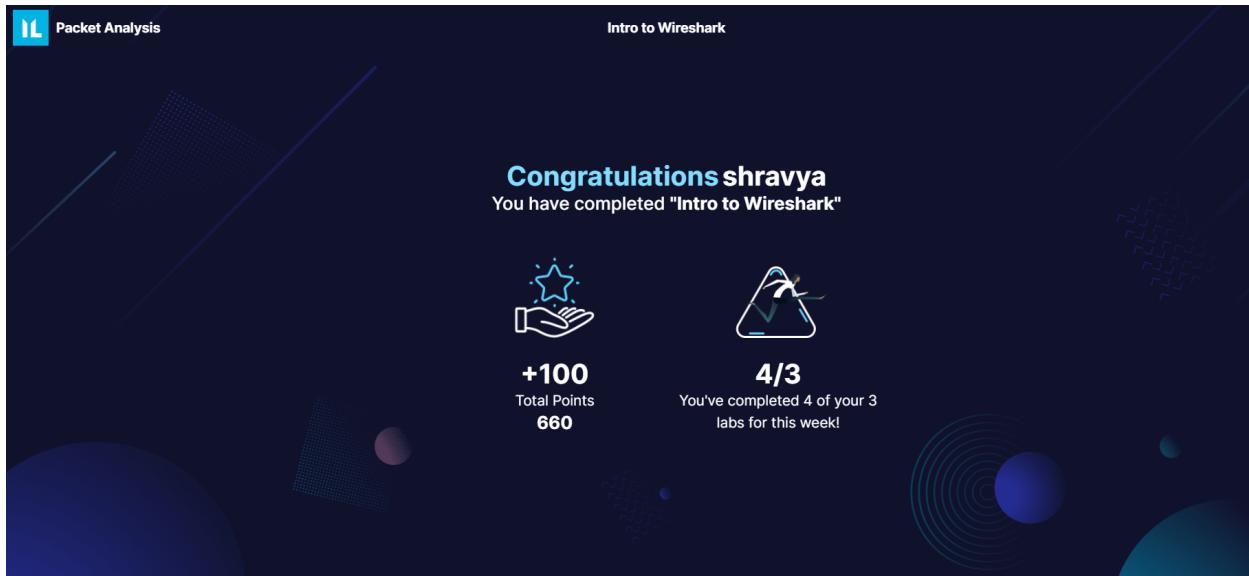


# Assignment - 2

## Lab 1: Wireshark and Network Traffic Analysis



**Tasks**

① What is the difference between resolved and unresolved ports on the Wireshark display setup?

Resolved ports display all information about the port (including destination and header data), whereas unresolved sources only show the raw data.

② Resolved ports display the name of the well-known service that runs on that port, whereas unresolved ports just display the number.

Correct

③ What is the correct syntax to use on Wireshark for showing only SMTP and ICMP traffic?

tcp.port eq 25 or icmp

tcp.show smtp & icmp

tcpdump.list 25

show 25 & icmp

Correct

④ Using wireshark\_setup.pcapng, filter the packets to view only HTTP requests. What is the source IP address shown on the last packet?

172.21.2.217

Correct

⑤ Within that same packet, what is the time shown? Your answer must be in YYYY-MM-DD HH:MM:SS format adjusted for UTC.

2017-12-12 13:04:10

Correct

⑥ What is the destination IP address of the last packet?

34.232.90.203

Check

**Desktop**

Applications The Wireshark Network... wireshark\_setup.pcapng Wireshark - Packet 79924 Tables - File Manager

File Edit View Go Capture Analyze Statistics Telephone Wireless Tools Help

wireshark\_setup.pcapng

Apply a display filter: <Ctrl>-

No.	Time	Source	Destination	Protocol	Length	Info
79903	176.07.04.213	172.21.2.217	185.21.2.216, 248	TCP	54	51844 -> 80 [ACK] Seq:12646 Ack:10957 Win:262144 Len:0
79904	176.07.09.059	172.21.2.217	172.21.2.217	TLSv1.2	126	Application Data
79905	176.07.09.059	172.21.2.217	172.21.2.217	TLSv1.2	126	Application Data
79906	176.07.09.059	172.21.2.217	172.21.2.217	TLSv1.2	126	Application Data
79907	176.07.09.059	172.21.2.217	172.21.2.217	TLSv1.2	126	Application Data
79908	176.07.09.059	172.21.2.217	172.21.2.217	TLSv1.2	126	Application Data
79909	176.07.09.059	172.21.2.217	172.21.2.217	TLSv1.2	126	Application Data
79910	176.07.09.059	172.21.2.217	172.21.2.217	TLSv1.2	126	Application Data
79911	176.07.09.059	172.21.2.217	172.21.2.217	TLSv1.2	126	Application Data
79912	176.07.09.059	172.21.2.217	172.21.2.217	TLSv1.2	126	Application Data
79913	176.07.09.059	172.21.2.217	172.21.2.217	TLSv1.2	126	Application Data
79914	176.07.09.059	172.21.2.217	172.21.2.217	TLSv1.2	126	Application Data
79915	176.09.01.013	172.21.2.217	172.21.2.217	TLSv1.2	567	Application Data
79916	176.09.02.012	172.21.2.217	172.21.2.217	TLSv1.2	93	Application Data
79917	176.09.02.012	172.21.2.217	172.21.2.217	TLSv1.2	93	Application Data
79918	176.09.03.013	172.21.2.217	172.21.2.217	TLSv1.2	100	Application Data
79919	176.09.03.013	172.21.2.217	172.21.2.217	TLSv1.2	100	Application Data
79920	176.09.03.013	172.21.2.217	172.21.2.217	TLSv1.2	100	Application Data
79921	176.10.00.020	172.21.2.217	172.21.2.217	TLSv1.2	103	Application Data
79922	176.10.00.022	172.21.2.217	172.21.2.217	TLSv1.2	100	Application Data
79923	176.10.00.022	172.21.2.217	172.21.2.217	TLSv1.2	100	Application Data
79924	176.10.03.422	172.21.2.217	34.232.90.203	HTTP	899	GET /f/.../event?bigClientId=4078&bigAction=synch&bigSourceId=

