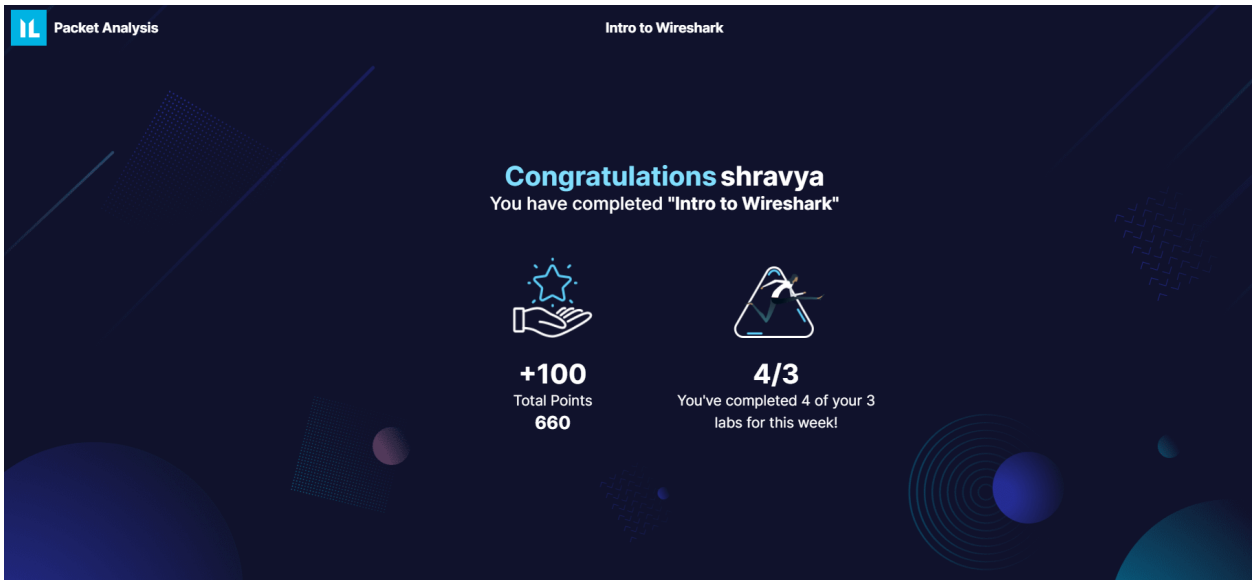
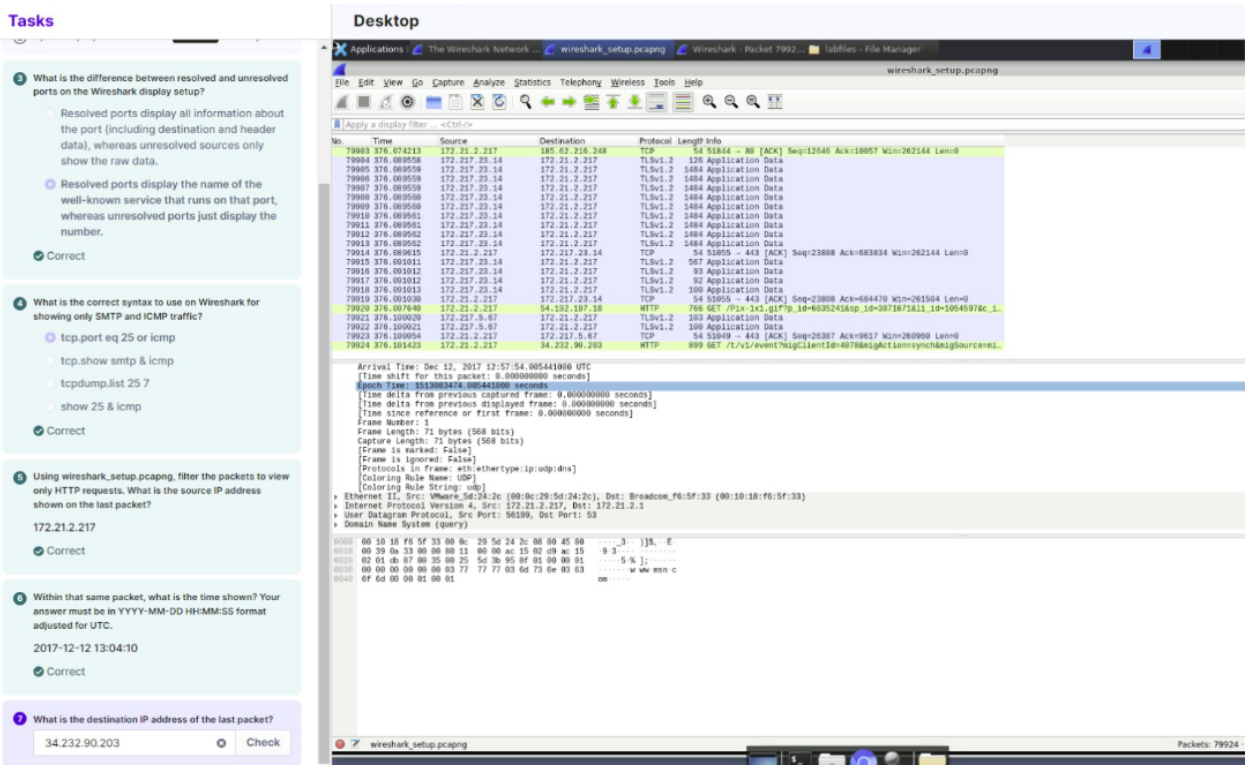
Assignment - 2

Lab 1: Wireshark and Network Traffic Analysis





Lab - 1 can be divided into 3 tasks namely:

Task 1: Wireshark Setup

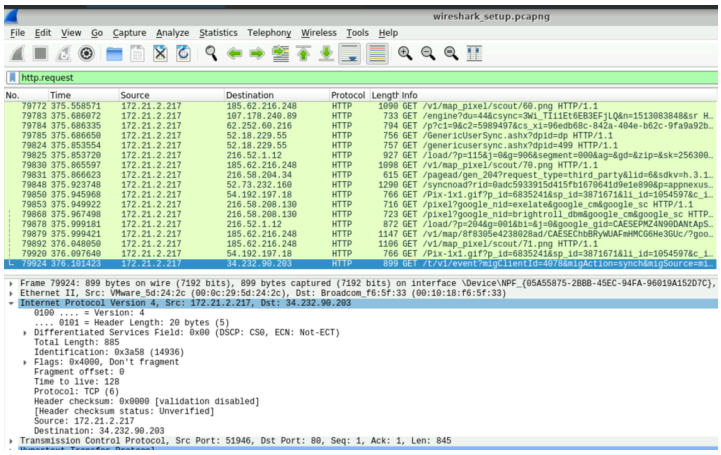
Question: What is the difference between resolved and unresolved ports on the Wireshark display setup?
Answer: Resolved ports display the name of the well-known service that runs on that port, whereas unresolved ports just display the number.
Explanation: In Wireshark, resolved ports show the well-known service names associated with the port numbers based on port-to-service mappings in the system. This makes it easier to identify the type of traffic, for example, "HTTP" for port 80, without having to refer to an external resource. Unresolved ports, on the other hand, only display the numerical port numbers, requiring manual mapping to understand the service being used.

