Kali	outside	192.168.146.26			8.8.8.8			53	not-applicable	deny	interzone-default	policy-deny	0	0		0

Figure 9: Kali Linux Log for denied HTTP access to Google.

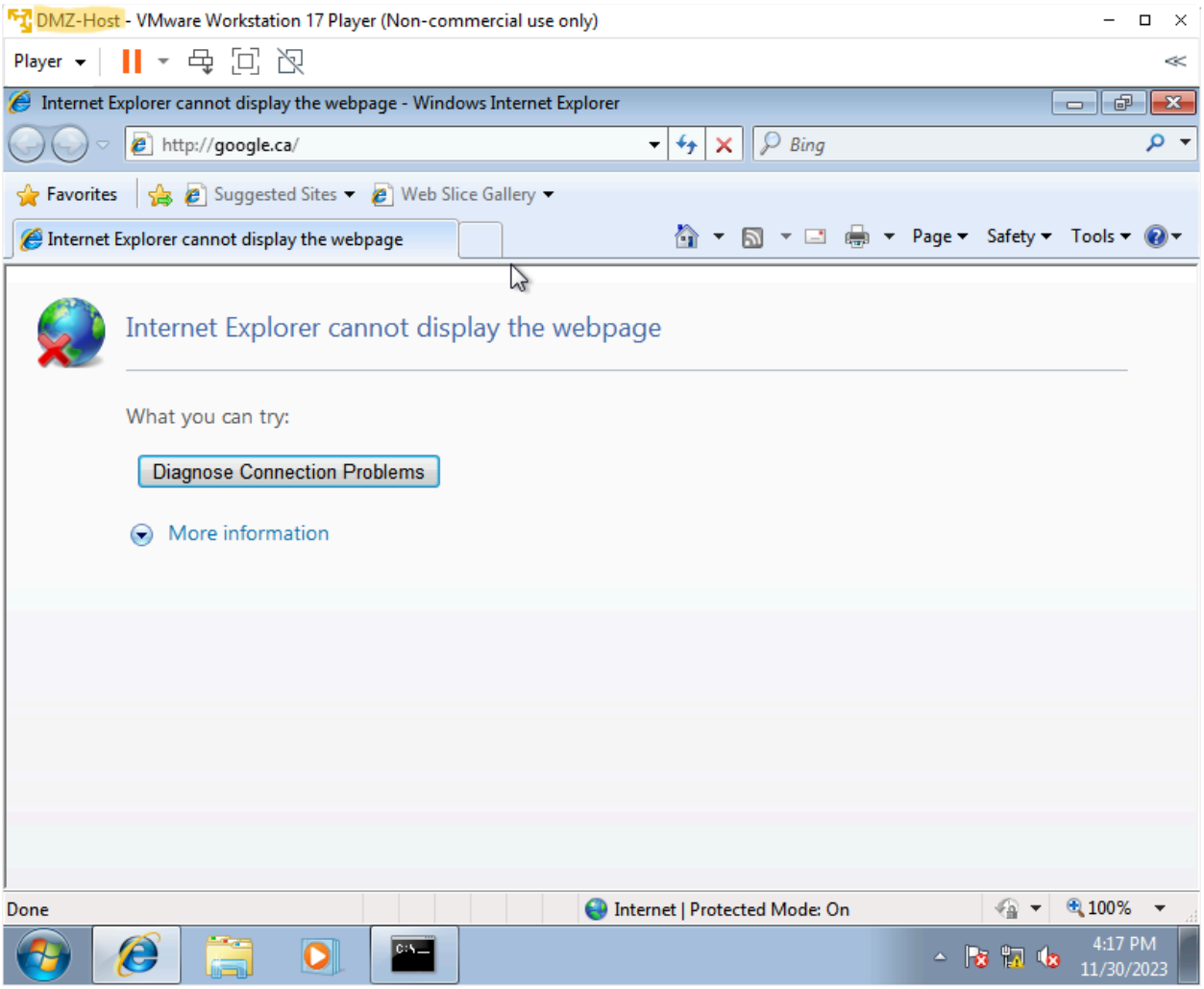

Figure 10: DMZ-Host Denied access to google.ca.

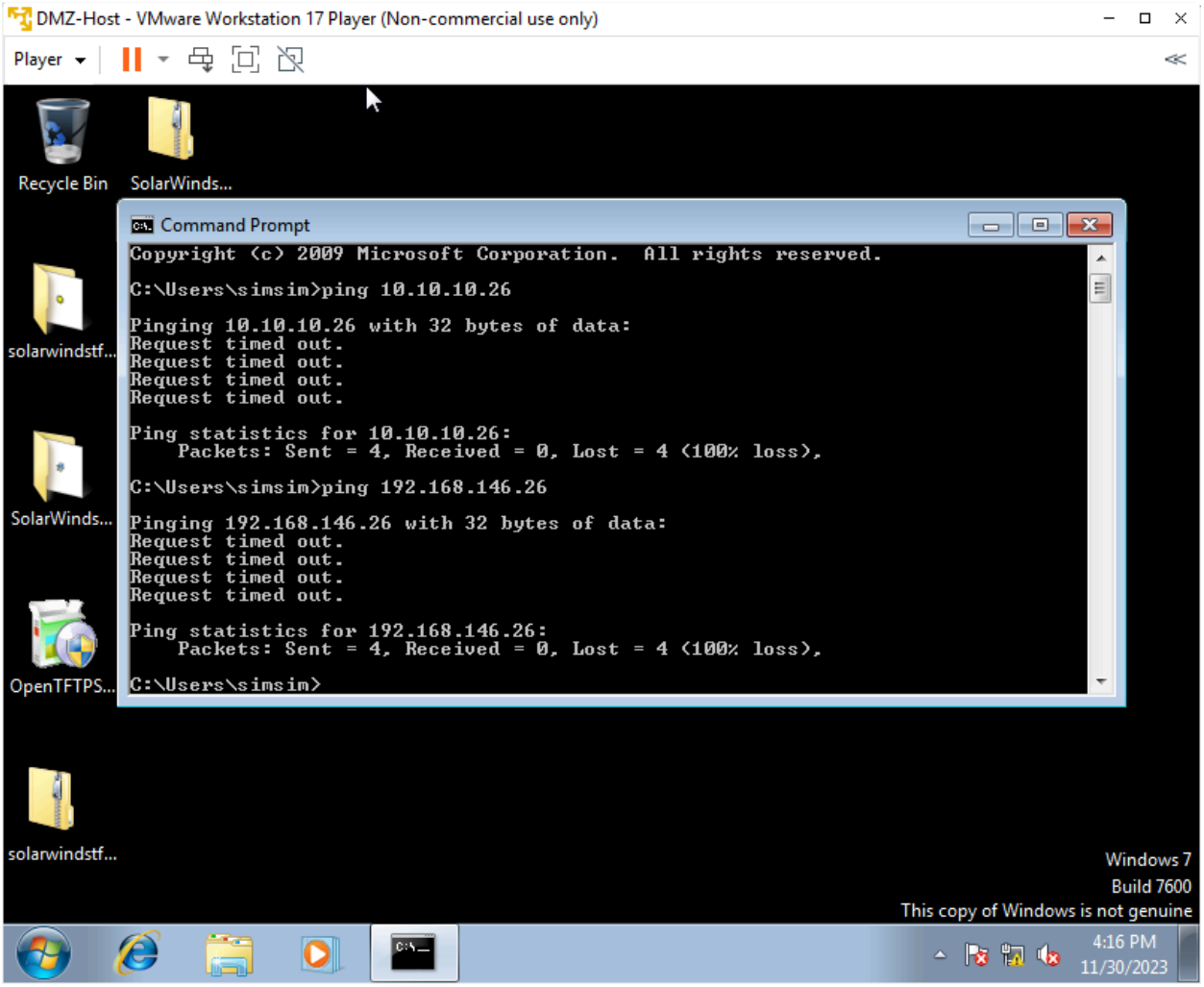

Figure 11: DMZ-Host denied pings to Inside-Host and Kali Linux.

		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	ACTION	RULE	SESSION END REASON	BYTES	HTTP/2 CONNECTION SESSION ID	SDWAN SITE NAME	APP FLA COUNT
		11/30 17:10:33	drop	DMZ	Inside	172.16.1.26			10.10.10.26			0	ping	deny	interzone-default	policy-deny	0	0		0
		11/30 17:10:33	drop	DMZ	outside	172.16.1.26			8.8.8.8			53	not-applicable	deny	interzone-default	policy-deny	0	0		0
		11/30 17:10:33	drop	DMZ	outside	172.16.1.26			8.8.8.8			53	not-applicable	deny	interzone-default	policy-deny	0	0		0
		11/30 17:10:33	drop	DMZ	outside	172.16.1.26			8.8.8.8			53	not-applicable	deny	interzone-default	policy-deny	0	0		0
		11/30 17:10:28	drop	DMZ	outside	172.16.1.26			8.8.8.8			53	not-applicable	deny	interzone-default	policy-deny	0	0		0
		11/30 17:10:23	drop	DMZ	inside	172.16.1.26			10.10.10.26			0	ping	deny	interzone-default	policy-deny	0	0		0

Figure 12: DMZ-Host Logs for denied ping to Inside-Host and DNS for HTTP.

		11/30 17:14:03	drop	DMZ	Kali	172.16.1.26			192.168.146.26			0	ping	deny	interzone-default	policy-deny	0	0		0
		11/30 17:14:03	drop	DMZ	outside	172.16.1.26			8.8.8.8			53	not-applicable	deny	interzone-default	policy-deny	0	0		0
		11/30 17:13:58	drop	DMZ	outside	172.16.1.26			8.8.8.8			53	not-applicable	deny	interzone-default	policy-deny	0	0		0
		11/30 17:13:58	drop	DMZ	outside	172.16.1.26			8.8.8.8			53	not-applicable	deny	interzone-default	policy-deny	0	0		0
		11/30 17:13:48	drop	DMZ	Kali	172.16.1.26			192.168.146.26			0	ping	deny	interzone-default	policy-deny	0	0		0

Figure 13: DMZ-Host Logs for denied ping to Kali Linux.

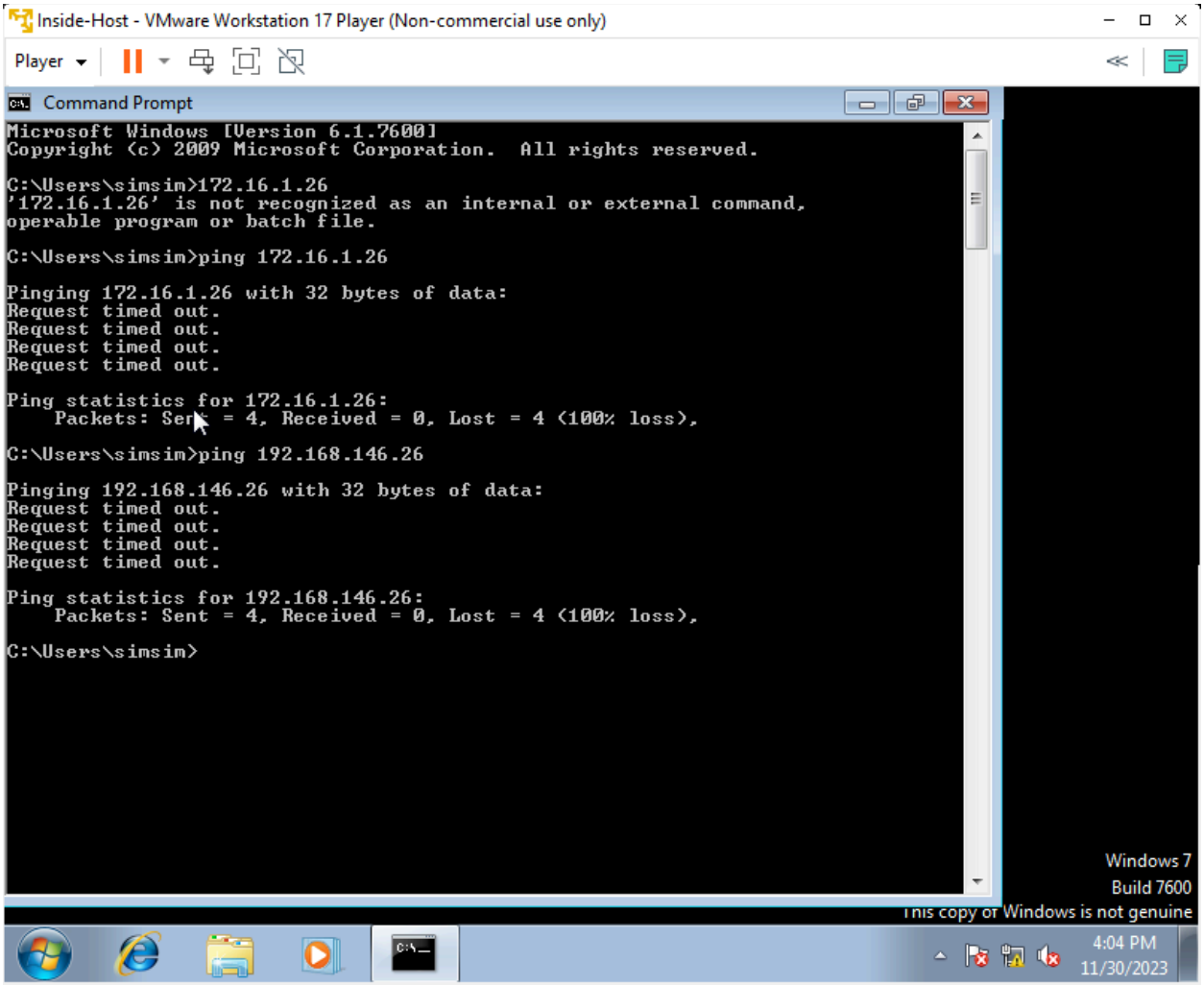

Figure 14: Inside-Host Denied Pings to Kali Linux and DMZ-Host

		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	ACTION	RULE	SESSION END REASON	BYTES	HTTP/2 CONNECTION SESSION ID	SDWAN SITE NAME	APP FLA COUNT
		11/30 17:06:48	drop	inside	Kali	10.10.10.26			192.168.146.26			0	ping	deny	interzone-default	policy-deny	0	0		0
		11/30 17:06:48	drop	inside	outside	10.10.10.26			8.8.8.8			53	not-applicable	deny	interzone-default	policy-deny	0	0		0
		11/30 17:06:33	drop	inside	Kali	10.10.10.26			192.168.146.26			0	ping	deny	interzone-default	policy-deny	0	0		0

Figure 15: Inside-Host log for denied Ping to Kali Linux.

		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	ACTION	RULE	SESSION END REASON	BYTES	HTTP/2 CONNECTION SESSION ID	SDWAN SITE NAME	APP FLA COUNT
		11/30 17:02:23	drop	inside	DMZ	10.10.10.26			172.16.1.26			0	ping	deny	interzone-default	policy-deny	0	0		0
		11/30 17:02:13	drop	DMZ	outside	172.16.1.26			8.8.8.8			53	not-applicable	deny	interzone-default	policy-deny	0	0		0
		11/30 17:02:08	drop	inside	DMZ	10.10.10.26			172.16.1.26			0	ping	deny	interzone-default	policy-deny	0	0		0

Figure 16: Inside-Host log for denied ping to DMZ-Host

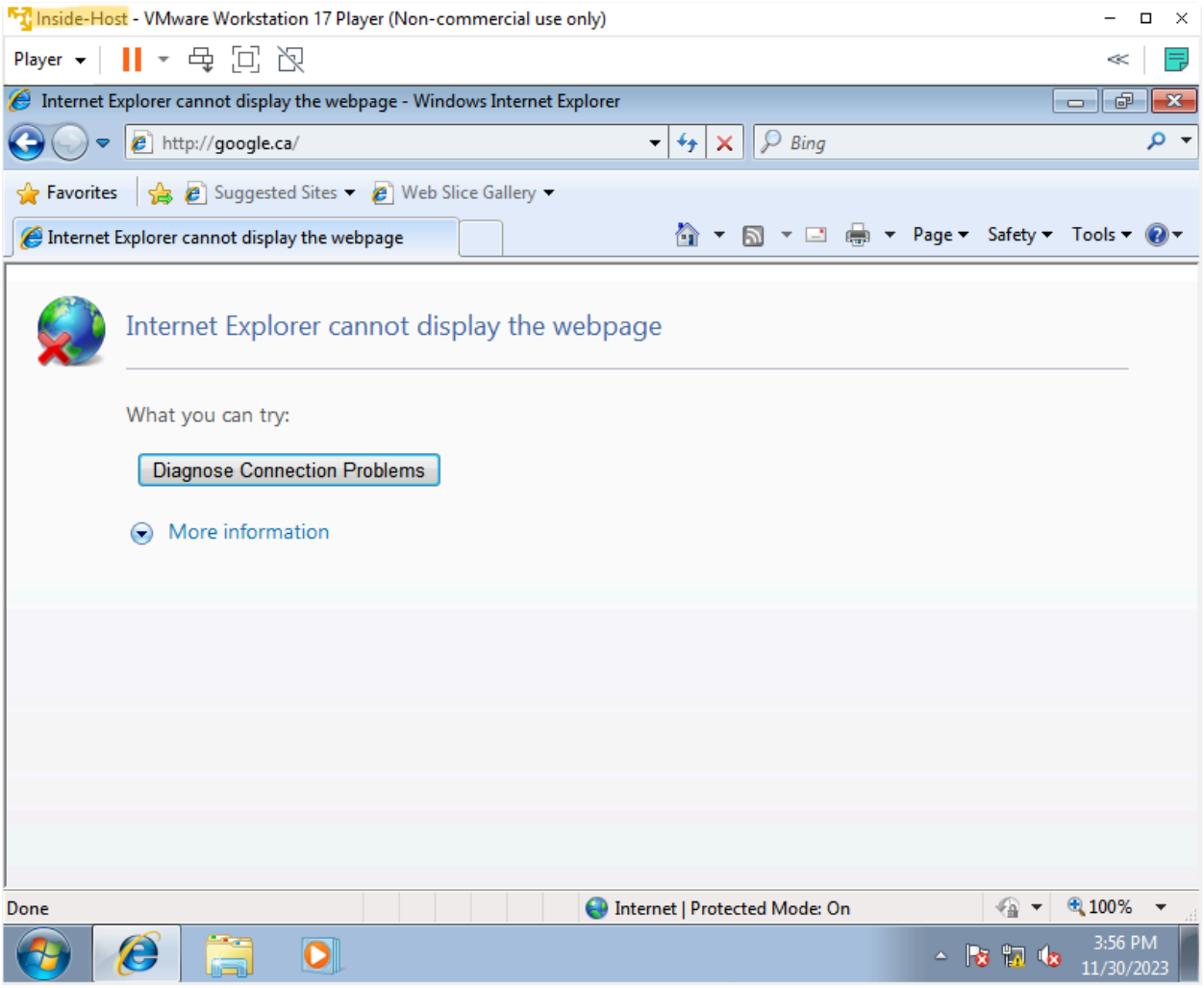

Figure 17: Inside-Host Denied HTTP request for google.ca.

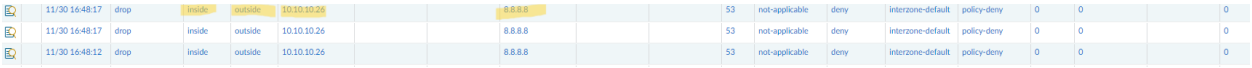

Figure 18: Inside-Host Log for Denied DNS/HTTP request.

3. Configure the required rules to allow the inside host and DMZ-Host to access the Internet using the DNS, HTTP, and HTTPS protocols.

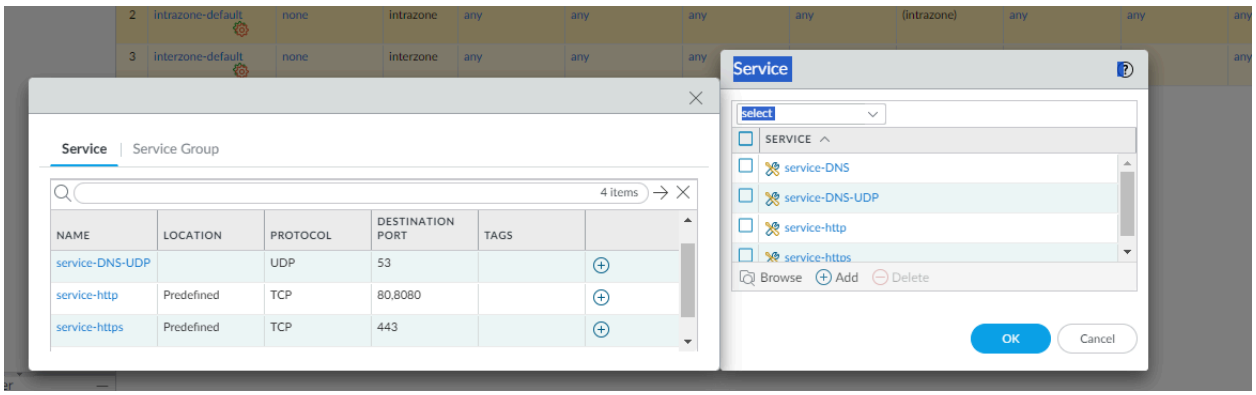

Figure 19: Service rules Applied to allow HTTP traffic.

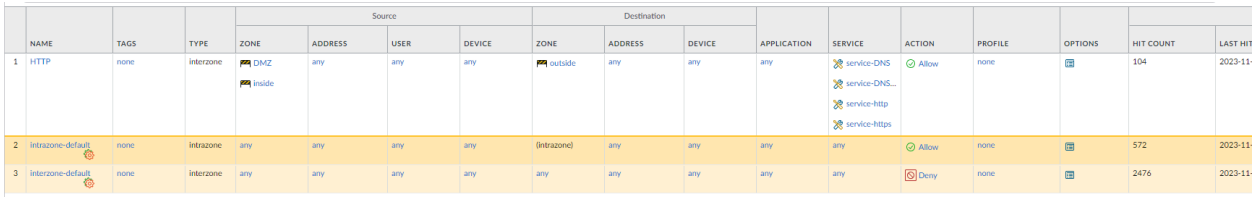

Figure 20: Table applied to the system.

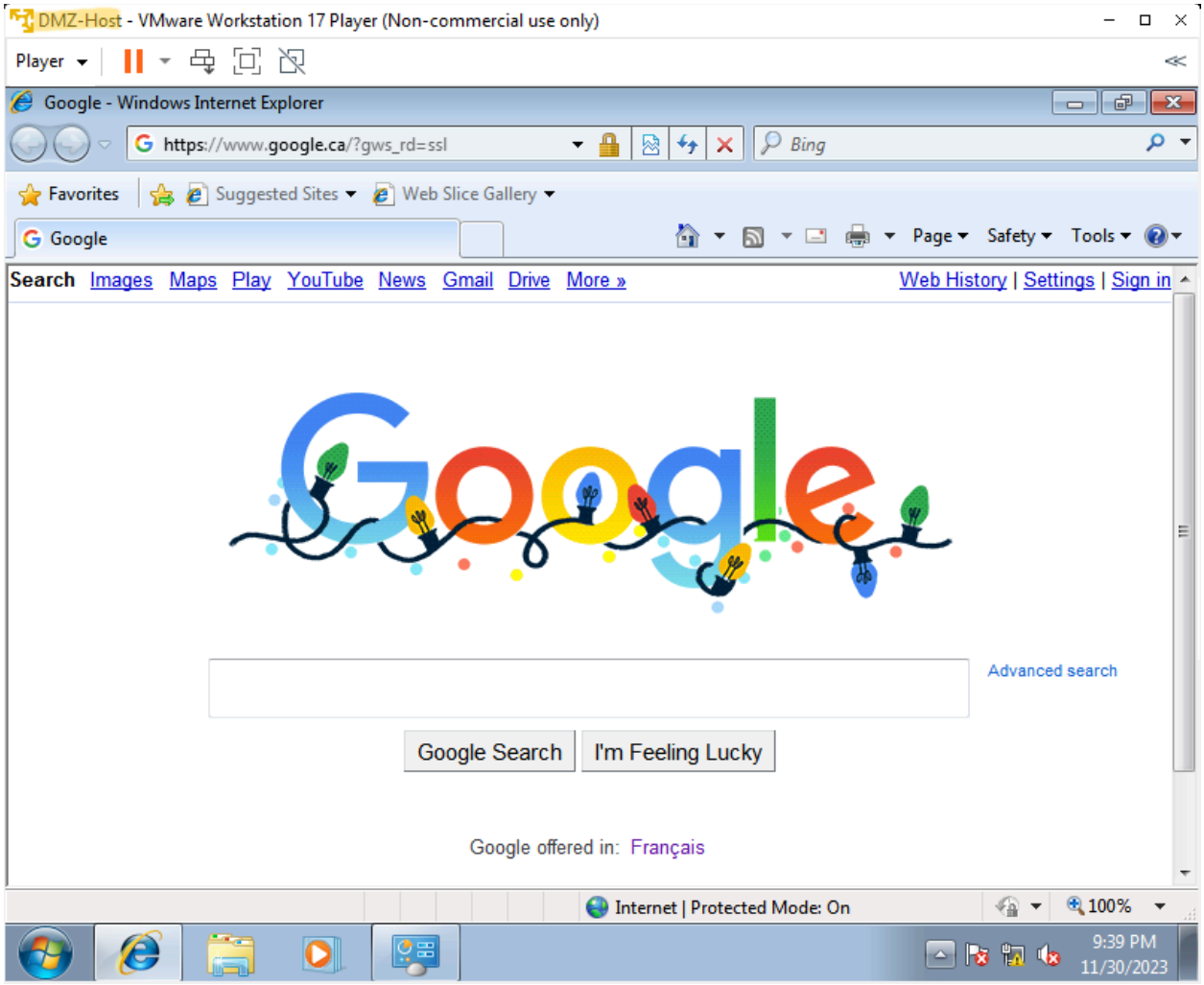

Figure 21: DMZ-Host showing Google and internet access.

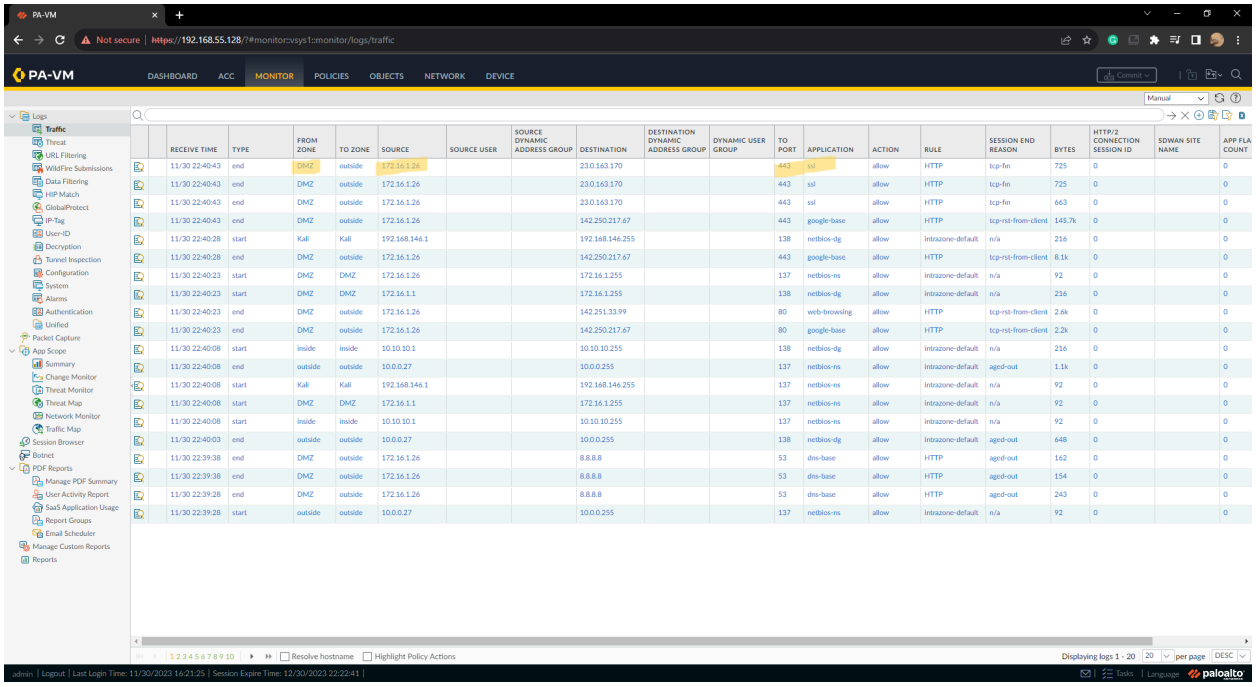

Figure 22: DMZ-Host internet access in logs.