Arrival Time: Dec 12, 2017 12:57:54.005441000 UTC  
[Time shift for this packet: 0.000000000 seconds]  
Epoch Time: 1513083474.005441000 Seconds  
Time since previous displayed frame: 0.000000000 seconds  
Time since reference or first frame: 0.000000000 seconds  
Frame ID: 1  
Frame length: 71 bytes (568 bits)  
Capture Length: 71 bytes (568 bits)  
Frame offset: 0 bytes (0 bits)  
Frame is ignored: False  
Protocols in frame with ethertype:ip:udp:dns  
User Datagram Protocol, Src Port: 56199, Dst Port: 53  
Domain Name System (query)

Ethernet II, Src: Wireshark (00:0c:29:5d:24:c1), Dst: Broadcast (ff:ff:ff:ff:ff:ff) [00:10:18:f6:5f:33]  
User Datagram Protocol, Src Port: 56199, Dst Port: 53  
User Datagram Protocol, Src Port: 56199, Dst Port: 53

0000 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 45 86 .3 )\$ .  
0001 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .9 3 .  
0002 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .5 % .  
0003 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0004 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0005 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0006 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0007 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0008 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0009 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0010 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0011 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0012 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0013 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0014 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0015 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0016 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0017 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0018 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0019 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0020 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0021 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0022 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0023 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0024 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0025 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0026 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0027 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0028 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0029 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0030 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0031 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0032 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0033 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0034 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0035 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0036 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0037 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0038 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0039 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0040 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0041 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0042 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0043 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0044 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0045 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0046 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0047 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0048 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0049 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0050 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0051 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0052 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0053 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0054 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0055 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0056 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0057 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0058 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0059 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0060 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0061 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0062 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0063 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0064 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0065 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0066 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0067 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0068 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0069 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0070 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0071 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0072 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0073 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0074 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0075 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0076 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0077 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0078 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0079 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0080 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0081 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0082 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0083 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0084 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0085 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0086 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0087 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0088 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0089 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0090 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0091 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0092 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0093 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0094 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0095 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0096 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0097 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0098 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0099 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0100 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0101 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0102 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0103 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0104 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0105 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0106 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0107 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0108 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0109 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0110 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0111 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0112 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0113 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0114 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0115 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0116 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0117 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0118 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0119 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0120 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0121 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0122 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0123 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0124 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0125 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0126 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0127 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0128 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0129 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0130 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0131 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0132 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0133 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0134 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0135 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0136 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0137 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0138 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0139 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0140 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0141 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0142 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0143 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0144 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0145 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0146 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0147 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0148 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0149 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0150 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0151 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0152 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0153 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0154 00 10 18 f6 5f 33 00 2c 20 5d 24 2c 08 09 ac 15 .  
0155 00 1

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`

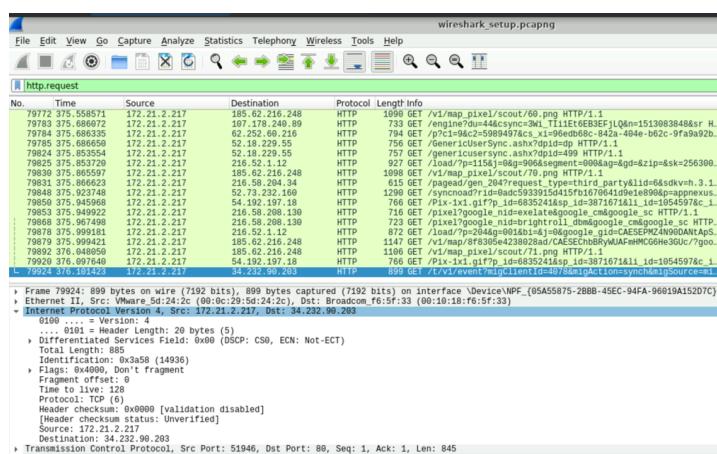
Explanation: In Wireshark, the correct syntax for filtering traffic is crucial. In this case, `tcp.port eq 25` filters for TCP traffic on port 25 (SMTP), while `icmp` filters for ICMP traffic. The logical "or" (`||`) is used to combine these conditions, ensuring that both SMTP and ICMP traffic are displayed.

## 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.

No.	Time	Source	Destination	Protocol	Length	Info
79772	375.558571	172.21.2.217	185.62.216.248	HTTP	1099	/v1/map_pixel/scout/60.png HTTP/1.1
79783	375.686972	172.21.2.217	107.178.240.89	HTTP	733	GET /engine?id=44&csync=3&l_T111ET
79784	375.686335	172.21.2.217	62.252.60.214	HTTP	794	GET /p?cl=9&c=2;5989497&c_xi=96ed6b
79785	375.686656	172.21.2.217	52.18.229.55	HTTP	756	/GenericUserSync.ashx?pid=dp
79824	375.853556	172.21.2.217	52.18.229.55	HTTP	757	/genericusersync.ashx?pid=dp
79825	375.853726	172.21.2.217	220.11.1.12	HTTP	927	GET /load/?p=115&i=0&s=906&se=0
79839	375.853777	172.21.2.217	185.62.216.248	HTTP	1099	/v1/map_pixel/scout/70.png HTTP/1.1
79821	376.866623	172.21.2.217	216.58.204.34	HTTP	615	/pagead/p?en_2047&request_type=
79848	376.923748	172.21.2.217	52.7.232.160	HTTP	1299	GET /syncnoad?rid=0adc5933915d415f
79850	376.945966	172.21.2.217	54.192.197.18	HTTP	766	/pix-1x1.gif?p_id=6835241&p_j
79853	376.949922	172.21.2.217	216.58.208.139	HTTP	716	/pix1?google_nid=exelate&poq
79868	376.967494	172.21.2.217	216.58.208.139	HTTP	723	/pix1?google_nid=brightroll_d
79878	376.999311	172.21.2.217	216.52.1.12	HTTP	872	GET /load/?p=204&g=0&s=1&j=0&gool
79880	376.999421	172.21.2.217	185.62.216.248	HTTP	1104	/v1/map_pixel/scout/71.png HTTP/1.1
79929	376.997640	172.21.2.217	54.192.197.18	HTTP	766	/pix-1x1.gif?p_id=6835241&p_j
L 79924	376.101423	172.21.2.217	34.232.90.203	HTTP	899	GET /l/v1/event/mplclientid=40784

Frame 79924: 899 bytes on wire (7192 bits), 899 bytes captured (7192 bits) on interface \Device\NPF\_{05A55875-2B8B-45EC-94FA-96019A152D7C}  
 Interface id: 0 (\Device\NPF\_{05A55875-2B8B-45EC-94FA-96019A152D7C})  
 Encapsulation type: Ethernet  
 Arrival Time: 2017-12-12 13:04:10.106864000 UTC  
 Time shift for this packet: 0.000000000 seconds  
 Epoch Time: 1513983850.106864000 seconds  
 [Time delta from previous captured frame: 0.001369000 seconds]  
 [Time delta from previous displayed frame: 0.003783000 seconds]  
 [Time since reference or first frame: 376.101423000 seconds]  
 Frame Number: 79924  
 Frame Length: 899 bytes (7192 bits)  
 Capture Length: 899 bytes (7192 bits)  
 [Frame is marked: False]  
 [Frame is ignored: False]

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.