Task 2: Wireshark Syntax

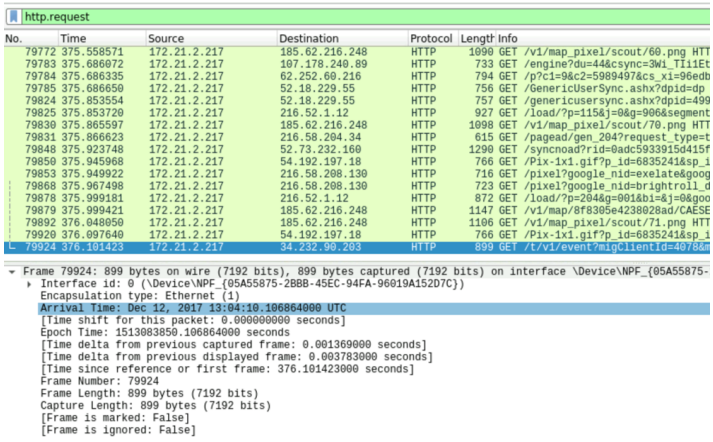
Question: What is the correct syntax to use on Wireshark for showing only SMTP and ICMP traffic?
Answer: tcp.port eq 25 or icmp
Explanation: In Wireshark, the correct syntax for filtering traffic is crucial. In this case, tcp.port eq 25 filters for TCP traffic on port 25 (SMTP), while icmp filters for ICMP traffic. The logical "or" (||) is used to combine these conditions, ensuring that both SMTP and ICMP traffic are displayed.

Task 3: Packet Analysis

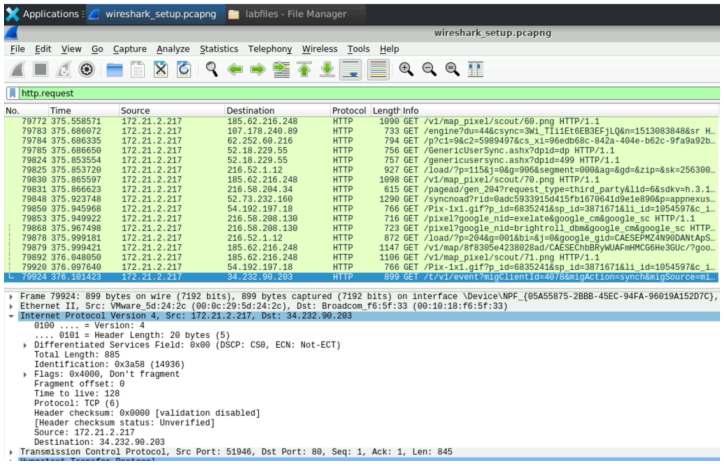
Question: Using wireshark_setup.pcapng, filter the packets to view only HTTP requests. What is the source IP address shown on the last packet?
Answer: The source IP address on the last packet is 172.21.2.217.
Explanation: After applying the filter to display only HTTP requests, the source IP address in the last packet is revealed. This information is essential for identifying the origin of the HTTP request.



Question: Within that same packet, what is the time shown? Your answer must be in YYYY-MM-DD HH:MM:SS format adjusted for UTC.
Answer: The time shown is 2017-12-12 13:04:10 (adjusted for UTC).
Explanation: The timestamp in the packet is expressed in the required format, adjusted to UTC from view tab. This timestamp is vital for understanding the temporal aspects of network traffic.



Question: What is the destination IP address of the last packet?
Answer: The destination IP address of the last packet is 34.232.90.203.
Explanation: By analyzing the packet content, one can extract the destination IP address. This information is crucial for determining where the HTTP request is directed.



Lab Writeup:

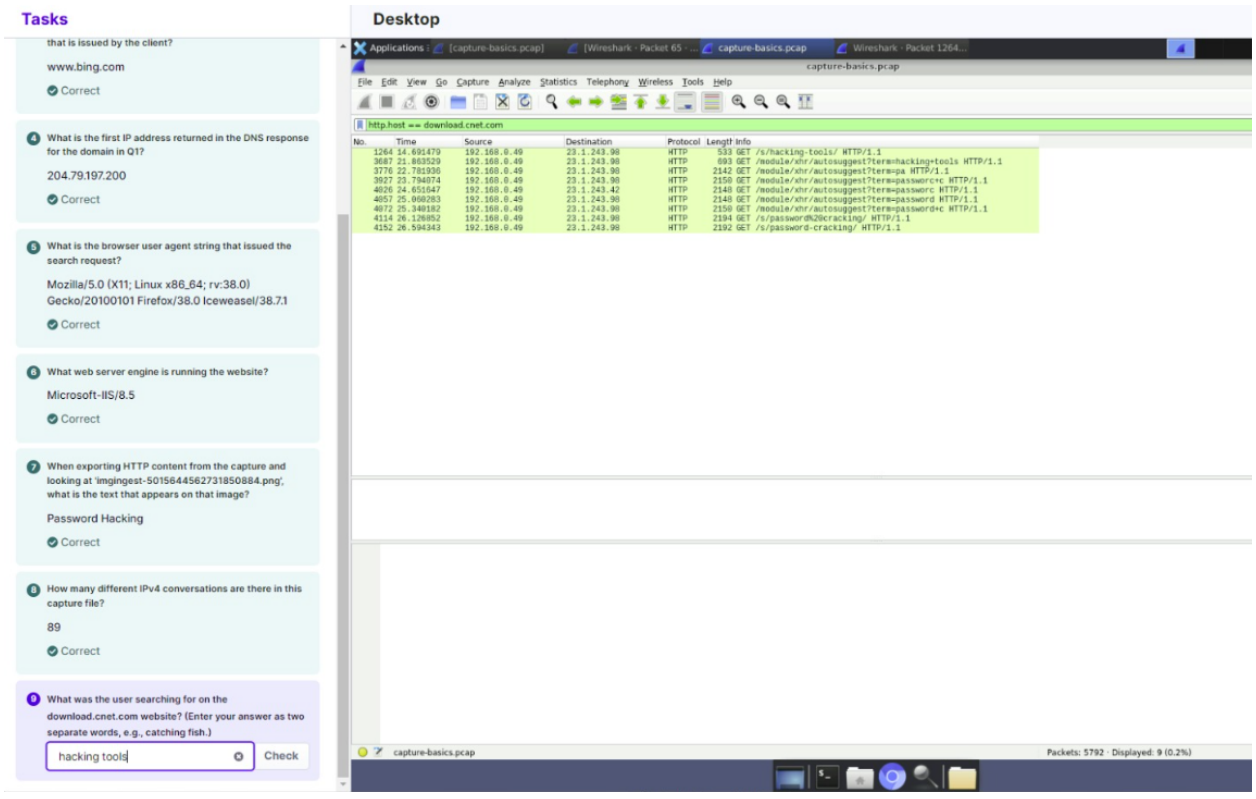
In Lab 1, I had hands-on with Wireshark, a powerful network traffic analysis tool. I delved into the concept of resolved and unresolved ports, discovering how resolved ports facilitate the quick identification of service names based on port numbers. I also practiced crafting Wireshark display filters, gaining hands-on experience in using syntax to isolate specific types of traffic, such as SMTP and ICMP. By applying these techniques, I learned the importance of technical

precision when analyzing network packets and understanding the details, like source and destination addresses, protocol types, and timestamps. This knowledge is invaluable for network administrators and security professionals, empowering them to gain deep insights into network activities.

Conclusion:

Lab 1 provided a strong foundation in using Wireshark for network traffic analysis. It sharpened the skills in configuring Wireshark to view distinct types of traffic and extracting essential technical details from network packets. These competencies will prove indispensable for anyone tasked with network monitoring, troubleshooting, and ensuring the robustness of network infrastructure.