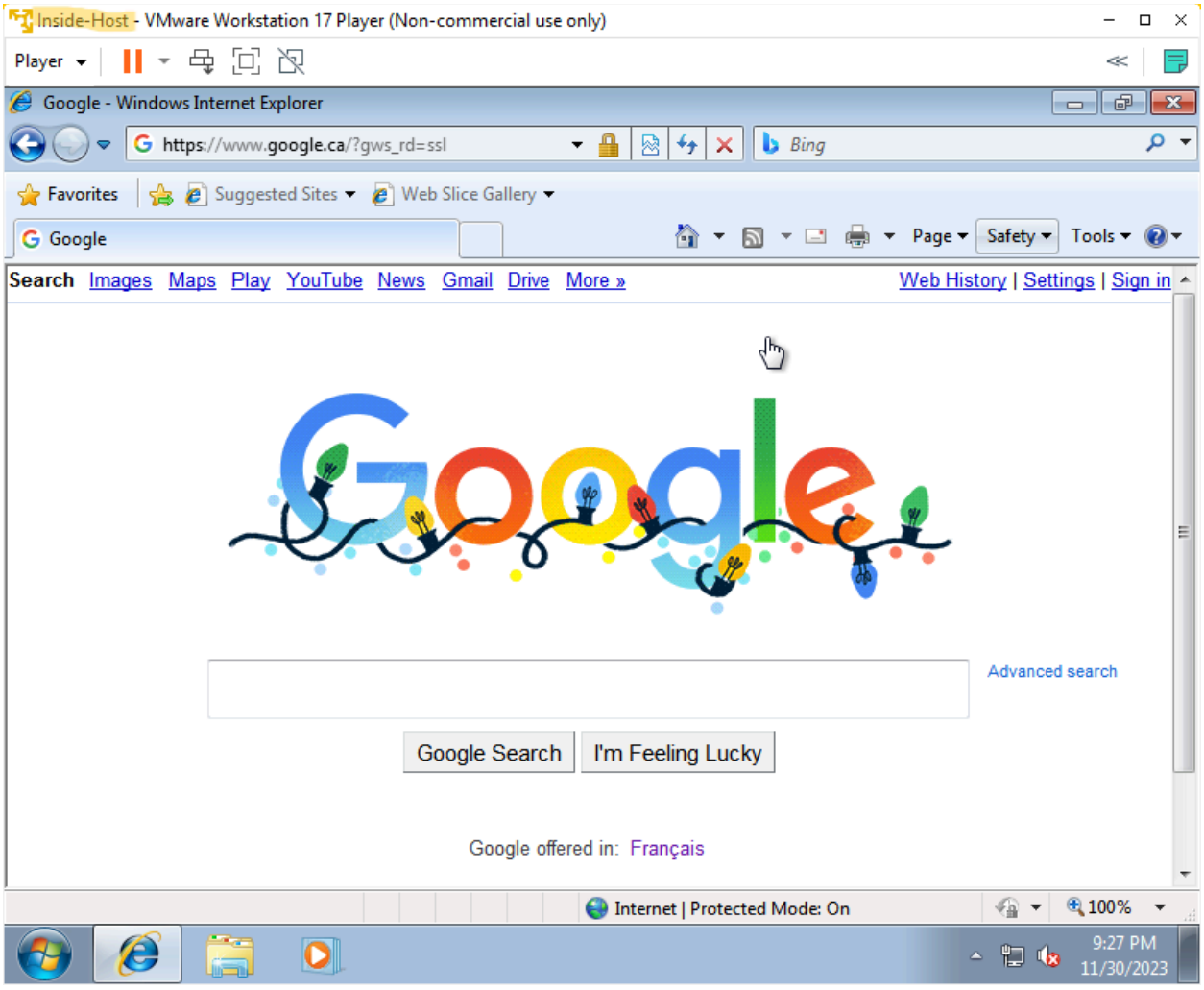

Figure 23: Inside-Host showing Internet access.

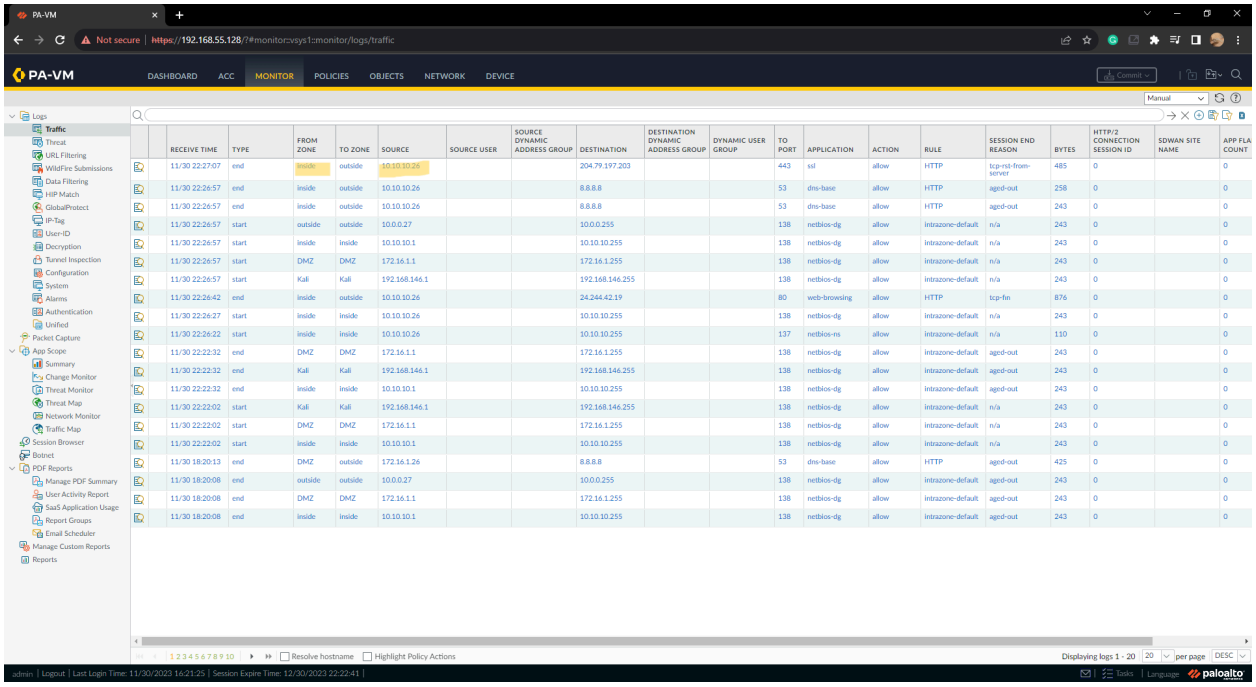

Figure 24: Inside-Host internet access in Logs.

4. Demonstrate Internet access over HTTP and HTTPS from Inside-Host and DMZ-Host, highlighting application awareness.

From our previous section, we can note the specific applications in the logs that have allowed google to run. Thus, we can add these applications to the application-aware entry.

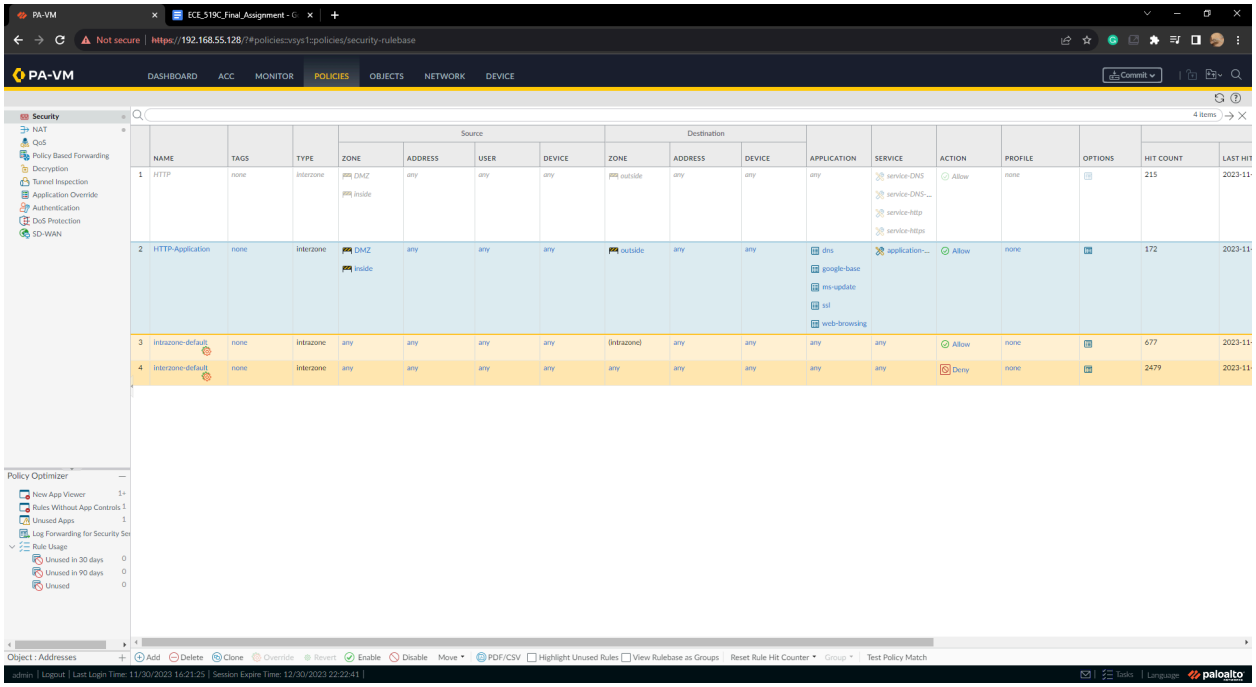

Figure 25: Switching from Service to Application-aware firewall rule.

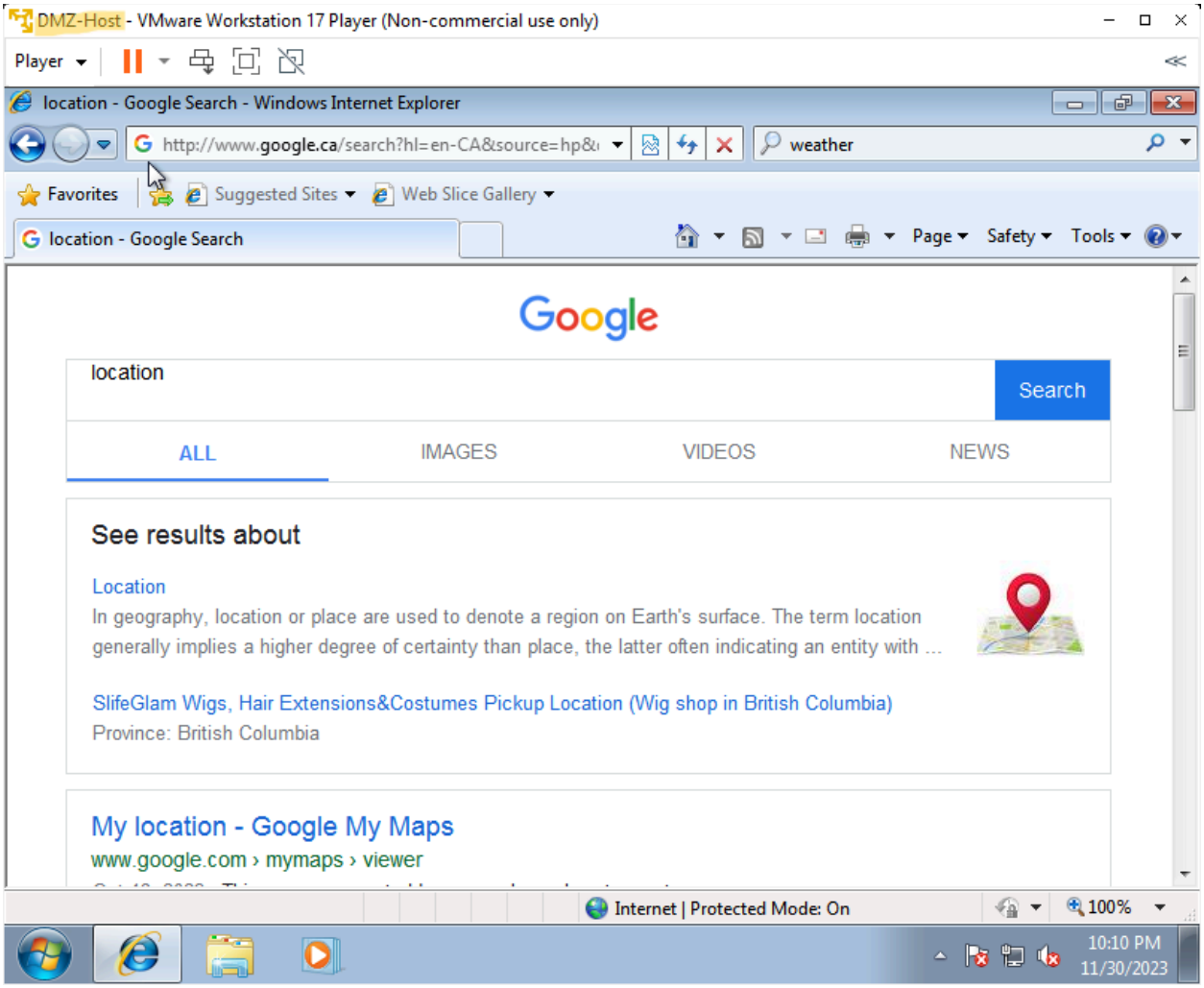

Figure 26: DMZ-Host internet search.

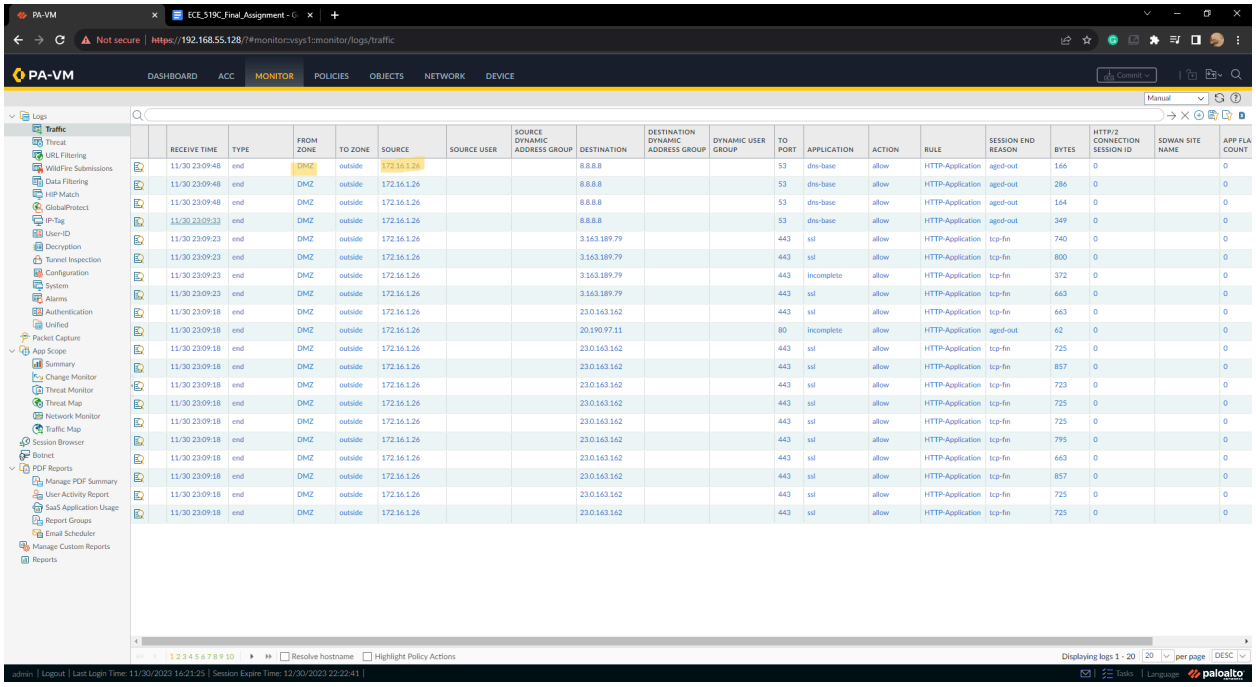

Figure 27: DMZ-Host logs showing application-aware firewall rule in action.

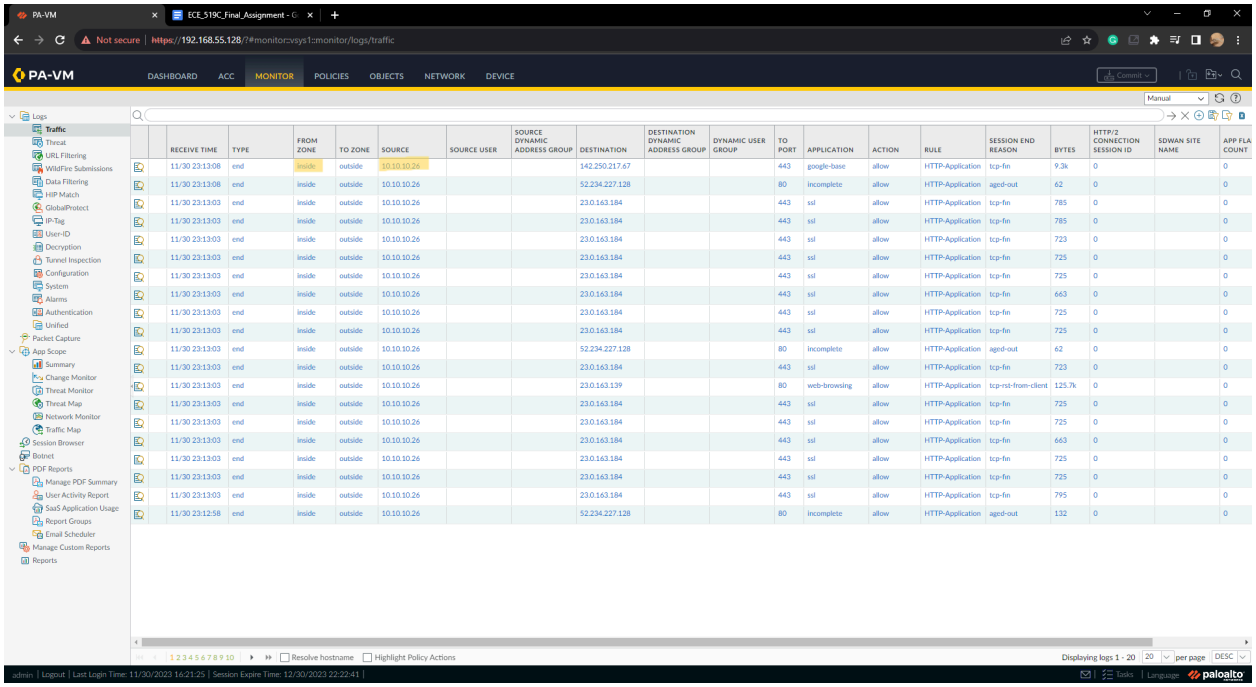

Figure 28: Inside-Host logs which show the application-aware firewall rule in action.

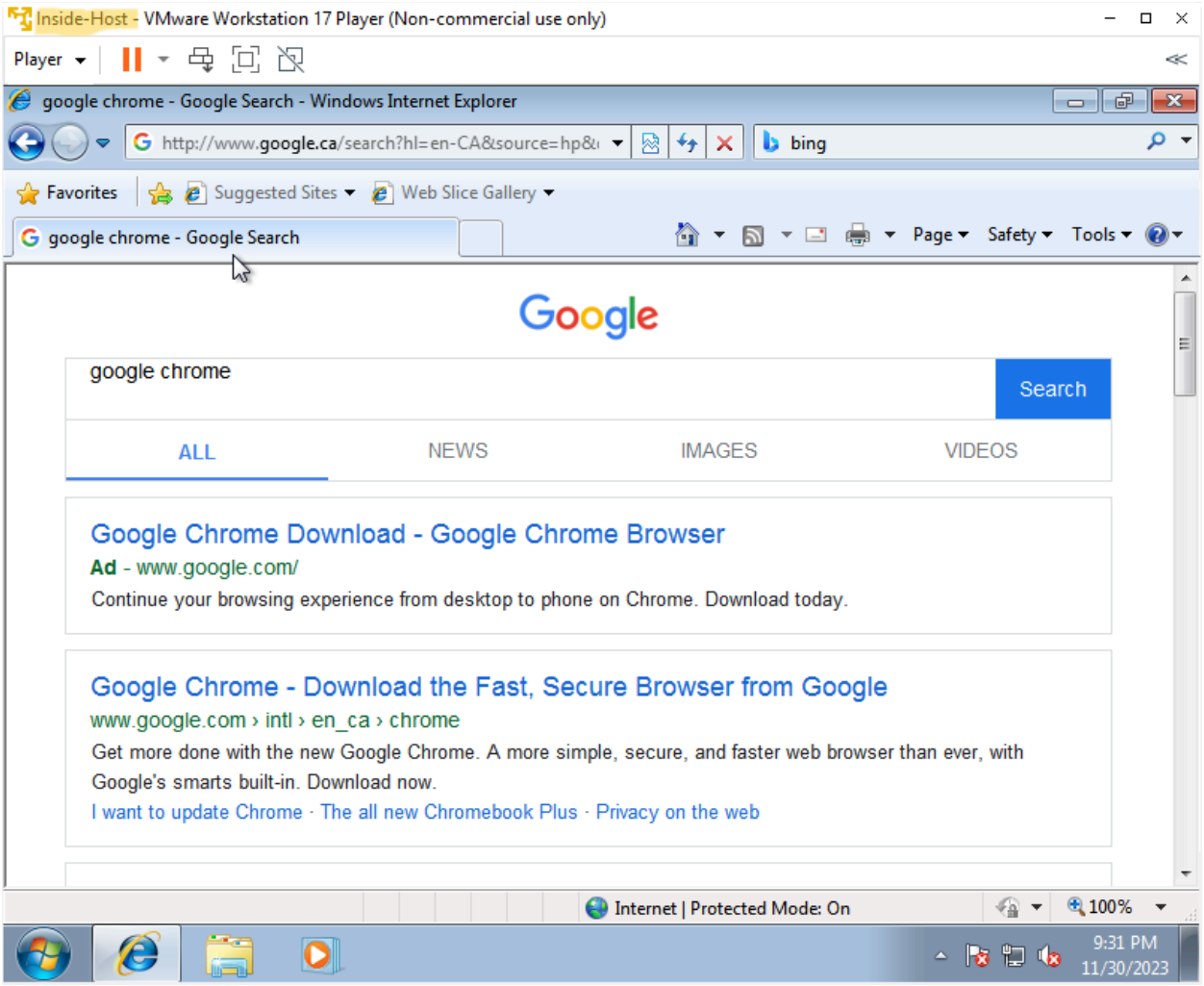

Figure 29: Inside-Host Internet search.

5. Download Google Chrome to the Inside-Host machine and try to access https://www.google.com using Google Chrome; why do you see denies in the monitor tab? Note: Google Chrome may fall back to using normal HTTPS rather than QUIC; however, in all cases, you will find some Drop logs in the “Monitor” tab.

Downloading Google Chrome was not a notable event as we needed to make sure that we found the version of Chrome available to Windows 7. It is important to note that since Internet Explorer is no longer supported by the most up-to-date Internet protocols, it is very refreshing to use

Google Chrome which is fully supported. It is also important to state that Chrome was able to connect to the internet as it used TCP protocols to function even with QUIC protocols disabled.

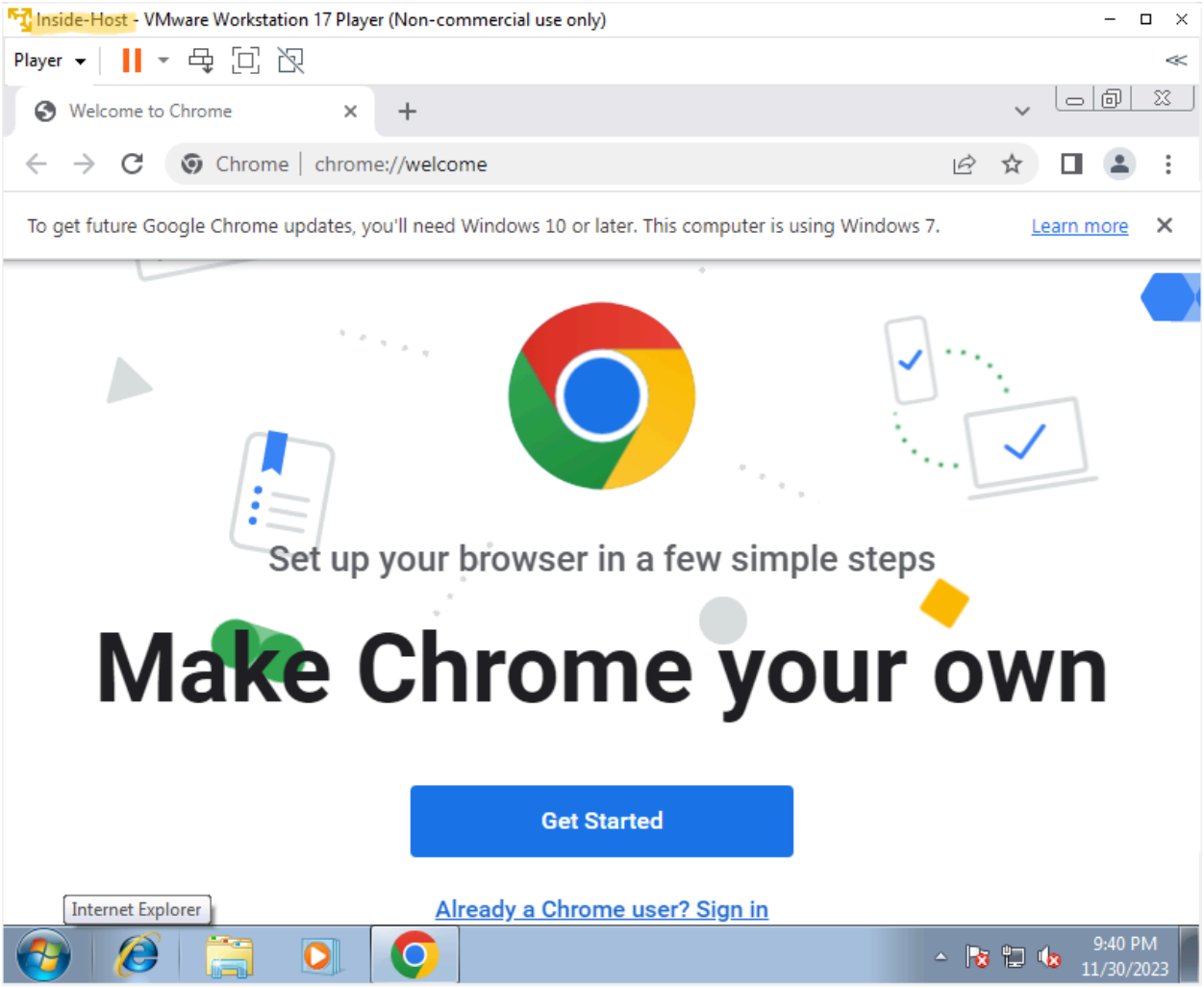

Figure 30: Chrome installed and running on Inside-Host.

