HTB Academy: Introduction to Network Traffic Analysis

Introduction

Understanding how data travels across a network is fundamental to both offensive and defensive security. This assignment focuses on developing practical skills in analyzing network traffic using industry-standard tools such as tcpdump and Wireshark. The modules I covered throughout this exercise: from the OSI model to advanced packet dissection provided the technical foundation necessary to interpret, filter, and analyze network communications effectively.

Through hands-on labs and guided exercises, I explored key concepts such as network layers, traffic capture, and protocol analysis, and gained exposure to techniques like capture and display filtering, session reassembly, and even decrypting RDP traffic. Each section not only helped reinforce theoretical knowledge but also simulated real-world investigative scenarios, where identifying suspicious or malicious traffic is critical.

Objectives

Understand Network Analysis both theory and practice

Gain Hands on skills with tools for Network analysis such as TCPdump and Wireshark

Sections and Questions

Network Traffic Analysis

This part covered the introduction to network analysis, including the skills needed for the module and extra resources that might be needed.

It also listed some tools used for Network analysis. These include:

Tcpdump, Tshark, Wireshark, Ngrep, Tcpick, Network Taps, Networking Span Ports, Elastic Stack and SIEMS

Many of the tools mentioned above have their syntax and commands to utilize, but one that is shared among them is Berkeley Packet Filter (BPF) syntax, and this was the syntax used through out the module.

The section also covered Workflow for performing Network traffic analysis:

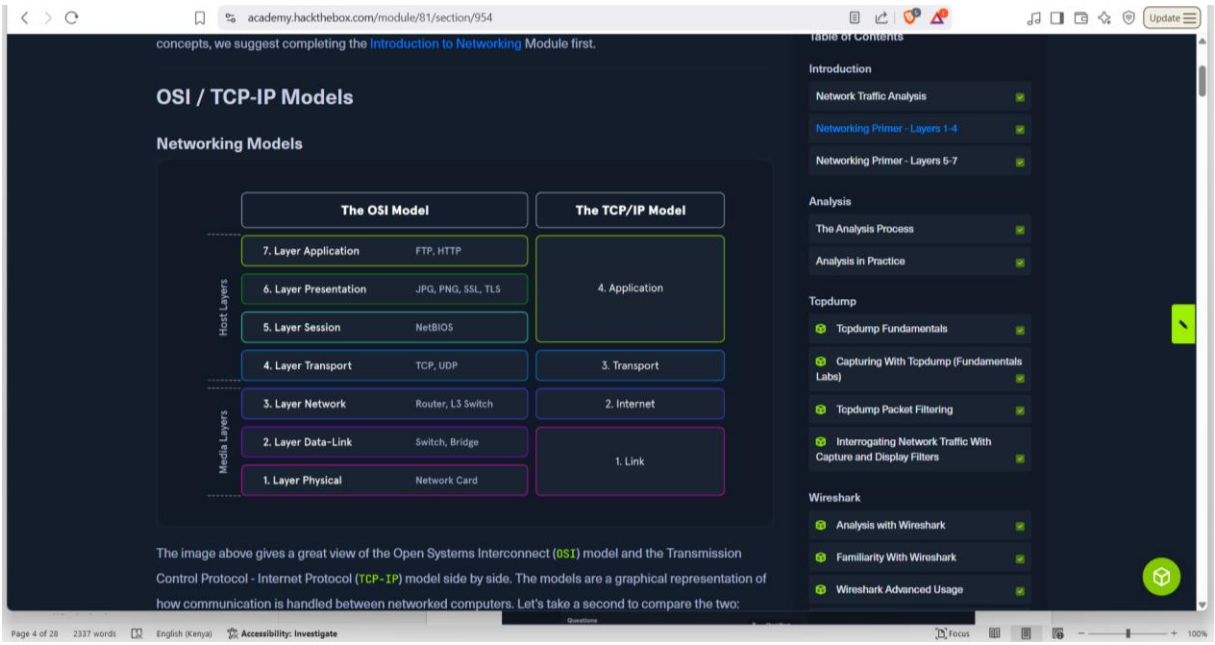
1. Ingest traffic
2. Reduce Noise
3. Analyse and Explore
4. Detect and Alert
5. Fix and Monitor

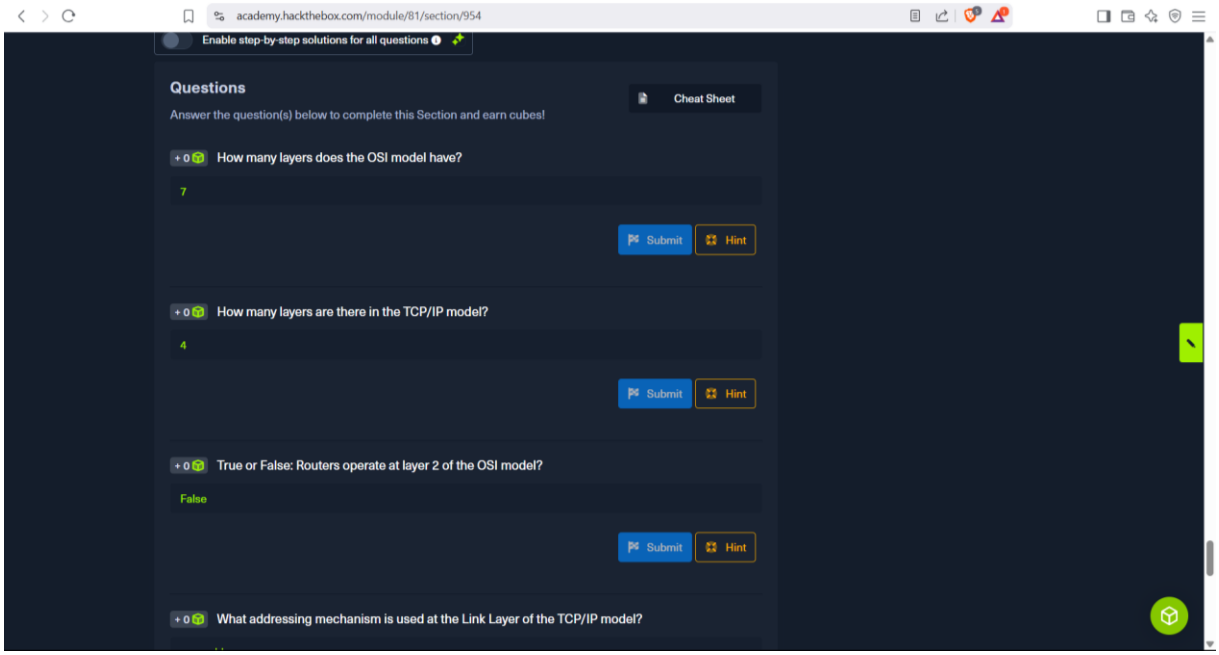
Networking Primer Layers 1-4

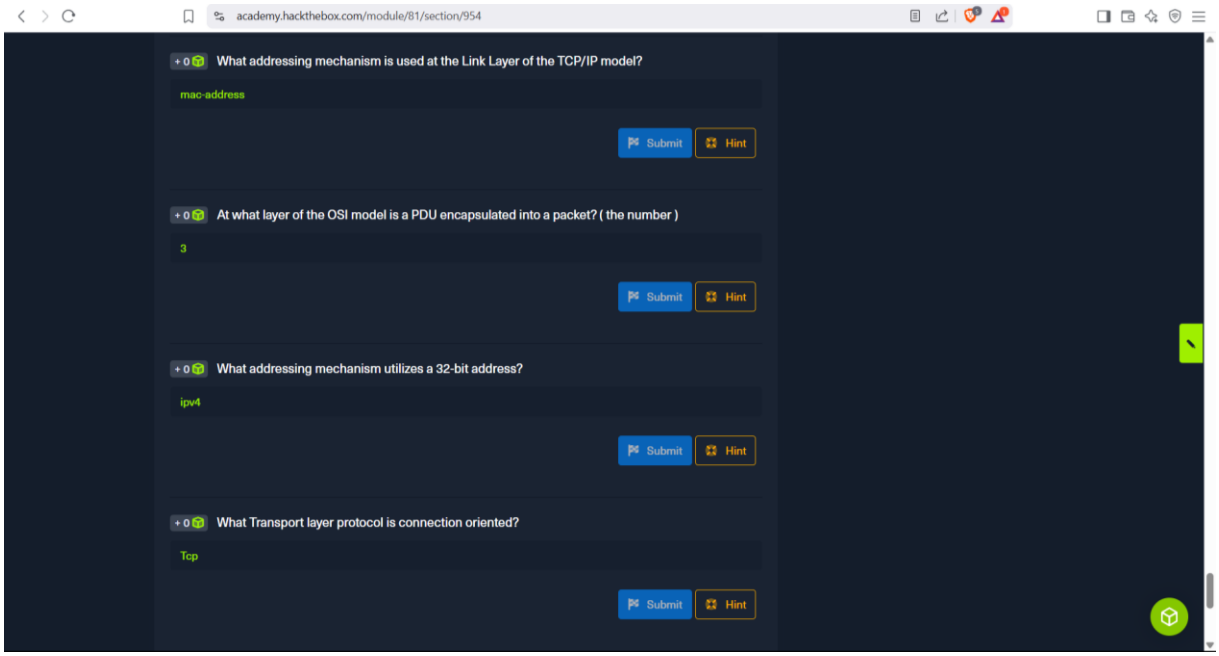
This section covered the relationship between OSI and TCP-IP Model, listing them down and explaining how they are all related.
It also explained the different forms of PDU at different layers of both OSI and TCP-IP.
It also covered a brief on mac addressing which is a 48 bit six octet address represented in hexadecimal format in the link layer of the TCP-IP model and Data link layer of the OSI.
The section also covered Ip addressing explaining both IPv4 and IPV6 briefly.
It finalized with TCP and udp explaining its difference and use cases.

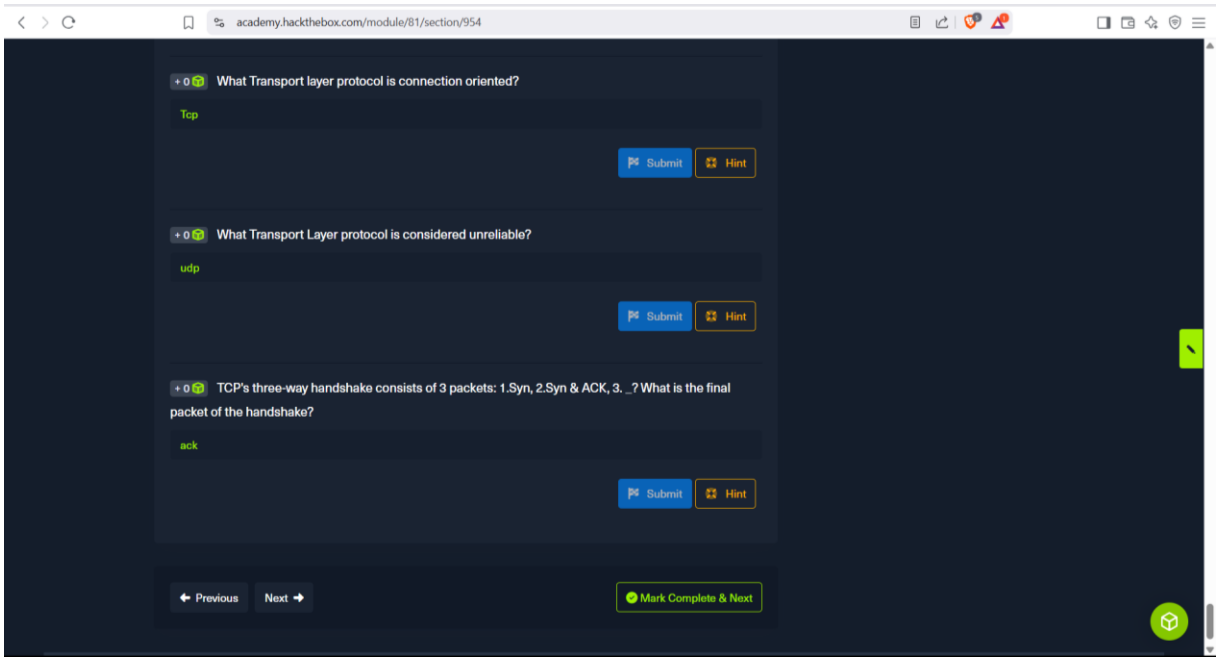
Questions:

How many layers does the OSI model have? 7
How many layers are there in the TCP/IP model? 4
True or False: Routers operate at layer 2 of the OSI model? False
What addressing mechanism is used at the Link Layer of the TCP/IP model? Mac-address
At what layer of the OSI model is a PDU encapsulated into a packet? 3
What addressing mechanism utilizes a 32-bit address? Ipv4
What Transport layer protocol is connection oriented? Tcp
What Transport Layer protocol is considered unreliable? Udp
TCP's three-way handshake consists of 3 packets: 1.Syn, 2.Syn & ACK, 3. _? What is the final packet of the handshake? Ack




academy.hackthebox.com/module/81/section/954
Enable step-by-step solutions for all questions
Questions
Answer the question(s) below to complete this Section and earn cubes!
Cheat Sheet
+0 How many layers does the OSI model have?
7
Submit
Hint
+0 How many layers are there in the TCP/IP model?
4
Submit
Hint
+0 True or False: Routers operate at layer 2 of the OSI model?
False
Submit
Hint
+0 What addressing mechanism is used at the Link Layer of the TCP/IP model?


academy.hackthebox.com/module/81/section/954
+0 What addressing mechanism is used at the Link Layer of the TCP/IP model?
mac-address
Submit
Hint
+0 At what layer of the OSI model is a PDU encapsulated into a packet? (the number)
3
Submit
Hint
+0 What addressing mechanism utilizes a 32-bit address?
ipv4
Submit
Hint
+0 What Transport layer protocol is connection oriented?
Tcp
Submit
Hint

