Cloud and Network Security-C1-2026

Student Name: **Felix Webbo**

Student No: **CS-CNS11-26044**

SUNDAY, FEB 01, 2026

Week 2: Assignment 2

Class Exercise: **HTB Academy: Introduction to Network Traffic Analysis**

1 ABSTRACT

This report examined network traffic analysis as a cybersecurity technique for identifying anomalies and malicious activities within a network environment. Packet captures were analyzed using established traffic analysis methodologies and tools to assess protocol behavior, communication patterns, and indicators of compromise. The findings revealed evidence of suspicious activity, including unauthorized privilege escalation commands originating from a compromised host. The study demonstrated the importance of network traffic analysis in detecting, investigating, and responding to security incidents effectively.

Table of Contents

2 INTRODUCTION

Network traffic analysis was used as a critical method for monitoring, detecting, and investigating security threats within computer networks. By examining data packets traversing a network, security practitioners were able to identify abnormal behavior, misconfigurations, and malicious communications. Both defensive and offensive security teams rely on this approach to gain visibility into network operations, detect intrusions, and understand attacker techniques. This report focused on the analysis of captured network traffic to determine the presence and scope of a security incident.

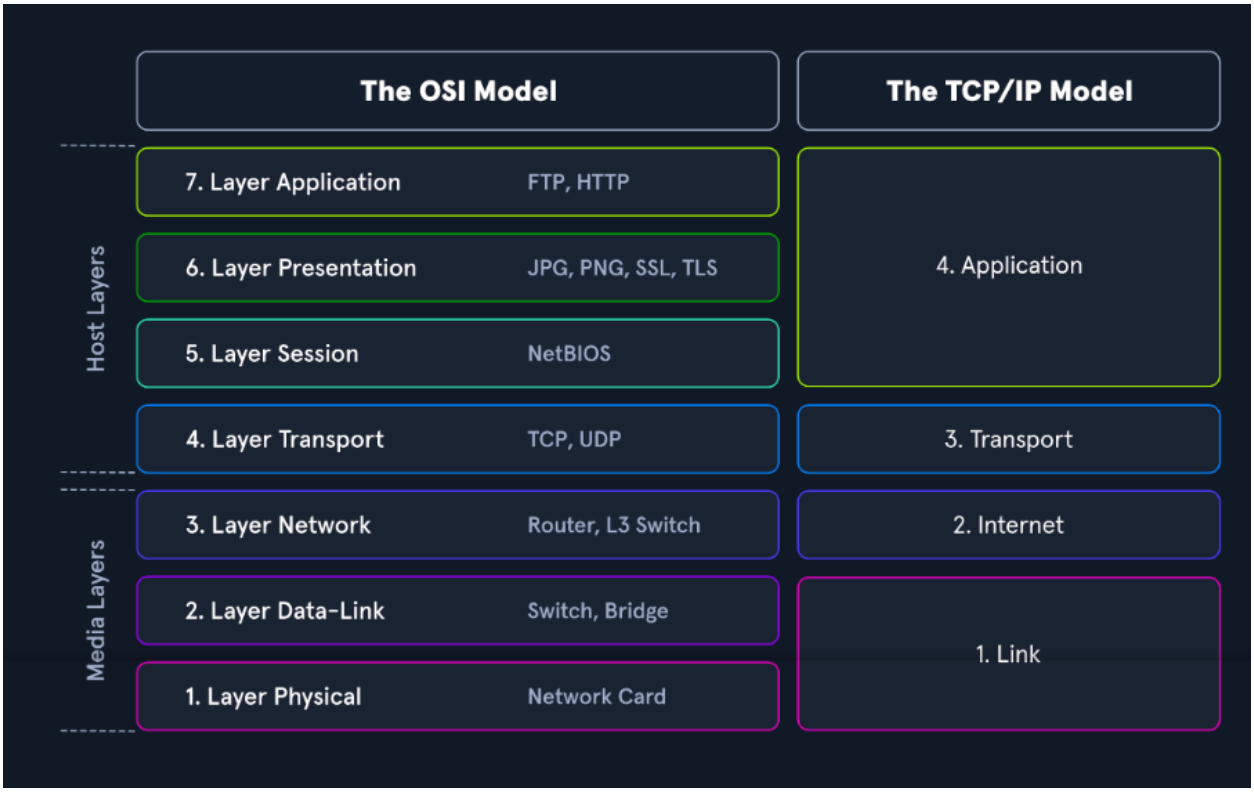
3 BACKGROUND ON NETWORK TRAFFIC ANALYSIS

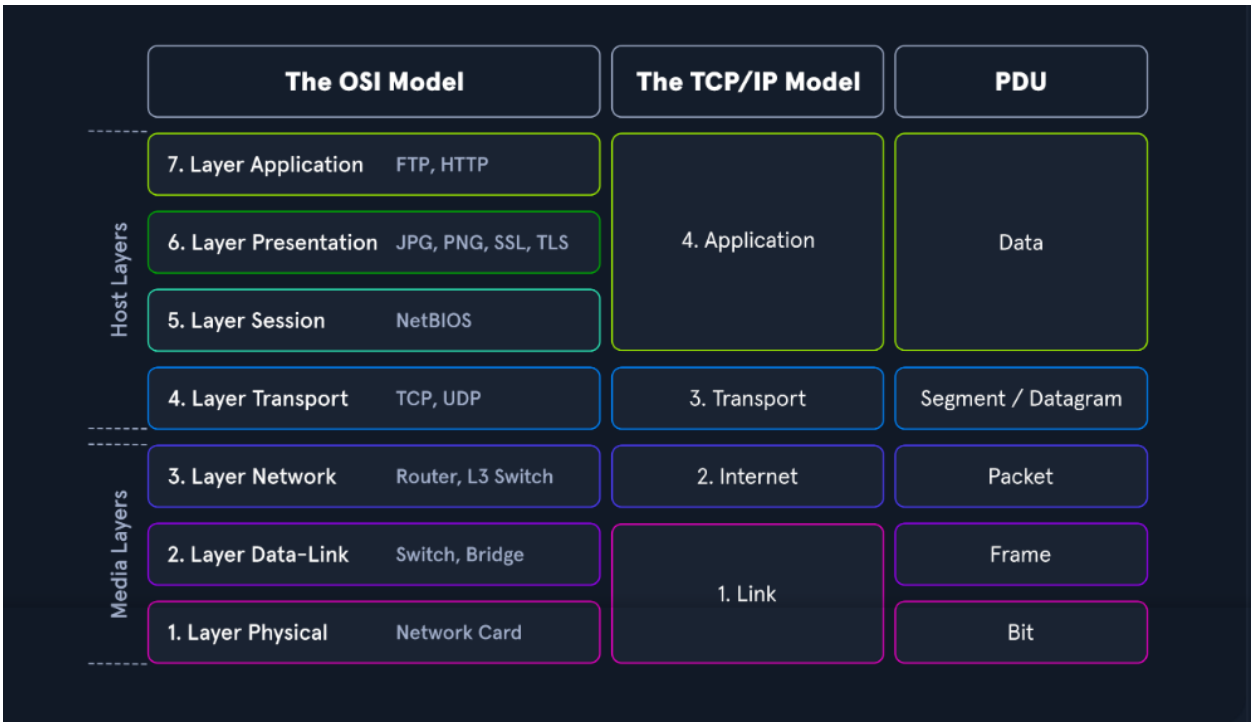
Network Traffic Analysis (NTA) was defined as the systematic inspection of network communications to establish normal traffic baselines, detect deviations, and identify security threats. The process involved understanding common ports, protocols, and traffic flows while leveraging tools such as Wireshark and tcpdump. Knowledge of the TCP/IP stack, OSI model, and protocol encapsulation was essential for interpreting packet-level data and reconstructing communication sessions accurately.

4 METHODOLOGY

The analysis was conducted using a packet capture (PCAP) file obtained from a monitored network segment. Traffic was examined using filtering techniques to reduce noise and isolate relevant packets. Protocol-specific filters were applied to distinguish between TCP, UDP, ARP, and other traffic types. Encapsulation concepts were considered while inspecting Protocol Data Units (PDUs) to understand interactions across network layers.