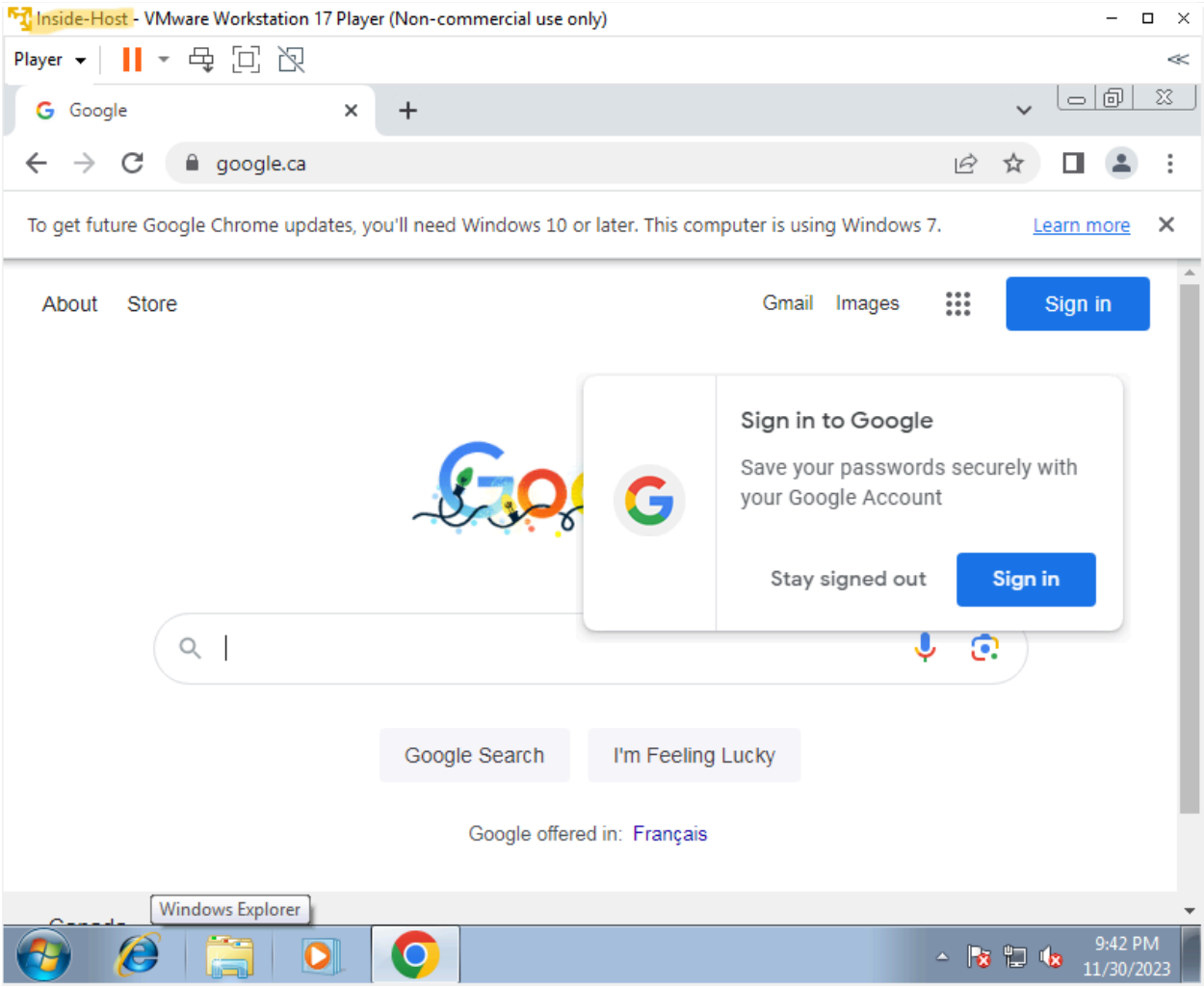

Figure 31: Inside-Host Google attempt.

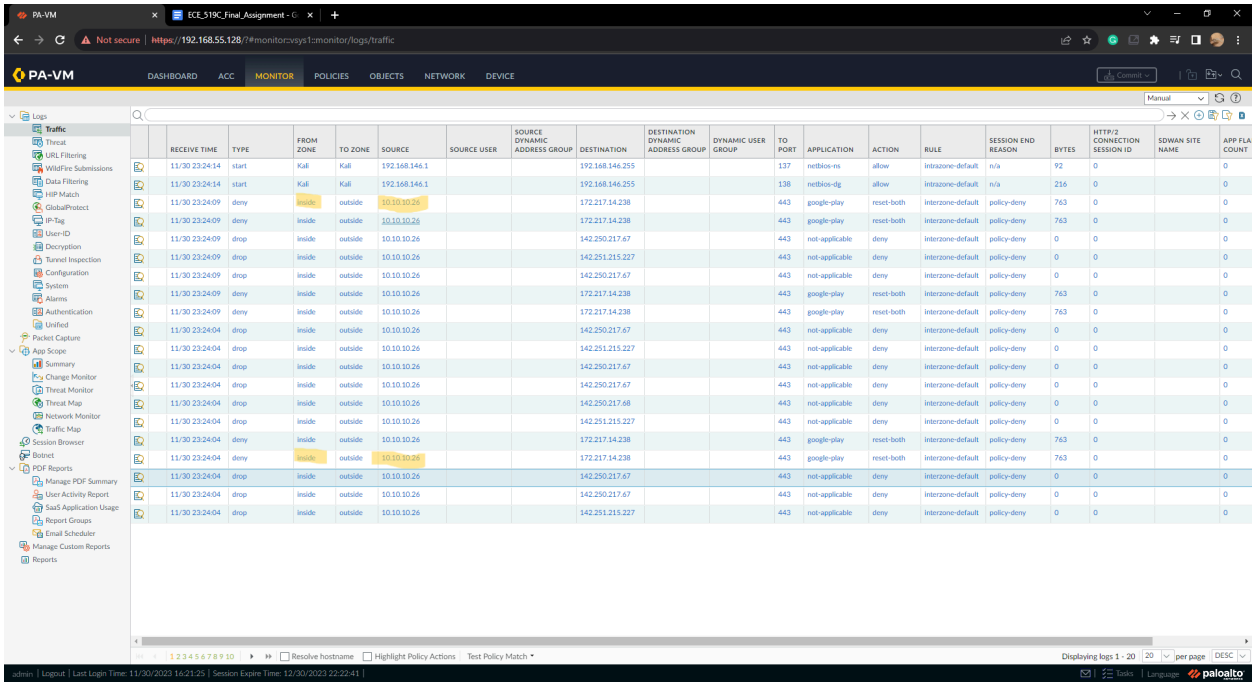

Figure 32: Log showing the dropped applications.

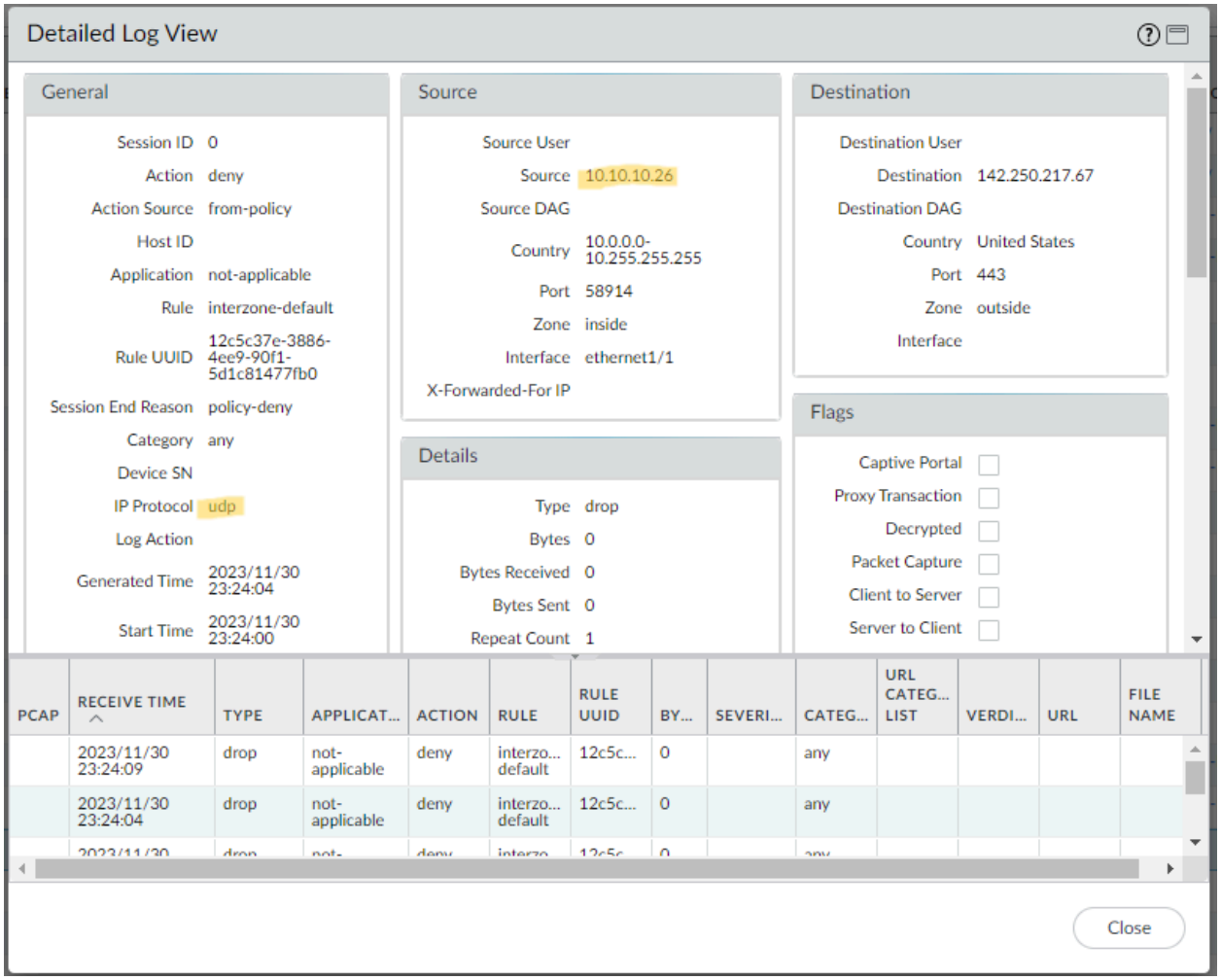

Figure 33: Detailed log showing dropped UDP protocol.

6. Allow Google websites access from the inside host using Google Chrome.

While application awareness is an important aspect of Next-generation firewalls, you need to understand every application which can contribute to internet connectivity. As such, I defaulted back to utilizing general service ports in order to complete the following tasks.

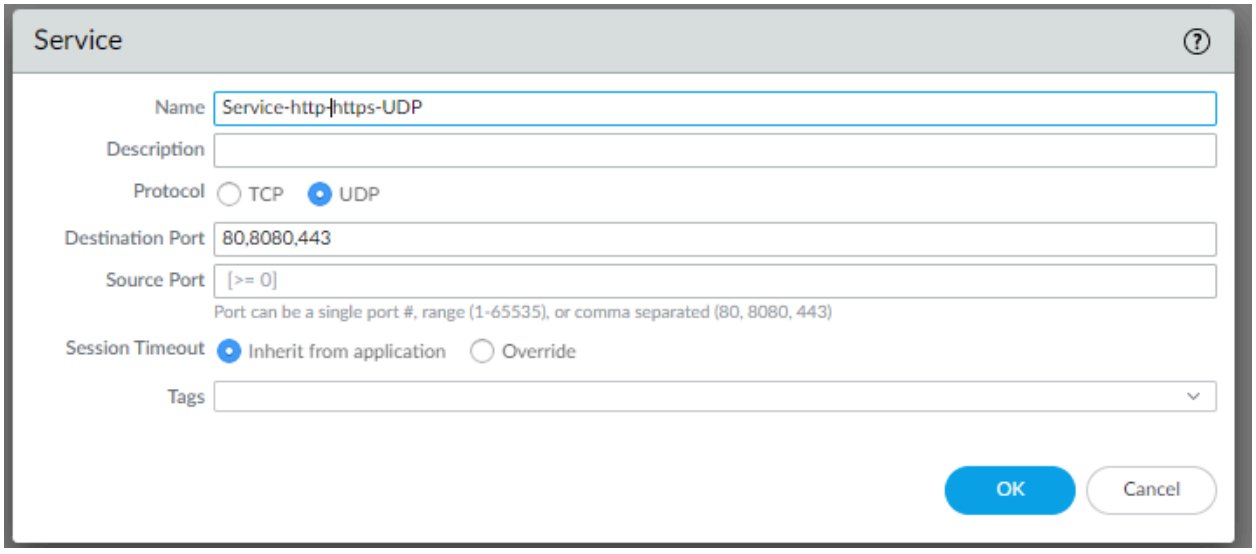

Figure 34: Configuring UDP port access in next-generation firewalls.

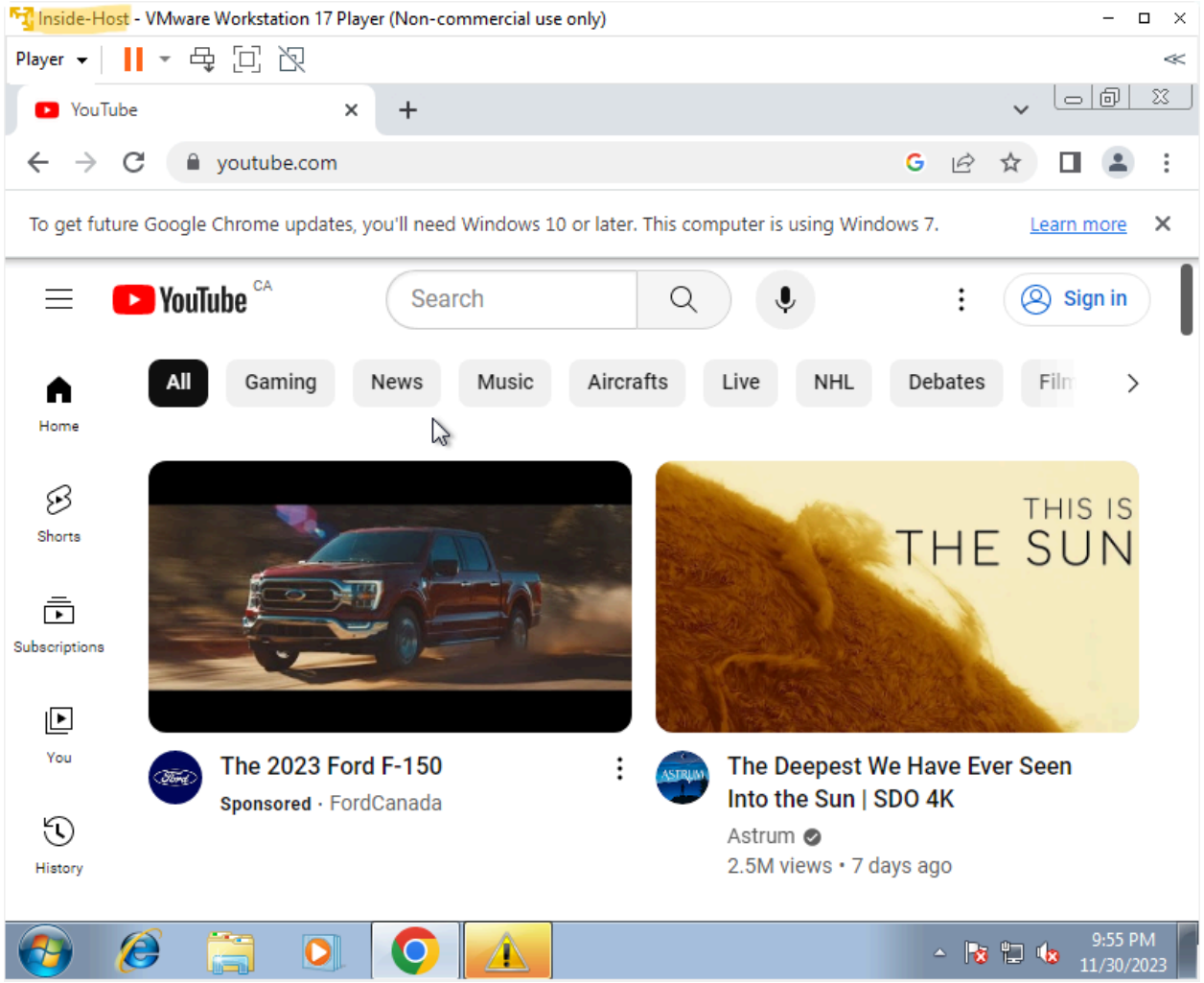

Figure 35: Chrome search from Inside Host.

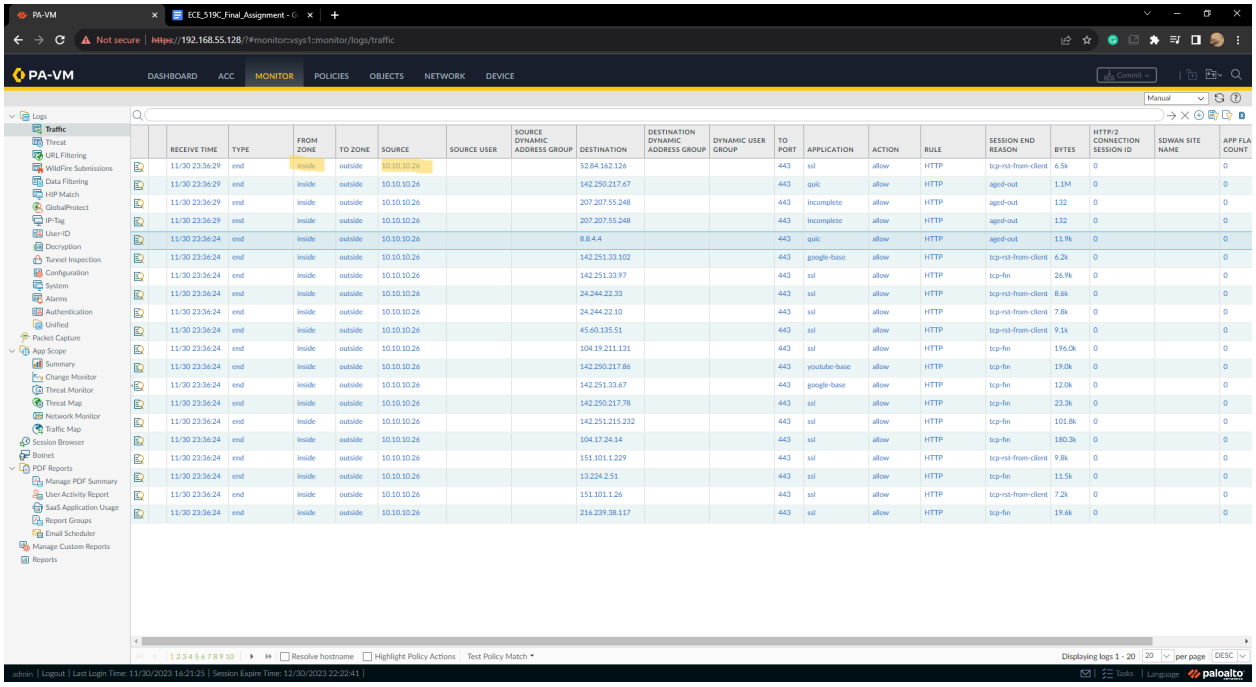

Figure 36: Logs showing QUIC and other UDP protocols allowed in Inside-Host.

7. Apply an HTTPS inspection rule for traffic from Inside-Host to the Internet.

We made a general SSL forward decryption object which allows for the decryption of outbound internet traffic so that we can inspect HTTPS traffic.

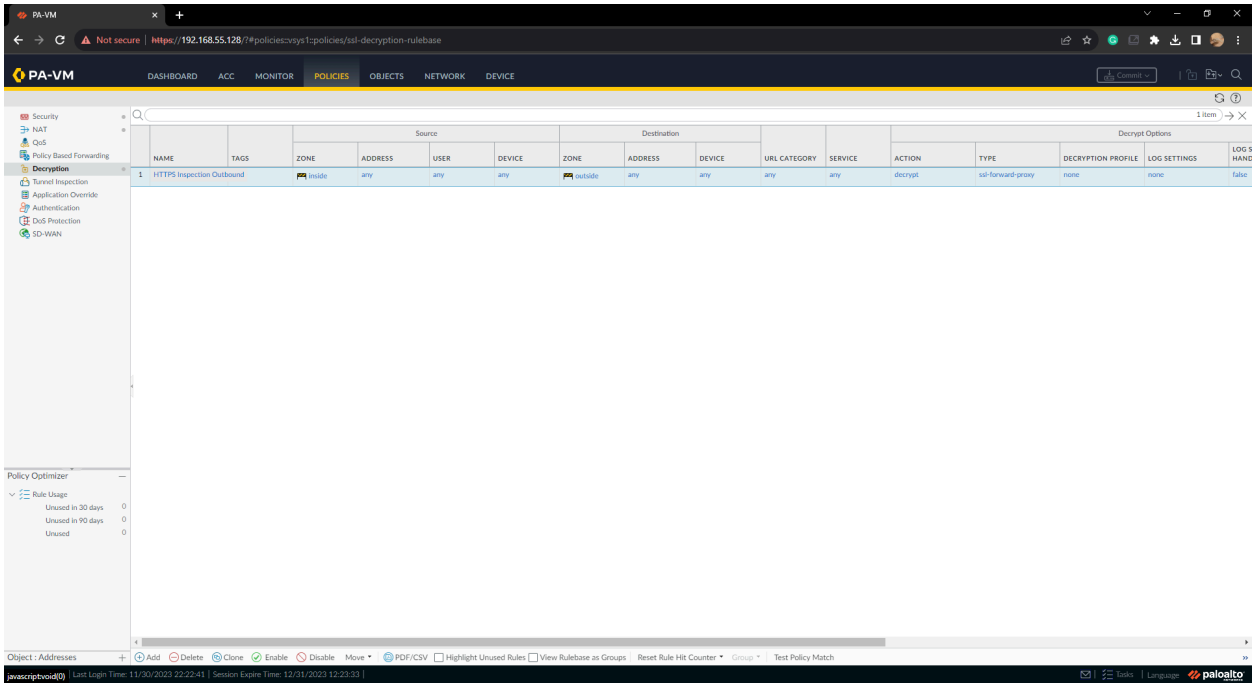

Figure 37: Enabling Forward SSL decryption policy for outbound traffic from inside to outside.

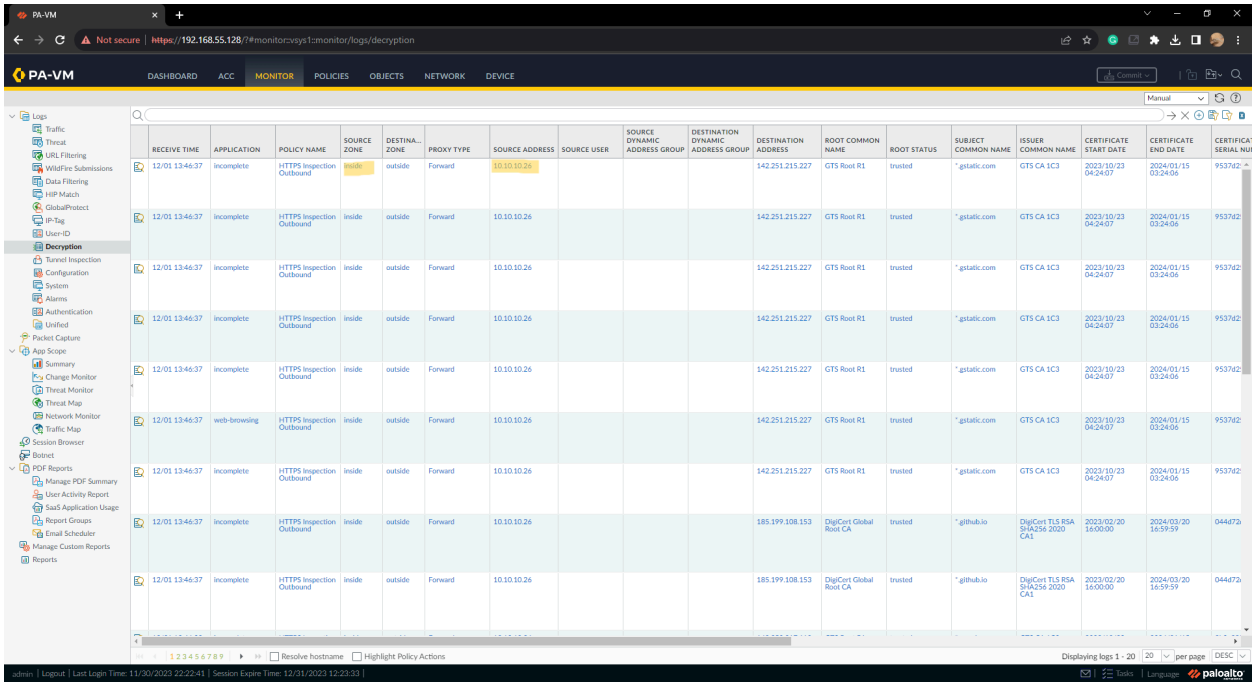

Figure 38: Decryption Log showing the HTTP inspection Rule in Action.

8. Allow the inside host to access Facebook but not Facebook Chat.

In order to do this, first we needed to see what kind of applications work when accessing https://facebook.ca and https://messenger.ca in order to determine what applications to block.

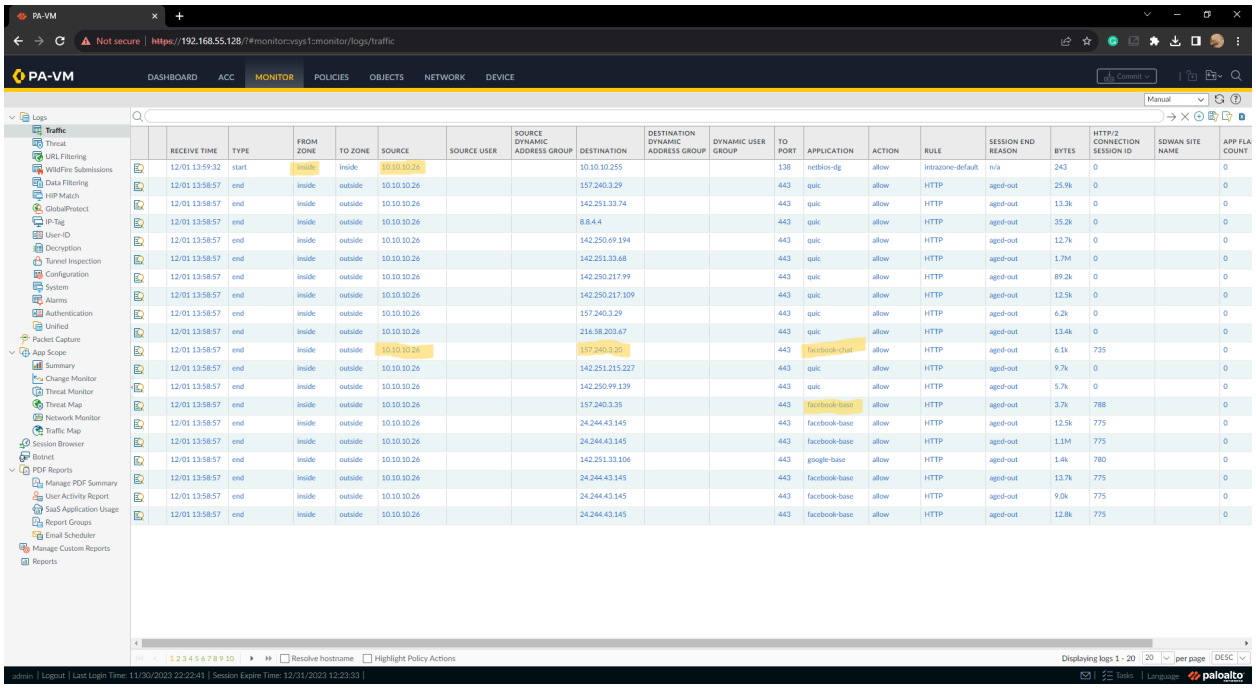

Figure 39: Logs showing Facebook applications.

