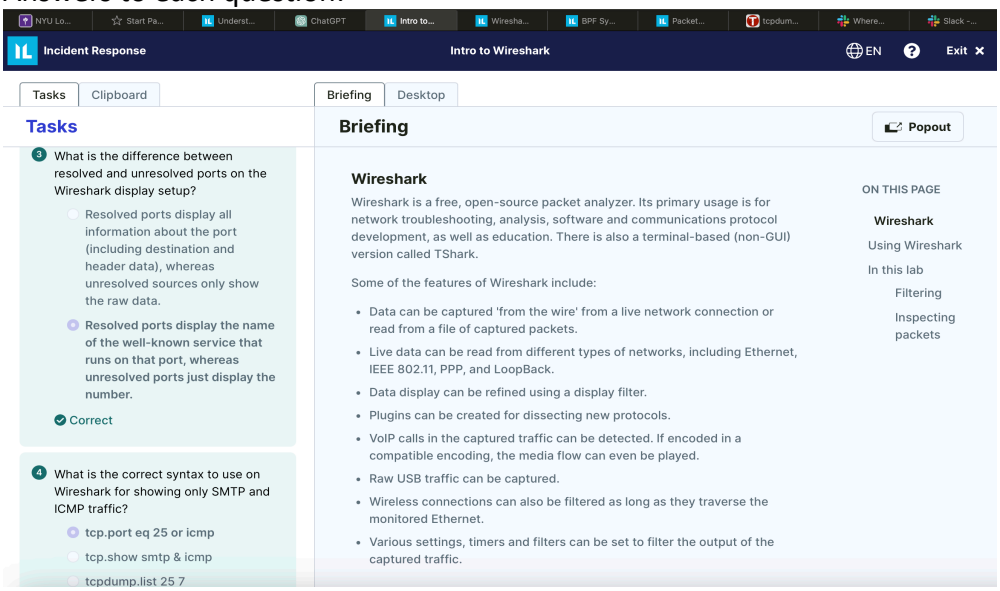
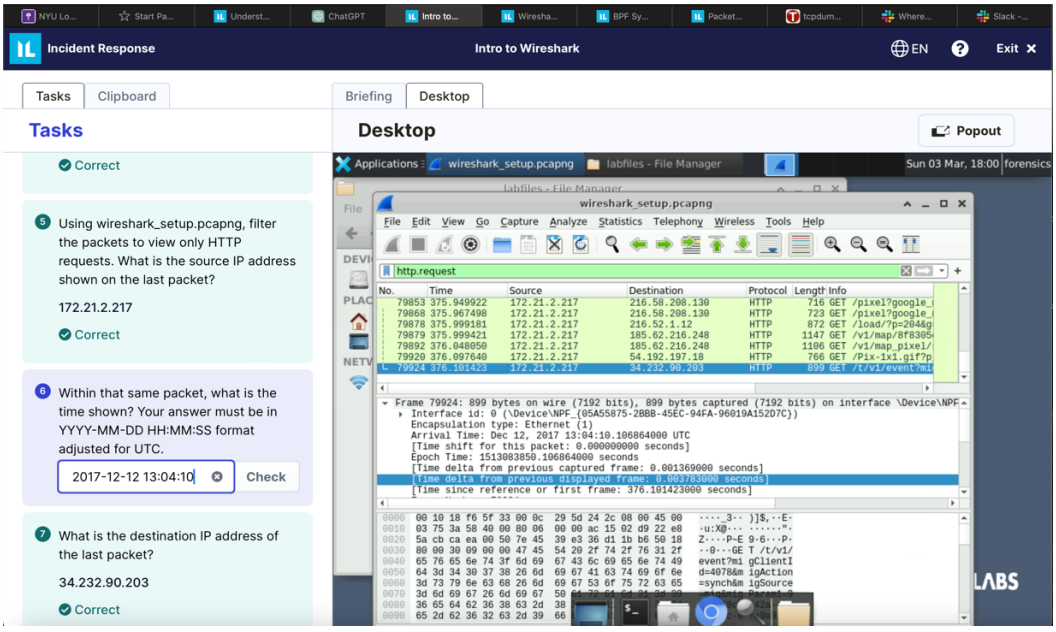
Network Security
Lab-2
Network Capture, Analysis, and Scanning

-Hari Kishan Reddy

1) Intro to Wireshark

Answers to each question:





Writeup and approach I followed to complete the lab:

Key Learnings:
1. Packet Capture*:* Wireshark can capture packets from live network connections or read from a file of captured packets. This allows us to analyze network traffic in real-time or analyze pre-captured data.

2. Display Filters*:* Display filters in Wireshark help narrow down the packets displayed based on specific criteria. For example, using `http` as a display filter allows us to view only HTTP traffic, making it easier to analyze.

3. Protocol Analysis: Wireshark can dissect and analyze various network protocols, such as TCP, UDP, HTTP, SMTP, and ICMP. This allows us to understand the communication patterns between devices on the network.

4. Port Resolution*:* Wireshark can resolve ports to display the name of the well-known service that runs on that port. This helps in understanding the purpose of each packet and the services being used.

How I Solved the Questions:
1. Familiarization with Wireshark: I read the briefing section to understand the basic features and uses of Wireshark.

2. Opening the pcap file: I opened the pcap file provided in the labfiles directory using Wireshark.

3. Difference between resolved and unresolved ports: I learned that resolved ports display the name of the service running on that port, while unresolved ports only display the port number.

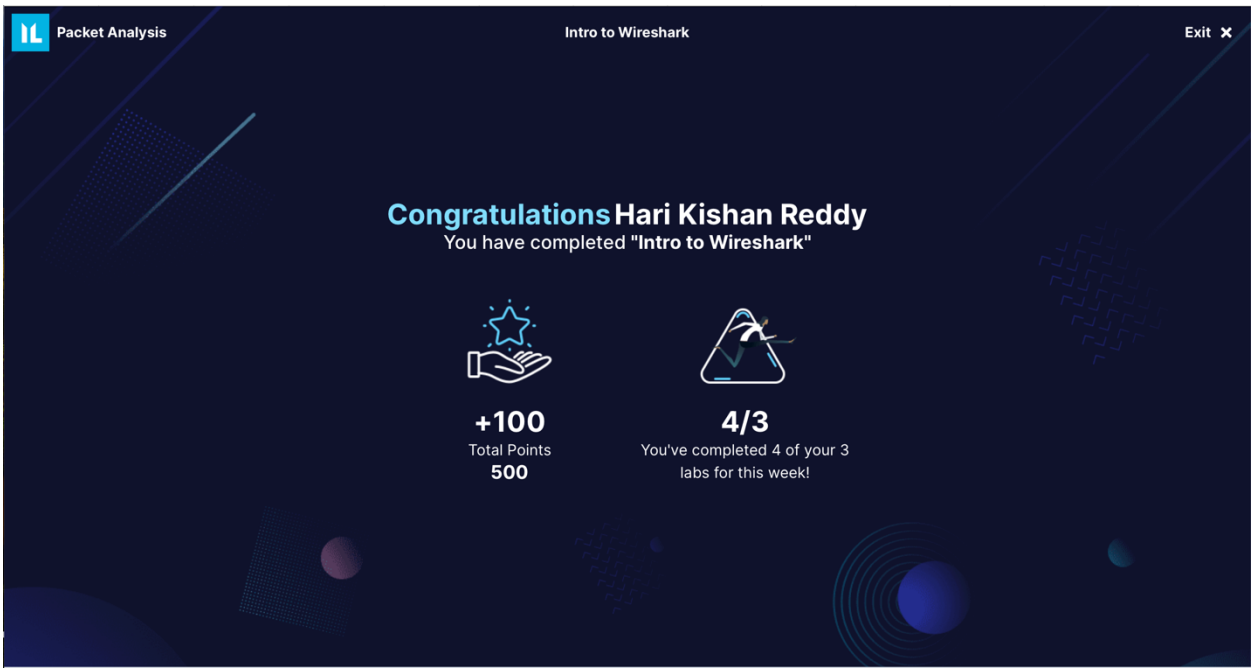
4. Syntax for showing SMTP and ICMP traffic: I used the correct syntax `tcp.port eq 25 or icmp` to display only SMTP and ICMP traffic.