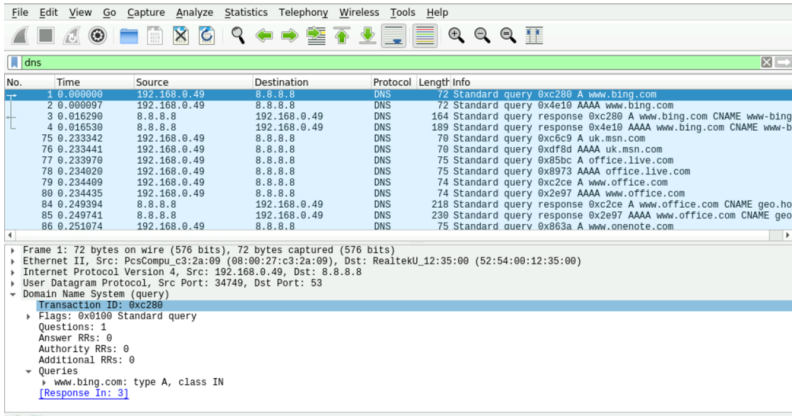
Lab 2 : Packet Capture Analysis Lab



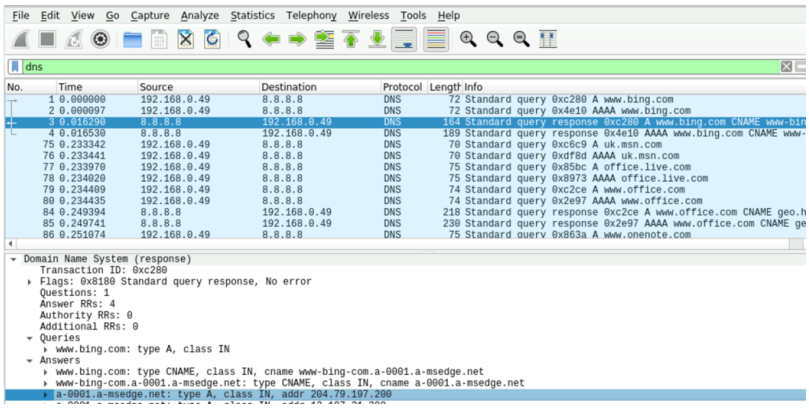
Lab-2 can be divided into 6 tasks as below:

Task 1: DNS Request Analysis

Question 1: What is the server name sought in the first DNS request issued by the client?
Answer: The server name sought in the first DNS request is "www.bing.com."
Explanation: In network traffic analysis, DNS (Domain Name System) requests are used to resolve domain names to IP addresses. The client initiated a DNS request seeking the IP address corresponding to "www.bing.com." This indicates a user's intent to access the Bing search engine.

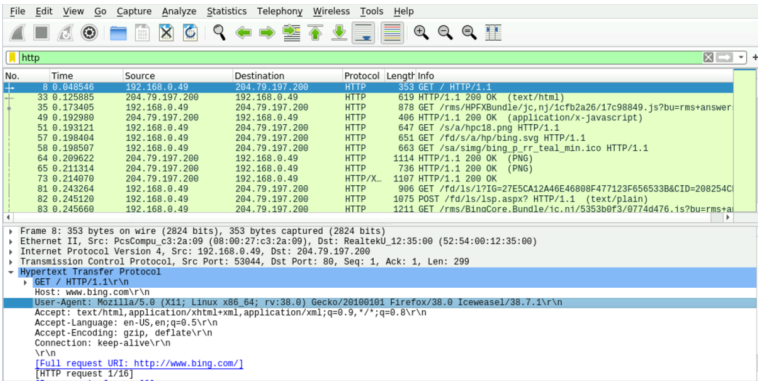


Question 2: What is the first IP address returned in the DNS response for the domain in Q1?
Answer: The first IP address returned in the DNS response for the domain "www.bing.com" is "204.79.197.200."
Explanation: The DNS response provides the mapping between a domain name and its associated IP address. In this case, "204.79.197.200" is the initial IP address provided for "www.bing.com."



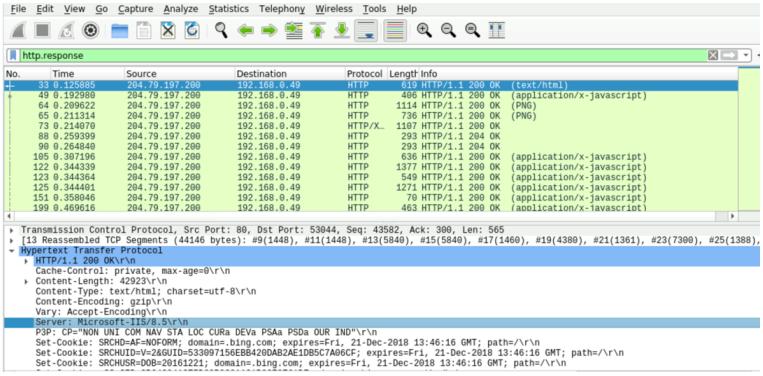
Task 2: Browser User Agent Analysis

Question 3: What is the browser user agent string that issued the search request?
Answer: The browser user agent string for the search request is "Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.7.1."
Explanation: The user agent string identifies the type and version of the web browser used for the search request. This string indicates the use of a Mozilla-based browser on a Linux system with specific versions.(User agent value is located in the Hypertext Transfer Protocol portion of the packet detail section.)



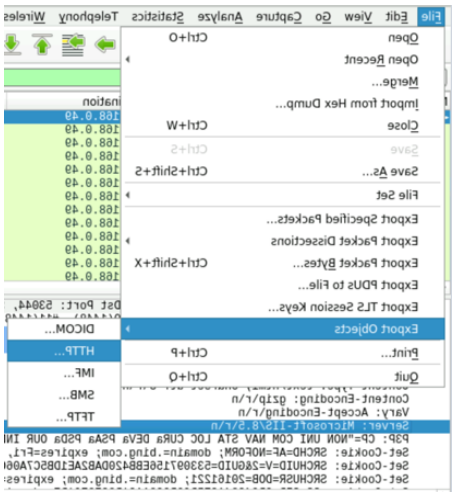
Task 3: Web Server Analysis

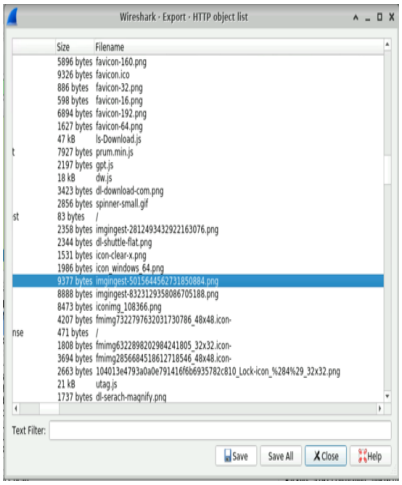
Question 4: What web server engine is running the website?
Answer: The web server engine running the website is "Microsoft-IIS/8.5."
Explanation: The "Microsoft-IIS" (Internet Information Services) web server engine is a product from Microsoft. The version "8.5" indicates the specific version of IIS running the website. (Apply the "http.response" display filter. The server engine can be discovered in the Hypertext Transfer Protocol portion of the first packet's detail section.)



Task 4: HTTP Content Analysis

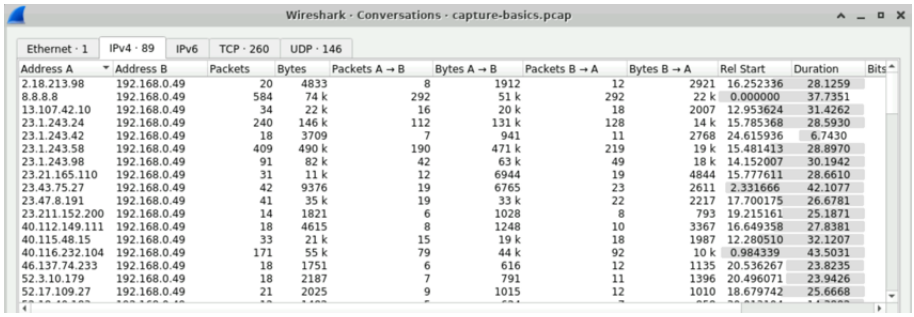
Question 5: When exporting HTTP content and looking at 'imgingest-5015644562731850884.png,' what is the text that appears on that image?
Answer: The text that appears on the image 'imgingest-5015644562731850884.png' is "Password Hacking."
Explanation: This answer reveals the content of the image. In this case, the image 'imgingest-5015644562731850884.png' contains the text "Password Hacking."(File->Export Objects-> HTTP ->select 'imgingest-5015644562731850884.png')





