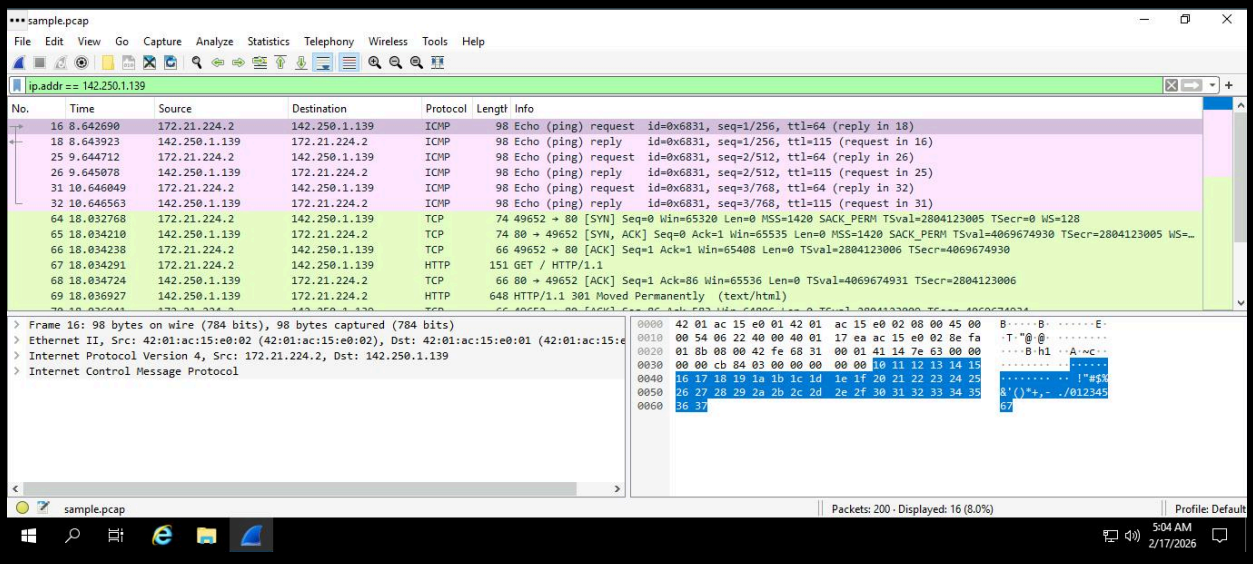
ANALYZING MY FIRST PACKET

A LAB BY DARNELL MAIDEN



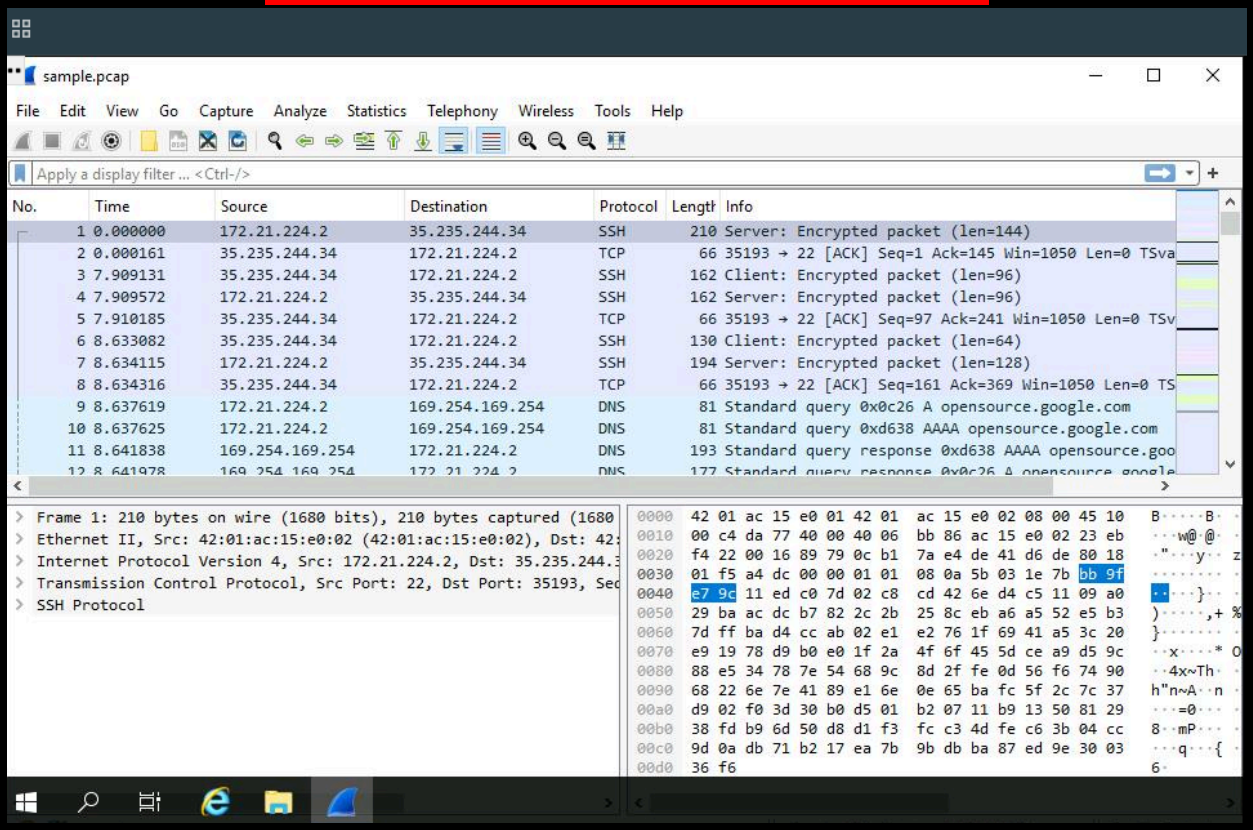
[SUMMARY]:

AFTER GETTING MY BADGE FOR ENTRY, MY ONBOARDING PAPER WORK, AND A FLIRTATIOUS GOODBYE FROM DELORIS FROM THE HR DEPT. (YEAH, REAL CONUDRUM RIGHT?), I FINALLY GOT MY FIRST ASSIGNMENT. THE LEAD WANTED ME TO ANALYZE PACKETS TO TEST MY ANALYST SKILLS AND SEE WHERE I FIT IN THE COMPANY. SO I SAT AT MY NEW DESK, OPENED UP MY WINDOWS VIRTUAL MACHINE, AND GOT TO WORK.

[OPENING WIRESHARK]:

I OPENED UP MY [VIRTUAL MACHINE] AND FOUND THE SAMPLE PCAP FILE THEY SENT TO ME A FEW MINUTES AGO. I CLICK ON IT AND IT DISPLAYS THE DATA FROM A SYSTEM THAT MADE WEB REQUESTS TO A SITE. I OPENED UP [WIRESHARK] TO DISPLAY THE DATA.

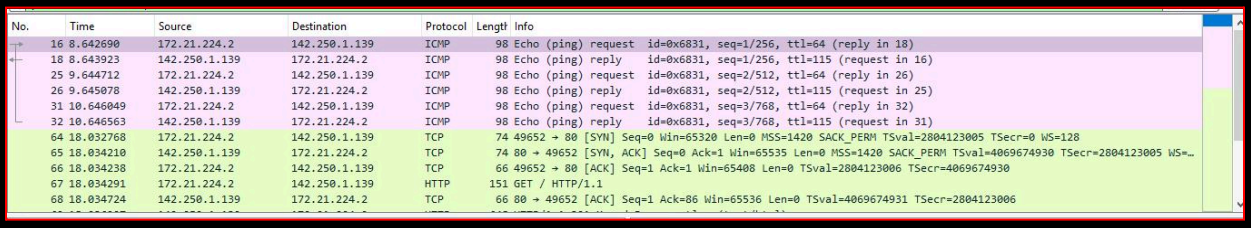
[ANALYZE THE DATA]:



ONCE [WIRESHARK] WAS OPEN I TOOK A MOMENT TO REFRESH MYSELF ON THE COMPONENTS OF EACH PACKET, WHICH IS:

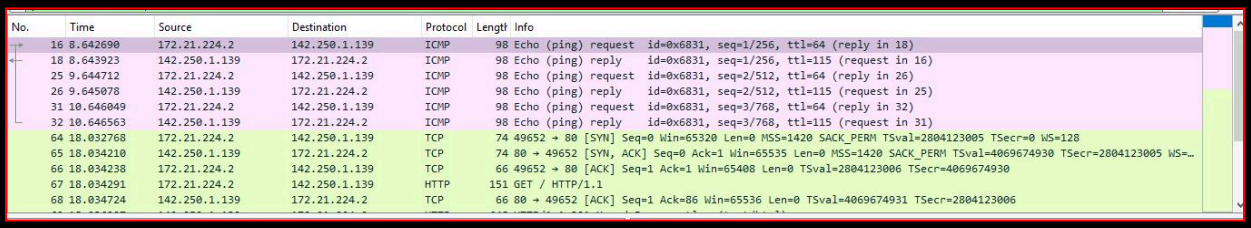
- NO.: THE INDEX OF THE NUMBER PACKET IN THIS PACKET CAPTURE FILE.
- TIME: THE TIMESTAMP OF THE PACKET.
- SOURCE: THE SOURCE IP ADDRESS.

- **DESTINATION:** THE DESTINATION IP ADDRESS.
- **PROTOCOL:** THE PROTOCOL CONTAINED IN THE PACKET
- **LENGTH:** THE TOTAL LENGTH OF THE PACKET
- **INFO:** SOME INFORMATION ABOUT THE DATA IN THE PACKET (THE PAYLOAD) . AS INTERPRETED BY WIRESHARK.



No.	Time	Source	Destination	Protocol	Length	Info
16	8.642690	172.21.224.2	142.250.1.139	ICMP	98	Echo (ping) request id=0x6831, seq=1/256, ttl=64 (reply in 18)
18	8.643923	142.250.1.139	172.21.224.2	ICMP	98	Echo (ping) reply id=0x6831, seq=1/256, ttl=115 (request in 16)
25	9.644712	172.21.224.2	142.250.1.139	ICMP	98	Echo (ping) request id=0x6831, seq=2/512, ttl=64 (reply in 26)
26	9.645078	142.250.1.139	172.21.224.2	ICMP	98	Echo (ping) reply id=0x6831, seq=2/512, ttl=115 (request in 25)
31	10.646049	172.21.224.2	142.250.1.139	ICMP	98	Echo (ping) request id=0x6831, seq=3/768, ttl=64 (reply in 32)
32	10.646563	142.250.1.139	172.21.224.2	ICMP	98	Echo (ping) reply id=0x6831, seq=3/768, ttl=115 (request in 31)
64	18.032768	172.21.224.2	142.250.1.139	TCP	74	49652 → 80 [SYN] Seq=0 Win=65320 Len=0 MSS=1420 SACK_PERM TSval=2804123005 TSecr=0 WS=128
65	18.034210	142.250.1.139	172.21.224.2	TCP	74	80 → 49652 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1420 SACK_PERM TSval=4069674930 TSecr=2804123005 WS=…
66	18.034238	172.21.224.2	142.250.1.139	TCP	66	49652 → 80 [ACK] Seq=1 Ack=1 Win=65408 Len=0 TSval=2804123006 TSecr=4069674930
67	18.034291	172.21.224.2	142.250.1.139	HTTP	151	GET / HTTP/1.1
68	18.034724	142.250.1.139	172.21.224.2	TCP	66	80 → 49652 [ACK] Seq=1 Ack=86 Win=65536 Len=0 TSval=4069674931 TSecr=2804123006