5. Filtering packets for HTTP requests: I filtered the packets using the `http` filter to view only HTTP requests. I then inspected the last packet to find the source IP address.

6. Finding the time and destination IP address: I inspected the last HTTP request packet to find the time and destination IP address.

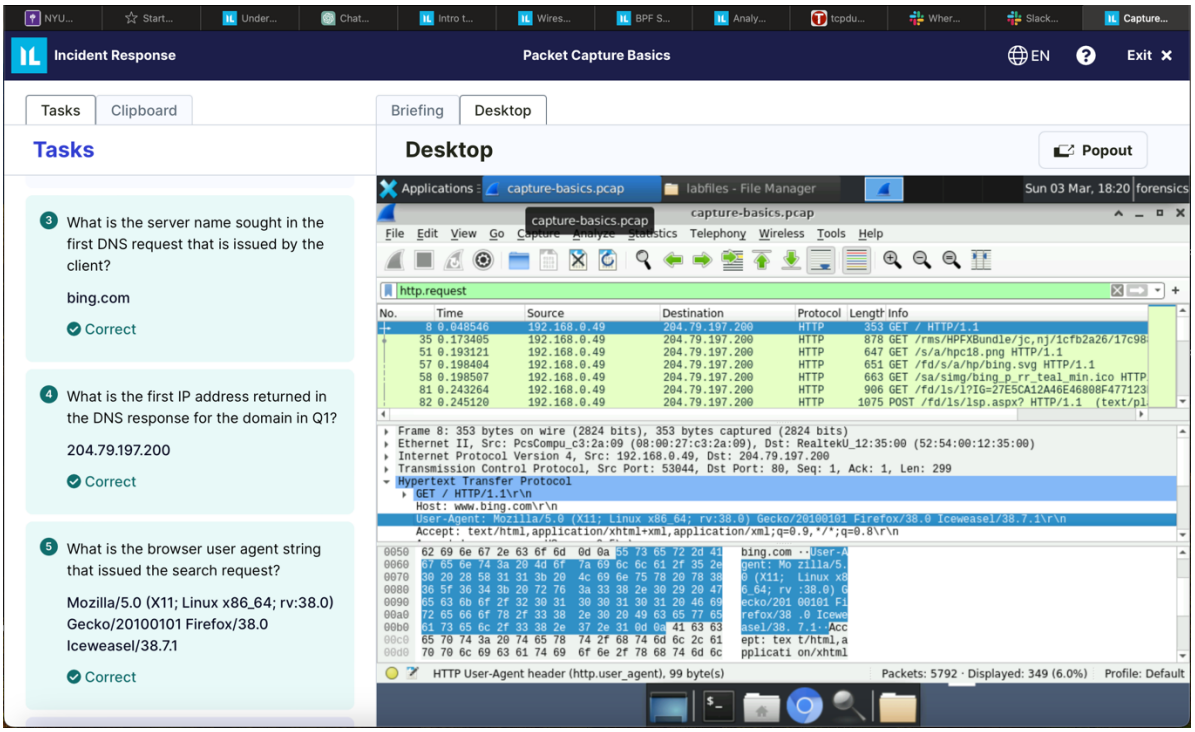
Conclusion:
Overall, this lab helped me understand the basics of network traffic capture and analysis using Wireshark. I learned how to capture and analyze network packets, use display filters, and identify specific types of traffic.

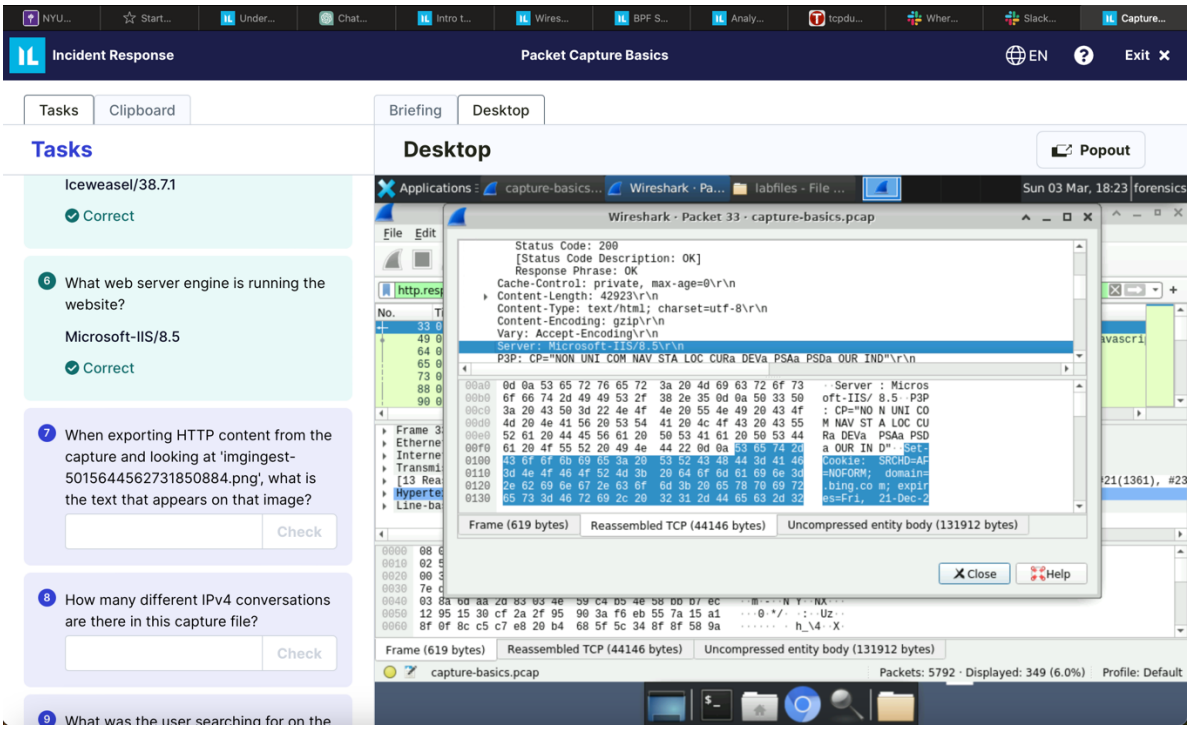
Screenshot of Lab Completion:

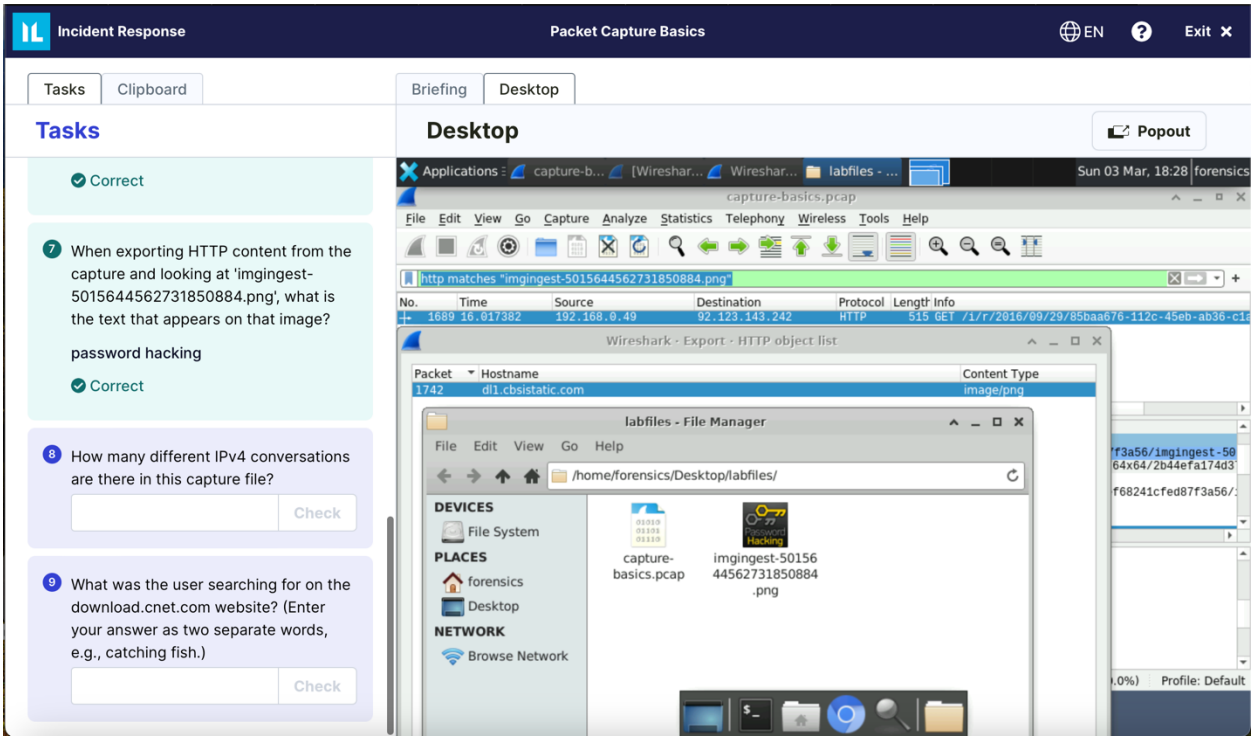


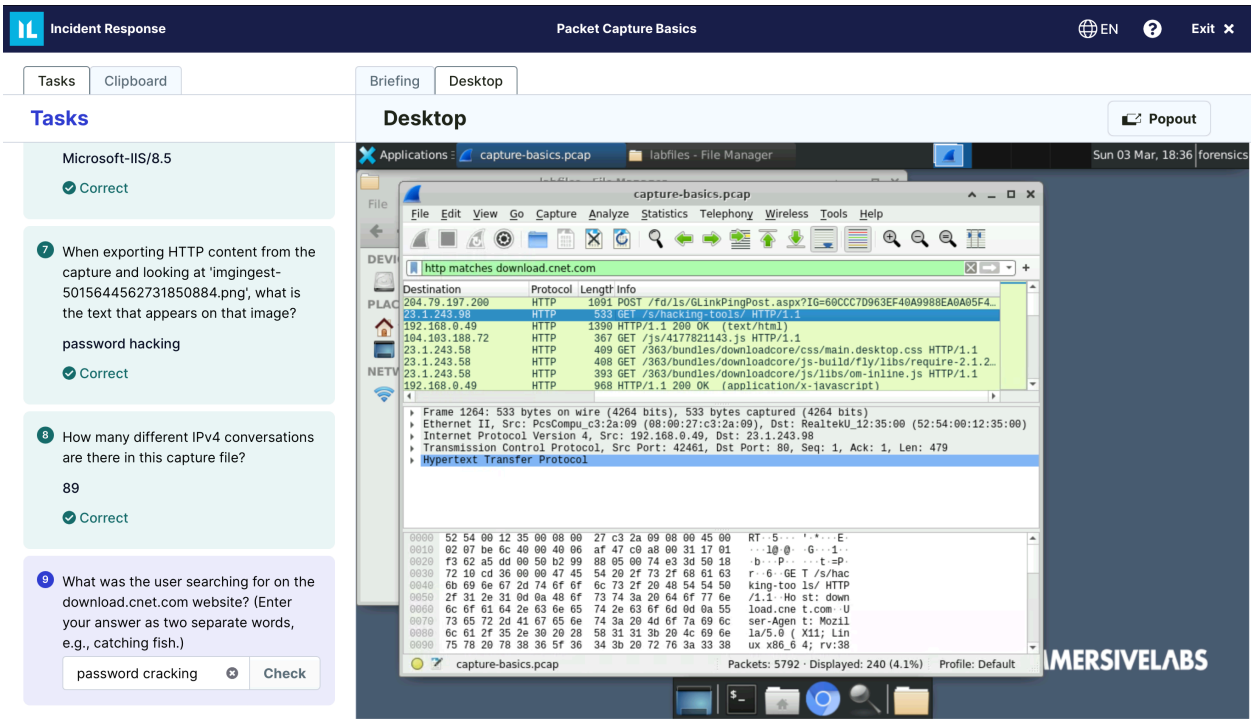
2) Packet Capture Basics

Answers to each question:






Incident Response
Packet Capture Basics
EN
Exit
Tasks
Clipboard
Briefing
Desktop
Tasks
Desktop
Popout
Correct
7 When exporting HTTP content from the capture and looking at 'imgingest-5015644562731850884.png', what is the text that appears on that image?
password hacking
Correct
8 How many different IPv4 conversations are there in this capture file?
Check
9 What was the user searching for on the download.cnet.com website? (Enter your answer as two separate words, e.g., catching fish.)
Check
Applications
capture-b...
[Wireshar...
Wireshar...
labfiles - ...
Sun 03 Mar, 18:28
forensics
capture-basics.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http matches "imgingest-5015644562731850884.png"
No. Time Source Destination Protocol Length Info
1689 16.017382 192.168.0.49 92.123.143.242 HTTP 515 GET /i/r/2016/09/29/85baa676-112c-45eb-ab36-c1
Wireshark · Export · HTTP object list
Packet Hostname Content Type
1742 dl1.cbsistatic.com image/png
labfiles - File Manager
File Edit View Go Help
/home/forensics/Desktop/labfiles/
DEVICES
File System
PLACES
forensics
Desktop
NETWORK
Browse Network
capture-basics.pcap
imgingest-5015644562731850884.png
Profile: Default


