CLOUD AND NETWORK SECURITY

CHARLES GITHINJI

CS-CNS07-24006

Week 2: Assignment 3

Hack the Box: Intro to Network Traffic Analysis

INTRODUCTIONS

The goal of this lab is to understand network traffic analysis. I will explore the fundamental concepts of network traffic analysis, learn the basics of Tcpdump and Wireshark, and how to use various filters in these tools. Through hands-on experiencewith Tcpdump and wireshark, I will gain the skills necessary to perform effective network traffic analysis. The specific objectives of this lab are;
This lab helps me learn how to implement network traffic analysis using Wireshark and Tcpdump.

1. Grasp the principles behind network analysis
2. Understand the key features of tcpdump

3. Learn to use Wireshark
4. Apply filters in Wireshark for more detailed analysis

Part 1: Network Traffic Analysis Introduction

Network Traffic Analysis (NTA) can be described as the act of examining network traffic to characterize common ports and protocols utilized, establish a baseline for our environment, monitor and respond to threats, and ensure the greatest possible insight into our organization's network.

Required Skills and Knowledge

1. TCP/IP Stack & OSI Model
2. Basic Network Concepts
3. Common Ports and Protocols
4. Concepts of IP Packets and the Sublayers
5. Protocol Transport Encapsulation

Environment and Equipment

Common Traffic Analysis Tools

Tool	Description

Tcpdump- *tcpdump i*s 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.

Performing Network Traffic Analysis

NTA Workflow

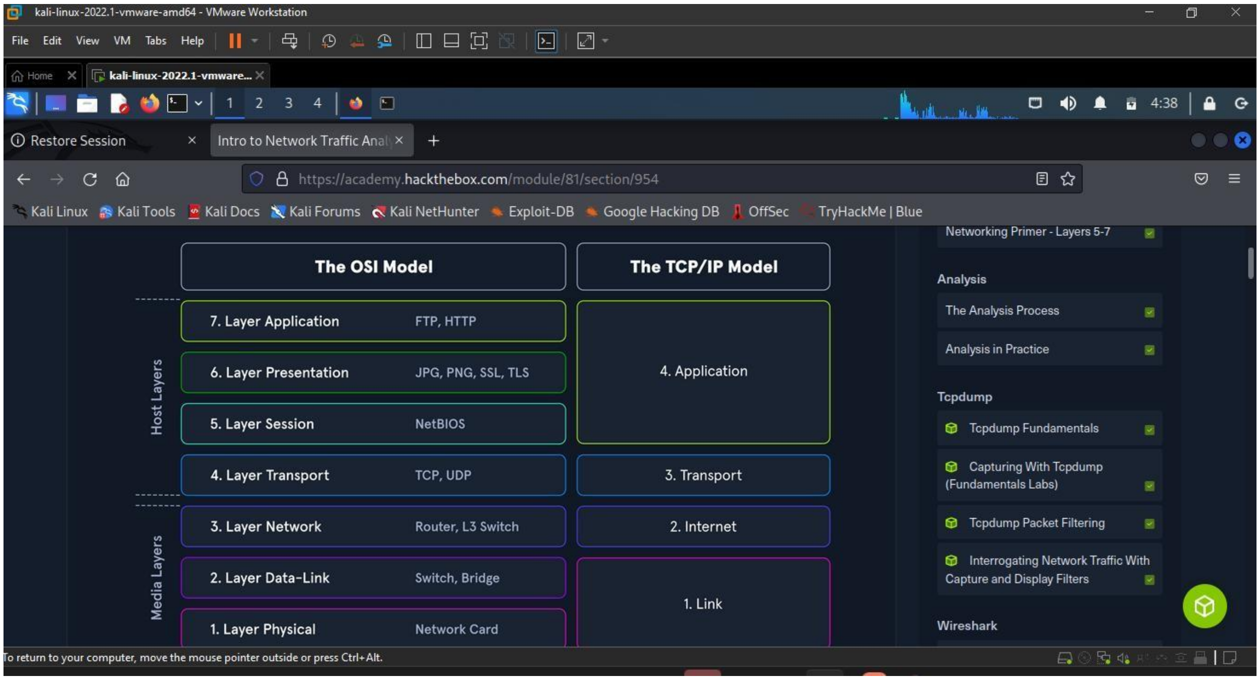
Traffic analysis, or NTA, is a method used to study and understand network traffic. It involves examining and analyzing the flow of data packets in a network to identify patterns, trends, and anomalies. NTA is not an exact science because it can be influenced by various factors, such as the goals of the analysis (whether it's detecting network errors or malicious actions) and the level of visibility into the network.

Networking Primer - Layers 1-4

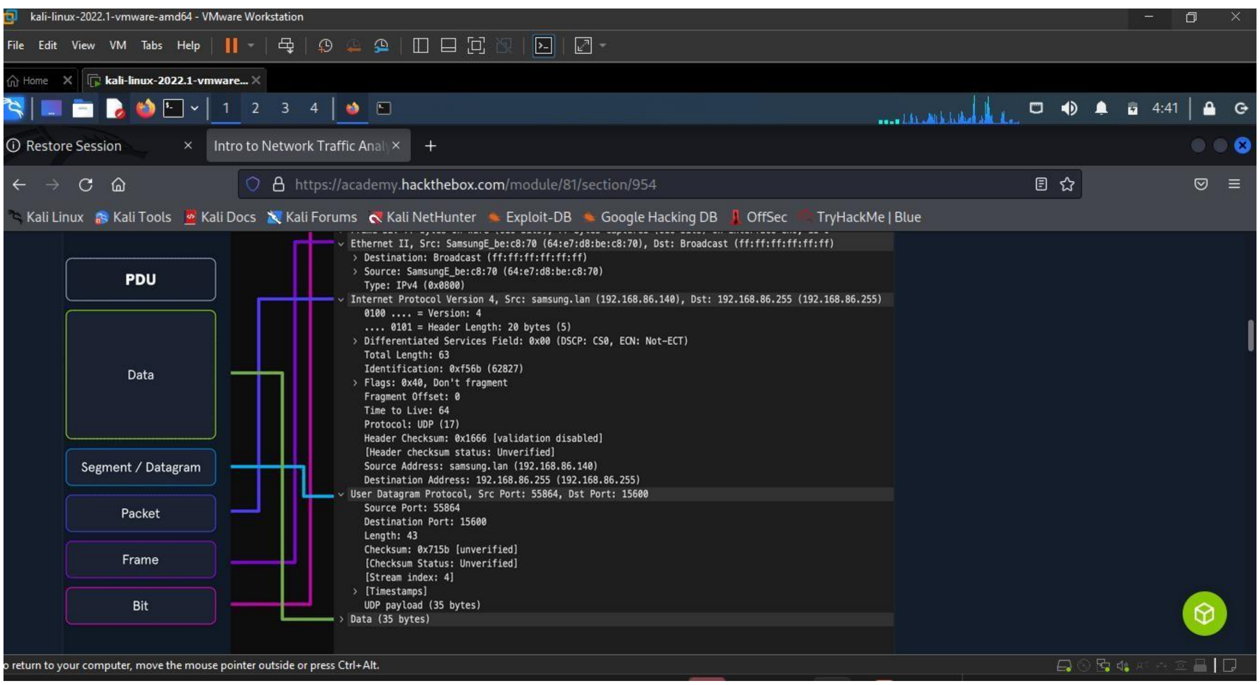
OSI / TCP-IP Models

The image above gives a great view of the Open Systems Interconnect (OSI) model and the Transmission Control Protocol - Internet Protocol (TCP-IP) model side by side.

PDU Example



PDU Packet Breakdown

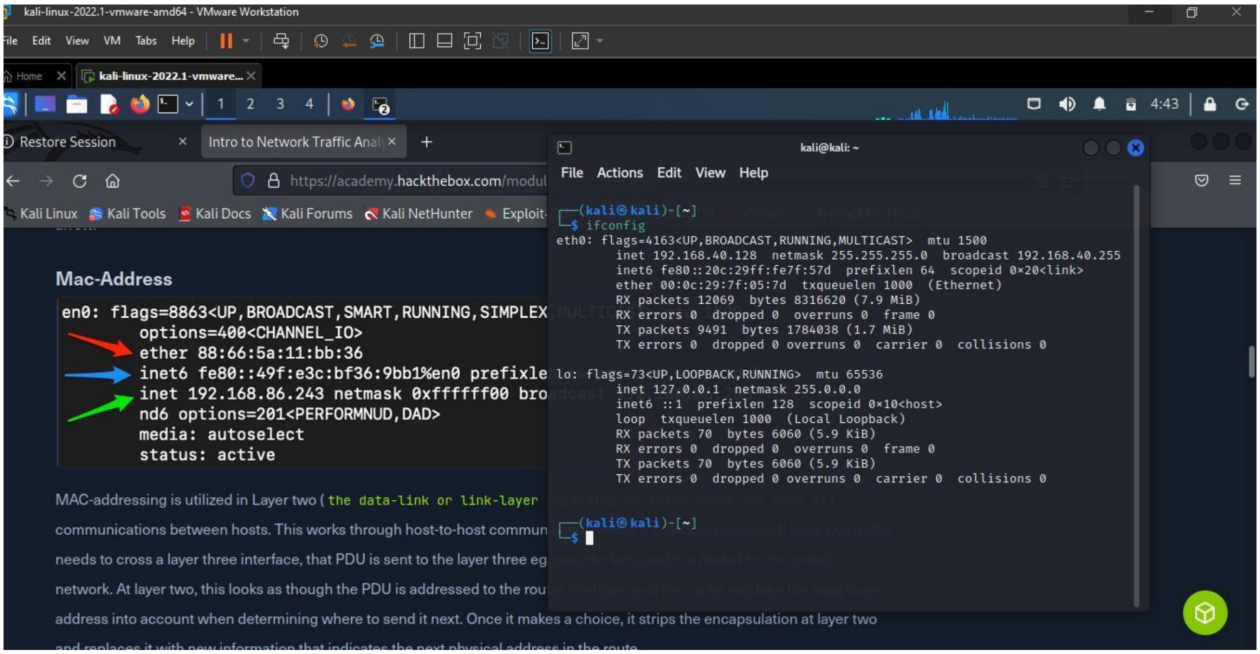


The image above shows the makeup of a PDU side by side with a packet breakout from Wireshark's Packet Details pane.

Addressing Mechanisms

MAC-Addressing

Each logical or physical interface attached to a host has a Media Access Control (MAC) address. This address is a 48-bit six octet address represented in hexadecimal format.

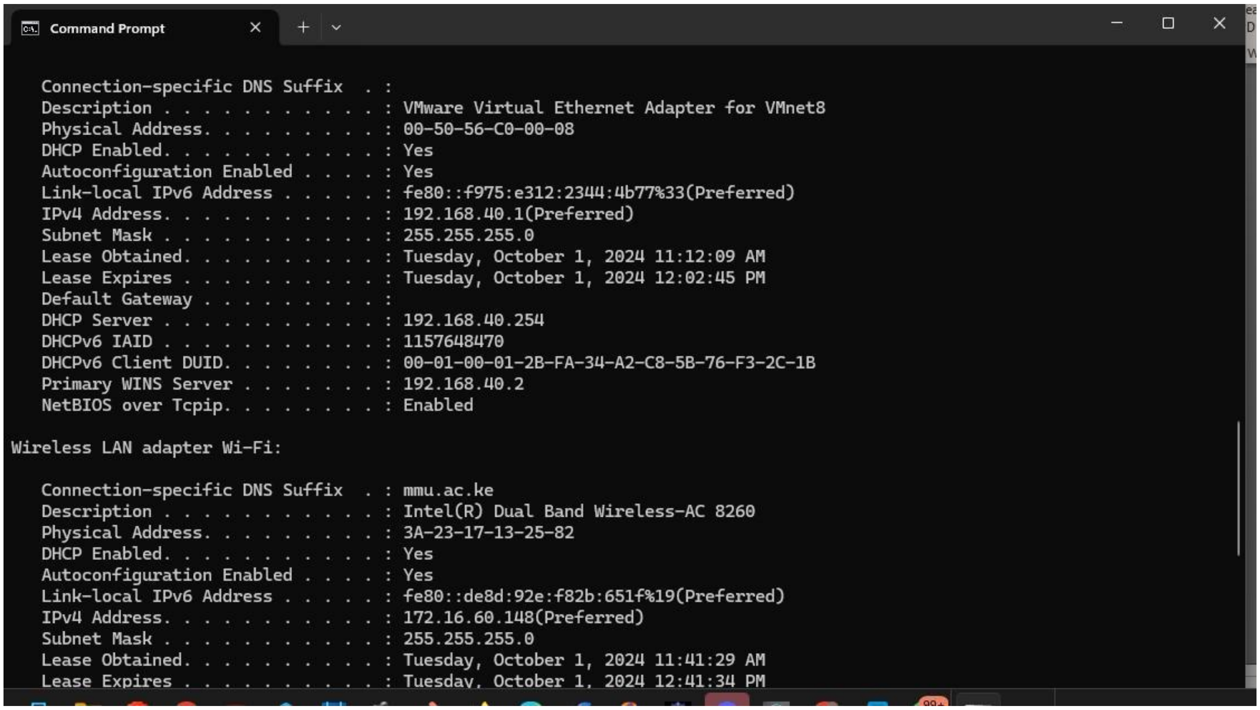


IP Addressing

The Internet Protocol (IP) was developed to deliver data from one host to another across network boundaries. IP is responsible for routing packets, the encapsulation of data, and fragmentation and reassembly of datagrams when they reach the destination host.