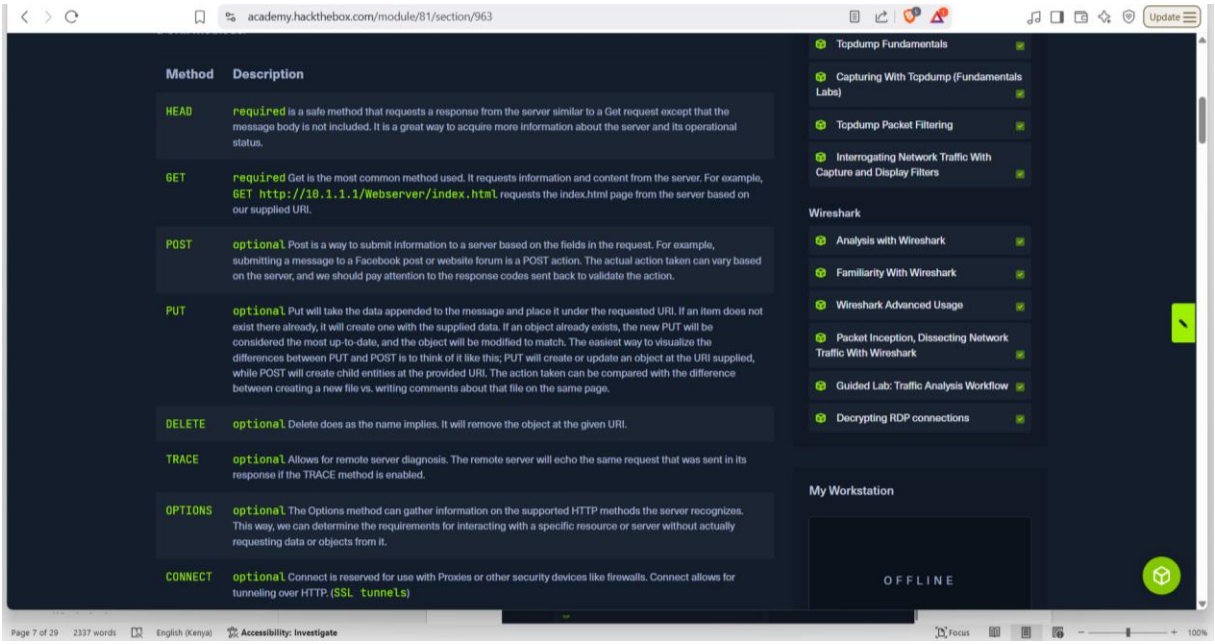


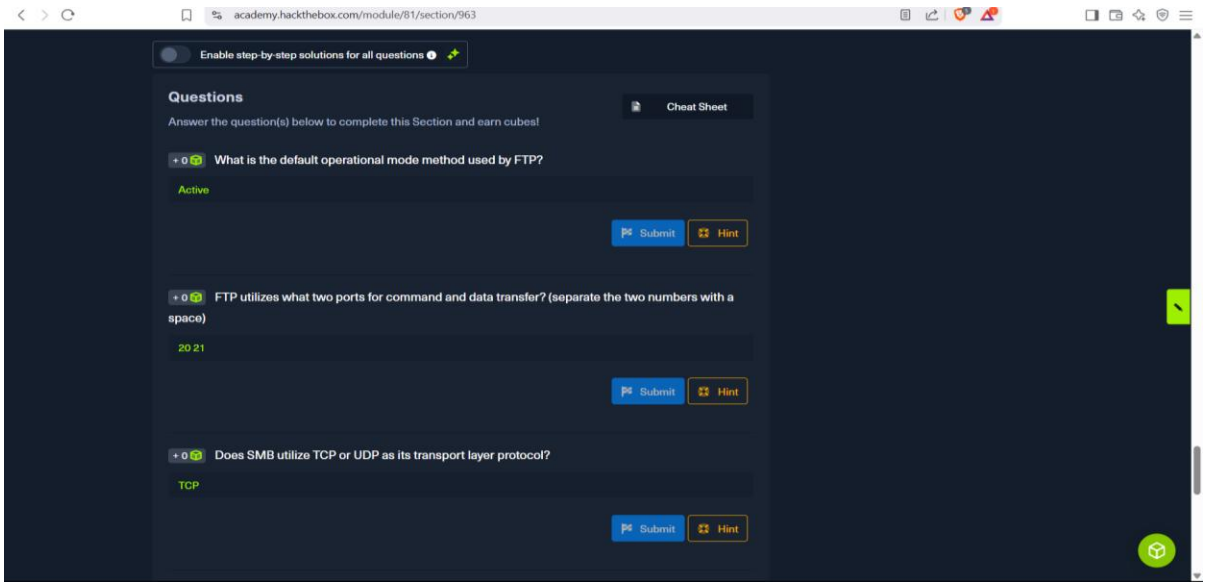
Networking Primer – Layer 5-7

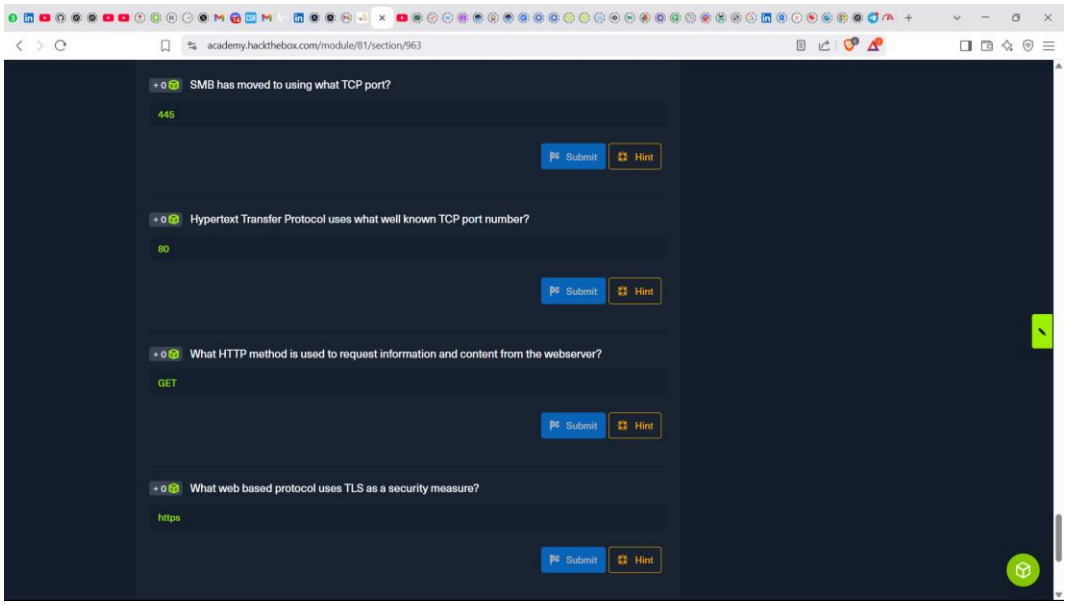
This section focused on upper layer protocols that handle applications. It started by defining Http, http methods, and briefly explaining the different http methods such as GET, POST HEAD, PUT, DELETE, TRACE, OPTIONS and CONNECT.
I also gained understanding on HTTPS a secure version of HTTP which is utilized to use TLS or SSL with older applications for data security.
FTP and some basic commands such as USER, PASS, PORT, PASV, LIST, CWD, PWD, SIZE RETR and QUIT were also outlined.
SMB a protocol used in windows enterprise for sharing resources was also covered.

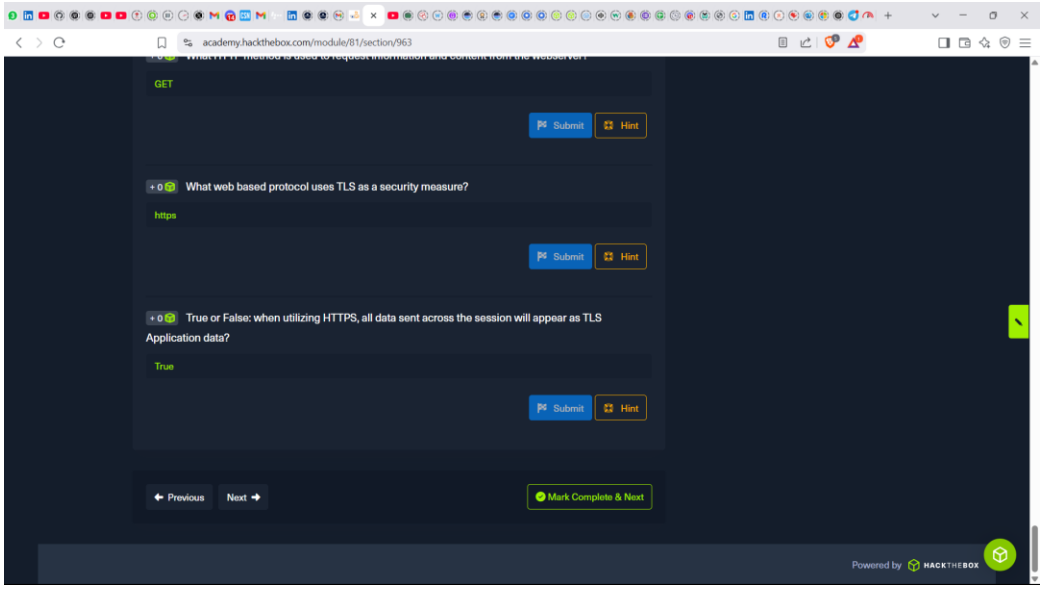
Questions:

What is the default operational mode method used by FTP? Active
FTP utilizes what two ports for command and data transfer? (separate the two numbers with a space) 20 21
Does SMB utilize TCP or UDP as its transport layer protocol? TCP
SMB has moved to using what TCP port? 445
Hypertext Transfer Protocol uses what well known TCP port number? 80
What HTTP method is used to request information and content from the webserver? GET
 What web based protocol uses TLS as a security measure? https
True or False: when utilizing HTTPS, all data sent across the session will appear as TLS Application data? True


academy.hackthebox.com/module/81/section/963
Update
Method
Description
HEAD
required is a safe method that requests a response from the server similar to a Get request except that the message body is not included. It is a great way to acquire more information about the server and its operational status.
GET
required Get is the most common method used. It requests information and content from the server. For example, GET http://10.1.1.1/Webserver/index.html requests the index.html page from the server based on our supplied URI.
POST
optional Post is a way to submit information to a server based on the fields in the request. For example, submitting a message to a Facebook post or website forum is a POST action. The actual action taken can vary based on the server, and we should pay attention to the response codes sent back to validate the action.
PUT
optional Put will take the data appended to the message and place it under the requested URI. If an item does not exist there already, it will create one with the supplied data. If an object already exists, the new PUT will be considered the most up-to-date, and the object will be modified to match. The easiest way to visualize the differences between PUT and POST is to think of it like this; PUT will create or update an object at the URI supplied, while POST will create child entities at the provided URI. The action taken can be compared with the difference between creating a new file vs. writing comments about that file on the same page.
DELETE
optional Delete does as the name implies. It will remove the object at the given URI.
TRACE
optional Allows for remote server diagnosis. The remote server will echo the same request that was sent in its response if the TRACE method is enabled.
OPTIONS
optional The Options method can gather information on the supported HTTP methods the server recognizes. This way, we can determine the requirements for interacting with a specific resource or server without actually requesting data or objects from it.
CONNECT
optional Connect is reserved for use with Proxies or other security devices like firewalls. Connect allows for tunneling over HTTP. (SSL tunnels)
Tcpdump Fundamentals
Capturing With Tcpdump (Fundamentals Labs)
Tcpdump Packet Filtering
Interrogating Network Traffic With Capture and Display Filters
Wireshark
Analysis with Wireshark
Familiarity With Wireshark
Wireshark Advanced Usage
Packet Inception, Dissecting Network Traffic With Wireshark
Guided Lab: Traffic Analysis Workflow
Decrypting RDP connections
My Workstation
OFFLINE
Page 7 of 29
2337 words
English (Kenya)
Accessibility: Investigate
Focus
100%


academy.hackthebox.com/module/81/section/963
Enable step-by-step solutions for all questions
Questions
Answer the question(s) below to complete this Section and earn cubes!
Cheat Sheet
+0 What is the default operational mode method used by FTP?
Active
Submit
Hint
+0 FTP utilizes what two ports for command and data transfer? (separate the two numbers with a space)
20 21
Submit
Hint
+0 Does SMB utilize TCP or UDP as its transport layer protocol?
TCP
Submit
Hint





The Analysis Process

This section dived into the details of Network analysis, explaining the need for having the skill to an organization, key being visibility.

Analysis Dependencies were discussed including Permission, Mirrored Port, Capture Tool, In-line Placement, Network Tap or Host with Multiple NIC's, Storage and Processing Power.

Analysis in Practice

This part focused on preparation for hands on practice where by it gave the process of the analysis starting with:

Descriptive analysis: which consists of the first 3 steps

Diagnostic analysis consisting of step 4-6

Predictive Analysis consisting of the last 2 steps 7 and 8

The comprehensive Prescriptive analysis (consisting of all the 8 steps) aims to narrow down what actions to take to eliminate or prevent a future problem or trigger a specific activity or process. The 8 steps in the process include

1. What is the Issue?(suspected breach? Networking Issue?)
2. Define our scope and the goal (what are we looking for? What is the time period?)
3. Define our target(s) (net/host(s) /protocol)
4. Capture Network Traffic
5. Identification of required network traffic components (filtering)
6. An understanding of captured network traffic
7. Note taking and Mind mapping of the found results
8. Summary of the analysis (what did we find?)

It also covered the 3 Key components of effective analysis:

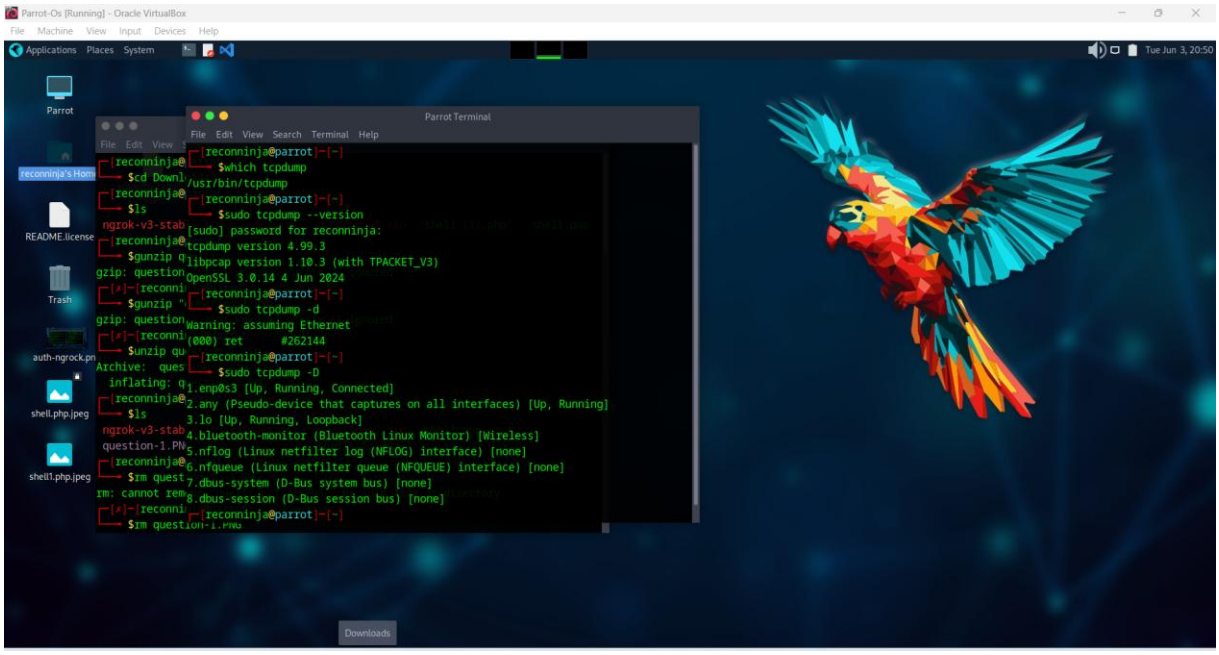
1. Know your environment
2. Placement is key
3. Persistence

It finalized by talking of analysis approach where its advisable to start with standard protocols first and note that our eyes are the best tools since everyday threat actors look for ways of bypassing present tools. Asking for help is also necessary as a second set of eyes can spot something overlooked by the first.

Tcpdump Fundamentals

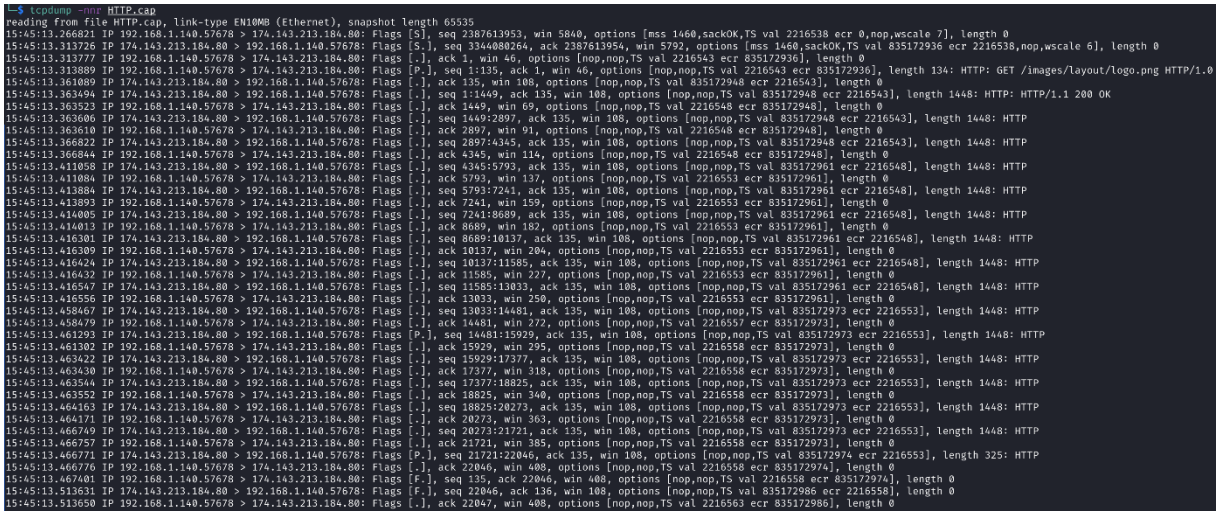
This part gave the fundamentals of tcpdump, what it is, how it is used and the operating system supporting it.

It also covered traffic captures with tcpdump, using man tcpdump , basic tcpdump commands and tcpdump switch combinations.



