Course: **Cloud and Network Security -C2 -2025**

Student Name: **Daniel-Caleb Cheruiyot**

Student No: **CS-CNS09-25014**

Saturday, May, 31st 2025

Week 2 Assignment 2:

Introduction to Network Traffic Analysis

https://academy.hackthebox.com/achievement/1095115/81

Table Of Contents

Introduction

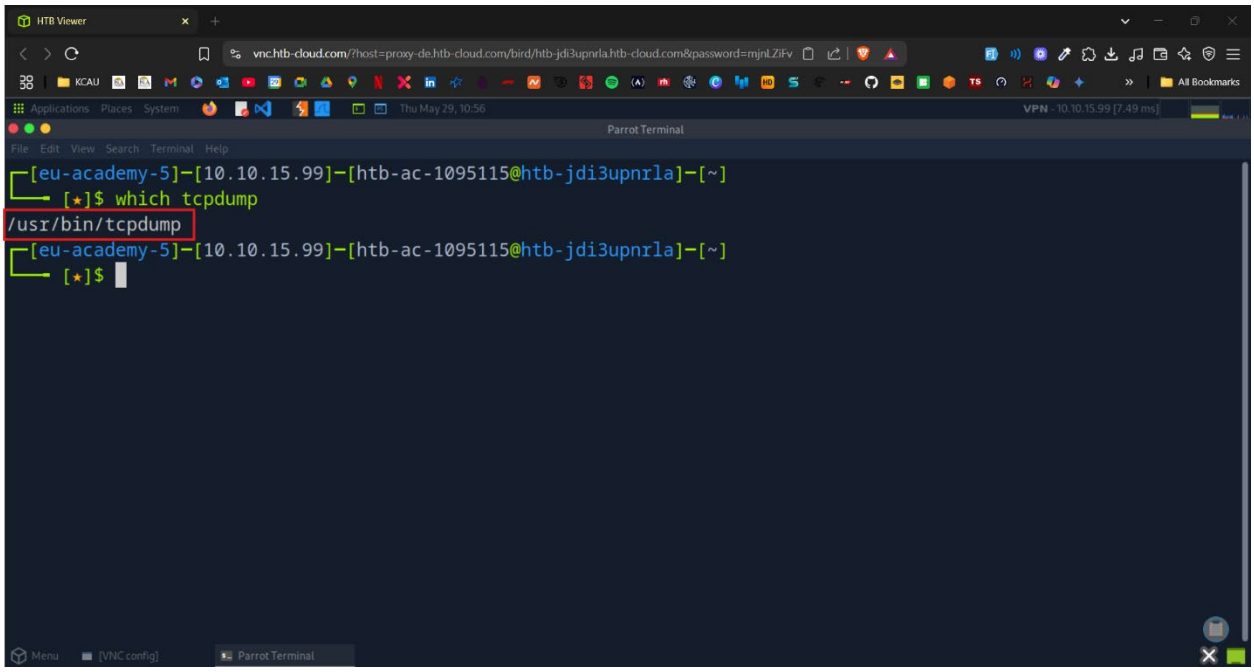
This report outlines my detailed investigation into suspicious network traffic and encrypted Remote Desktop Protocol (RDP) sessions associated with an internal host and also undertook multiple analysis tasks using Wireshark to:

i. Extract transferred images over HTTP suspected of hiding data.
ii. Investigate ongoing FTP and HTTP traffic during a live capture.
iii. Conduct a full traffic analysis workflow to identify signs of intrusion.
iv. Decrypt and analyze an RDP session using a recovered RSA private key.

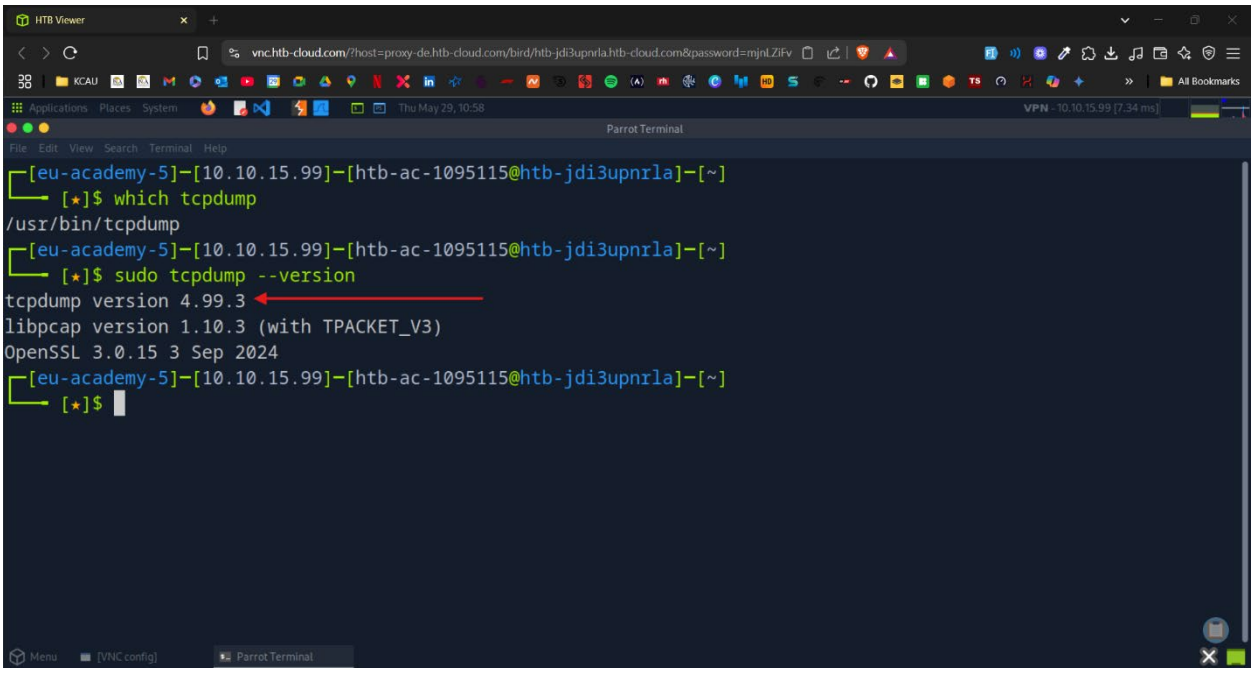
TCP Fundamentals

Tcpdump is a command-line packet sniffer that can directly capture and interpret data frames from a file or network interface.

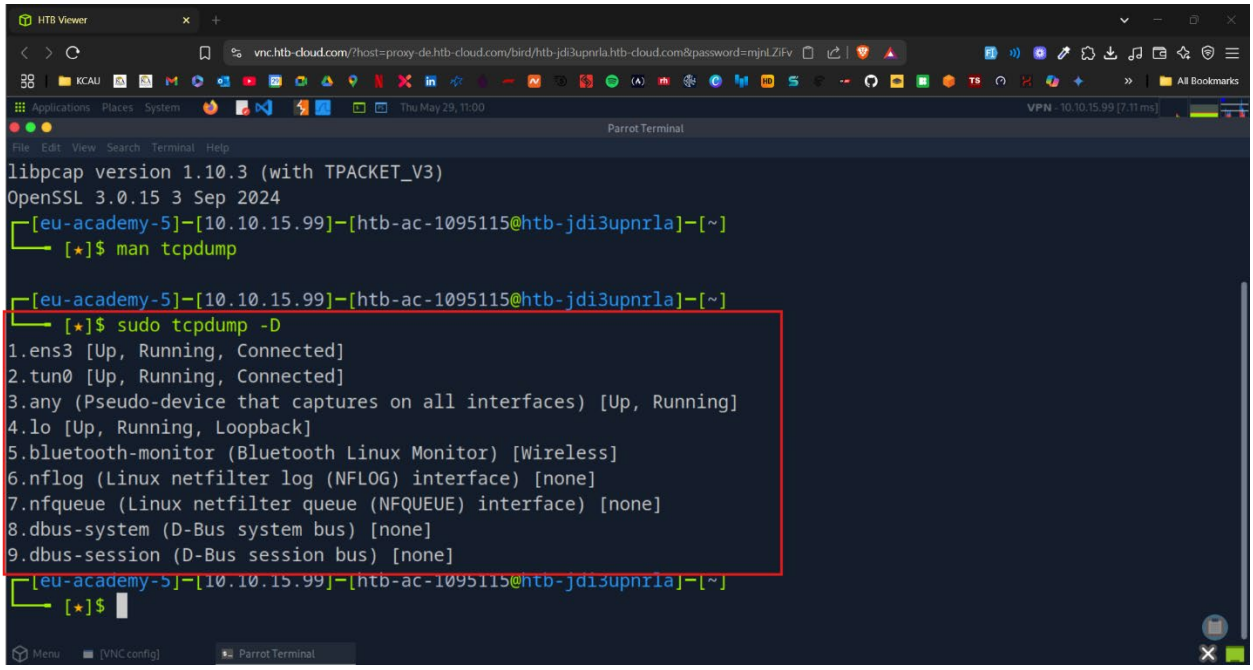
First things first, is to locate tcpdump and validate if it exists



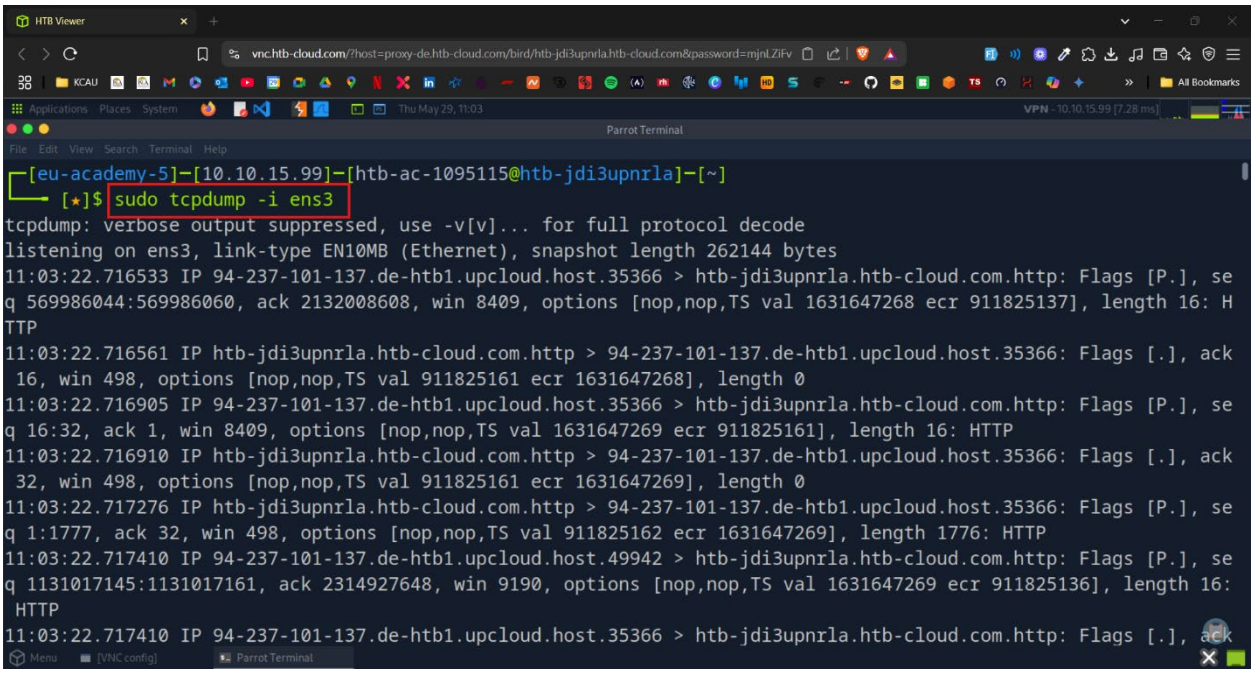
Hence no need for me to install. I then proceeded to check the version of the tcpdump installed which was **4.99.3**

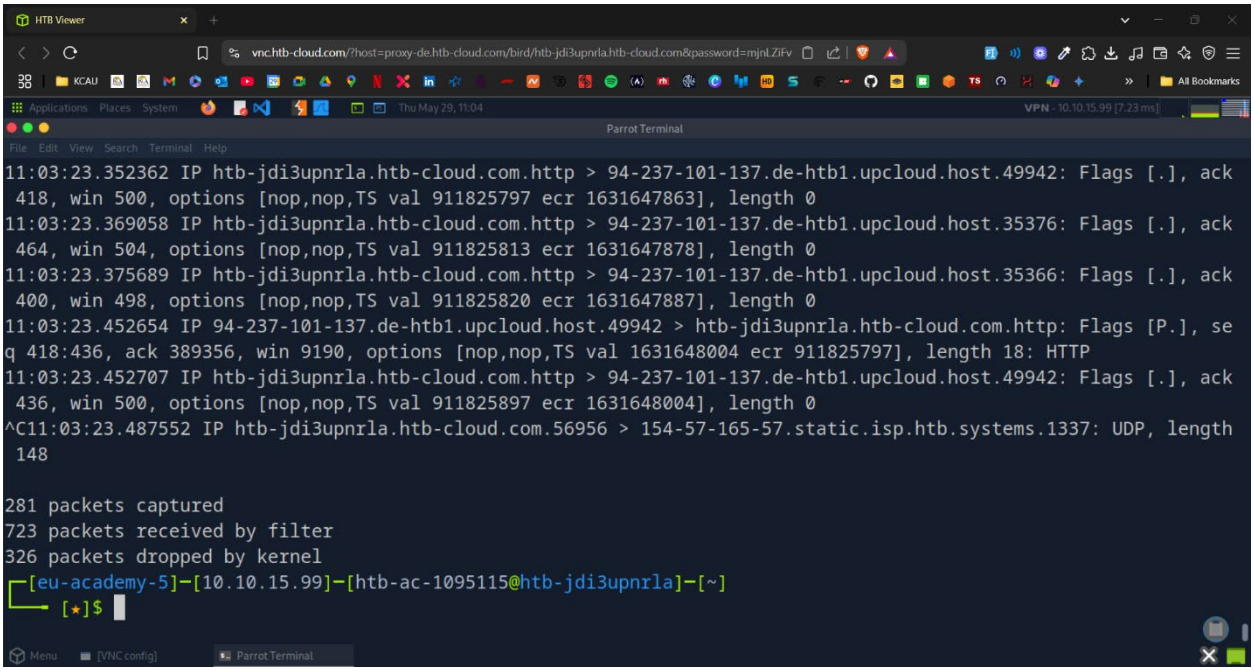


To list the available interfaces, I used the command ***tcpdump -D*** which displayed a list of network interfaces to choose from as shown below.