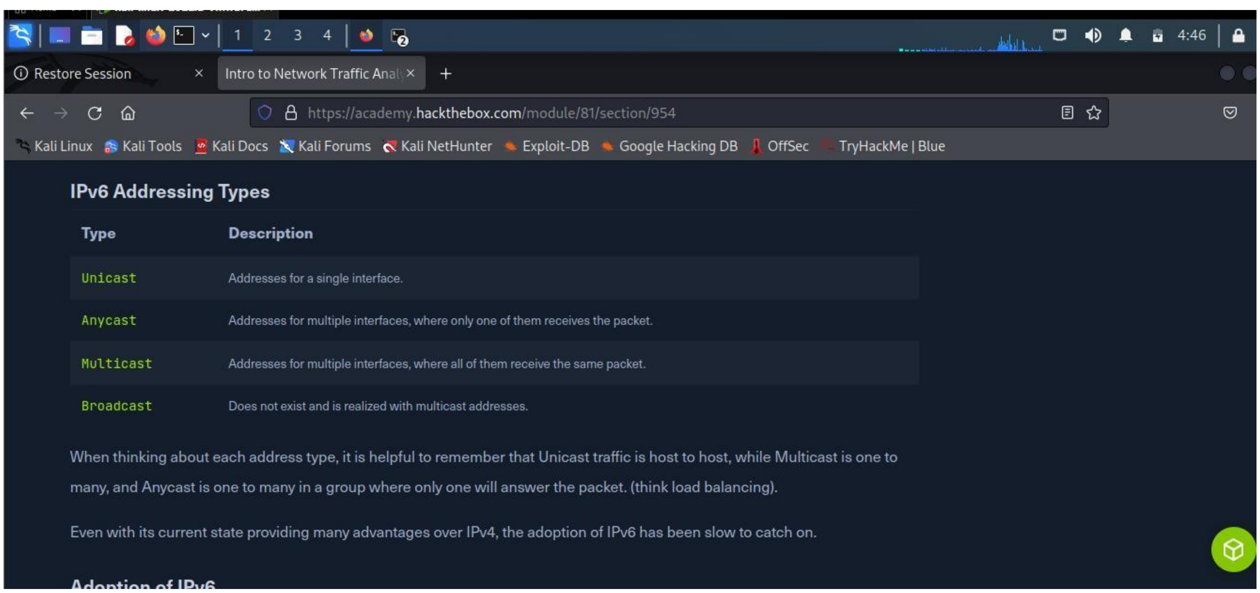
IPv4

The most common addressing mechanism most are familiar with is the Internet Protocol address version 4 (IPv4). An IPv4 address is made up of a 32-bit four octet number represented in decimal format. In my example, we can see the address 172.16.60.69.



IPv6

IPv6 provides us a much larger address space that can be utilized for any networked purpose. IPv6 is a 128-bit address 16 octets represented in Hexadecimal format.



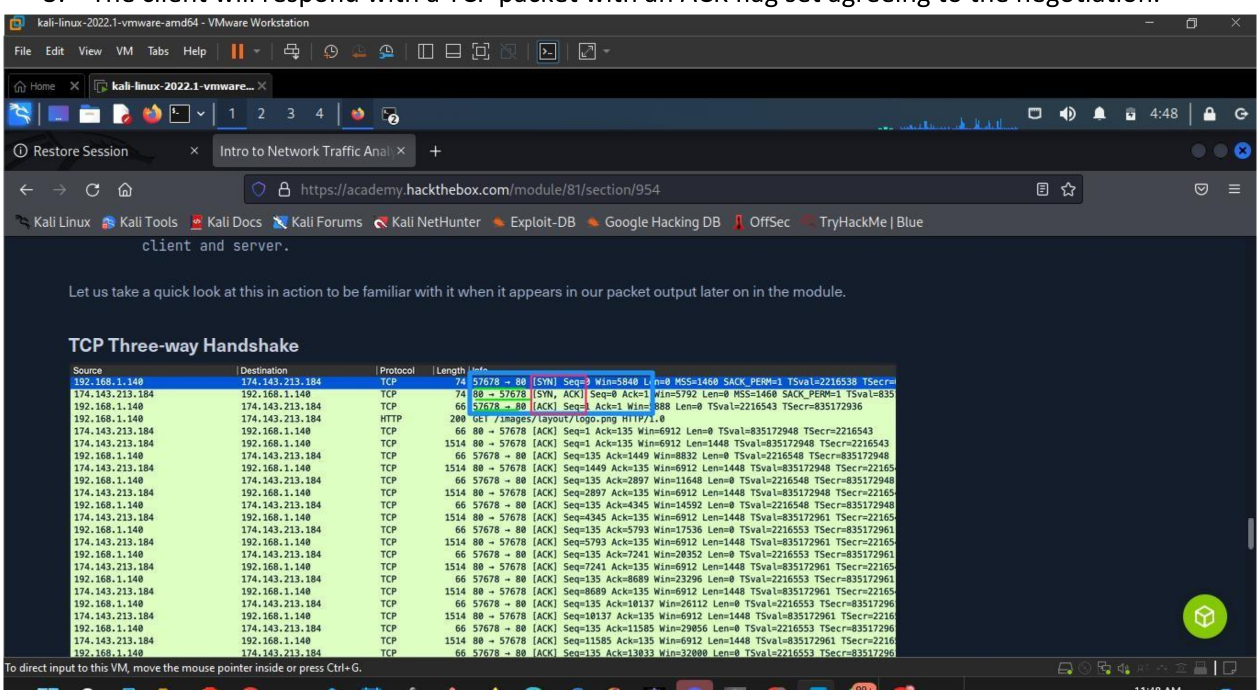
TCP / UDP, Transport Mechanisms

Characteristic	TCP	UDP

Transmission	Connection-oriented	Connectionless. Fire and forget.
Connection Establishment	TCP uses a three-way handshake to ensure that a connection is established.	UDP does not ensure the destination is listening.
Data Delivery	Stream-based conversations	packet by packet, the source does not care if the destination is active
Receipt of data	Sequence and Acknowledgement numbers are utilized to account for data.	UDP does not care
Speed	TCP has more overhead and is slower because of its built-in functions.	UDP is fast but unreliable.

TCP Three-way Handshake

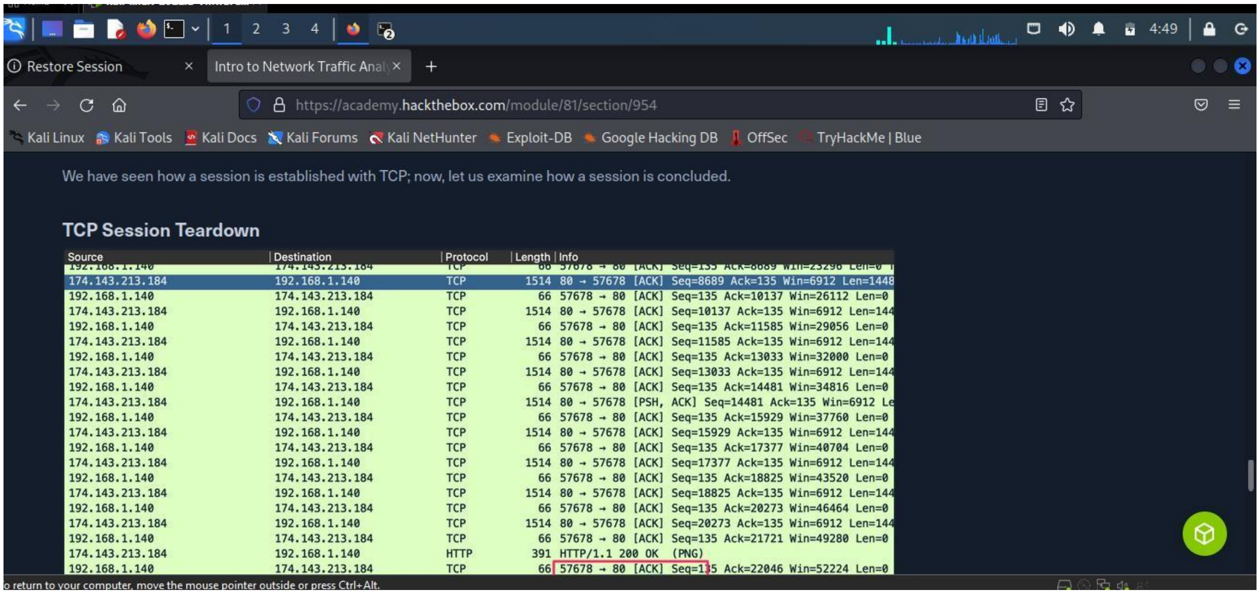
1. The client sends a packet with the SYN flag set to on along with other negotiable options in the TCP header.
2. The server will respond with a TCP packet that includes a SYN flag set for the sequence number negotiation and an ACK flag set to acknowledge the previous SYN packet sent by the host.
3. The client will respond with a TCP packet with an ACK flag set agreeing to the negotiation.



TCP Session Teardown

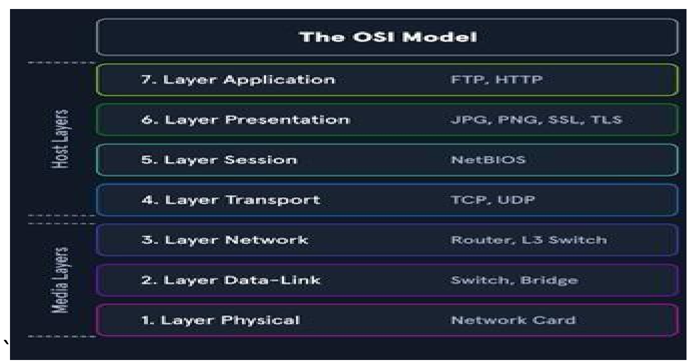
Another flag we will see with TCP is the FIN flag.

1. **FIN, ACK**
2. **FIN, ACK,**
3. **ACK**

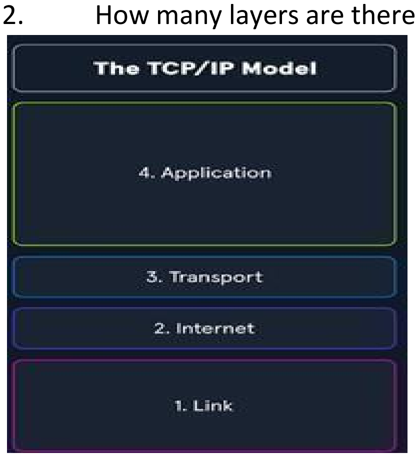


Questions

1. How many layers does the OSI model have? 7



2. How many layers are there in the TCP/IP model? 4



3. True or False: Routers operate at layer 2 of the OSI model? False

4. What addressing mechanism is used at the Link Layer of the TCP/IP model? MAC-address

MAC-addressing is utilized in Layer two (`the data-link or link-layer depending on which model you look at`) communications between hosts. This works through host-to-host communication within a broadcast domain. If layer two traffic needs to cross a layer three interface, that PDU is sent to the layer three egress interface, and it is

5. At what layer of the OSI model is a PDU encapsulated into a packet? (the number) 3

3. Layer Network Router, L3 Switch	2. Internet	Packet

6. What addressing mechanism utilizes a 32-bit address? IPv4

An IPv4 address is made up of a 32-bit `four octet` number represented in decimal format. In our example, we can see the address `192.168.86.243`. Each octet of an IP address can be represented by a number ranging from `0` to `255`. When examining a PDU, we will find IP addresses in layer three (`Network`) of the OSI model and layer two (`internet`) of the TCP-IP model. We will not deep dive into IPv4 here, but for the sake of this module, understand what these addresses are, what they do for us, and at which layer they are used.

7. What Transport layer protocol is connection oriented? TCP

8. What Transport Layer protocol is considered unreliable? UDP

TCP VS. UDP

Characteristic	TCP	UDP
`Transmission`	Connection-oriented	Connectionless. Fire and forget.