Incident Response
Packet Capture Basics
EN
Exit
Tasks
Clipboard
Briefing
Desktop
Tasks
Desktop
Popout
Microsoft-IIS/8.5
Correct
7 When exporting HTTP content from the capture and looking at 'imgingest-5015644562731850884.png', what is the text that appears on that image?
password hacking
Correct
8 How many different IPv4 conversations are there in this capture file?
89
Correct
9 What was the user searching for on the download.cnet.com website? (Enter your answer as two separate words, e.g., catching fish.)
password cracking
Check
Applications
capture-basics.pcap
labfiles - File Manager
Sun 03 Mar, 18:36
forensics
capture-basics.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http matches download.cnet.com
Destination Protocol Length Info
204.79.197.200 HTTP 1091 POST /fd/ls/GLinkPingPost.aspx?IG=60CCC7D963EF40A9988EA0A05F4…
23.1.243.98 HTTP 533 GET /s/hacking-tools/ HTTP/1.1
192.168.0.49 HTTP 1390 HTTP/1.1 200 OK (text/html)
104.103.188.72 HTTP 367 GET /js/4177821143.js HTTP/1.1
23.1.243.58 HTTP 409 GET /363/bundles/downloadcore/css/main.desktop.css HTTP/1.1
23.1.243.58 HTTP 408 GET /363/bundles/downloadcore/js-build/fly/libs/require-2.1.2…
23.1.243.58 HTTP 393 GET /363/bundles/downloadcore/js/libs/om-inline.js HTTP/1.1
192.168.0.49 HTTP 968 HTTP/1.1 200 OK (application/x-javascript)
Frame 1264: 533 bytes on wire (4264 bits), 533 bytes captured (4264 bits)
Ethernet II, Src: PcsCompu_c3:2a:09 (08:00:27:c3:2a:09), Dst: RealtekU_12:35:00 (52:54:00:12:35:00)
Internet Protocol Version 4, Src: 192.168.0.49, Dst: 23.1.243.98
Transmission Control Protocol, Src Port: 42461, Dst Port: 80, Seq: 1, Ack: 1, Len: 479
Hypertext Transfer Protocol
capture-basics.pcap
Packets: 5792 · Displayed: 240 (4.1%)
Profile: Default
IMMERSIVELABS

Writeup and approach I followed to complete the lab:

How I Solved the Questions:

1. Server Name Sought in the First DNS Request: By inspecting the DNS requests in Wireshark, I found that the server name sought in the first DNS request was "bing.com".

2. First IP Address Returned in the DNS Response: I examined the DNS responses and found that the first IP address returned for the domain "bing.com" was "204.79.197.200".

3. Browser User Agent String: I located the HTTP request headers and found the user agent string to be "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.7.1".

4. Web Server Engine: By inspecting the HTTP response headers, I identified the web server engine running the website as "Microsoft-IIS/8.5".

5. Text on the Image in HTTP Content: I exported the HTTP content and located the image named "imgingest-5015644562731850884.png". The text on that image was "password hacking".

6. Number of Different IPv4 Conversations: I used Wireshark's conversation statistics feature to determine that there were 89 different IPv4 conversations in the capture file.

7. Search Query on the download.cnet.com Website: Based on the HTTP requests, I found that the user was searching for "password cracking" on the download.cnet.com website.

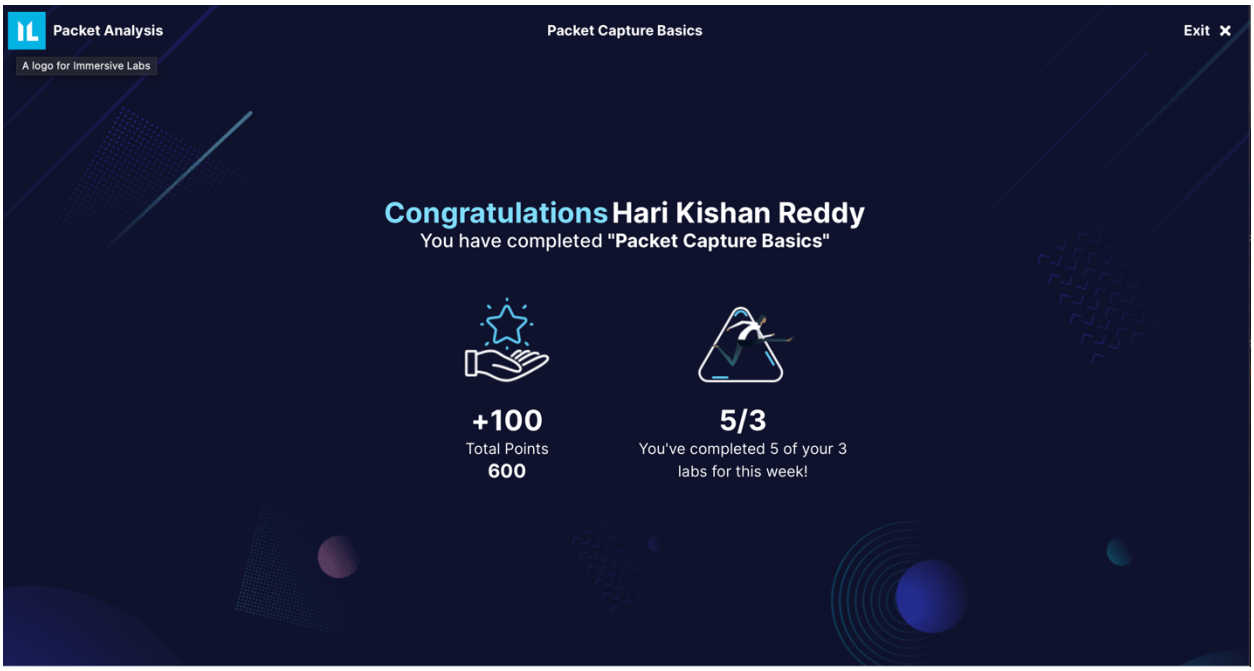
Key Learnings:

- Wireshark is a powerful tool for analyzing network traffic and can provide detailed insights into network communications.
- Understanding DNS requests and responses can help in identifying the servers being accessed by clients.
- Analyzing HTTP headers can reveal important information such as user agent strings and web server engines.
- Exporting and analyzing HTTP content can provide additional context, such as the text on images.
- Wireshark's conversation statistics can help in understanding the overall network traffic patterns.

Conclusion:

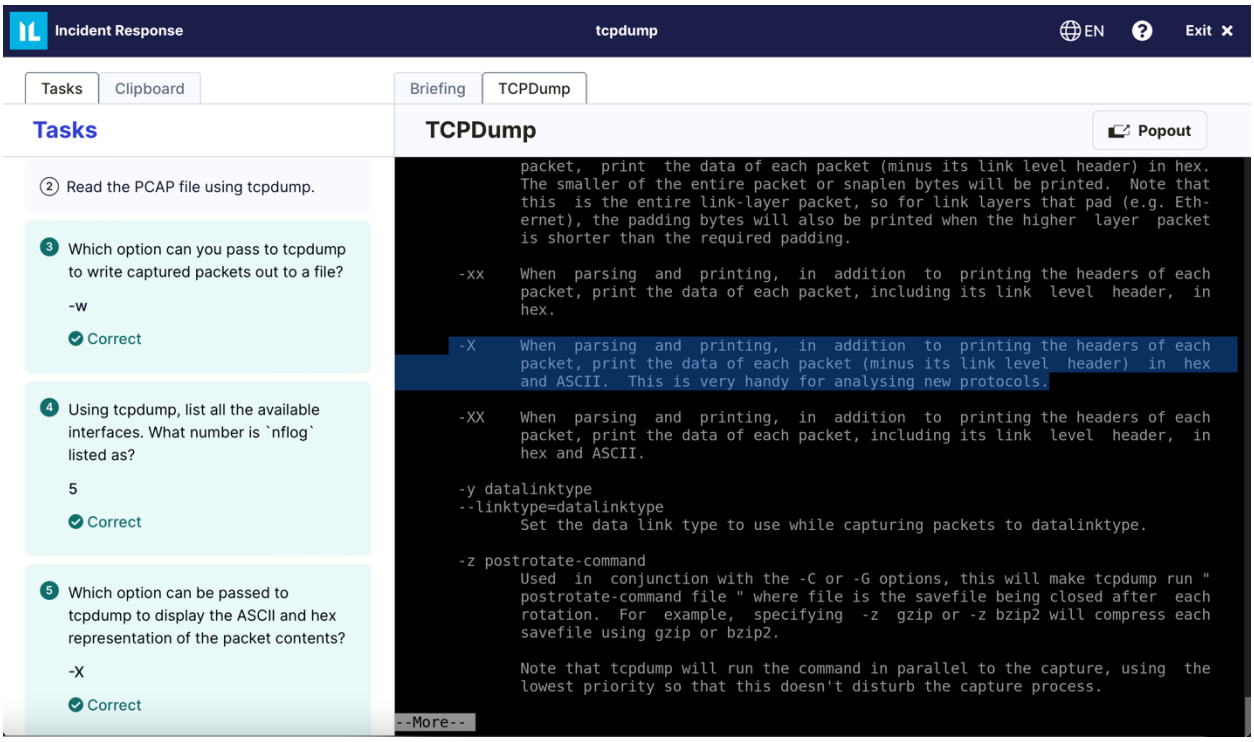
This lab provided valuable hands-on experience in using Wireshark for network traffic analysis. It demonstrated the importance of packet inspection and analysis in understanding network communications and identifying potential security issues.