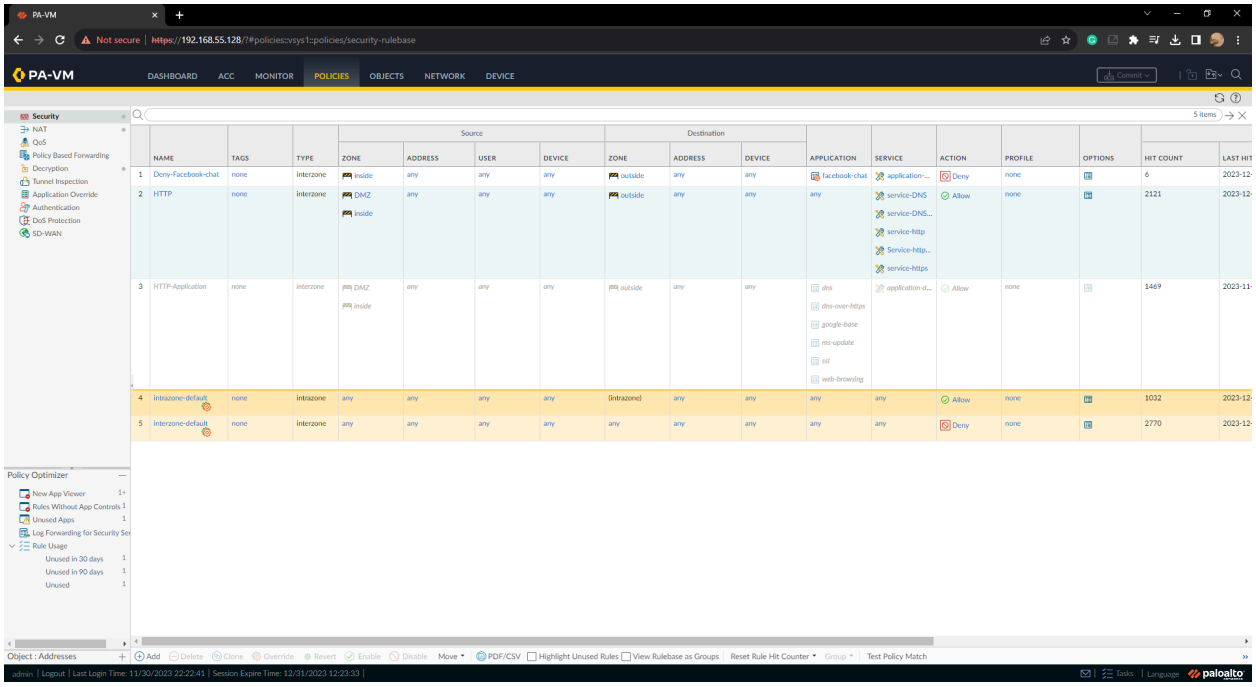

Figure 40: Configuring access rule to deny access for Facebook-chat application from the inside host.

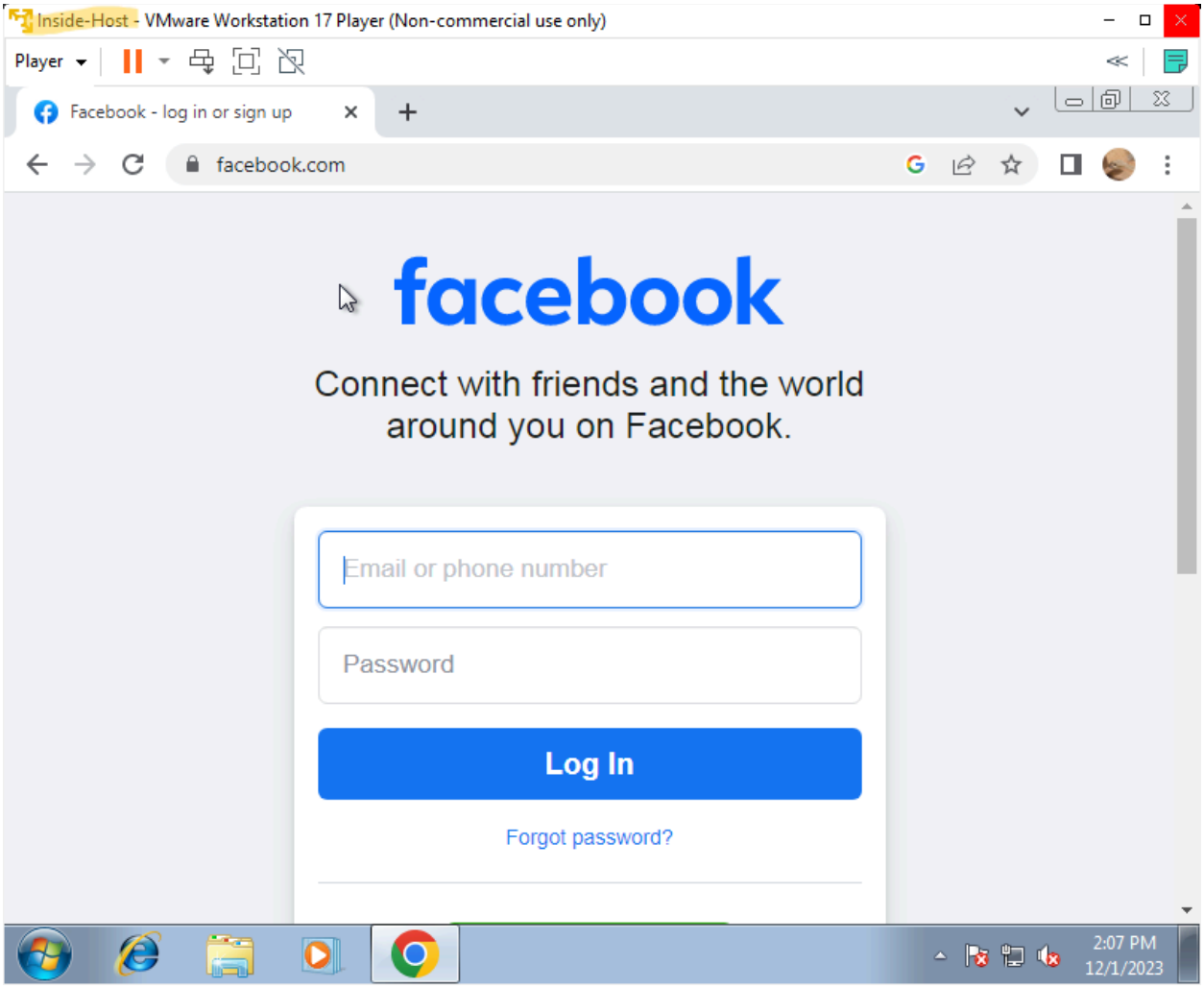

Figure 41: Inside-Host connecting to Facebook.

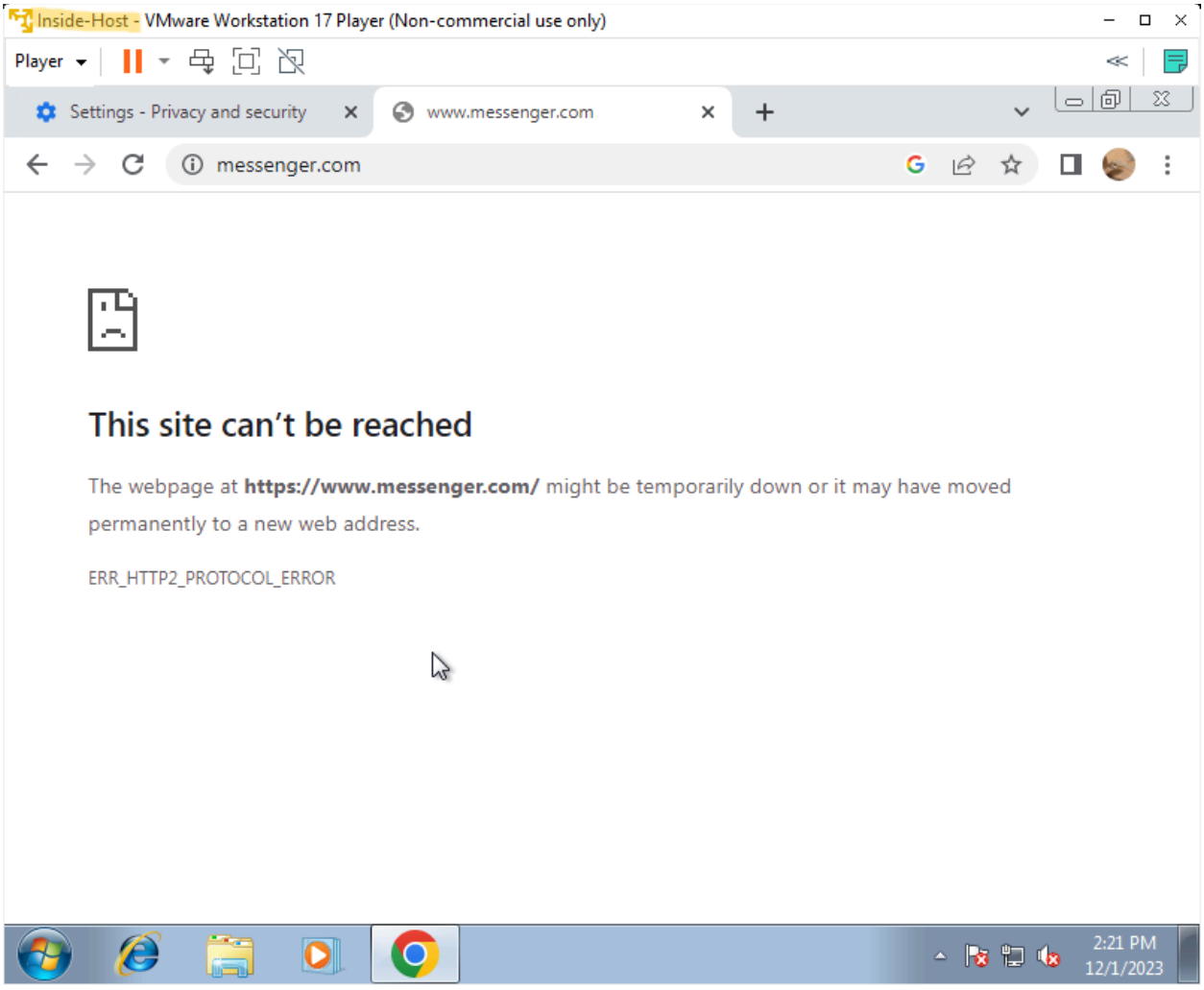

Figure 42: Inside-Host blocked access to messenger/facebook chat.

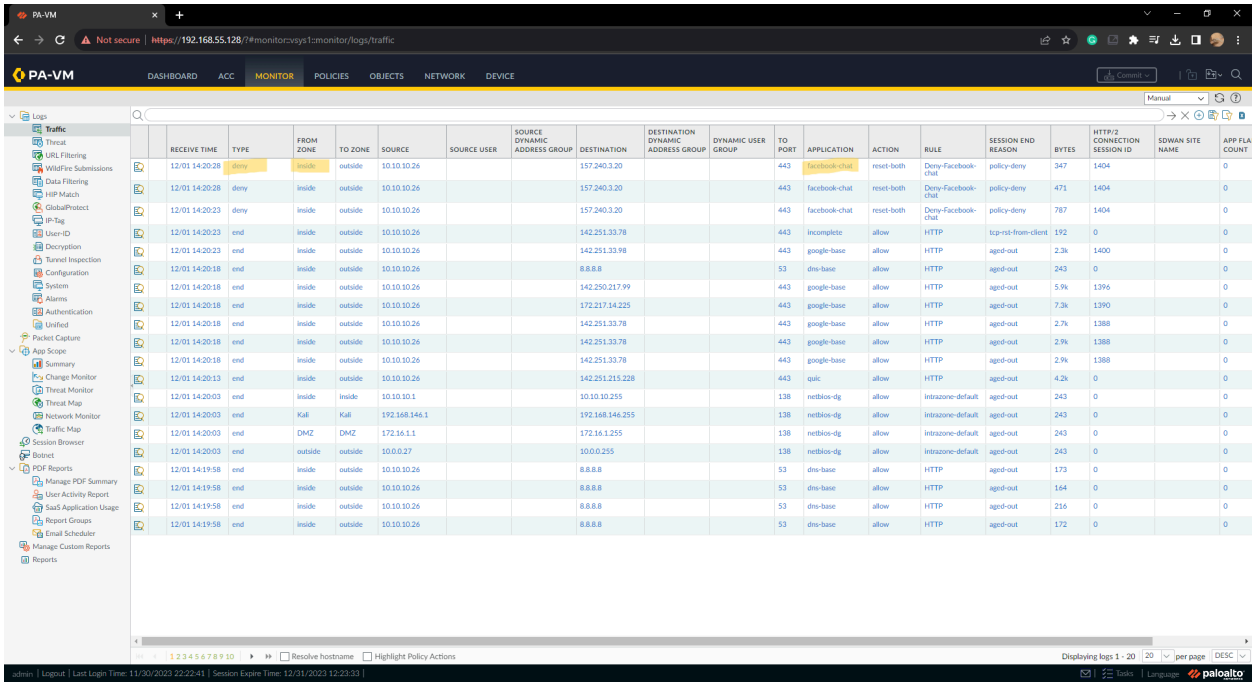

Figure 43: Logs showing denied access to Facebook chat.

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

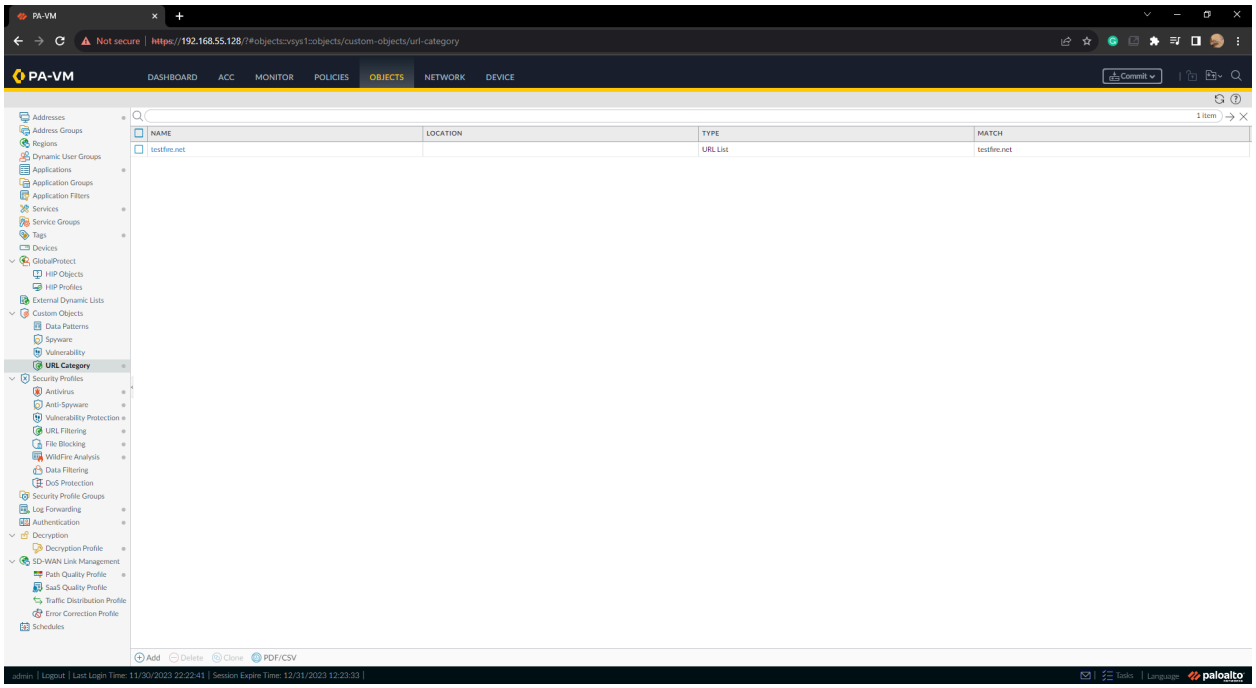

Figure 44: URL Category solely with Testfire.net.

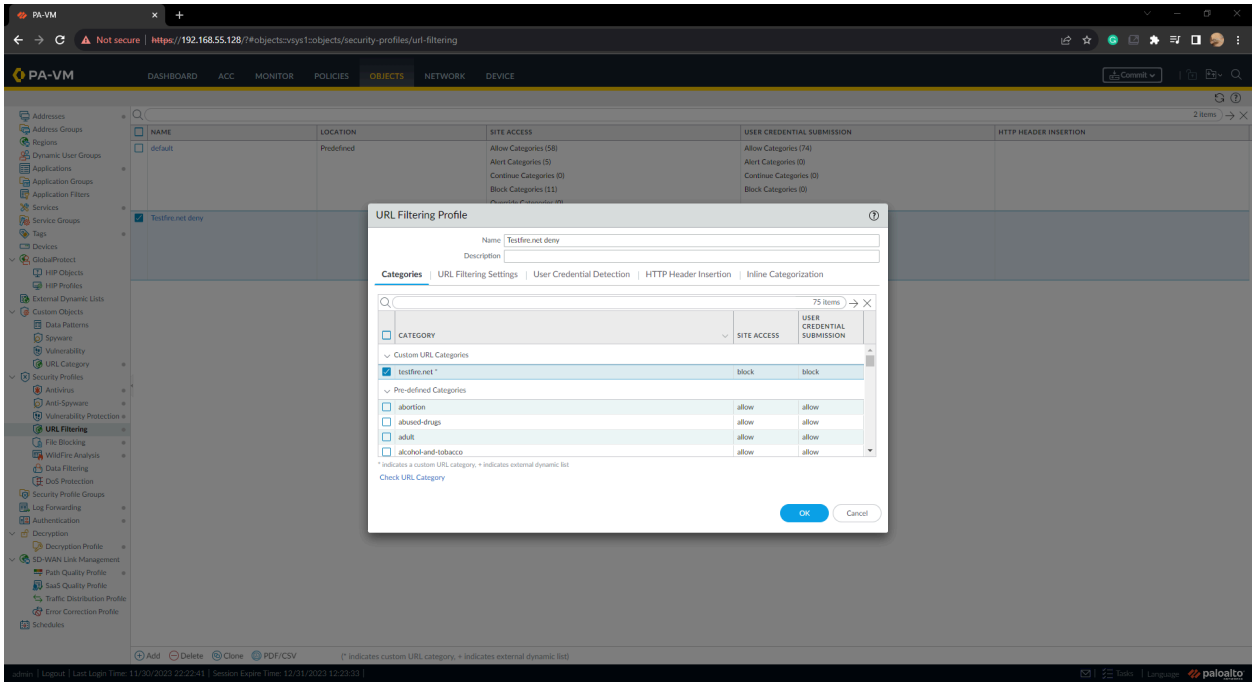

Figure 45: URL filtering profile with testfire.net being blocked access.

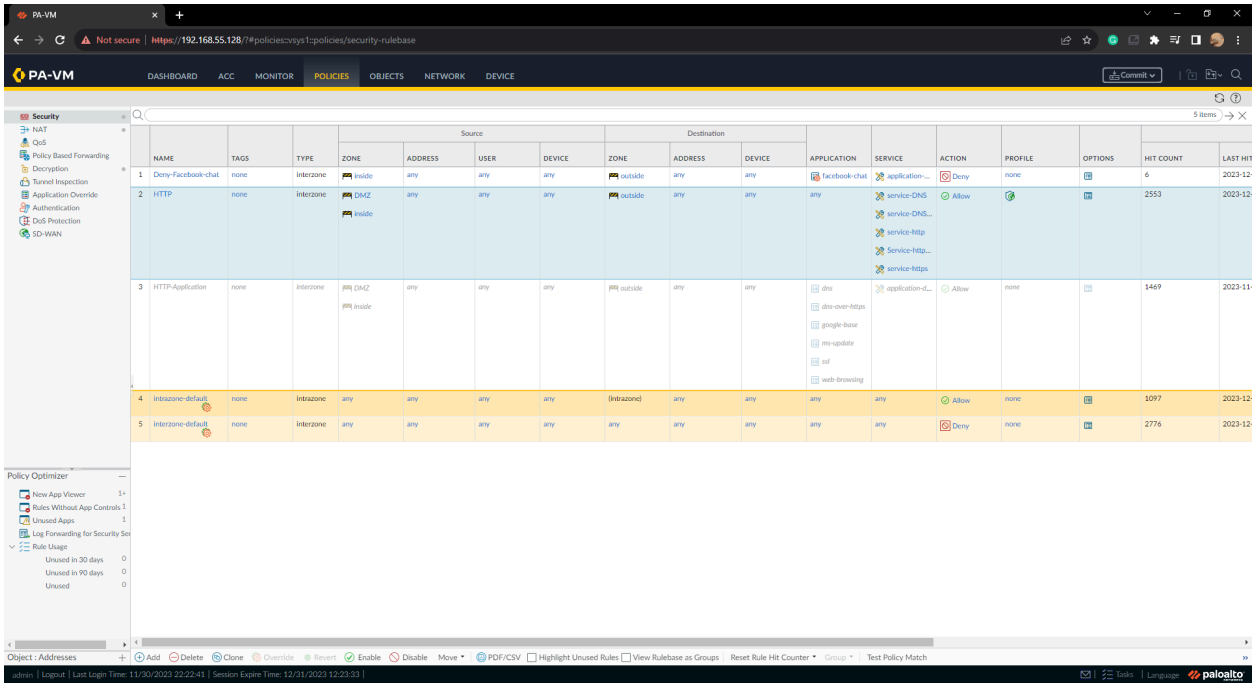

Figure 46: Profile column displays URL filtering, which uses the aforementioned profile.

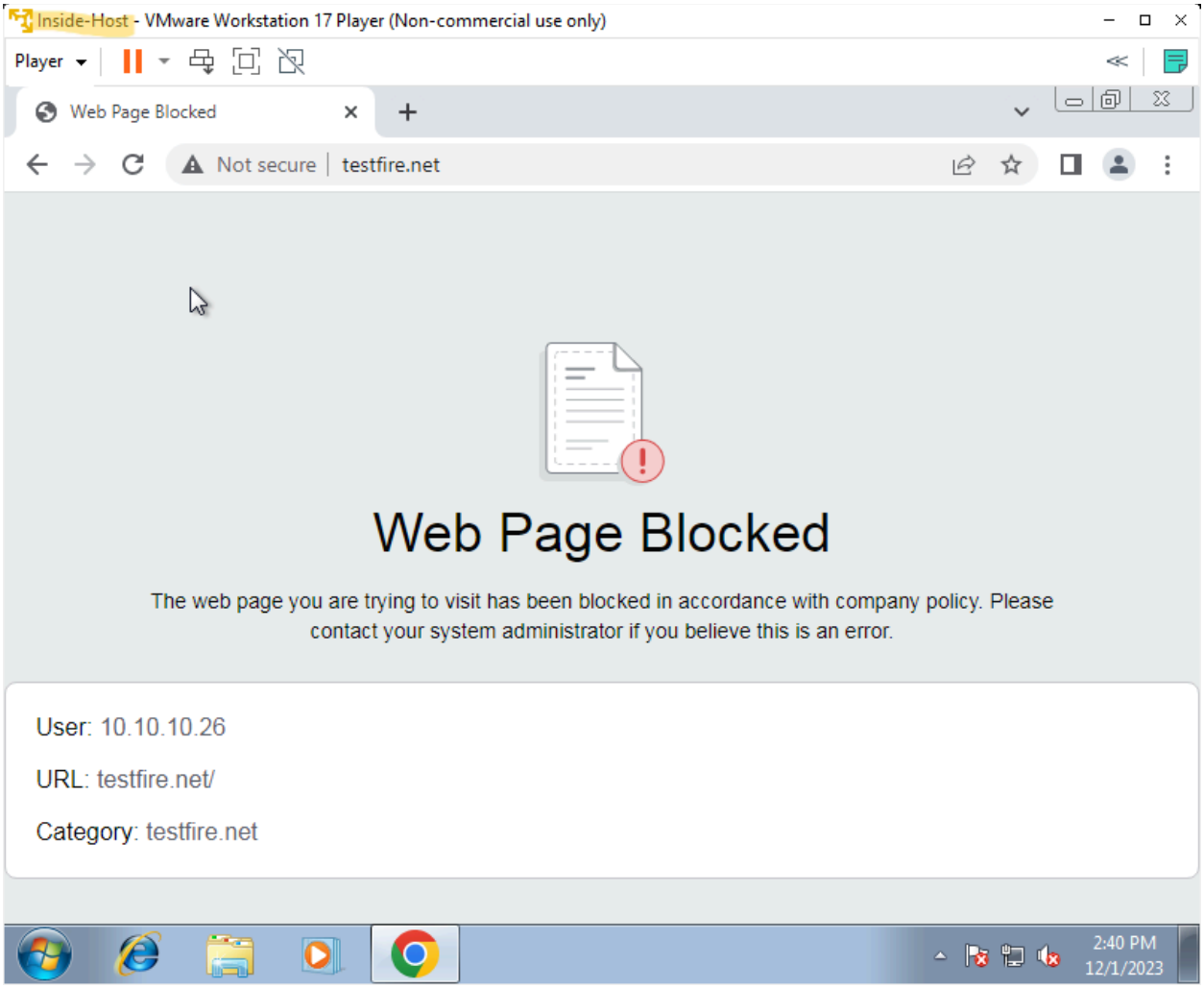

Figure 47: testfire.net Web Page blocked on Inside-Host

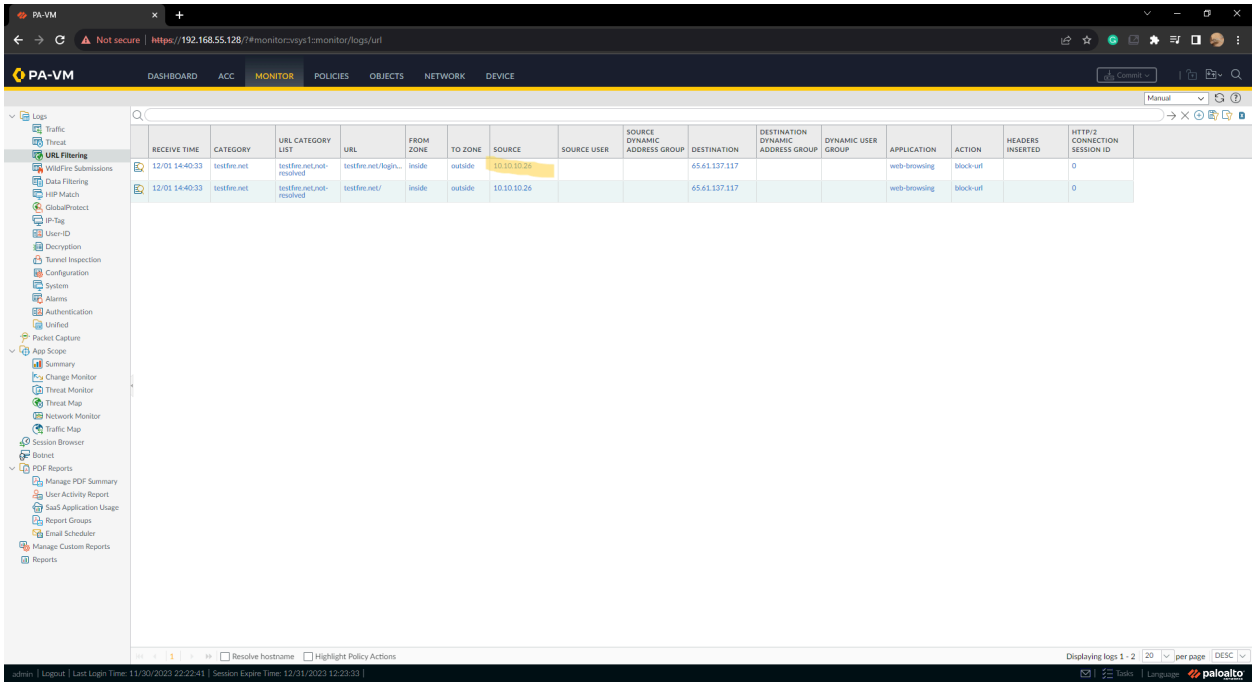

Figure 48: URL filtering log showing blocked access.