9. TCP's three-way handshake consists of 3 packets: 1.Syn, 2.Syn & ACK, 3. _? What is the final packet of the handshake? ACK

3. The `client` will respond with a TCP packet with an ACK flag set agreeing to the negotiation.

    1. This packet is the end of the three-way handshake and established the connection between client and server.

Networking Prime – Layers 5-7

Questions

1. What is the default operational mode method used by FTP? Active

> FTP is capable of running in two different modes, `active` or `passive`. Active is the default operational method utilized by FTP, meaning that the server listens for a control command `PORT` from the client, stating what port to use for data transfer. Passive mode enables us to access FTP servers located behind firewalls or a NAT-enabled link that makes direct TCP connections impossible. In this instance, the client would send the `PASV` command and wait for a response from the server informing the client what IP and port to utilize for the data transfer channel connection.

2. FTP utilizes what two ports for command and data transfer? (separate the two numbers with a space) 20 21

> When we think about communication between hosts, we typically think about a client and server talking over a single socket. Through this socket, both the client and server send commands and data over the same link. In this aspect, FTP is unique since it utilizes multiple ports at a time. FTP uses ports 20 and 21 over TCP. Port 20 is used for data transfer, while port 21 is utilized for issuing commands controlling the FTP session. In regards to authentication, FTP supports user authentication as well as allowing anonymous access if configured.

3. Does SMB utilize TCP or UDP as its transport layer protocol? TCP

> resource or perform actions. In the past, SMB utilized NetBIOS as its transport mechanism over UDP ports 137 and 138. Since modern changes, SMB now supports direct TCP transport over port 445, NetBIOS over TCP port 139, and even the QUIC protocol.

4. SMB has moved to using what TCP port? 445

> resource or perform actions. In the past, SMB utilized NetBIOS as its transport mechanism over UDP ports 137 and 138. Since modern changes, SMB now supports direct TCP transport over port 445, NetBIOS over TCP port 139, and even the QUIC protocol.

5. Hypertext Transfer Protocol uses what well known TCP port number? 80

> media (HTML, images, hyperlinks, video). HTTP utilizes ports 80 or 8000 over TCP during normal operations. In exceptional circumstances, it can be modified to use alternate ports, or even at times, UDP.

6. What HTTP method is used to request information and content from the webserver? Get

> `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.

7. What web based protocol uses TLS as a security measure? HTTPS

> **HTTPS**
>
> HTTP Secure (`HTTPS`) is a modification of the HTTP protocol designed to utilize Transport Layer Security (`TLS`) or Secure Sockets Layer (`SSL`) with older applications for data security. TLS is utilized as an encryption mechanism to secure the communications between a client and a server. TLS can wrap regular HTTP traffic within TLS, which

8. True or False: when utilizing HTTPS, all data sent across the session will appear as TLS Application data? True

> HTTP Secure (`HTTPS`) is a modification of the HTTP protocol designed to utilize Transport Layer Security (`TLS`) or Secure Sockets Layer (`SSL`) with older applications for data security. TLS is utilized as an encryption mechanism to secure the communications between a client and a server. TLS can wrap regular HTTP traffic within TLS, which means that we can encrypt our entire conversation, not just the data sent or requested. Before the TLS mechanism was in place, we were vulnerable to Man-in-the-middle attacks and other types of reconnaissance or hijacking, meaning anyone in the same LAN as the client or server could view the web traffic if they were listening on the wire. We can now have security implemented in the browser enabling everyone to encrypt their web habits, search requests, sessions or data transfers, bank transactions, and much more.

THE ANALYSIS PROCESS

Network Traffic Analysis (NTA) is a dynamic process aimed at examining network data to detect anomalies and potential threats. It involves breaking down traffic into understandable chunks, identifying deviations from normal patterns, and flagging suspicious activities such as unauthorized remote access (e.g., via RDP, SSH, or Telnet). The analysis helps set a baseline for typical network behavior, making it easier to spot unusual trends.

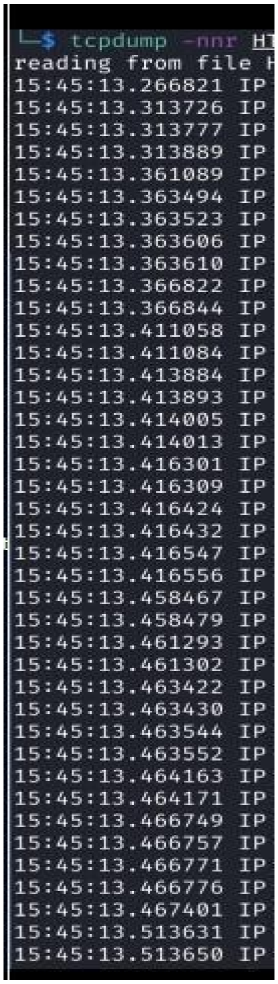
NTA is crucial for both defense and daily operations, providing visibility into network usage, top-talking hosts, and internal communications. It aids in troubleshooting, detecting issues, and ensuring protocols function correctly. Tools like IDS/IPS, firewalls, and logging systems (e.g., Splunk or ELK Stack) enhance NTA by quickly alerting on known attacks, but human oversight remains essential, as attackers constantly find ways to bypass automated defenses. Combining automated tools with manual checks ensures more effective monitoring and protection.

Part 2: Tcpdump Fundamentals

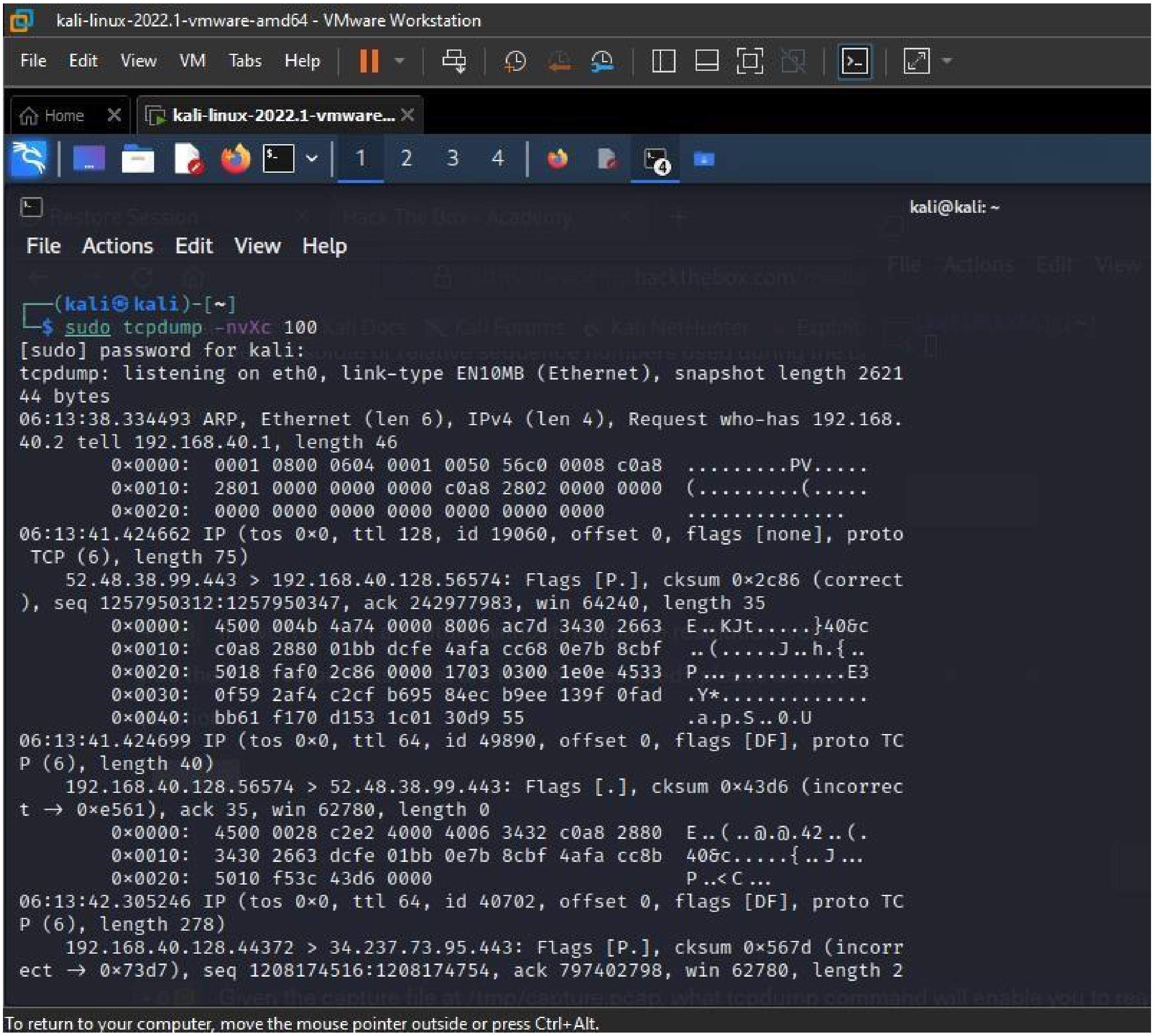
1. Utilizing the output shown in question-1.png, who is the server in this communication? (IP Address)
   174.143.213.184

```
└─$ 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
```

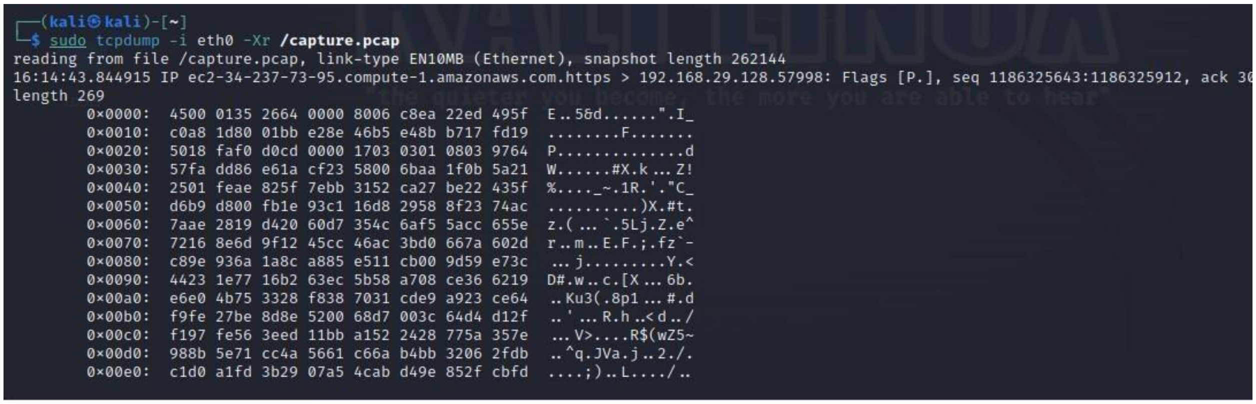
2. Were absolute or relative sequence numbers used during the capture? (see question-1.zip to answer)
   relative



3. 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

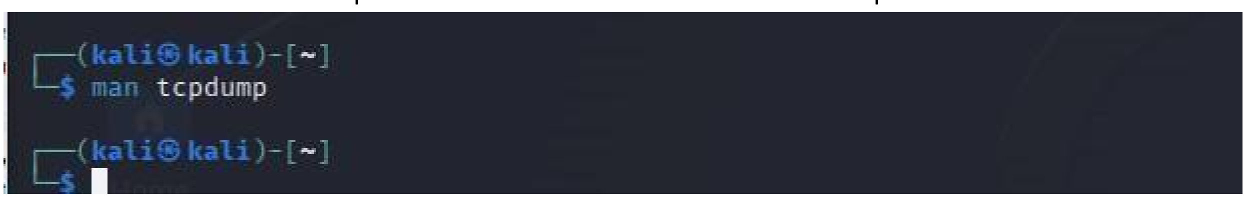


4. Given the capture file at /tmp/capture.pcap, what tcpdump command will enable you to read from the capture and show the output contents in Hex and ASCII? (Please use best practices when using switches)
   sudo tcpdump -Xr /tmp/capture.pcap