No.	Time	Source	Destination	Protocol	Length	Info
79772	375.558571	172.21.2.217	185.62.216.248	HTTP	1099	/v1/map_pixel/scout/60.png HTTP/1.1
79783	375.686972	172.21.2.217	107.178.240.89	HTTP	733	GET /engine?id=44&csync=3&l_T111ET
79784	375.686335	172.21.2.217	62.252.60.218	HTTP	794	GET /p?cl=9&c=2;5989497&c_xi=96ed6b
79785	375.686656	172.21.2.217	52.18.229.55	HTTP	756	/GenericUserSync.ashx?pid=dp
79824	375.853556	172.21.2.217	52.18.229.55	HTTP	757	/genericusersync.ashx?pid=dp
79825	375.853726	172.21.2.217	220.11.1.12	HTTP	927	GET /load/?p=115&i=0&s=906&se=0
79839	375.853777	172.21.2.217	185.62.216.248	HTTP	1099	GET /v1/map_pixel/scout/70.png HTTP/1.1
79850	376.866623	172.21.2.217	216.58.204.34	HTTP	615	/pagead/p?en_2047&request_type=third_party&id=8&dv=an_3_L
79853	376.945966	172.21.2.217	52.18.229.55	HTTP	766	GET /pix-1x1.gif?p_id=6835241&p_j
79856	376.949922	172.21.2.217	54.192.197.18	HTTP	766	/pix1?google_nid=exelate&poq
79858	376.967494	172.21.2.217	216.58.208.139	HTTP	716	/pix1?google_nid=brightroll_d
79878	376.999311	172.21.2.217	216.52.1.12	HTTP	872	GET /load/?p=204&g=0&s=1&j=0&gool
79879	376.999381	172.21.2.217	185.62.216.248	HTTP	1104	/v1/map_pixel/scout/71.png HTTP/1.1
79880	376.999421	172.21.2.217	185.62.216.248	HTTP	1147	GET /v1/map/8f8395e4288028d/CAESEhBByWUAFHMHGHe3GUc/?gool
79881	376.999440	172.21.2.217	54.192.197.18	HTTP	766	GET /pix-1x1.gif?p_id=6835241&p_j
79929	376.997640	172.21.2.217	54.192.197.18	HTTP	766	/pix1?google_nid=caesePM24N06AN1ApS
L 79924	376.101423	172.21.2.217	34.232.90.203	HTTP	899	GET /l/v1/event/mplclientid=40784

Frame 79924: 899 bytes on wire (7192 bits), 899 bytes captured (7192 bits) on interface \Device\NPF\_{05A55875-2B8B-45EC-94FA-96019A152D7C}  
 Interface id: 0 (\Device\NPF\_{05A55875-2B8B-45EC-94FA-96019A152D7C})  
 Encapsulation type: Ethernet  
 Arrival Time: 2017-12-12 13:04:10.106864000 UTC  
 Time shift for this packet: 0.000000000 seconds  
 Epoch Time: 1513983850.106864000 seconds  
 [Time delta from previous captured frame: 0.001369000 seconds]  
 [Time delta from previous displayed frame: 0.003783000 seconds]  
 [Time since reference or first frame: 376.101423000 seconds]  
 Frame Number: 79924  
 Frame Length: 899 bytes (7192 bits)  
 Capture Length: 899 bytes (7192 bits)  
 [Frame is marked: False]  
 [Frame is ignored: False]

## Lab Writeup:

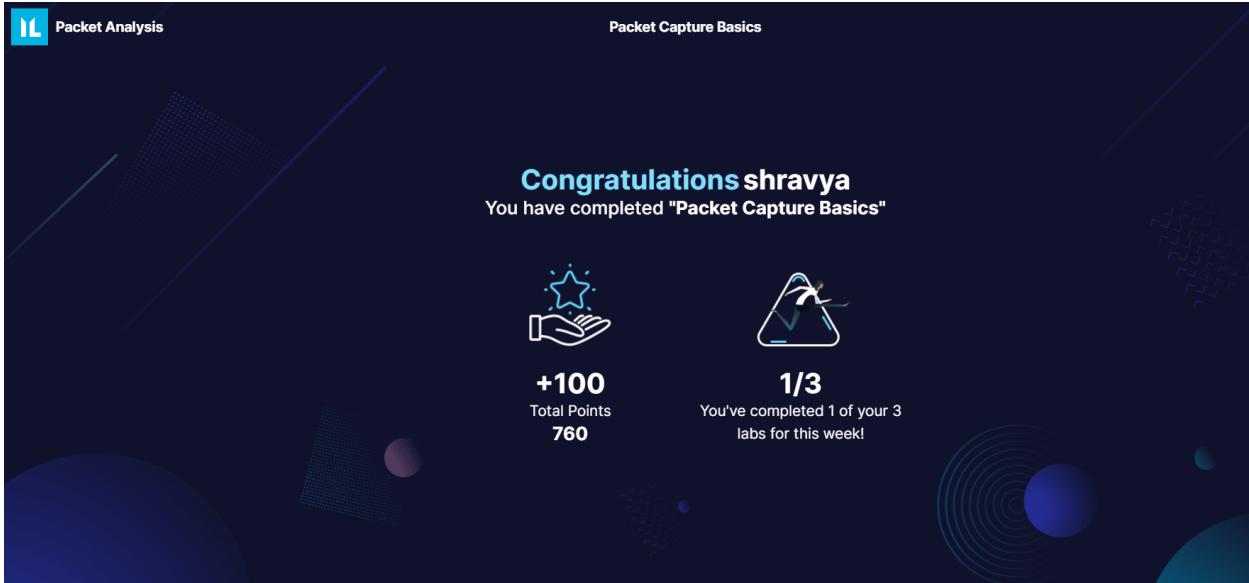
In Lab 1, I had hands-on with Wireshark, a powerful network traffic analysis tool. I delved into the concept of resolved and unresolved ports, discovering how resolved ports facilitate the quick identification of service names based on port numbers. I also practiced crafting Wireshark display filters, 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



**Tasks**

What is issued by the client?

www.bing.com

Correct

What is the first IP address returned in the DNS response for the domain in Q1?