Questions:

The output below was to be used in answering the questions that will follow:



```
└─$ tcpdump -nnr HTTP.cap
reading from file HTTP.cap, link-type EN10MB (Ethernet), snapshot length 65535
15:45:13.266821 IP 192.168.1.140.57678 > 174.143.213.184.80: Flags [S], seq 2387613953, win 5840, options [mss 1460,sackOK,TS val 2216538 ecr 0,nop,wscale 7], length 0
15:45:13.313726 IP 174.143.213.184.80 > 192.168.1.140.57678: Flags [S.], seq 3344080264, ack 2387613954, win 5792, options [mss 1460,sackOK,TS val 835172936 ecr 2216538,nop,wscale 6], length 0
15:45:13.313777 IP 192.168.1.140.57678 > 174.143.213.184.80: Flags [.], ack 1, win 46, options [nop,nop,TS val 2216543 ecr 835172936], length 0
15:45:13.313889 IP 192.168.1.140.57678 > 174.143.213.184.80: Flags [P.], seq 1:135, ack 1, win 46, options [nop,nop,TS val 2216543 ecr 835172936], length 134: HTTP: GET /images/layout/logo.png HTTP/1.0
15:45:13.361089 IP 174.143.213.184.80 > 192.168.1.140.57678: Flags [.], ack 135, win 108, options [nop,nop,TS val 835172948 ecr 2216543], length 0
15:45:13.363494 IP 174.143.213.184.80 > 192.168.1.140.57678: Flags [.], seq 1:1449, ack 135, win 108, options [nop,nop,TS val 835172948 ecr 2216543], length 1448: HTTP: HTTP/1.1 200 OK
15:45:13.363523 IP 192.168.1.140.57678 > 174.143.213.184.80: Flags [.], ack 1449, win 69, options [nop,nop,TS val 2216548 ecr 835172948], length 0
15:45:13.363606 IP 174.143.213.184.80 > 192.168.1.140.57678: Flags [.], seq 1449:2897, ack 135, win 108, options [nop,nop,TS val 835172948 ecr 2216543], length 1448: HTTP
15:45:13.363610 IP 192.168.1.140.57678 > 174.143.213.184.80: Flags [.], ack 2897, win 91, options [nop,nop,TS val 2216548 ecr 835172948], length 0
15:45:13.366822 IP 174.143.213.184.80 > 192.168.1.140.57678: Flags [.], seq 2897:4345, ack 135, win 108, options [nop,nop,TS val 835172948 ecr 2216543], length 1448: HTTP
15:45:13.366844 IP 192.168.1.140.57678 > 174.143.213.184.80: Flags [.], ack 4345, win 114, options [nop,nop,TS val 2216548 ecr 835172948], length 0
15:45:13.411058 IP 174.143.213.184.80 > 192.168.1.140.57678: Flags [.], seq 4345:5793, ack 135, win 108, options [nop,nop,TS val 835172961 ecr 2216548], length 1448: HTTP
15:45:13.411084 IP 192.168.1.140.57678 > 174.143.213.184.80: Flags [.], ack 5793, win 137, options [nop,nop,TS val 2216553 ecr 835172961], length 0
15:45:13.413884 IP 174.143.213.184.80 > 192.168.1.140.57678: Flags [.], seq 5793:7241, ack 135, win 108, options [nop,nop,TS val 835172961 ecr 2216548], length 1448: HTTP
15:45:13.413893 IP 192.168.1.140.57678 > 174.143.213.184.80: Flags [.], ack 7241, win 159, options [nop,nop,TS val 2216553 ecr 835172961], length 0
15:45:13.414005 IP 174.143.213.184.80 > 192.168.1.140.57678: Flags [.], seq 7241:8689, ack 135, win 108, options [nop,nop,TS val 835172961 ecr 2216548], length 1448: HTTP
15:45:13.414013 IP 192.168.1.140.57678 > 174.143.213.184.80: Flags [.], ack 8689, win 182, options [nop,nop,TS val 2216553 ecr 835172961], length 0
15:45:13.416301 IP 174.143.213.184.80 > 192.168.1.140.57678: Flags [.], seq 8689:10137, ack 135, win 108, options [nop,nop,TS val 835172961 ecr 2216548], length 1448: HTTP
15:45:13.416309 IP 192.168.1.140.57678 > 174.143.213.184.80: Flags [.], ack 10137, win 204, options [nop,nop,TS val 2216553 ecr 835172961], length 0
15:45:13.416424 IP 174.143.213.184.80 > 192.168.1.140.57678: Flags [.], seq 10137:11585, ack 135, win 108, options [nop,nop,TS val 835172961 ecr 2216548], length 1448: HTTP
15:45:13.416432 IP 192.168.1.140.57678 > 174.143.213.184.80: Flags [.], ack 11585, win 227, options [nop,nop,TS val 2216553 ecr 835172961], length 0
15:45:13.416547 IP 174.143.213.184.80 > 192.168.1.140.57678: Flags [.], seq 11585:13033, ack 135, win 108, options [nop,nop,TS val 835172961 ecr 2216548], length 1448: HTTP
15:45:13.416556 IP 192.168.1.140.57678 > 174.143.213.184.80: Flags [.], ack 13033, win 250, options [nop,nop,TS val 2216553 ecr 835172961], length 0
15:45:13.458467 IP 174.143.213.184.80 > 192.168.1.140.57678: Flags [.], seq 13033:14481, ack 135, win 108, options [nop,nop,TS val 835172973 ecr 2216553], length 1448: HTTP
15:45:13.458479 IP 192.168.1.140.57678 > 174.143.213.184.80: Flags [.], ack 14481, win 272, options [nop,nop,TS val 2216557 ecr 835172973], length 0
15:45:13.461293 IP 174.143.213.184.80 > 192.168.1.140.57678: Flags [P.], seq 14481:15929, ack 135, win 108, options [nop,nop,TS val 835172973 ecr 2216553], length 1448: HTTP
15:45:13.461302 IP 192.168.1.140.57678 > 174.143.213.184.80: Flags [.], ack 15929, win 295, options [nop,nop,TS val 2216558 ecr 835172973], length 0
15:45:13.463422 IP 174.143.213.184.80 > 192.168.1.140.57678: Flags [.], seq 15929:17377, ack 135, win 108, options [nop,nop,TS val 835172973 ecr 2216553], length 1448: HTTP
15:45:13.463430 IP 192.168.1.140.57678 > 174.143.213.184.80: Flags [.], ack 17377, win 318, options [nop,nop,TS val 2216558 ecr 835172973], length 0
15:45:13.463544 IP 174.143.213.184.80 > 192.168.1.140.57678: Flags [.], seq 17377:18825, ack 135, win 108, options [nop,nop,TS val 835172973 ecr 2216553], length 1448: HTTP
15:45:13.463552 IP 192.168.1.140.57678 > 174.143.213.184.80: Flags [.], ack 18825, win 340, options [nop,nop,TS val 2216558 ecr 835172973], length 0
15:45:13.464163 IP 174.143.213.184.80 > 192.168.1.140.57678: Flags [.], seq 18825:20273, ack 135, win 108, options [nop,nop,TS val 835172973 ecr 2216553], length 1448: HTTP
15:45:13.464171 IP 192.168.1.140.57678 > 174.143.213.184.80: Flags [.], ack 20273, win 363, options [nop,nop,TS val 2216558 ecr 835172973], length 0
15:45:13.466749 IP 174.143.213.184.80 > 192.168.1.140.57678: Flags [.], seq 20273:21721, ack 135, win 108, options [nop,nop,TS val 835172973 ecr 2216553], length 1448: HTTP
15:45:13.466757 IP 192.168.1.140.57678 > 174.143.213.184.80: Flags [.], ack 21721, win 385, options [nop,nop,TS val 2216558 ecr 835172973], length 0
15:45:13.466771 IP 174.143.213.184.80 > 192.168.1.140.57678: Flags [P.], seq 21721:22046, ack 135, win 108, options [nop,nop,TS val 835172974 ecr 2216553], length 325: HTTP
15:45:13.466776 IP 192.168.1.140.57678 > 174.143.213.184.80: Flags [.], ack 22046, win 408, options [nop,nop,TS val 2216558 ecr 835172974], length 0
15:45:13.467401 IP 192.168.1.140.57678 > 174.143.213.184.80: Flags [F.], seq 135, ack 22046, win 408, options [nop,nop,TS val 2216558 ecr 835172974], length 0
15:45:13.513631 IP 174.143.213.184.80 > 192.168.1.140.57678: Flags [F.], seq 22046, ack 136, win 108, options [nop,nop,TS val 835172986 ecr 2216558], length 0
15:45:13.513650 IP 192.168.1.140.57678 > 174.143.213.184.80: Flags [.], ack 22047, win 408, options [nop,nop,TS val 2216563 ecr 835172986], length 0
```

Utilizing the output shown in question-1.png, who is the server in this communication? (IP Address) 174.143.213.184

Were absolute or relative sequence numbers used during the capture? (see question-1.zip to answer) relative

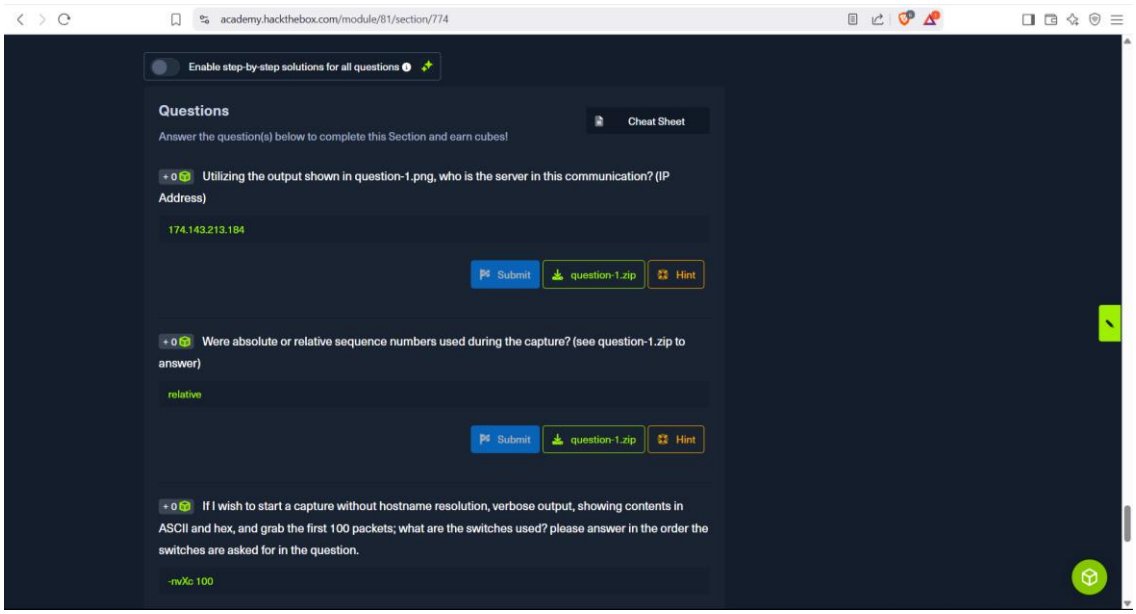
If I wish to start a capture without hostname resolution, verbose output, showing contents in ASCII and hex, and grab the first 100 packets; what are the switches used? please answer in the order the switches are asked for in the question Ans: -nvXc 100

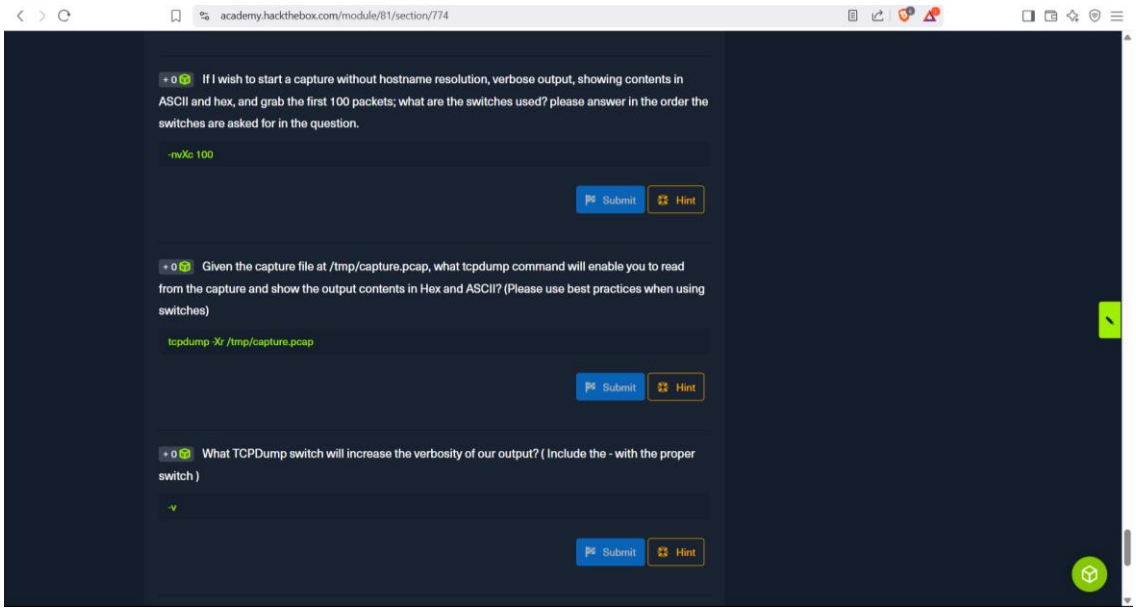
Given the capture file at /tmp/capture.pcap, what tcpdump command will enable you to read from the capture and show the output contents in Hex and ASCII? (Please use best practices when using switches) Ans : tcpdump -Xr /tmp/capture.pcap

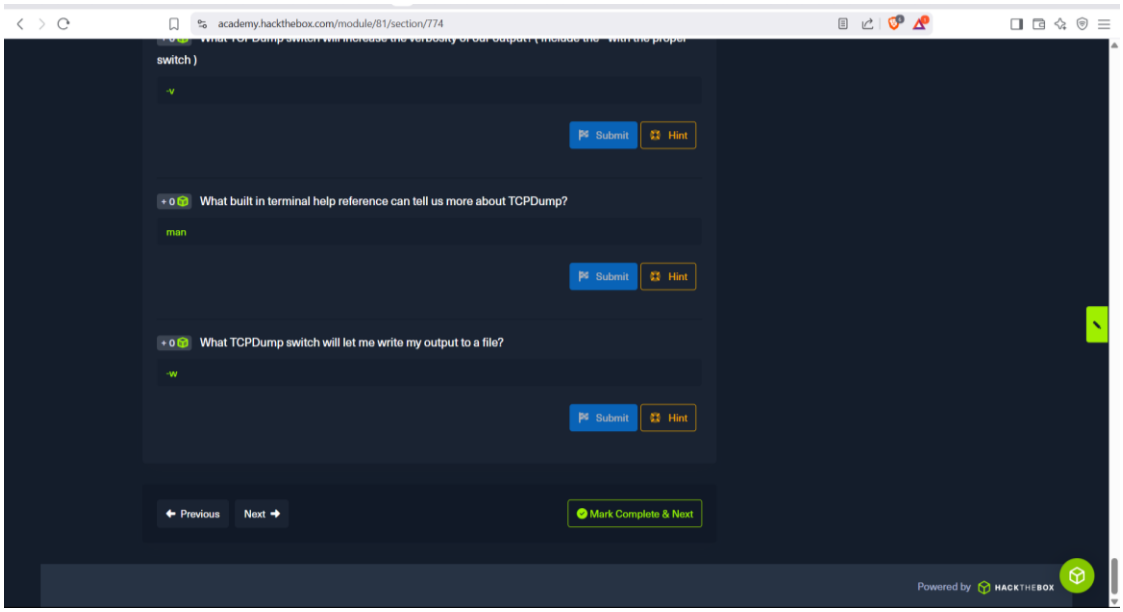
What TCPDump switch will increase the verbosity of our output? (Include the - with the proper switch) Ans: -v