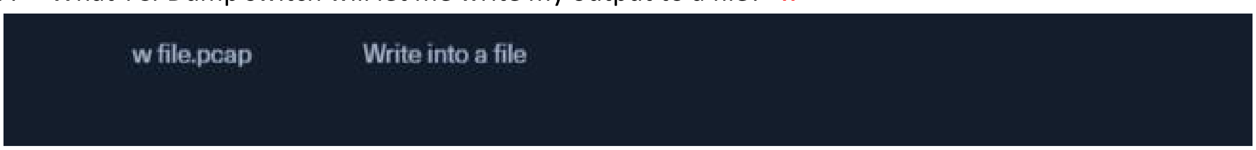


5. What TCPDump switch will increase the verbosity of our output? (Include the - with the proper switch) –v

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



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

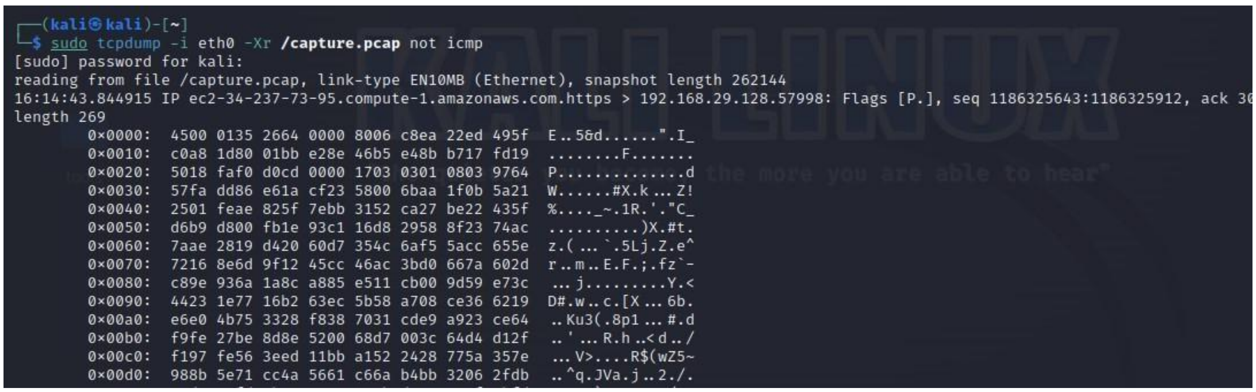


Fundamentals Lab

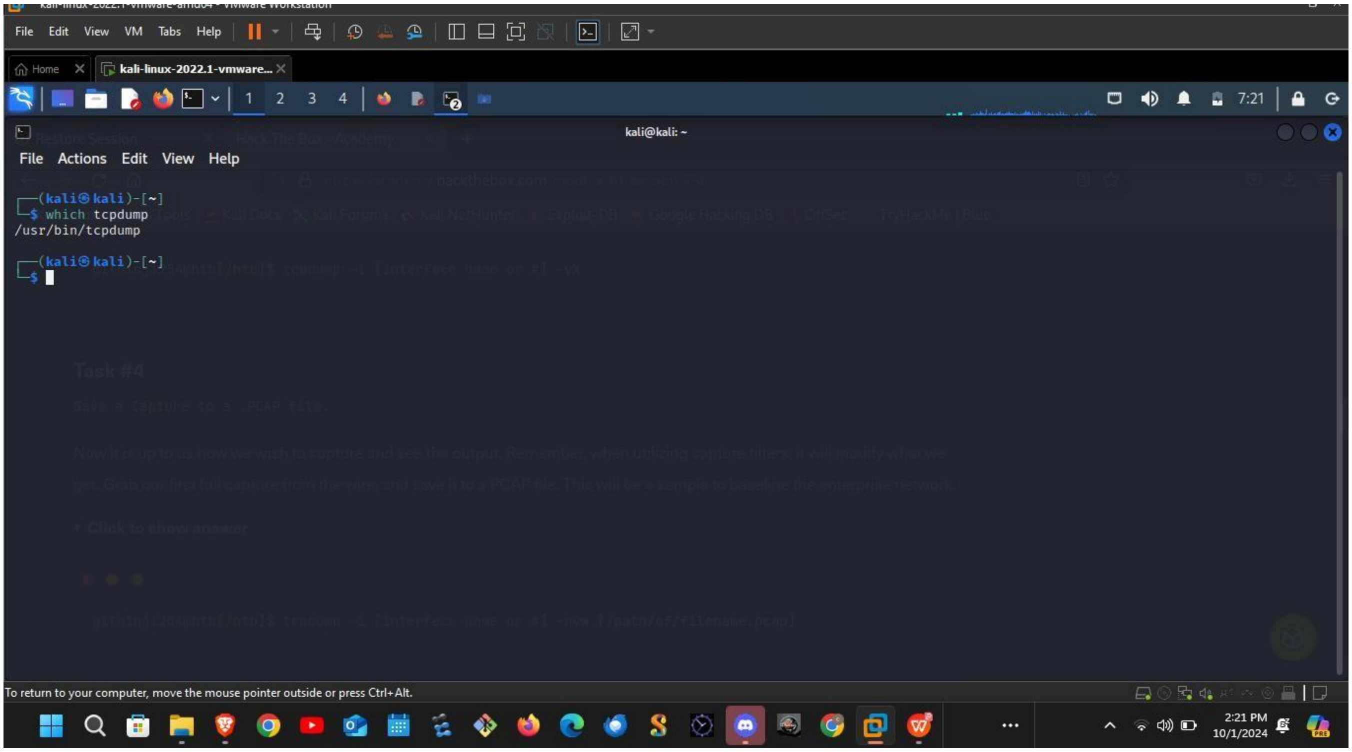
1. What TCPDump switch will allow us to pipe the contents of a pcap file out to another function such as 'grep'? –l

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

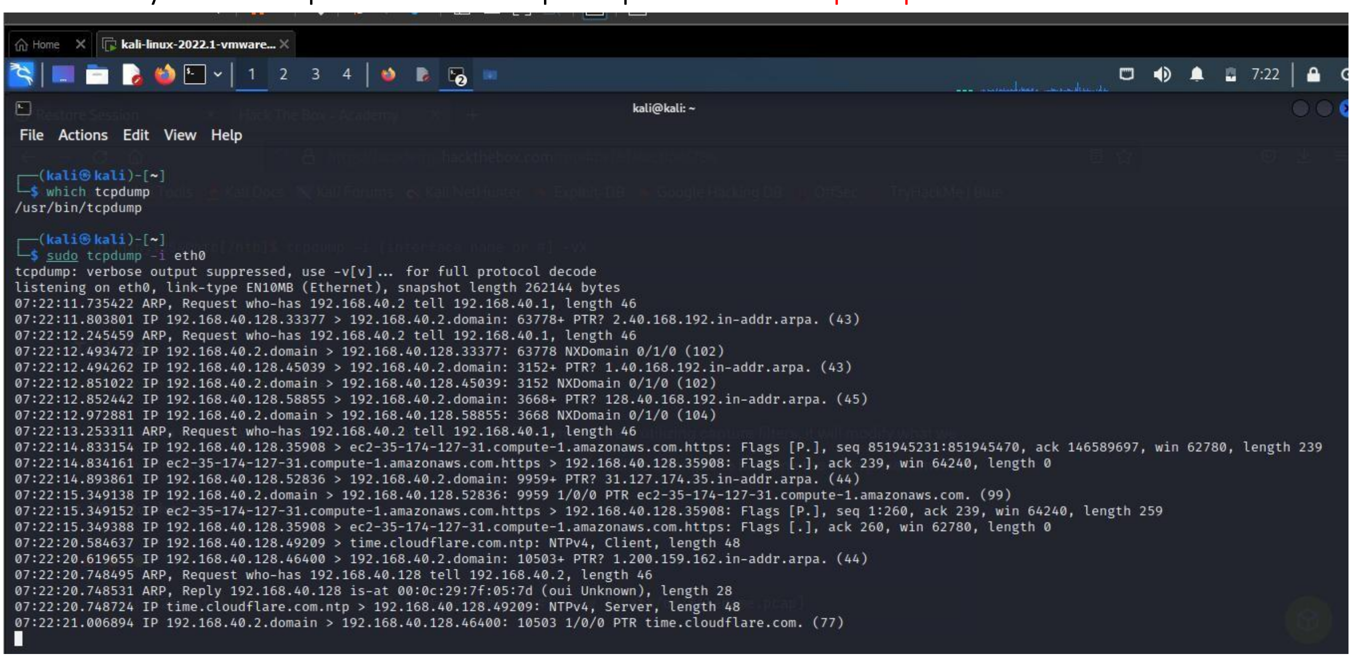
3. If we wished to filter out ICMP traffic from our capture, what filter could we use? (word only, not symbol please.) not ICMP



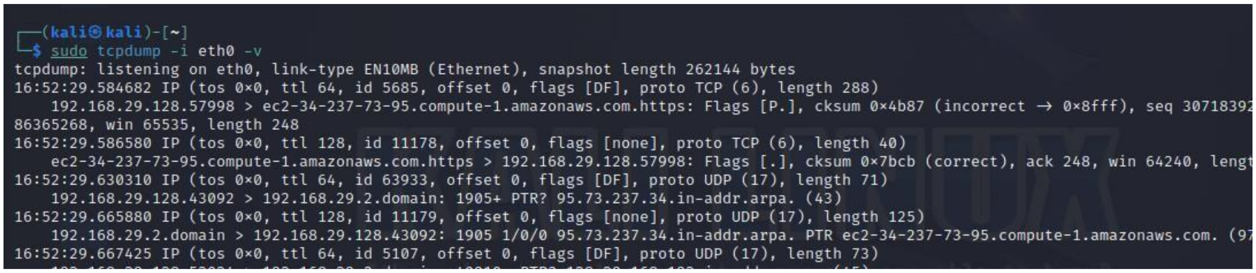
4. What command will show you where / if TCPDump is installed? which tcpdump



5. How do you start a capture with TCPDump to capture on eth0? Tcpdump –i eth0

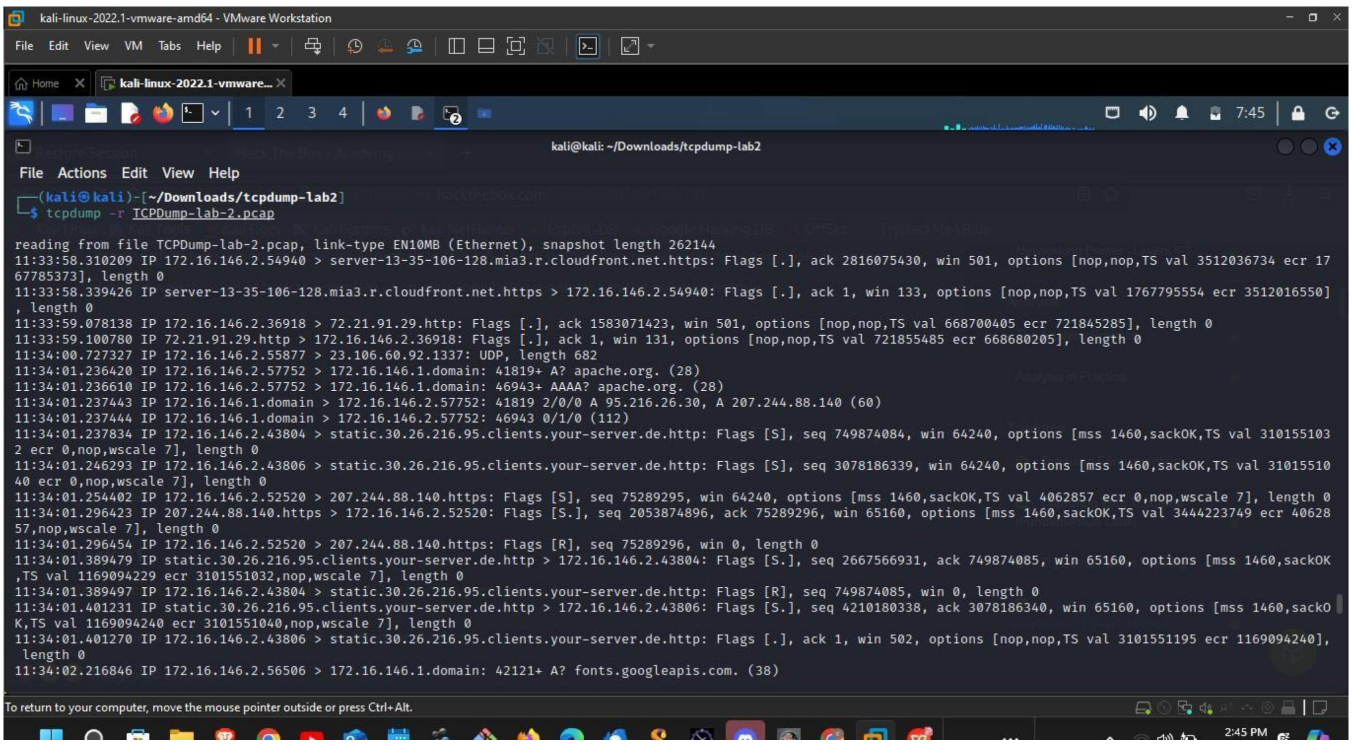


6. What switch will provide more verbosity in your output? –v



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

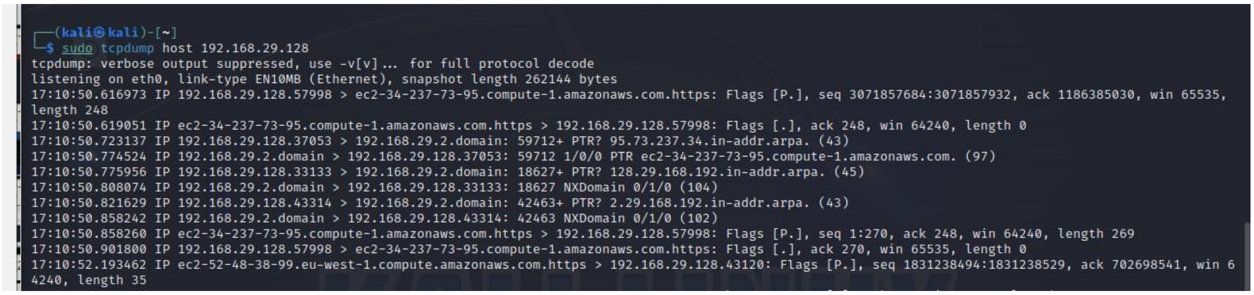
8. What switch will read a capture from a .pcap file? –r