PACKETS COME IN SPECIFIC COLORS TO CLASSIFY THE SPECIFIC TYPES OF DATA. FOR EXAMPLE, [LIGHT GREEN] CONTAINS A MIXTURE OF [TCP] AND [HTTP] TRAFFIC AND [LIGHT PURPLE] CONTAINS [ICMP] TRAFFIC.



No.	Time	Source	Destination	Protocol	Length	Info
16	8.642690	172.21.224.2	142.250.1.139	ICMP	98	Echo (ping) request id=0x6831, seq=1/256, ttl=64 (reply in 18)
18	8.643923	142.250.1.139	172.21.224.2	ICMP	98	Echo (ping) reply id=0x6831, seq=1/256, ttl=115 (request in 16)
25	9.644712	172.21.224.2	142.250.1.139	ICMP	98	Echo (ping) request id=0x6831, seq=2/512, ttl=64 (reply in 26)
26	9.645078	142.250.1.139	172.21.224.2	ICMP	98	Echo (ping) reply id=0x6831, seq=2/512, ttl=115 (request in 25)
31	10.646049	172.21.224.2	142.250.1.139	ICMP	98	Echo (ping) request id=0x6831, seq=3/768, ttl=64 (reply in 32)
32	10.646563	142.250.1.139	172.21.224.2	ICMP	98	Echo (ping) reply id=0x6831, seq=3/768, ttl=115 (request in 31)
64	18.032768	172.21.224.2	142.250.1.139	TCP	74	49652 → 80 [SYN] Seq=0 Win=65320 Len=0 MSS=1420 SACK_PERM TSval=2804123005 TSecr=0 WS=128
65	18.034210	142.250.1.139	172.21.224.2	TCP	74	80 → 49652 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1420 SACK_PERM TSval=4069674930 TSecr=2804123005 WS=…
66	18.034238	172.21.224.2	142.250.1.139	TCP	66	49652 → 80 [ACK] Seq=1 Ack=1 Win=65408 Len=0 TSval=2804123006 TSecr=4069674930
67	18.034291	172.21.224.2	142.250.1.139	HTTP	151	GET / HTTP/1.1
68	18.034724	142.250.1.139	172.21.224.2	TCP	66	80 → 49652 [ACK] Seq=1 Ack=86 Win=65536 Len=0 TSval=4069674931 TSecr=2804123006

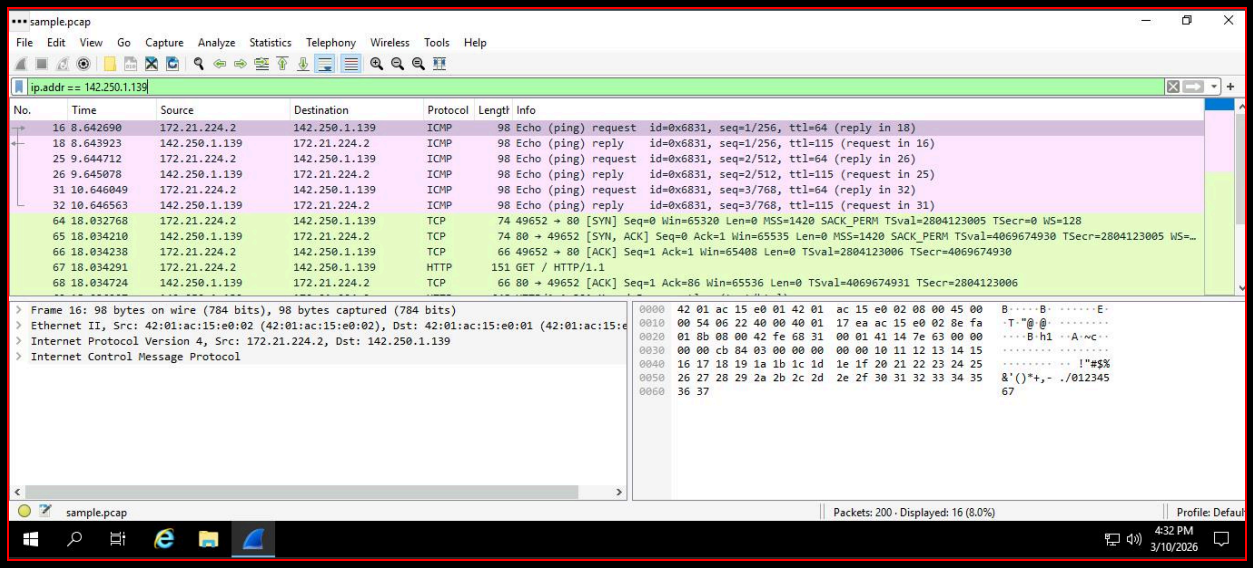
IF YOU LOOK CLOSELY AT THE FIRST PACKET AT THE TOP, IT BELONGS TO A GROUP OF ICMP TRAFFIC.