What built in terminal help reference can tell us more about TCPDump? Man

What TCPDump switch will let me write my output to a file? -w

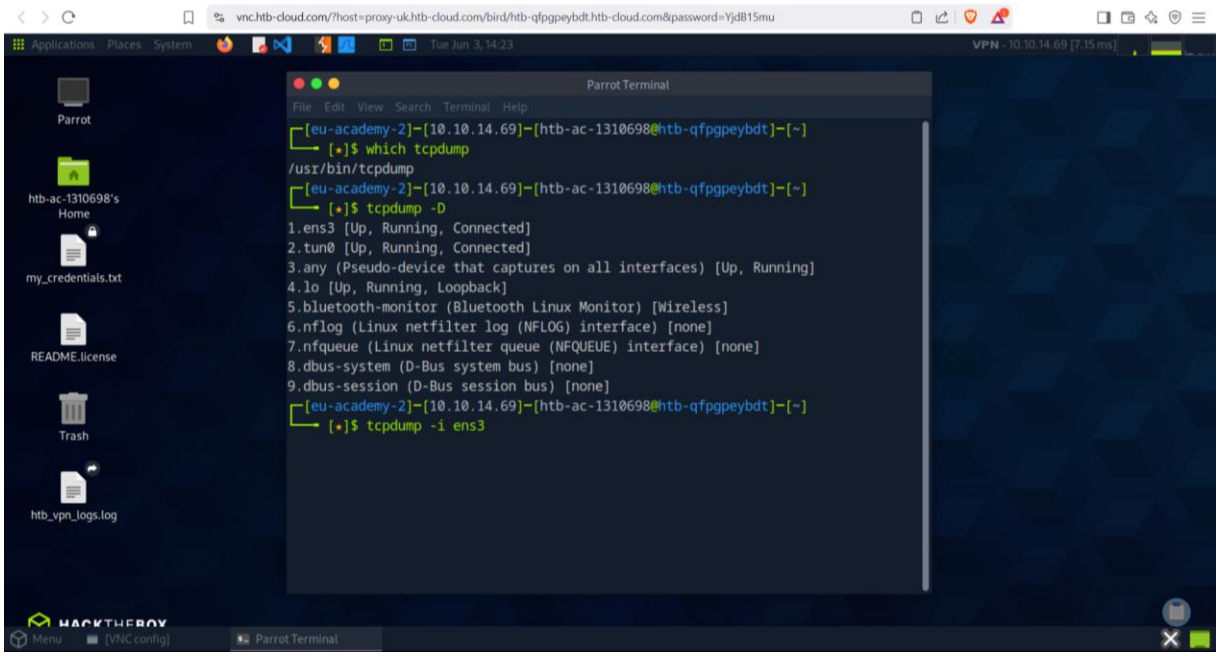

academy.hackthebox.com/module/81/section/774
Enable step-by-step solutions for all questions
Questions
Cheat Sheet
Answer the question(s) below to complete this Section and earn cubes!
+0 Utilizing the output shown in question-1.png, who is the server in this communication? (IP Address)
174.143.213.184
Submit
question-1.zip
Hint
+0 Were absolute or relative sequence numbers used during the capture? (see question-1.zip to answer)
relative
Submit
question-1.zip
Hint
+0 If I wish to start a capture without hostname resolution, verbose output, showing contents in ASCII and hex, and grab the first 100 packets; what are the switches used? please answer in the order the switches are asked for in the question.
-nvXc 100


academy.hackthebox.com/module/81/section/774
+0 If I wish to start a capture without hostname resolution, verbose output, showing contents in ASCII and hex, and grab the first 100 packets; what are the switches used? please answer in the order the switches are asked for in the question.
-nvXc 100
Submit
Hint
+0 Given the capture file at /tmp/capture.pcap, what tcpdump command will enable you to read from the capture and show the output contents in Hex and ASCII? (Please use best practices when using switches)
tcpdump -Xr /tmp/capture.pcap
Submit
Hint
+0 What TCPDump switch will increase the verbosity of our output? (Include the - with the proper switch)
-v
Submit
Hint

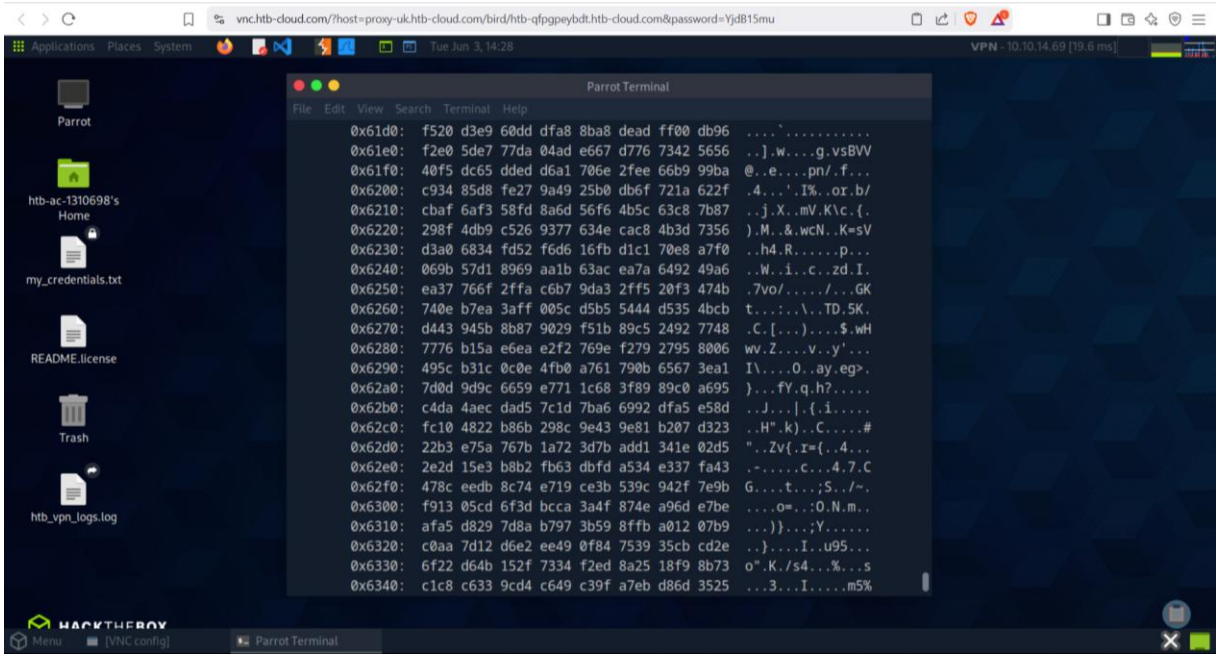


Capturing with tcpdump (Lab fundamentals)

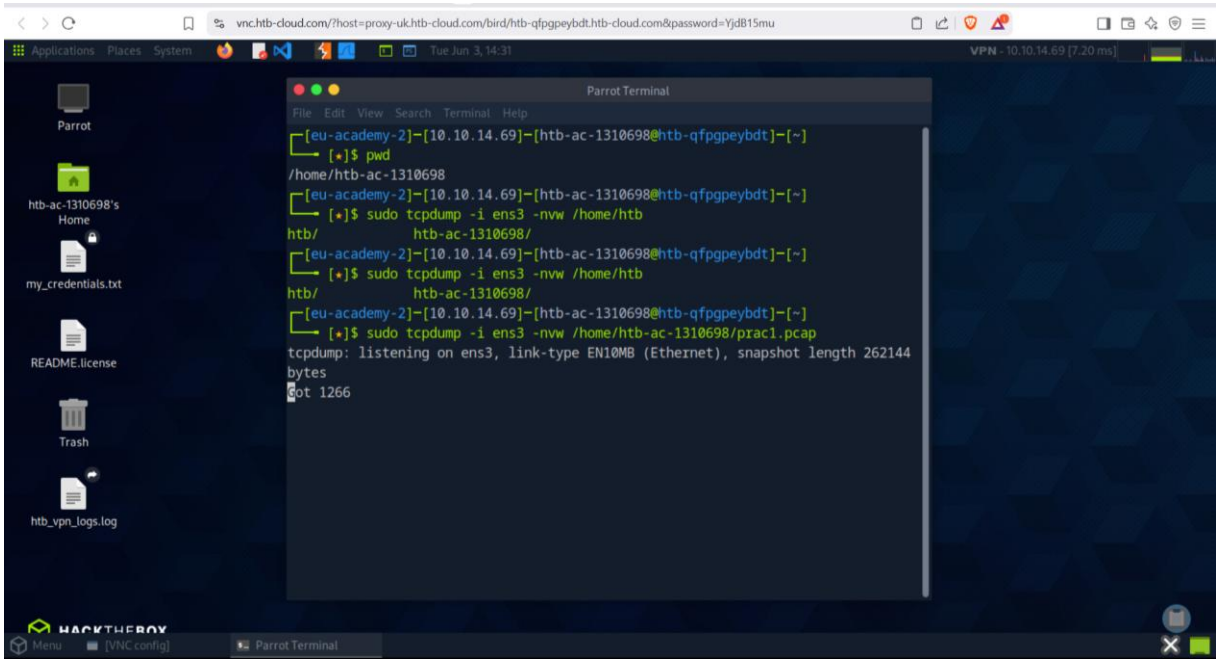
Here I learned how to put the commands from the previous session to practice:



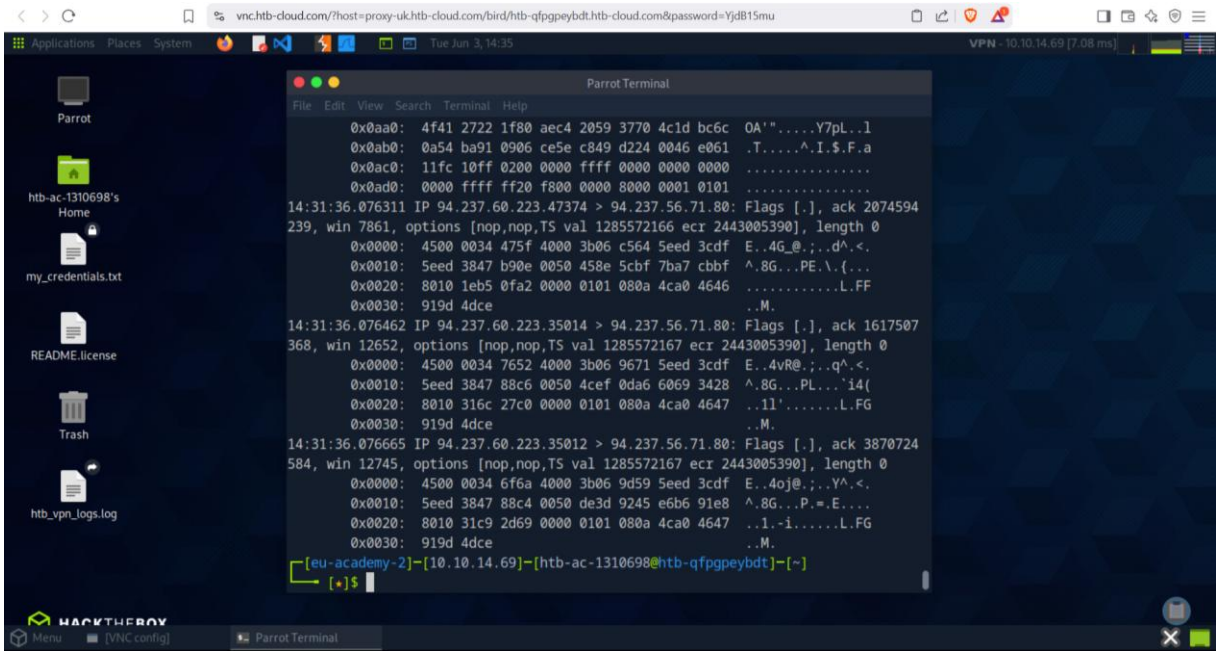
Basic command to capture results in a more visually appealing format: sudo tcpdump -i ens3 -vX



Saving the tcpdump output to a file:



Reading tcpdump output saved on a file using switches that will not resolve hostnames or port numbers: tcpdump -nnSXr prac1.pcap



Questions:

What TCPDump switch will allow us to pipe the contents of a pcap file out to another function such as 'grep'? -l

True or False: The filter "port" looks at source and destination traffic. Ans True

If we wished to filter out ICMP traffic from our capture, what filter could we use? (word only, not symbol please.) Ans : not icmp

What command will show you where / if TCPDump is installed? Which tcpdump

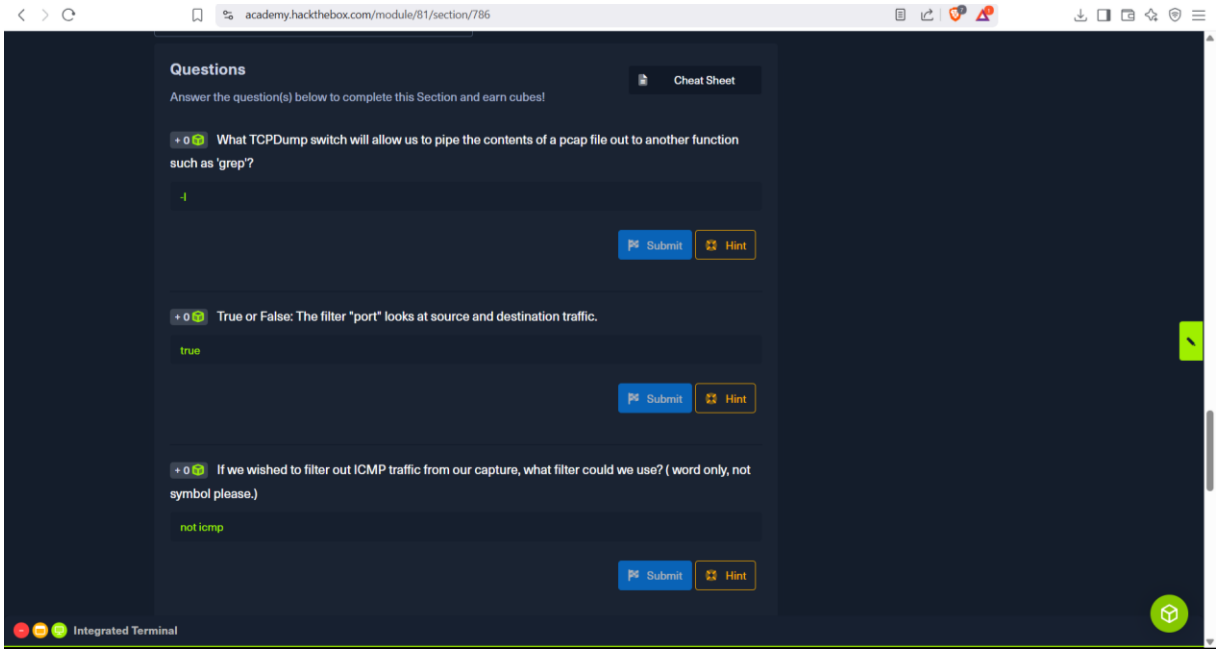
How do you start a capture with TCPDump to capture on eth0? tcpdump -i eth0

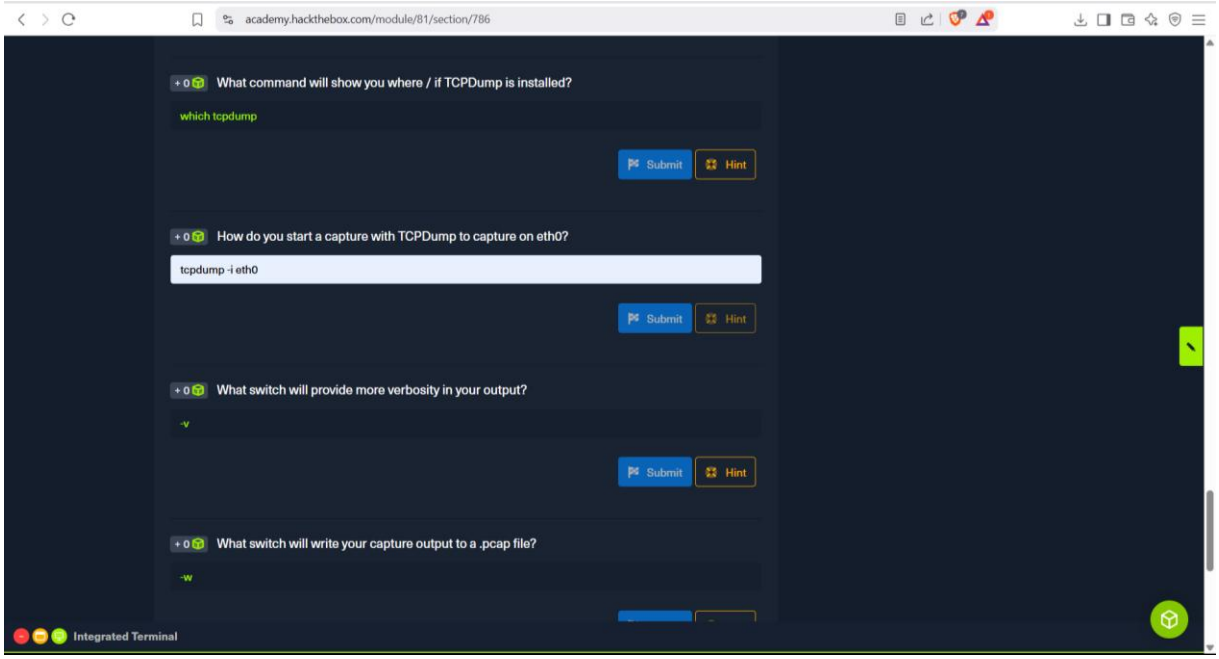
What switch will provide more verbosity in your output? -v