9. What switch will show the contents of a capture in Hex and ASCII? –X

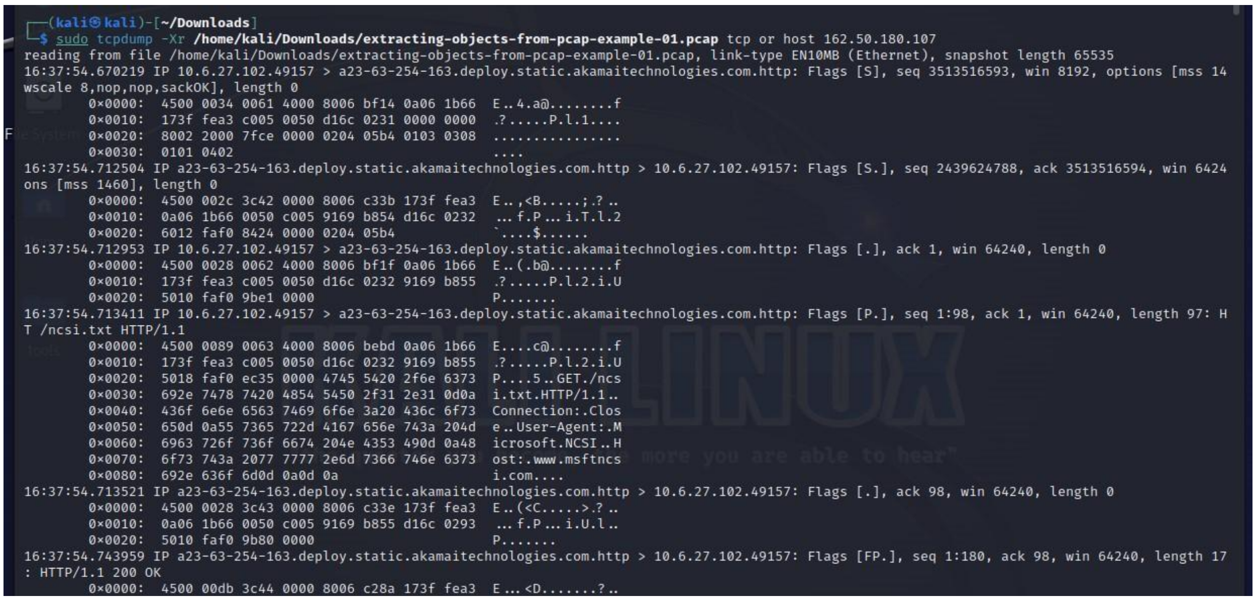
Tcpdump Packet Filtering

1. 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



```
┌──(kali㉿kali)-[~]
└─$ sudo tcpdump host 192.168.29.128
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
17:10:50.616973 IP 192.168.29.128.57998 > ec2-34-237-73-95.compute-1.amazonaws.com.https: Flags [P.], seq 3071857684:3071857932, ack 1186385030, win 65535,
length 248
17:10:50.619051 IP ec2-34-237-73-95.compute-1.amazonaws.com.https > 192.168.29.128.57998: Flags [.], ack 248, win 64240, length 0
17:10:50.723137 IP 192.168.29.128.37053 > 192.168.29.2.domain: 59712+ PTR? 95.73.237.34.in-addr.arpa. (43)
17:10:50.774524 IP 192.168.29.2.domain > 192.168.29.128.37053: 59712 1/0/0 PTR ec2-34-237-73-95.compute-1.amazonaws.com. (97)
17:10:50.775956 IP 192.168.29.128.33133 > 192.168.29.2.domain: 18627+ PTR? 128.29.168.192.in-addr.arpa. (45)
17:10:50.808074 IP 192.168.29.2.domain > 192.168.29.128.33133: 18627 NXDomain 0/1/0 (104)
17:10:50.821629 IP 192.168.29.128.43314 > 192.168.29.2.domain: 42463+ PTR? 2.29.168.192.in-addr.arpa. (43)
17:10:50.858242 IP 192.168.29.2.domain > 192.168.29.128.43314: 42463 NXDomain 0/1/0 (102)
17:10:50.858260 IP ec2-34-237-73-95.compute-1.amazonaws.com.https > 192.168.29.128.57998: Flags [P.], seq 1:270, ack 248, win 64240, length 269
17:10:50.901800 IP 192.168.29.128.57998 > ec2-34-237-73-95.compute-1.amazonaws.com.https: Flags [.], ack 270, win 65535, length 0
17:10:52.193462 IP ec2-52-48-38-99.eu-west-1.compute.amazonaws.com.https > 192.168.29.128.43120: Flags [P.], seq 1831238494:1831238529, ack 702698541, win 6
4240, length 35
```

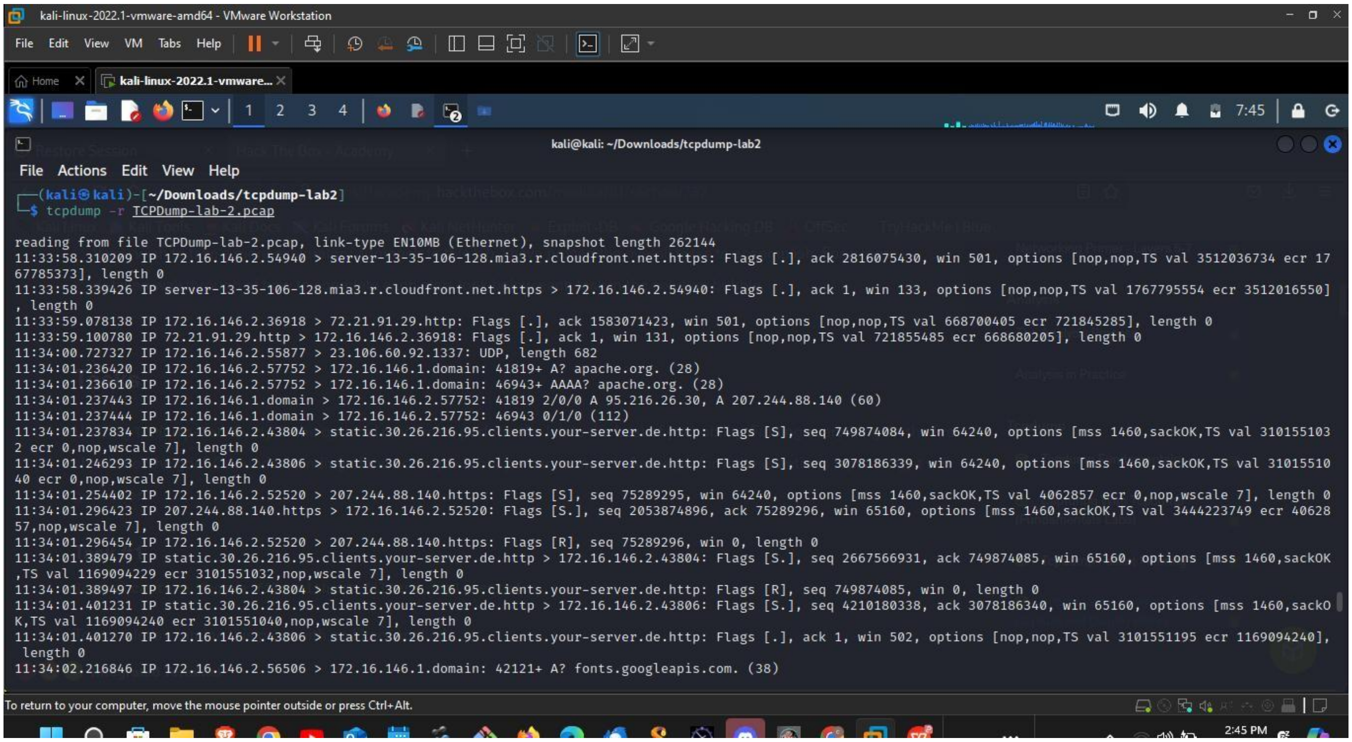
2. What filter will allow me to capture based on either of two options? Or

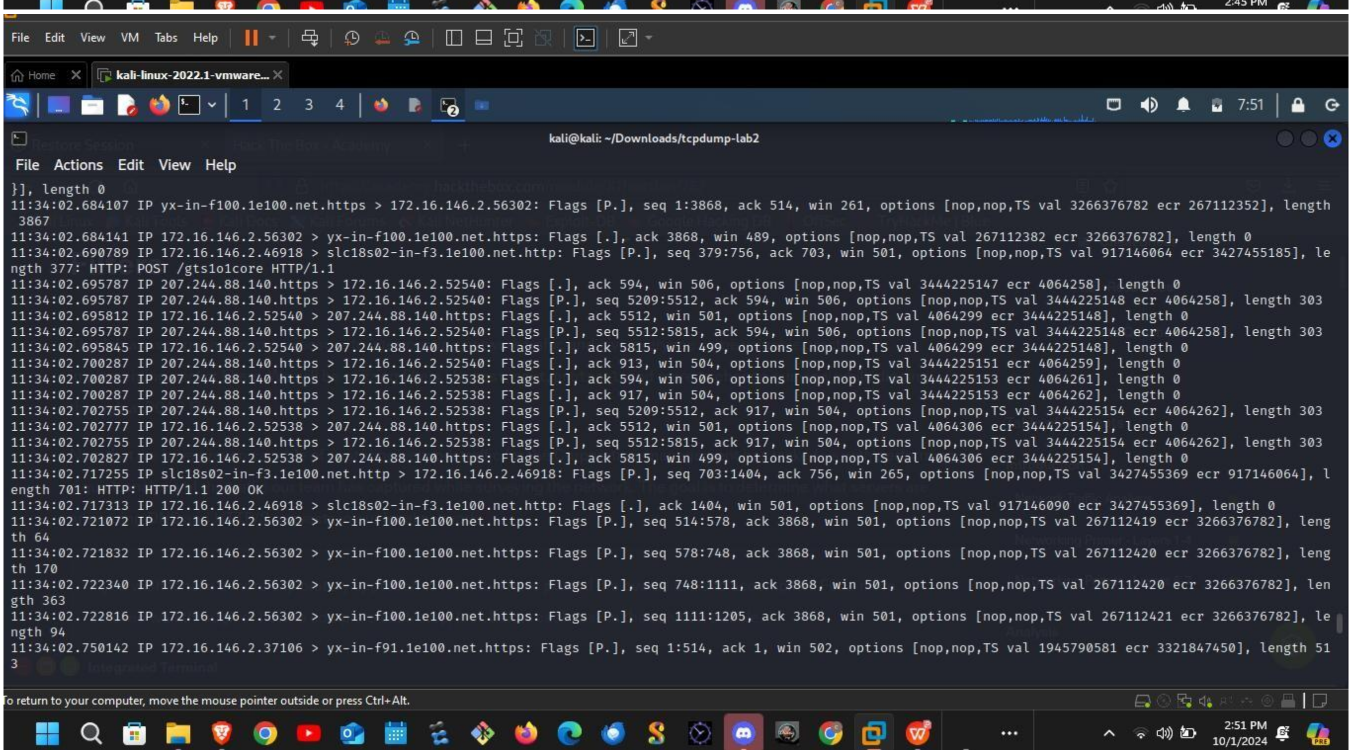


```
┌──(kali㉿kali)-[~/Downloads]
└─$ sudo tcpdump -Xr /home/kali/Downloads/extracting-objects-from-pcap-example-01.pcap tcp or host 162.50.180.107
reading from file /home/kali/Downloads/extracting-objects-from-pcap-example-01.pcap, link-type EN10MB (Ethernet), snapshot length 65535
16:37:54.670219 IP 10.6.27.102.49157 > a23-63-254-163.deploy.static.akamaitechnologies.com.http: Flags [S], seq 3513516593, win 8192, options [mss 14
wscale 8,nop,nop,sackOK], length 0
        0x0000:  4500 0034 0061 4000 8006 bf14 0a06 1b66  E..4.a@........f
        0x0010:  173f fea3 c005 0050 d16c 0231 0000 0000  .?.....P.l.1....
        0x0020:  8002 2000 7fce 0000 0204 05b4 0103 0308  ................
        0x0030:  0101 0402                                ....
16:37:54.712504 IP a23-63-254-163.deploy.static.akamaitechnologies.com.http > 10.6.27.102.49157: Flags [S.], seq 2439624788, ack 3513516594, win 6424
ons [mss 1460], length 0
        0x0000:  4500 002c 3c42 0000 8006 c33b 173f fea3  E..,<B.....;.?..
        0x0010:  0a06 1b66 0050 c005 9169 b854 d16c 0232  ...f.P...i.T.l.2
        0x0020:  6012 faf0 8424 0000 0204 05b4            `....$......
16:37:54.712953 IP 10.6.27.102.49157 > a23-63-254-163.deploy.static.akamaitechnologies.com.http: Flags [.], ack 1, win 64240, length 0
        0x0000:  4500 0028 0062 4000 8006 bf1f 0a06 1b66  E..(.b@........f
        0x0010:  173f fea3 c005 0050 d16c 0232 9169 b855  .?.....P.l.2.i.U
        0x0020:  5010 faf0 9be1 0000                      P.......
16:37:54.713411 IP 10.6.27.102.49157 > a23-63-254-163.deploy.static.akamaitechnologies.com.http: Flags [P.], seq 1:98, ack 1, win 64240, length 97: H
T /ncsi.txt HTTP/1.1
        0x0000:  4500 0089 0063 4000 8006 bebd 0a06 1b66  E....c@........f
        0x0010:  173f fea3 c005 0050 d16c 0232 9169 b855  .?.....P.l.2.i.U
        0x0020:  5018 faf0 ec35 0000 4745 5420 2f6e 6373  P....5..GET./ncs
        0x0030:  692e 7478 7420 4854 5450 2f31 2e31 0d0a  i.txt.HTTP/1.1..
        0x0040:  436f 6e6e 6563 7469 6f6e 3a20 436c 6f73  Connection:.Clos
        0x0050:  650d 0a55 7365 722d 4167 656e 743a 204d  e..User-Agent:.M
        0x0060:  6963 726f 736f 6674 204e 4353 490d 0a48  icrosoft.NCSI..H
        0x0070:  6f73 743a 2077 7777 2e6d 7366 746e 6373  ost:.www.msftncs
        0x0080:  692e 636f 6d0d 0a0d 0a                   i.com....
16:37:54.713521 IP a23-63-254-163.deploy.static.akamaitechnologies.com.http > 10.6.27.102.49157: Flags [.], ack 98, win 64240, length 0
        0x0000:  4500 0028 3c43 0000 8006 c33e 173f fea3  E..(<C.....>.?..
        0x0010:  0a06 1b66 0050 c005 9169 b855 d16c 0293  ...f.P...i.U.l..
        0x0020:  5010 faf0 9b80 0000                      P.......
16:37:54.743959 IP a23-63-254-163.deploy.static.akamaitechnologies.com.http > 10.6.27.102.49157: Flags [FP.], seq 1:180, ack 98, win 64240, length 17
: HTTP/1.1 200 OK
        0x0000:  4500 00db 3c44 0000 8006 c28a 173f fea3  E...<D.......?..
```