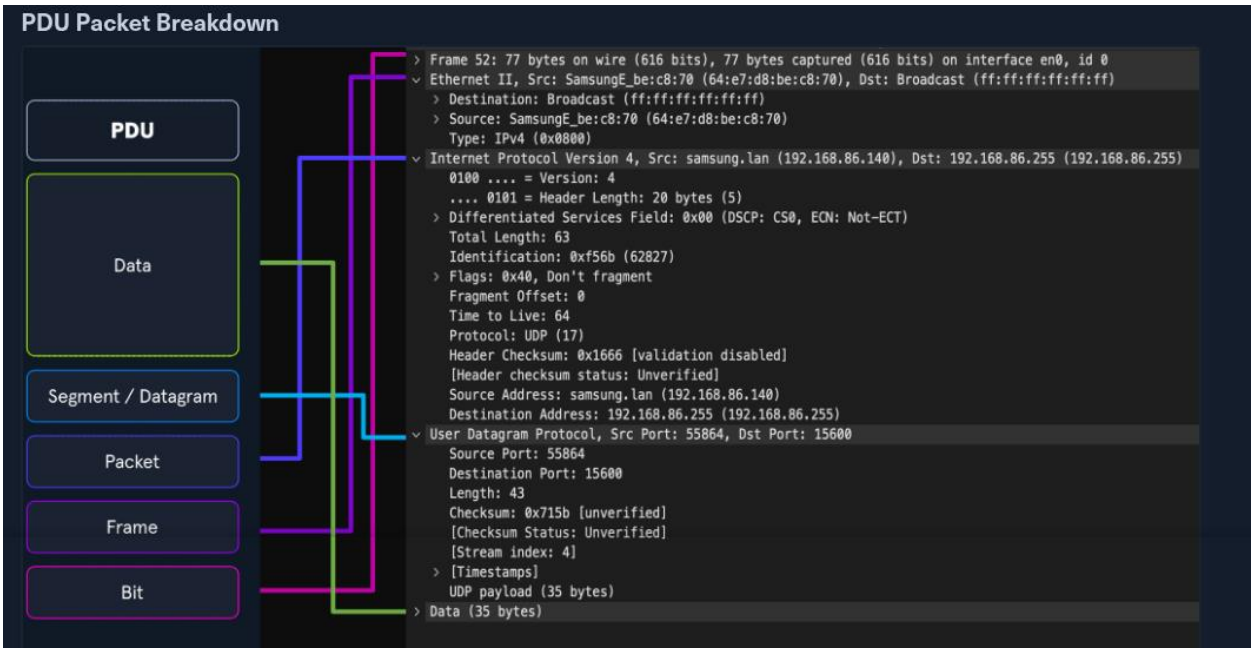
4.1 Networking Primer - Layers 1-4

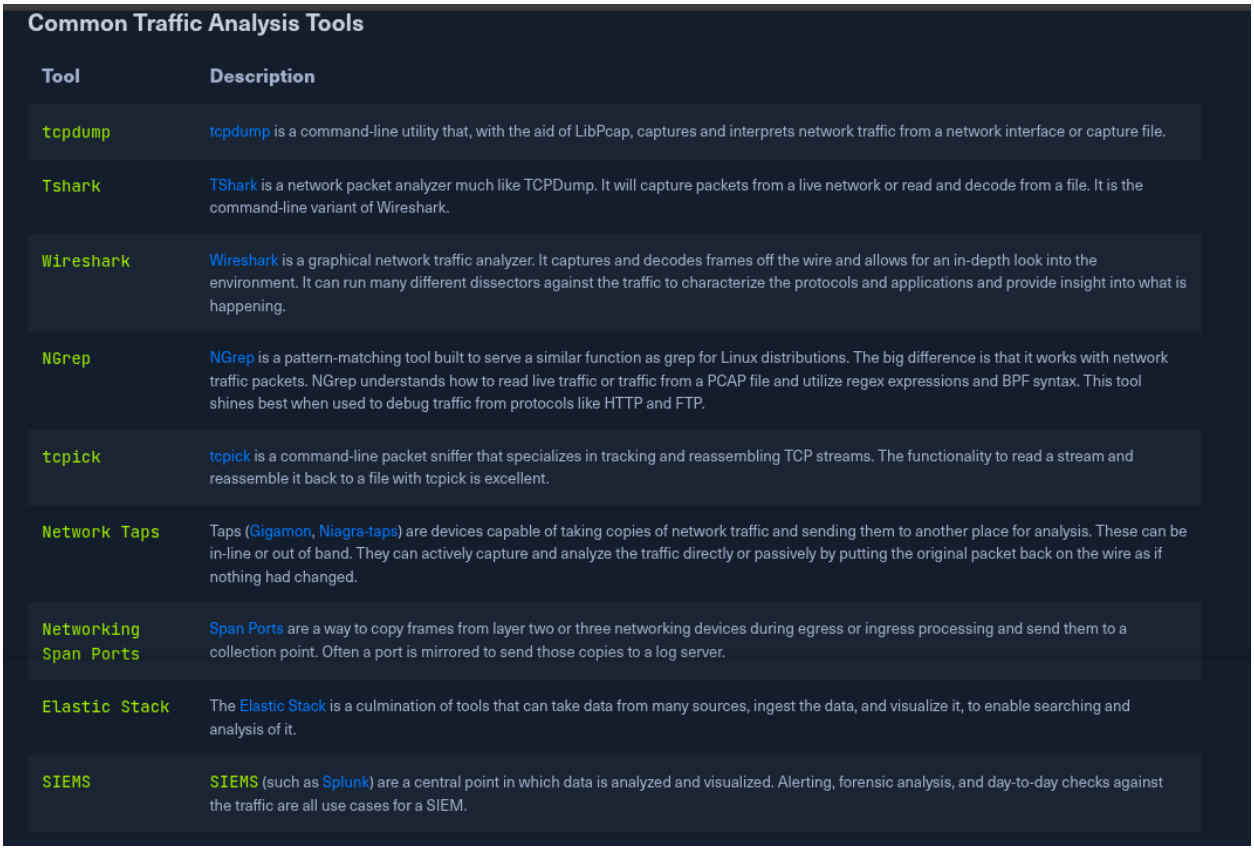




When inspecting a Protocol Data Unit (PDU), the concept of encapsulation was kept in mind. As data moved down the protocol stack, each layer wrapped the previous layer's data in a new

encapsulation. This encapsulation added layer-specific information into the PDU header. The information varied by layer but typically included details about the payload from the previous layer, operational flags, communication negotiation options, source and destination IP addresses, port numbers, transport layer protocols, and application layer protocols.