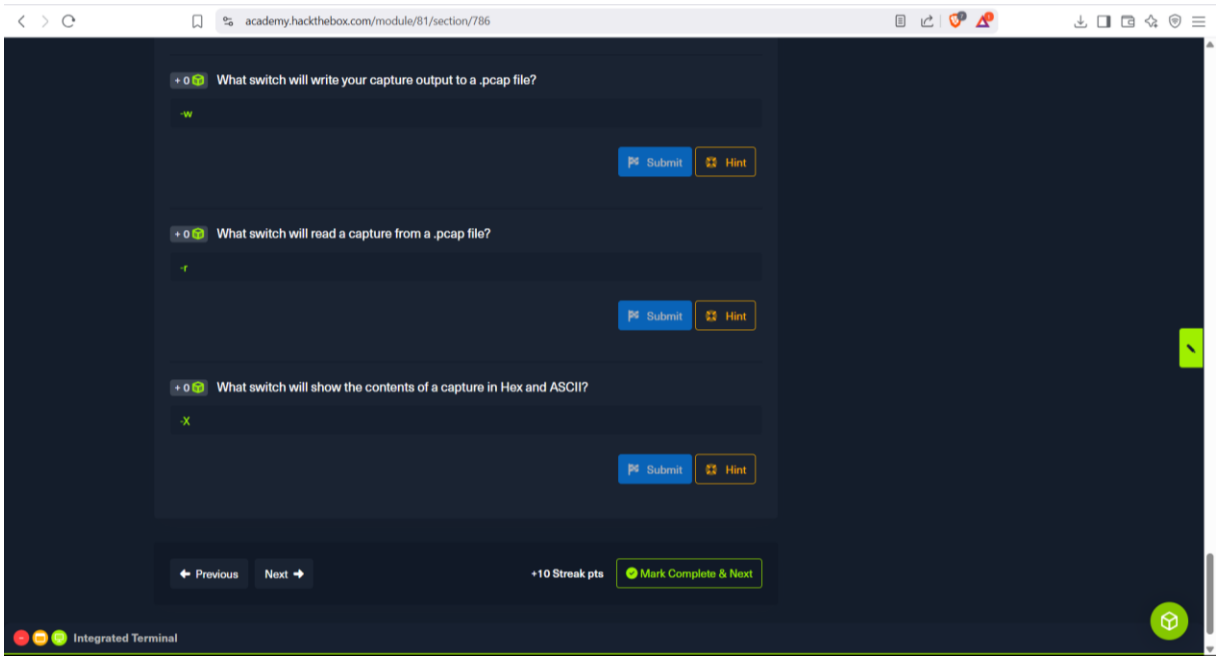
What switch will write your capture output to a .pcap file? -w

What switch will read a capture from a .pcap file? -r

What switch will show the contents of a capture in Hex and ASCII? -X


academy.hackthebox.com/module/81/section/786
Questions
Cheat Sheet
Answer the question(s) below to complete this Section and earn cubes!
+ 0 What TCPDump switch will allow us to pipe the contents of a pcap file out to another function such as 'grep'?
-l
Submit
Hint
+ 0 True or False: The filter "port" looks at source and destination traffic.
true
Submit
Hint
+ 0 If we wished to filter out ICMP traffic from our capture, what filter could we use? (word only, not symbol please.)
not icmp
Submit
Hint
Integrated Terminal


academy.hackthebox.com/module/81/section/786
+ 0 What command will show you where / if TCPDump is installed?
which tcpdump
Submit
Hint
+ 0 How do you start a capture with TCPDump to capture on eth0?
tcpdump -i eth0
Submit
Hint
+ 0 What switch will provide more verbosity in your output?
-v
Submit
Hint
+ 0 What switch will write your capture output to a .pcap file?
-w
Integrated Terminal



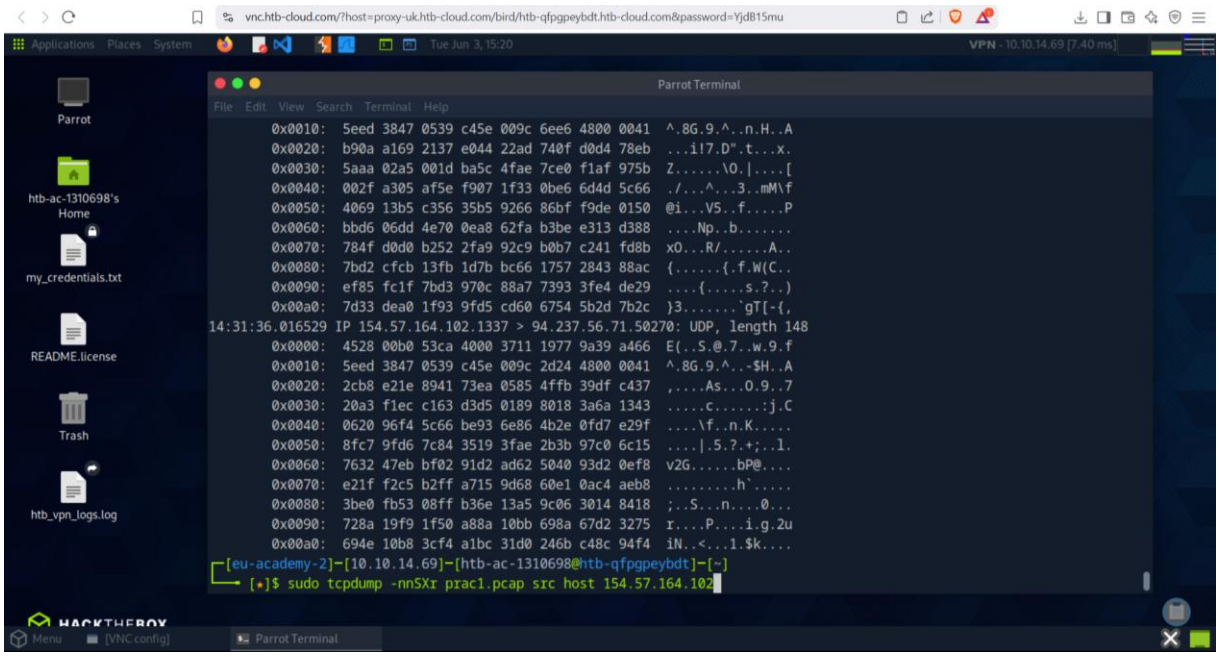
Tcpdump packet filtering

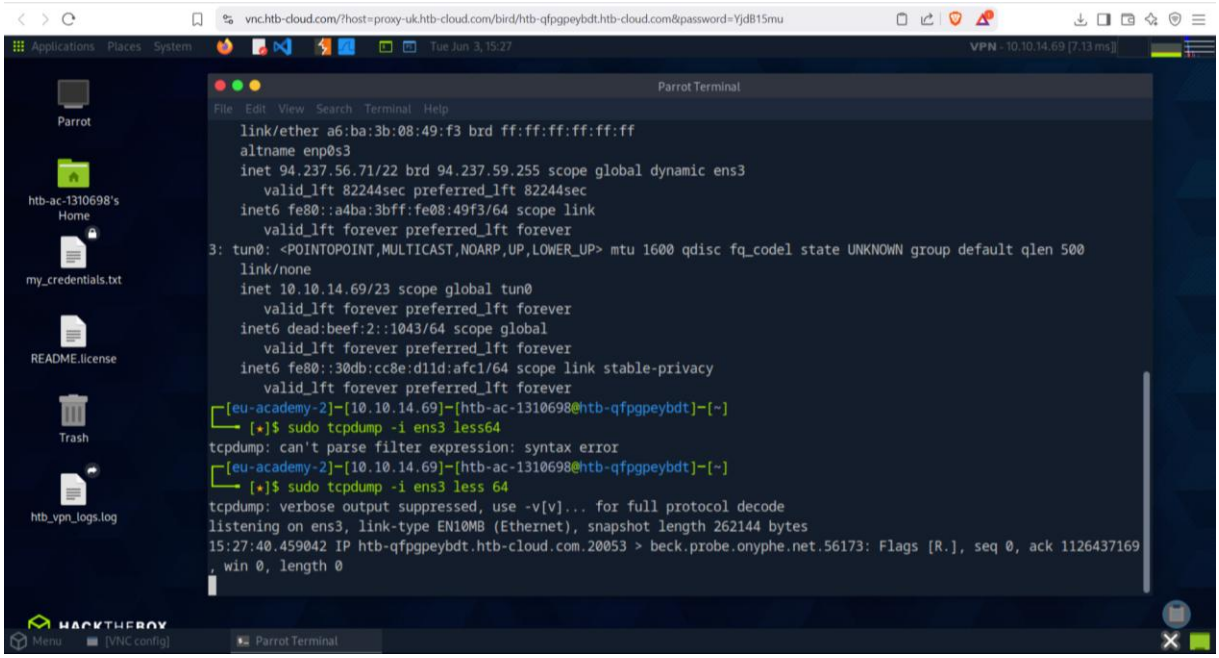
This section took me to advance syntax and filtering options on tcpdump. Some helpful filters like host net, proto, port and many more were now introduced in this section. Some are shown below:

Helpful TCPDump Filters

Filter	Result
host	`host` will filter visible traffic to show anything involving the designated host. Bi-directional
src / dest	`src` and `dest` are modifiers. We can use them to designate a source or destination host or port.
net	`net` will show us any traffic sourcing from or destined to the network designated. It uses / notation.
proto	will filter for a specific protocol type. (ether, TCP, UDP, and ICMP as examples)
port	`port` is bi-directional. It will show any traffic with the specified port as the source or destination.
portrange	`portrange` allows us to specify a range of ports. (0-1024)
less / greater "< >"	`less` and `greater` can be used to look for a packet or protocol option of a specific size.
and / &&	`and` `&&` can be used to concatenate two different filters together. for example, src host AND port.
or	`or` allows for a match on either of two conditions. It does not have to meet both. It can be tricky.
not	`not` is a modifier saying anything but x. For example, not UDP.

Practice on htb labs





After this i got some tips:
The -v, -X, and -e switches can help you increase the amount of data captured, while the -c, -n, -s, -S, and -q switches can help reduce and modify the amount of data written and seen.

Using the -S switch will display absolute sequence numbers, which can be extremely long. Typically, tcpdump displays relative sequence numbers, which are easier to track and read

Many handy options that can be used but are not always directly valuable for everyone are the -A and -l switches. A will show only the ASCII text after the packet line, instead of both ASCII and Hex. L will tell tcpdump to output packets in a different mode. L will line buffer instead of pooling and pushing in chunks. It allows us to send the output directly to another tool such as grep using a pipe |

Eg to view only email headers on a http.cap file: sudo tcpdump -Ar http.cap -l | grep 'mailto:*'

Looking for tcp protocol flags: tcpdump -i eth0 'tcp[13] &2 != 0'

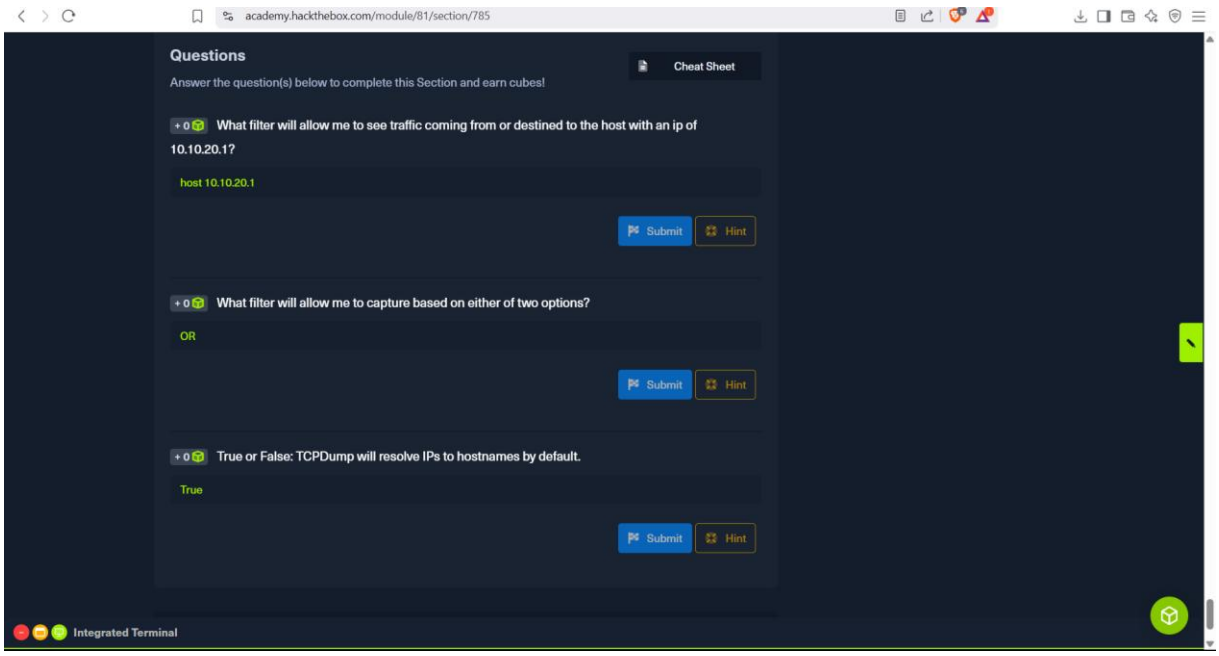
Hunting for SYN flags: sudo tcpdump -i eth0 'tcp[13] &2 != 0'

Questions:

What filter will allow me to see traffic coming from or destined to the host with an ip of 10.10.20.1? host 10.10.20.1

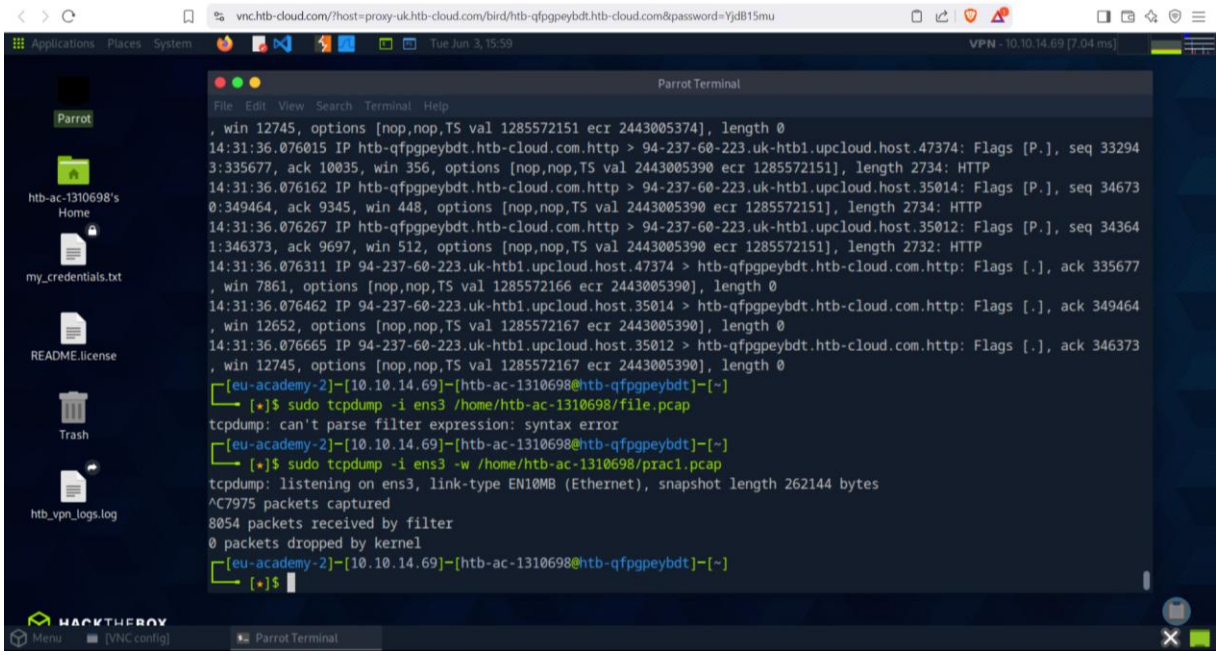
What filter will allow me to capture based on either of two options? OR