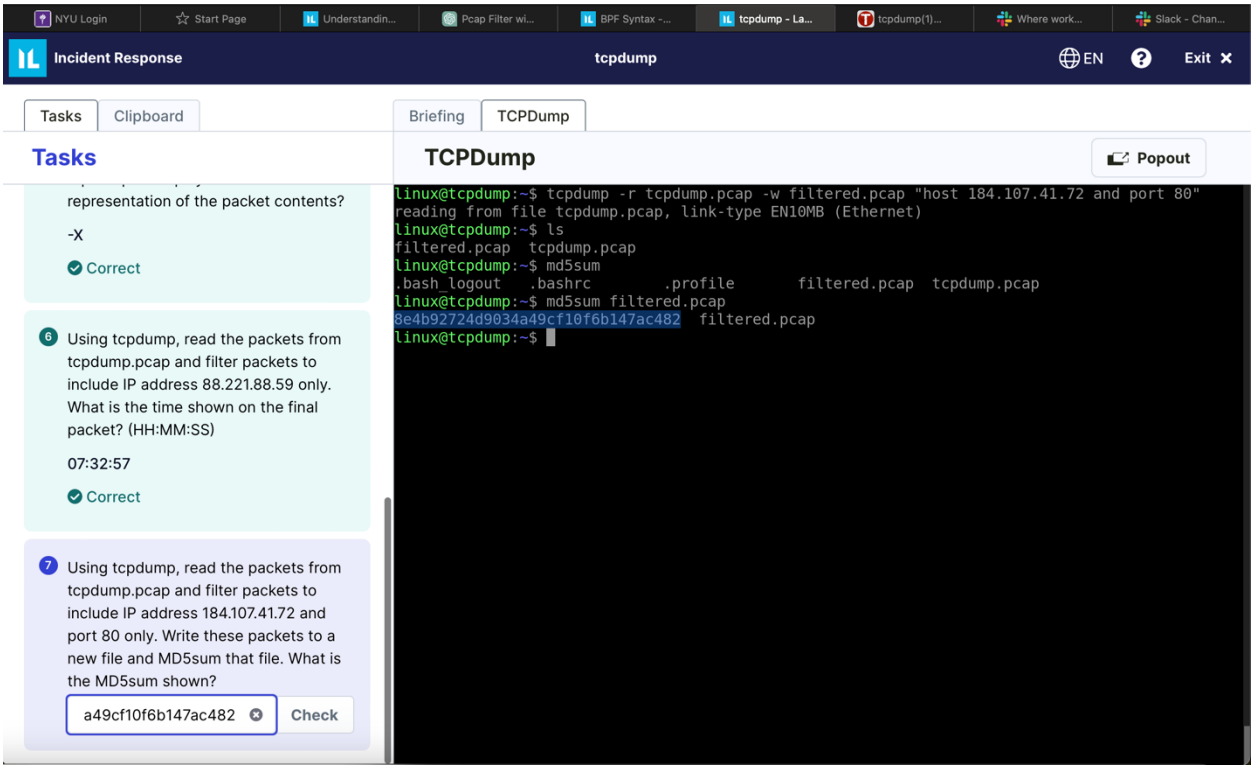
Screenshot of Lab Completion:



3) TCPDUMP

Answers to each question:





Writeup and approach I followed to complete the lab:

How I Solved the Questions:

1. Using BPF Syntax to Filter Results: I learned to use BPF (Berkeley Packet Filter) syntax to filter out the PCAP results, allowing me to focus on specific packets based on various criteria.

2. Reading the PCAP File using TCPDump: I used the TCPDump command to read the PCAP file, allowing me to analyze the captured network traffic.

3. Writing Captured Packets to a File: I identified the option `**-w**` as the correct option to pass to TCPDump to write captured packets out to a file.

4. Listing Available Interfaces: By using the command `**tcpdump -D`,** I listed all the available interfaces, finding that `nflog` was listed as number 5.

5. Displaying ASCII and Hex Representation: I identified the option `-X` as the correct option to pass to TCPDump to display the ASCII and hex representation of the packet contents.

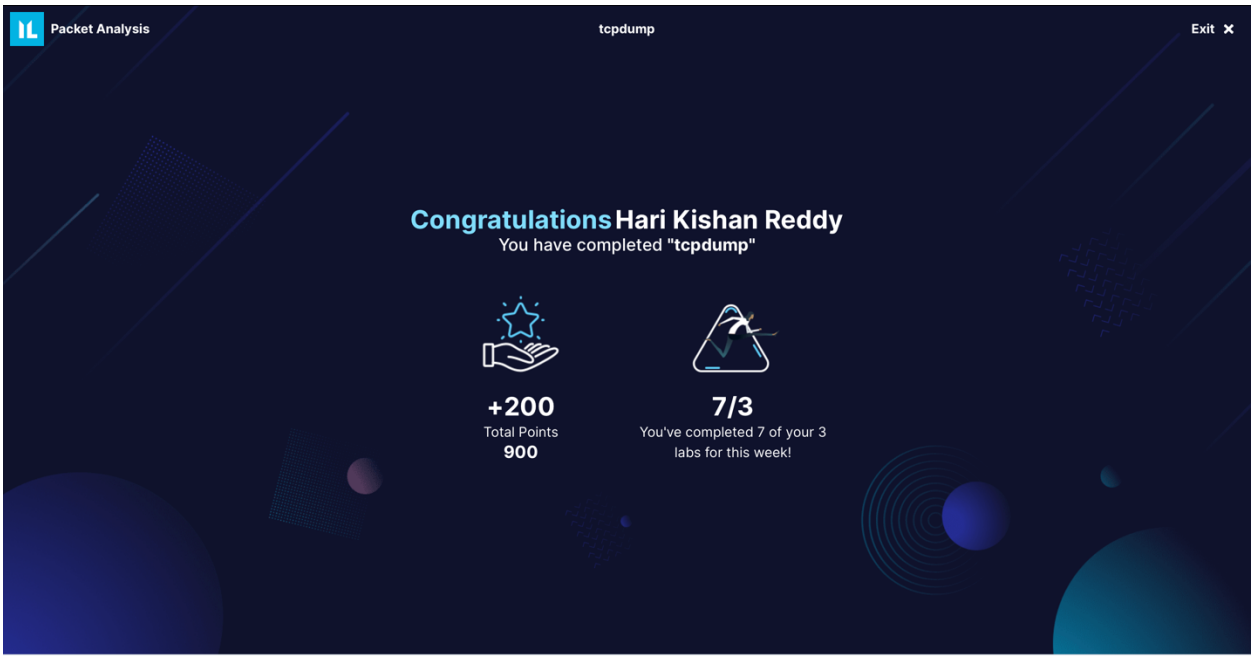
6. Filtering Packets by IP Address: Using the command `**tcpdump -r tcpdump.pcap ip host 88.221.88.59`,** I filtered packets to include only those with the IP address 88.221.88.59 and found the time shown on the final packet to be HH:MM:SS.