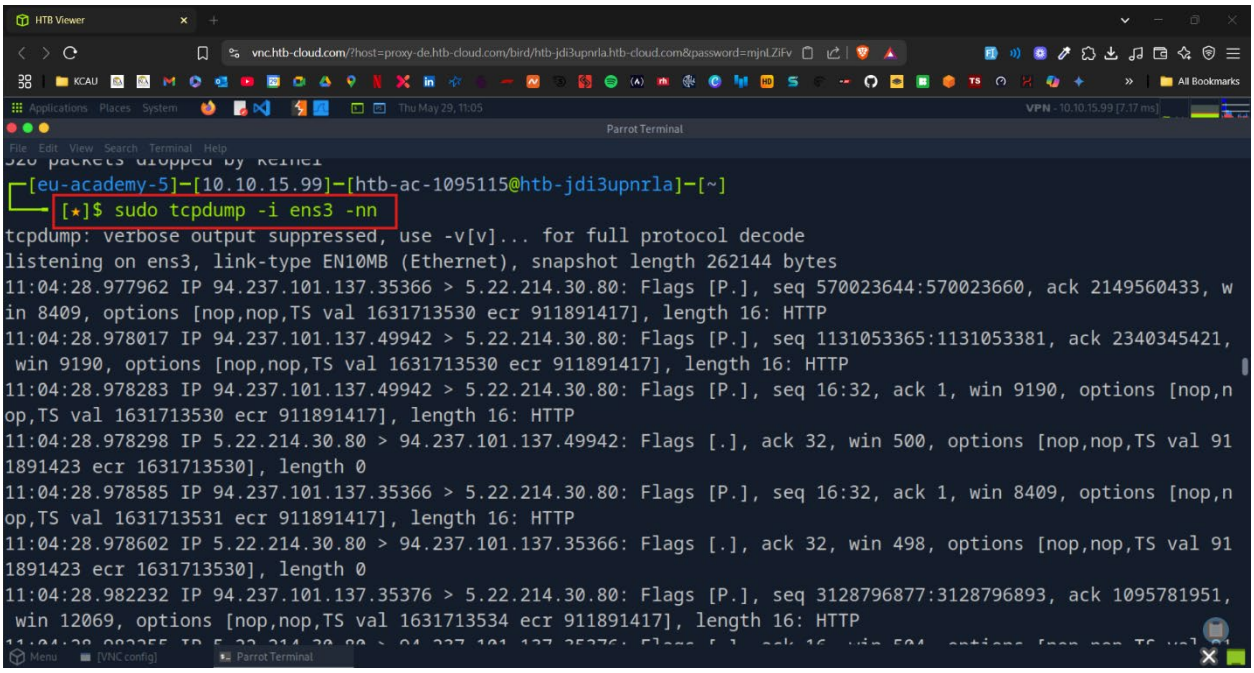


In my case, I chose interface ***ens3***:

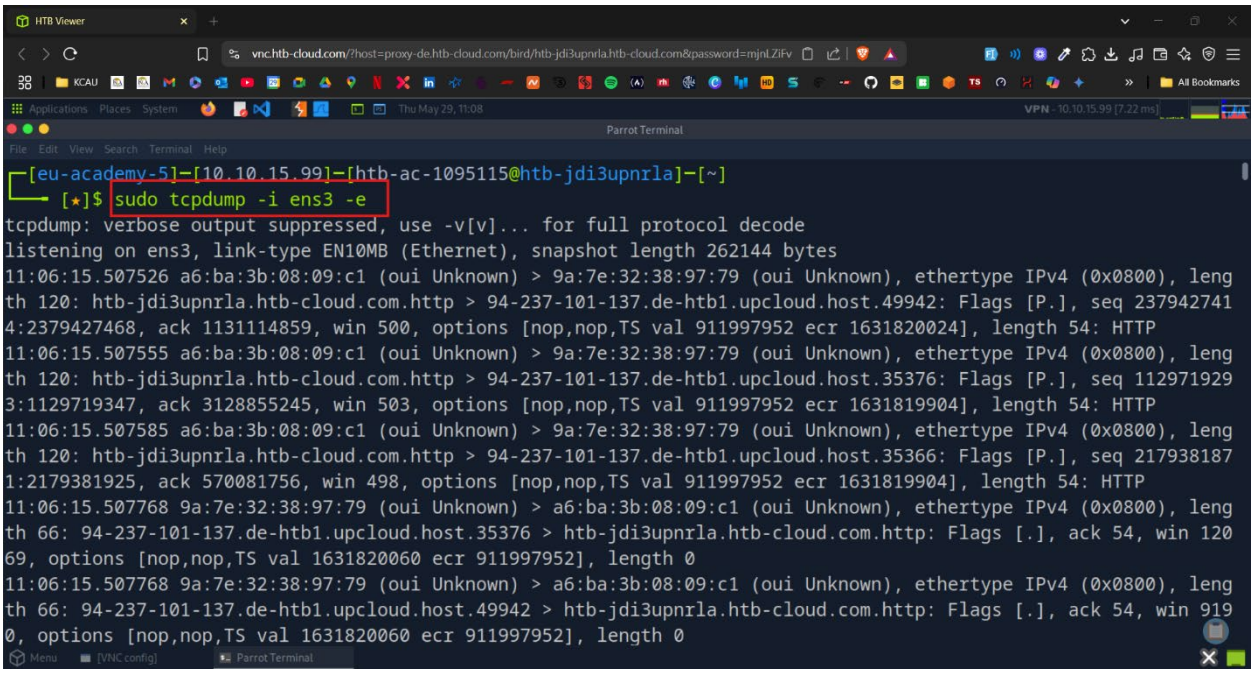




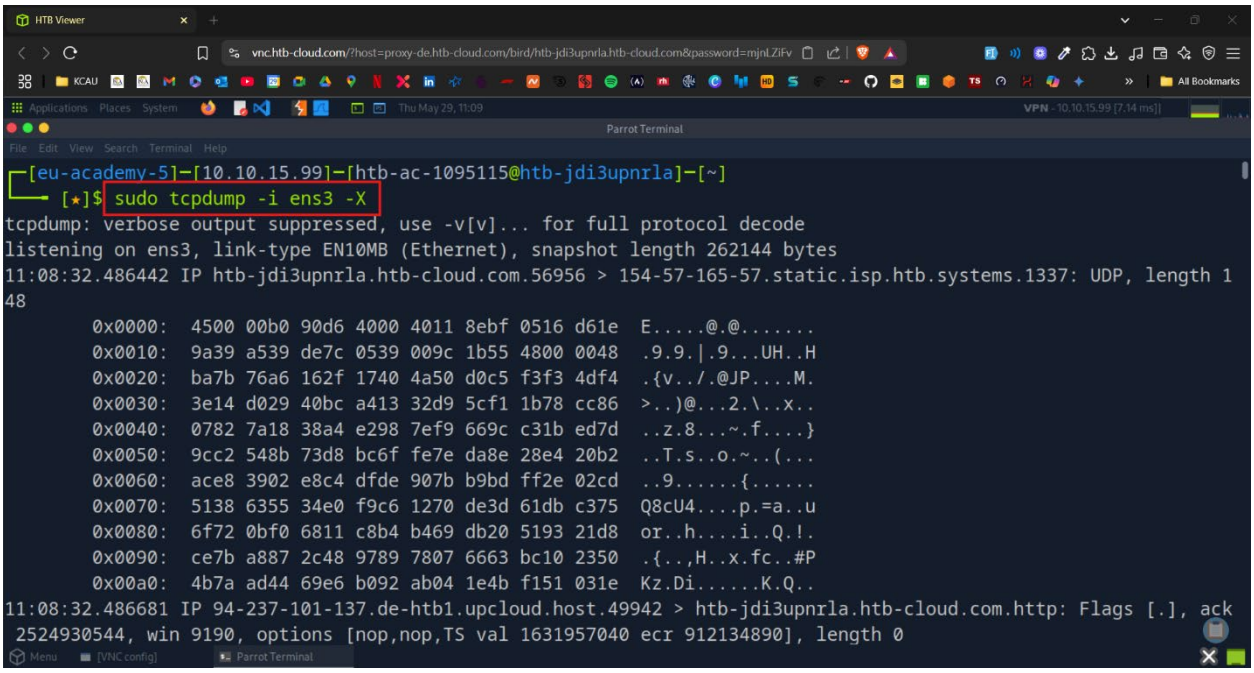
By issuing the ***-nn*** switches as seen below, we tell Tcpdump to refrain from resolving IP addresses and port numbers to their hostnames and common port names.



When utilizing the ***-e*** switch, we are tasking tcpdump to include the ethernet headers in the capture's output along with its regular content. We can see this worked by examining the output.

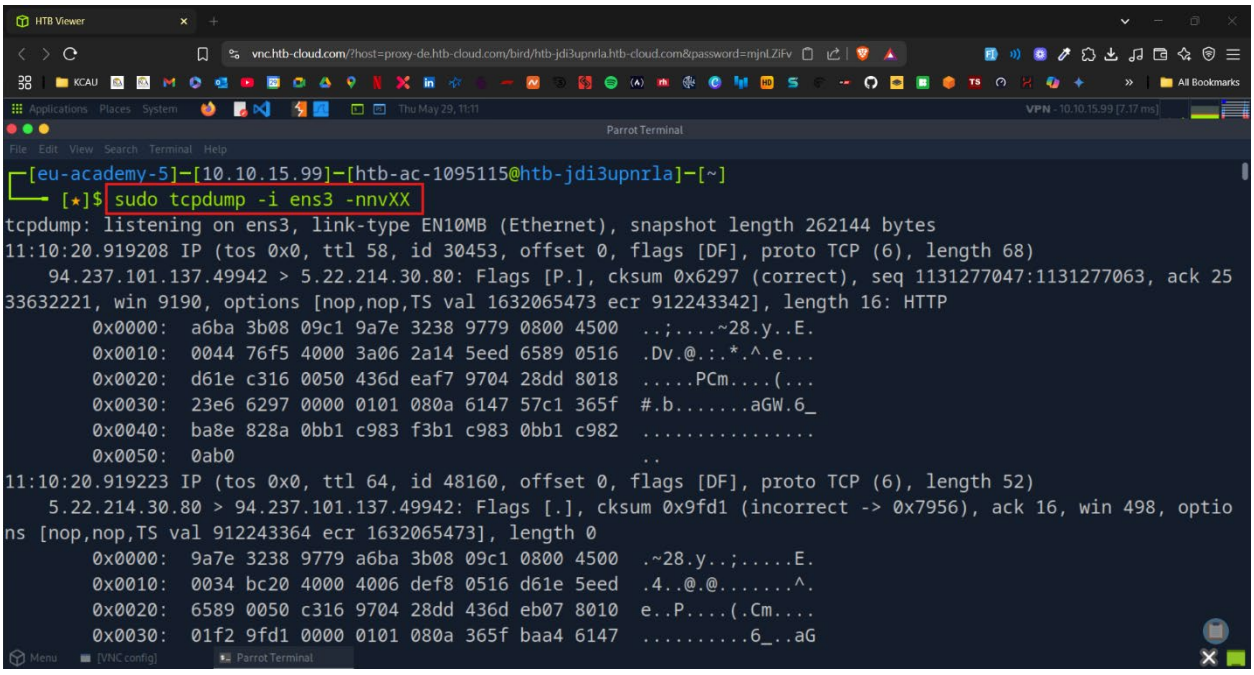


By issuing the ***-X*** switch, we can see the packet a bit clearer now. We get an ASCII output on the right to interpret anything in clear text that corresponds to the hexadecimal output on the left.



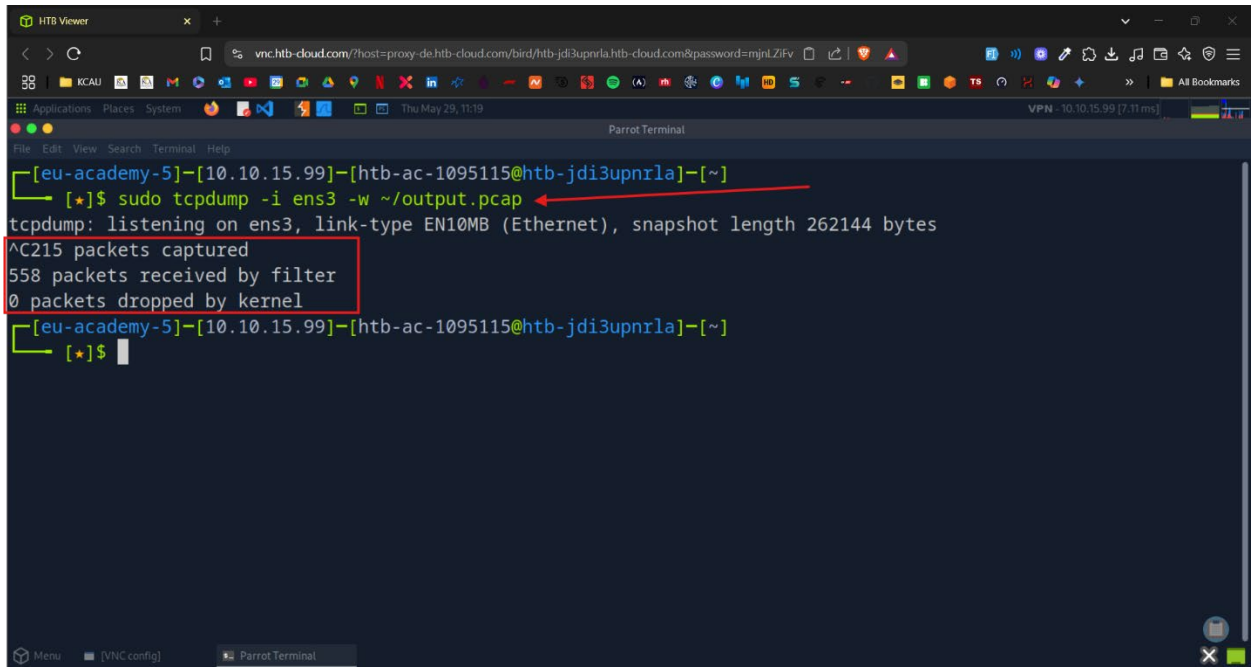
After trying the few different switches shown above, I learnt that we can also chain them e.g ***sudo tcpdump -i ens3 -nnvXX***

as shown below:



File Input/Output with Tcpdump

Using -w will write our capture to a file and in my case I named the file ***output.pcap***

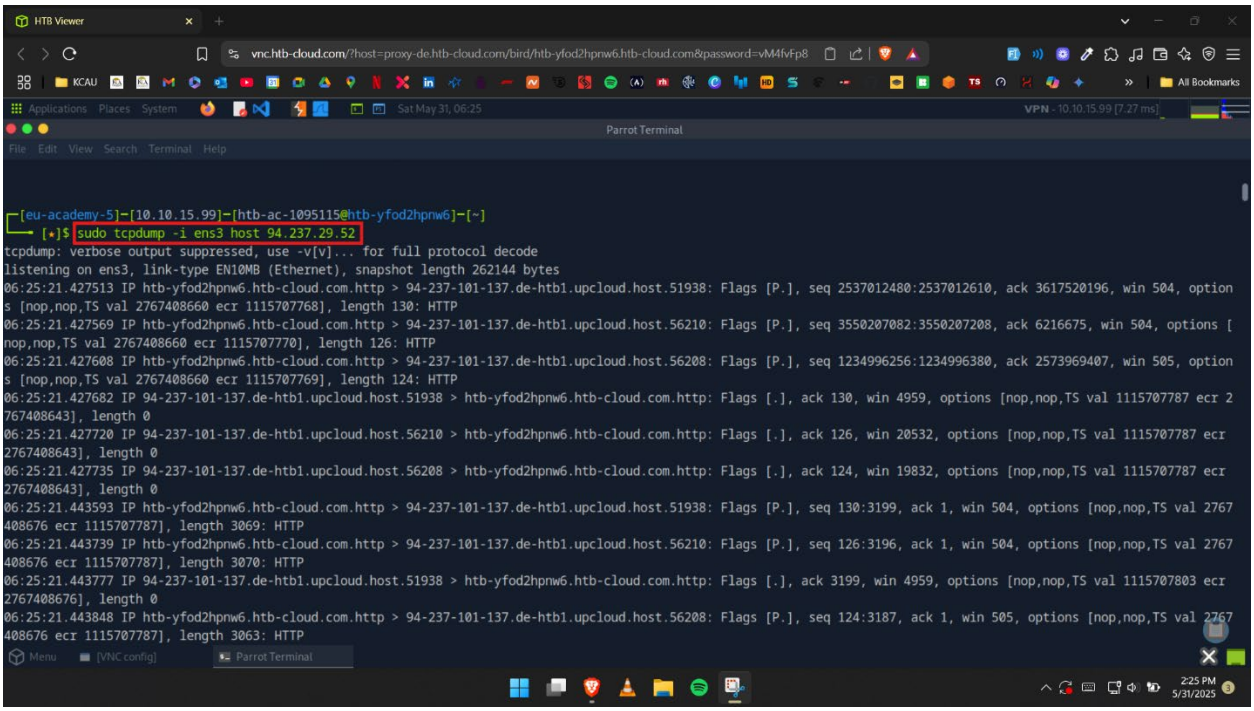


Tcpdump Packet Filtering

Tcpdump provides a robust and efficient way to parse the data included in our captures via packet filters.

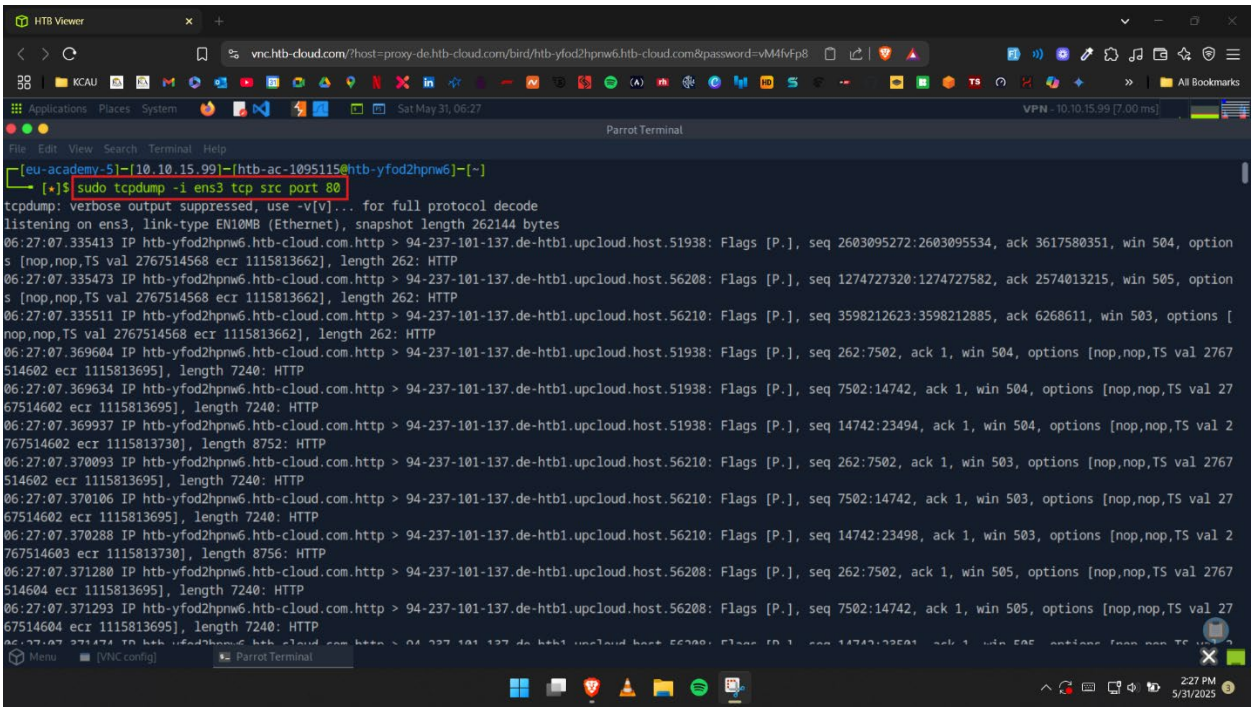
Host Filter

This filter is often used when we want to examine only a specific host or server. With this, we can identify with whom this host or server communicates and in which way.

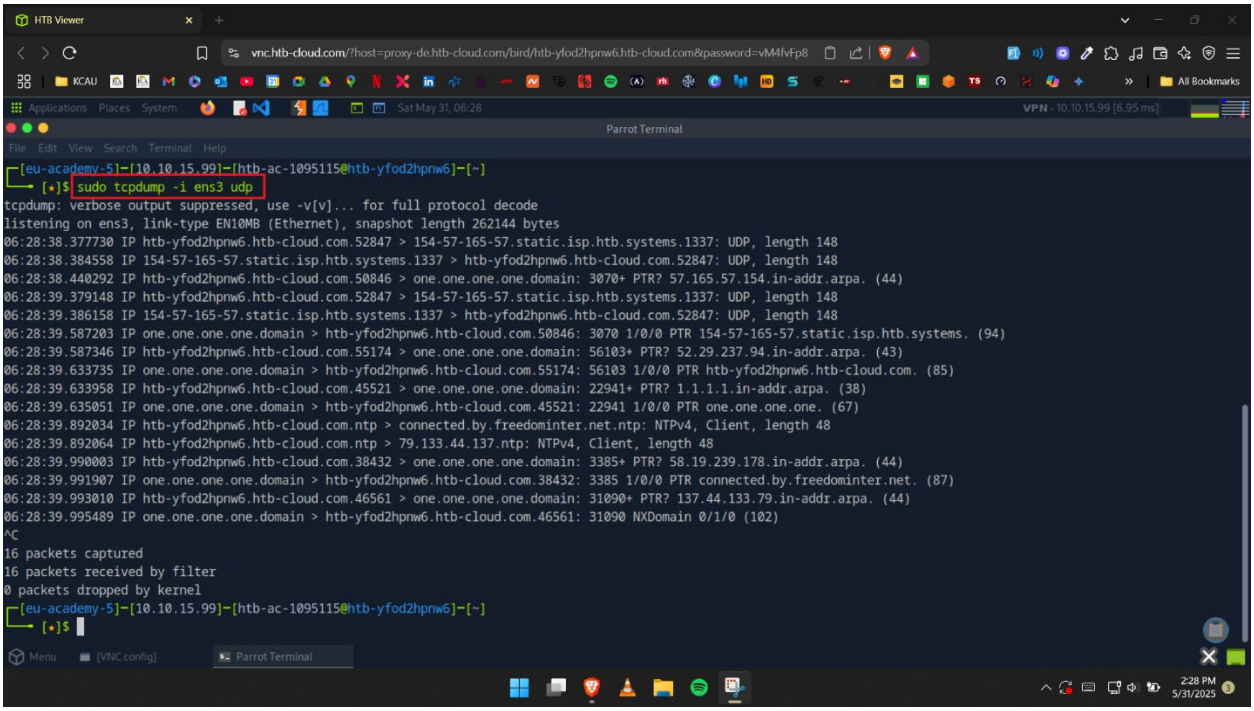


Source/Destination Filter

Source and destination allow us to work with the directions of communication. For example, in the last output, we have specified that our source host is ***94.237.29.52***, and only packets sent from this host will be intercepted. This can be done for ports, and network ranges as well. An example of this utilizing ***src port 80*** would look something like this:

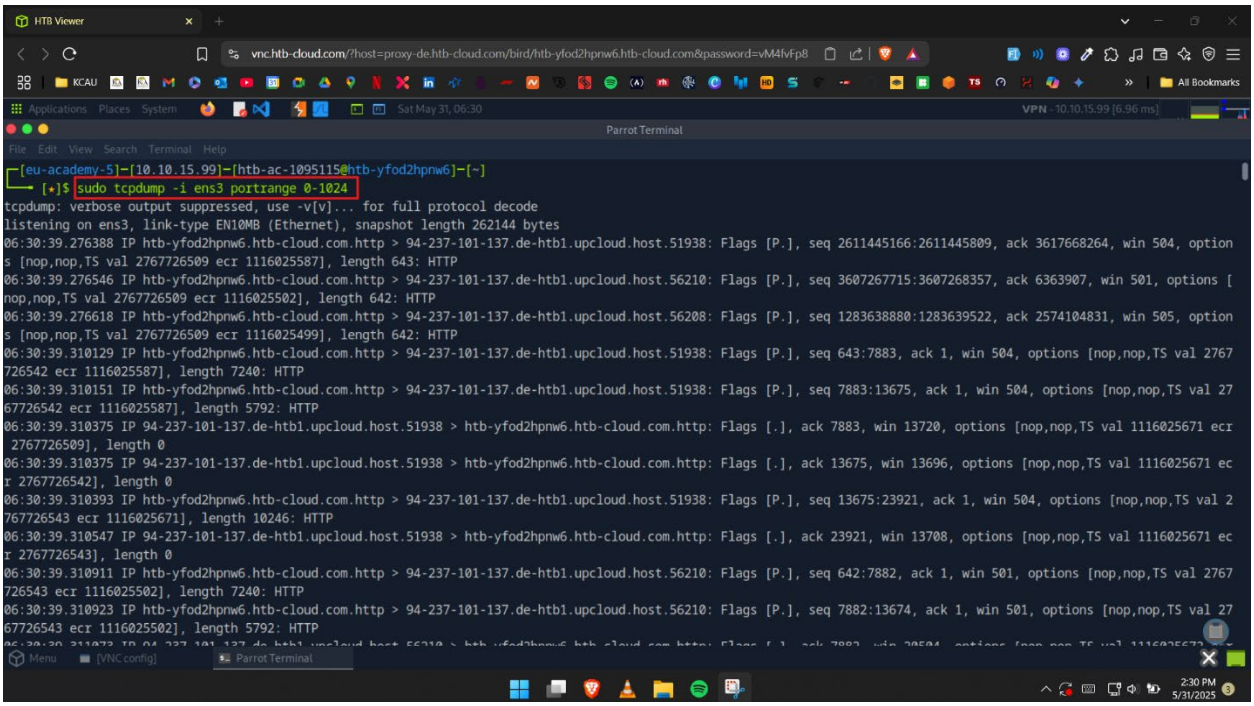


Protocol Filter - Common Name

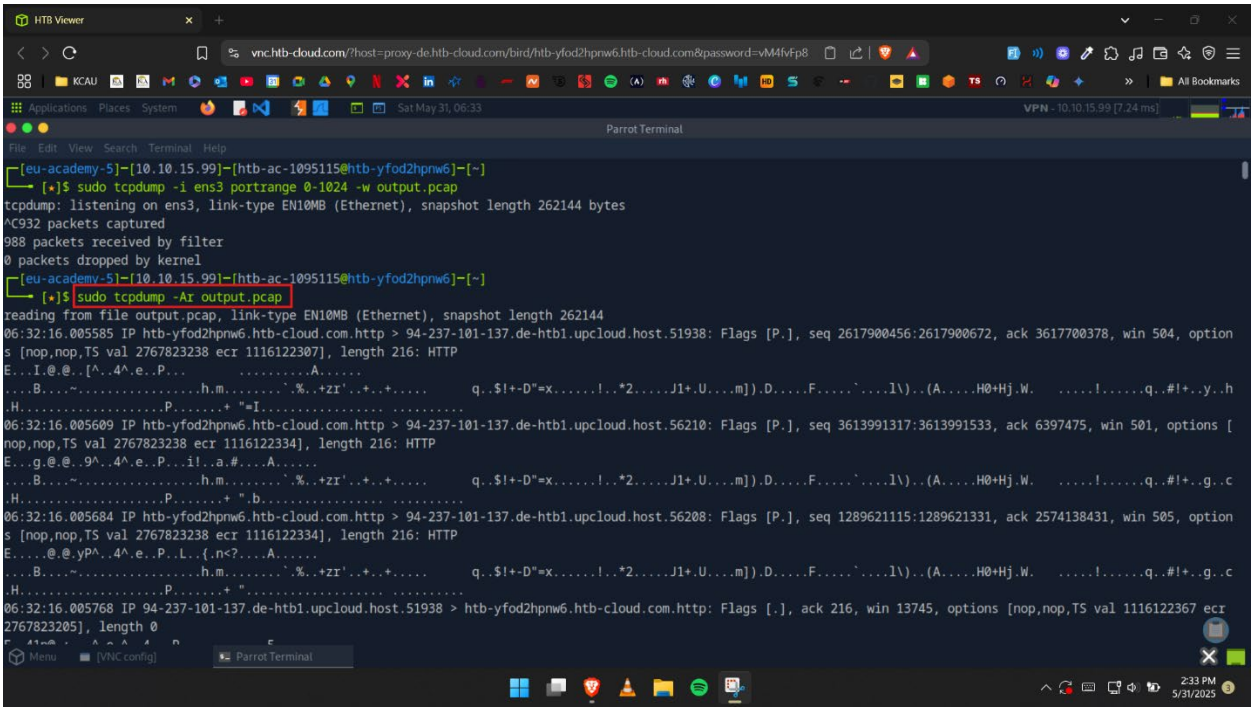


Port Range Filter

This allows us to everything from within the port range.