True or False: TCPDump will resolve IPs to hostnames by default. Ans: True



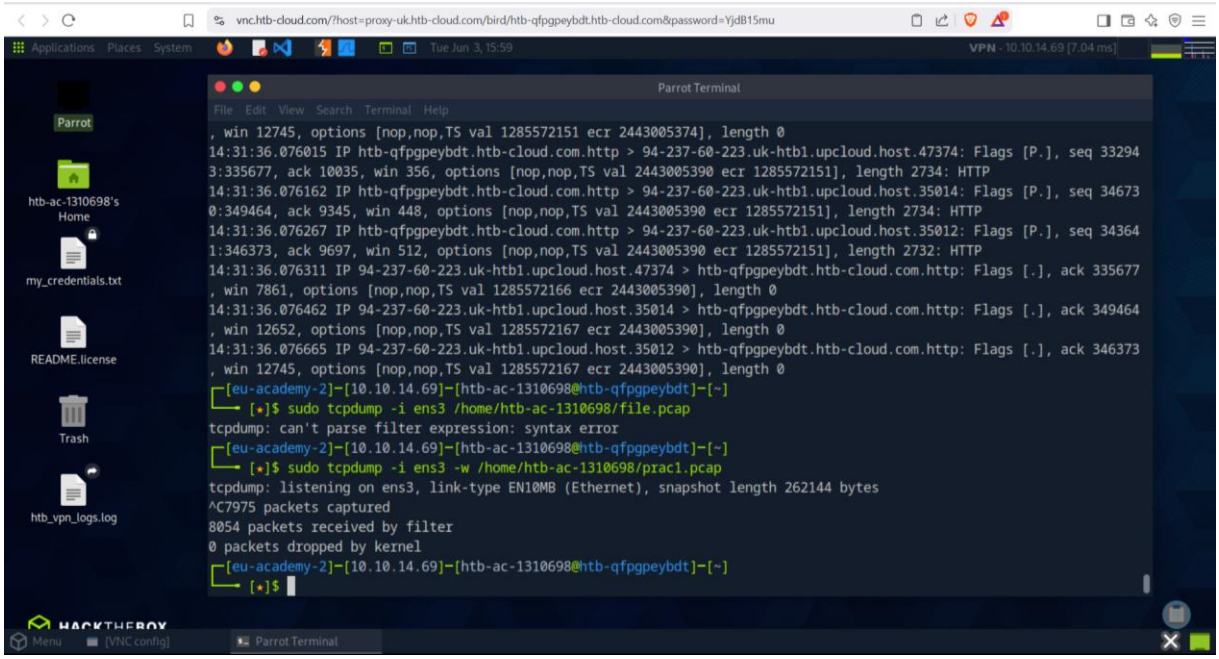
Interrogating Network Traffic With Capture and Display Filters

This lab game me some exposure to interrogating network traffic and some valuable practice implementing packet filters. I started by initiating a tcpdump capture without filters and saved it on file.pcap



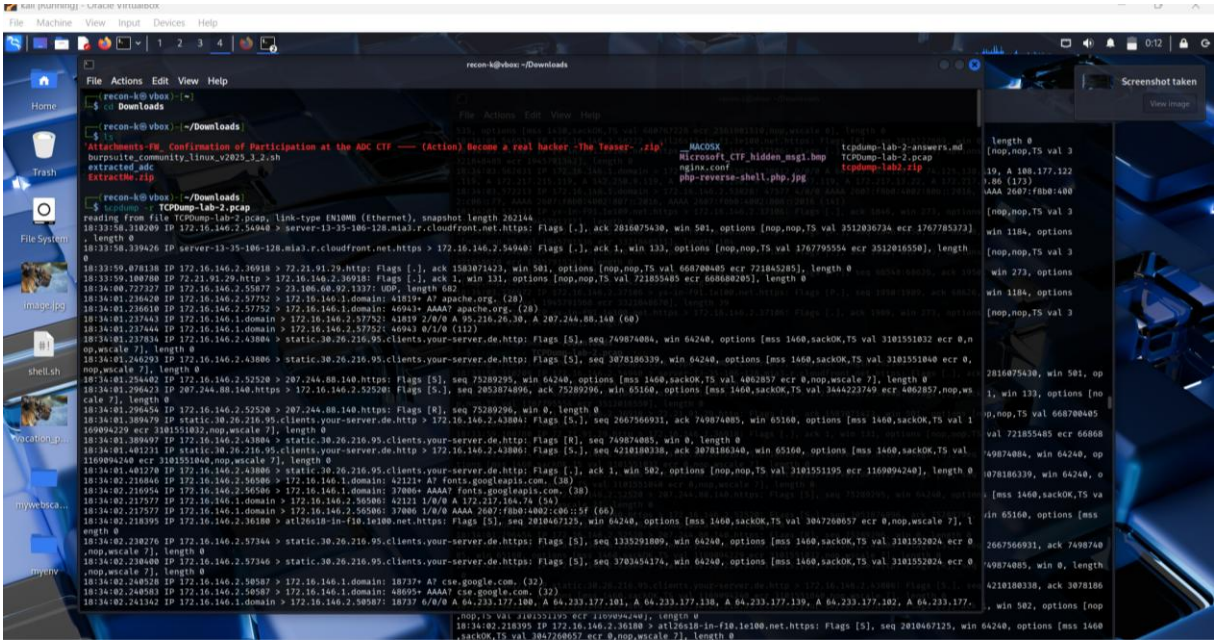
Reading a capture from a file without filters applied

Tcpdump -r prac1.pcap

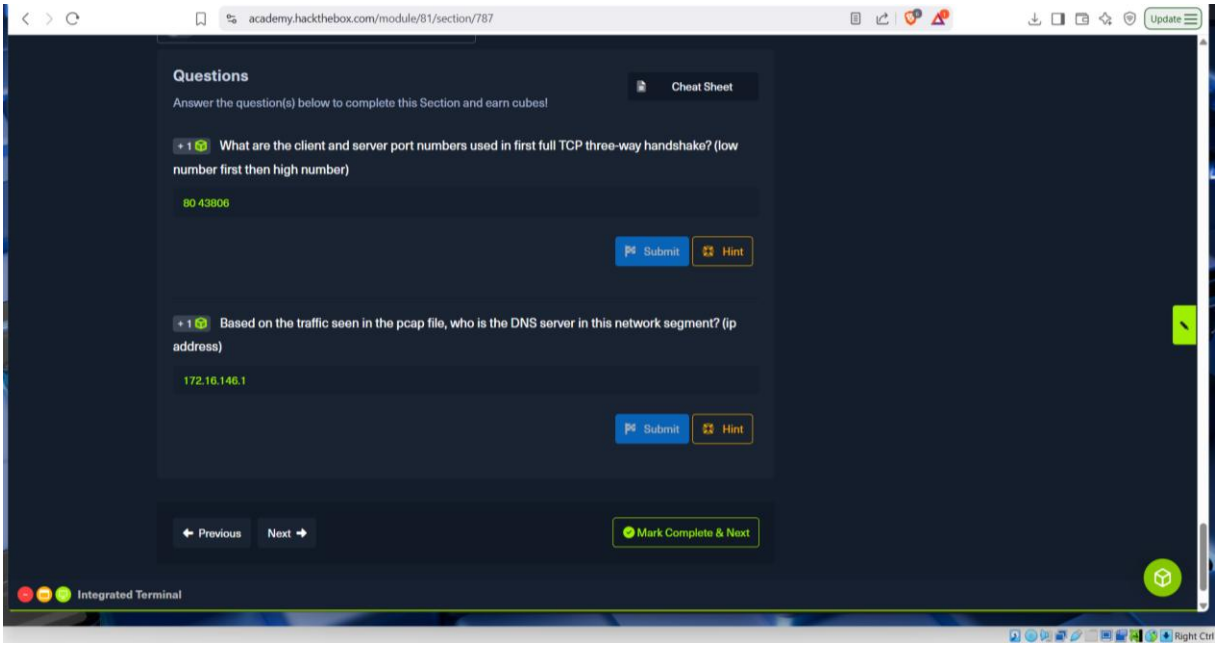


Questions:

After Downloading the Tcpdump zip file i analyzed the .pcap file in kali linux vm in order to answer the questions that follow:



1. What are the client and server port numbers used in first full TCP three-way handshake? (low number first then high number) 80, 43806
2. Based on the traffic seen in the pcap file, who is the DNS server in this network segment? (ip address) 172.16.146.1



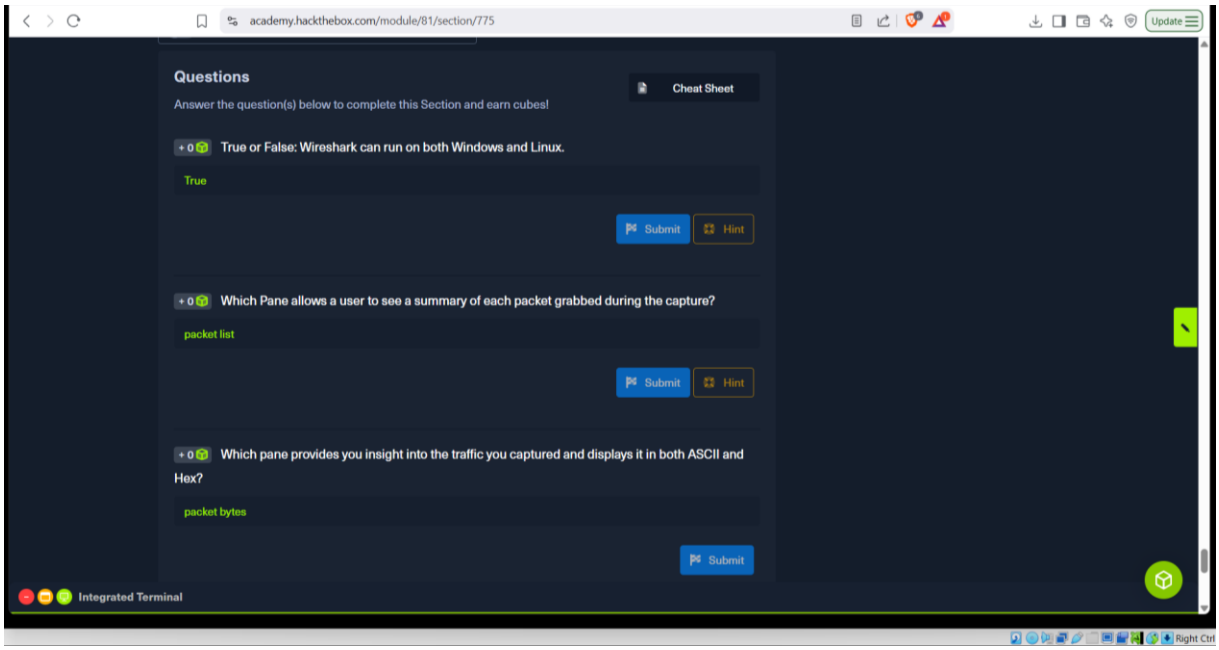
Wireshark

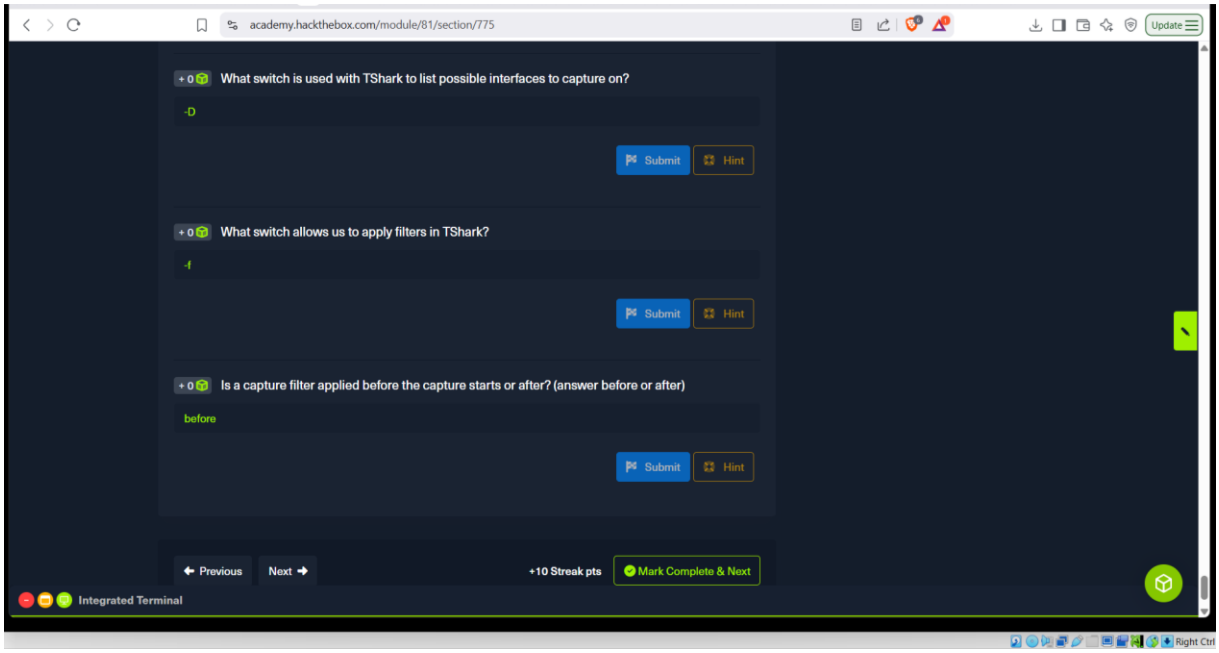
Covered the GUI based wireshark and terminal T-shark, including its basic switches.
Termshark was also gone through.
Basic usage of the tools was covered including application of filters.