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

Interrogating Network Traffic With Capture and Display Filters





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)

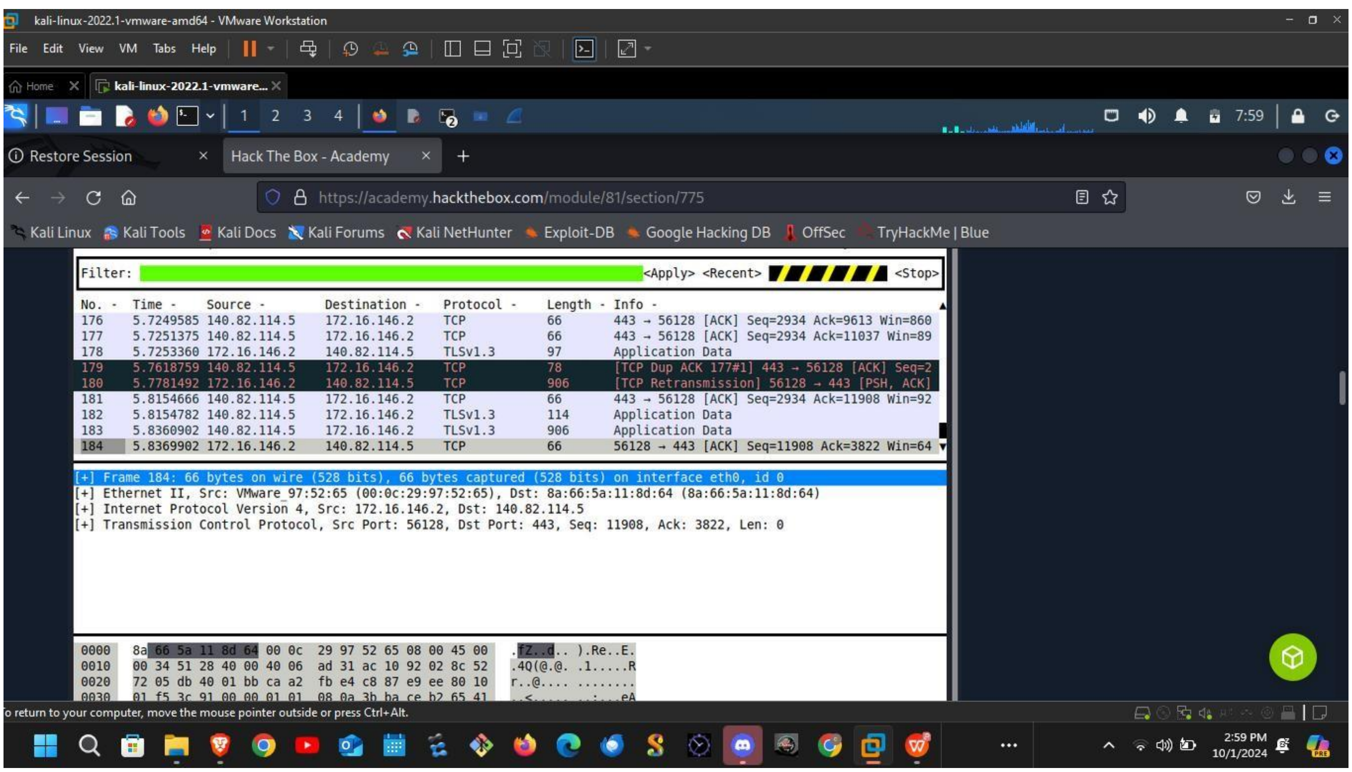
172.16.146.1

PART 3: Analysis with Wireshark

1. True or False: Wireshark can run on both Windows and Linux. True

> `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. Wireshark allows the user to dive much deeper into the inspection of network packets than other tools. What makes Wireshark truly powerful is the analysis capability it provides, giving a detailed insight into the traffic.

2. Which Pane allows a user to see a summary of each packet grabbed during the capture? Packet List



3. Which pane provides your insight into the traffic you captured and displays it in both ASCII and Hex? Packet Bytes

```
0000  ff ff ff ff ff ff 78 2b  cb 6a a8 a1 08 06 00 01   ······x+ ·j······
0010  08 00 06 04 00 01 78 2b  cb 6a a8 a1 c0 a8 01 01   ······x+ ·j······
0020  00 00 00 00 00 00 c0 a8  01 fe 00 00 00 00 00 00   ········ ········
0030  00 00 00 00 00 00 00 00  00 00 00 00               ········ ····
```

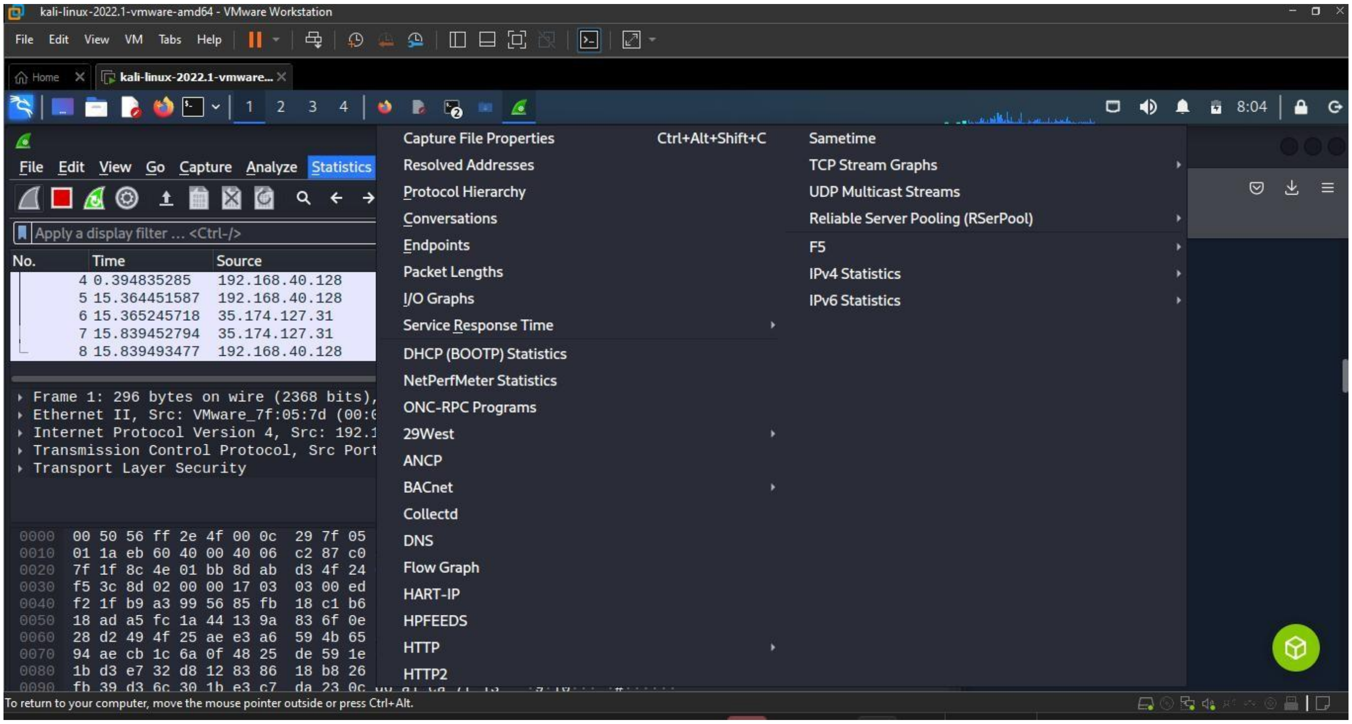
4. What switch is used with TShark to list possible interfaces to capture on? –D

5. What switch allows us to apply filters in TShark? –f

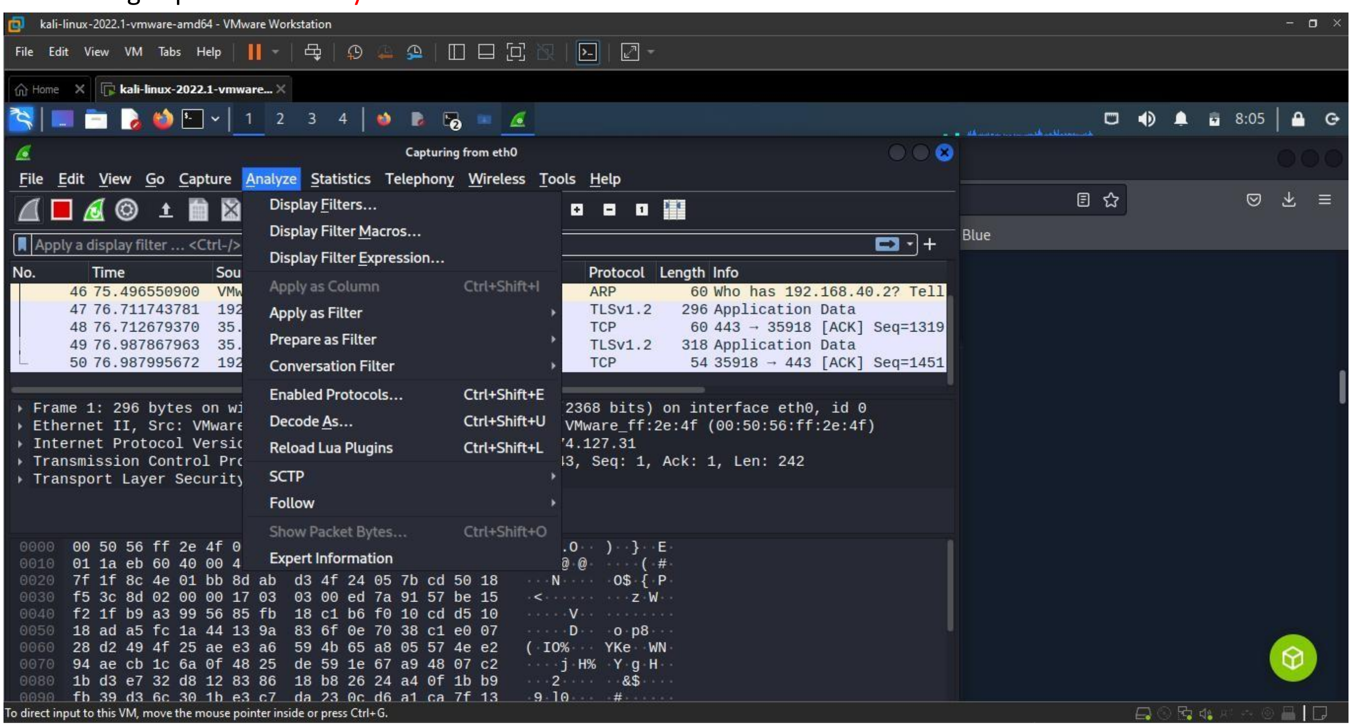
6. Is a capture filter applied before the capture starts or after? (answer before or after) before