Task 5: IPv4 Conversations Analysis

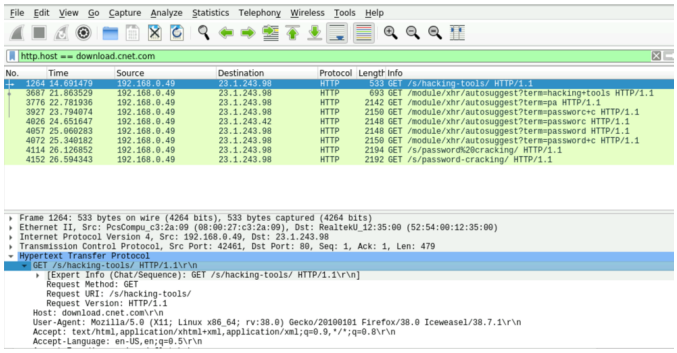
Question 6: How many different IPv4 conversations are there in this capture file?
Answer: There are 89 different IPv4 conversations in this capture file.
Explanation: IPv4 conversations represent unique interactions between source and destination IP addresses. In this capture, there are 89 distinct pairs of IP addresses engaged in conversations. (statistics in the top menu ->Conversations -> HTTP -> go to IPV4 section)



Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits
2.18.213.98	192.168.0.49	20	4833	8	1912	12	2921	16.252336	28.1259	
8.8.8.8	192.168.0.49	584	74 k	292	51 k	292	22 k	0.000000	37.7351	
13.107.42.10	192.168.0.49	34	22 k	16	20 k	18	2007	12.953624	31.4262	
23.1.243.24	192.168.0.49	240	146 k	112	131 k	128	14 k	15.785368	28.5930	
23.1.243.42	192.168.0.49	18	3709	7	941	11	2768	24.615936	6.7430	
23.1.243.58	192.168.0.49	409	490 k	190	471 k	219	19 k	15.481413	28.8970	
23.1.243.98	192.168.0.49	91	82 k	42	63 k	49	18 k	14.152007	30.1942	
23.21.165.110	192.168.0.49	31	11 k	12	6944	19	4844	15.777611	28.6610	
23.43.75.27	192.168.0.49	42	9376	19	6765	23	2611	2.331666	42.1077	
23.47.8.191	192.168.0.49	41	35 k	19	33 k	22	2217	17.700175	26.6781	
23.211.152.200	192.168.0.49	14	1821	6	1028	8	793	19.215161	25.1871	
40.112.149.111	192.168.0.49	18	4615	8	1248	10	3367	16.649358	27.8381	
40.115.48.15	192.168.0.49	33	21 k	15	19 k	18	1987	12.280510	32.1207	
40.116.232.104	192.168.0.49	171	55 k	79	44 k	92	10 k	0.984339	43.5031	
46.137.74.233	192.168.0.49	18	1751	6	616	12	1135	20.536267	23.8235	
52.3.10.179	192.168.0.49	18	2187	7	791	11	1396	20.496071	23.9426	
52.17.109.27	192.168.0.49	21	2025	9	1015	12	1010	18.679742	25.6668	

Task 6: User Search Query Analysis

Question 7: What was the user searching for on the download.cnet.com website?
Answer: The user was searching for "hacking tools" on the download.cnet.com website.
Explanation: The search query "hacking tools" was issued by the user on the download.cnet.com website, indicating the user's specific search intent.(Apply filter “http.host == download.cnet.com” ->open first packet ->Answer can be found in the
info section of the packet under GET.)



Lab Writeup:

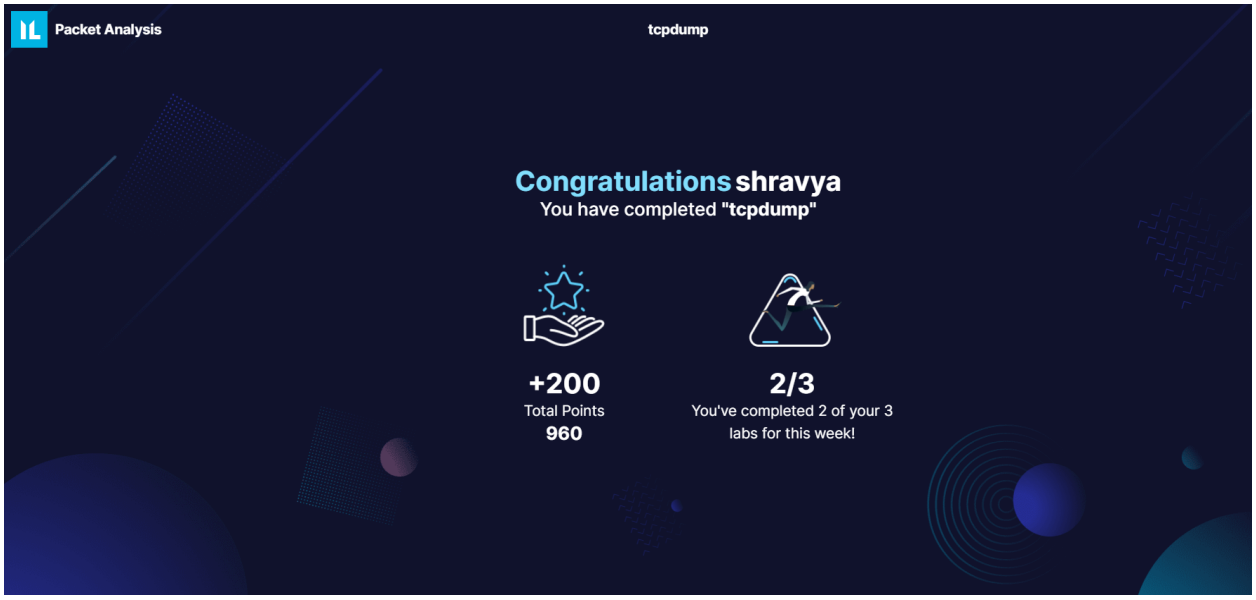
In the Packet Capture Analysis Lab, I delved into the intricate world of network packet analysis, leveraging Wireshark to scrutinize a packet capture file. The primary objective was to sharpen the skills of a Cyber SOC Analyst by extracting precise information from network traffic. The practical exposure to network activity and packet analysis tools offered several technical insights.
DNS Request Analysis revealed the ability to identify sought server names and DNS responses. This proficiency is invaluable for network troubleshooting and security investigation.
Browser User Agent Analysis enlightened us about the significance of user agent strings in profiling user interactions. Recognizing browser details aids in understanding user behavior, a vital aspect of cybersecurity.
Web Server Analysis allowed us to identify the server engines behind websites. This knowledge is critical for assessing vulnerabilities.
HTTP Content Analysis showcased how image content can bear hidden messages, emphasizing the importance of scrutinizing all aspects of network traffic.
IPv4 Conversations Analysis sharpened the skills in tracking network conversations, aiding in identifying anomalies and threats.
User Search Query Analysis revealed user intent, crucial for threat assessment.
In conclusion, the lab imparted the skills needed to investigate network activities, enhancing one’scapabilities as Cyber SOC Analysts. Learning outcomes encompassed DNS analysis, user profiling, web server assessment, content scrutiny, conversation tracking, and understanding

user intent. These skills are invaluable for monitoring, analyzing, and safeguarding network infrastructure.
This lab further solidified the significance of meticulous network analysis in the realm of cybersecurity, emphasizing the importance of technical proficiency in packet capture and analysis.

Conclusion:

The Packet Capture Analysis Lab was an immersive journey into network traffic analysis. It empowered us with advanced skills in dissecting packets, understanding DNS, user agents, web servers, and content. We now possess the knowledge to unravel complex network scenarios, contributing to enhanced security and proactive threat detection. This lab epitomizes the essential training needed for a Cyber SOC Analyst in the dynamic realm of network security.