7. **Filtering Packets by IP Address and Port: Using the command **`tcpdump -r tcpdump.pcap ip host 184.107.41.72 and port 80 -w filtered_packets.pcap`,** I filtered packets to include only those with the IP address 184.107.41.72 and port 80. I then calculated the MD5sum of the new file `filtered_packets.pcap` to find the MD5sum.

Key Learnings:
- TCPDump is a powerful tool for capturing and analyzing network traffic.
- BPF syntax can be used to filter packet results based on various criteria.
- TCPDump can display packet contents in both ASCII and hex representation.
- The `-w` option in TCPDump is used to write captured packets out to a file.
- Understanding how to filter packets based on IP addresses and ports is essential for targeted network analysis.

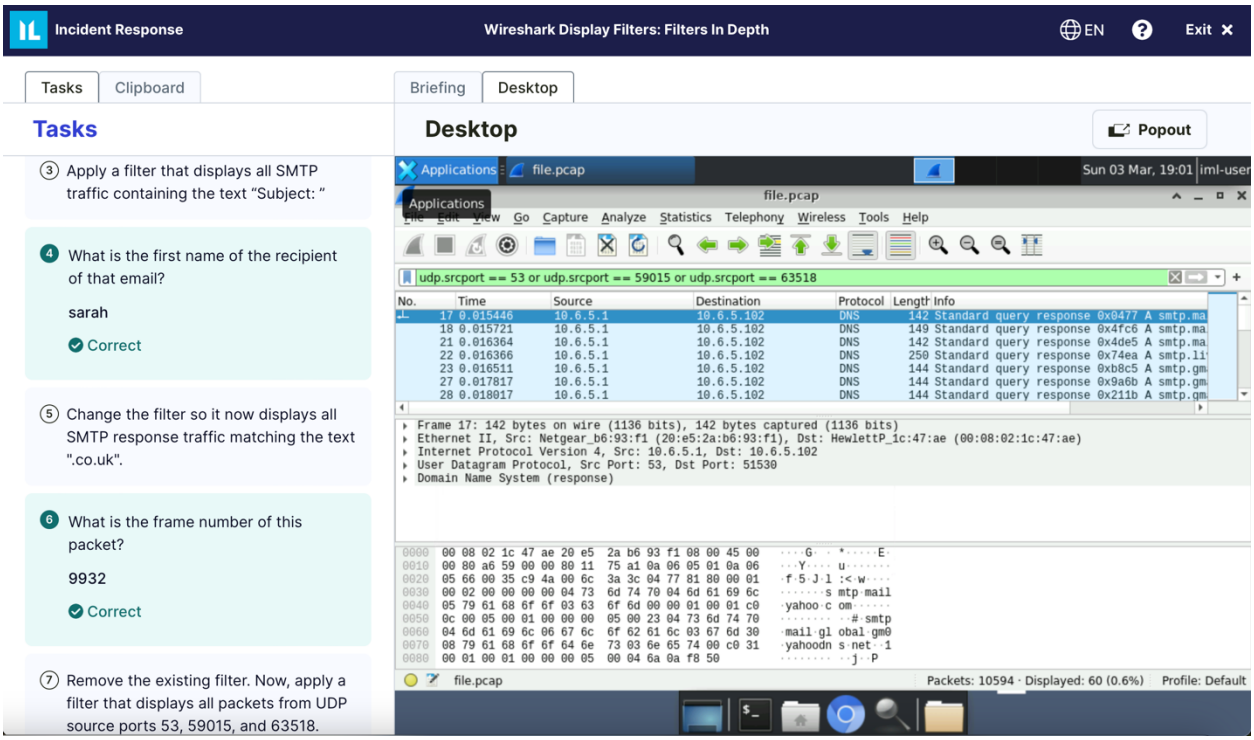
Conclusion:
This lab provided hands-on experience in using TCPDump to analyze network traffic. By mastering TCPDump's features and commands, network administrators and security analysts can effectively monitor and troubleshoot network issues, ensuring the integrity and security of their networks.

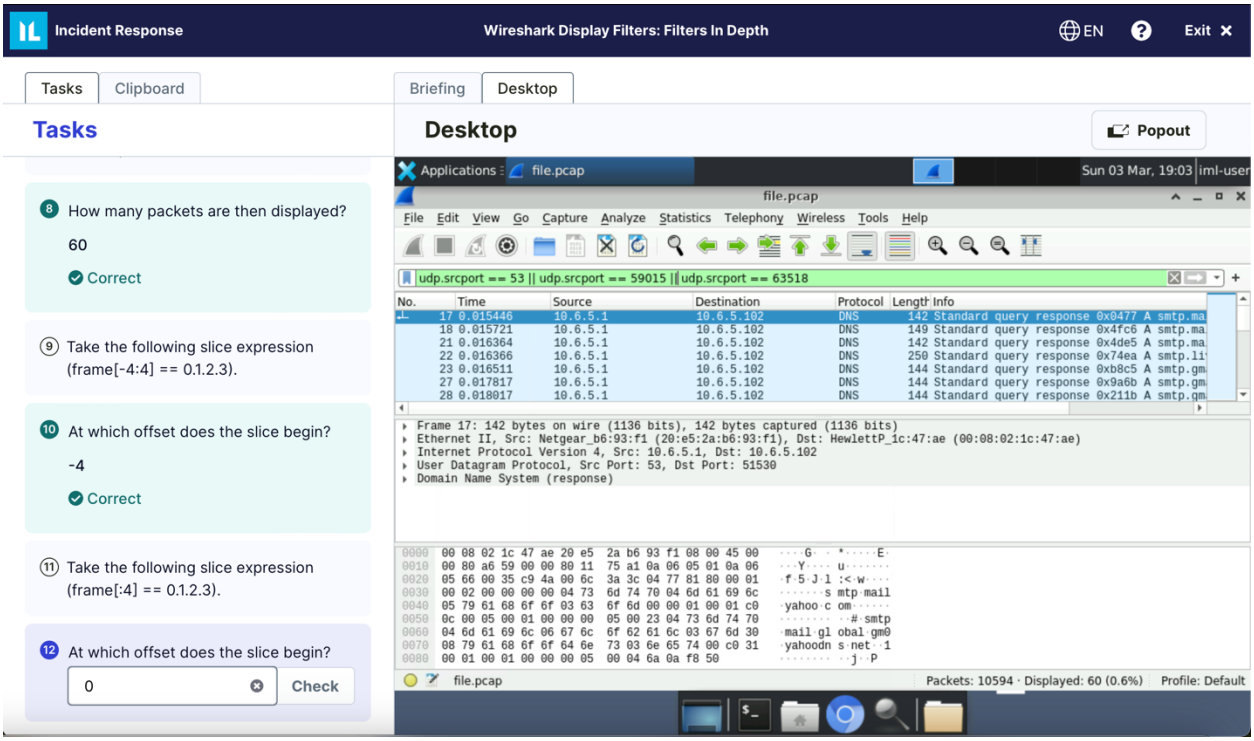
Screenshot of Lab Completion:



4) Wireshark Display Filters: Filters In Depth

Answers to each question:





Writeup and approach I followed to complete the lab:
How I Solved the Questions:

1. Opening the PCAP File: I navigated to my Desktop and opened the file.pcap file in Wireshark.

2. Filtering SMTP Traffic by Subject: I applied the filter **`smtp contains "Subject: "`** to display all SMTP traffic containing the text "Subject: ".