ICMP (INTERNET CONTROL MESSAGE PROTOCOL) IS A NETWORK LAYER PROTOCOL USED BY ROUTERS AND DEVICES TO SEND ERROR MESSAGES AND OPERATIONAL INFORMATION.

[FILTER AND INSPECT]:

AFTER GETTING A REFRESHER ON THE COMPONENTS AND DATA LAYOUTS I WENT TO THE “APPLY A DISPLAY FILTER” BOX AND TYPED IN A IP ADDRESS THAT WAS GIVEN TO ME BY THE COMPANY.

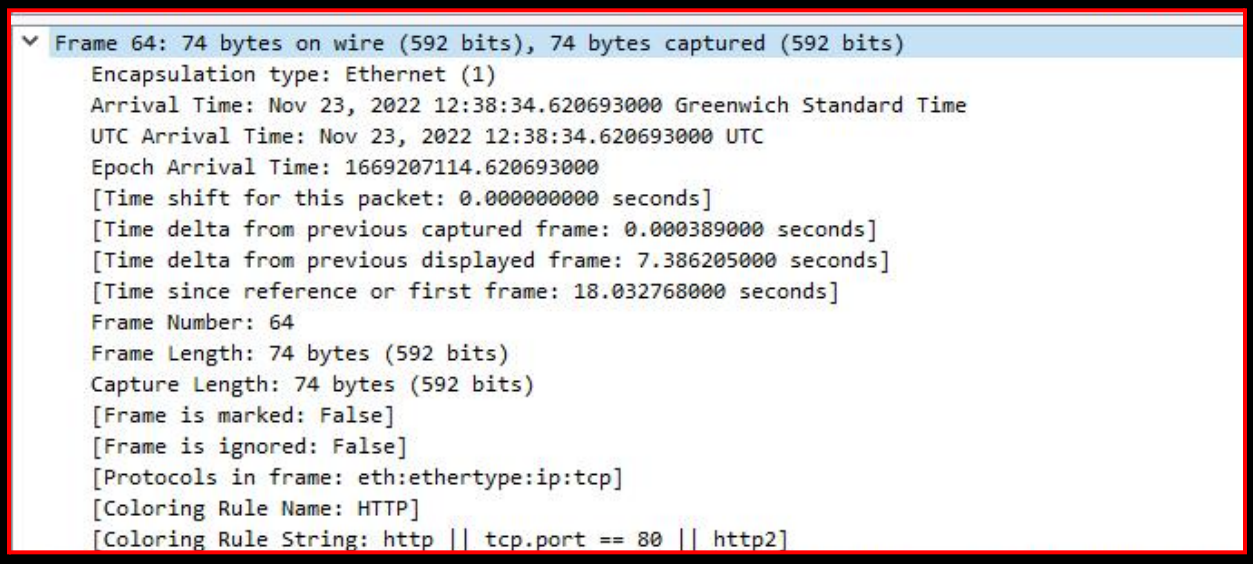
[IP.ADDR == 142.250.1.139]



AFTER ENTERING MY SEARCH I HEADED TO THE FIRST TCP ENTRY FOR THE IP ADDRESS GIVEN AND EXAMINED IT CAREFULLY USING THE BOTTOM LEFT WINDOW. THESE ARE THE COMPONENTS:

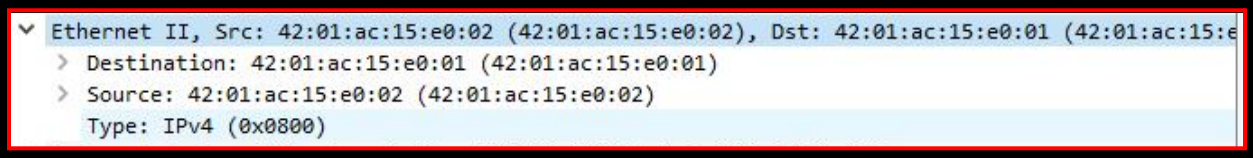
[FRAME]:

THIS IS THE FRAME. THESE ARE THE OVERALL DETAILS OF THE PACKET SUCH AS THE FRAME RATE AND ARRIVAL TIME OF THE PACKET ALONG WITH THE ENTIRETY OF THE PACKET DATA PRESENTED.



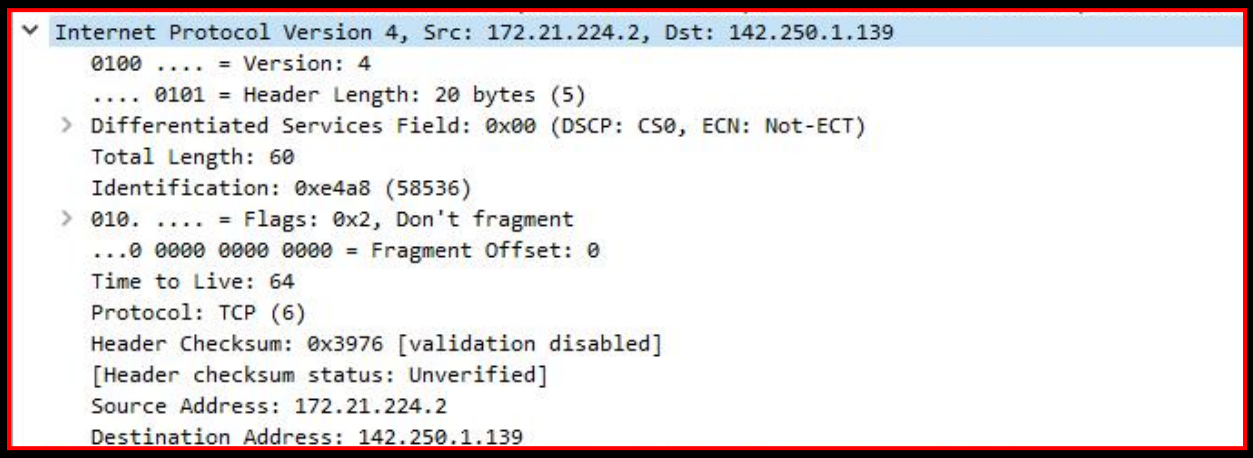
[ETHERNET II]:

THIS SECTION CONTAINS THE DETAILS ABOUT THE PACKET AT THE ETHERNET LEVEL, INCLUDING THE SOURCE AND DESTINATION MAC ADDRESSES AND THE TYPE OF INTERNAL PROTOCOL THAT THE ETHERNET PACKET CONTAINS.



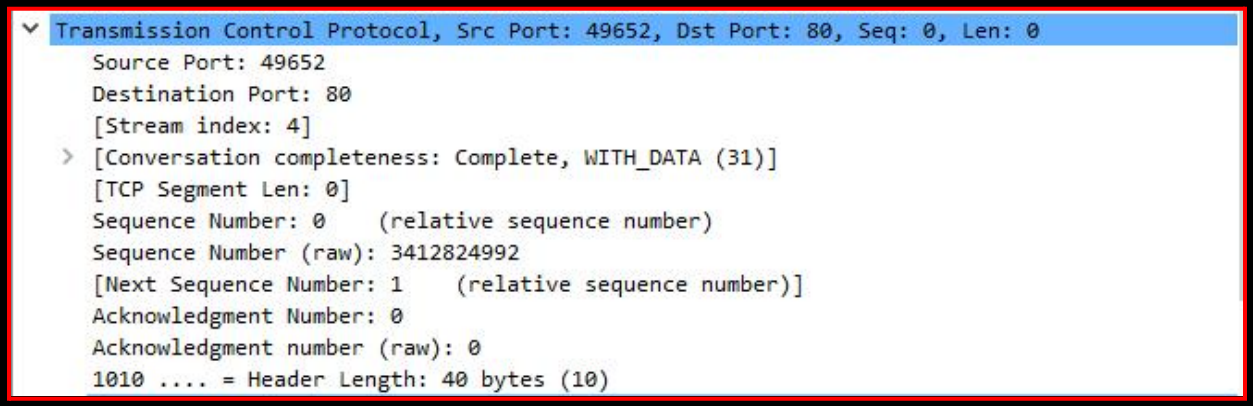
[IPV4]:

THIS IS THE IP DATA SECTION WHICH CONTAINS INFORMATION ABOUT THE TIME TO LIVE, THE PROTOCOL, SOURCE ADDRESS, AND THE DESTINATION ADDRESS AS WELL.



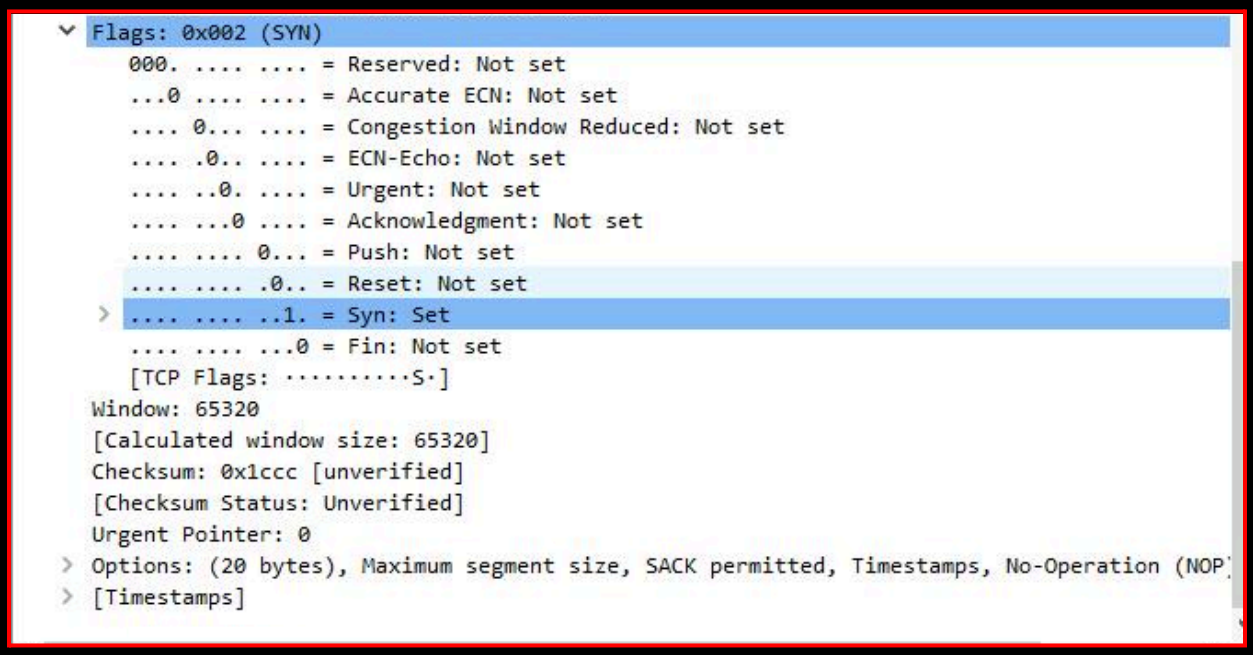
[PROTOCOL INFORMATION]:

THIS SECTION CONTAINS PROTOCOL INFORMATION FOR THE PACKET YOU'RE INSPECTING SUCH AS THE SOURCE PORTS, DESTINATION PORTS, ETC.