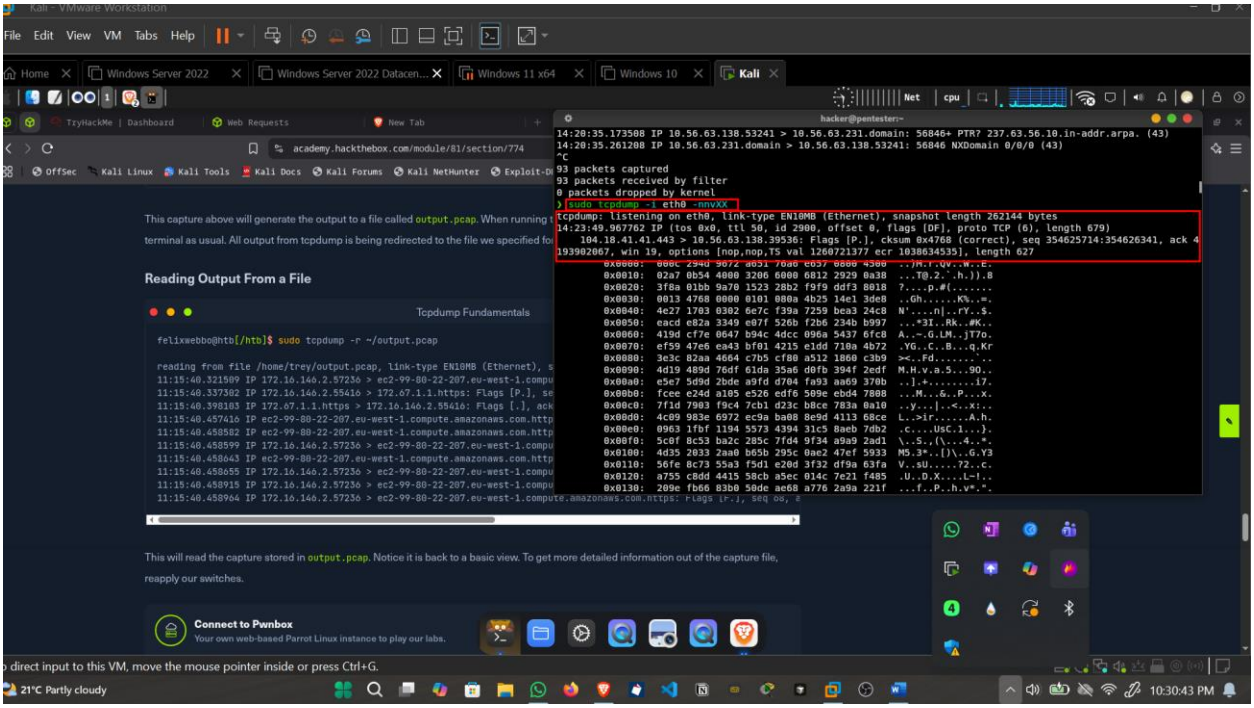


Common Traffic Analysis Tools

Tool	Description
tcpdump	tcpdump is a command-line utility that, with the aid of LibPcap, captures and interprets network traffic from a network interface or capture file.
Tshark	TShark is a network packet analyzer much like TCPDump. It will capture packets from a live network or read and decode from a file. It is the command-line variant of Wireshark.
Wireshark	Wireshark is a graphical network traffic analyzer. It captures and decodes frames off the wire and allows for an in-depth look into the environment. It can run many different dissectors against the traffic to characterize the protocols and applications and provide insight into what is happening.
NGrep	NGrep is a pattern-matching tool built to serve a similar function as grep for Linux distributions. The big difference is that it works with network traffic packets. NGrep understands how to read live traffic or traffic from a PCAP file and utilize regex expressions and BPF syntax. This tool shines best when used to debug traffic from protocols like HTTP and FTP.
tcpick	tcpick is a command-line packet sniffer that specializes in tracking and reassembling TCP streams. The functionality to read a stream and reassemble it back to a file with tcpick is excellent.
Network Taps	Taps (Gigamon, Niagra-taps) are devices capable of taking copies of network traffic and sending them to another place for analysis. These can be in-line or out of band. They can actively capture and analyze the traffic directly or passively by putting the original packet back on the wire as if nothing had changed.
Networking Span Ports	Span Ports are a way to copy frames from layer two or three networking devices during egress or ingress processing and send them to a collection point. Often a port is mirrored to send those copies to a log server.
Elastic Stack	The Elastic Stack is a culmination of tools that can take data from many sources, ingest the data, and visualize it, to enable searching and analysis of it.
SIEMS	SIEMS (such as Splunk) are a central point in which data is analyzed and visualized. Alerting, forensic analysis, and day-to-day checks against the traffic are all use cases for a SIEM.

5 TCPDUMP FUNDAMENTALS

Tcpdump was a command-line packet capture tool used to capture and analyze network traffic from a file or live network interface. It operated on Unix-like systems and required no graphical interface, making it suitable for terminal and remote (SSH) environments. The tool relied on libpcap and promiscuous mode to capture packets across the local network, not only traffic destined for the host system. Due to its low-level hardware access, administrative privileges were required, and it was commonly preinstalled on most Linux distributions.