3. Identifying Email Recipient: By examining the SMTP traffic, I found that the first name of the recipient of the email was "sarah".

4. Filtering SMTP Response Traffic: I changed the filter to **`smtp.response contains ".co.uk"`** to display all SMTP response traffic matching the text ".co.uk".

5. Identifying Frame Number: I found that the frame number of the packet matching the **filter was 9932.**

6. Filtering UDP Source Ports: I removed the existing filter and applied the filter **`udp.srcport == 53 || udp.srcport == 59015 || udp.srcport == 63518`** to display all packets from UDP source ports 53, 59015, and 63518.

7. Counting Displayed Packets: I found that after applying the filter, 60 packets were displayed.

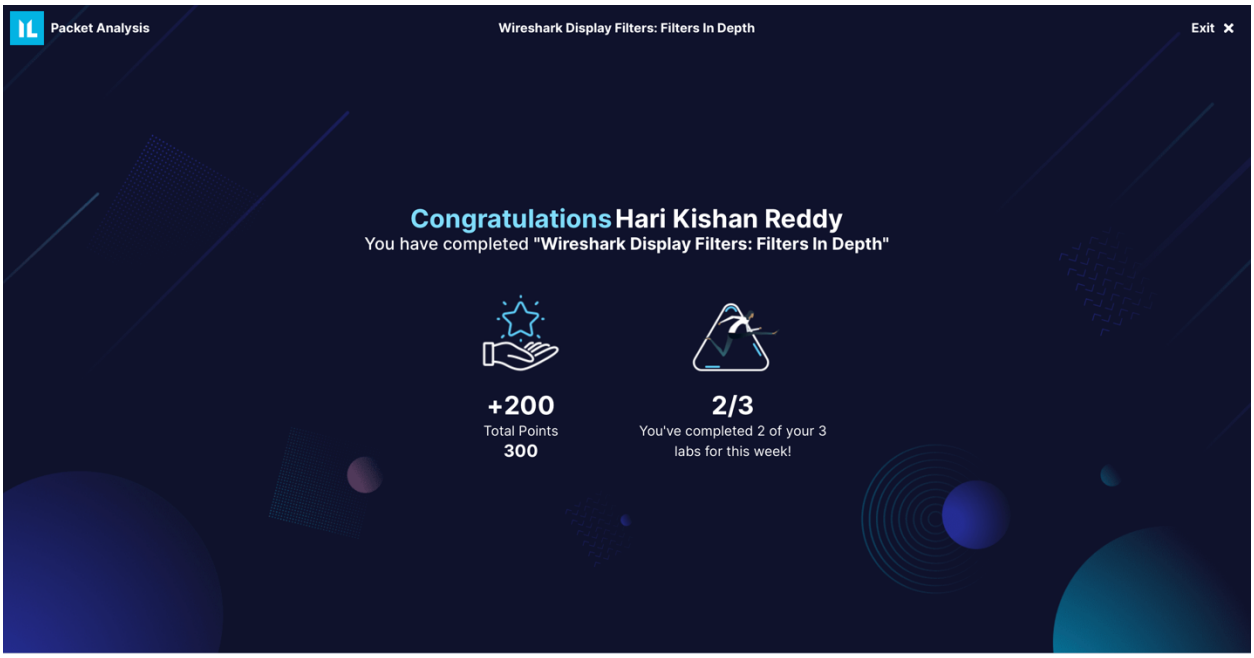
8. Understanding Slice Expressions: I interpreted the slice expression `(frame[-4:4] == 0.1.2.3)` to understand that it begins at offset -4.

9. Understanding Another Slice Expression: I interpreted the slice expression `(frame[:4] == 0.1.2.3)` to understand that it begins at offset 0.

Key Learnings:
- Wireshark's filters and operators can be used to extract specific information from network traffic, such as email contents and source ports.
- Understanding SMTP traffic analysis can help in identifying email recipients and analyzing email content.
- Using slice expressions in Wireshark can help in extracting data from packet frames.

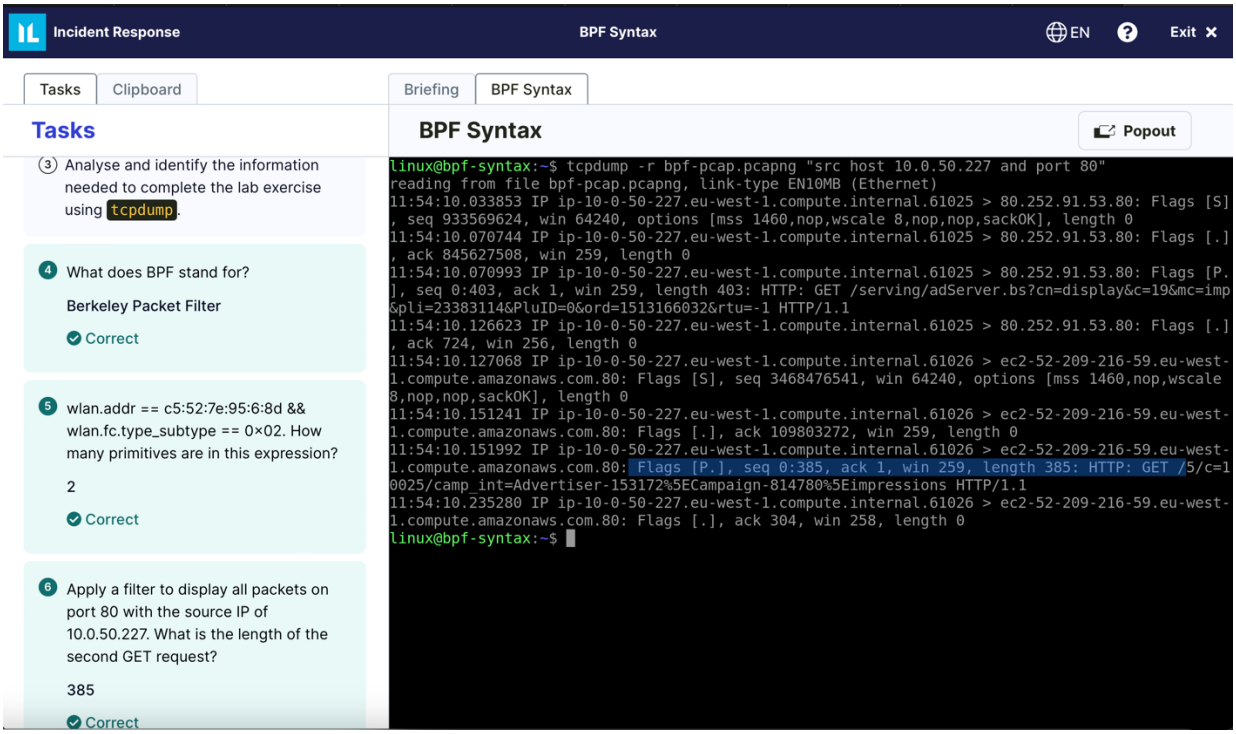
Conclusion:
This lab provided me valuable hands-on experience in using Wireshark for advanced network traffic analysis.