204.79.197.200

Correct

What is the browser user agent string that issued the search request?

Mozilla/5.0 (X11; Linux x86\_64; rv:38.0) Gecko/20100101 Firefox/38.0 Iceweasel/38.7.1

Correct

What web server engine is running the website?

Microsoft-IIS/8.5

Correct

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

How many different IPv4 conversations are there in this capture file?

89

Correct

What was the user searching for on the download.cnet.com website? (Enter your answer as two separate words, e.g., catching fish.)

hacking tools

Check

**Desktop**

Applications: {capture-basics.pcap} [Wireshark - Packet 65] capture-basics.pcap [Wireshark - Packet 1264... capture-basics.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

[http://host == download.cnet.com]

No.	Time	Source	Destination	Protocol	Length	Info
3687	14.14.14.14.479	192.168.0.49	23.1.243.98	HTTP	693	/s/hacking-tools/1 HTTP/2.1
3687	14.14.14.85329	192.168.0.49	23.1.243.98	HTTP	693	/s/hacking-tools/1 HTTP/2.1
3776	22.781936	192.168.0.49	23.1.243.98	HTTP	2142	GET /module/xhr/autosuggest?term=ga HTTP/1.1
3937	22.819374	192.168.0.49	23.1.243.98	HTTP	2159	GET /module/xhr/autosuggest?term=passwordrc HTTP/1.1
3937	22.819374	192.168.0.49	23.1.243.42	HTTP	2159	GET /module/xhr/autosuggest?term=passwordrc HTTP/1.1
4857	25.455047	192.168.0.49	23.1.243.98	HTTP	2148	GET /module/xhr/autosuggest?term=password HTTP/1.1
4872	25.348182	192.168.0.49	23.1.243.98	HTTP	2150	GET /module/xhr/autosuggest?term=passwordrc HTTP/1.1
4872	25.348182	192.168.0.49	23.1.243.98	HTTP	2150	GET /module/xhr/autosuggest?term=passwordrc HTTP/1.1
4152	26.594343	192.168.0.49	23.1.243.98	HTTP	2192	GET /s/password-cracking/ HTTP/1.1

Packets: 5792 · Displayed: 9 (0.2%)

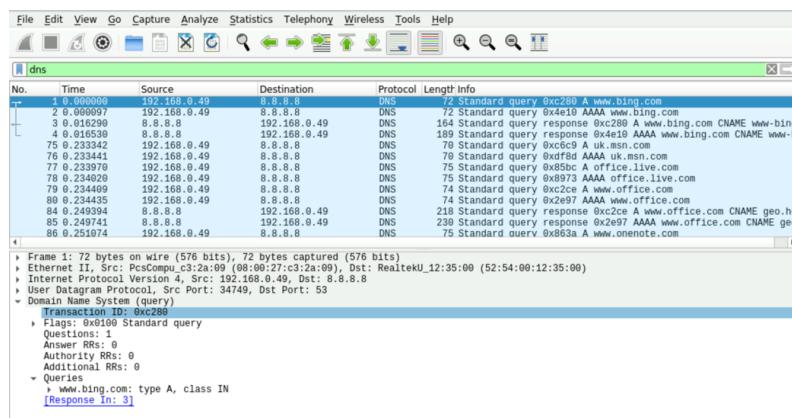
Lab-2 can be divided into 6 tasks as below:

## 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."

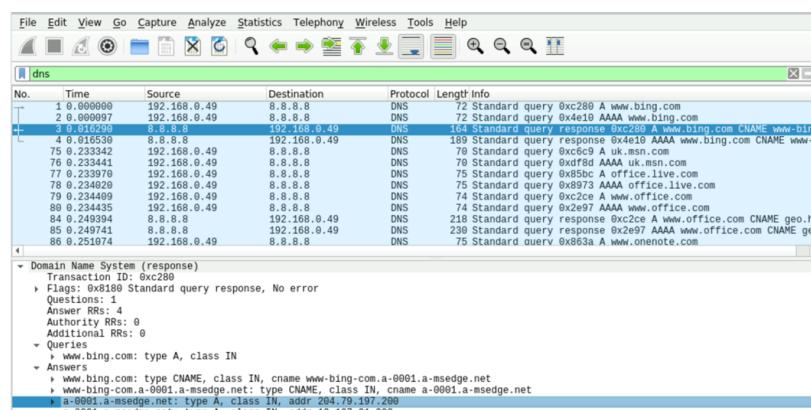
Explanation: In network traffic analysis, DNS (Domain Name System) requests are used to resolve domain names to IP addresses. The client initiated a DNS request seeking the IP address corresponding to "www.bing.com." This indicates a user's intent to access the Bing search engine.



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](http://www.bing.com)".

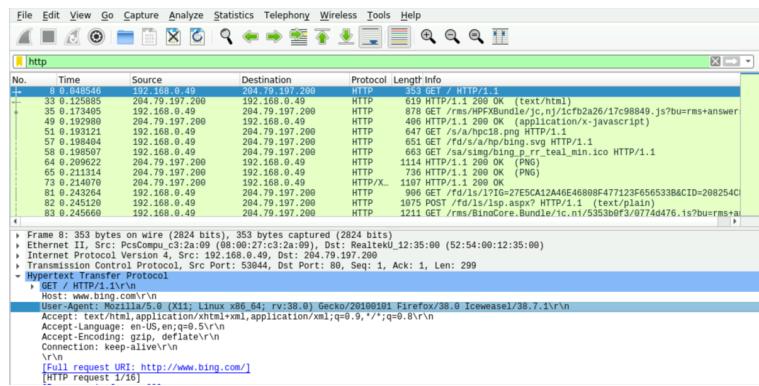


## Task 2: Browser User Agent Analysis

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.)

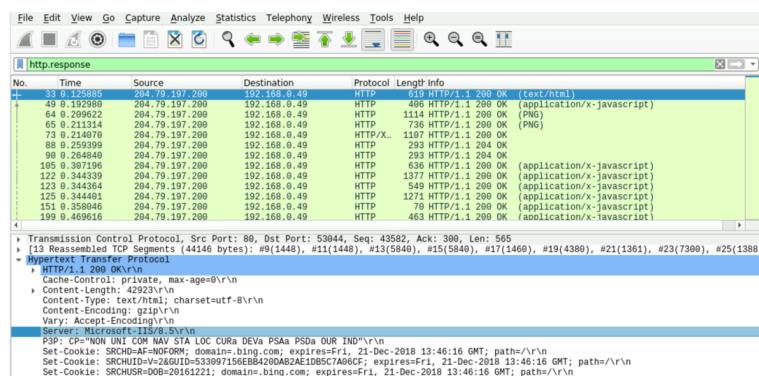


## Task 3: Web Server Analysis

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.)