Lab 3: Tcp dump



Tasks

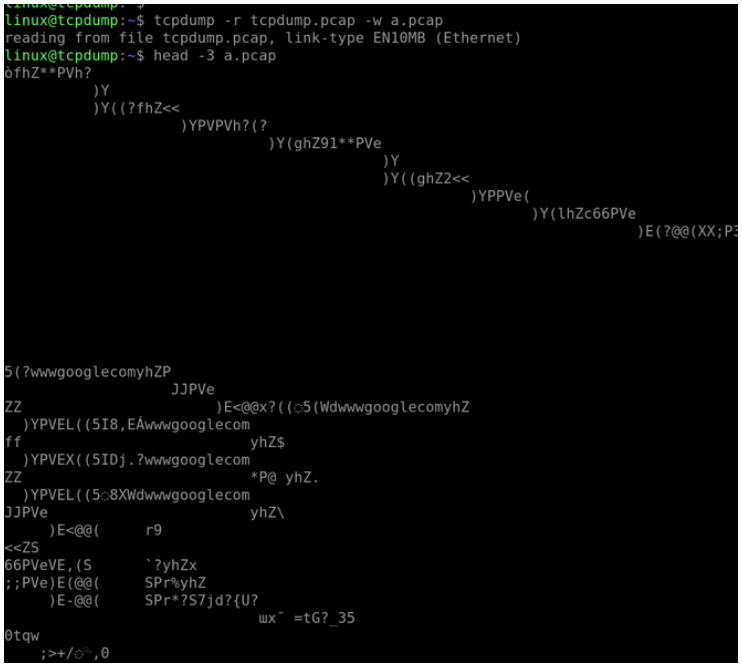
Machines — Reset

TCPDump — Open >

1. Learn to use BPF syntax to filter out the PCAP results.
2. Read the PCAP file using tcpdump.
3. Which option can you pass to tcpdump to write captured packets out to a file?
   -W
   Correct
4. Using tcpdump, list all the available interfaces. What number is `nflog` listed as?
   5
   Correct
5. Which option can be passed to tcpdump to display the ASCII and hex representation of the packet contents?
   -X
   Correct
6. Using tcpdump, read the packets from tcpdump.pcap and filter packets to include IP address 88.221.88.59 only. What is the time shown on the final packet? (HH:MM:SS)
   07:32:57
   Correct
7. Using tcpdump, read the packets from tcpdump.pcap and filter packets to include IP address 184.107.41.72 and port 80 only. Write these packets to a new file and MD5sum that file. What is the MD5sum shown?
   8e4b92724d9034a49cf10f6b147ac482
   Correct

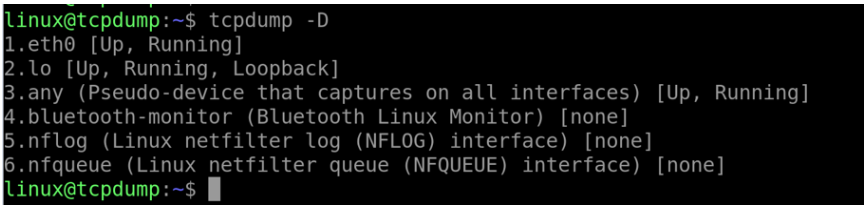
Task 1: Reading the PCAP File

Question: What option can you pass to tcpdump to write captured packets out to a file?
Answer: The option is -w.
Explanation: To write captured packets to a file, one can use the -w option, followed by the desired filename. For example: tcpdump -r [filename.pcapng] -w [output_filename].



Task 2: Listing Available Interfaces

Question: Using tcpdump, list all the available interfaces. What number is nflog listed as?
Answer: nflog is listed as interface number 5.
Explanation: I used the command tcpdump -D to list all available interfaces, and interface number 5 was assigned to nflog.



Task 3: Displaying ASCII and Hex Representation

Question: Which option can be passed to tcpdump to display the ASCII and hex representation of the packet contents?
Answer: The option is -X.
Explanation: To display both ASCII and hex representation of packet contents, I used the -X option with tcpdump.

```
linux@tcpdump:~$ tcpdump -r tcpdump.pcap -X
```

```
        0x4290:  c706 c717 4a67 f916 4d67 3195 ebb4 3bf7  ....Jg..Mg1...;.
        0x42a0:  223f 56ac 6aec 34a3 7a81 bbb0 84bc 68d9  "?V.j.4.z.....h.
        0x42b0:  404c 2654 30cc 95f0 4736 303c 2607 b695  @L&T0...G60<&...
        0x42c0:  1702 666f d816 3e41 a51c 252a d374 cf63  ..fo..>A..%*.t.c
        0x42d0:  7b64 5831 da61 36b2 1cba 12f0 fb96 0ecc  {dX1.a6.........
        0x42e0:  fc92 69da 509d 2d23 cb74 48b4 2f96 5e21  ..i.P.-#.tH./.^!
        0x42f0:  e8e2 70ec e670 bc79 f1b7 a9e1 af64 11c3  ..p..p.y.....d..
        0x4300:  4a40 3aad ce6f f1d9 0083 68ca 9817 3e63  J@:..o....h...>c
        0x4310:  90b3 2aab 4363 6233 409c 04d9 db08 79a4  ..*.Ccb3@.....y.
        0x4320:  0ebe 96e1 76b6 9d47 fe87 c021 b277 b3c3  ....v..G...!.w..
        0x4330:  8b18 4f3d 19bc c8dc ac65 5b34 2fe9 6173  ..O=.....e[4/.as
        0x4340:  5d4a 0177 2044 ae04 a16e 72ce 0a6b ad72  ]J.w.D...nr..k.r
        0x4350:  090e 0df0 d620 66ba 8218 1e11 3a0a 0b40  ......f.....:..@
        0x4360:  8041 68ff bbbe 72ba 87a3 c50a f9fb 5d93  .Ah...r.......].
        0x4370:  1288 0276 d4e3 5128 f3e1 ae8a c002 dc3d  ...v..Q(.......=
        0x4380:  514f 9dbf 71ba 3f13 fb79 a4cb 3c9c c067  QO..q.?..y..<..g
        0x4390:  82fb 6203 f6d3 c495 76b8 02b1 7c3e d8ad  ..b.....v...|>..
        0x43a0:  c595 4b0b ebfe bedb 15fe 6357 e44d 56a9  ..K.......cW.MV.
        0x43b0:  734a 8c5a 8484 c9aa 7ffe 4c69 0ae6 f711  sJ.Z......Li....
        0x43c0:  d6ac 27bb 9013 c084 3d12 4c5c 6cef fe21  ..'.....=.L\l..!
        0x43d0:  0392 9ce8 93f2 308c 5b3d d4f9 4021 2cff  ......0.[=..@!,.
        0x43e0:  51e1 8017 49b9 9ce9 1c2e be23 3d60 1aa9  Q...I......#=`..
        0x43f0:  e796 2222 7620 ff63 83c7 ad37 e317 f471  ..""v..c...7...q
        0x4400:  9c45 8967 5db0 8ebf 50ed 7e31 fdce a795  .E.g]...P.~1....
        0x4410:  4056 5790 f7a9 8d2e a0e7 f255 4661 93ff  @VW........UFa..
        0x4420:  3b2c 6c5c 997a 9121 77e1 b668 19c6 de9b  ;,l\.z.!w..h....
        0x4430:  e864 66b2 0c26 f129 6437 2b20 57cc 94e8  .df..&.)d7+.W...
        0x4440:  6e2c bb81 2183 10d0 ce1b 3ca8 aa70 31ec  n,..!.....<..p1.
        0x4450:  e07f 63b0 62c5 7ab8 4fe2 10f1 a114 065a  ..c.b.z.O......Z
        0x4460:  68d0 78dc fdf4 b4f2 459c 5615 c3cd d35f  h.x.....E.V....
```