Wireshark Advanced Usage

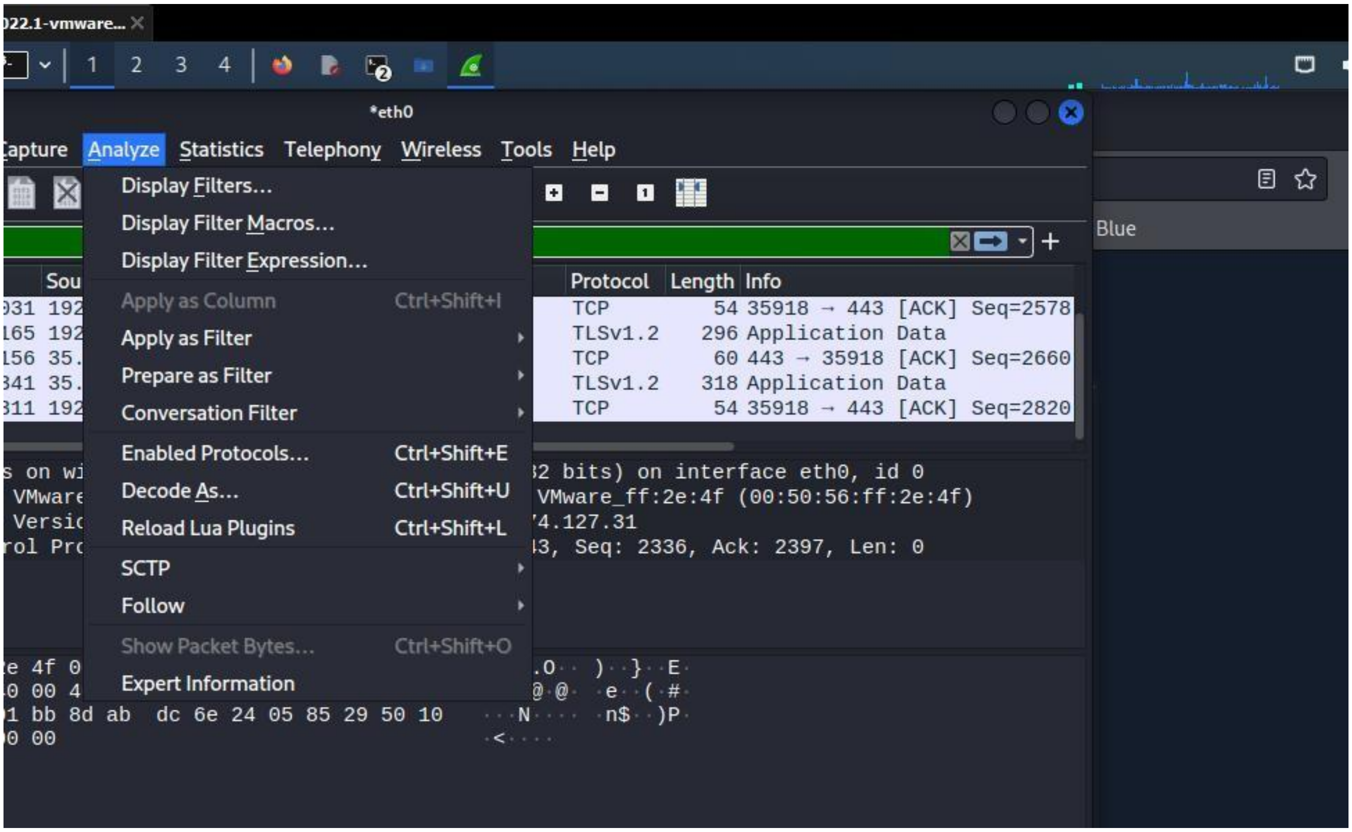
1. Which plugin tab can provide us with a way to view conversation metadata and even protocol breakdowns for the entire PCAP file? Statistics



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

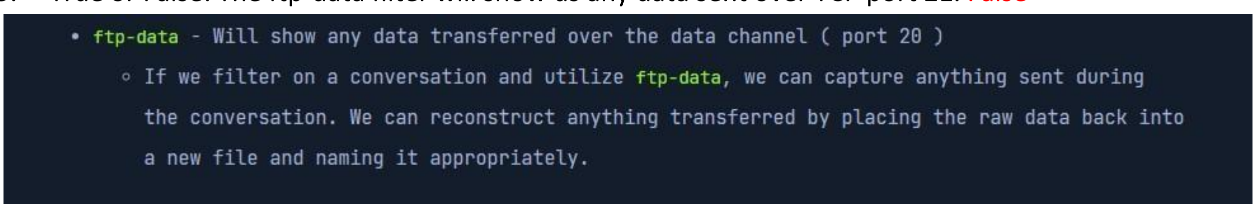


3. What stream oriented Transport protocol enables us to follow and rebuild conversations and the included data? TCP



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

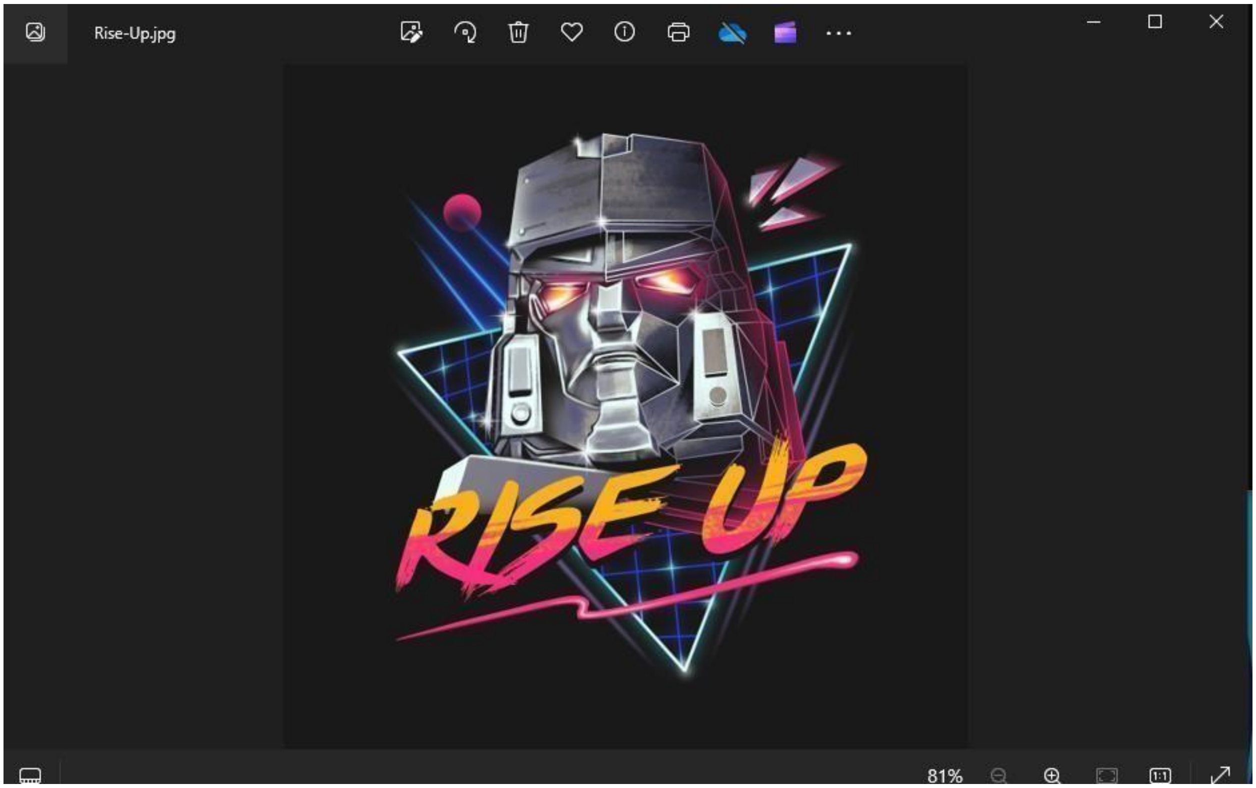
5. True or False: The ftp-data filter will show us any data sent over TCP port 21. False



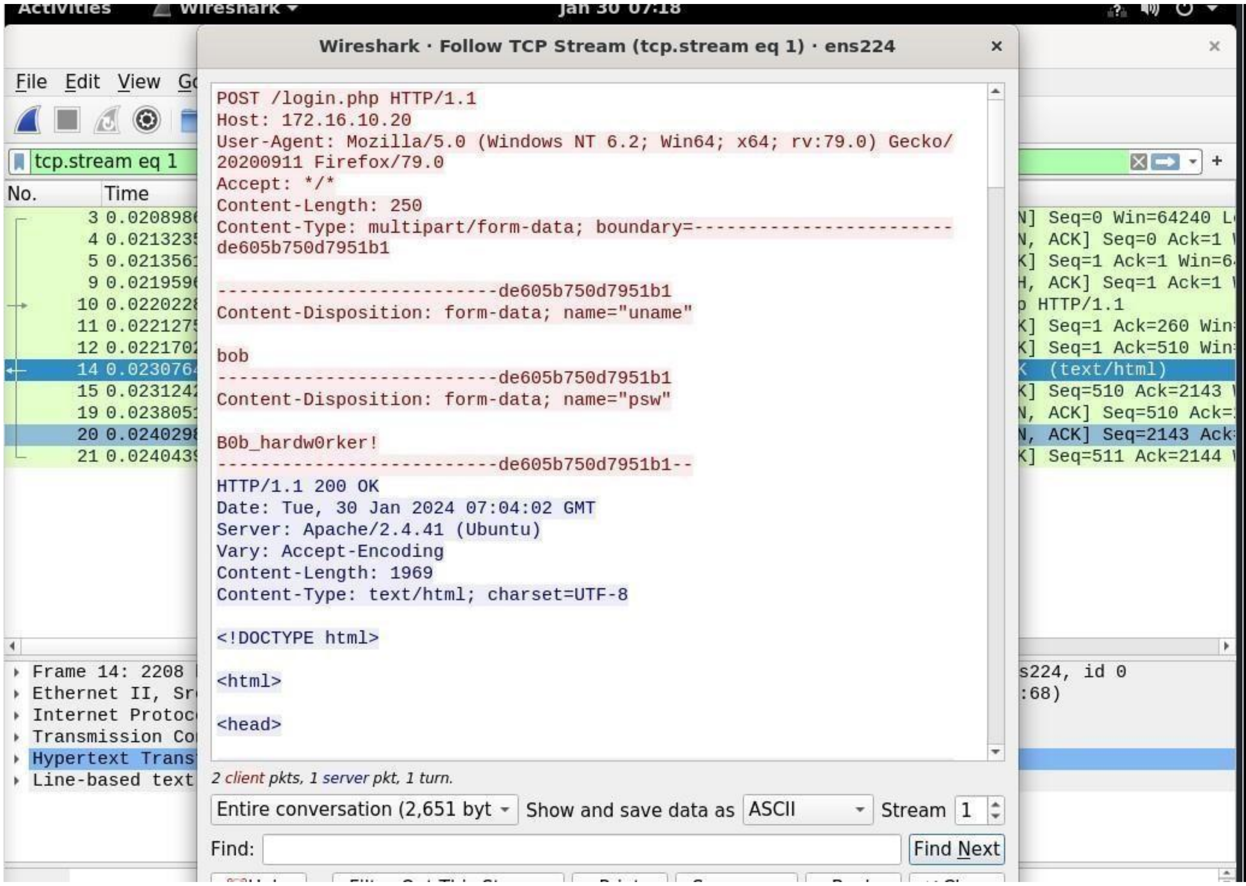
- ftp-data - Will show any data transferred over the data channel (port 20)
  - If we filter on a conversation and utilize ftp-data, we can capture anything sent during the conversation. We can reconstruct anything transferred by placing the raw data back into a new file and naming it appropriately.