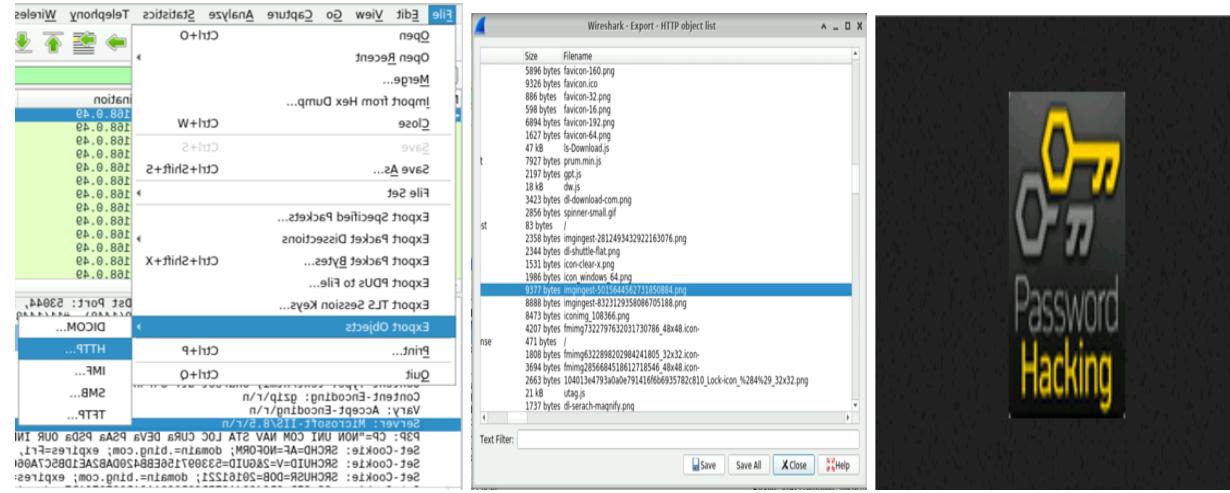


## Task 4: HTTP Content Analysis

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)

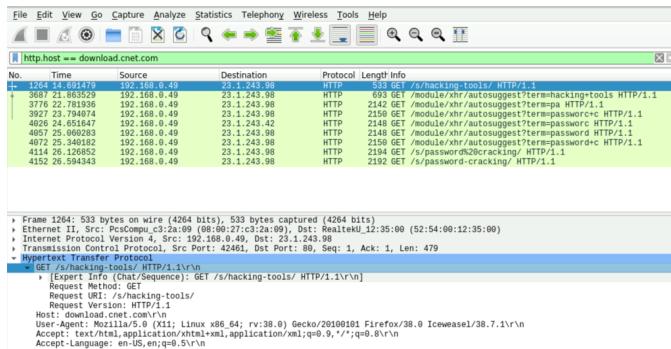
Wireshark - Conversations - capture-basics.pcap										
Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits
2.18.213.98	192.168.0.49	20	4833	8	1912	12	2921	16.252336	28.1259	
8.8.8.8	192.168.0.49	584	74 k	292	51 k	292	22 k	0.000000	37.7351	
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	
23.43.25.27	192.168.0.49	42	9376	19	6765	23	2611	2.331680	42.1077	
23.7.9.11	192.168.0.49	41	35 k	19	5 k	22	2211	20.536267	26.6668	
23.21.152.200	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	139	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's capabilities as Cyber SOC Analysts. Learning outcomes encompassed DNS analysis, user profiling, web server assessment, content scrutiny, conversation tracking, and understanding

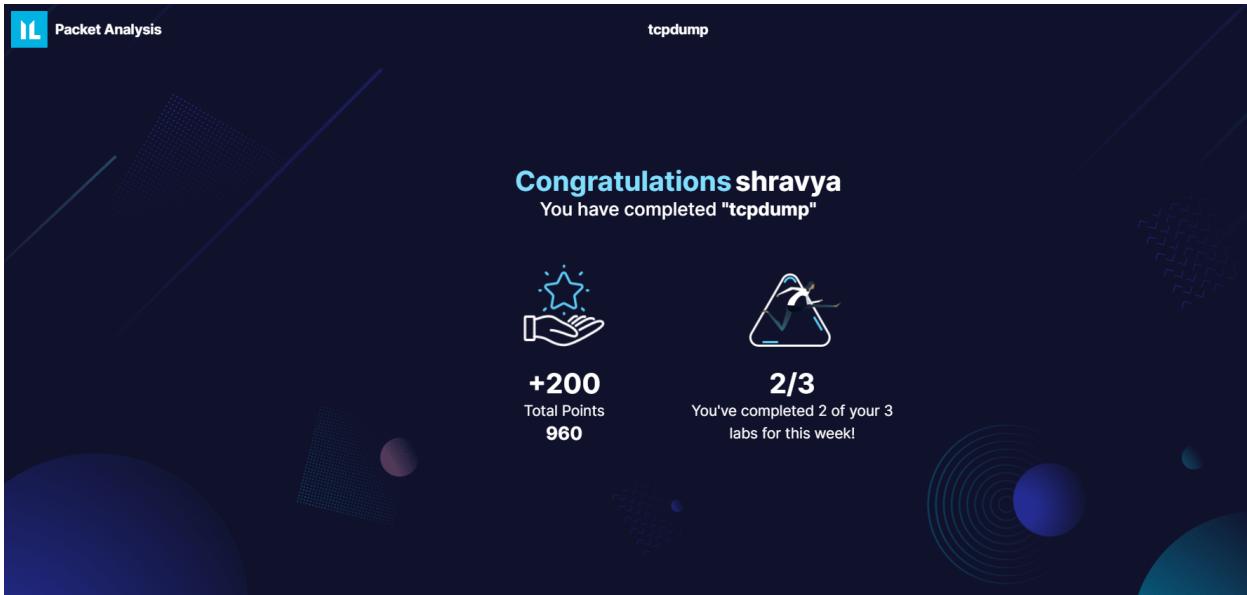
user intent. These skills are invaluable for monitoring, analyzing, and safeguarding network infrastructure.