Task 4: Filtering Packets by IP Address

Question: Using tcpdump, read the packets from tcpdump.pcap and filter packets to include IP address 88.221.88.59 only. What is the time shown on the final packet? (HH:MM:SS)
Answer: The time shown on the final packet is 07:32:57.
Explanation: I used the command tcpdump -r tcpdump.pcap host 88.221.88.59 to filter packets by the specified IP address and obtained the final packet's timestamp.

```
linux@tcpdump:~$
linux@tcpdump:~$
linux@tcpdump:~$ tcpdump -r tcpdump.pcap -nn host 88.221.88.59
reading from file tcpdump.pcap, link-type EN10MB (Ethernet)
07:31:56.197987 IP 192.168.21.133.40646 > 88.221.88.59.80: Flags [.], ack 2054538429, win 30016, length 0
07:31:56.198136 IP 88.221.88.59.80 > 192.168.21.133.40646: Flags [.], ack 1, win 64240, length 0
07:32:06.438054 IP 192.168.21.133.40646 > 88.221.88.59.80: Flags [.], ack 1, win 30016, length 0
07:32:06.438365 IP 88.221.88.59.80 > 192.168.21.133.40646: Flags [.], ack 1, win 64240, length 0
07:32:16.679955 IP 192.168.21.133.40646 > 88.221.88.59.80: Flags [.], ack 1, win 30016, length 0
07:32:16.678082 IP 88.221.88.59.80 > 192.168.21.133.40646: Flags [.], ack 1, win 64240, length 0
07:32:26.921868 IP 192.168.21.133.40646 > 88.221.88.59.80: Flags [.], ack 1, win 30016, length 0
07:32:26.921990 IP 88.221.88.59.80 > 192.168.21.133.40646: Flags [.], ack 1, win 64240, length 0
07:32:37.158275 IP 192.168.21.133.40646 > 88.221.88.59.80: Flags [.], ack 1, win 30016, length 0
07:32:37.158725 IP 88.221.88.59.80 > 192.168.21.133.40646: Flags [.], ack 1, win 64240, length 0
07:32:47.397977 IP 192.168.21.133.40646 > 88.221.88.59.80: Flags [.], ack 1, win 30016, length 0
07:32:47.398547 IP 88.221.88.59.80 > 192.168.21.133.40646: Flags [.], ack 1, win 64240, length 0
07:32:57.638112 IP 192.168.21.133.40646 > 88.221.88.59.80: Flags [.], ack 1, win 30016, length 0
07:32:57.638538 IP 88.221.88.59.80 > 192.168.21.133.40646: Flags [.], ack 1, win 64240, length 0
linux@tcpdump:~$
linux@tcpdump:~$
```

Task 5: Filtering Packets by IP Address and Port

Question: Using tcpdump, read the packets from tcpdump.pcap and filter packets to include IP address 184.107.41.72 and port 80 only. Write these packets to a new file and MD5sum that file. What is the MD5sum shown?
Answer: The MD5sum is 8e4b92724d9034a49cf10f6b147ac482.
Explanation: I filtered packets by IP address and port using the command tcpdump -r tcpdump.pcap host 184.107.41.72 and port 80 -w filtered.pcap and then calculated the MD5sum of the resulting file filtered.pcap.

Lab Writeup:

In this lab, I delved into the intricacies of network analysis using tcpdump, a command-line packet analysis tool. This technical exploration deepened one's understanding of network traffic capture and analysis.

I began by grasping the basics, including writing captured packets to a file using -w and identifying available interfaces with -D. Moving forward, I unlocked the power of tcpdump with -X, revealing both ASCII and hex representations of packet contents.

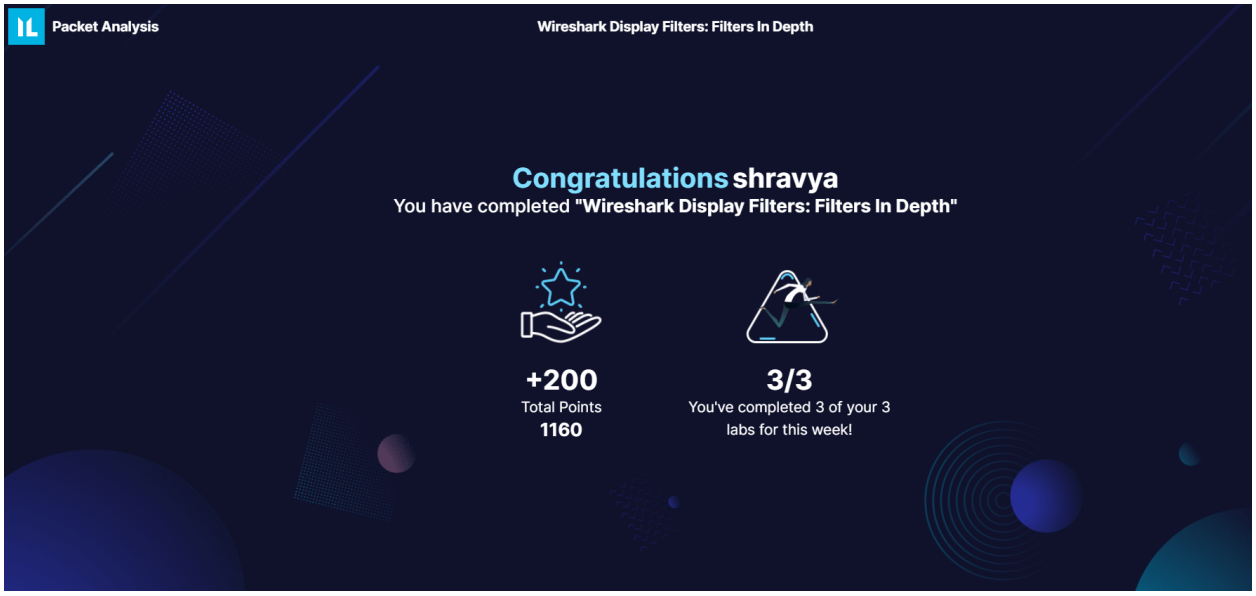
The real essence of network analysis came to the fore when I harnessed tcpdump's filtering capabilities. Filtering packets by IP address and port sharpened one's skills, culminating in MD5sum verification.

Overall, this lab demonstrated that tcpdump, though devoid of a GUI, is a formidable ally for network administrators and security analysts. It underscored the importance of command-line tools for network analysis, enriching one's capabilities in dissecting and understanding the network's inner workings.

Conclusion:

This lab underscored the significance of tcpdump as a valuable tool for network analysis. I gained insights into filtering packets and capturing specific network traffic, skills essential for system administrators and network security analysts.

Lab 4: Wireshark display filters: Filters in depth



3 Analyse and identify the information needed to complete the lab exercise.

4 From the PCAP provided, apply a filter which displays all SMTP traffic containing the text "Subject: ". What is the first name of the recipient of that email?

Sarah

Correct

5 From the PCAP provided, apply a filter which displays all SMTP response traffic matching the text ".co.uk". What is the frame number of this packet?

9932

Correct

6 From the PCAP provided, apply a filter which displays all packets from UDP source ports 53, 59015, and 63518. How many packets are then displayed?

60

Correct

7 Take the following slice expression (frame[-4:4] == 0.1.2.3). At which offset does the slice begin?

4 Check

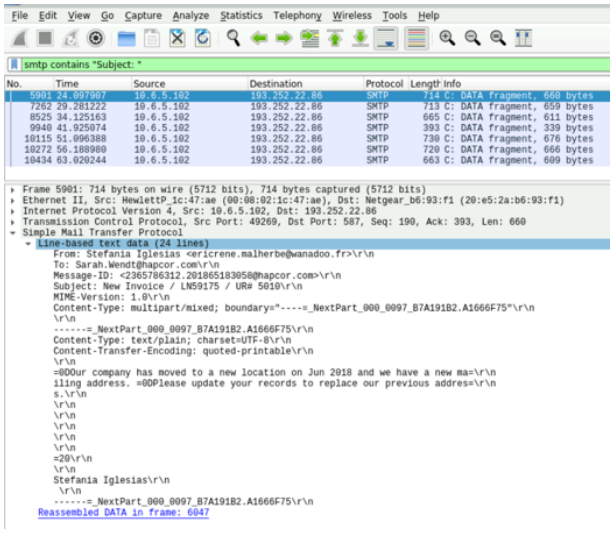
8 Take the following slice expression (frame[:4] == 0.1.2.3). At which offset does the slice begin?