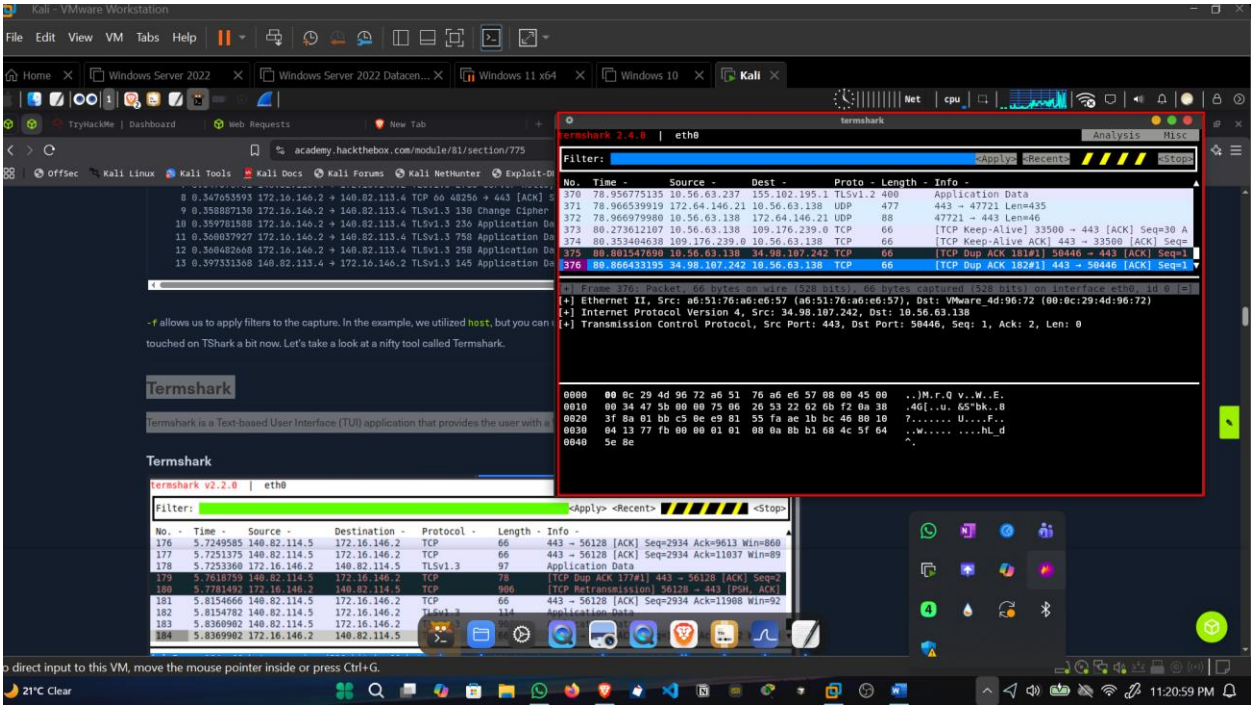


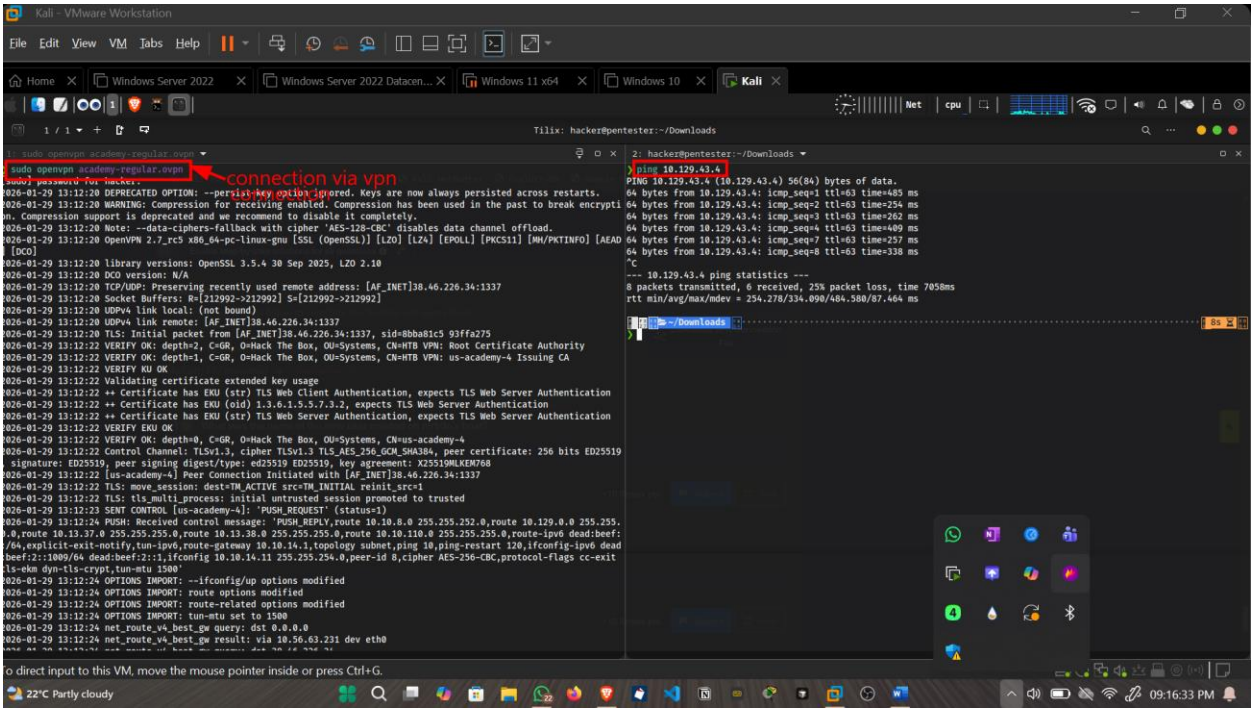
6 ANALYSIS WITH WIRESHARK

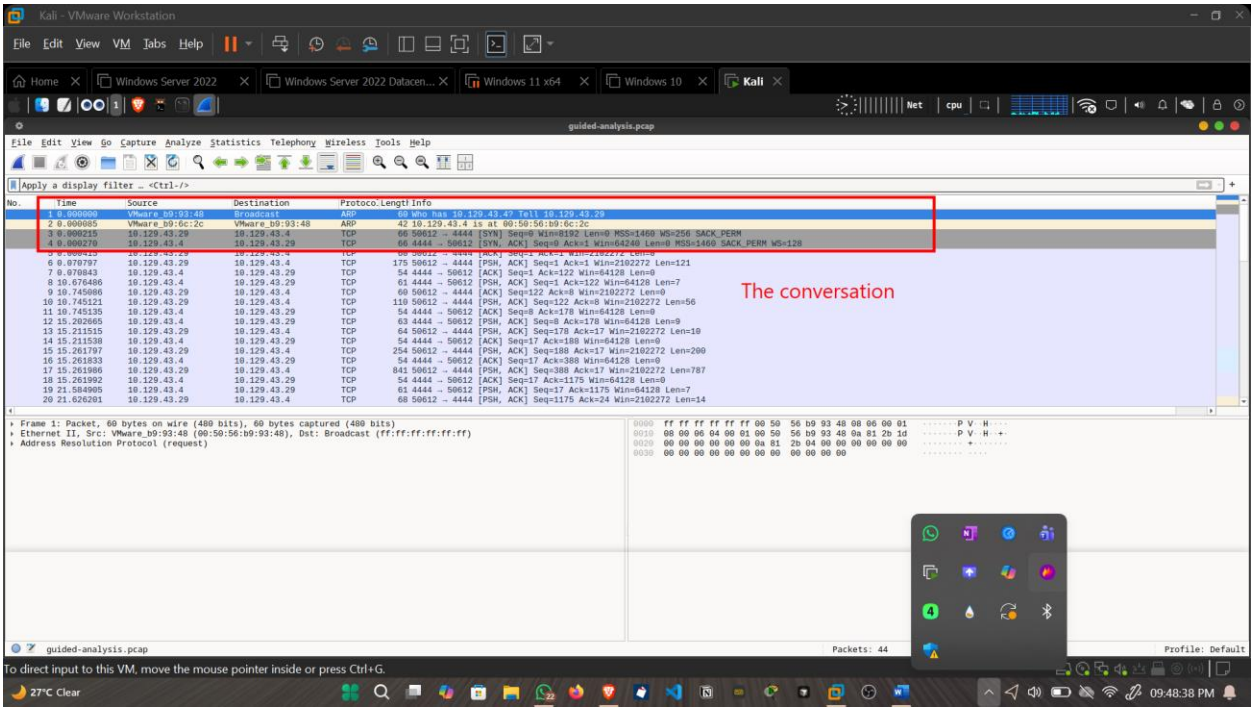
Wireshark is a free, open-source network traffic analyzer similar to tcpdump but with a graphical interface. It supports multiple platforms, captures traffic from various interfaces (such as Wi-Fi, USB, and Bluetooth), and provides powerful, in-depth packet analysis. When a GUI is unavailable, command-line variants can be used instead.

6.1 Termshark

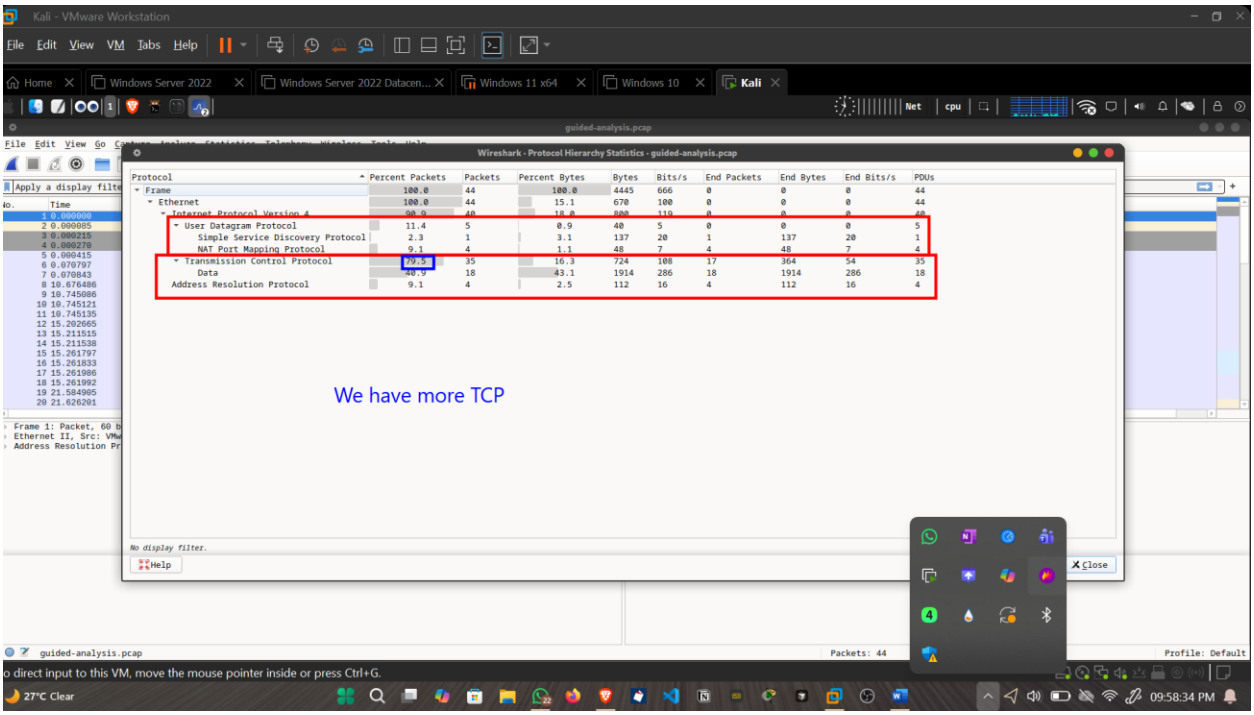
Termshark is a Text-based User Interface (TUI) application that provides the user with a Wireshark-like interface right in your terminal window.