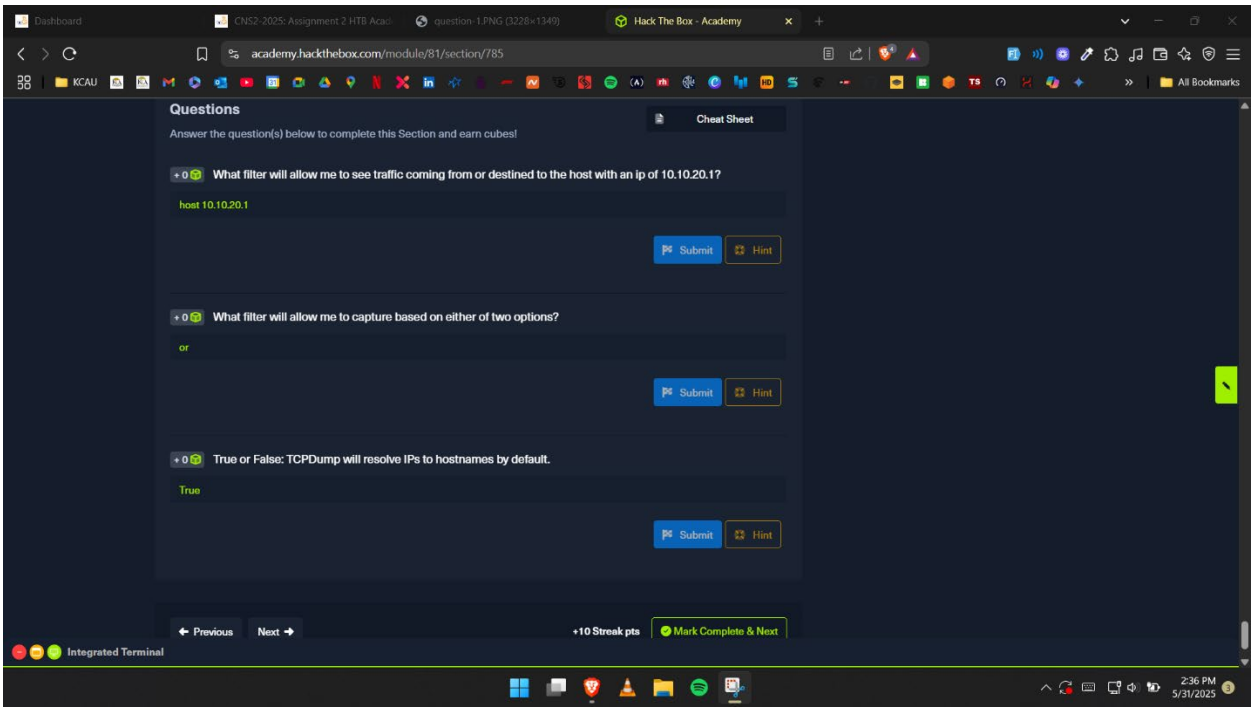


Tips and Tricks



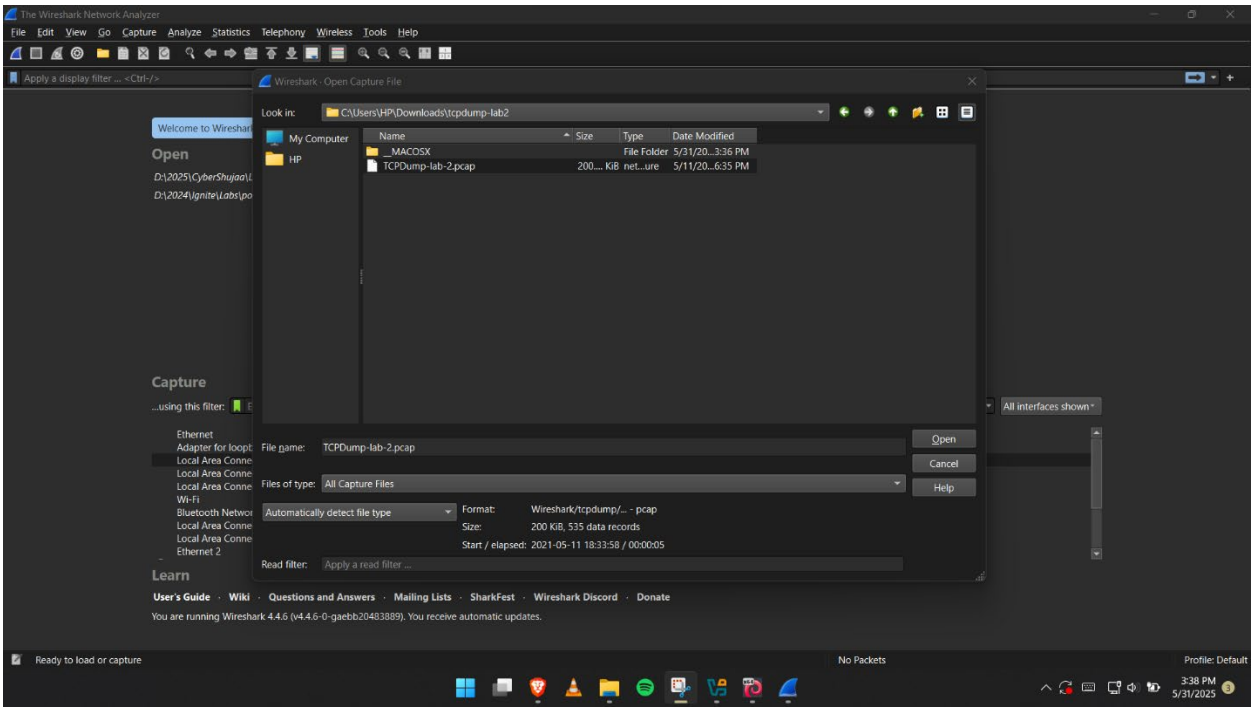
Notice how it has the ASCII values shown below each output line because of our use of -A. This can be helpful when quickly looking for something human-readable in the output.

Questions

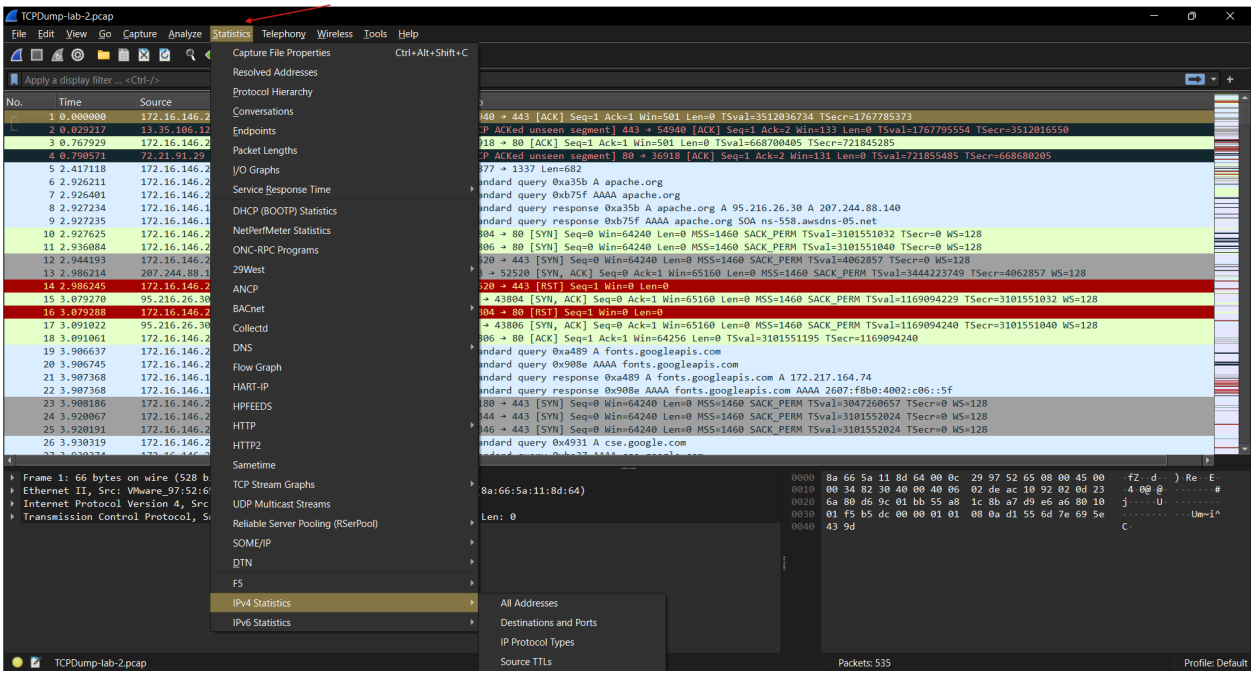


Analysis with Wireshark

Wireshark is a free and open-source network traffic analyzer much like tcpdump but with a graphical interface. Wireshark is multi-platform and capable of capturing live data off many different interface types (to include WiFi, USB, and Bluetooth) and saving the traffic to several different formats.

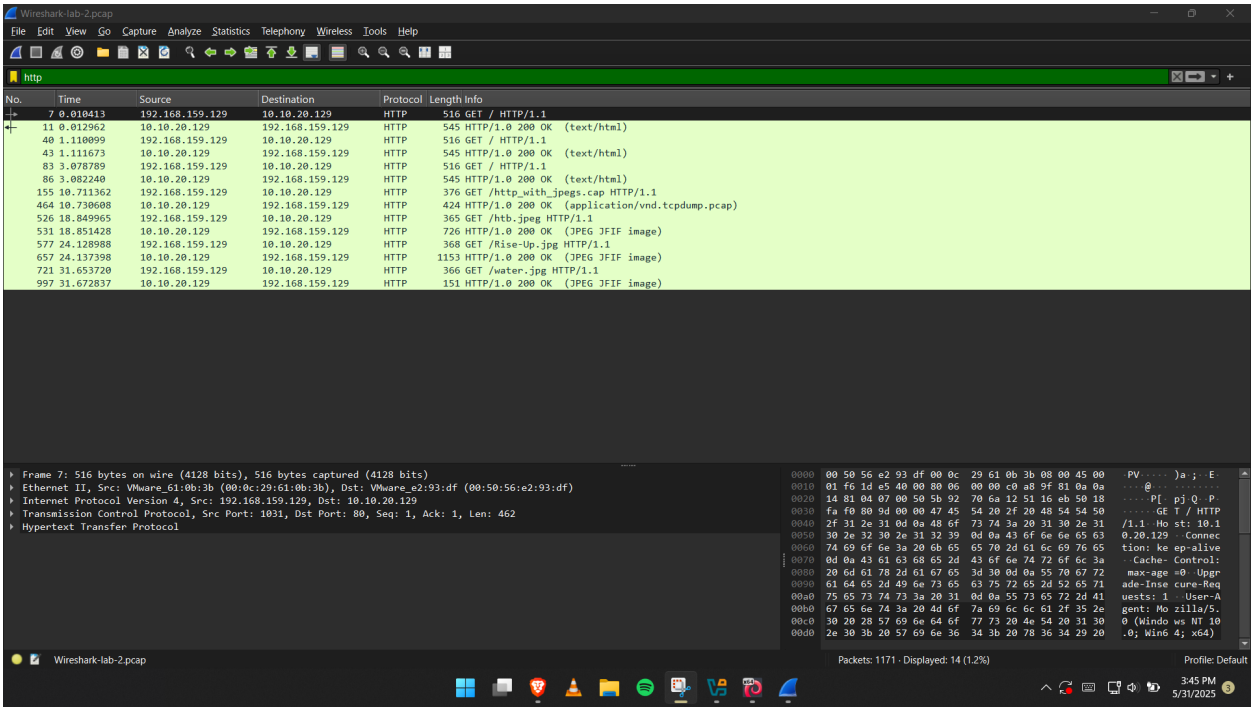


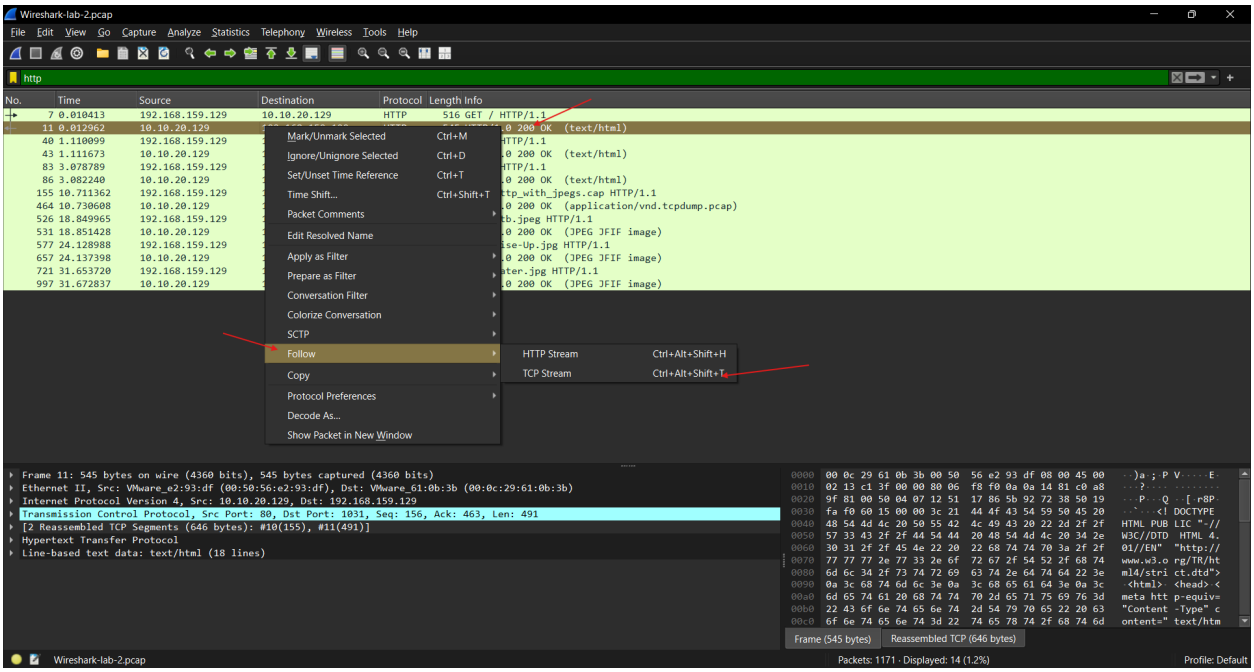
The Statistics tab provides us with great insight into the data we are examining.