Questions

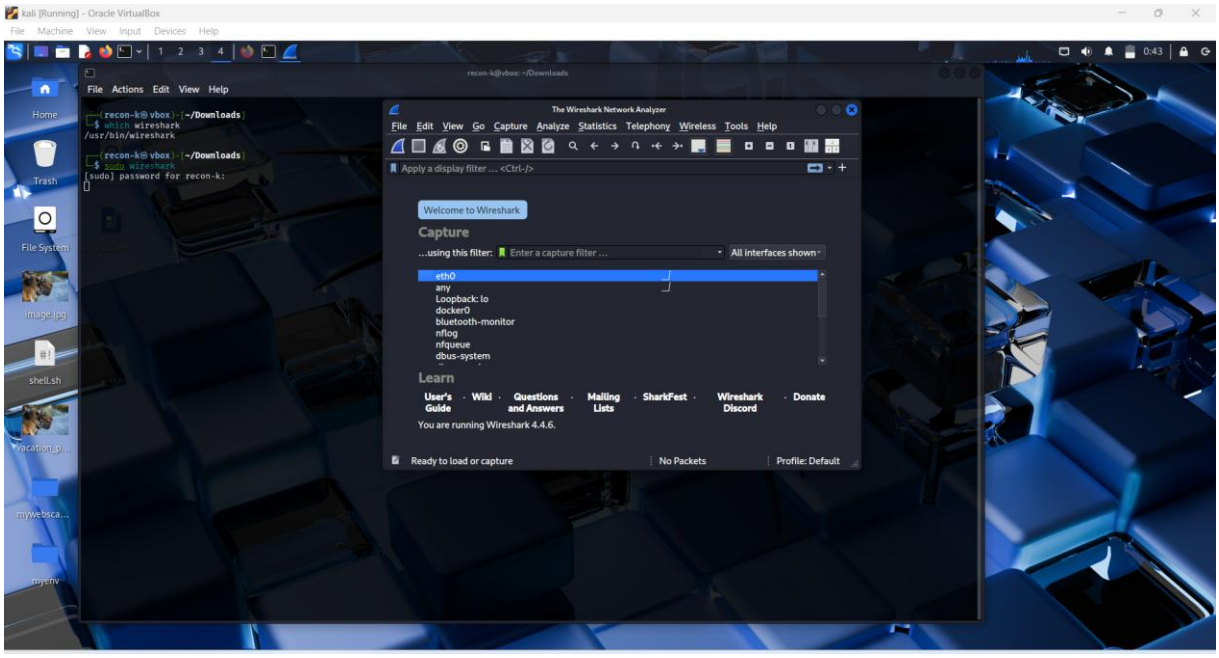
Analysis with Wireshark

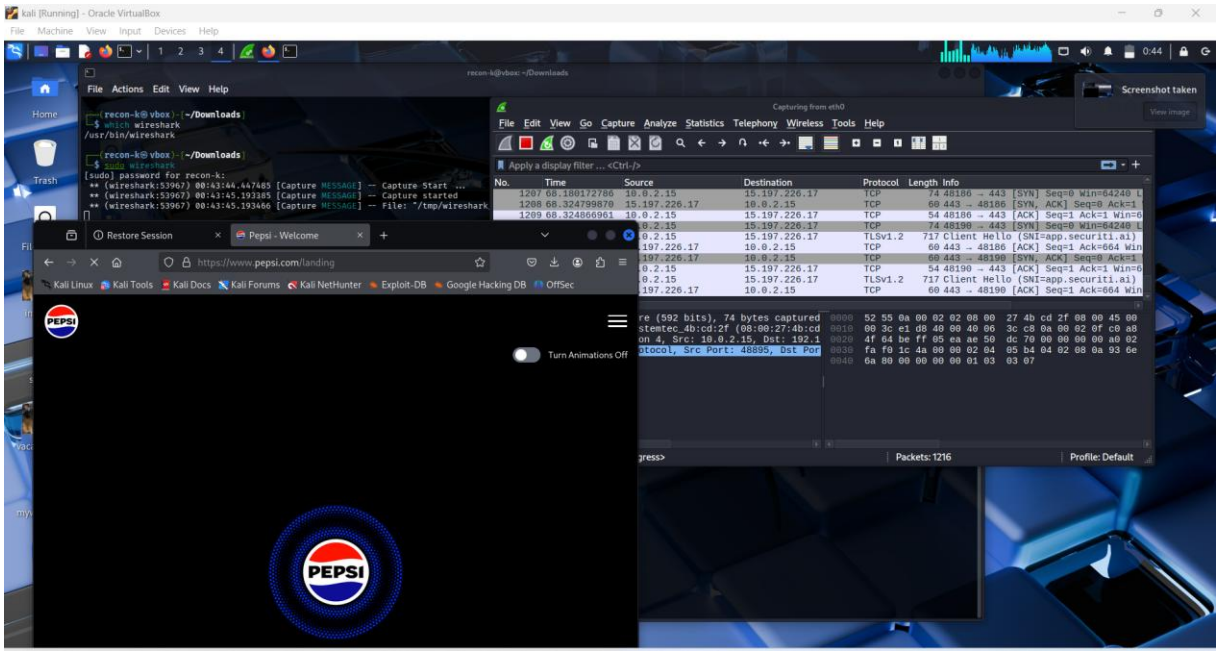
True or False: Wireshark can run on both Windows and Linux.: True
Which Pane allows a user to see a summary of each packet grabbed during the capture? Packet list
Which pane provides you insight into the traffic you captured and displays it in both ASCII and Hex? Packet bytes
What switch is used with TShark to list possible interfaces to capture on?
What switch allows us to apply filters in TShark? -f
Is a capture filter applied before the capture starts or after? (answer before or after) before

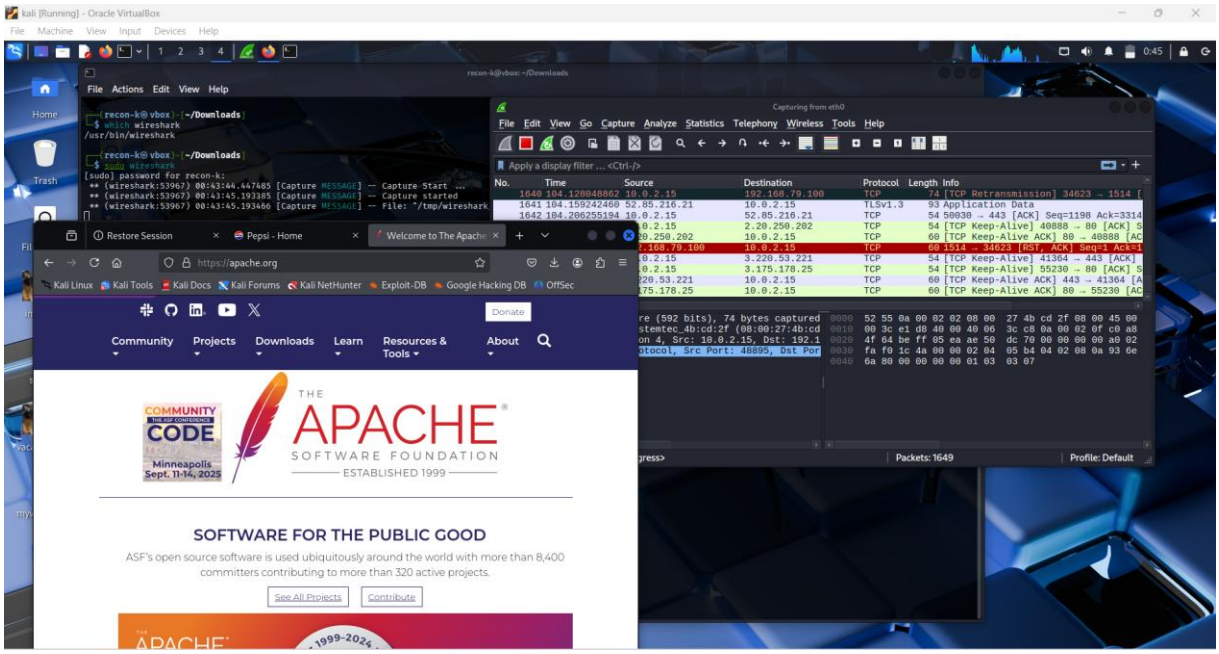




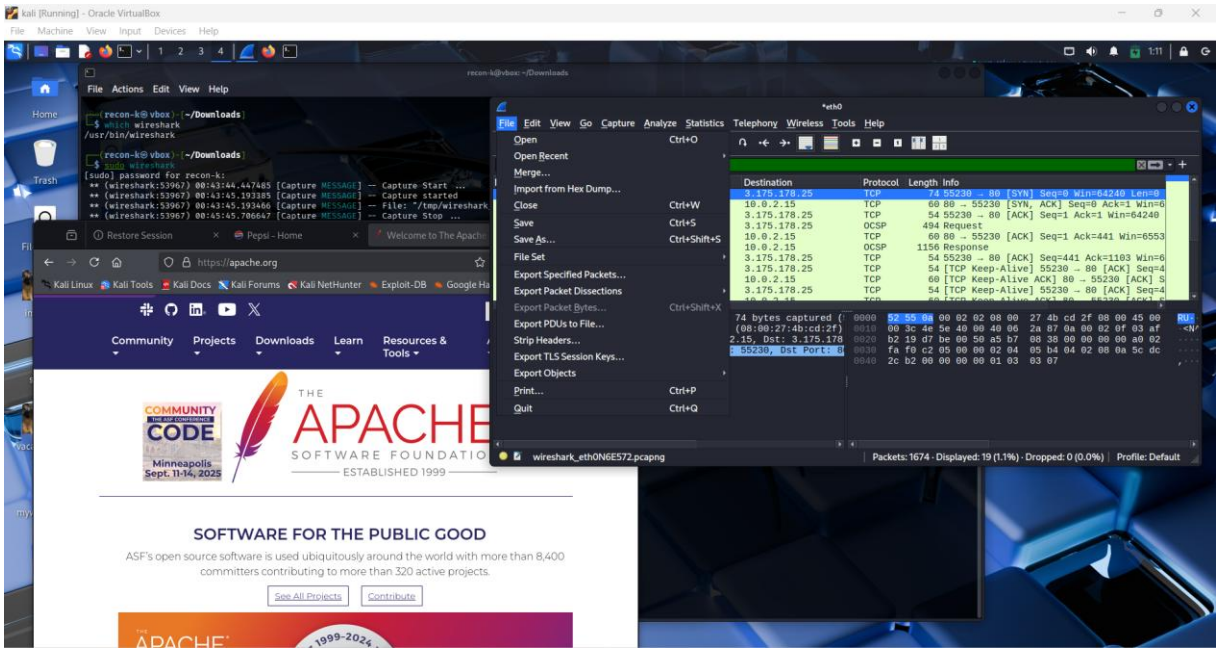
Farmiliarity with Wireshark







Wireshark advance usage



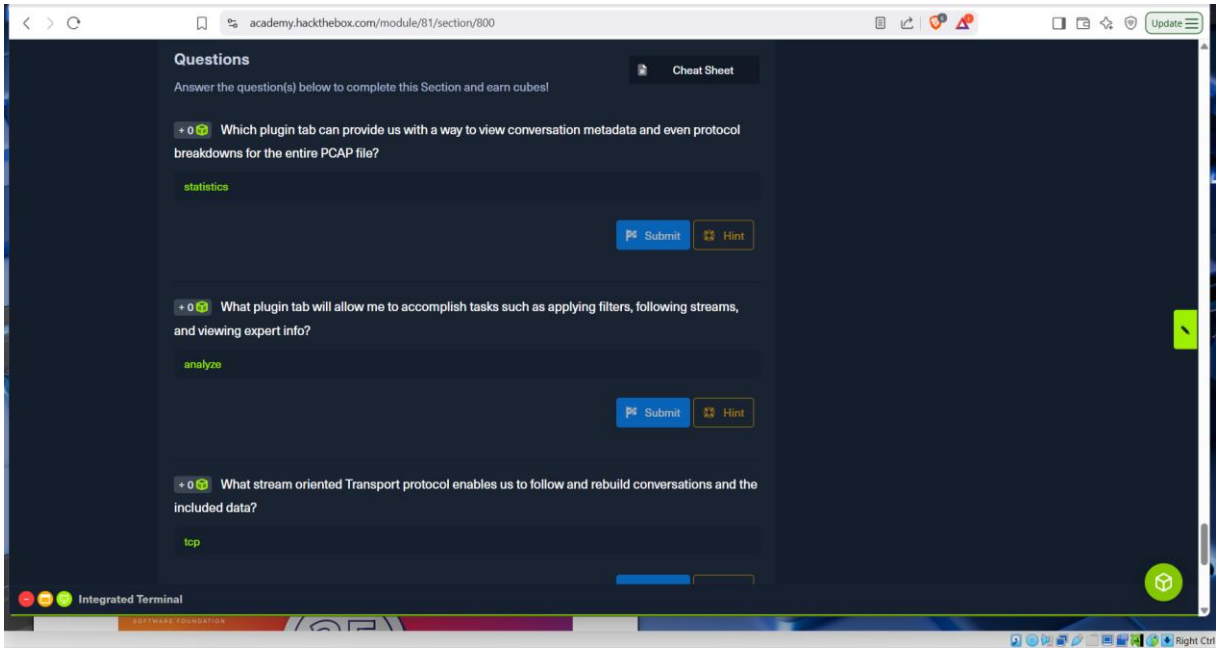
Which plugin tab can provide us with a way to view conversation metadata and even protocol breakdowns for the entire PCAP file? Statistics

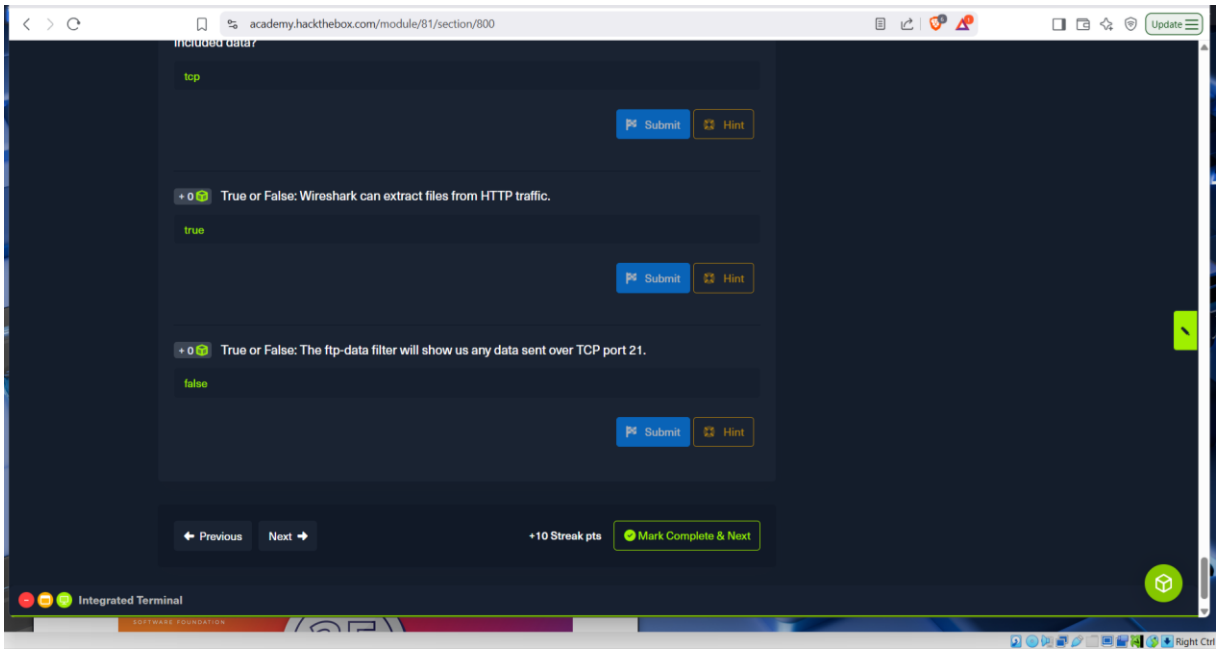
What plugin tab will allow me to accomplish tasks such as applying filters, following streams, and viewing expert info? Analyze

What stream oriented Transport protocol enables us to follow and rebuild conversations and the included data? Tcp

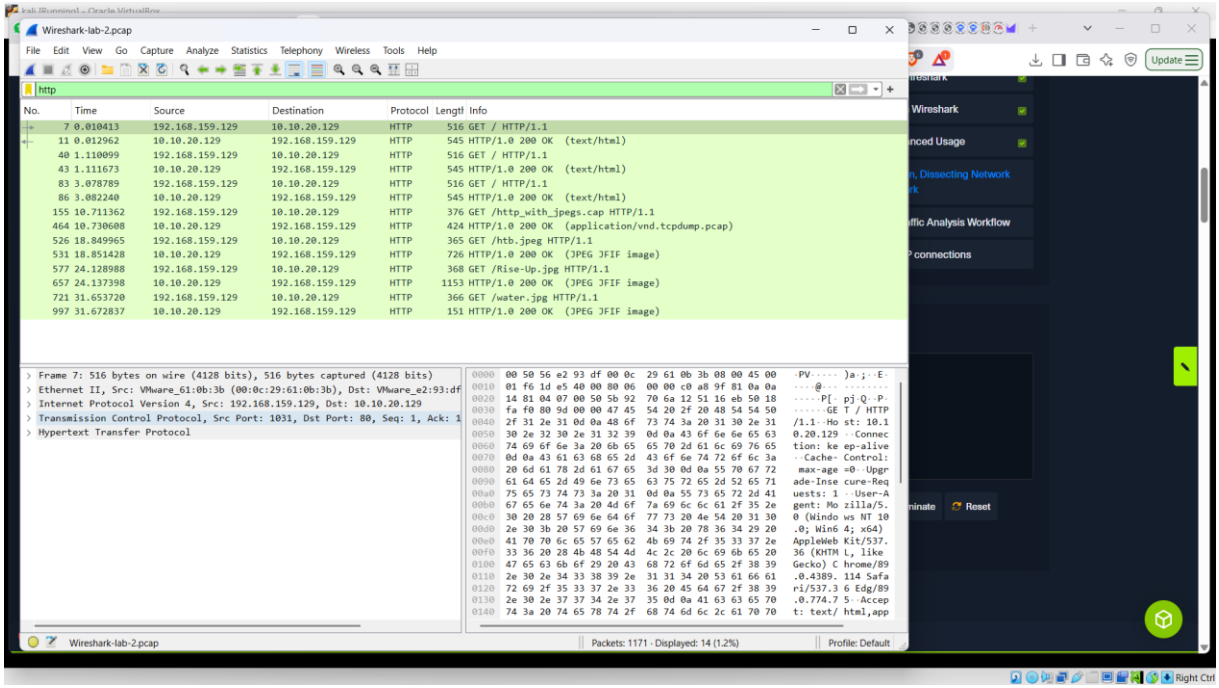
True or False: Wireshark can extract files from HTTP traffic. True