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	<input type="button" value="Reset"/>
<input checked="" type="checkbox"/> TCPDump	<a href="#">Open &gt;</a>

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].

```
linux@tcpdump:~$ tcpdump -r tcpdump.pcap -w a.pcap
reading from file tcpdump.pcap, link-type EN10MB (Ethernet)
linux@tcpdump:~$ head -3 a.pcap
 fhZ**PVh?
)Y
)Y((?fhZ<<
)YPVPVh?(?
)Y(ghZ91**PVe
)Y
)Y((ghZ2<<
)YPPVe(
)Y(lhZc66PVe
)E(@(XX;P;

5(?wwwgooglecomyhzP
JJPVe
)E<@x?((o5(\dwwwgooglecomyhz
)YPVEL((518,\dwwwgooglecom
ff
)YPVEX((5IDj.?wwwgooglecom
zz
)YPVEL((5:8X\wwwgooglecom
JJPVe
)E<@(
r9
yhz\
)E<@(
`?yhZx
;;PVe){@(
SPr@yhz
)E-@(
SPr@?S7jd?{U?
wx` =tG7_35
@tqw
;+/\o^,0
```

## 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.

```
linux@tcpdump:~$ tcpdump -D
1.eth0 [Up, Running]
2.lo [Up, Running, Loopback]
3.any (Pseudo-device that captures on all interfaces) [Up, Running]
4.bluetooth-monitor (Bluetooth Linux Monitor) [none]
5.nflog (Linux netfilter log (NFLOG) interface) [none]
6.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]
linux@tcpdump:~$ █
```

## 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 6a6c 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 fid9 0083 68ca 9817 3e63 J@:..o....h...>c
0x4310: 9b03 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 ..0=....e[4/.as
0x4340: 5d4a 0177 2044 ae04 a16e 72ce 0a6b ad72 ]J.w.D...nr..k.r
0x4350: 090e 0df0 d620 66ba 8218 le11 3a0a 0b40 .....f.....@.
0x4360: 8041 68ff bbbe 72ba 87a3 c56a f9fb 5d93 .Ah....r.....].
0x4370: 1288 0276 d4e3 5128 f3e1 ae8a c002 dc3d ...v..Q(.....=
0x4380: 514f 9dbf 71ba 3f13 fb79 a4cb 3c9c c067 Q0..q..y..<..g
0x4390: 82f2 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 df49 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..~...
0x4410: 4056 5790 f7a9 8d2e a0e7 f255 4661 93ff @W.....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.0.....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
07:32:16.677955 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
07:32:26.921866 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`.

```
linux@tcpdump:~$  
linux@tcpdump:~$  
linux@tcpdump:~$ tcpdump -r tcpdump.pcap -nn host 184.107.41.72 and port 80 -w new.pcap  
reading from file tcpdump.pcap, link-type EN10MB (Ethernet)  
linux@tcpdump:~$ md5sum new.pcap  
8e4b92724d9034a49cf10f6b147ac482  new.pcap  
linux@tcpdump:~$  
linux@tcpdump:~$
```

### 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.

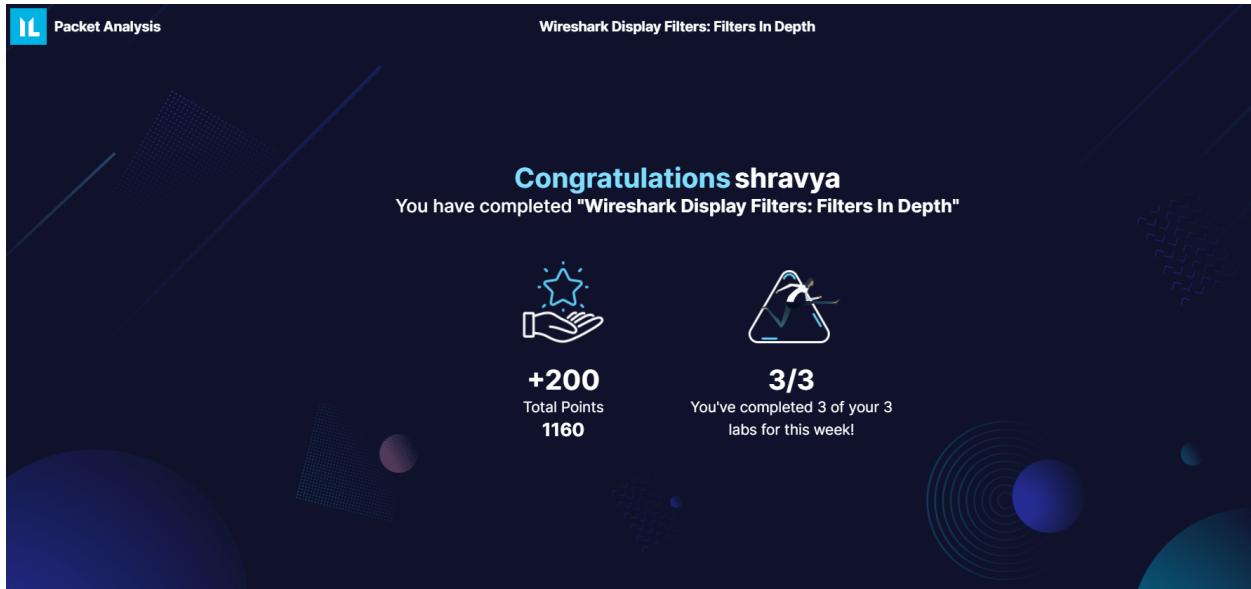
The real essence of network analysis came to the fore when I harnessed tcpdump's filtering capabilities. Filtering packets by IP address and port sharpened one's skills, culminating in MD5sum verification.

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



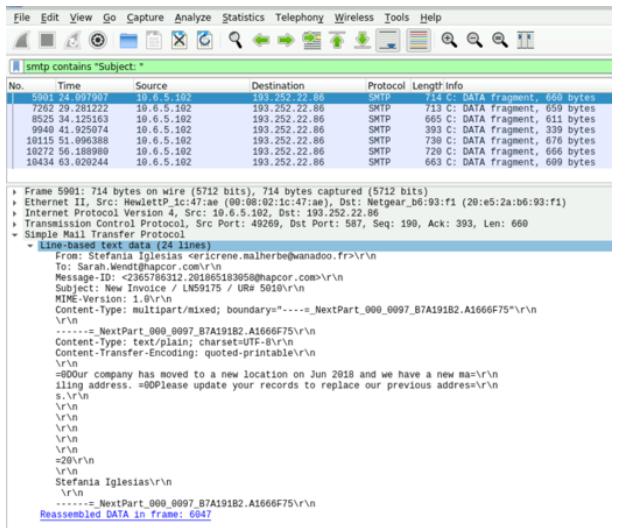
- ③ Analyse and identify the information needed to complete the lab exercise.
- ④ From the PCAP provided, apply a filter which displays all SMTP traffic containing the text "Subject: ". What is the first name of the recipient of that email?  
Sarah  
✓ Correct
- ⑤ From the PCAP provided, apply a filter 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?
- 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."