0

Correct

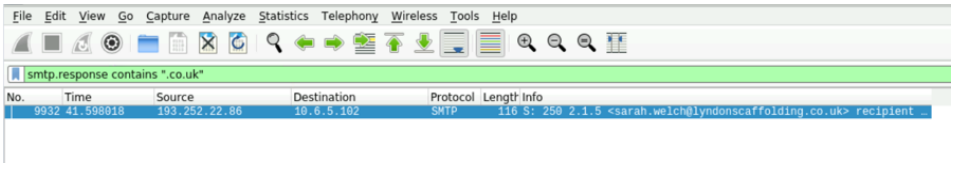
Task 1: Filtering SMTP Traffic by "Subject":

Question: Apply a filter to display all SMTP traffic containing the text "Subject: ". What is the first name of the recipient of that email?
Answer: The first name of the recipient is "Sarah."
Explanation: I applied the filter "smtp contains "Subject: "" to isolate SMTP traffic with the specific text in the subject. Scanning through the results, I found the name "Sarah" in the relevant email.



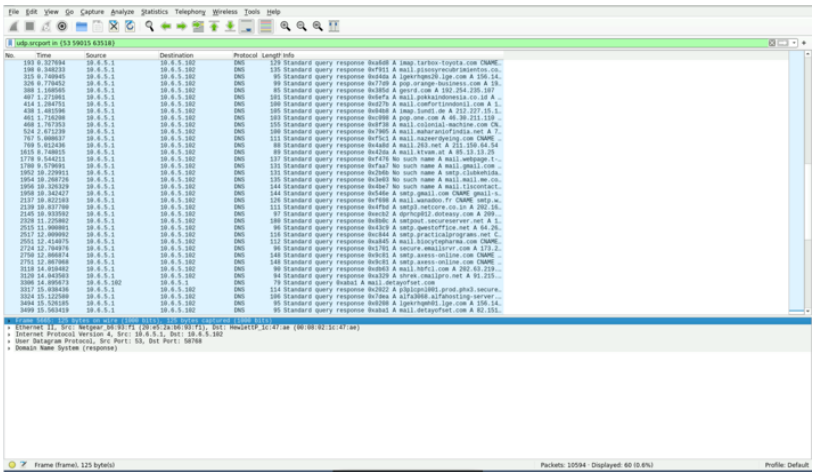
Task 2: Filtering SMTP Response with ".co.uk":

Question: Apply a filter to display all SMTP response traffic matching the text ".co.uk". What is the frame number of this packet?
Answer: The frame number of this packet is "9932."
Explanation: I used the filter "smtp.response contains ".co.uk"" to locate SMTP response traffic containing the specified text. I then identified the frame number as "9932."



Task 3: Filtering UDP Packets from Specific Source Ports:

Question: Apply a filter to display all packets from UDP source ports 53, 59015, and 63518. How many packets are displayed?
Answer: The filter displayed a total of "60" packets.
Explanation: I created a filter using the syntax "udp.srcport == 53 || udp.srcport == 59015 || udp.srcport == 63518" to isolate packets from the specified UDP source ports. The result showed "60" relevant packets.



Task 4: Understanding Slice Expressions:

Question 4.1: Take the following slice expression (frame[-4:4] == 0.1.2.3). At which offset does the slice begin?
Answer 4.1: The slice begins at the offset of "-4."

Question 4.2: Take the following slice expression (frame[:4] == 0.1.2.3). At which offset does the slice begin?
Answer 4.2: The slice begins at the offset of "0."
Explanation: Slices in Wireshark allow us to filter specific sections of a field. In the first example, the slice begins at an offset of "-4," while in the second example, it starts at an offset of "0."

Lab Writeup:

An essential tool in the network analyst's toolbox is a set of Wireshark display filters. I learned about the difficulties of filtering network traffic in this lab, which improved my ability to separate important data from a deluge of packets.

These filters are used for more than just convenience; they are essential for the effective analysis of network data. We can quickly locate pertinent information thanks to the filters, which is especially useful when dealing with traffic brought on by malware. Any network analyst needs to be able to filter packets by protocol, source, destination, or content.

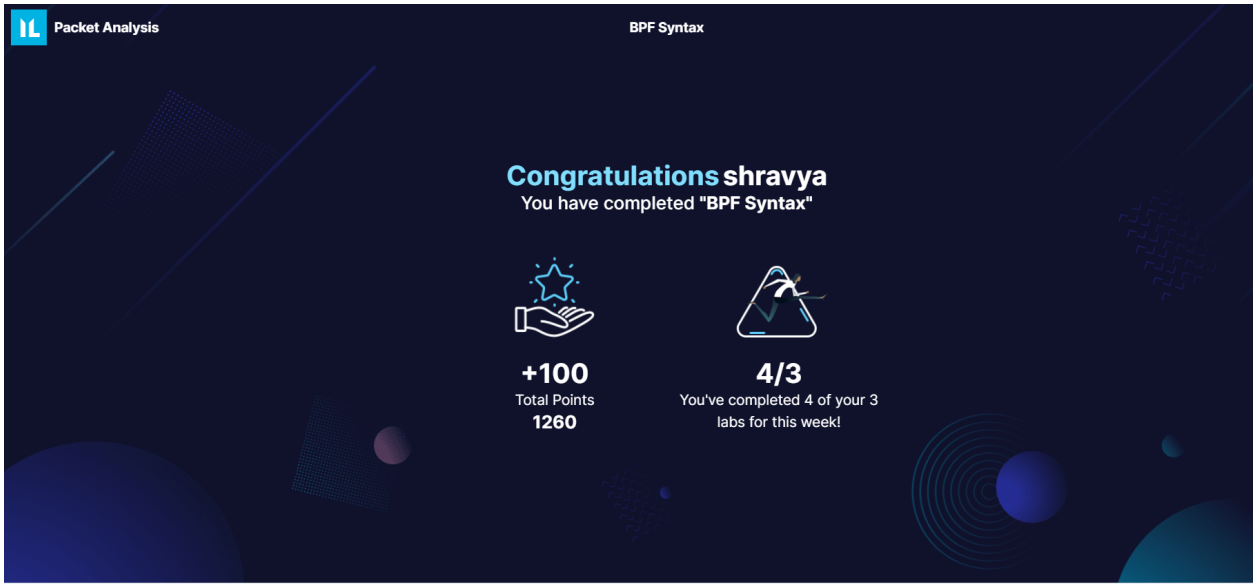
Additionally, sophisticated methods to create exact filters are provided by complex operators like comparison operators and slice operators, enabling us to look for patterns, values, and segments within packets.