Following TCP Streams

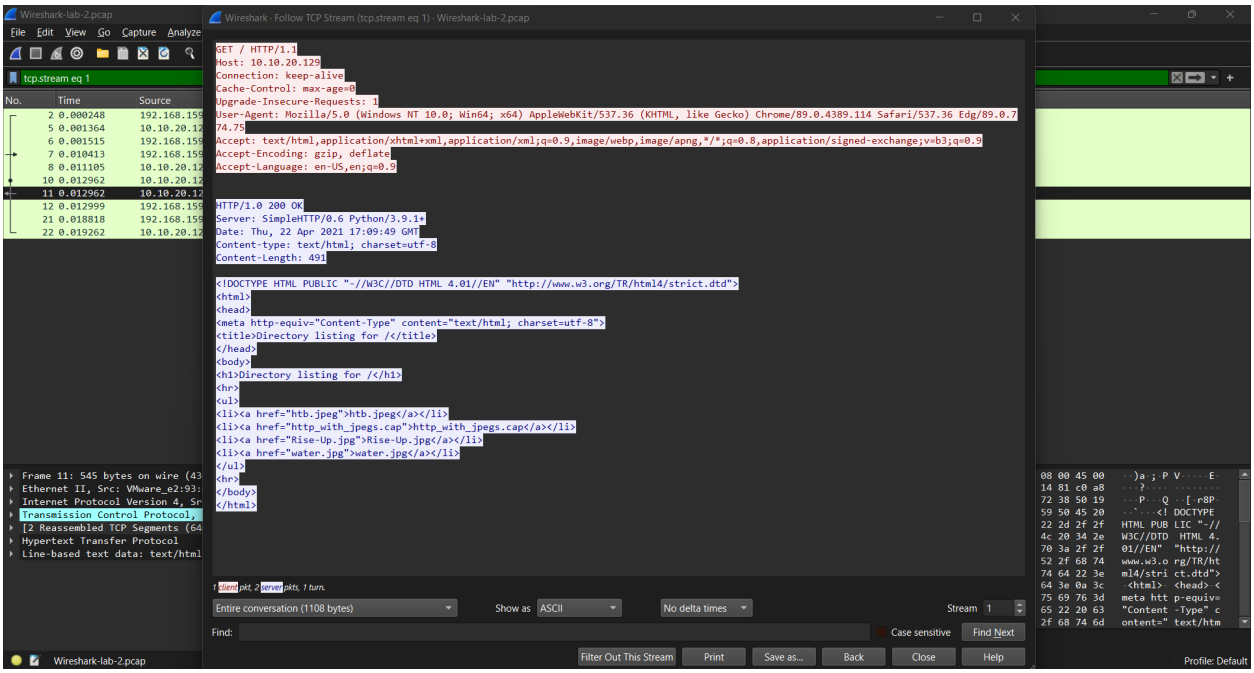
Wireshark can stitch TCP packets back together to recreate the entire stream in a readable format. This ability also allows us to pull data (images, files, etc.) out of the capture. This works for almost any protocol that utilizes TCP as a transport mechanism.





Filter For A Specific TCP Stream

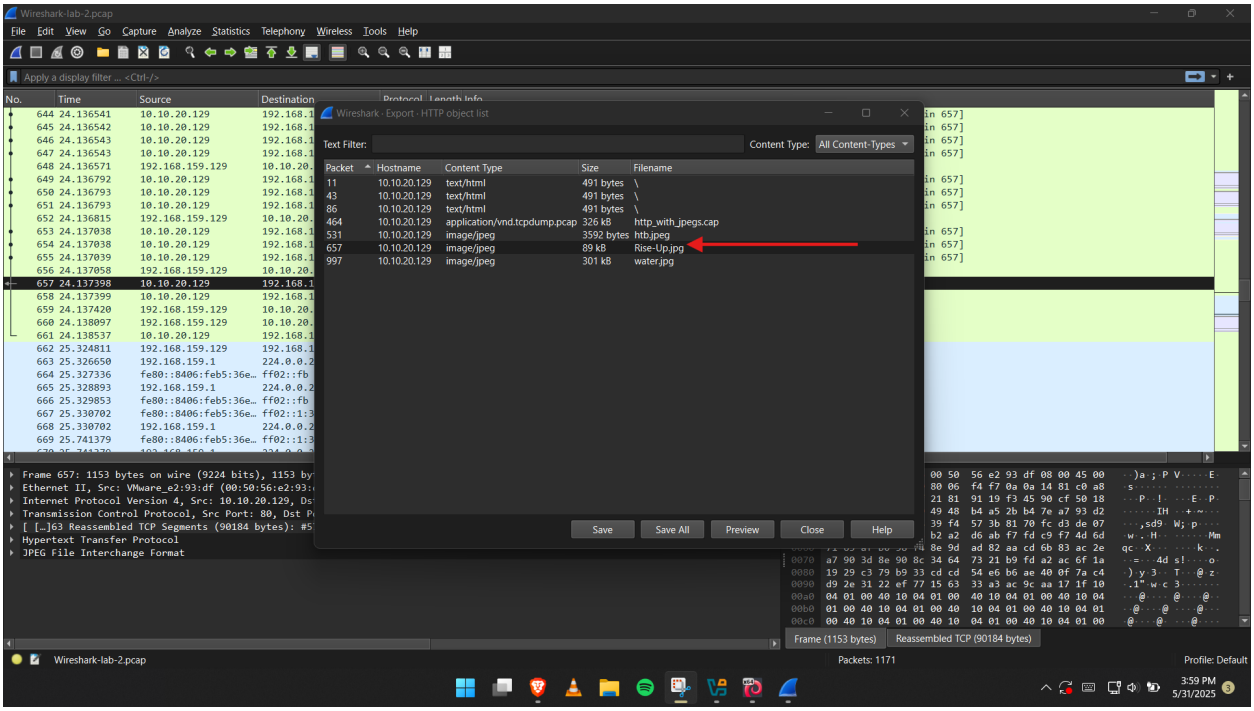
Alternatively, we can utilize the filter ***tcp.stream eq …*** to find and track conversations captured in the pcap file for example:

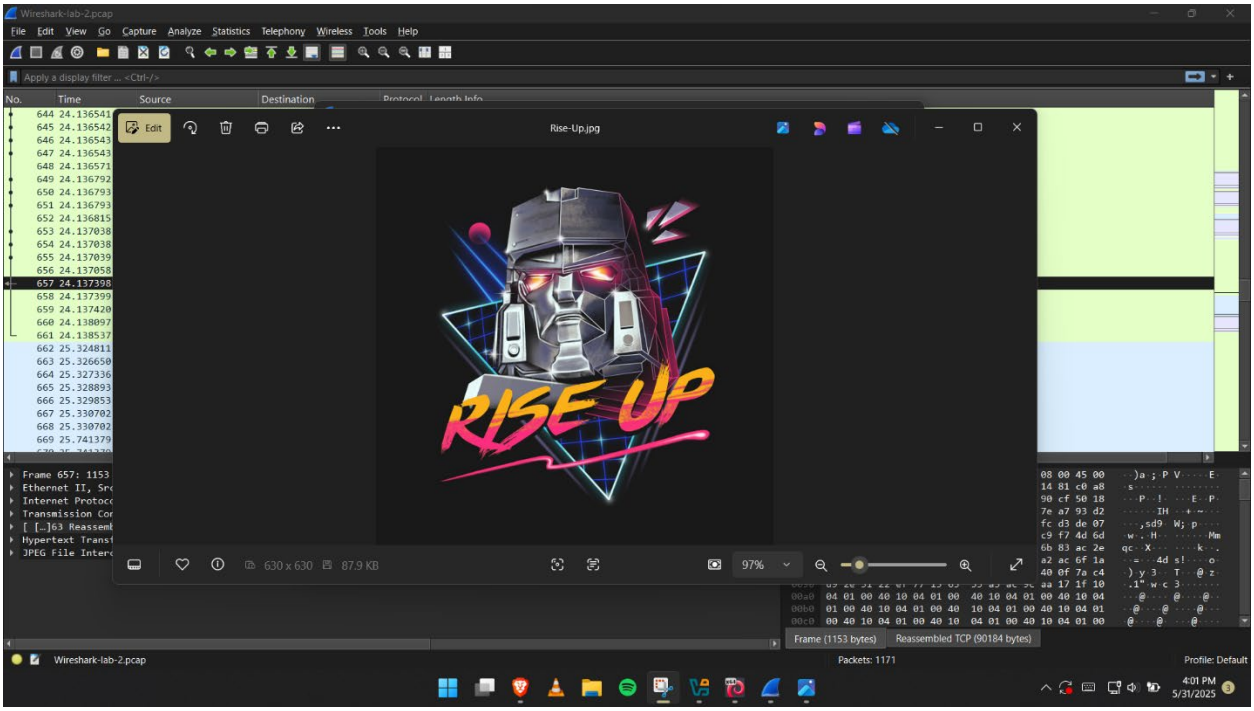


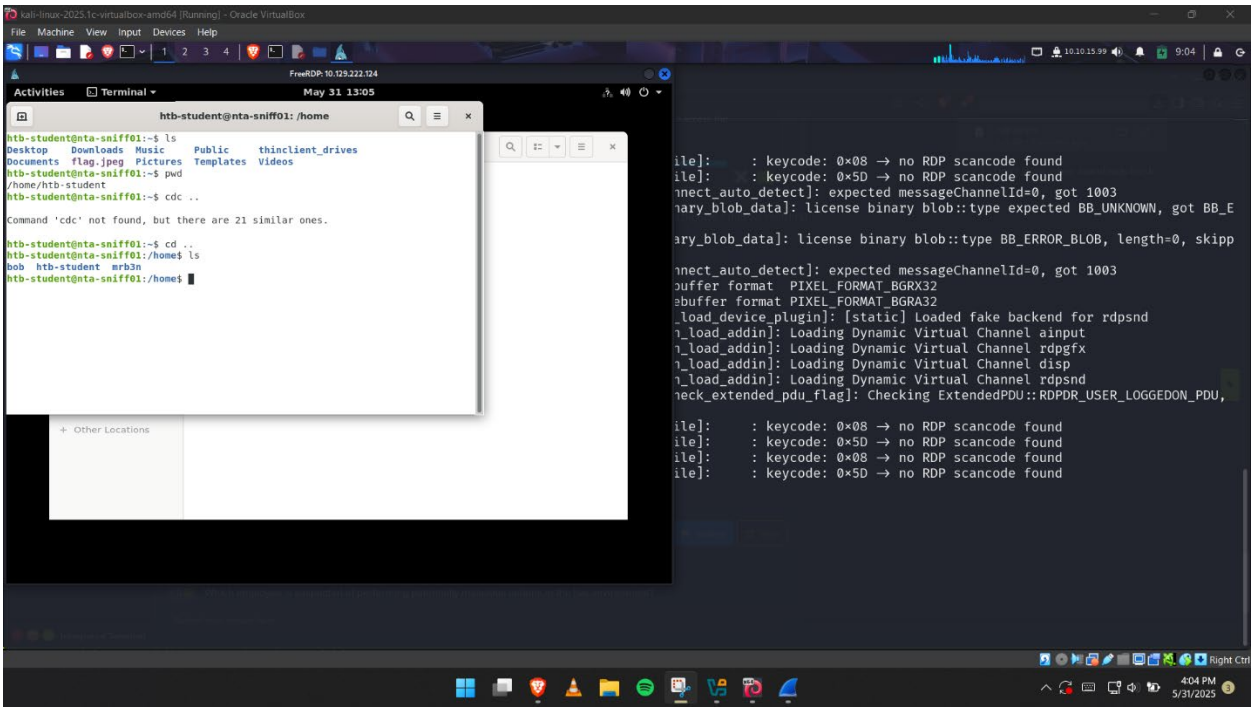
Packet Inception, Dissecting Network Traffic With Wireshark

Tasks

i. Open a pre-captured file (HTTP extraction).
ii. Filter the results.
iii. Follow the stream and extract the item(s) found.