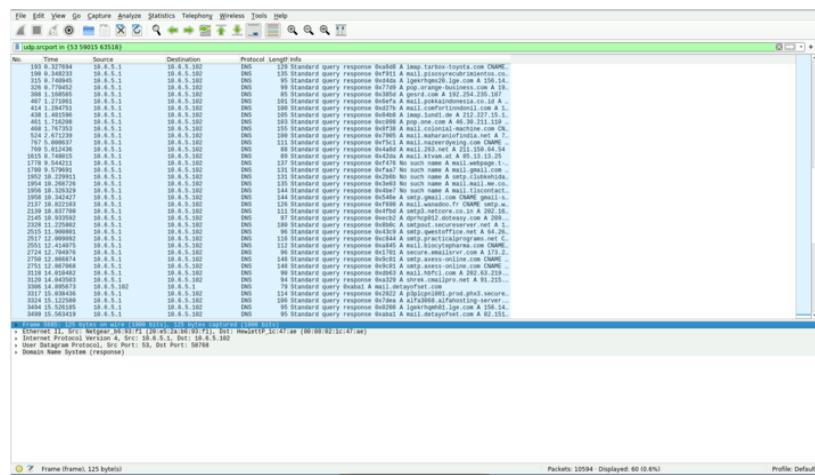
## Task 3: Filtering UDP Packets from Specific Source Ports:

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

**Congratulations shrawya**  
You have completed "BPF Syntax"

**+100**  
Total Points  
**1260**

**4/3**  
You've completed 4 of your 3  
labs for this week!

Tasks	BPF Syntax
① Understand what BPF syntax is and how it's used to filter packets.	<pre>linux@bpf: syntax -s ls bpf-pcap pcapng output.pcap linux@bpf:syntax -s tcpdump -r bpf-pcap.pcapng -nn 'src 10.0.50.227 and tcp port 80' -A   grep -A 5 "GET" reading from file bpf-pcap.pcapng, link-type EN10MB (Ethernet) 11:54:10.078993 IP 10.0.50.227-&gt; 80.252.91.53.80: Flags [P..], seq 0:403, ack 1, win 259, length 403: HTTP: GET /serving/adServer.bs?cn=display&amp;c=195mc HTTP/1.1 Host: Bs.serving-sys.com Connection: keep-alive User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/62.0.3202.94 Safari/537.36 Accept: image/webp,image/*,*/*;q=0.8 DNT: 1  11:54:10.151992 IP 10.0.50.227.61028 &gt; 52.209.216.59.80: Flags [P..], seq 0:385, ack 1, win 259, length 385: HTTP: GET /5/c=10025/camp_int=Advertiser-153172N pressions HTTP/1.1 E..... 2.P..... 2. P.....W..GET /5/c=10025/camp_int=Advertiser-153172N5Campaign-81478085Eimpressions HTTP/1.1 Host: Bcp.crwcdntr.net Connection: keep-alive User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/62.0.3202.94 Safari/537.36 Accept: image/webp,image/*,*/*;q=0.8 DNT: 1  linux@bpf:syntax -s tcpdump -r bpf-pcap.pcapng 'udp port 57190' -nn -ttt   tail -n 1 reading from file bpf-pcap.pcapng, link-type EN10MB (Ethernet) 2017:12:10 10:54:43.010909 10.0.50.227-&gt; 52.209.216.59.443: UDP, length 41 linux@bpf:syntax -s tcpdump -r bpf-pcap.pcapng 'not type 1' -w output.pcap reading from file bpf-pcap.pcapng, link-type EN10MB (Ethernet) linux@bpf:syntax -s nd5sum output.pcap 0942d25b012745422c1719ac26419da6 output.pcap linux@bpf:syntax -s c linux@bpf:syntax -s</pre>
② Use the PCAP located in /home/linux/bpf-pcap.pcapng.	
③ Analyse and identify the information needed to complete the lab exercise using <a href="#">Kali</a> .	
④ What does BPF stand for?	Berkeley Packet Filter
⑤ Berkeley Packet Filter	Correct
⑥ wlan.addr == c5:52:7e:05:68:d & wlan.fc.type_subtype == 0x02. How many primitives are in this expression?	2
⑦ Berkeley Packet Filter	Correct
⑧ 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?	385
⑨ Berkeley Packet Filter	Correct
⑩ Apply a filter to display all UDP packets on port 57190. What is the timestamp of the final packet?	11:54:43
⑪ Berkeley Packet Filter	Correct
⑫ Apply a filter which reads all traffic apart from DNS and TCP, and output this to a file. What is the filesize of this file?	b942d25b012745422c1719ac26419da6
	<input checked="" type="radio"/> Check