Kali - VMware Workstation
File Edit View VM Tabs Help
Home
Windows Server 2022
Windows Server 2022 Datacen...
Windows 11 x64
Windows 10
Kali
termshark
termshark 2.4.0 | eth0
Filter:
No. Time Source Dest Proto Length Info
370 78.956775135 10.56.63.237 155.102.195.1 TLSv1.2 400 Application Data
371 78.966539919 172.64.146.21 10.56.63.138 UDP 477 443 → 47721 Len=435
372 78.966979980 10.56.63.138 172.64.146.21 UDP 88 47721 → 443 Len=46
373 80.273612107 10.56.63.138 109.176.239.0 TCP 66 [TCP Keep-Alive] 33500 → 443 [ACK] Seq=30 A
374 80.353404638 109.176.239.0 10.56.63.138 TCP 66 [TCP Keep-Alive ACK] 443 → 33500 [ACK] Seq=
375 80.801547690 10.56.63.138 34.98.107.242 TCP 66 [TCP Dup ACK 181#1] 50446 → 443 [ACK] Seq=1
376 80.866433195 34.98.107.242 10.56.63.138 TCP 66 [TCP Dup ACK 182#1] 443 → 50446 [ACK] Seq=1
[+] Ethernet II, Src: a6:51:76:a6:e6:57 (a6:51:76:a6:e6:57), Dst: VMware_4d:96:72 (00:0c:29:4d:96:72)
[+] Internet Protocol Version 4, Src: 34.98.107.242, Dst: 10.56.63.138
[+] Transmission Control Protocol, Src Port: 443, Dst Port: 50446, Seq: 1, Ack: 2, Len: 0
0000 00 0c 29 4d 96 72 a6 51 76 a6 e6 57 08 00 45 00
0010 00 34 47 5b 00 00 75 06 26 53 22 62 6b f2 0a 38
0020 3f 8a 01 bb c5 0e e9 81 55 fa ae 1b bc 46 80 10
0030 04 13 77 fb 00 00 01 01 08 0a 8b b1 68 4c 5f 64
0040 5e 8e
-f allows us to apply filters to the capture. In the example, we utilized host, but you can
touched on TShark a bit now. Let's take a look at a nifty tool called Termshark.
Termshark
Termshark is a Text-based User Interface (TUI) application that provides the user with a
Termshark
termshark v2.2.0 | eth0
Filter:
No. Time Source Destination Protocol Length Info
176 5.7249585 140.82.114.5 172.16.146.2 TCP 66 443 → 56128 [ACK] Seq=2934 Ack=9613 Win=860
177 5.7251375 140.82.114.5 172.16.146.2 TCP 66 443 → 56128 [ACK] Seq=2934 Ack=11037 Win=89
178 5.7253360 172.16.146.2 140.82.114.5 TLSv1.3 97 Application Data
179 5.7618759 140.82.114.5 172.16.146.2 TCP 78 [TCP Dup ACK 177#1] 443 → 56128 [ACK] Seq=2
180 5.7781492 172.16.146.2 140.82.114.5 TCP 906 [TCP Retransmission] 56128 → 443 [PSH, ACK]
181 5.8154666 140.82.114.5 172.16.146.2 TCP 66 443 → 56128 [ACK] Seq=2934 Ack=11908 Win=92
182 5.8154782 140.82.114.5 172.16.146.2 TLSv1.3 114 Application Data
183 5.8360902 140.82.114.5 172.16.146.2
184 5.8369902 172.16.146.2 140.82.114.5
o direct input to this VM, move the mouse pointer inside or press Ctrl+G.
21°C Clear
11:20:59 PM

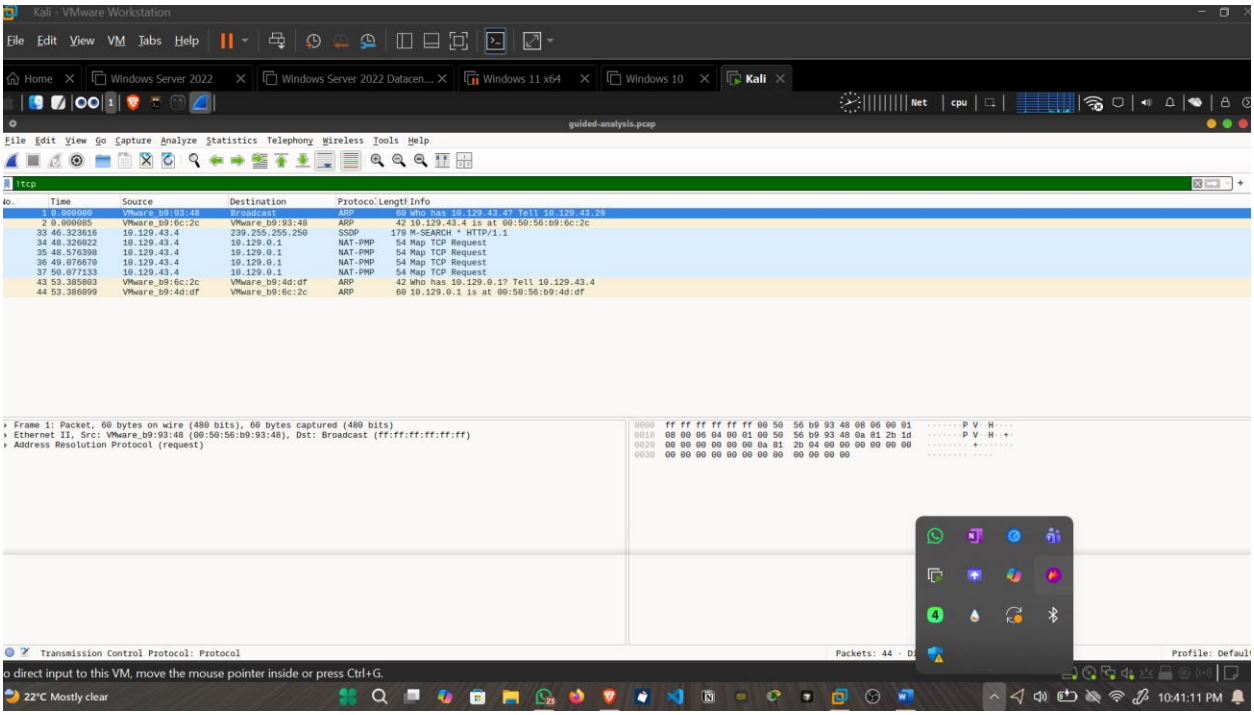




After checking out the conversation's plugin pictured above, it can see there are only three conversations captured in this pcap file, and they all pertain to our suspicious host.

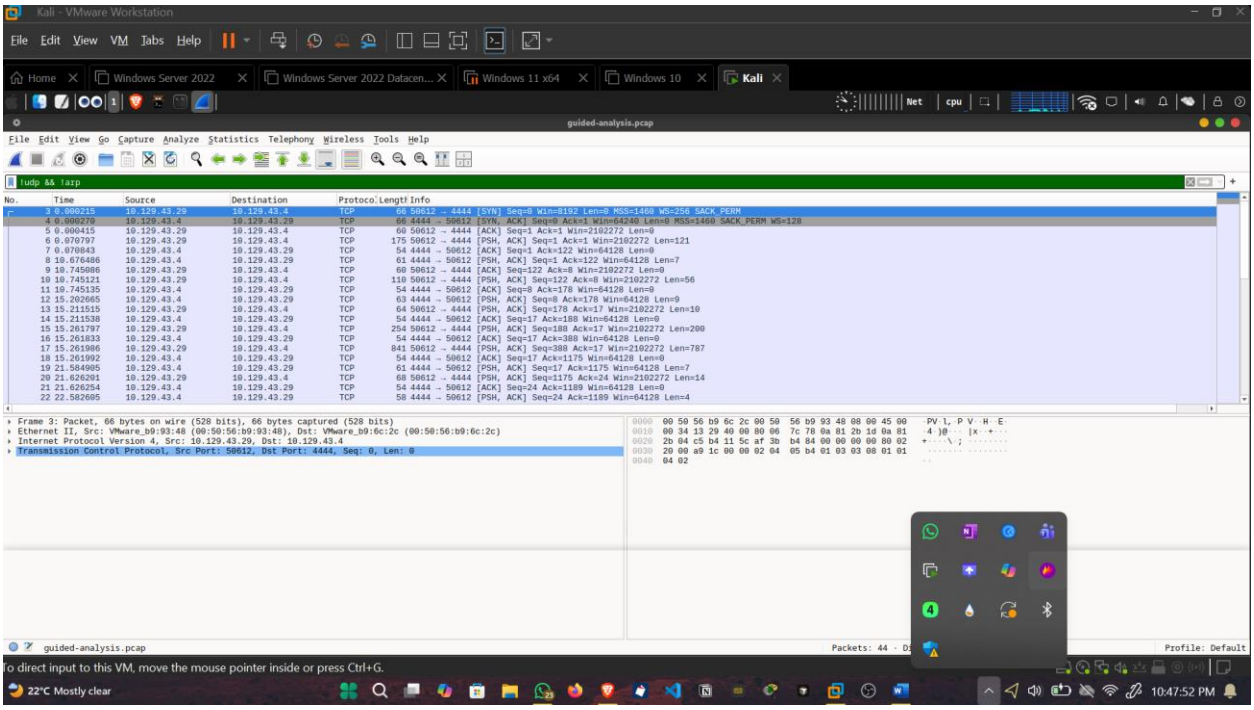


It can see here that this PCAP is mostly TCP traffic, with a bit of UDP traffic. Since there is less UDP than TCP traffic, analyzed UDP first by applying a filter.