10. Apply antivirus inspection to traffic from Inside-Host to the Internet.

Luckily, our system has antivirus software installed, therefore, we can simply add the Default antivirus profile to our HTTPS access rule.

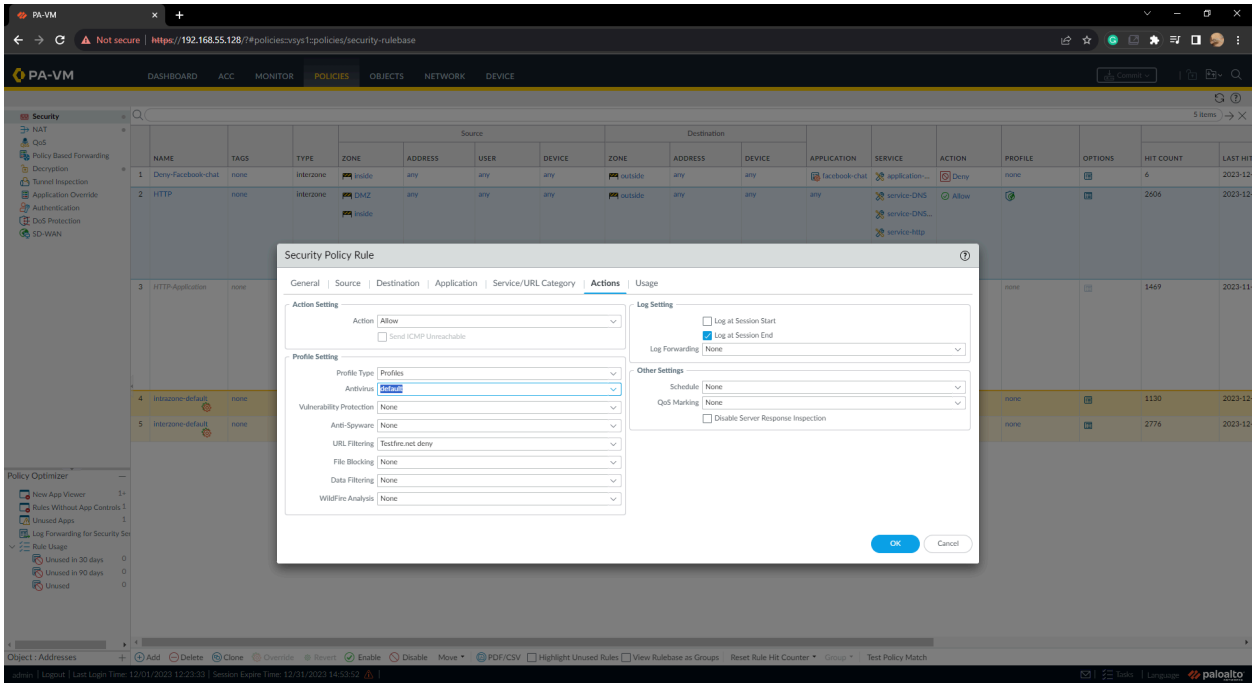

Figure 49: Adding antivirus inspection to HTTP rule.

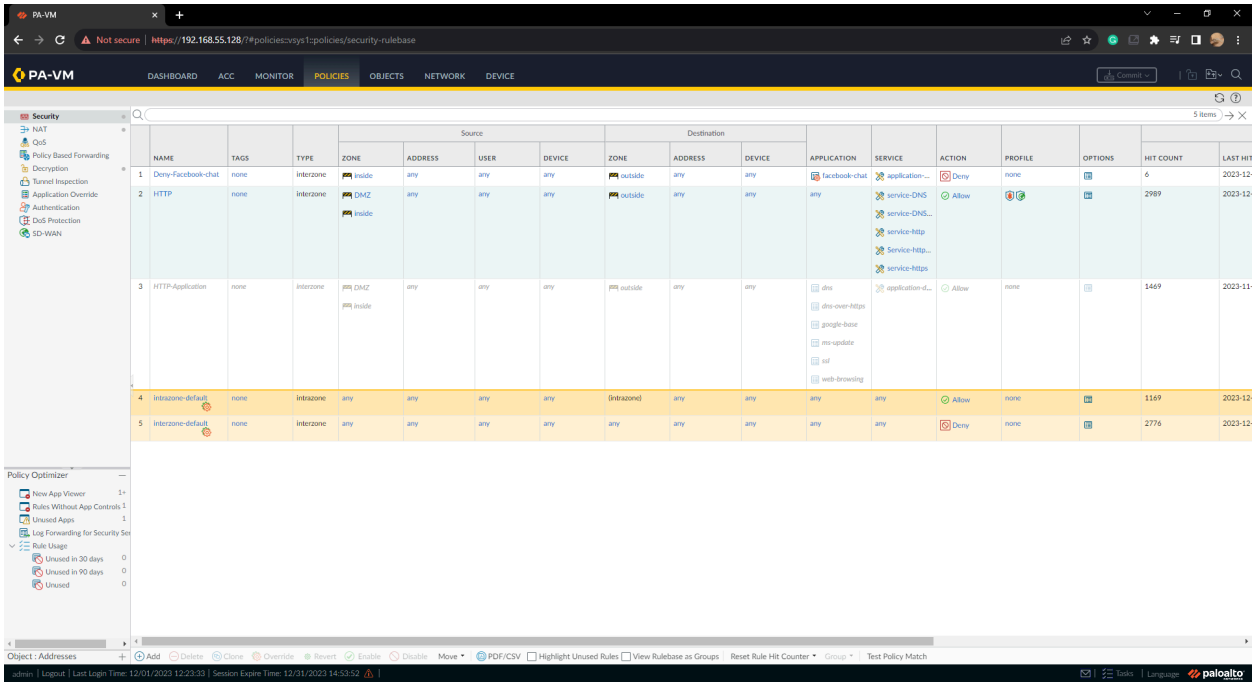

Figure 50: Access rule now configured with Anti-Virus inspection in the profile section.

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

We downloaded all versions of the Eicar test file. All downloads were blocked by the antivirus inspection policy, and the logs were then displayed in the threats log and noted as a virus.

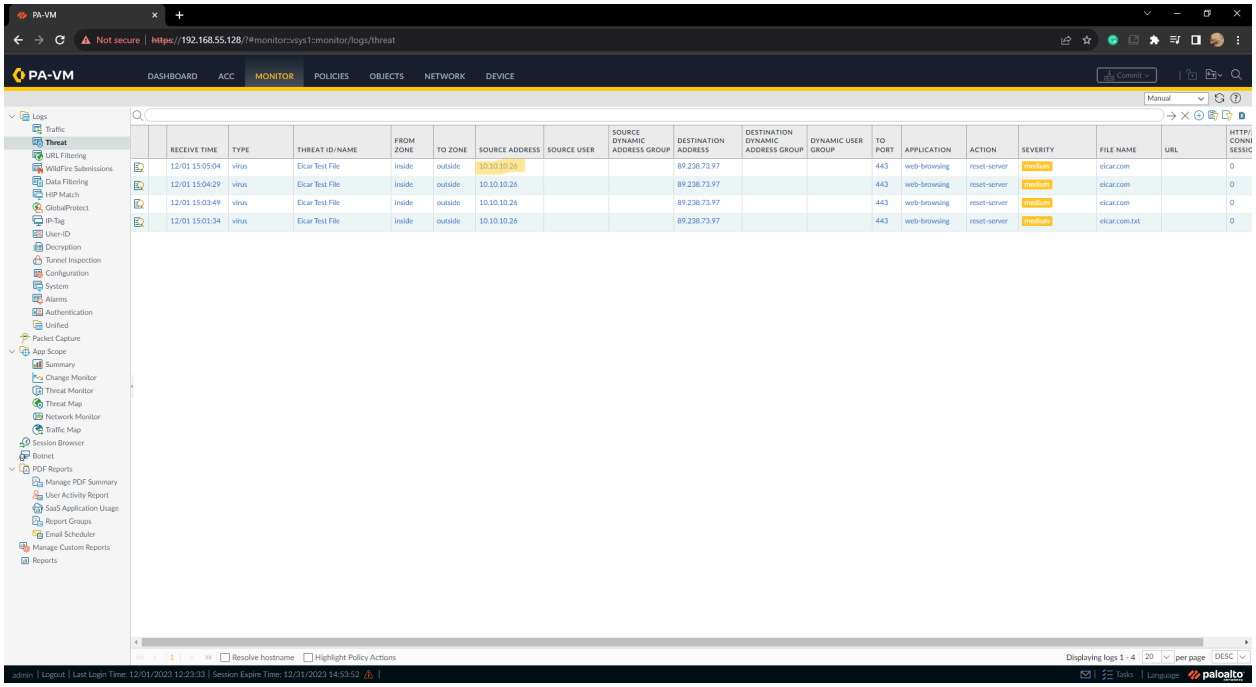

Figure 51: Threats logs showing the Eicar test blocked and labelled as a virus.

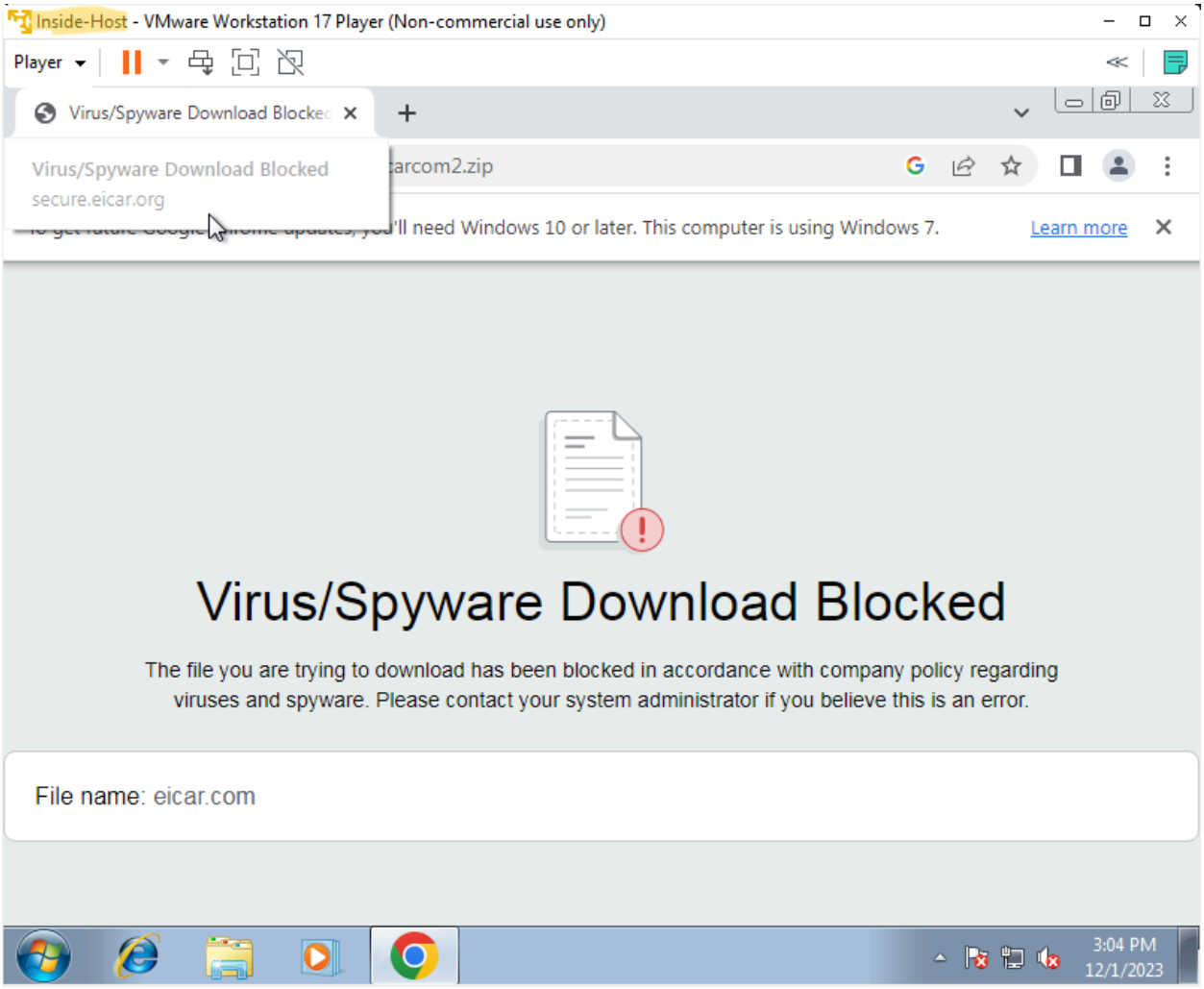

Figure 52: The Eicar.com2.zip file download was blocked on Inside-Host.

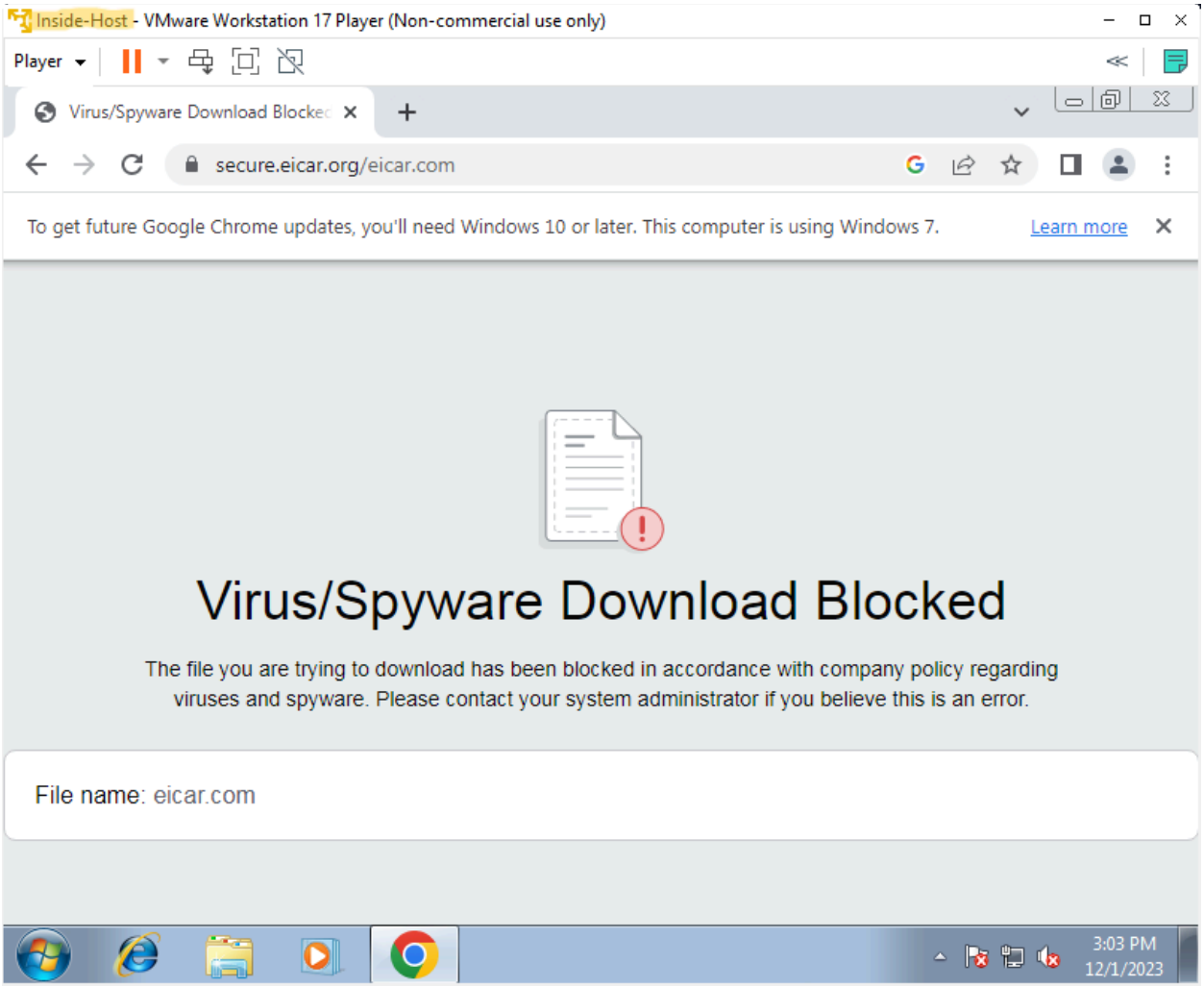

Figure 53: The Eicar.com test file was blocked on Inside-Host.

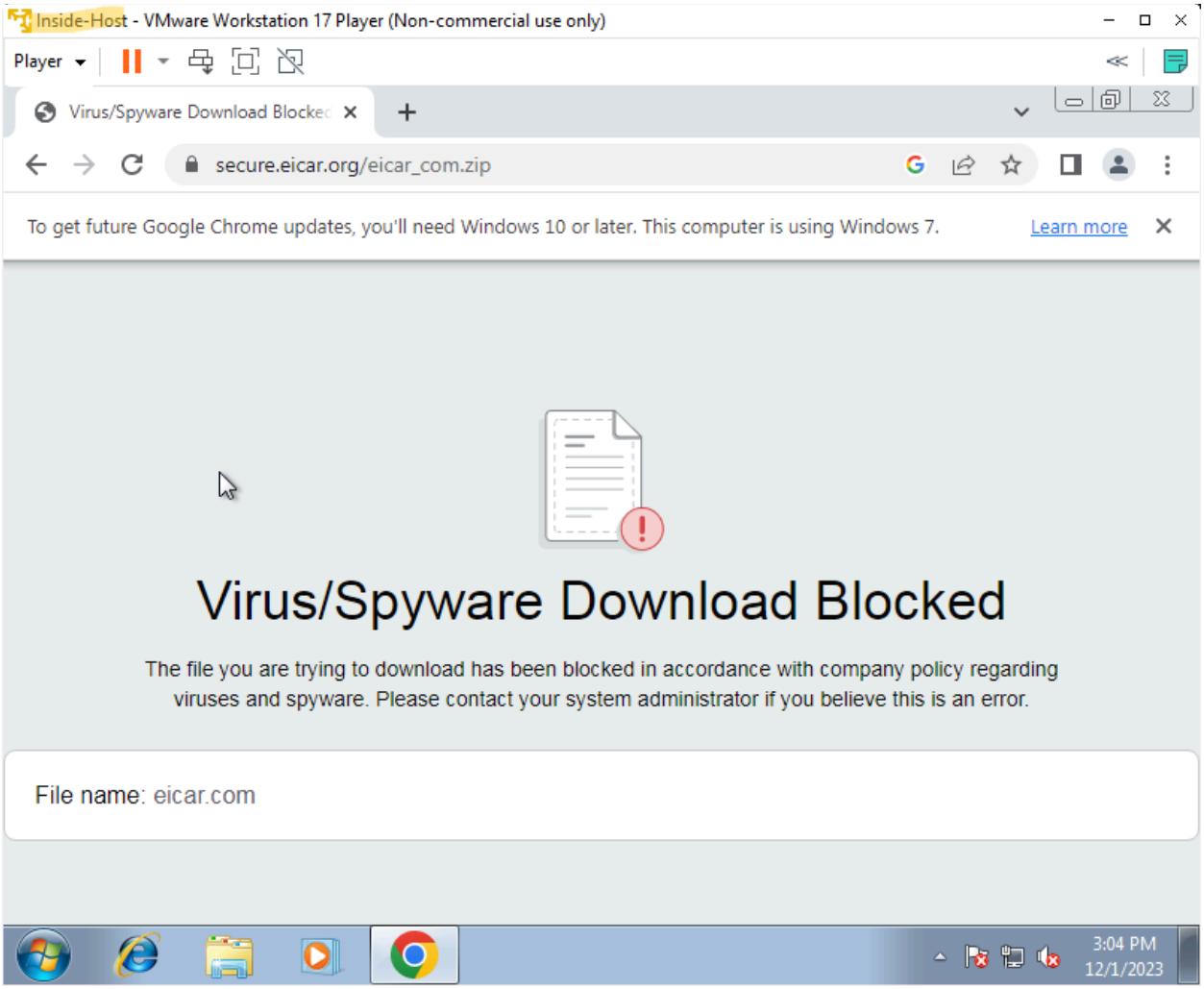

Figure 54: Eicar_com.zip file blocked on inside host.

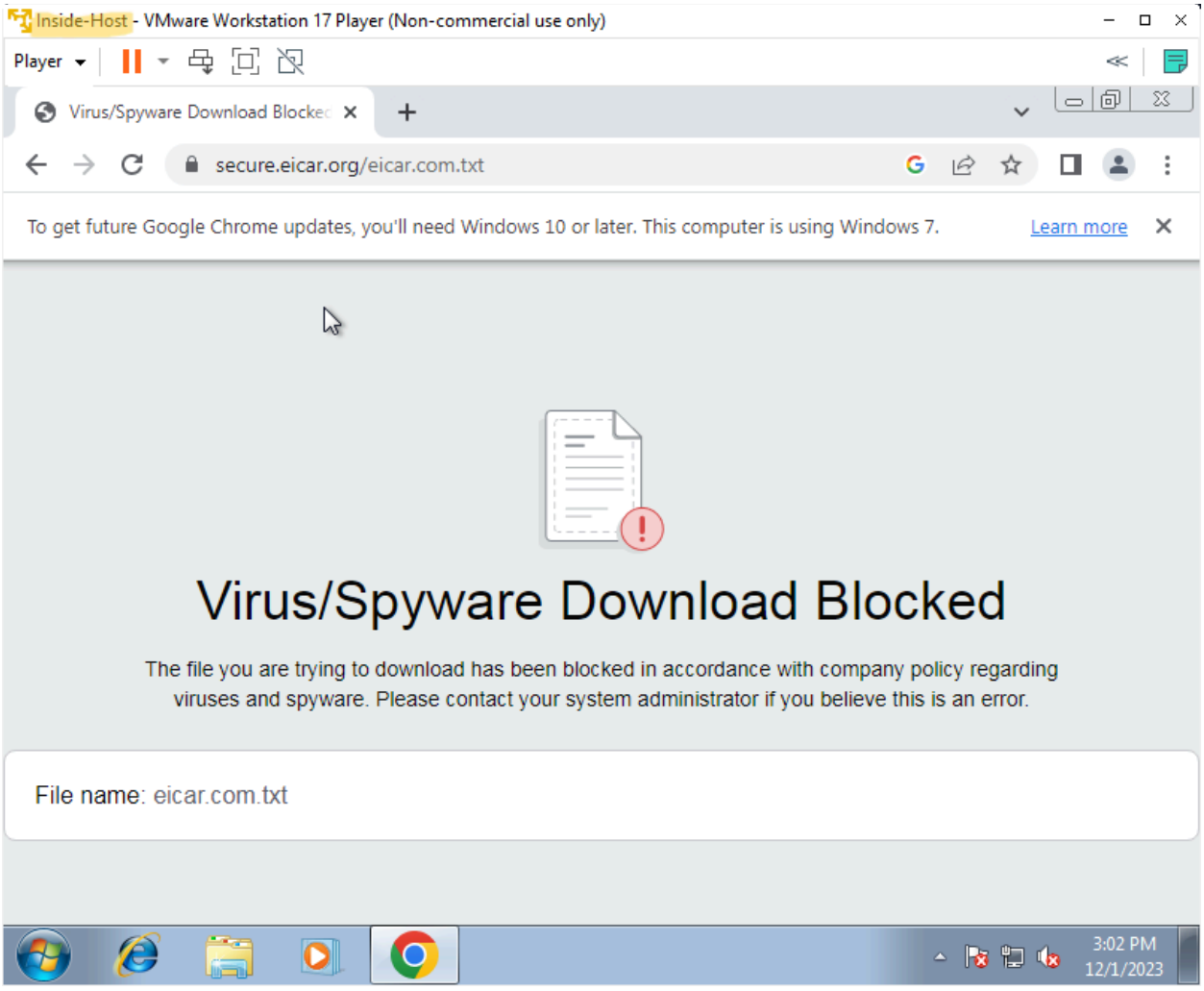

Figure 55: eicar.com.txt file blocked on inside host.

12. On DMZ-Host, run the TFTP Server on non-standard port "1069."

Luckily we had TFTP64 installed on the DMZ-Host, this allows us to run a TFTP server from a designated address 172.16.1.26. Even though this is the private IP of the DMZ-Host, because Kali is within the perimeter firewall, we set up on VMnet 4 and not linked to the outside network we can directly link to the private address.

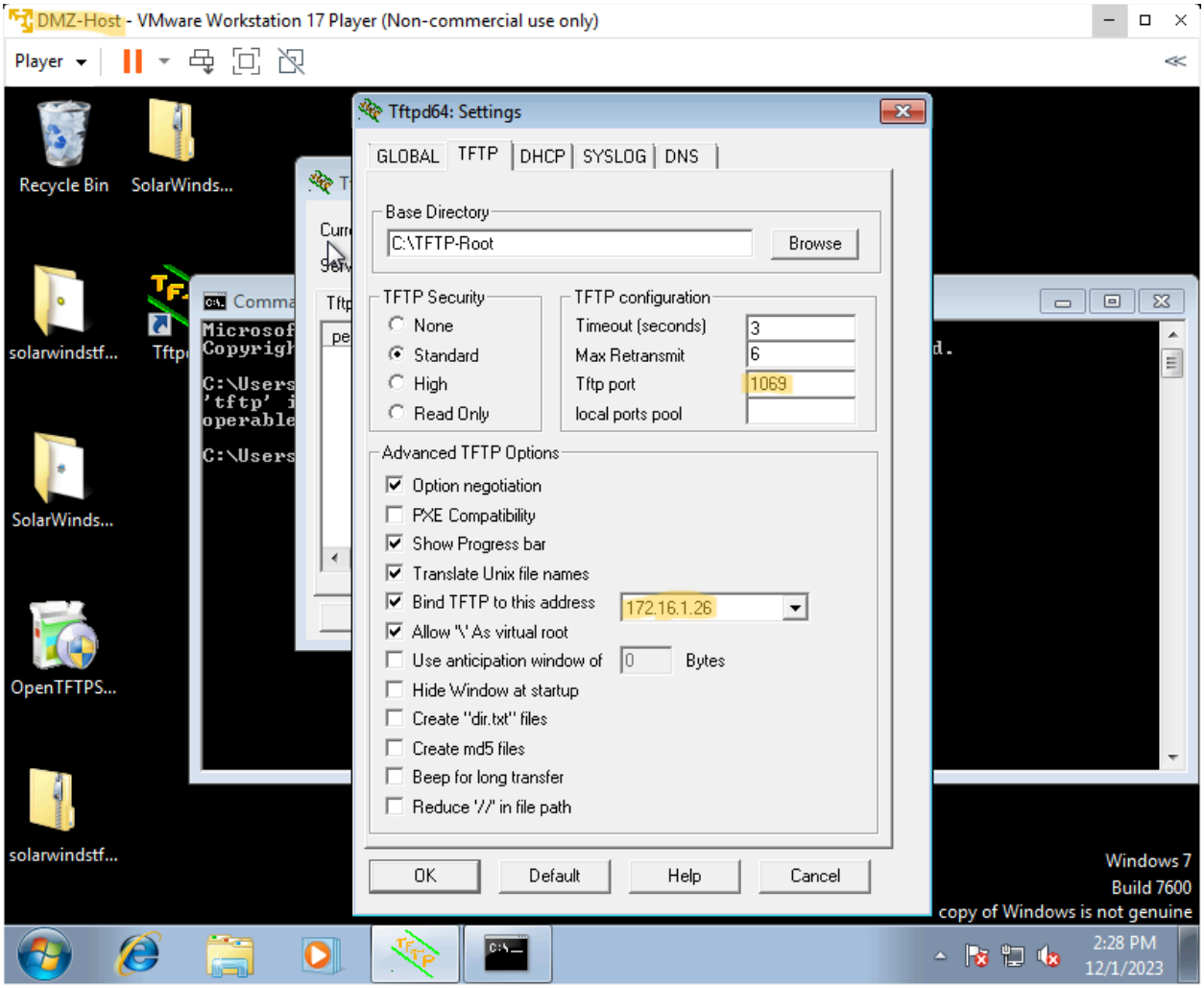

Figure 56: TFTP settings on DMZ-Host, binding it to the private IP and nonstandard Port.