[FLAG INFORMATION]:

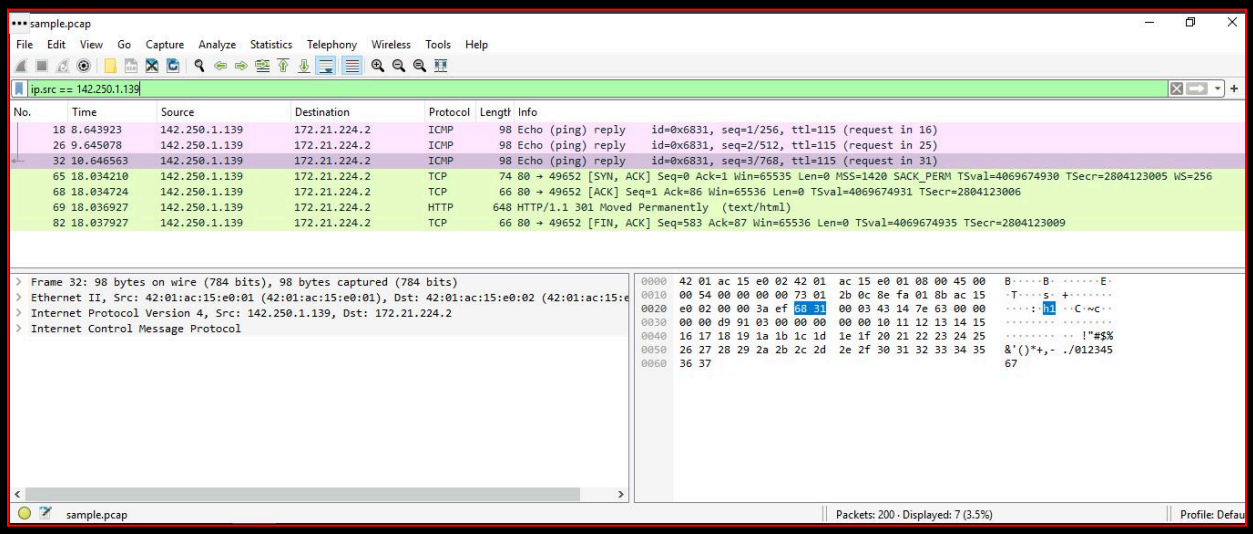
THIS SECTION DISPLAYS THE PROTOCOL FLAG INFORMATION.



[FILTER AND SELECT]:

MY NEXT ASSIGNMENT IS ANALYZING SPECIFIC NETWORK PACKETS BASED ON WHERE THEY CAME FROM OR WHERE THEY WERE SENT TO. I'LL BE SELECTING PACKETS BASED ON THEIR PHYSICAL ETHERNET MEDIA ACCESS CONTROL (MAC) ADDRESS OR THEIR INTERNET PROTOCOL (IP) ADDRESS. LET'S TAKE THIS IP SOURCE HERE FOR EXAMPLE.

[IP.SRC == 142.250.1.139]

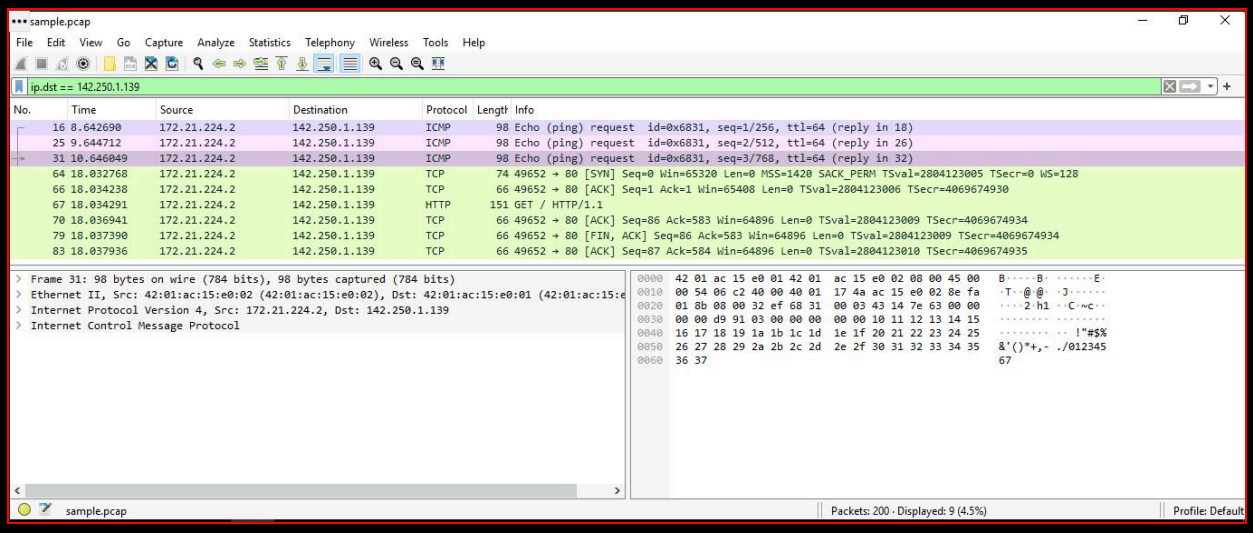


I HAVE THE OVERVIEW OF EVERY ENTRY THAT CAME FROM 142.250.1.139

I CAN USE THIS INFORMATION FOR ANY ANALYSIS , SUCH AS UNUSUAL SOURCE TRAFFIC.