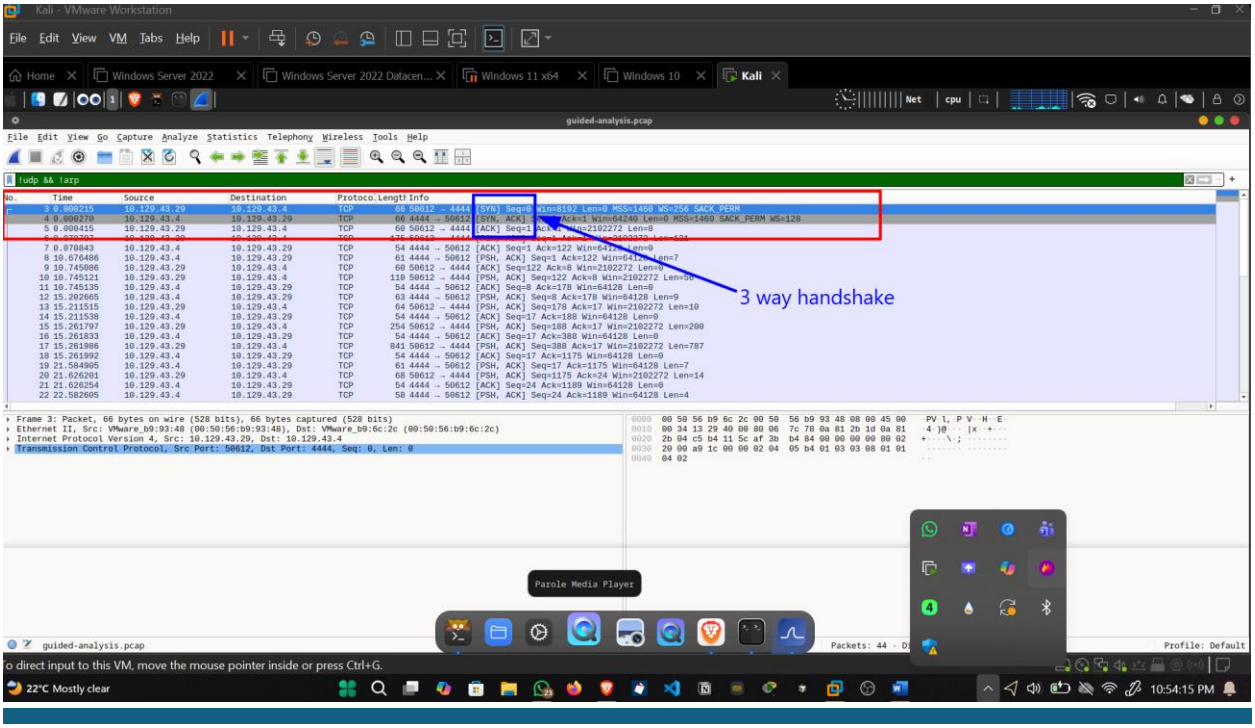


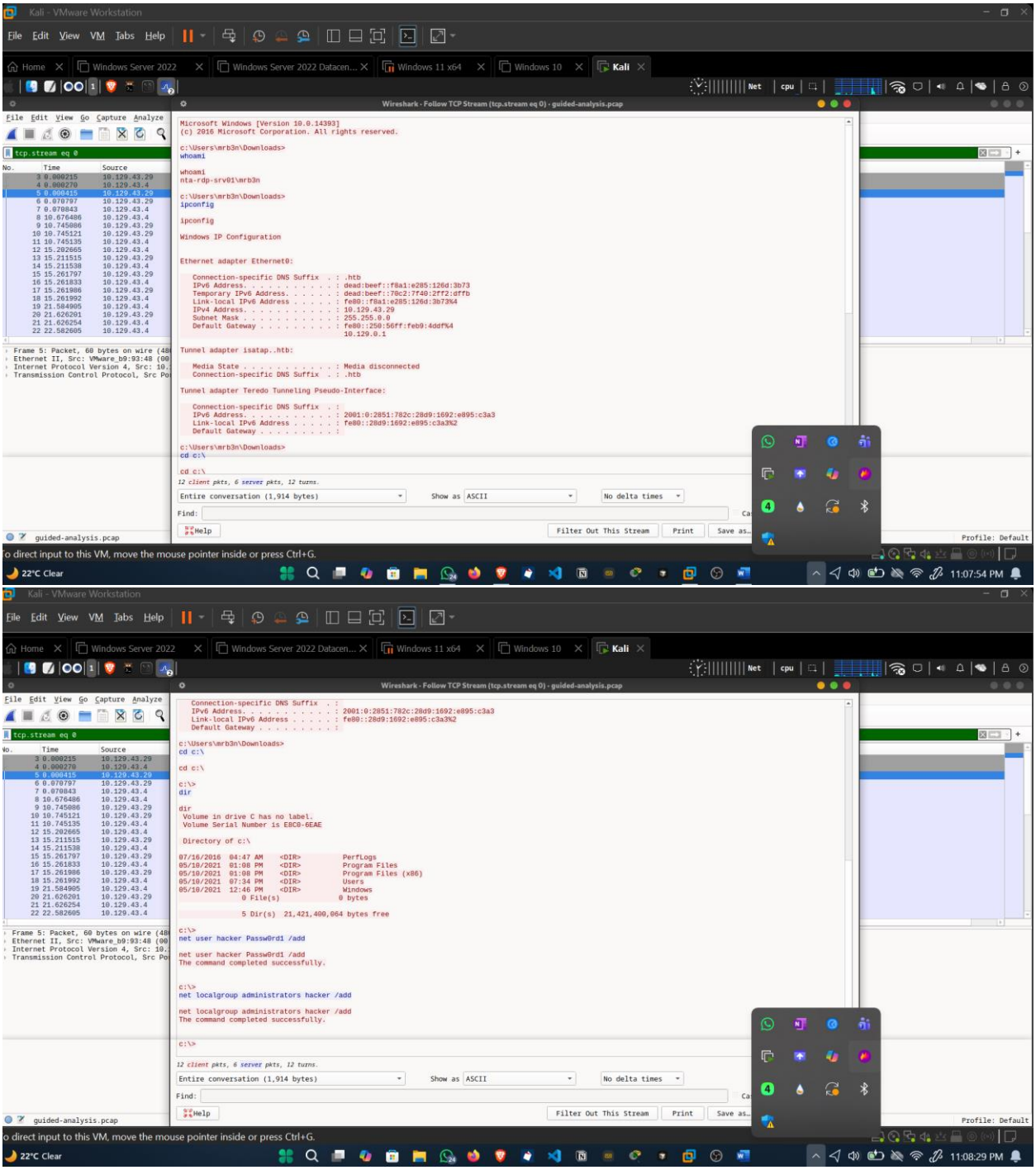
When filtering on just UDP traffic, Only nine packets are seen. Four arp packets, four Network Address Translation NAT, and one Simple Sevice Discovery Protocol SSDP packet. It can be

concluded based on their packet types and information they contain that this traffic is normal network traffic and nothing to be concerned about.