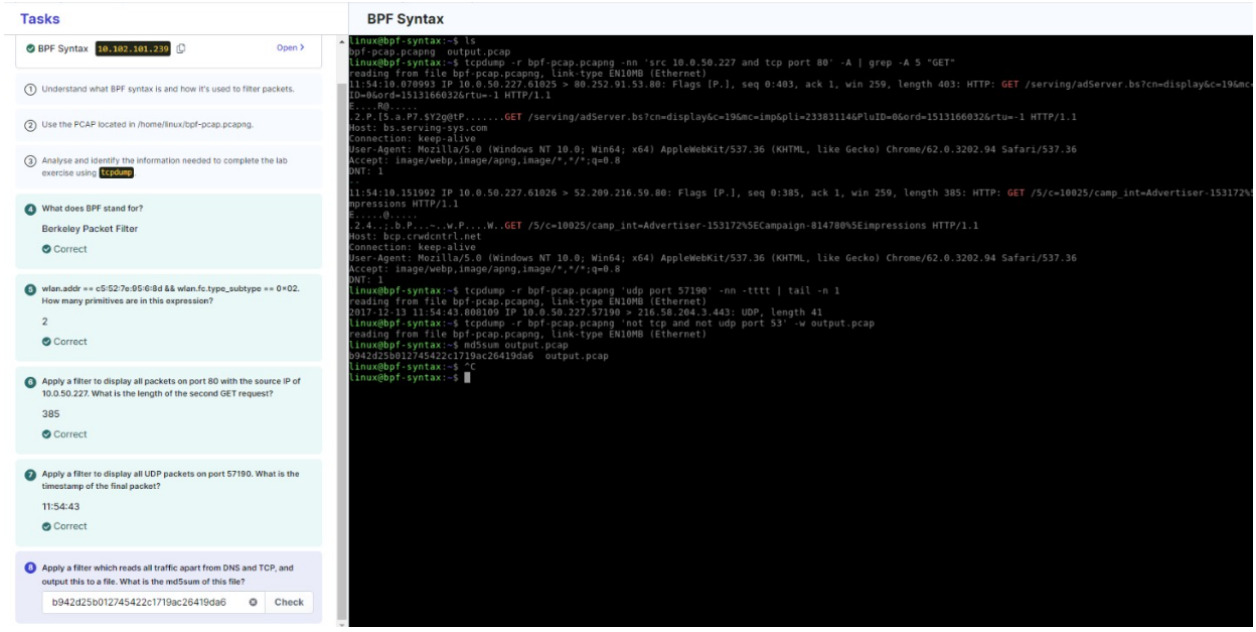
Conclusion:

This lab experience has provided a deep dive into Wireshark display filters, enabling precise packet filtering. By understanding complex operators, I can craft filters to extract essential information from a network capture efficiently. Wireshark's powerful filtering capabilities are indispensable for network analysis, helping to detect anomalies, troubleshoot issues, and uncover crucial indicators of compromise (IOCs).

In a highly technical context, the importance of Wireshark in network security and analysis cannot be overstated. It equips analysts with the tools to dissect and make sense of network traffic, a critical skill in today's cybersecurity landscape.

Lab 5: BPF Syntax





Task 1: Understand BPF Syntax

Question: What does BPF stand for?
Answer: Berkeley Packet Filter
Explanation: In this task, I gained an understanding of the Berkeley Packet Filter (BPF) and its significance in filtering network packets. BPF, originating from BSD Unix Systems, serves to filter unwanted packets as early as possible when capturing live network traffic. It is utilized as a capture filter in tools like Wireshark and in Linux terminal utilities like tcpdump. I grasped that

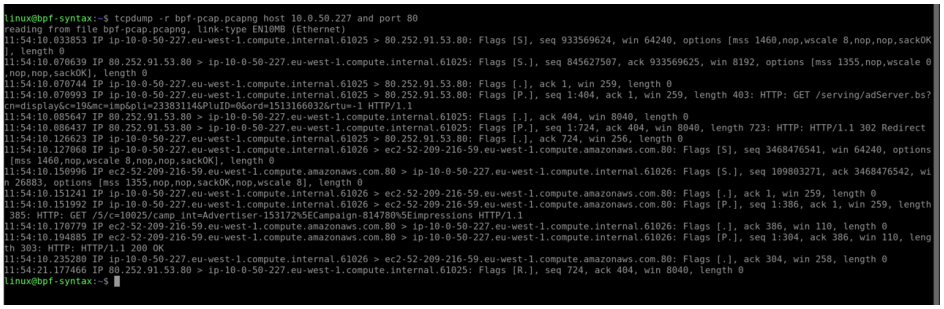
BPF differs from display filters; capture filters are set before packet capture and cannot be changed during the capture, reducing the raw size of the capture. Display filters, in contrast, are employed post-capture to analyze packets.

Task 2: Number of Primitives in Expression

Question: "wlan.addr == c5:52:7e:95:6:8d && wlan.fc.type_subtype == 0x02." How many primitives are in this expression?
Answer: 2
Explanation: The provided BPF expression "wlan.addr == c5:52:7e:95:6:8d && wlan.fc.type_subtype == 0x02" contains two primitives. This insight illustrates the capability to construct complex filtering expressions by combining qualifiers and operators, essential for precise packet filtering.

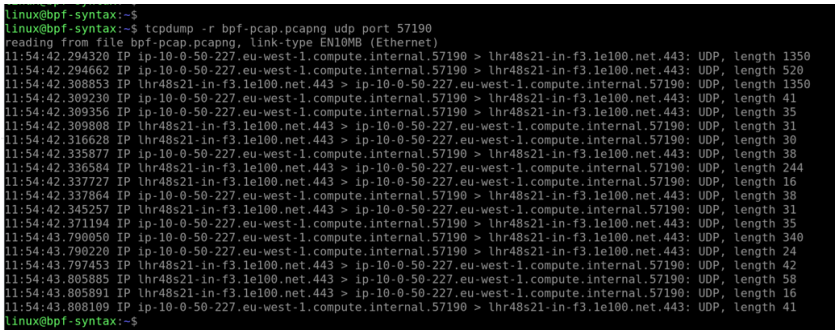
Task 3: Length of Second GET Request

Question: Apply a filter to display all packets on port 80 with the source IP of 10.0.50.227. What is the length of the second GET request?
Answer: 385
Explanation: I applied a filter to display packets originating from source IP 10.0.50.227 on port 80. The length of the second GET request in the filtered packets was 385 bytes. This task exemplifies the practical use of BPF syntax for filtering network traffic based on specific criteria.



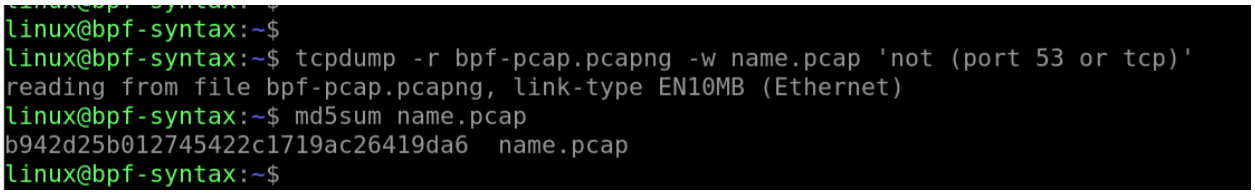
Task 4: Timestamp of Final Packet

Question: Apply a filter to display all UDP packets on port 57190. What is the timestamp of the final packet?
Answer: 11:54:43
Explanation: By applying a filter to display all UDP packets on port 57190, I determined that the timestamp of the final packet was "11:54:43." This task demonstrates BPF's role in analyzing network packets with precision, in this case, focusing on UDP traffic.



Task 5: MD5sum of Filtered Traffic File

Question: Apply a filter which reads all traffic apart from DNS and TCP, and output this to a file. What is the md5sum of this file?
Answer: b942d25b012745422c1719ac26419da6
Explanation: I used a filter to exclude DNS and TCP traffic and saved the resulting traffic to a file. The MD5sum of this file was "b942d25b012745422c1719ac26419da6." This illustrates how BPF syntax can be employed to filter and save specific network traffic for further analysis or archival.



Lab Writeup:

In completing this lab, I acquired a comprehensive understanding of BPF syntax and its critical role in packet filtering and network analysis. BPF, rooted in the Berkeley Packet Filter, is instrumental in capturing and analyzing network traffic efficiently. I learned to construct filtering expressions using BPF, combining qualifiers, operators, and primitives to precisely filter packets based on source, destination, protocol, and more. Additionally, I understood the difference between capture filters and display filters, recognizing that capture filters reduce the size of packet captures from the outset. This lab emphasized the importance of accurate and efficient network traffic analysis, ensuring that only relevant data is captured and scrutinized. Such skills are invaluable for network troubleshooting, security monitoring, and optimizing network performance.

Conclusion:

The BPF Syntax and Packet Filtering lab proved to be an instructive experience, demonstrating the practical application of BPF syntax in network analysis. The ability to construct precise filters allows for the efficient capture and analysis of network traffic, a critical skill in modern networking and security contexts.