Wireshark-lab-2.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Wireshark · Export · HTTP object list
Text Filter:
Content Type: All Content-Types
Packet Hostname Content Type Size Filename
11 10.10.20.129 text/html 491 bytes \
43 10.10.20.129 text/html 491 bytes \
86 10.10.20.129 text/html 491 bytes \
464 10.10.20.129 application/vnd.tcpdump.pcap 326 kB http_with_jpegs.cap
531 10.10.20.129 image/jpeg 3592 bytes htb.jpeg
657 10.10.20.129 image/jpeg 89 kB Rise-Up.jpg
997 10.10.20.129 image/jpeg 301 kB water.jpg
Save
Save All
Preview
Close
Help
Frame 657: 1153 bytes on wire (9224 bits), 1153 by
Ethernet II, Src: VMware_e2:93:df (00:50:56:e2:93:
Internet Protocol Version 4, Src: 10.10.20.129, Ds
Transmission Control Protocol, Src Port: 80, Dst P
[[…]63 Reassembled TCP Segments (90184 bytes): #5
Hypertext Transfer Protocol
JPEG File Interchange Format
Frame (1153 bytes)
Reassembled TCP (90184 bytes)
Packets: 1171
Profile: Default
3:59 PM
5/31/2025


Wireshark-lab-2.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Edit
Rise-Up.jpg
RISE UP
630 x 630
87.9 KB
97%
Frame 657: 1153
Ethernet II, Src
Internet Protocc
Transmission Cor
[[…]63 Reassembl
Hypertext Transf
JPEG File Interc
Frame (1153 bytes)
Reassembled TCP (90184 bytes)
Packets: 1171
Profile: Default
4:01 PM
5/31/2025



Questions

Answer the question(s) below to complete this Section and earn cubes!

Cheat Sheet

Download VPN Connection File

Target(s): 10.129.222.124 (ACADEMY-NTA-SNIFF01)

Life Left: 17 minute(s) + Terminate

RDP to 10.129.222.124 (ACADEMY-NTA-SNIFF01) with user "htb-student" and password "HTB_@cademy_stdnt!"

+2 What was the filename of the image that contained a certain Transformer Leader? (name.filetype)

Rise-up.jpg

Submit Hint

+0 Which employee is suspected of performing potentially malicious actions in the live environment?

bob

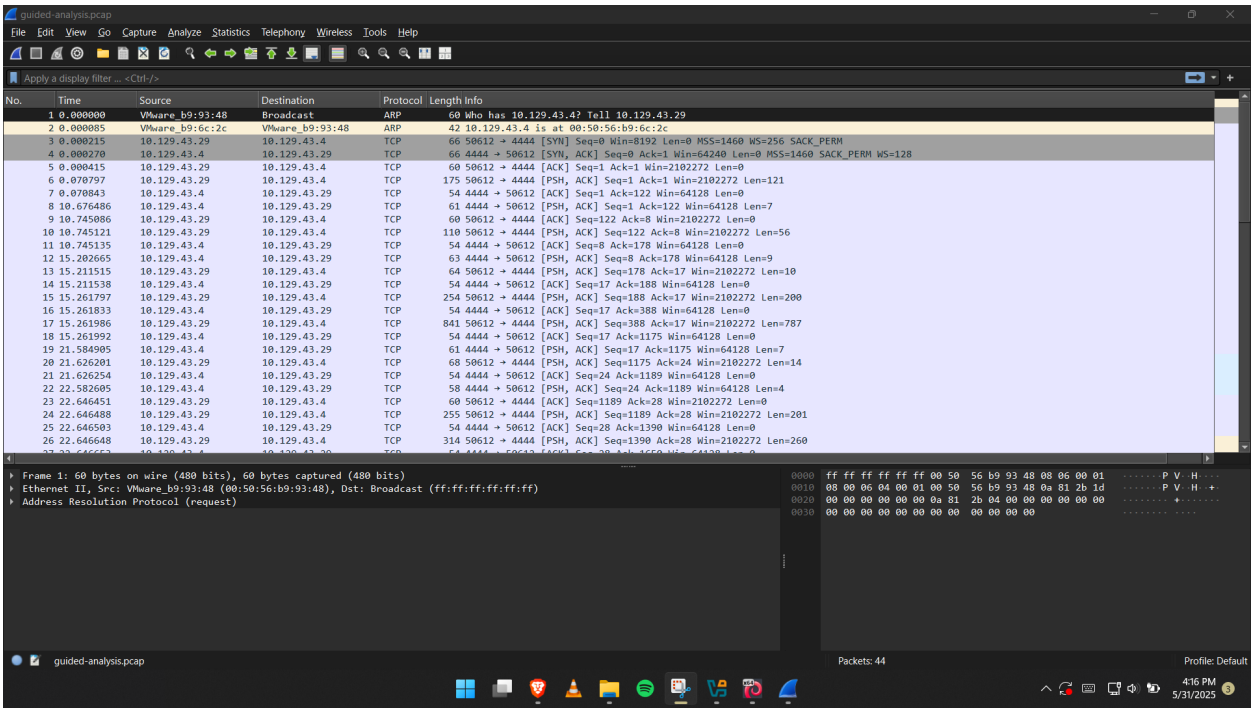
Submit Hint

Previous Next

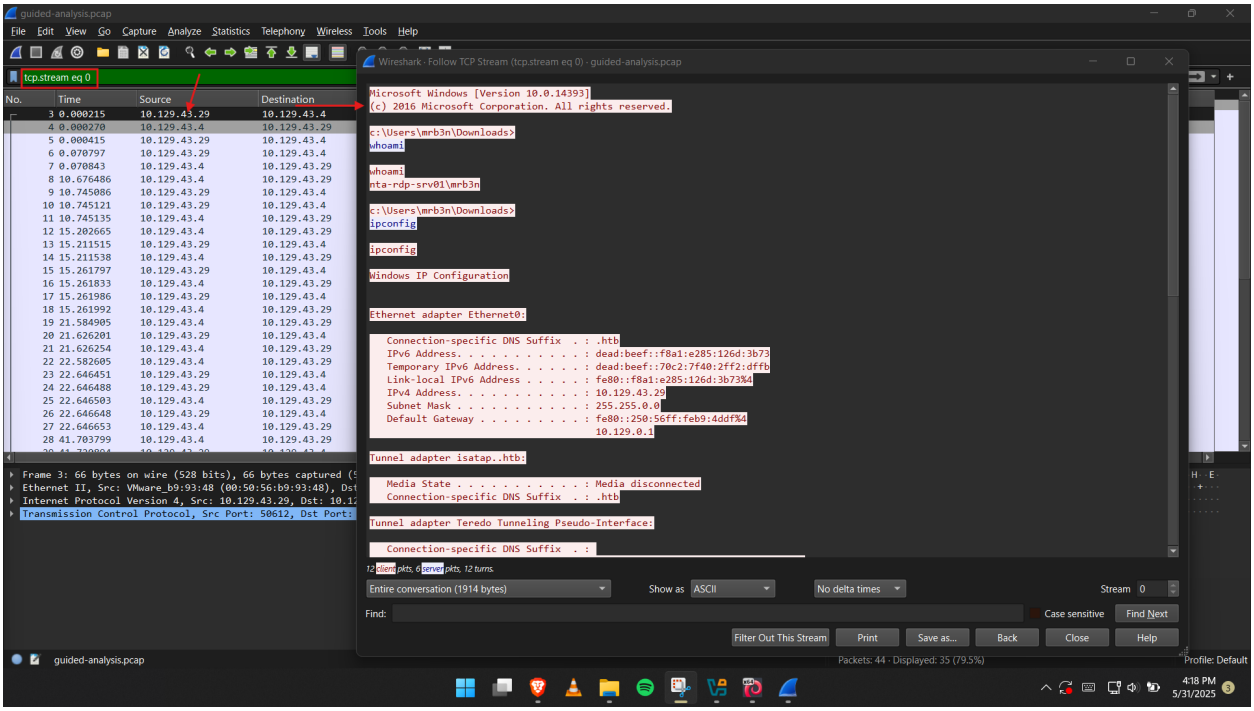
Mark Complete & Next

Guided Lab: Traffic Analysis Workflow

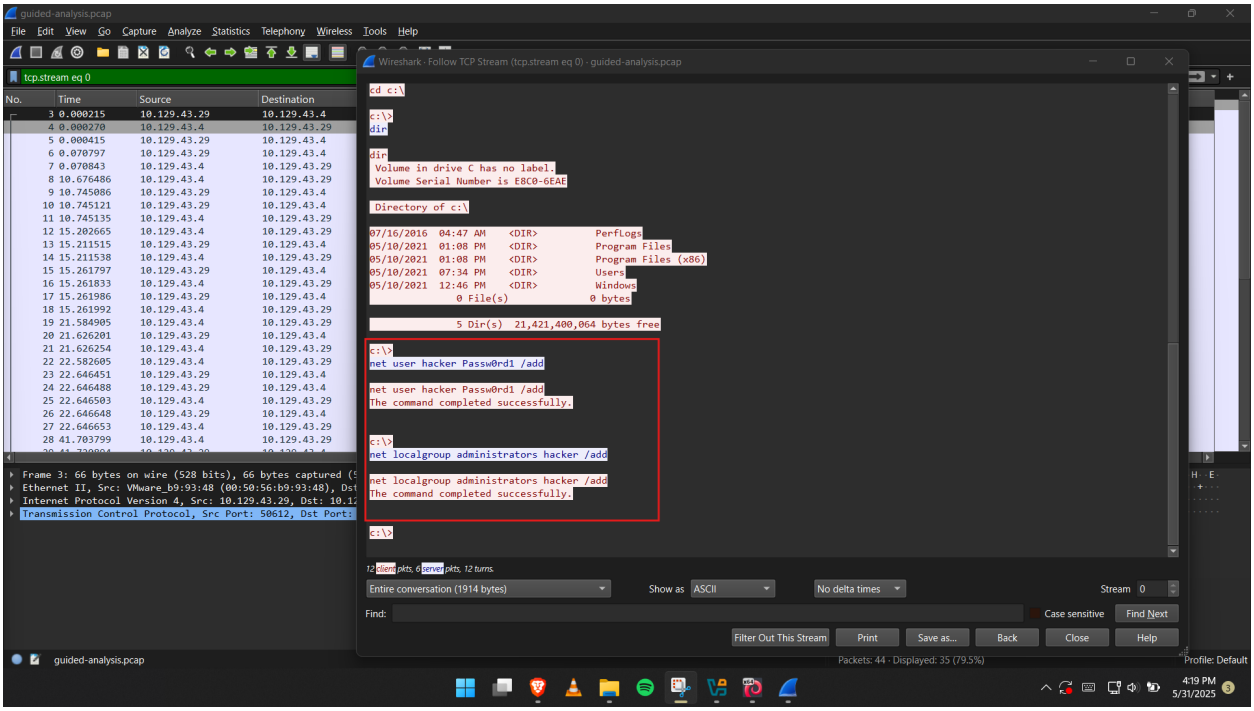
This is an attempt to utilize the concepts from the Analysis Process sections to complete an analysis of the guided-analysis.zip provided in the optional resources and live traffic from the academy network.

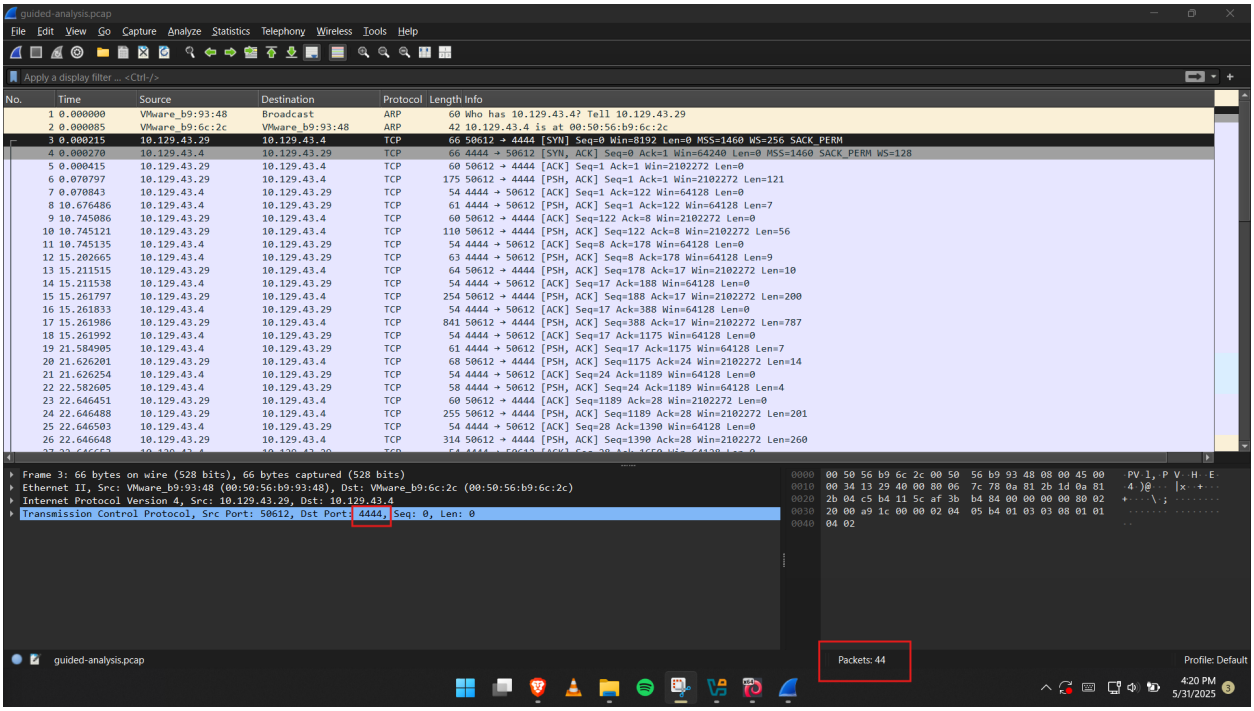


(Overview of the ***guided-analysis.pcap*** file)