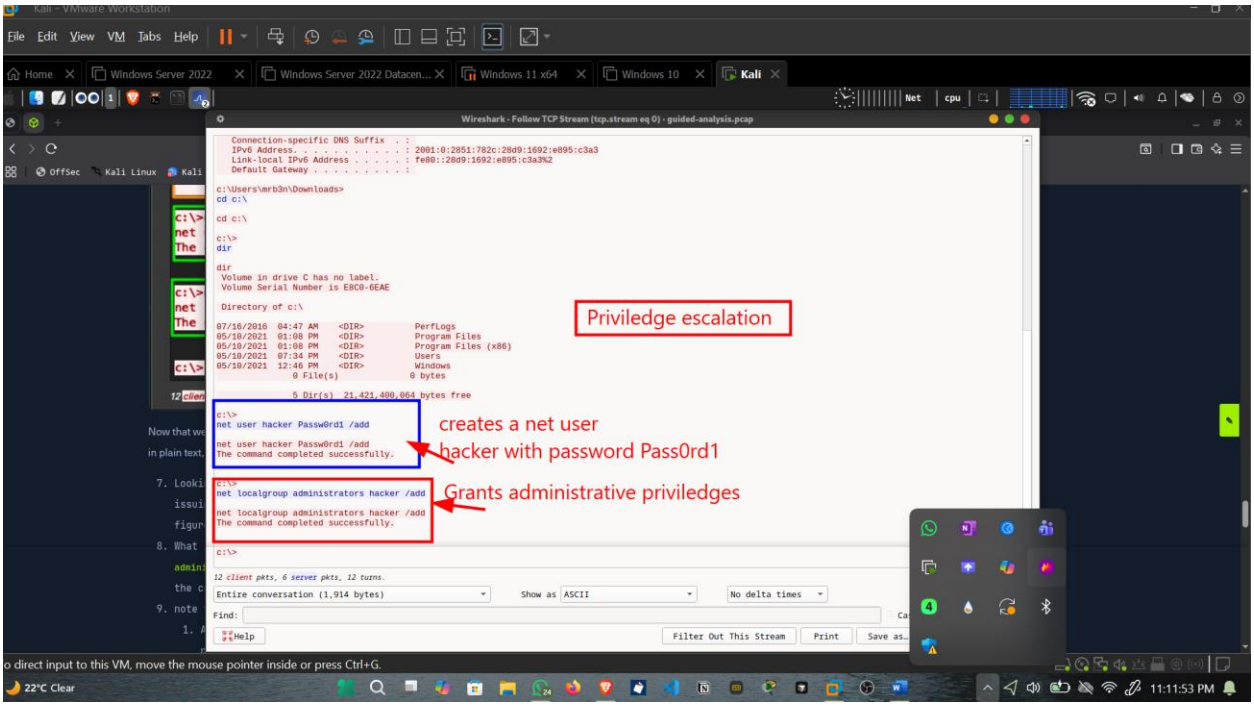


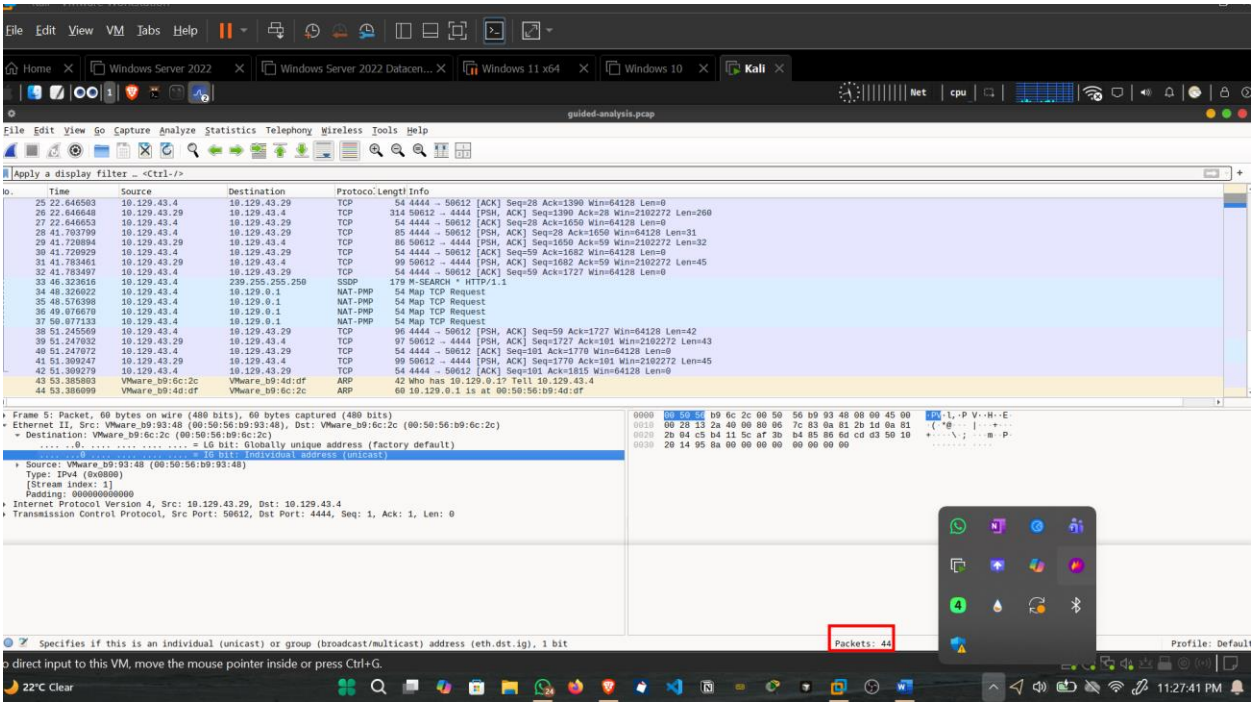
After clearing the view a bit, the remaining packets are all TCP, and all appear to be the same conversation between hosts 10.129.43.4 and 10.129.43.29. This was determined since the session establishment via a three-way handshake at packet 3, and the same ports are used through the rest of the packets in the output below.