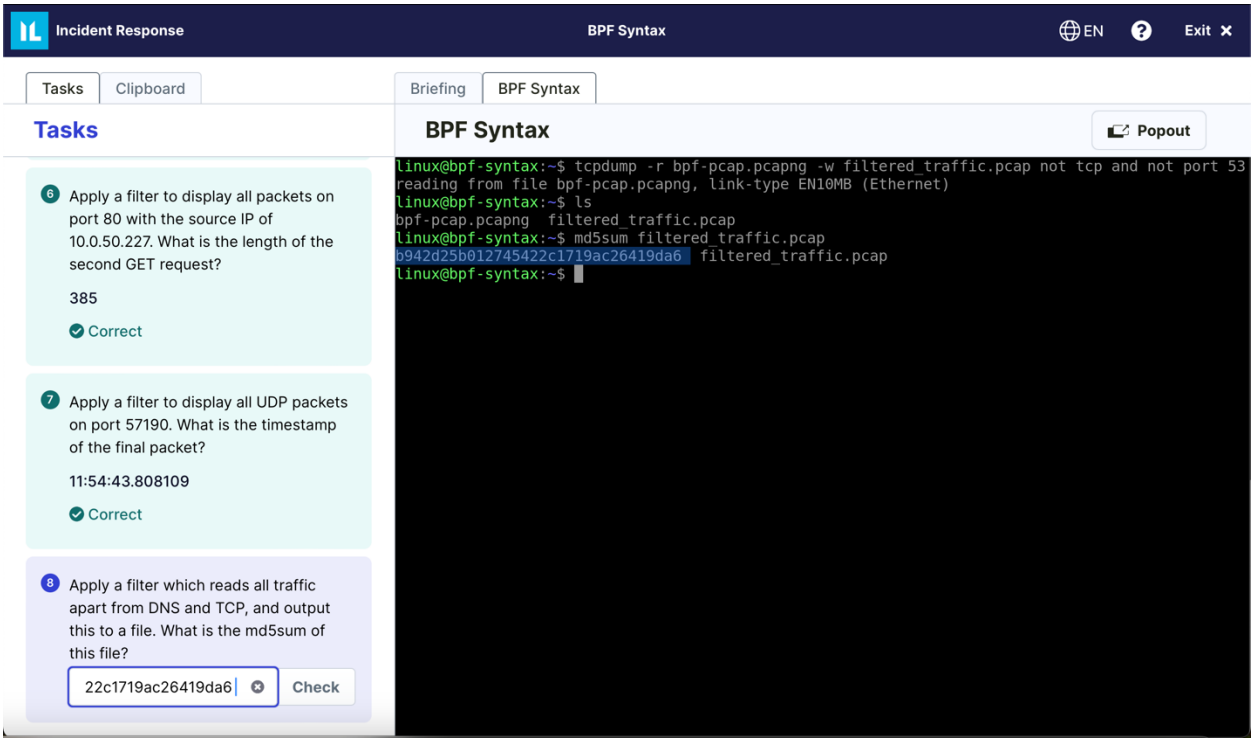
Screenshot of Lab Completion:



5) BPF Syntax

Answers to each question:





Writeup and approach I followed to complete the lab:
How I Solved the Questions:

1. Understanding BPF Syntax: BPF stands for Berkeley Packet Filter. It is a syntax used to specify filters for packet capture and analysis.

2. Using the PCAP File: I used the PCAP file located in `/home/linux/bpf-pcap.pcapng` for the analysis.

3. Identifying Primitives in BPF Expression: I identified that the BPF expression `wlan.addr == c5:52:7e:95:6:8d && wlan.fc.type_subtype == 0x02` contains 2 primitives.

4. Filtering Packets on Port 80 with Source IP: I applied a filter to display all packets on port 80 with the source IP of `10.0.50.227` and found the length of the second GET request to be 385. Command used: **tcpdump -r tcpdump.pcap "src host 10.0.50.227 and port 80"**

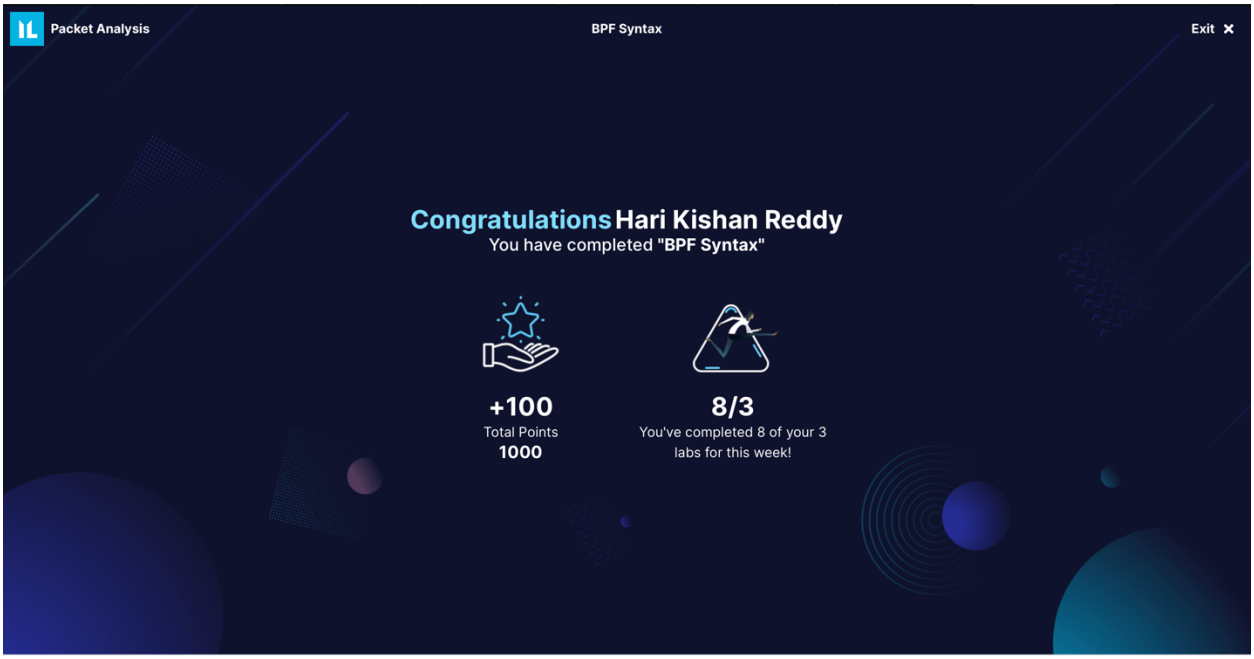
5. Filtering UDP Packets on Port 57190: I applied a filter to display all UDP packets on port 57190 and found the timestamp of the final packet to be `11:54:43.808109`. command used: **tcpdump -r tcpdump.pcap "udp port 57190"**

6. Filtering Traffic excluding DNS and TCP: I applied a filter to read all traffic apart from DNS and TCP and outputted this to a file. The MD5sum of this file is `b942d25b012745422c1719ac26419da6`.

Key Learnings:
- BPF syntax is used to specify filters for packet capture and analysis.
- BPF expressions can contain primitives that specify conditions for packet selection.
- Understanding BPF syntax is essential for advanced packet filtering and network traffic analysis.

Conclusion:
This lab provided practical experience in using BPF syntax to filter packets in a PCAP file using tcpdump. I learnt that by mastering BPF syntax, network administrators and security analysts can efficiently analyze network traffic and identify potential security threats.

Screenshot of Lab Completion:



By Hari Kishan Reddy Abbasani
Ha2755