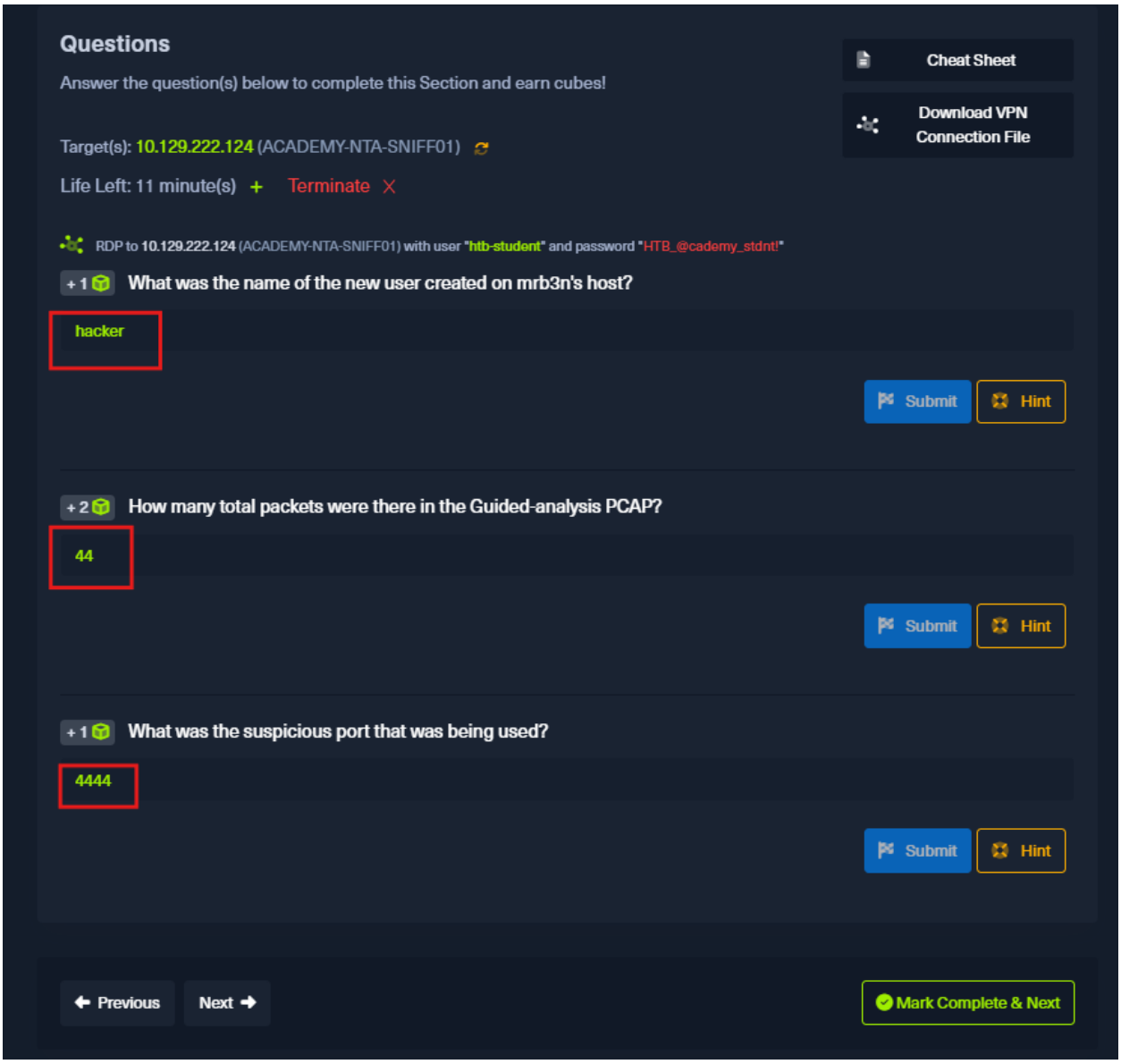


In the above screenshot, I looked at the TCP traffic where I utilized the display filter ***!udp && !arp*** . the next thing I did was follow the TCP stream and examined it and look at what we got, it appears that someone is performing basic recon of the host. They are issuing commands like ***whoami***, ***ipconfig***, ***dir***. We also see that they created an account ***hacker***



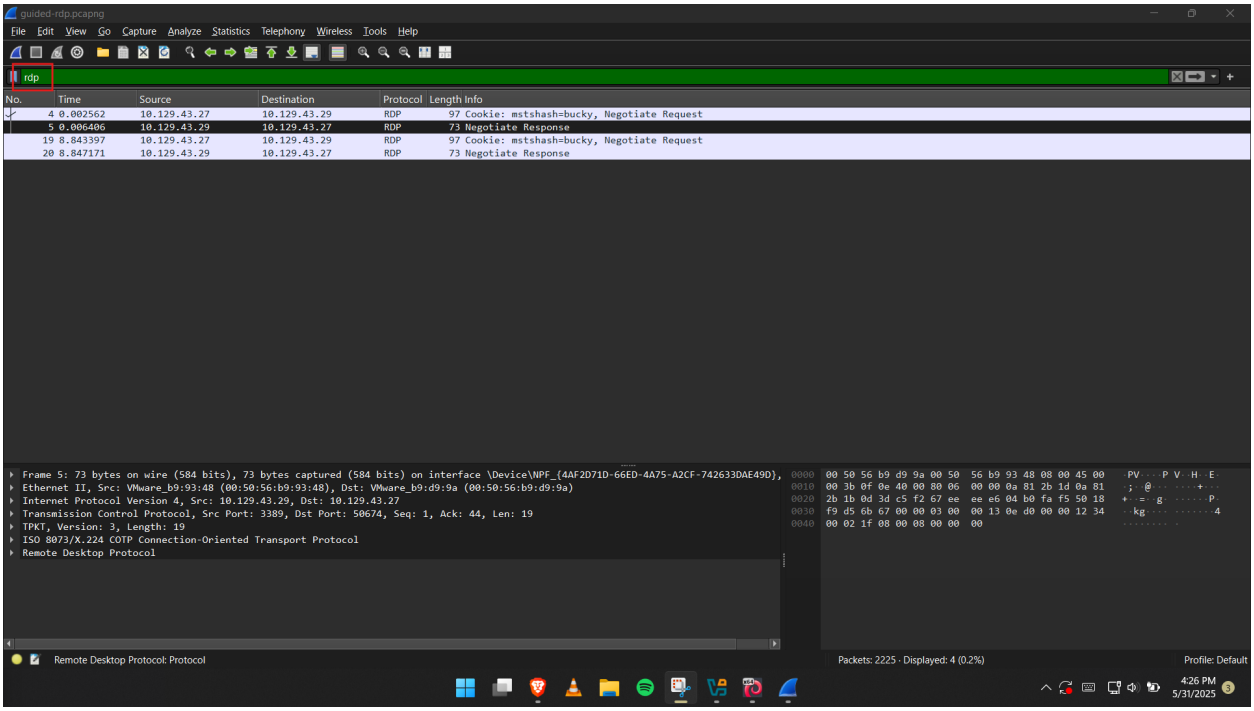


We see the that the total packets of the file are ***44*** and found the suspicious port that was being used as port ***4444***. Which were the answers to the task questions below.

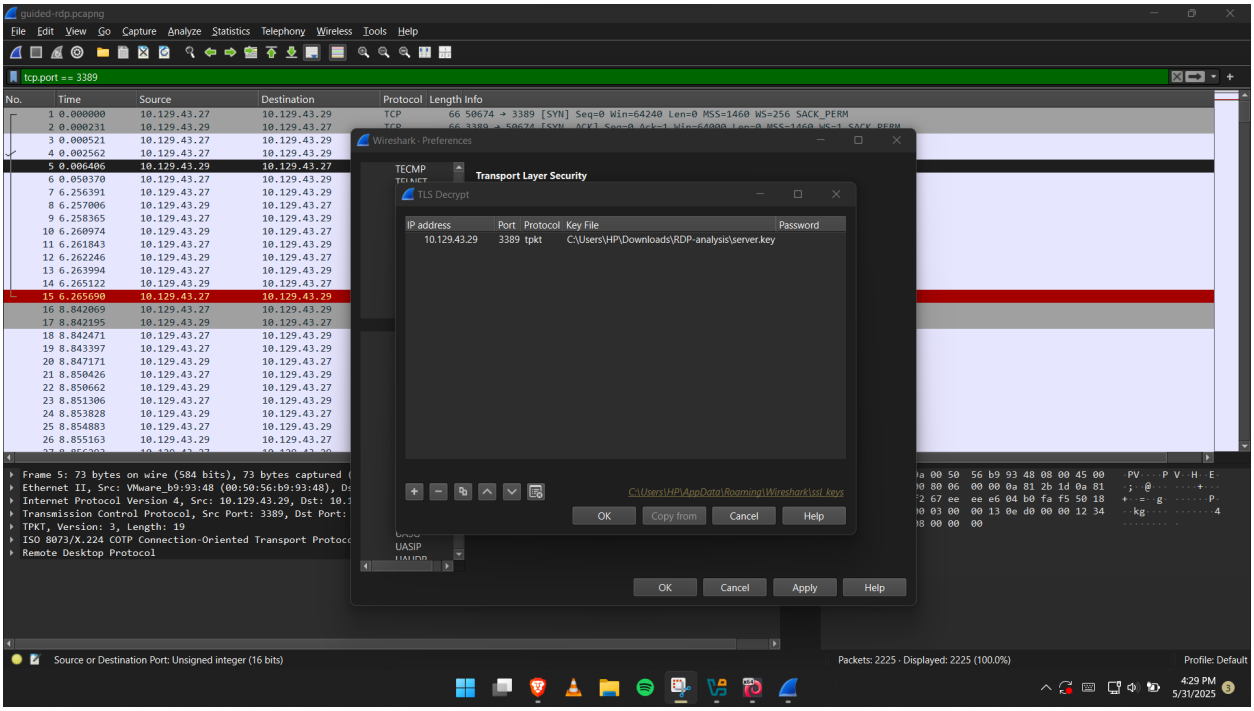


Decrypting RDP connections

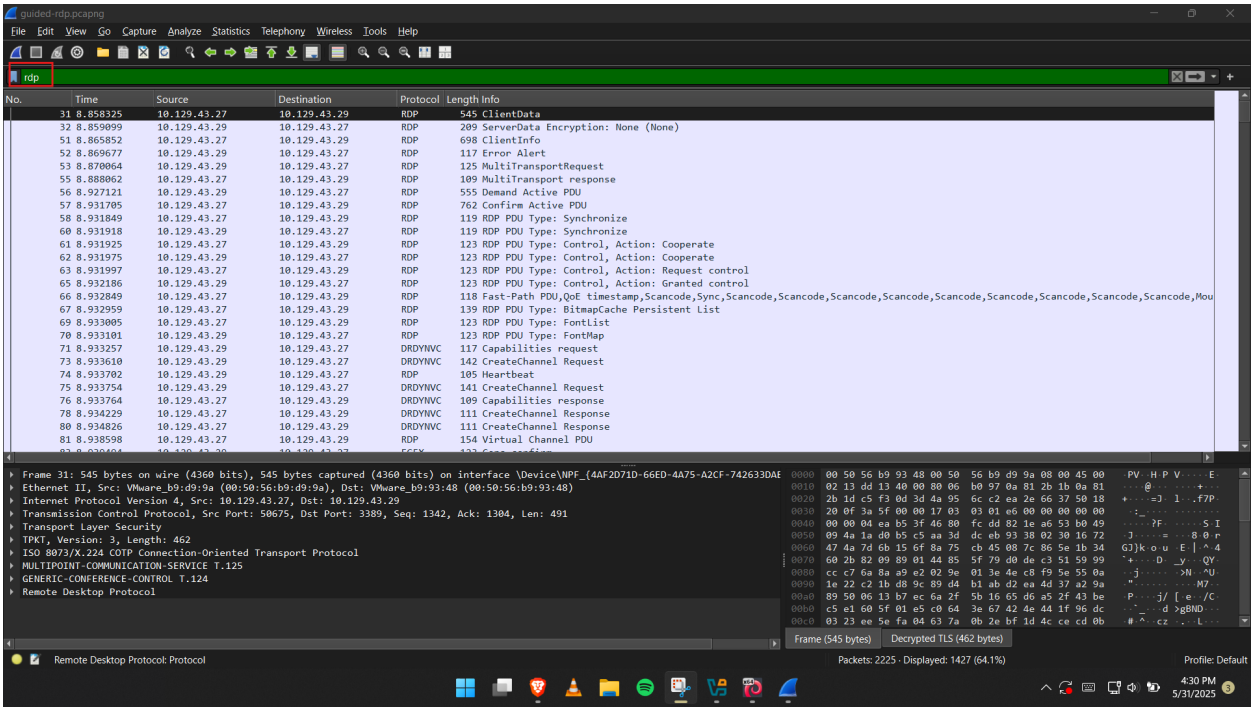
Here we attempt to analyze the RDP-analysis.zip as tasked and first thing first we download the resource and then open the file in Wireshark. Our focus is on RDP so we filtered only ***rdp.***