Packet Inception, Dissecting Network Traffic with Wireshark

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



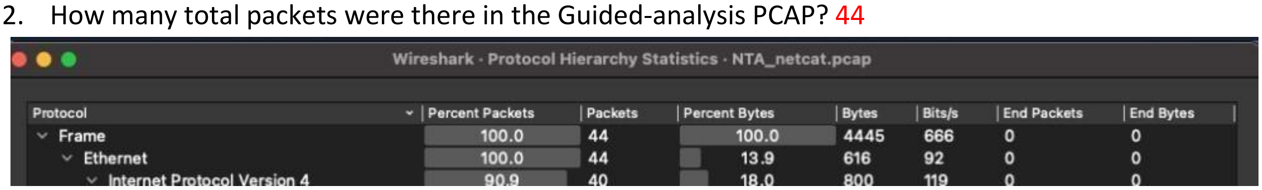
2. Which employee is suspected of performing potentially malicious actions in the live environment?



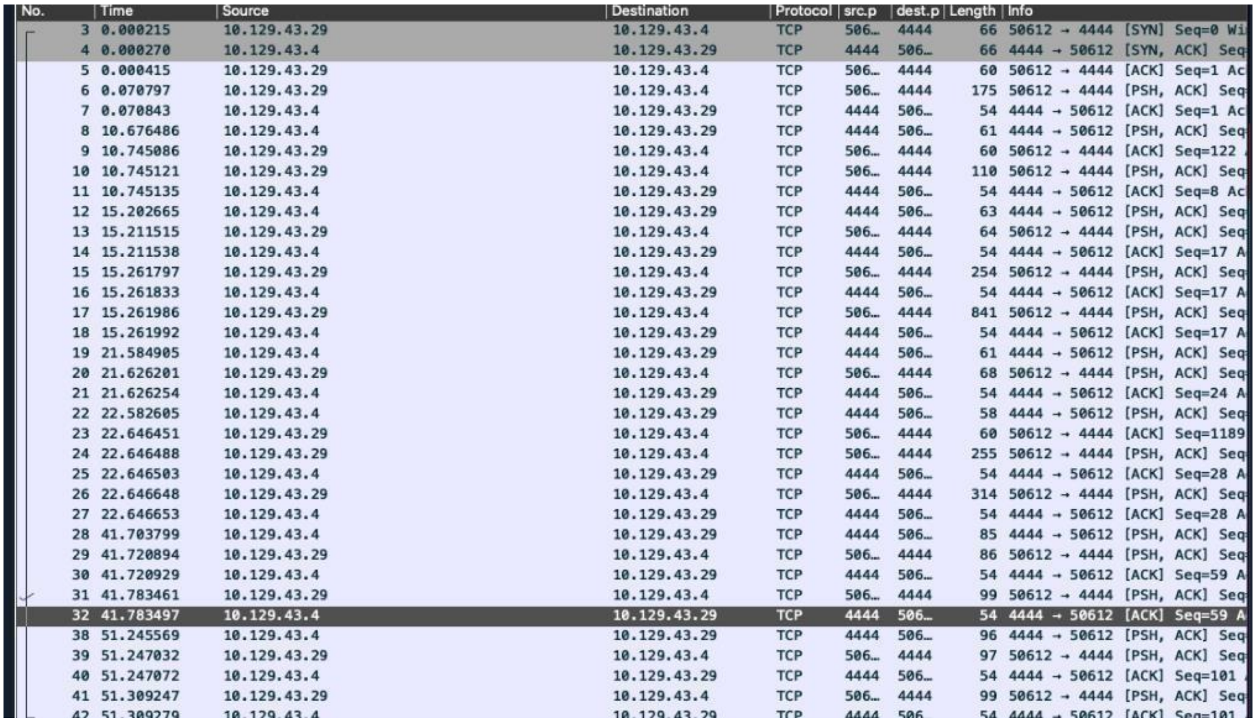
Guided Lab: Traffic Analysis Workflow

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

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



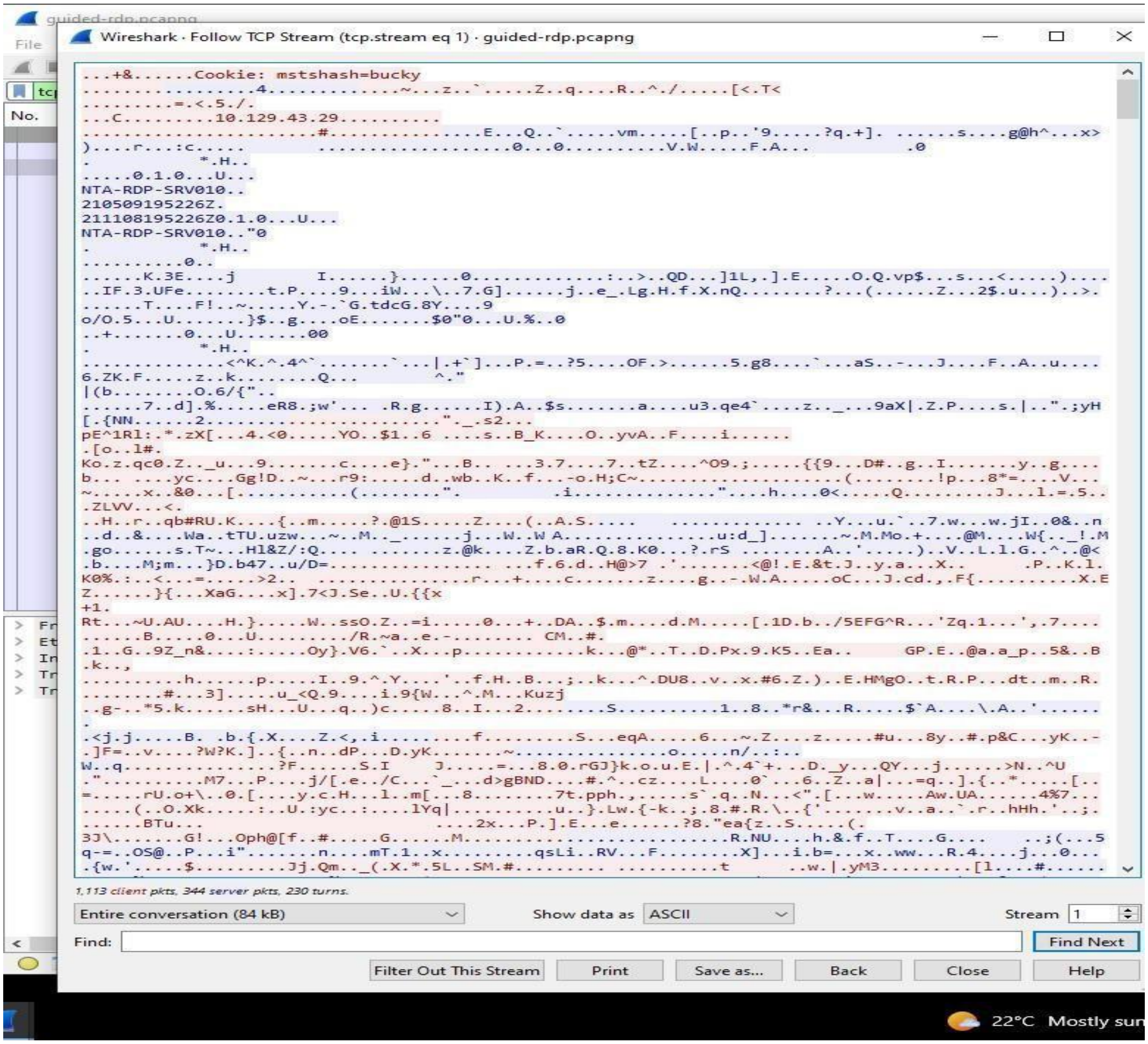
3. What was the suspicious port that was being used? 4444



No.	Time	Source	Destination	Protocol	src.p	dest.p	Length	Info
3	0.000215	10.129.43.29	10.129.43.4	TCP	506…	4444	66	50612 → 4444 [SYN] Seq=0 Wi
4	0.000270	10.129.43.4	10.129.43.29	TCP	4444	506…	66	4444 → 50612 [SYN, ACK] Seq
5	0.000415	10.129.43.29	10.129.43.4	TCP	506…	4444	60	50612 → 4444 [ACK] Seq=1 Ac
6	0.070797	10.129.43.29	10.129.43.4	TCP	506…	4444	175	50612 → 4444 [PSH, ACK] Seq
7	0.070843	10.129.43.4	10.129.43.29	TCP	4444	506…	54	4444 → 50612 [ACK] Seq=1 Ac
8	10.676486	10.129.43.4	10.129.43.29	TCP	4444	506…	61	4444 → 50612 [PSH, ACK] Seq
9	10.745086	10.129.43.29	10.129.43.4	TCP	506…	4444	60	50612 → 4444 [ACK] Seq=122
10	10.745121	10.129.43.29	10.129.43.4	TCP	506…	4444	110	50612 → 4444 [PSH, ACK] Seq
11	10.745135	10.129.43.4	10.129.43.29	TCP	4444	506…	54	4444 → 50612 [ACK] Seq=8 Ac
12	15.202665	10.129.43.4	10.129.43.29	TCP	4444	506…	63	4444 → 50612 [PSH, ACK] Seq
13	15.211515	10.129.43.29	10.129.43.4	TCP	506…	4444	64	50612 → 4444 [PSH, ACK] Seq
14	15.211538	10.129.43.4	10.129.43.29	TCP	4444	506…	54	4444 → 50612 [ACK] Seq=17 A
15	15.261797	10.129.43.29	10.129.43.4	TCP	506…	4444	254	50612 → 4444 [PSH, ACK] Seq
16	15.261833	10.129.43.4	10.129.43.29	TCP	4444	506…	54	4444 → 50612 [ACK] Seq=17 A
17	15.261986	10.129.43.29	10.129.43.4	TCP	506…	4444	841	50612 → 4444 [PSH, ACK] Seq
18	15.261992	10.129.43.4	10.129.43.29	TCP	4444	506…	54	4444 → 50612 [ACK] Seq=17 A
19	21.584905	10.129.43.4	10.129.43.29	TCP	4444	506…	61	4444 → 50612 [PSH, ACK] Seq
20	21.626201	10.129.43.29	10.129.43.4	TCP	506…	4444	68	50612 → 4444 [PSH, ACK] Seq
21	21.626254	10.129.43.4	10.129.43.29	TCP	4444	506…	54	4444 → 50612 [ACK] Seq=24 A
22	22.582605	10.129.43.4	10.129.43.29	TCP	4444	506…	58	4444 → 50612 [PSH, ACK] Seq
23	22.646451	10.129.43.29	10.129.43.4	TCP	506…	4444	60	50612 → 4444 [ACK] Seq=1189
24	22.646488	10.129.43.29	10.129.43.4	TCP	506…	4444	255	50612 → 4444 [PSH, ACK] Seq
25	22.646503	10.129.43.4	10.129.43.29	TCP	4444	506…	54	4444 → 50612 [ACK] Seq=28 A
26	22.646648	10.129.43.29	10.129.43.4	TCP	506…	4444	314	50612 → 4444 [PSH, ACK] Seq
27	22.646653	10.129.43.4	10.129.43.29	TCP	4444	506…	54	4444 → 50612 [ACK] Seq=28 A
28	41.703799	10.129.43.4	10.129.43.29	TCP	4444	506…	85	4444 → 50612 [PSH, ACK] Seq
29	41.720894	10.129.43.29	10.129.43.4	TCP	506…	4444	86	50612 → 4444 [PSH, ACK] Seq
30	41.720929	10.129.43.4	10.129.43.29	TCP	4444	506…	54	4444 → 50612 [ACK] Seq=59 A
31	41.783461	10.129.43.29	10.129.43.4	TCP	506…	4444	99	50612 → 4444 [PSH, ACK] Seq
32	41.783497	10.129.43.4	10.129.43.29	TCP	4444	506…	54	4444 → 50612 [ACK] Seq=59 A
38	51.245569	10.129.43.4	10.129.43.29	TCP	4444	506…	96	4444 → 50612 [PSH, ACK] Seq
39	51.247032	10.129.43.29	10.129.43.4	TCP	506…	4444	97	50612 → 4444 [PSH, ACK] Seq
40	51.247072	10.129.43.4	10.129.43.29	TCP	4444	506…	54	4444 → 50612 [ACK] Seq=101
41	51.309247	10.129.43.29	10.129.43.4	TCP	506…	4444	99	50612 → 4444 [PSH, ACK] Seq
42	51.309279	10.129.43.4	10.129.43.29	TCP	4444	506…	54	4444 → 50612 [ACK] Seq=101

Decrypting RDP connections

1. What user account was used to initiate the RDP connection? bucky



https://academy.hackthebox.com/achievement/868719/81

CONCLUSION

In this lab, I learned basics of network traffic analysis, how to use tcpdump and Wireshark for implementing network traffic analysis.

As most command I had never interacted with they really brought to me a challenge. However, with the help of online research and study, I was able to overcome this.