13. Allow access from Kali-Linux to the TFTP Server using application awareness regardless of the port number.

My first impulse was to enable the TFTP application, which would immediately allow this transaction. However, this did not yield the proper results. After some research, I realized that this occurred due to the application-default service, which is enabled by default with application awareness. Application default states that it would expect TFTP from the standard port 69. Therefore switching this to any, instead allows for the TFTP to occur on non-standard ports.

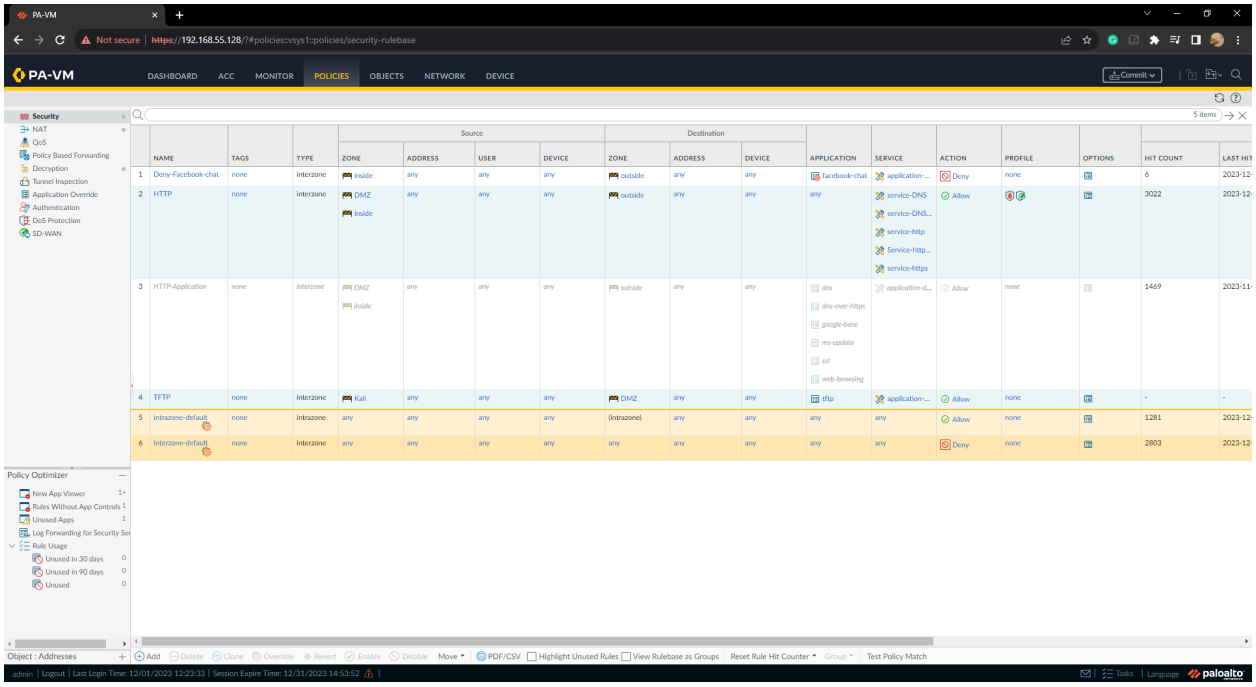

Figure 57: Enabling TFTP access rule which uses the TFTP application to enable traffic.

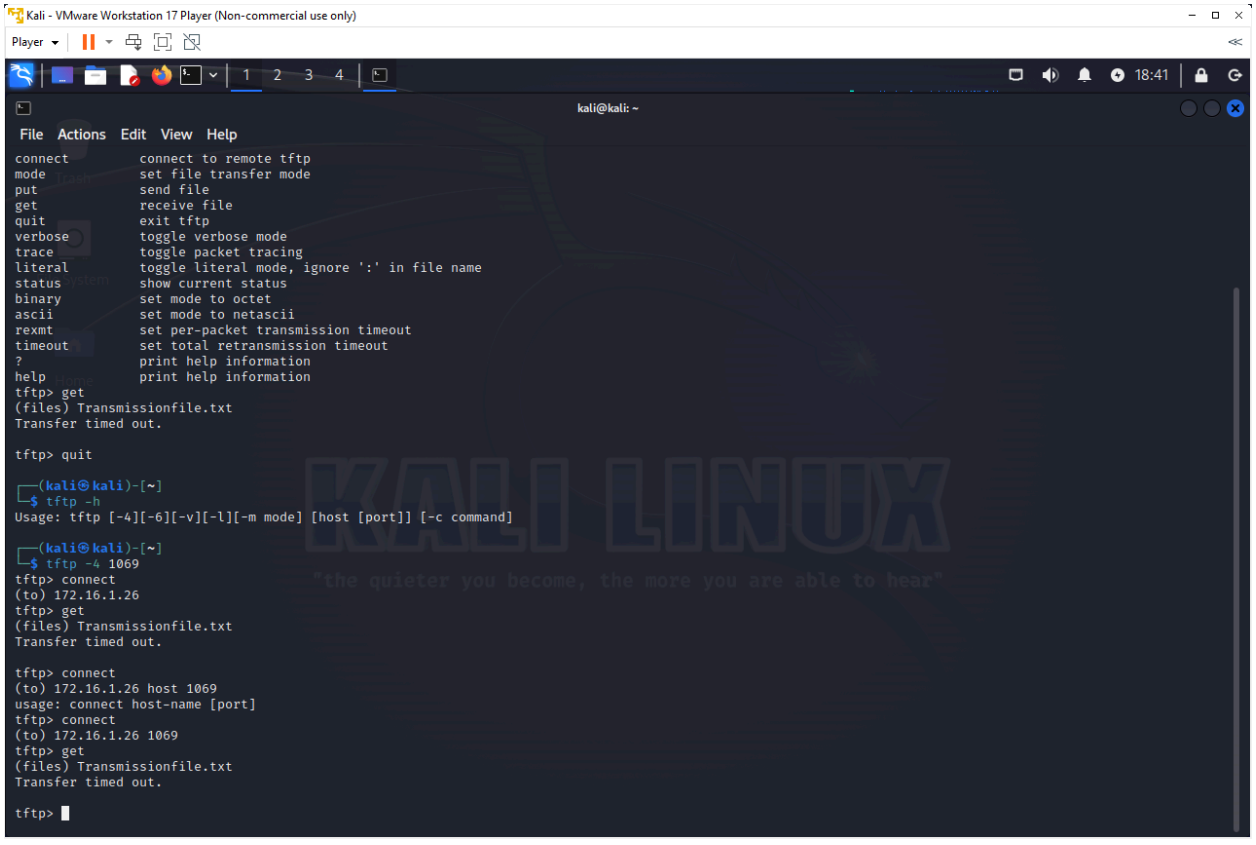

Figure 58: Kali failed to attempt to receive a file from TFTP

	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	ACTION	RULE	SESSION END REASON	BYTES	HTTP/2 CONNECTION SESSION ID	SDWAN SITE NAME	APP FLA COUNT
	12/01 15:40:35	drop	Kali	DMZ	192.168.146.26			172.16.1.26			1069	not-applicable	deny	interzone-default	policy-deny	0	0		0
	12/01 15:40:30	drop	Kali	DMZ	192.168.146.26			172.16.1.26			1069	not-applicable	deny	interzone-default	policy-deny	0	0		0

Figure 59: Log displaying this failed attempt.

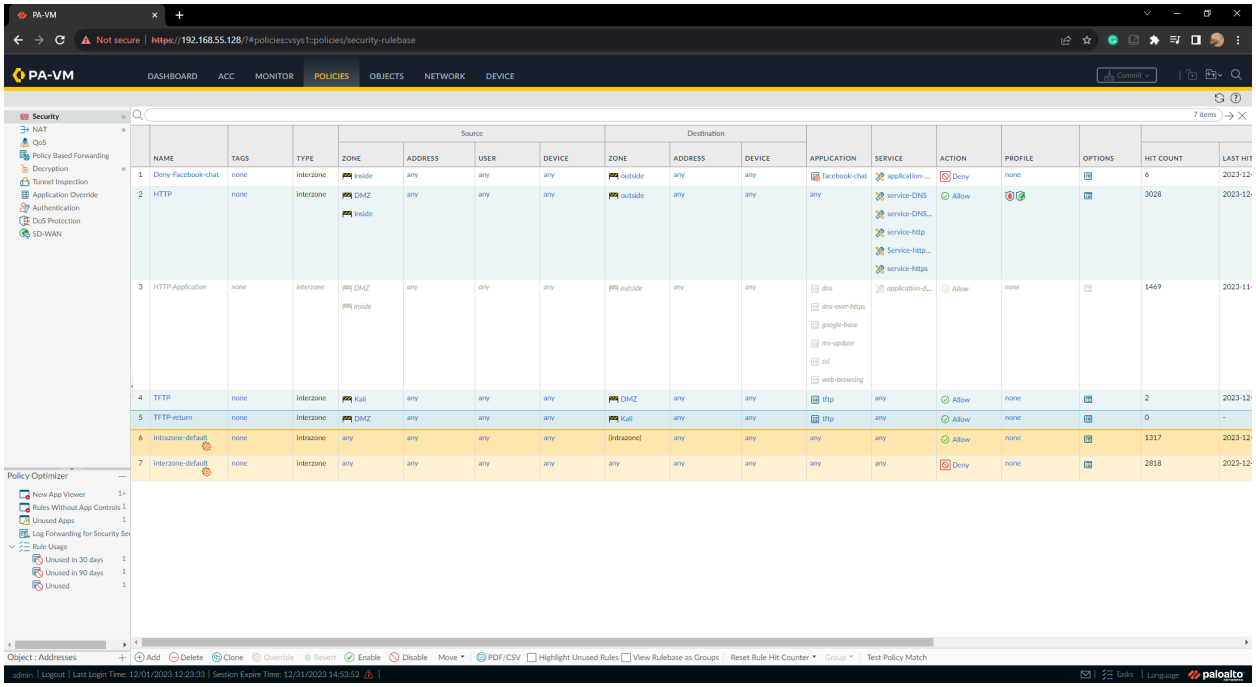

Figure 60: Changed TFTP protocol to change service to any (TFTP return not required).

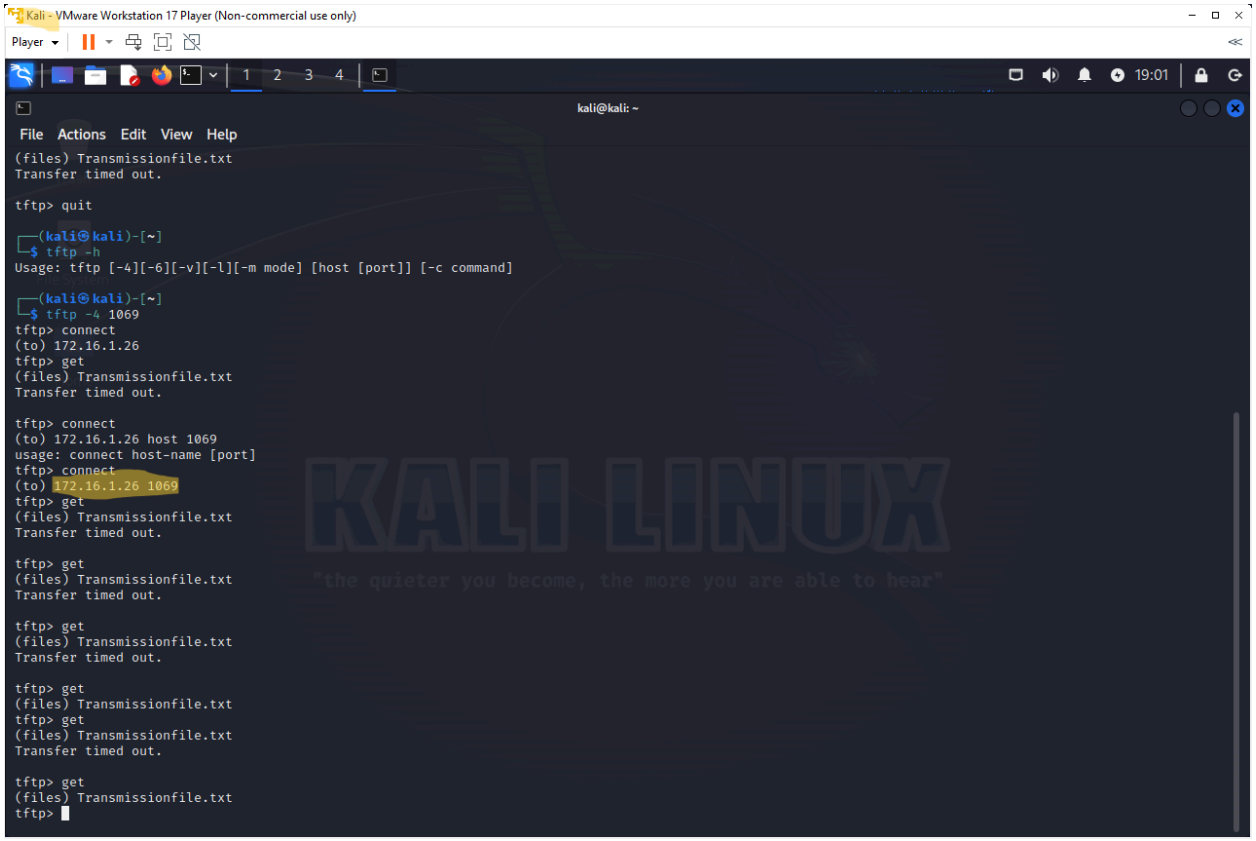

Figure 61: Successful TFTP file transfer.

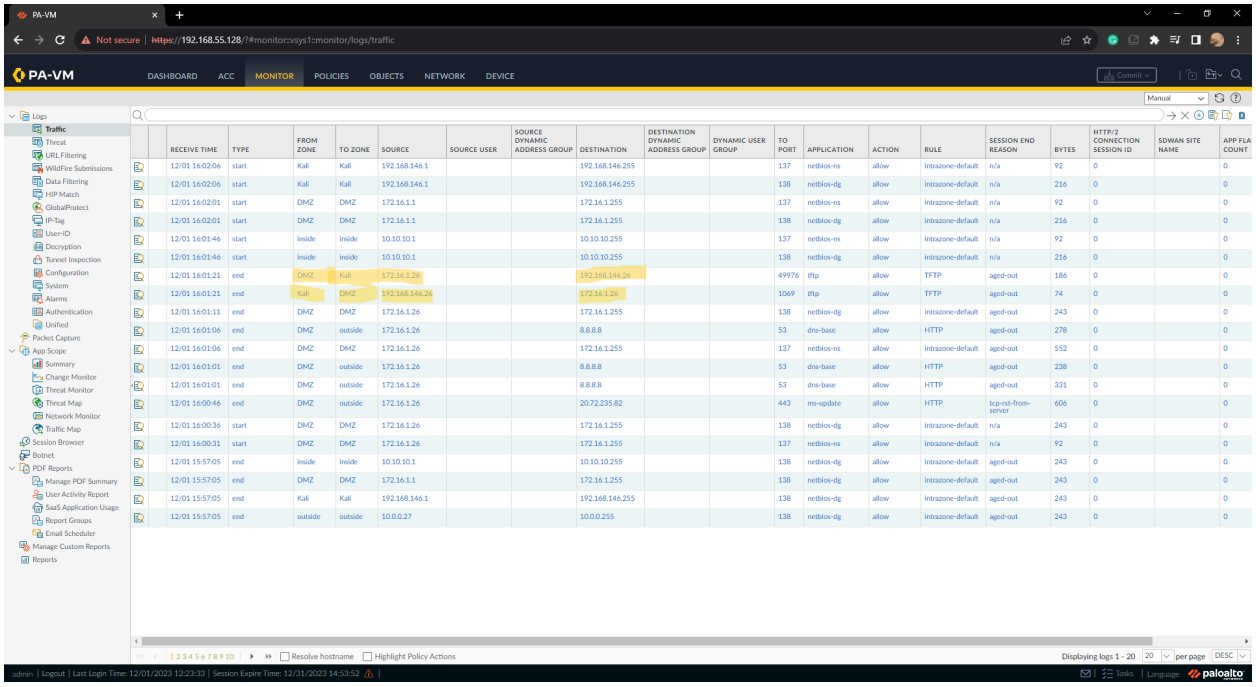

Figure 62: Log showing this successful TFTP file transfer.

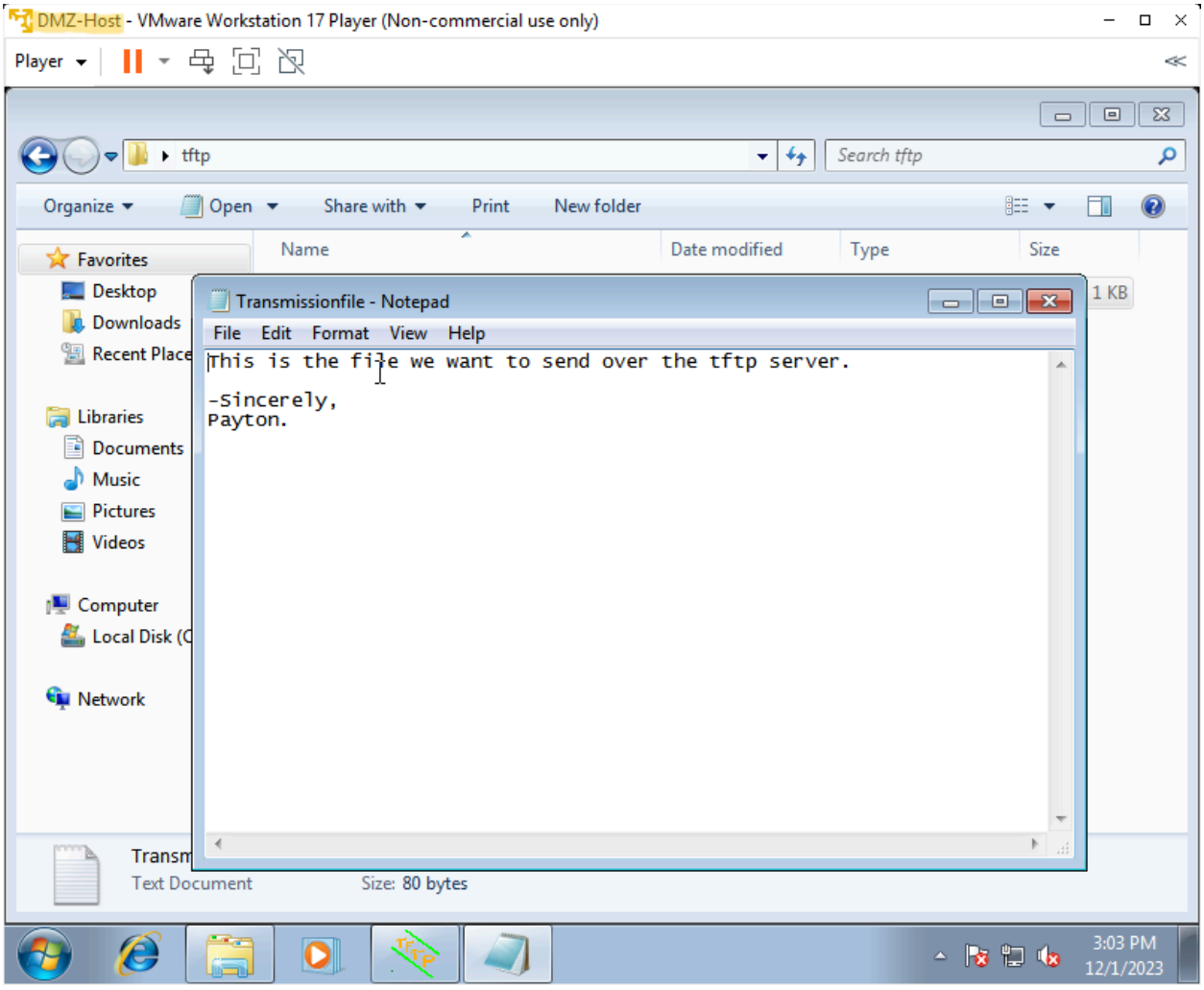

Figure 63: Transmission file to be transferred between server and client.

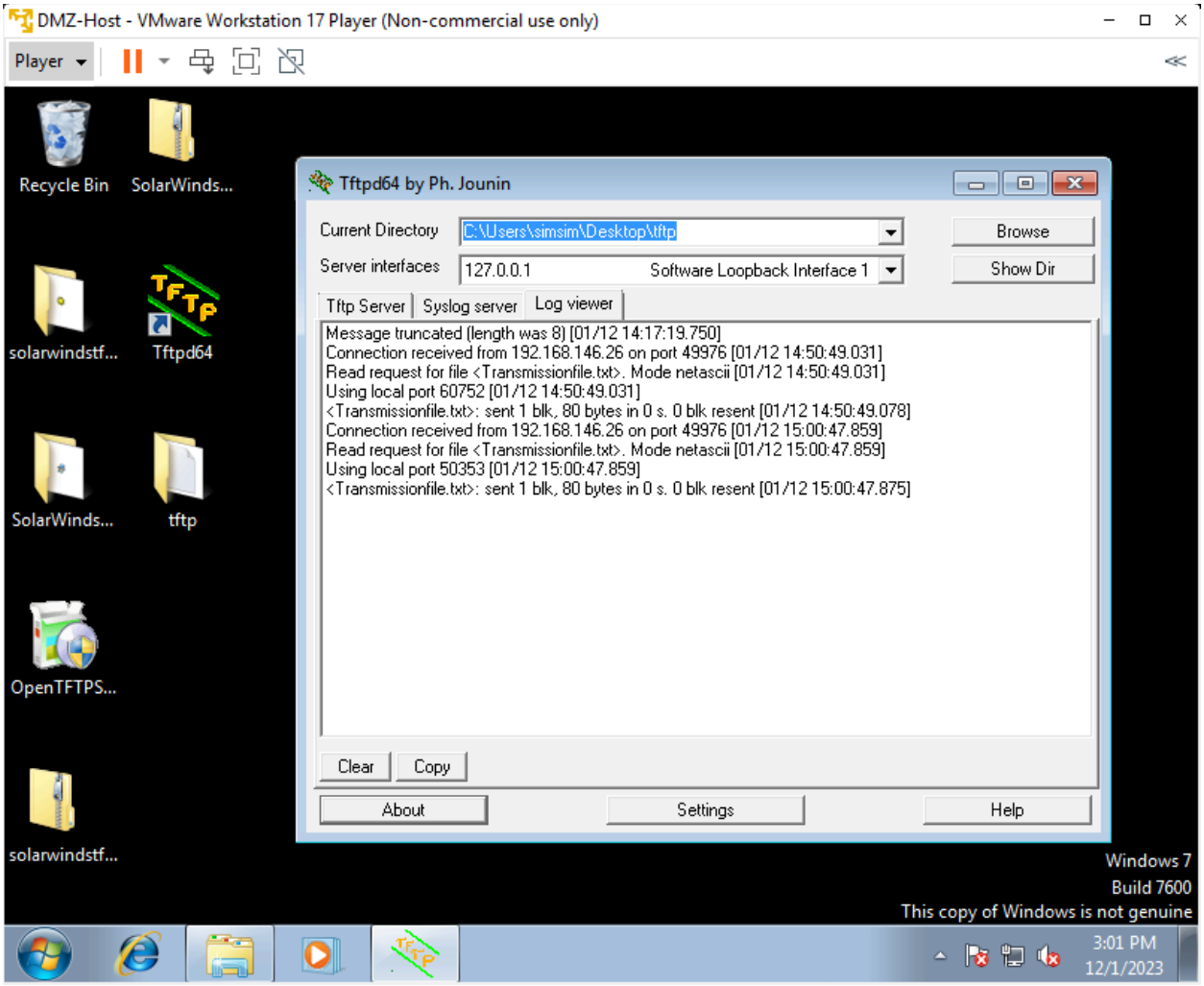

Figure 64: Log on TFTP64 DMZ-Host showing file transfer.

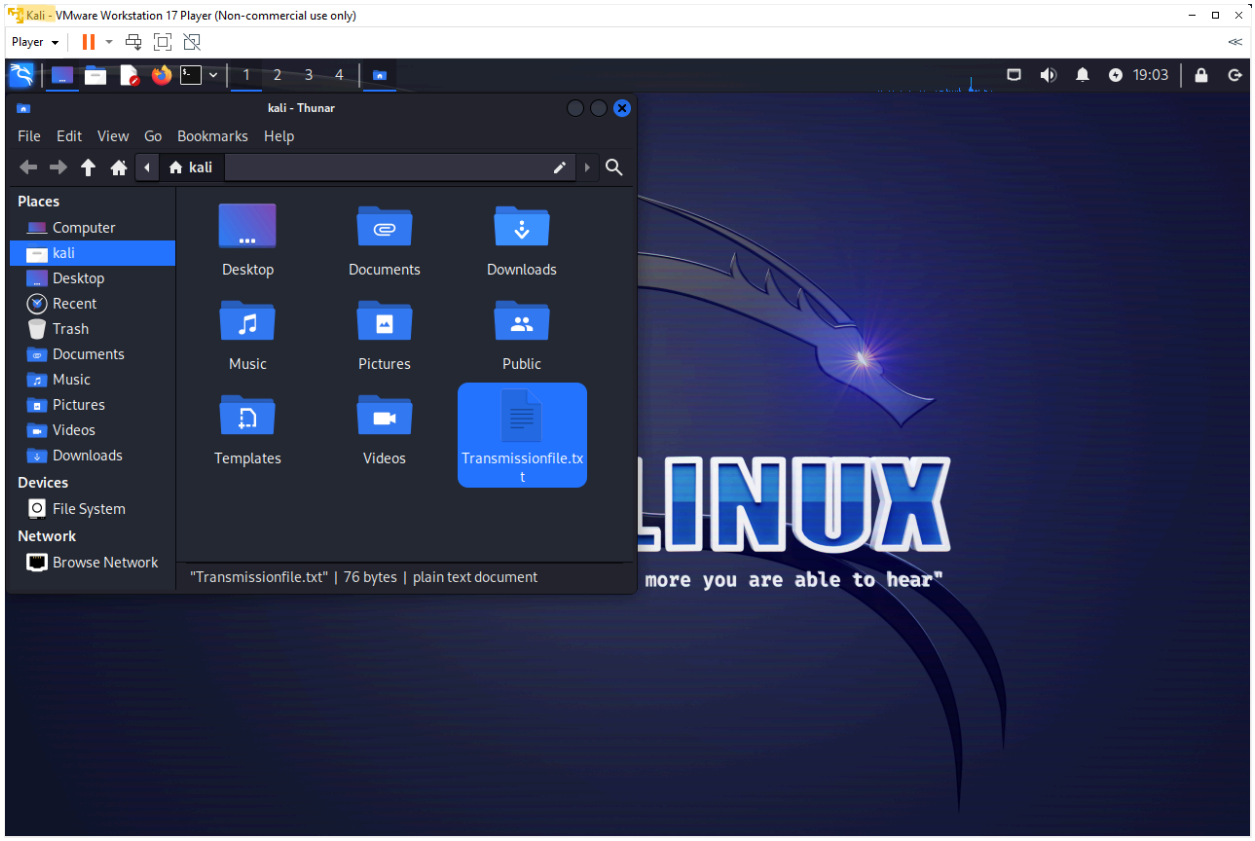

Figure 65: Transmissionfile.txt successfully loaded to Kali system.