NOW LET'S FILTER THAT SAME IP FOR THE DESTINATION TRAFFIC.

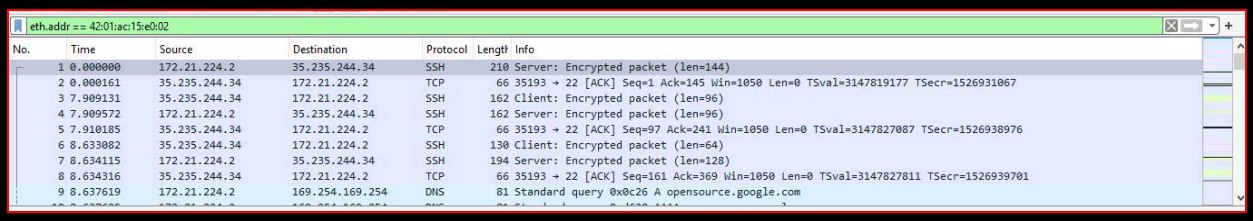
[IP.DST == 142.250.1.139]



NOW IT CONTAINS ALL THE PACKETS THAT WE'RE SENT TO 142.250.1.139

NOW I WANT TO FILTER AND SELECT TRAFFIC TO OR FROM A SPECIFIC ETHERNET MAC ADDRESS

[ETH.ADDR == 42:01:AC:15:E0:02]

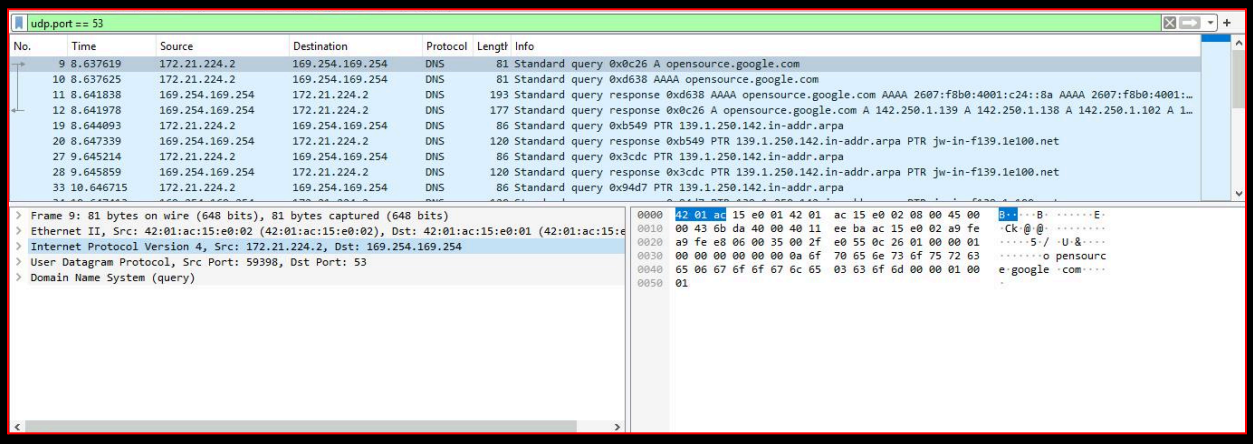


EVERY ENTRY LISTED HERE IS RELATED TO THE ETHERNET MAC ADDRESS. THIS COULD BE USED FOR FURTHER ANALYSIS AS WELL.

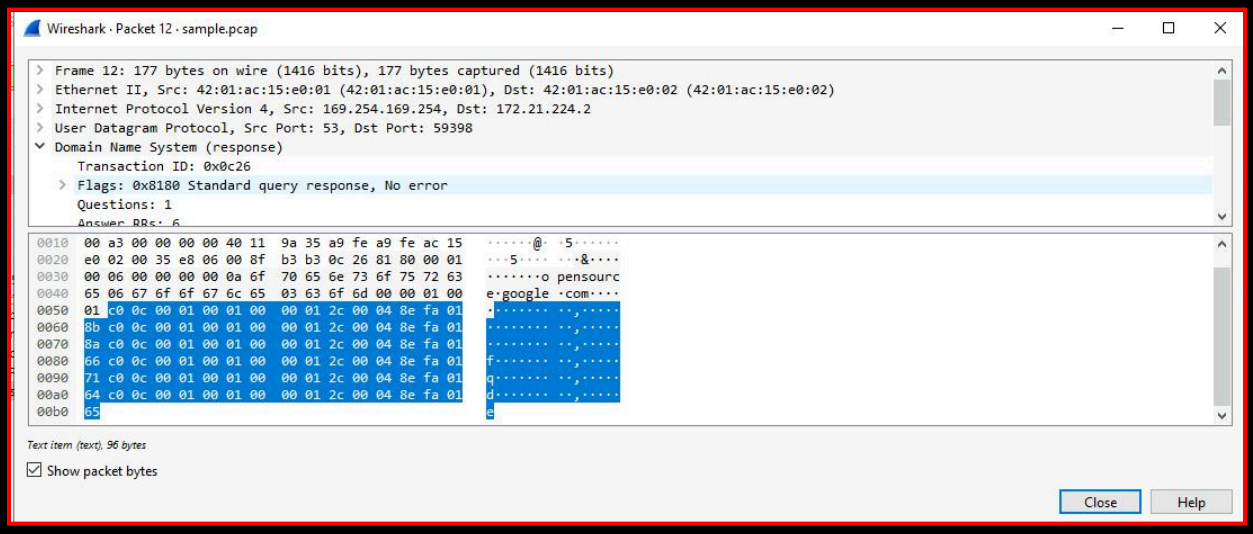
[DNS EXPLORATION]