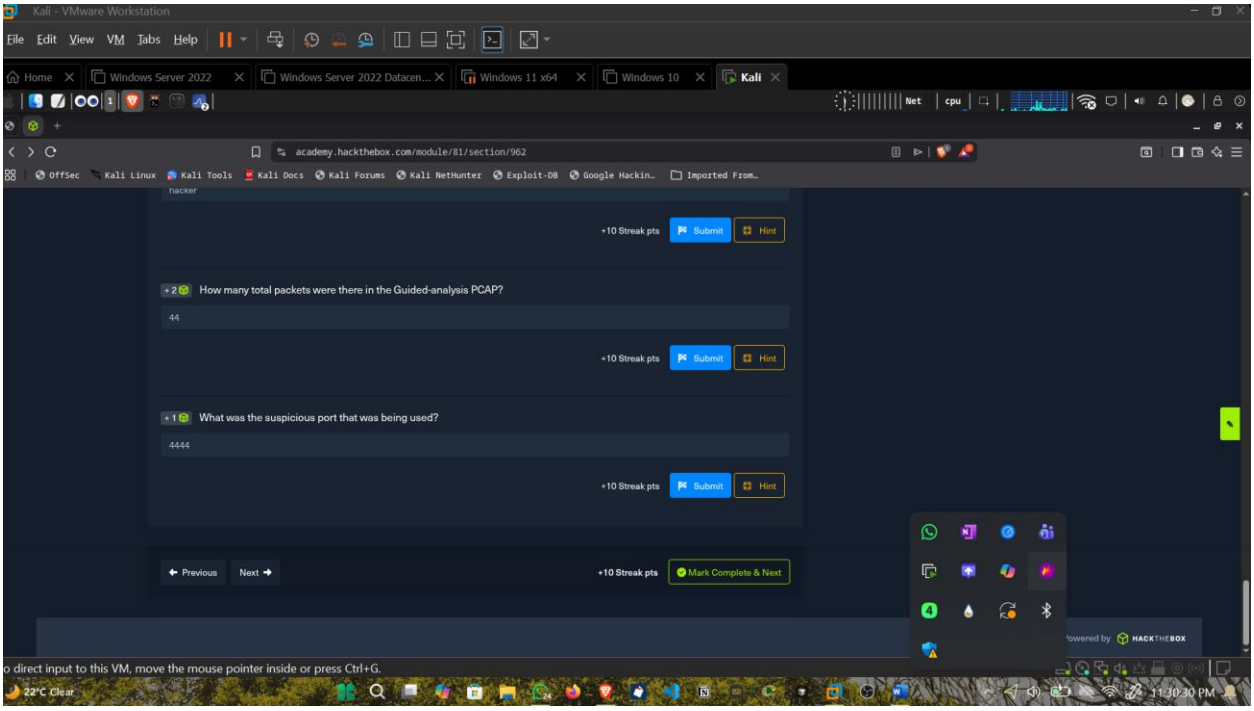



Kali - VMware Workstation
Wireshark - Follow TCP Stream (tcp.stream eq 0) - guided-analysis.pcap
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.
c:\Users\mrb3n\Downloads>
whoami
whoami
nta-rdp-srv01\mrb3n
c:\Users\mrb3n\Downloads>
ipconfig
ipconfig
Windows IP Configuration
Ethernet adapter Ethernet0:
Connection-specific DNS Suffix . : .htb
IPv6 Address. . . . . : dead:beef::f8a1:e285:126d:3b73
Temporary IPv6 Address. . . . . : dead:beef::70c2:7f40:2ff2:dffb
Link-local IPv6 Address . . . . . : fe80::f8a1:e285:126d:3b73%4
IPv4 Address. . . . . : 10.129.43.29
Subnet Mask . . . . . : 255.255.0.0
Default Gateway . . . . . : fe80::250:56ff:feb9:4ddf%4
10.129.0.1
Tunnel adapter isatap.htb:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . : .htb
Tunnel adapter Teredo Tunneling Pseudo-Interface:
Connection-specific DNS Suffix . :
IPv6 Address. . . . . : 2001:0:2851:782c:28d9:1692:e895:c3a3
Link-local IPv6 Address . . . . . : fe80::28d9:1692:e895:c3a3%2
Default Gateway . . . . . :
c:\Users\mrb3n\Downloads>
cd c:\
cd c:\
12 client pkts, 6 server pkts, 12 turns.
Entire conversation (1,914 bytes)
Show as ASCII
No delta times
Find:
Help
Filter Out This Stream
Print
Save as...
guided-analysis.pcap
Profile: Default
o direct input to this VM, move the mouse pointer inside or press Ctrl+G.
22°C Clear
11:07:54 PM

Kali - VMware Workstation
Wireshark - Follow TCP Stream (tcp.stream eq 0) - guided-analysis.pcap
Connection-specific DNS Suffix . :
IPv6 Address. . . . . : 2001:0:2851:782c:28d9:1692:e895:c3a3
Link-local IPv6 Address . . . . . : fe80::28d9:1692:e895:c3a3%2
Default Gateway . . . . . :
c:\Users\mrb3n\Downloads>
cd c:\
cd c:\
c:\>
dir
dir
Volume in drive C has no label.
Volume Serial Number is E8C0-6EAE
Directory of c:\
07/16/2016 04:47 AM <DIR> PerfLogs
05/10/2021 01:08 PM <DIR> Program Files
05/10/2021 01:08 PM <DIR> Program Files (x86)
05/10/2021 07:34 PM <DIR> Users
05/10/2021 12:46 PM <DIR> Windows
0 File(s) 0 bytes
5 Dir(s) 21,421,400,064 bytes free
c:\>
net user hacker Passw0rd1 /add
net user hacker Passw0rd1 /add
The command completed successfully.
c:\>
net localgroup administrators hacker /add
net localgroup administrators hacker /add
The command completed successfully.
c:\>
12 client pkts, 6 server pkts, 12 turns.
Entire conversation (1,914 bytes)
Show as ASCII
No delta times
Find:
Help
Filter Out This Stream
Print
Save as...
guided-analysis.pcap
Profile: Default
o direct input to this VM, move the mouse pointer inside or press Ctrl+G.
22°C Clear
11:08:29 PM


Priviledge escalation
creates a net user
hacker with password Pass0rd1
Grants administrative priviledges





Based on network analysis, signs of malicious activity were detected originating from host 10.129.43.29. Commands consistent with privilege escalation were observed, including the creation of a new user account and the assignment of local administrator permissions using net commands. The activity appeared to be routed through a host belonging to Bob, who had previously been investigated for the exfiltration of corporate secrets disguised as web traffic. The screenshots documented the network traffic and commands used during the incident. This evidence indicated that the initial security breach had expanded beyond its original scope.

7 ANALYSIS AND FINDINGS

Initial inspection revealed three primary conversations within the PCAP file, all associated with a single suspicious host. The majority of the traffic consisted of TCP packets, with a small number of UDP packets observed.

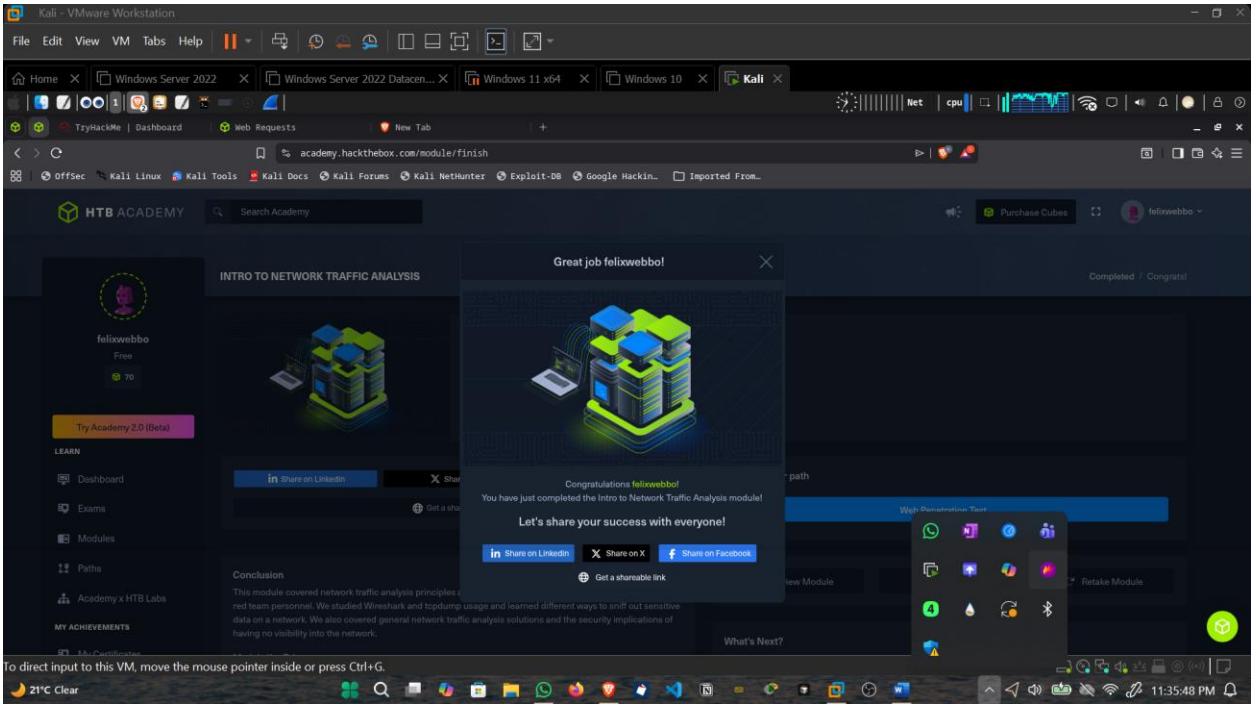
UDP traffic analysis identified nine packets, including ARP, Network Address Translation (NAT), and Simple Service Discovery Protocol (SSDP) packets. Based on packet structure and content, this traffic was classified as normal background network activity and did not indicate malicious behavior.

Further analysis showed that the remaining packets were TCP-based and belonged to a single session between hosts **10.129.43.4** and **10.129.43.29**. This was confirmed through the observation of a TCP three-way handshake and consistent source and destination ports throughout the session. Examination of the payload revealed commands consistent with privilege escalation activities.

8 INCIDENT EVIDENCE

The network traffic indicated malicious actions originating from host **10.129.43.29**. Commands associated with unauthorized privilege escalation were detected, including the creation of a new user account and the assignment of local administrator privileges using system-level commands. The activity appeared to have been routed through a host linked to a previously investigated individual, Bob, who had been suspected of exfiltrating sensitive corporate data disguised as legitimate web traffic. The evidence suggested that the initial breach had escalated into a broader compromise.

The findings highlighted how attackers leveraged legitimate protocols and encrypted-looking traffic to evade detection. Without baseline traffic knowledge and detailed packet analysis, such activity could have remained unnoticed. The case reinforced the importance of continuous monitoring, protocol awareness, and deep packet inspection in modern cybersecurity environments.



Visit [this link]

9 CONCLUSION

Network traffic analysis plays a critical role in identifying and understanding malicious activities within a network. By applying effective filtering techniques, examining protocols, and reconstructing sessions, security teams can uncover unauthorized access and suspicious behavior. Strong networking fundamentals combined with continuous monitoring enable timely detection and mitigation of threats. Ultimately, network traffic analysis is a powerful tool that enhances an organization's ability to protect systems and respond effectively to modern cyber threats.

10 REFERENCES

- Cyber Shujaa LLM
- Wireshark User Guide.