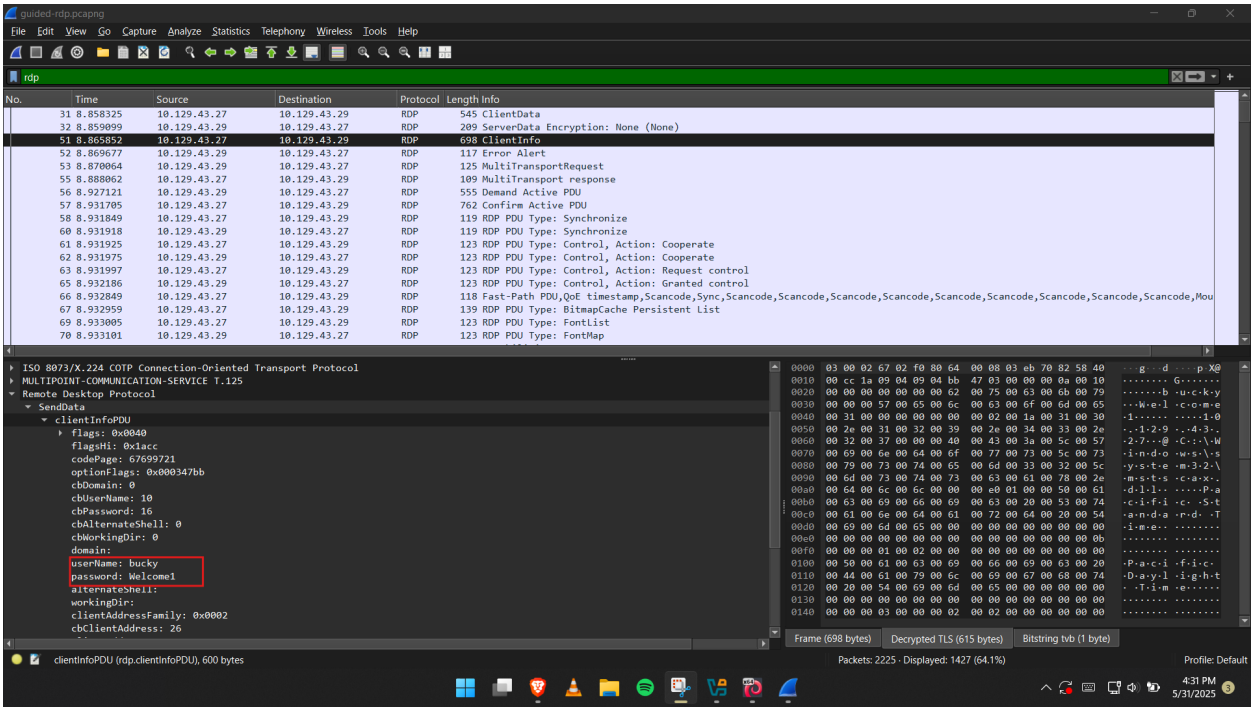


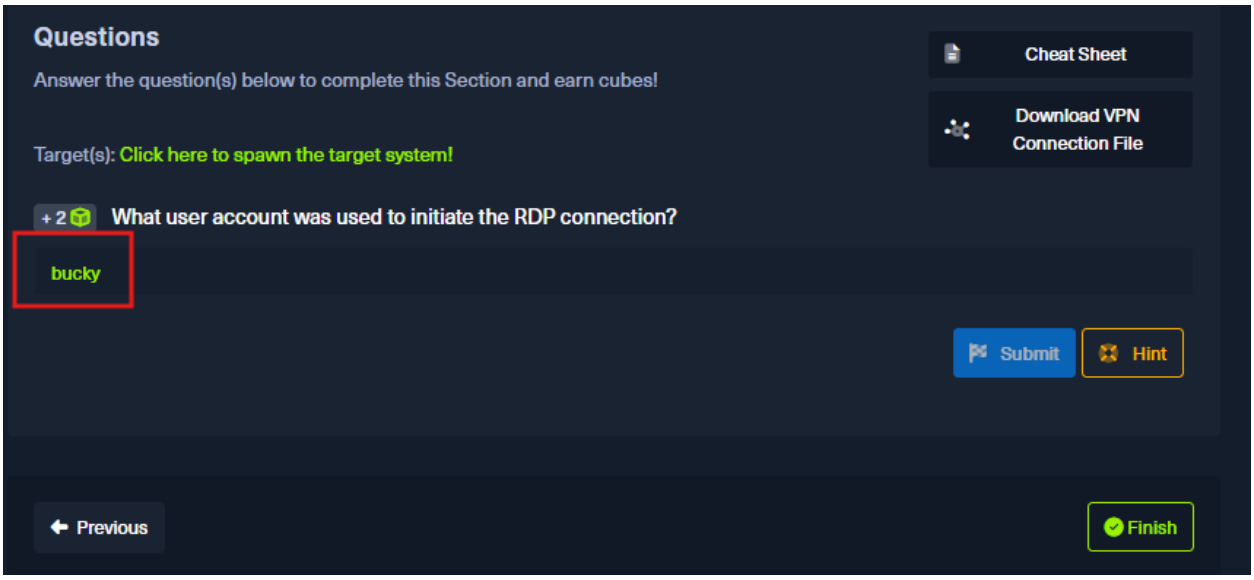
Filter on ***port 3389*** to determine if any RDP traffic encrypted or otherwise exists and provided the RDP-key to Wireshark so it can decrypt the traffic.



We now able to see more traffic which we couldn't see before as shown in the previous screenshot with ***rdp*** filter.


guided-rdp.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
rdp
No. Time Source Destination Protocol Length Info
31 8.858325 10.129.43.27 10.129.43.29 RDP 545 ClientData
32 8.859099 10.129.43.29 10.129.43.27 RDP 209 ServerData Encryption: None (None)
51 8.865852 10.129.43.27 10.129.43.29 RDP 698 ClientInfo
52 8.869677 10.129.43.29 10.129.43.27 RDP 117 Error Alert
53 8.870064 10.129.43.29 10.129.43.27 RDP 125 MultiTransportRequest
55 8.888062 10.129.43.27 10.129.43.29 RDP 109 MultiTransport response
56 8.927121 10.129.43.29 10.129.43.27 RDP 555 Demand Active PDU
57 8.931705 10.129.43.27 10.129.43.29 RDP 762 Confirm Active PDU
58 8.931849 10.129.43.27 10.129.43.29 RDP 119 RDP PDU Type: Synchronize
60 8.931918 10.129.43.29 10.129.43.27 RDP 119 RDP PDU Type: Synchronize
61 8.931925 10.129.43.27 10.129.43.29 RDP 123 RDP PDU Type: Control, Action: Cooperate
62 8.931975 10.129.43.29 10.129.43.27 RDP 123 RDP PDU Type: Control, Action: Cooperate
63 8.931997 10.129.43.27 10.129.43.29 RDP 123 RDP PDU Type: Control, Action: Request control
65 8.932186 10.129.43.29 10.129.43.27 RDP 123 RDP PDU Type: Control, Action: Granted control
66 8.932849 10.129.43.27 10.129.43.29 RDP 118 Fast-Path PDU,QoE timestamp,Scancode,Sync,Scancode,Scancode,Scancode,Scancode,Scancode,Scancode,Scancode,Scancode,Scancode,Mou
67 8.932959 10.129.43.29 10.129.43.27 RDP 139 RDP PDU Type: BitmapCache Persistent List
69 8.933005 10.129.43.27 10.129.43.29 RDP 123 RDP PDU Type: FontList
70 8.933101 10.129.43.29 10.129.43.27 RDP 123 RDP PDU Type: FontMap
71 8.933257 10.129.43.29 10.129.43.27 DRDYNVC 117 Capabilities request
73 8.933610 10.129.43.29 10.129.43.27 DRDYNVC 142 CreateChannel Request
74 8.933702 10.129.43.29 10.129.43.27 RDP 105 Heartbeat
75 8.933754 10.129.43.29 10.129.43.27 DRDYNVC 141 CreateChannel Request
76 8.933764 10.129.43.27 10.129.43.29 DRDYNVC 109 Capabilities response
78 8.934229 10.129.43.27 10.129.43.29 DRDYNVC 111 CreateChannel Response
80 8.934826 10.129.43.27 10.129.43.29 DRDYNVC 111 CreateChannel Response
81 8.938598 10.129.43.27 10.129.43.29 RDP 154 Virtual Channel PDU
Frame 31: 545 bytes on wire (4360 bits), 545 bytes captured (4360 bits) on interface \Device\NPF_{4AF2D71D-66ED-4A75-A2CF-742633DAE
Ethernet II, Src: VMware_b9:d9:9a (00:50:56:b9:d9:9a), Dst: VMware_b9:93:48 (00:50:56:b9:93:48)
Internet Protocol Version 4, Src: 10.129.43.27, Dst: 10.129.43.29
Transmission Control Protocol, Src Port: 50675, Dst Port: 3389, Seq: 1342, Ack: 1304, Len: 491
Transport Layer Security
TPKT, Version: 3, Length: 462
ISO 8073/X.224 COTP Connection-Oriented Transport Protocol
MULTIPOINT-COMMUNICATION-SERVICE T.125
GENERIC-CONFERENCE-CONTROL T.124
Remote Desktop Protocol
0000 00 50 56 b9 93 48 00 50 56 b9 d9 9a 08 00 45 00
0010 02 13 dd 13 40 00 80 06 b0 97 0a 81 2b 1b 0a 81
0020 2b 1d c5 f3 0d 3d 4a 95 6c c2 ea 2e 66 37 50 18
0030 20 0f 3a 5f 00 00 17 03 03 01 e6 00 00 00 00 00
0040 00 00 04 ea b5 3f 46 80 fc dd 82 1e a6 53 b0 49
0050 09 4a 1a d0 b5 c5 aa 3d dc eb 93 38 02 30 16 72
0060 47 4a 7d 6b 15 6f 8a 75 cb 45 08 7c 86 5e 1b 34
0070 60 2b 82 09 89 01 44 85 5f 79 d0 de c3 51 59 99
0080 cc c7 6a 8a a9 e2 02 9e 01 3e 4e c8 f9 5e 55 0a
0090 1e 22 c2 1b d8 9c 89 d4 b1 ab d2 ea 4d 37 a2 9a
00a0 89 50 06 13 b7 ec 6a 2f 5b 16 65 d6 a5 2f 43 be
00b0 c5 e1 60 5f 01 e5 c0 64 3e 67 42 4e 44 1f 96 dc
00c0 03 23 ee 5e fa 04 63 7a 0b 2e bf 1d 4c ce cd 0b
Frame (545 bytes)
Decrypted TLS (462 bytes)
Remote Desktop Protocol: Protocol
Packets: 2225 · Displayed: 1427 (64.1%)
Profile: Default
4:30 PM
5/31/2025


guided-rdp.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
rdp
No. Time Source Destination Protocol Length Info
31 8.858325 10.129.43.27 10.129.43.29 RDP 545 ClientData
32 8.859099 10.129.43.29 10.129.43.27 RDP 209 ServerData Encryption: None (None)
51 8.865852 10.129.43.27 10.129.43.29 RDP 698 ClientInfo
52 8.869677 10.129.43.29 10.129.43.27 RDP 117 Error Alert
53 8.870064 10.129.43.29 10.129.43.27 RDP 125 MultiTransportRequest
55 8.888062 10.129.43.27 10.129.43.29 RDP 109 MultiTransport response
56 8.927121 10.129.43.29 10.129.43.27 RDP 555 Demand Active PDU
57 8.931705 10.129.43.27 10.129.43.29 RDP 762 Confirm Active PDU
58 8.931849 10.129.43.27 10.129.43.29 RDP 119 RDP PDU Type: Synchronize
60 8.931918 10.129.43.29 10.129.43.27 RDP 119 RDP PDU Type: Synchronize
61 8.931925 10.129.43.27 10.129.43.29 RDP 123 RDP PDU Type: Control, Action: Cooperate
62 8.931975 10.129.43.29 10.129.43.27 RDP 123 RDP PDU Type: Control, Action: Cooperate
63 8.931997 10.129.43.27 10.129.43.29 RDP 123 RDP PDU Type: Control, Action: Request control
65 8.932186 10.129.43.29 10.129.43.27 RDP 123 RDP PDU Type: Control, Action: Granted control
66 8.932849 10.129.43.27 10.129.43.29 RDP 118 Fast-Path PDU,QoE timestamp,Scancode,Sync,Scancode,Scancode,Scancode,Scancode,Scancode,Scancode,Scancode,Scancode,Scancode,Mou
67 8.932959 10.129.43.29 10.129.43.27 RDP 139 RDP PDU Type: BitmapCache Persistent List
69 8.933005 10.129.43.27 10.129.43.29 RDP 123 RDP PDU Type: FontList
70 8.933101 10.129.43.29 10.129.43.27 RDP 123 RDP PDU Type: FontMap
ISO 8073/X.224 COTP Connection-Oriented Transport Protocol
MULTIPOINT-COMMUNICATION-SERVICE T.125
Remote Desktop Protocol
SendData
clientInfoPDU
flags: 0x0040
flagsHi: 0x1acc
codePage: 67699721
optionFlags: 0x000347bb
cbDomain: 0
cbUserName: 10
cbPassword: 16
cbAlternateShell: 0
cbWorkingDir: 0
domain:
userName: bucky
password: Welcome1
alternateShell:
workingDir:
clientAddressFamily: 0x0002
cbClientAddress: 26
0000 03 00 02 67 02 f0 80 64 00 08 03 eb 70 82 58 40
0010 00 cc 1a 09 04 09 04 bb 47 03 00 00 00 0a 00 10
0020 00 00 00 00 00 00 00 62 00 75 00 63 00 6b 00 79
0030 00 00 00 57 00 65 00 6c 00 63 00 6f 00 6d 00 65
0040 00 31 00 00 00 00 00 00 00 02 00 1a 00 31 00 30
0050 00 2e 00 31 00 32 00 39 00 2e 00 34 00 33 00 2e
0060 00 32 00 37 00 00 00 40 00 43 00 3a 00 5c 00 57
0070 00 69 00 6e 00 64 00 6f 00 77 00 73 00 5c 00 73
0080 00 79 00 73 00 74 00 65 00 6d 00 33 00 32 00 5c
0090 00 6d 00 73 00 74 00 73 00 63 00 61 00 78 00 2e
00a0 00 64 00 6c 00 6c 00 00 00 e0 01 00 00 50 00 61
00b0 00 63 00 69 00 66 00 69 00 63 00 20 00 53 00 74
00c0 00 61 00 6e 00 64 00 61 00 72 00 64 00 20 00 54
00d0 00 69 00 6d 00 65 00 00 00 00 00 00 00 00 00 00
00e0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0b
00f0 00 00 00 01 00 02 00 00 00 00 00 00 00 00 00 00
0100 00 50 00 61 00 63 00 69 00 66 00 69 00 63 00 20
0110 00 44 00 61 00 79 00 6c 00 69 00 67 00 68 00 74
0120 00 20 00 54 00 69 00 6d 00 65 00 00 00 00 00 00
0130 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0140 00 00 00 03 00 00 00 02 00 02 00 00 00 00 00 00
Frame (698 bytes)
Decrypted TLS (615 bytes)
Bitstring tvb (1 byte)
clientInfoPDU (rdp.clientInfoPDU), 600 bytes
Packets: 2225 · Displayed: 1427 (64.1%)
Profile: Default
4:31 PM
5/31/2025



Summary

This lab was to serve as an example of what Wireshark can do with captured data and its plugins. Wireshark's capability to ingest information and illuminate the obscure is robust. Having the ability to decrypt data after ingestion is a powerful capability. This concept could be applied to any protocol that utilizes encryption as long as we have the key that will be utilized to establish the connections.