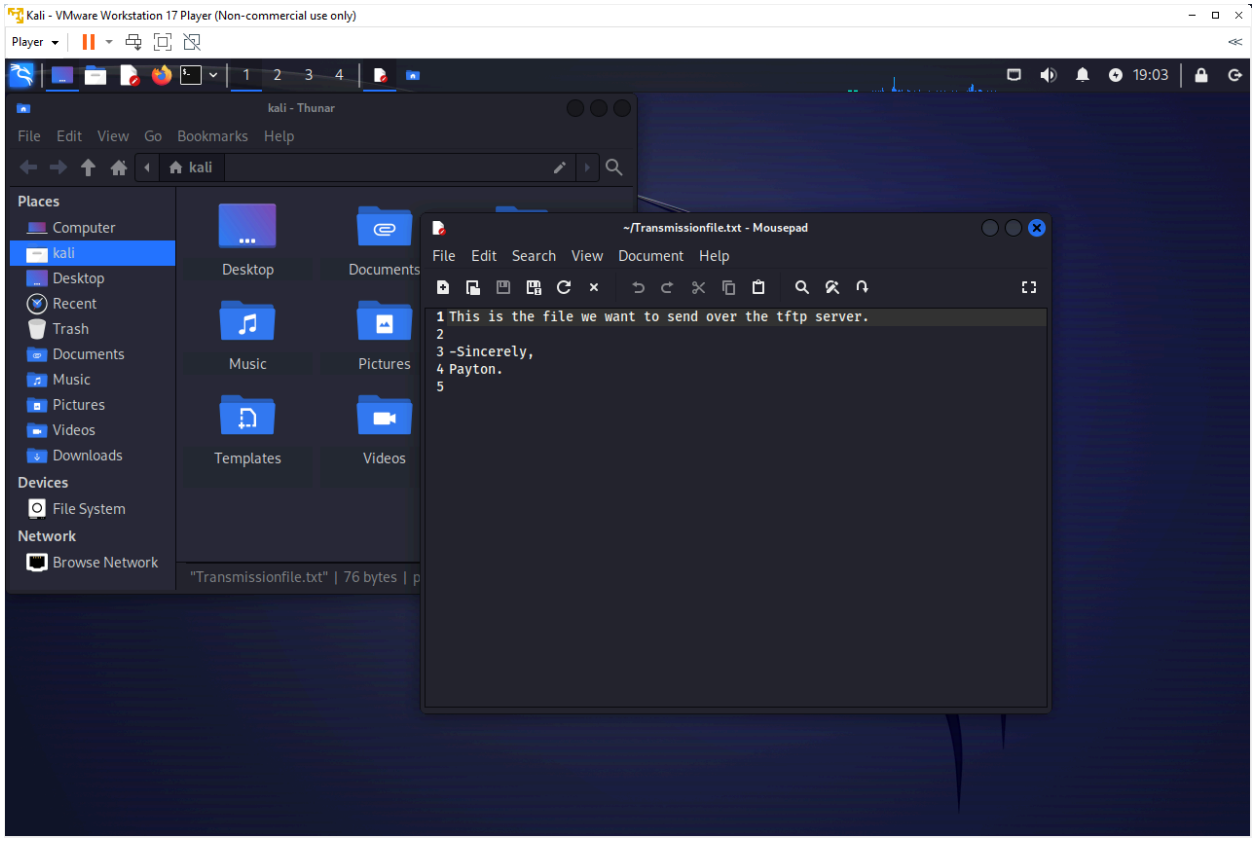

Figure 66: Content of Transmissionfile.txt on Kali.

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

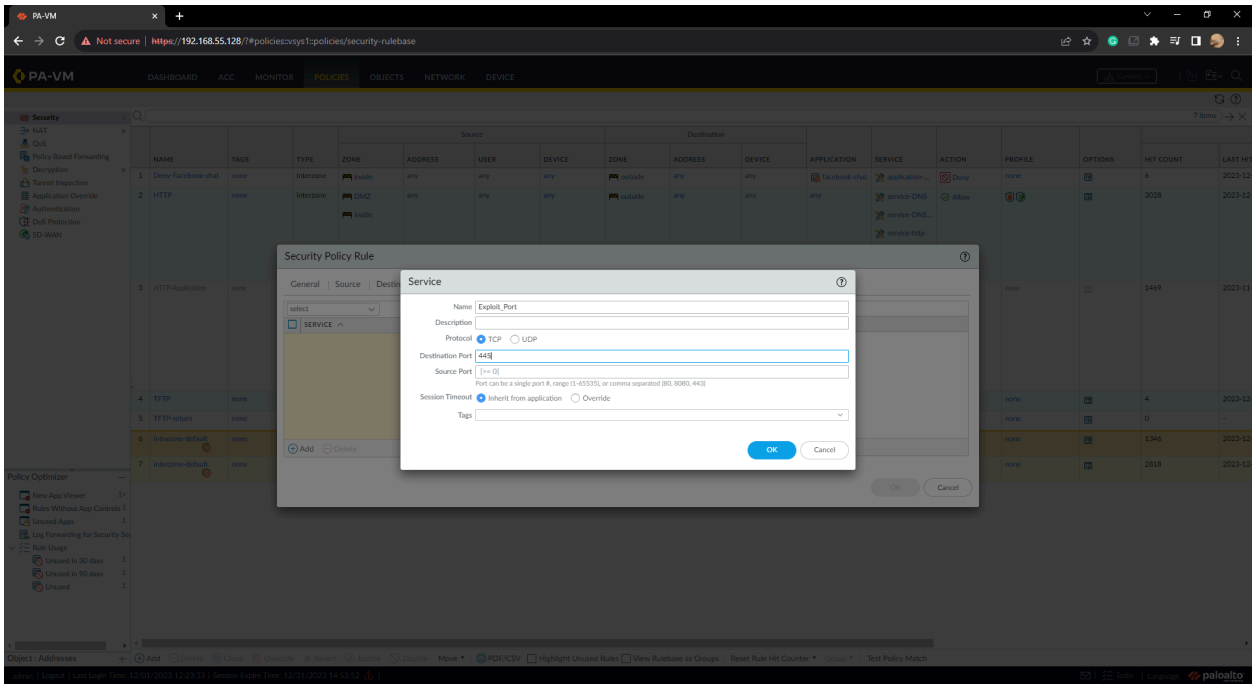

Figure 67: Configuring policy access rule for exploit port tcp 445.

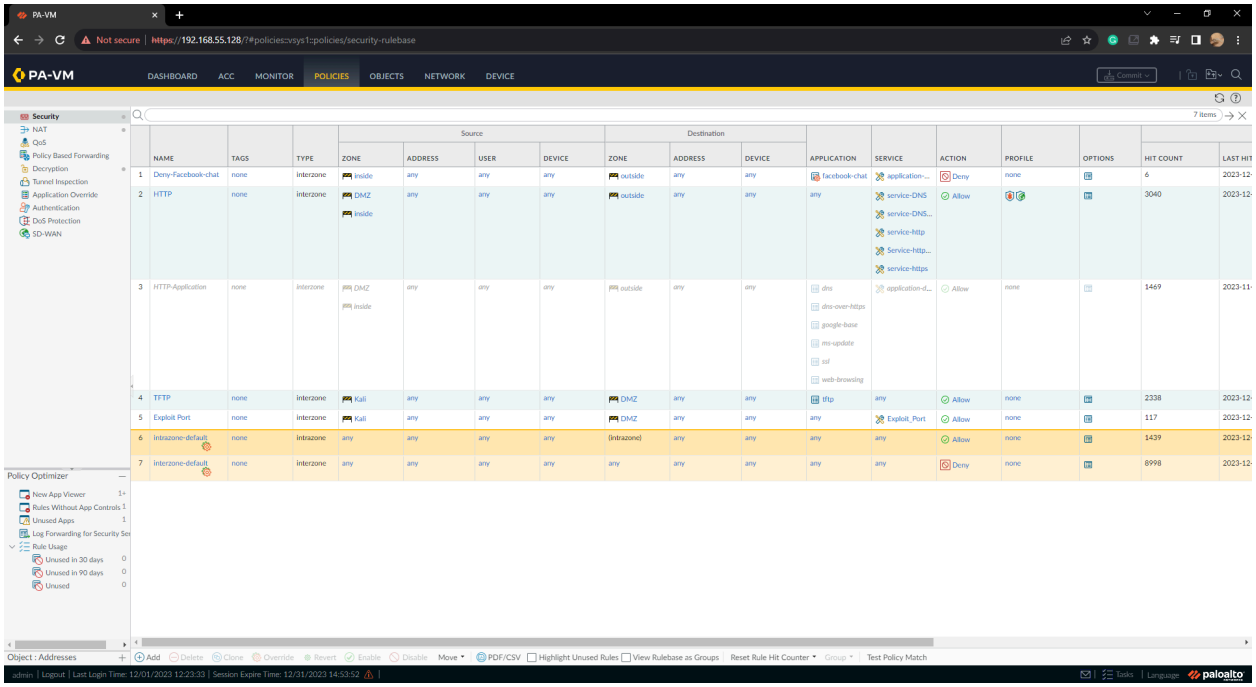

Figure 68: Exploit Port enabled between Kali and DMZ.

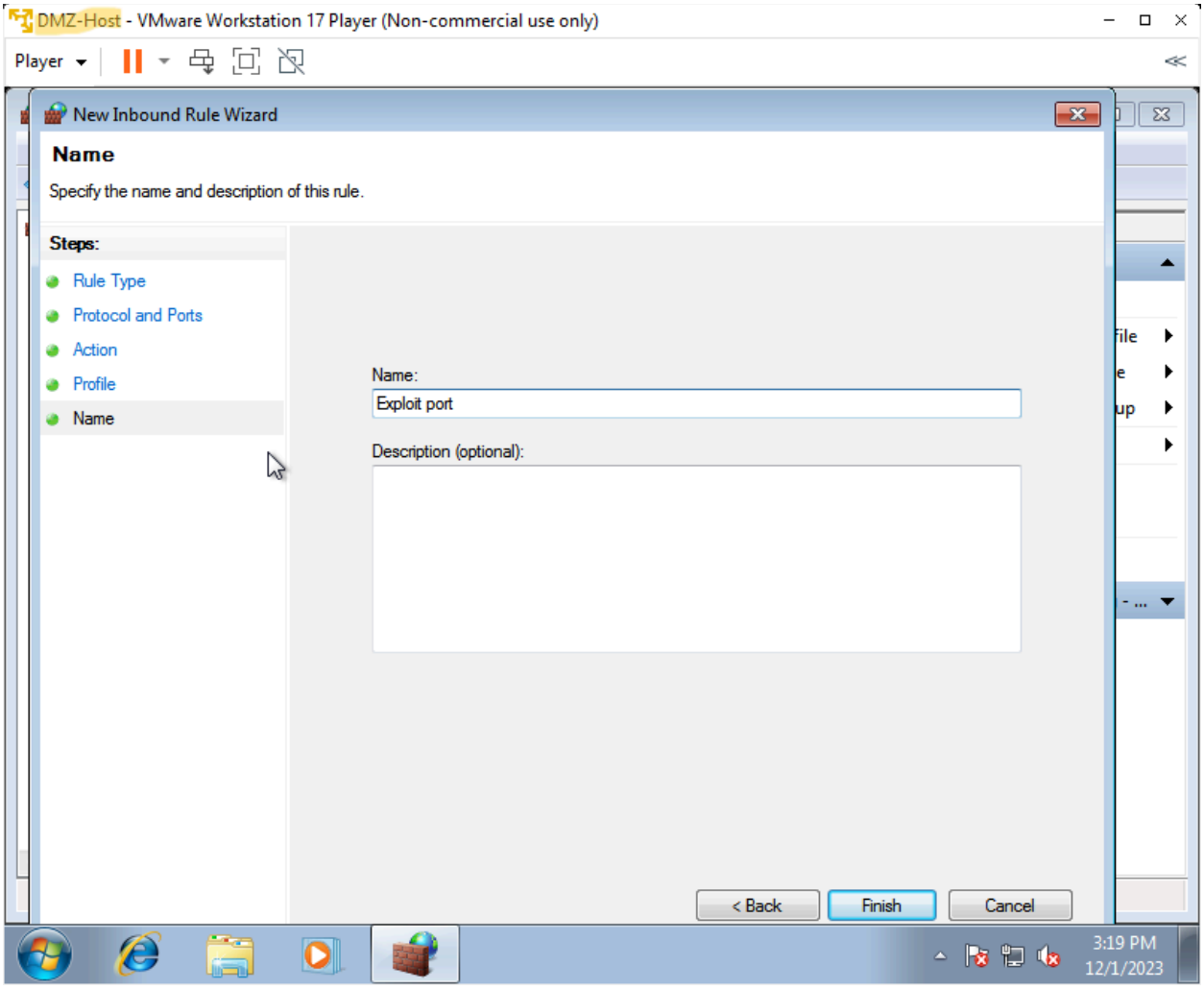

Figure 69: Configuring new inbound access rule in firewall allowing for port 445 access.

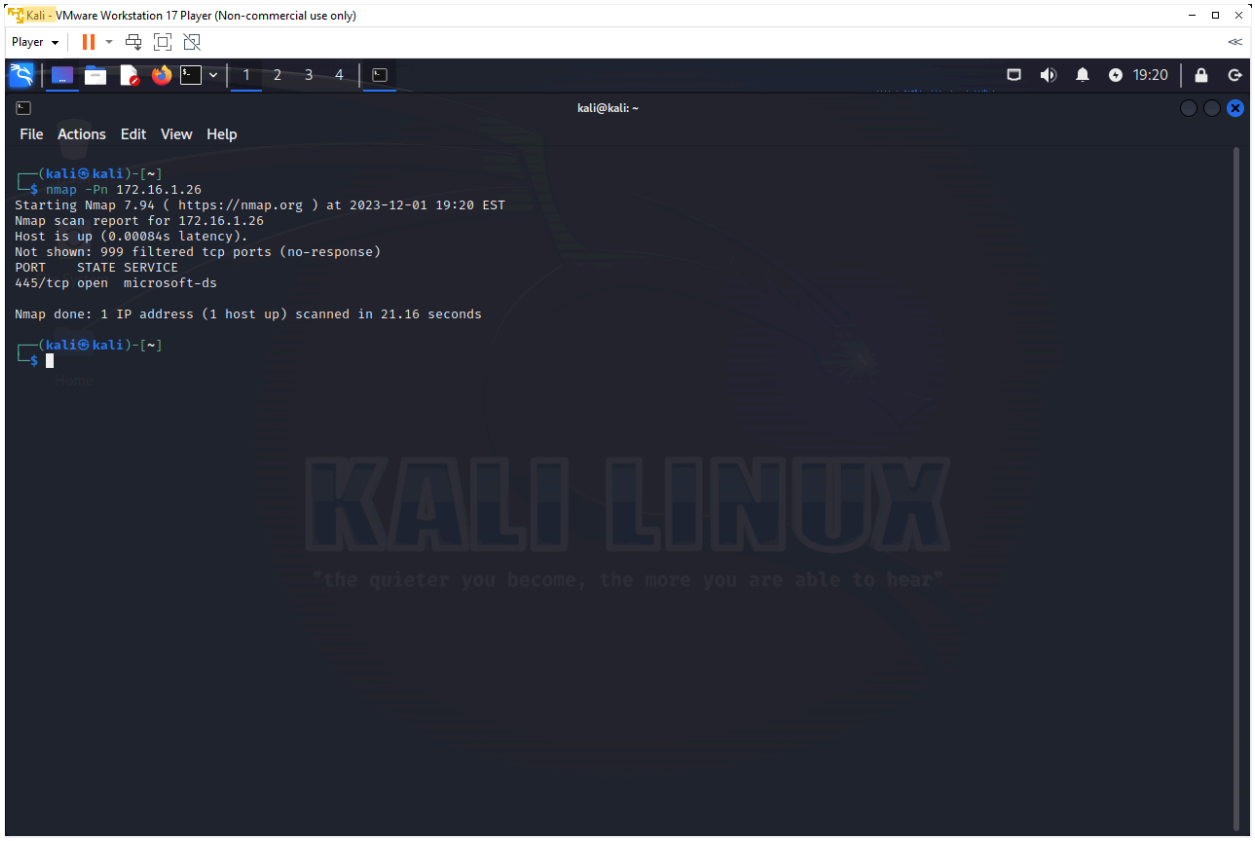

Figure 70: Nmap on Kali linux showing Port open.

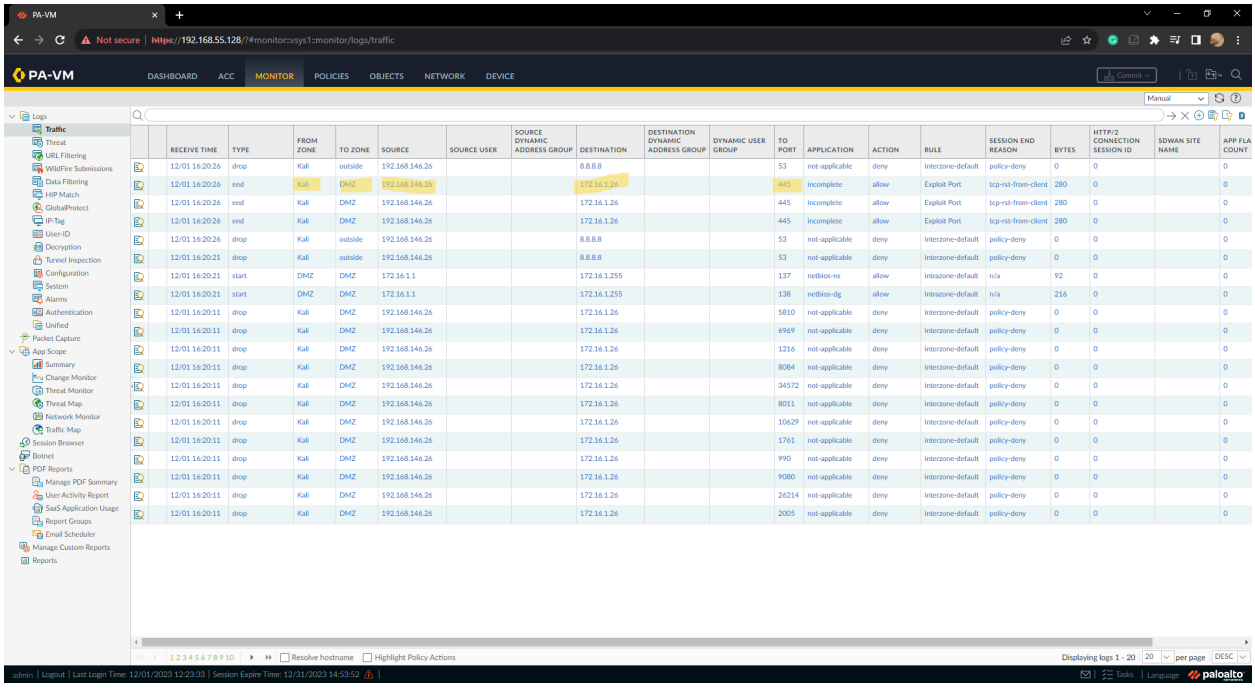

Figure 71: Log showing Kali Linux access to port 445 (TFTP rule disabled so non-standard ports denied are denied access)

15. Use the Metasploit framework on Kali Linux to demonstrate that DMZ-Host is vulnerable to MS17-010.

Similar to what we did in Assignment 2, we configured Metasploit eternal blue scanner to 172.16.1.26 and ran a scan to see if the host is vulnerable.

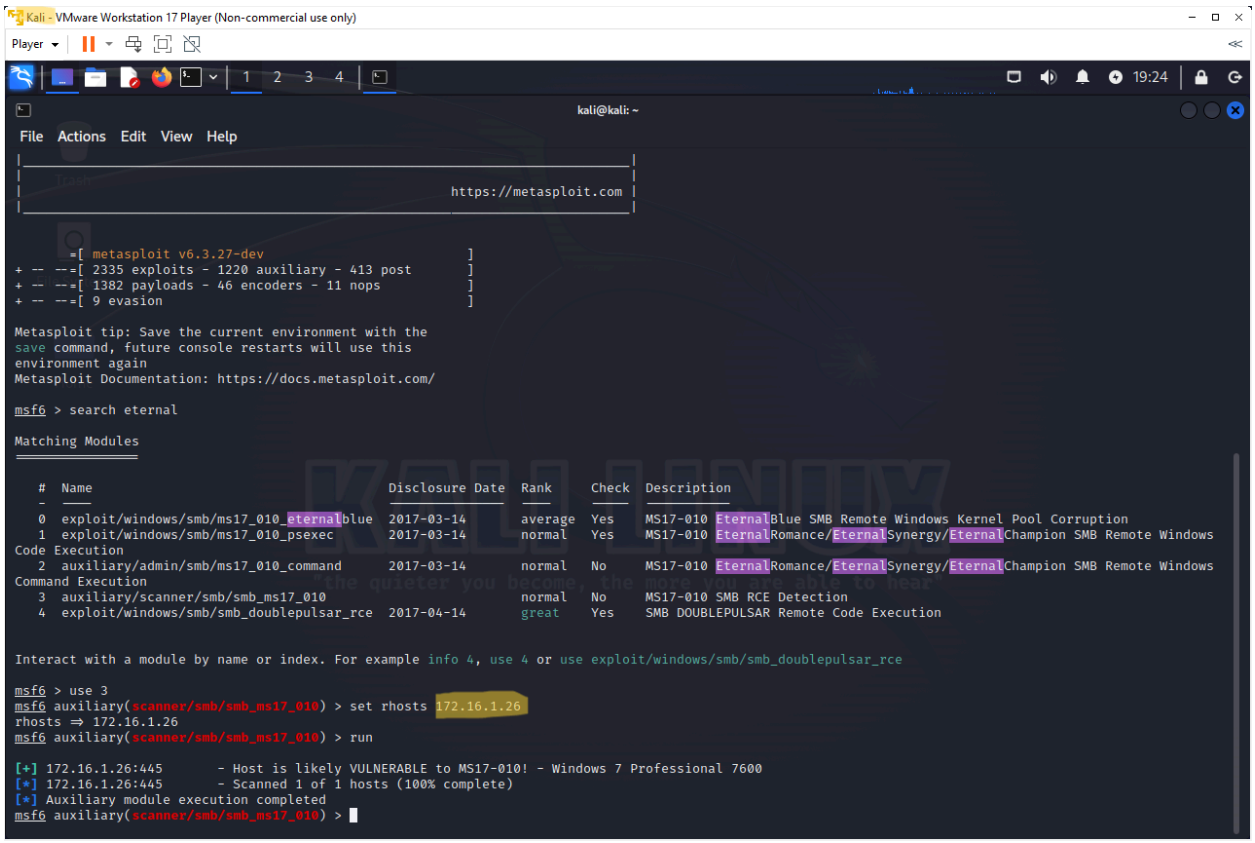

Figure 72: Kali Linux executing the scanner determining that the host is likely vulnerable.

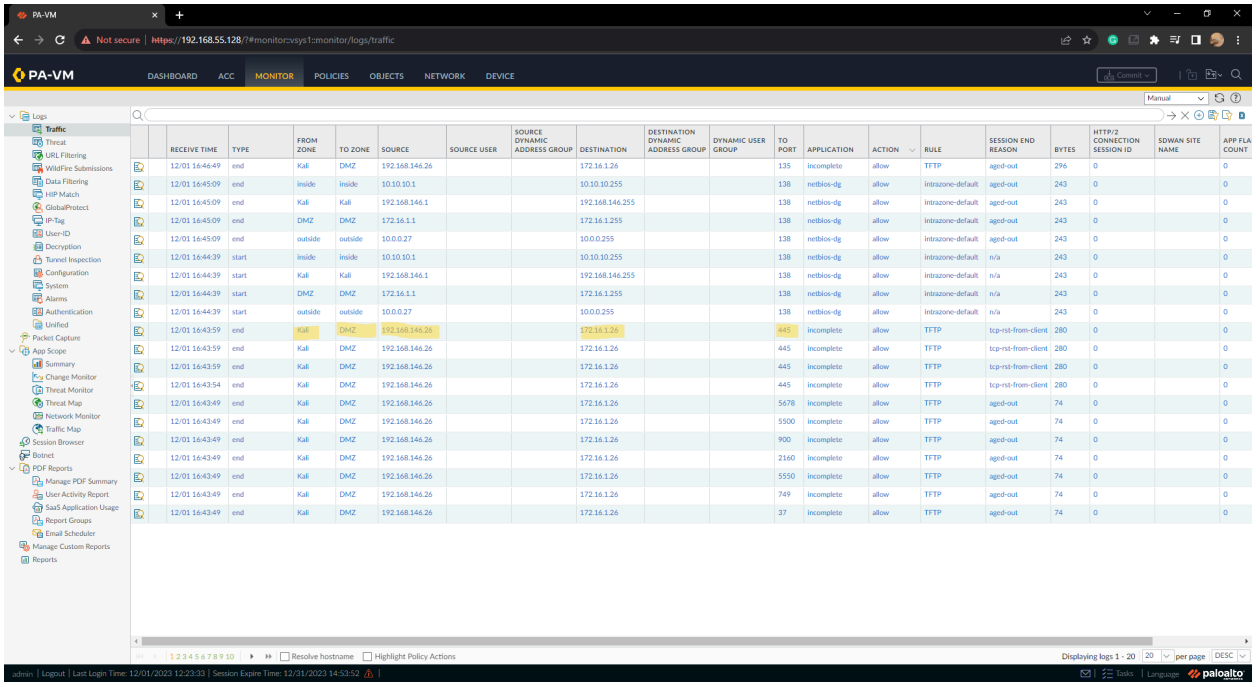

Figure 73: Log showing Kali Linux access to port 445 during scanner (TFTP access rule was reenabled in this step).

16. Exploit the MS17-010 vulnerability on DMZ-Host using default Meterpreter payload (reverse TCP) from Kali Linux. Was the attack successful? Why?

This attack was not successful by default. Meterpreter was able to drop the payload into the DMZ-Host, indicating some level of success. However, even with the TFTP port rule in place, for some reason, port 4444 return traffic was blocked by the firewall, and thus the connection could not be firmly established. As such, the msfconsole deemed the attack a failure.