## 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.

```
linux@bpf-syntax:~$ tcpcdump -r bpf-pcap.pcapng host 10.0.50.227 and port 80
read from file bpf-pcap.pcapng, link-type EN10MB (Ethernet)
11:54:10.033853 IP ip-10-0-50-227.eu-west-1.compute.internal.61025 > 80.252.91.53.80: Flags [S], seq 933569624, win 64240, options [mss 1460,nop,wscale 8,nop,nop,sackOK], length 0
11:54:10.070639 IP 80.252.91.53.80 > ip-10-0-50-227.eu-west-1.compute.internal.61025: Flags [S.], seq 845627507, ack 933569625, win 8192, options [mss 1355,nop,wscale 8,nop,nop,sackOK], length 0
11:54:10.070744 IP ip-10-0-50-227.eu-west-1.compute.internal.61025 > 80.252.91.53.80: Flags [.], ack 1, win 259, length 0
11:54:10.070993 IP 80.252.91.53.80 > ip-10-0-50-227.eu-west-1.compute.internal.61025 > 80.252.91.53.80: Flags [P.], seq 1:404, ack 1:404, win 259, length 403: HTTP: GET /server/adServer.bsf?cmdisplayc=195mc=impSpl+23831146PuID@0&ord=1513166932&rtrv=1 HTTP/1.1
11:54:10.085647 IP 80.252.91.53.80 > ip-10-0-50-227.eu-west-1.compute.internal.61025: Flags [.], ack 404, win 8040, length 0
11:54:10.126623 IP ip-10-0-50-227.eu-west-1.compute.internal.61025 > 80.252.91.53.80: Flags [.], ack 724, win 256, length 723: HTTP: HTTP/1.1 302 Redirect
11:54:10.127068 IP ip-10-0-50-227.eu-west-1.compute.internal.61026 > ec2-52-209-216-59.eu-west-1.compute.amazonaws.com:80: Flags [S.], seq 3468476541, win 64240, options [mss 1460,nop,wscale 8,nop,nop,sackOK], length 0
11:54:10.150996 IP ec2-52-209-216-59.eu-west-1.compute.amazonaws.com:80 > ip-10-0-50-227.eu-west-1.compute.internal.61026: Flags [S.], seq 109803271, ack 3468476542, win 64240, options [mss 1460,nop,wscale 8,nop,nop,sackOK], length 0
11:54:10.151241 IP ip-10-0-50-227.eu-west-1.compute.internal.61026 > ec2-52-209-216-59.eu-west-1.compute.amazonaws.com:80: Flags [.], ack 1, win 259, length 0
11:54:10.151992 IP ip-10-0-50-227.eu-west-1.compute.internal.61026 > ec2-52-209-216-59.eu-west-1.compute.amazonaws.com:80: Flags [P.], seq 1:386, ack 1, win 259, length 385: HTTP: GET /<1025/>camp_intAdvertiser-15317245Campaign-814780n5EImpressions HTTP/1.1
11:54:10.194085 IP ip-10-0-50-227.eu-west-1.compute.internal.61026 > 80.252.91.53.80: Flags [.], ack 306, win 110, length 0
11:54:10.194085 IP ip-10-0-50-227.eu-west-1.compute.internal.61026 > 80.252.91.53.80: Flags [P.], seq 1:384, ack 386, win 110, length 0: HTTP/1.1 200 OK
11:54:10.235280 IP ip-10-0-50-227.eu-west-1.compute.internal.61026 > ec2-52-209-216-59.eu-west-1.compute.amazonaws.com:80: Flags [.], ack 304, win 258, length 0
11:54:21.177466 IP 80.252.91.53.80 > ip-10-0-50-227.eu-west-1.compute.internal.61025: Flags [R.], seq 724, ack 404, win 8040, length 0
linux@bpf-syntax:~$
```

## 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.

```
Linux@bpf-syntax:~$ 
Linux@bpf-syntax:~$ tcpdump -r bpf-pcap.pcapng udp port 57190
reading from file bpf-pcap.pcapng, link-type EN10MB (Ethernet)
11:54:42.294320 IP ip-10-0-50-227.eu-west-1.compute.internal.57190 > lhr48s21-in-f3.le100.net.443: UDP, length 1350
11:54:42.294662 IP ip-10-0-50-227.eu-west-1.compute.internal.57190 > lhr48s21-in-f3.le100.net.443: UDP, length 520
11:54:42.308853 IP lhr48s21-in-f3.le100.net.443 > ip-10-0-50-227.eu-west-1.compute.internal.57190: UDP, length 1350
11:54:42.309230 IP ip-10-0-50-227.eu-west-1.compute.internal.57190 > lhr48s21-in-f3.le100.net.443: UDP, length 41
11:54:42.309358 IP ip-10-0-50-227.eu-west-1.compute.internal.57190 > lhr48s21-in-f3.le100.net.443: UDP, length 35
11:54:42.309860 IP lhr48s21-in-f3.le100.net.443 > ip-10-0-50-227.eu-west-1.compute.internal.57190: UDP, length 31
11:54:42.316628 IP lhr48s21-in-f3.le100.net.443 > ip-10-0-50-227.eu-west-1.compute.internal.57190: UDP, length 30
11:54:42.335877 IP ip-10-0-50-227.eu-west-1.compute.internal.57190 > lhr48s21-in-f3.le100.net.443: UDP, length 38
11:54:42.336582 IP lhr48s21-in-f3.le100.net.443 > ip-10-0-50-227.eu-west-1.compute.internal.57190: UDP, length 244
11:54:42.337727 IP lhr48s21-in-f3.le100.net.443 > ip-10-0-50-227.eu-west-1.compute.internal.57190: UDP, length 16
11:54:42.337865 IP ip-10-0-50-227.eu-west-1.compute.internal.57190 > lhr48s21-in-f3.le100.net.443: UDP, length 38
11:54:42.345257 IP lhr48s21-in-f3.le100.net.443 > ip-10-0-50-227.eu-west-1.compute.internal.57190: UDP, length 31
11:54:42.371194 IP ip-10-0-50-227.eu-west-1.compute.internal.57190 > lhr48s21-in-f3.le100.net.443: UDP, length 35
11:54:42.379659 IP ip-10-0-50-227.eu-west-1.compute.internal.57190 > lhr48s21-in-f3.le100.net.443: UDP, length 340
11:54:43.796220 IP ip-10-0-50-227.eu-west-1.compute.internal.57190 > lhr48s21-in-f3.le100.net.443: UDP, length 24
11:54:43.797453 IP lhr48s21-in-f3.le100.net.443 > ip-10-0-50-227.eu-west-1.compute.internal.57190: UDP, length 42
11:54:43.805885 IP lhr48s21-in-f3.le100.net.443 > ip-10-0-50-227.eu-west-1.compute.internal.57190: UDP, length 58
11:54:43.805891 IP lhr48s21-in-f3.le100.net.443 > ip-10-0-50-227.eu-west-1.compute.internal.57190: UDP, length 16
11:54:43.808109 IP ip-10-0-50-227.eu-west-1.compute.internal.57190 > lhr48s21-in-f3.le100.net.443: UDP, length 41
Linux@bpf-syntax:~$
```

## 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.

```
Linux@bpf-syntax:~$ 
Linux@bpf-syntax:~$ tcpdump -r bpf-pcap.pcapng -w name.pcap 'not (port 53 or tcp)'
reading from file bpf-pcap.pcapng, link-type EN10MB (Ethernet)
Linux@bpf-syntax:~$ md5sum name.pcap
b942d25b012745422c1719ac26419da6  name.pcap
Linux@bpf-syntax:~$
```

## 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.