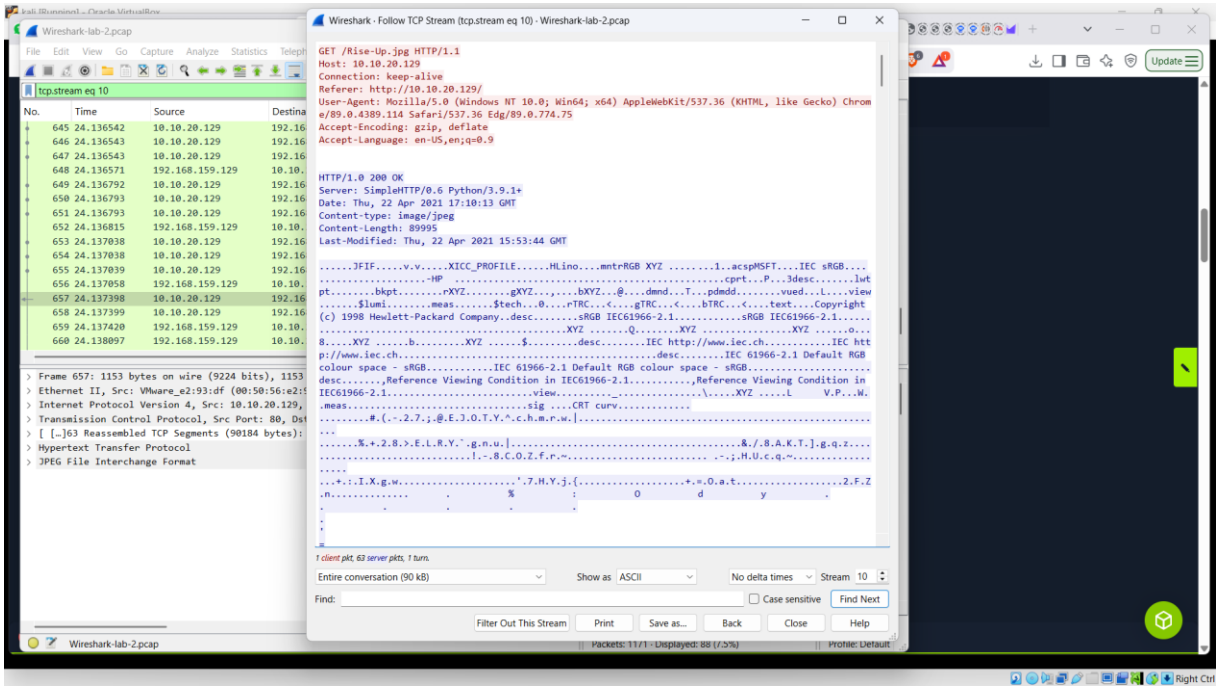
True or False: The ftp-data filter will show us any data sent over TCP port 21. false


academy.hackthebox.com/module/81/section/800
Update
Questions
Cheat Sheet
Answer the question(s) below to complete this Section and earn cubes!
+0 Which plugin tab can provide us with a way to view conversation metadata and even protocol breakdowns for the entire PCAP file?
statistics
Submit
Hint
+0 What plugin tab will allow me to accomplish tasks such as applying filters, following streams, and viewing expert info?
analyze
Submit
Hint
+0 What stream oriented Transport protocol enables us to follow and rebuild conversations and the included data?
tcp
Integrated Terminal
Right Ctrl


academy.hackthebox.com/module/81/section/800
Update
tcp
Submit
Hint
+0 True or False: Wireshark can extract files from HTTP traffic.
true
Submit
Hint
+0 True or False: The ftp-data filter will show us any data sent over TCP port 21.
false
Submit
Hint
Previous
Next
+10 Streak pts
Mark Complete & Next
Integrated Terminal
Right Ctrl

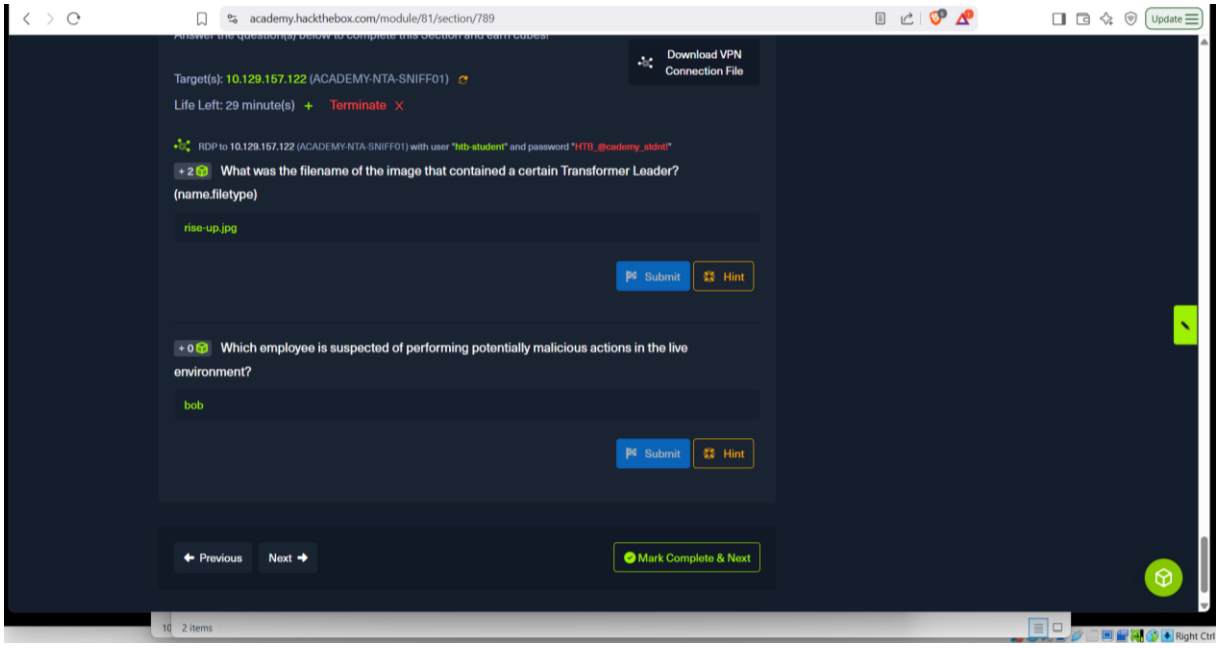
Packet Inception, Dissecting Network Traffic With Wireshark





What was the filename of the image that contained a certain Transformer Leader? (name.filetype) rise-up.jpg

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

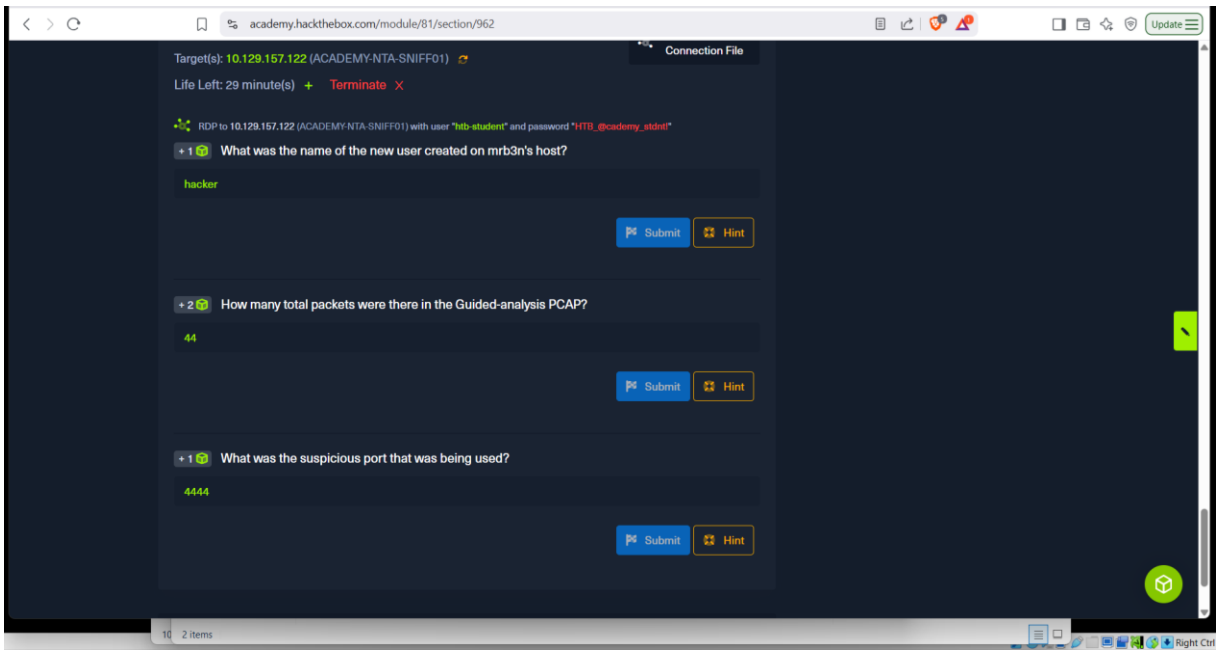


Guided lab : Traffic analysis workflow

What was the name of the new user created on mrb3n's host? Hacker

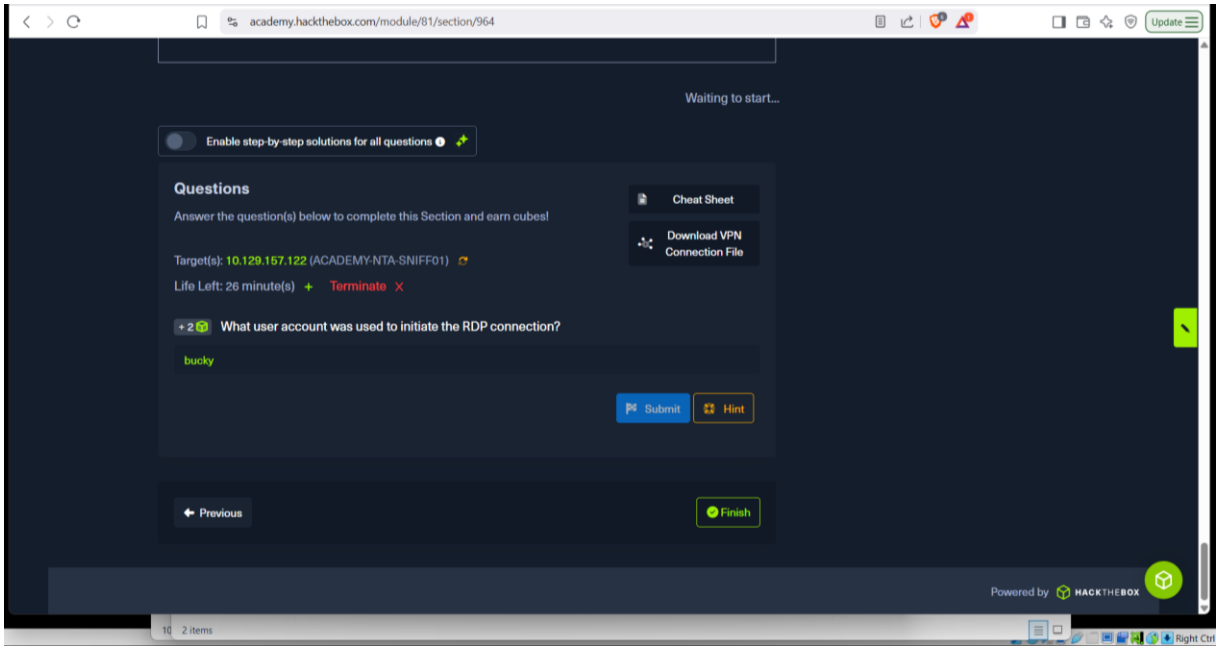
How many total packets were there in the Guided-analysis PCAP? 44

What was the suspicious port that was being used? 4444

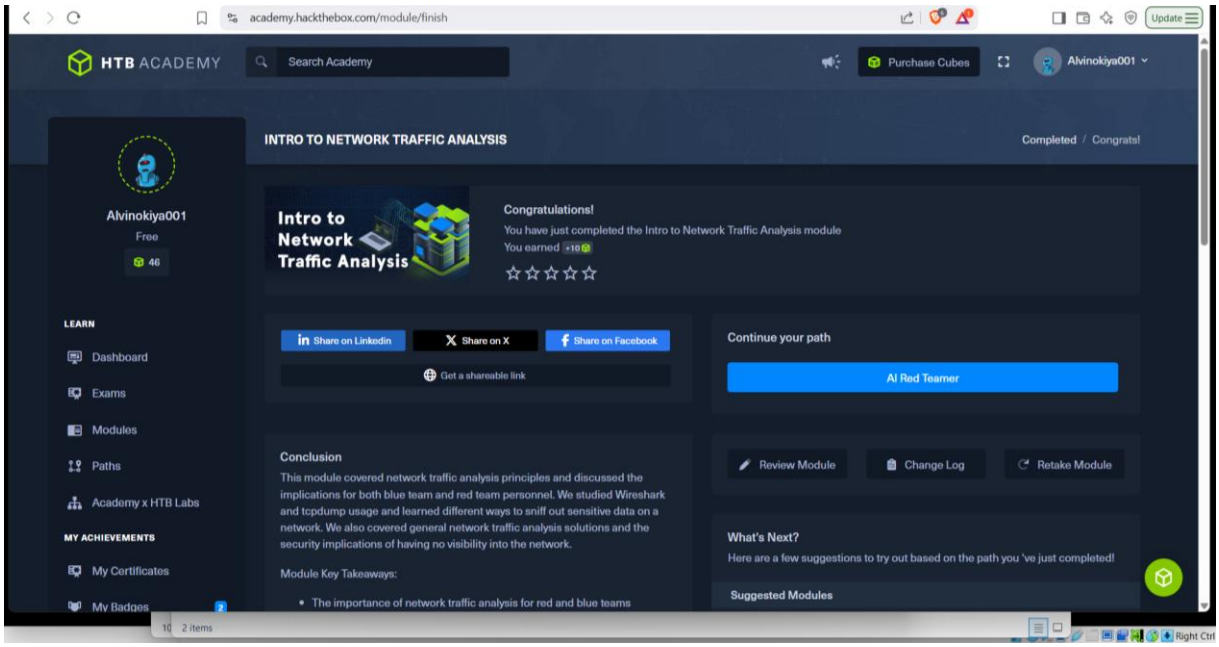


Decrypting RDP Connections

What user account was used to initiate the RDP connection? Bucky



Conclusion



Doing and completing this assignment provided valuable insight into the structure and

behavior of network traffic. Working with tcpdump allowed me to grasp the raw fundamentals of packet capture and filtering, while Wireshark offered a deeper, more visual perspective into the intricacies of protocols and session analysis.

I now feel more confident in navigating the network analysis workflow from capturing packets to interpreting them across the OSI layers. The final module on RDP decryption was particularly eye-opening, showing how encrypted traffic can still be analyzed under the right conditions. These skills are essential not just for monitoring and troubleshooting networks, but also for uncovering potential threats during security assessments or incident response.