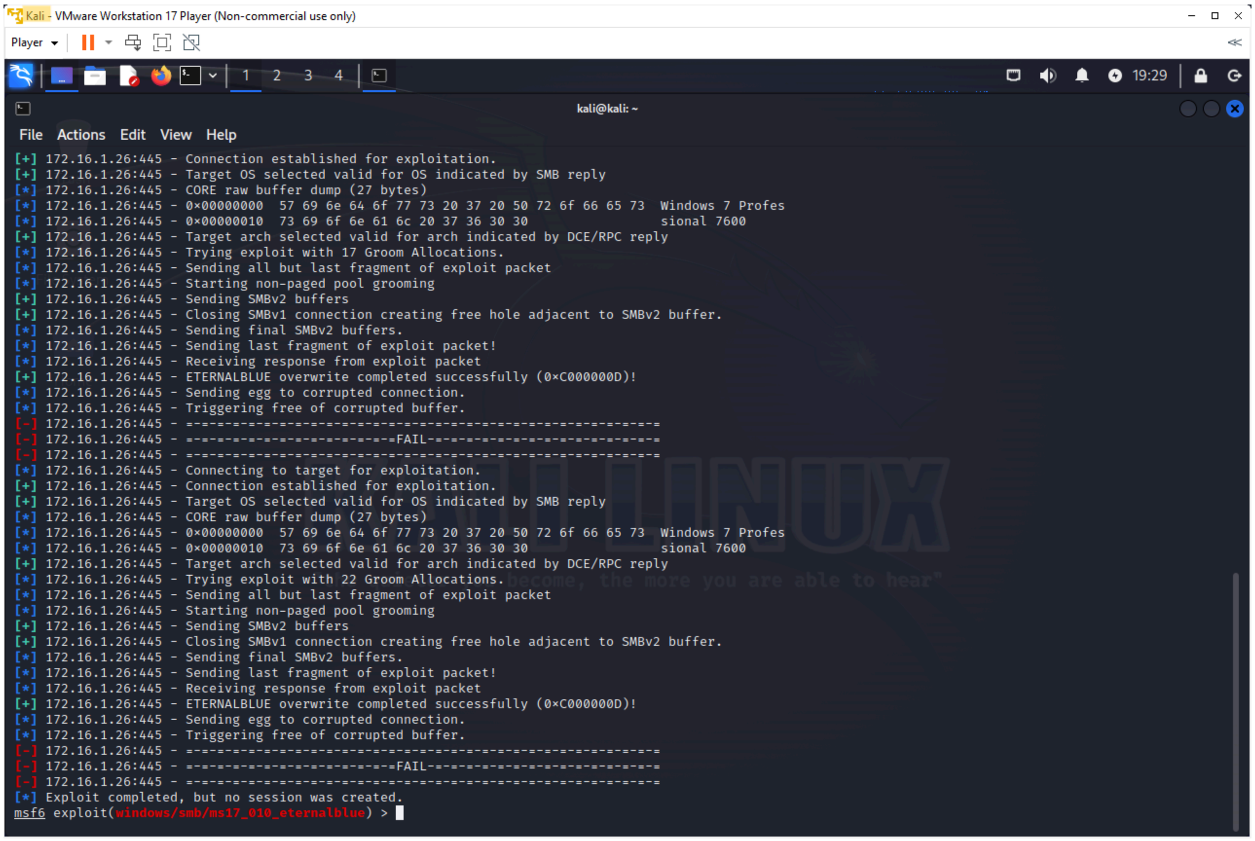

Figure 74: Kali msfconsole showing failed attack attempt.

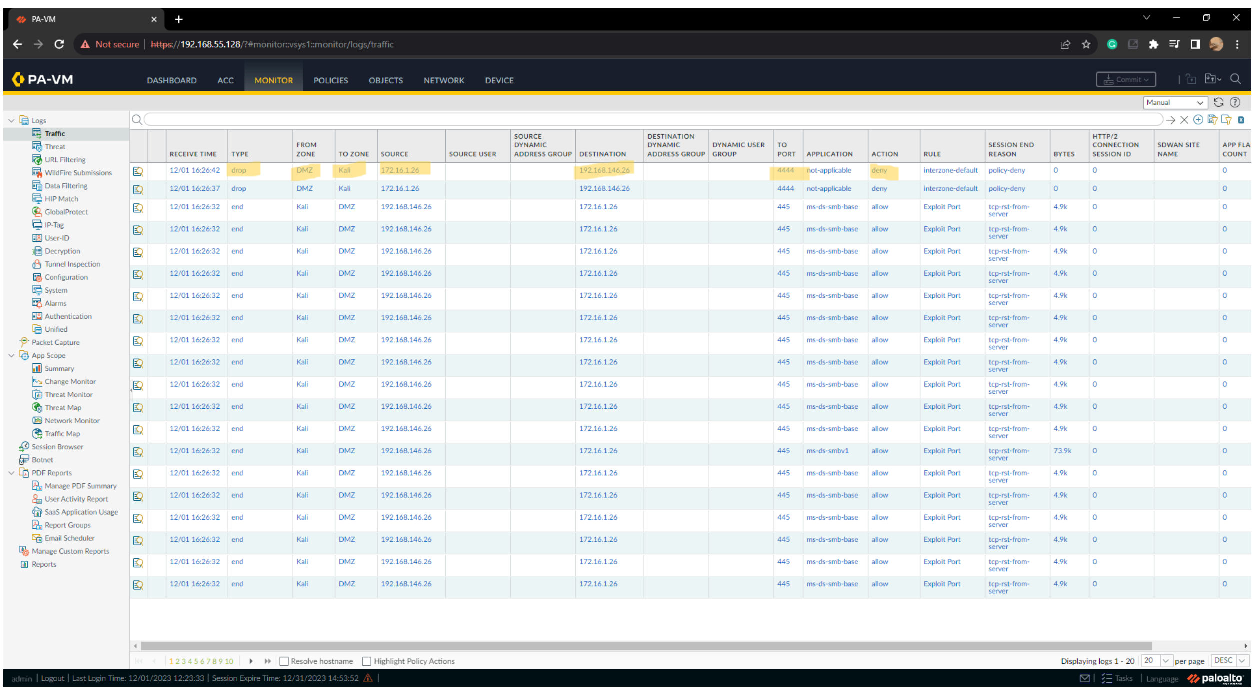

Figure 75: Log showing despite access to port 445, the firewall denied port 4444.

17. Assess the attack's success and perform required tasks if unsuccessful.

As stated before, the Kali Linux machine was able to drop the payload into the System files of the target computer. As such, this attack did have some success. If the attack had been simple malware or ransomware, it would have been entirely successful over this port and could have overridden system files. However, this exploit establishes the reverse TCP connection over an open port 4444. The firewall initially blocks this port. This is because port 4444 in the firewall logs is designated an application labelled “unknown port” as the firewall is unfamiliar with the application attempting to be executed, such that it is denied by the default Zero Trust rule. Therefore, in order to make this task successful, we must open port 4444 within the firewall access rules.

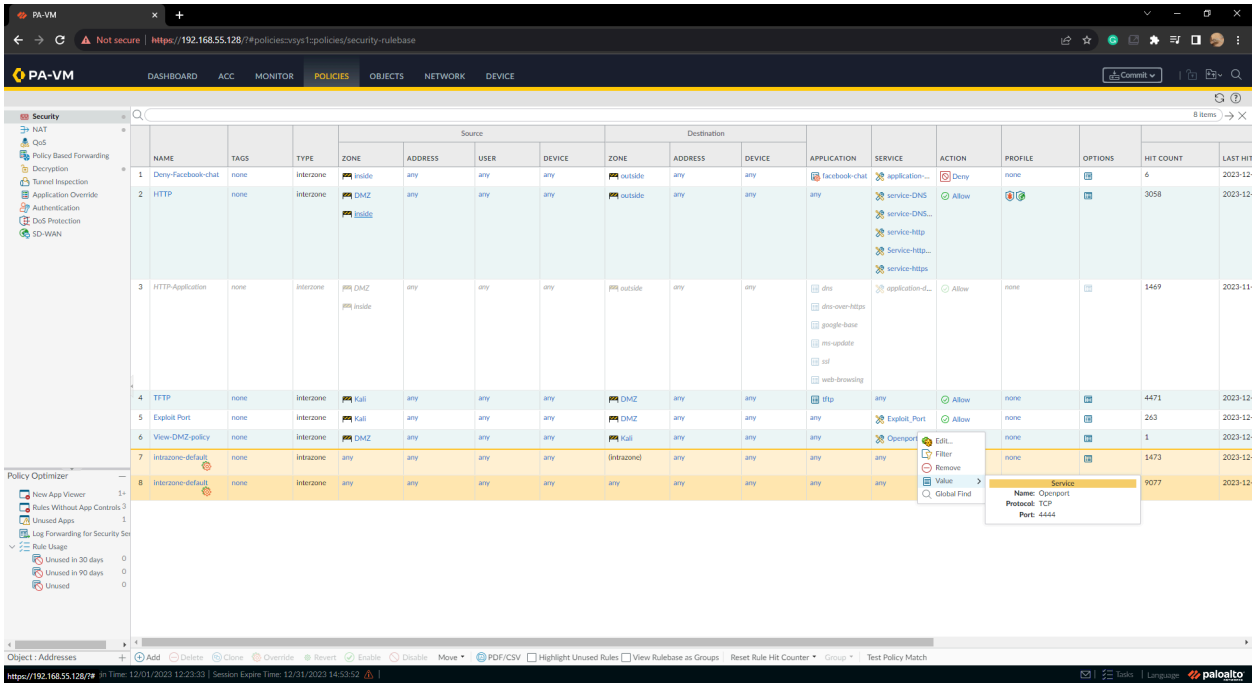

Figure 76: Allowing for TCP port 4444 to be opened.

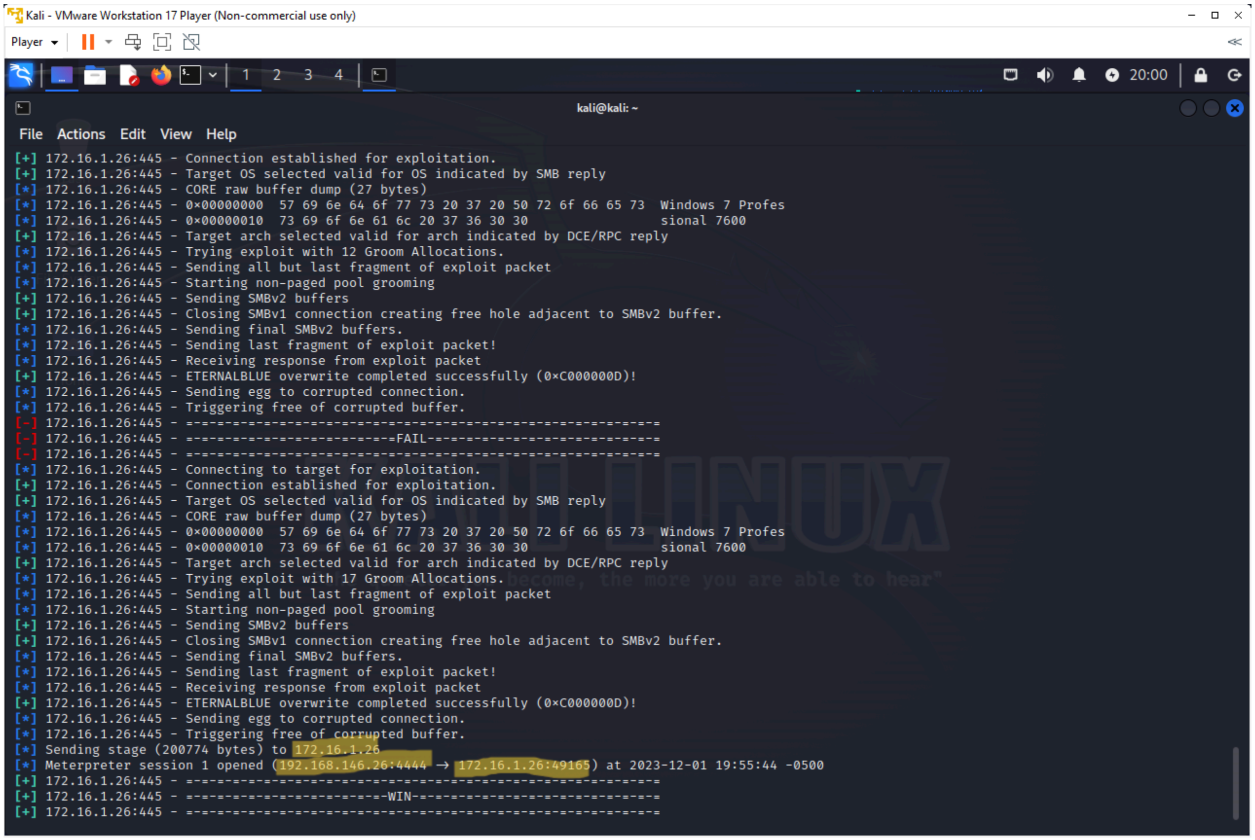

Figure 77: Attack deemed successful, and DMZ-Host becomes a zombie system.

18. Block the applications used in the previous attack and show that the attack was prevented even if port 445 is still open.

Examining the logs provided by the monitor tab, we can see that ms-ds-smb is the application the exploit uses. Thus, by blocking this specific application, we can see that even with the port still open, the exploit can no longer run and will not even pass the scanner phase to attempt the exploit.

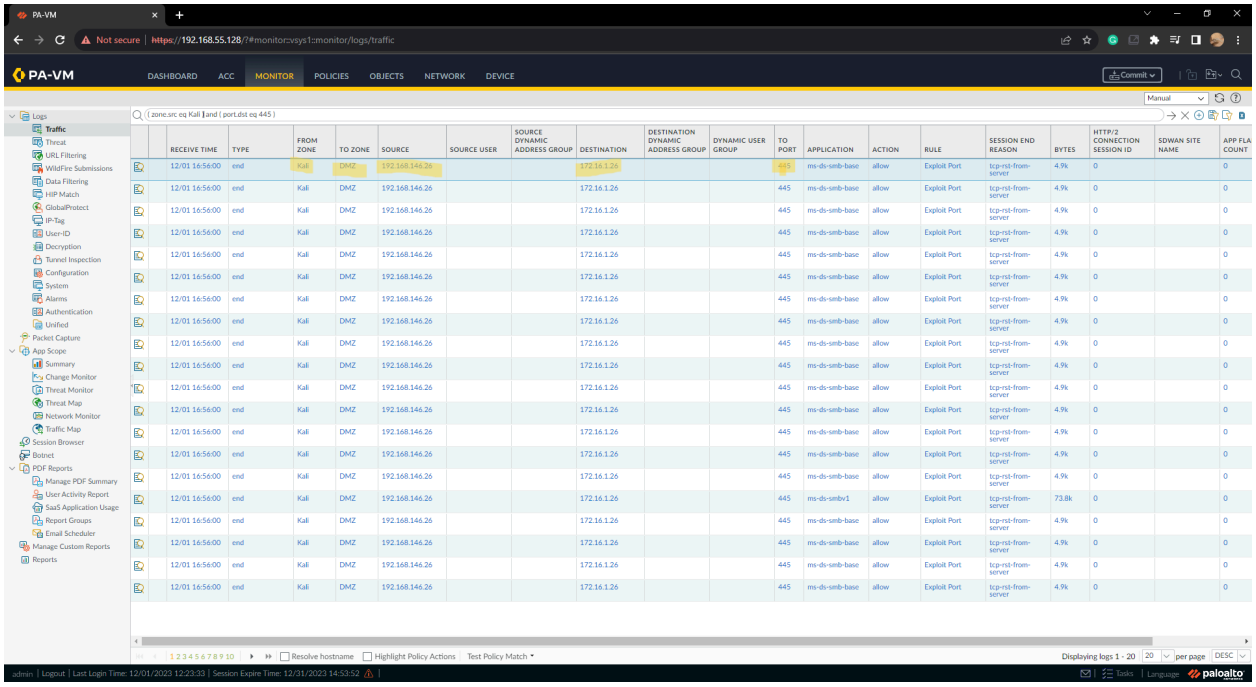

Figure 78: Log showing the applications allowed during Exploit.

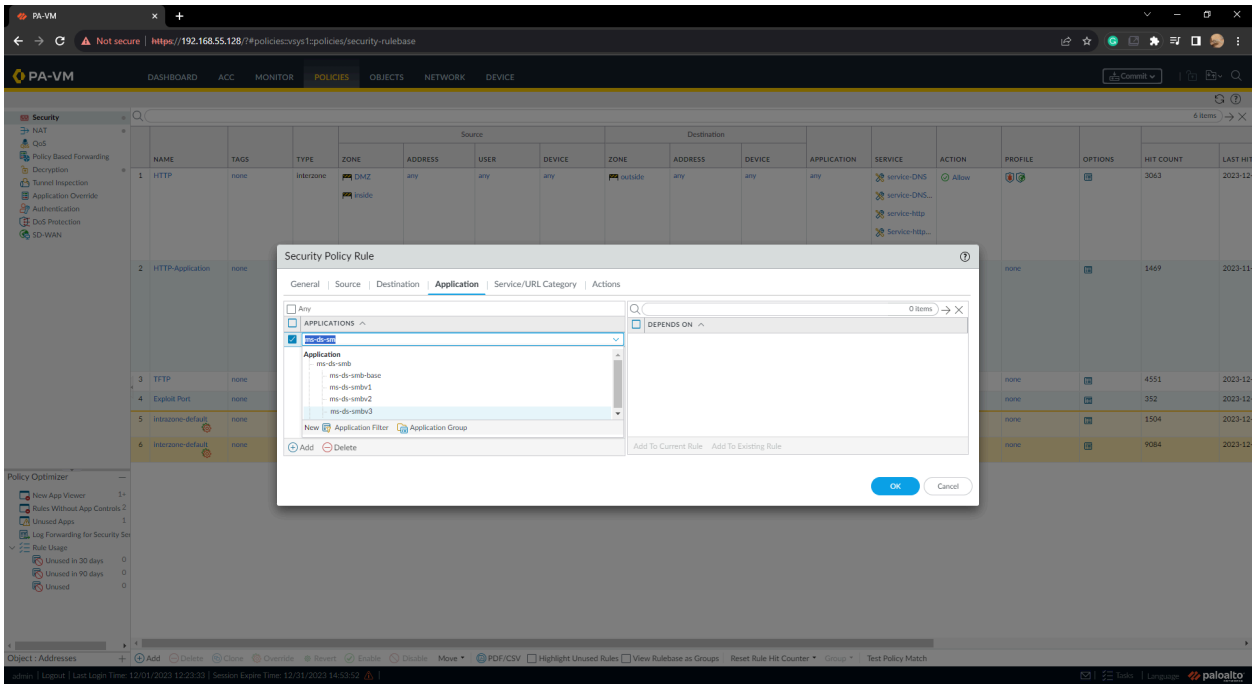

Figure 79: Configuring security policy rule with ms-ds-smb applications blocked.

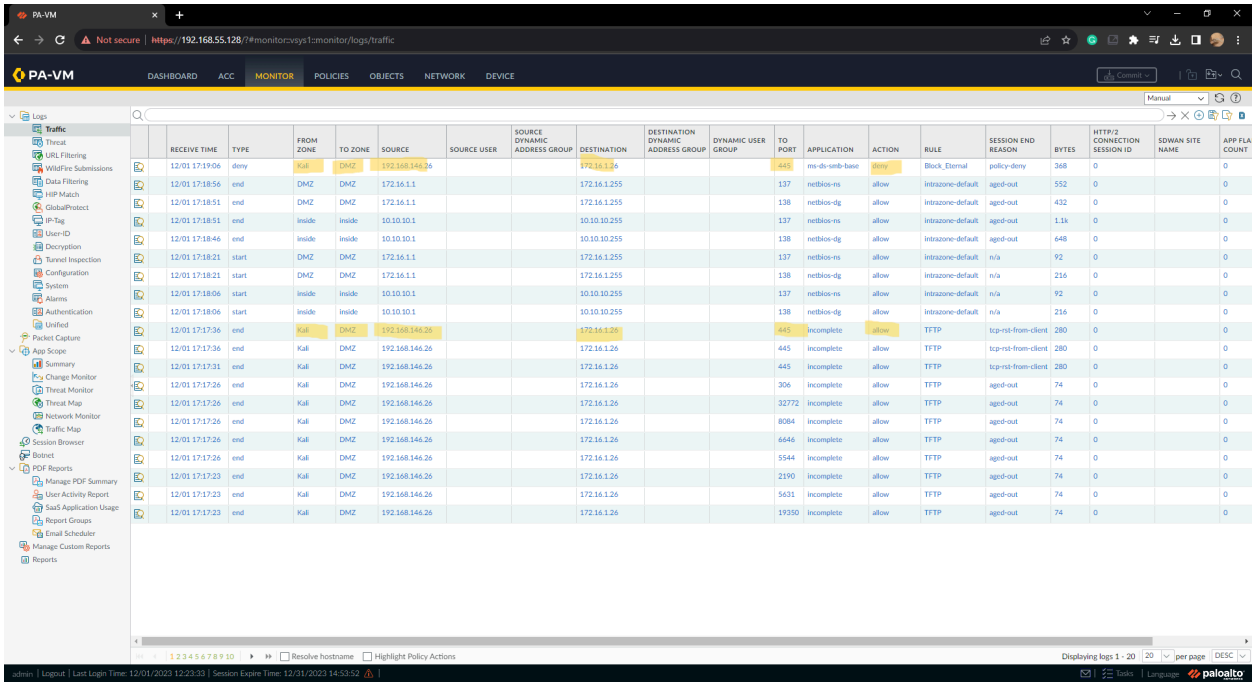

Figure 80: Log showing the change in system form allowing SMB to block all applications.

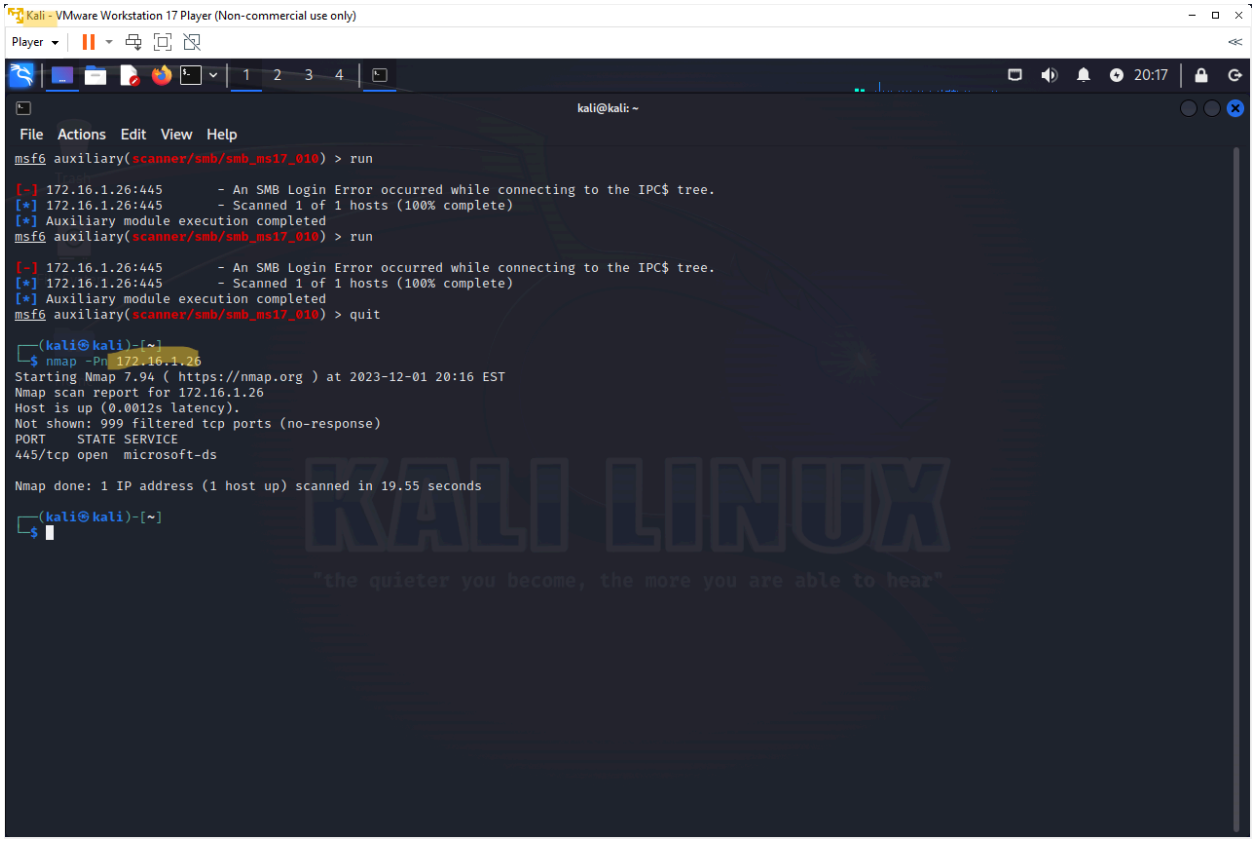

Figure 81: Kali Linux Nmap scan showing that the port is still open.

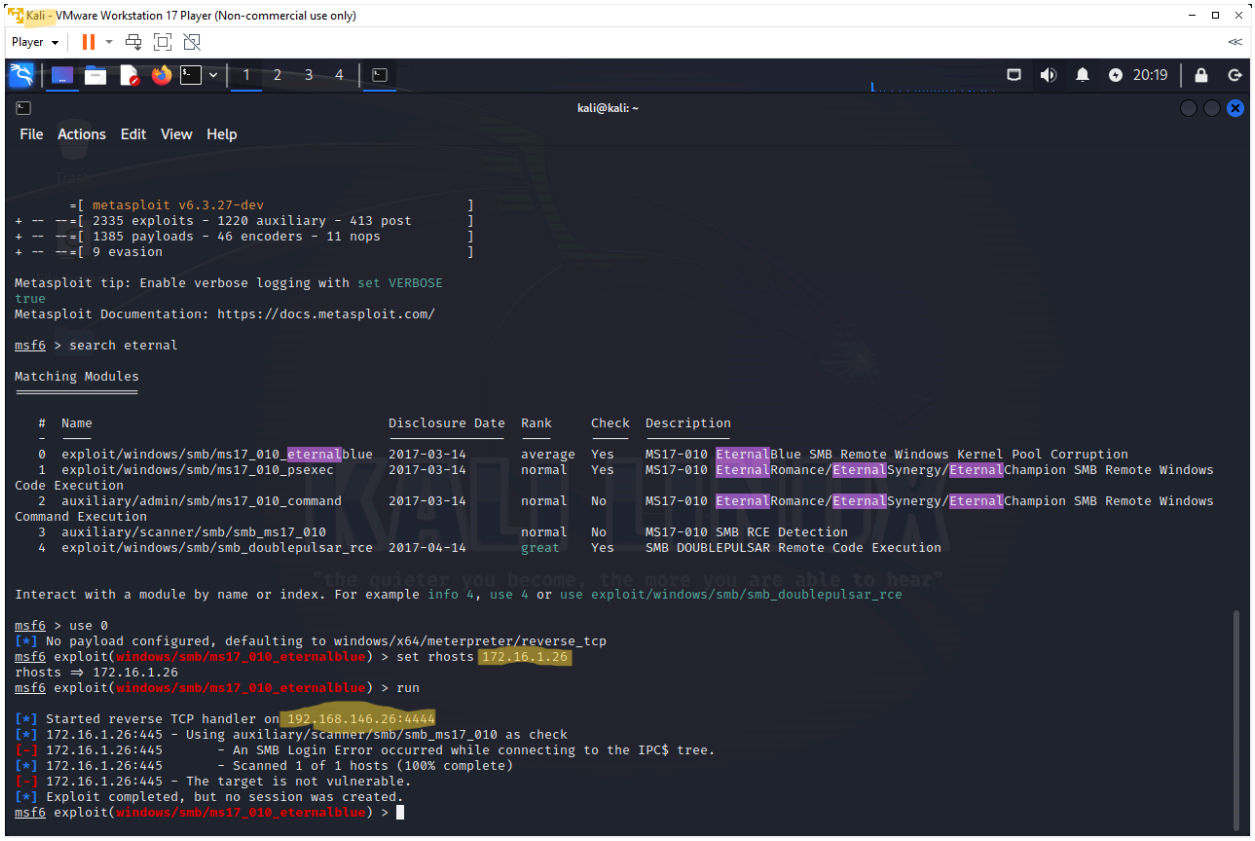

Figure 82: Kali Linux machine showing that target cannot be attacked.

The End.

It has been a pleasure taking this course, and I adored the practical aspects of it and all the experience it has given me.