I NEEDED TO FIND WHAT SITES PEOPLE WERE LOOKING UP ALONG WITH THE ANSWERS, WHICH ARE THE IP ADDRESSES THAT ARE BEING SENT BACK BY THE DNS SERVER WHEN A NAME IS SUCCESSFULLY RESOLVED. MY LEAD TOLD ME TO EXAMINE THIS PORT FOR FURTHER ANALYSIS.

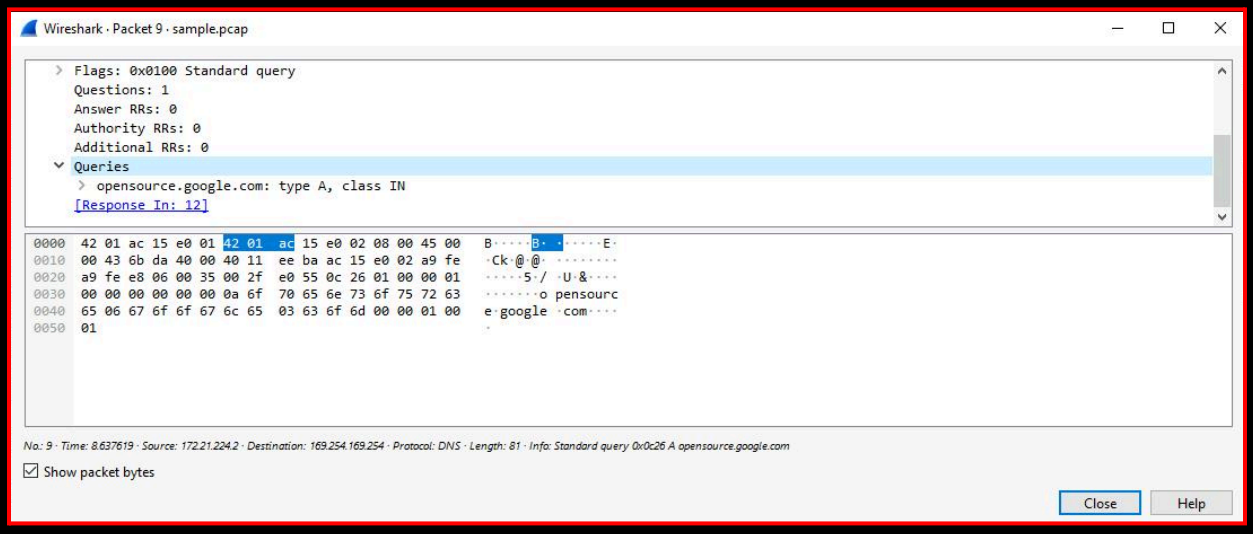
[UDP.PORT == 53]

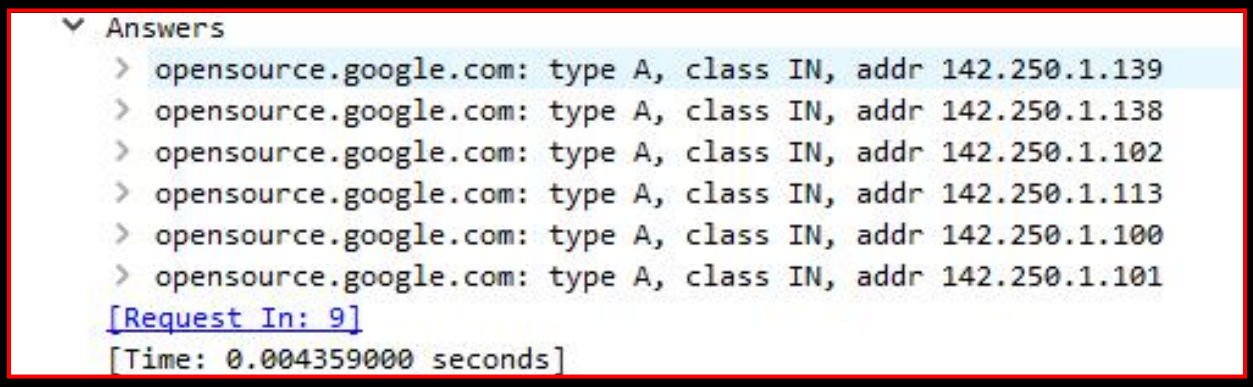


THIS IS A GOOD START BUT I NEED THE TO KNOW WHERE THE SITE IS, SO I DOUBLE CLICK THE DOMAIN NAME SYSTEM (QUERY) , THEN CLICK THE QUERIES SECTION...



....AND VOLIA, THE ANSWERS DATA SHOWS ME THAT THE QUERIED NAME WAS OPENSOURCE.GOOGLE.COM AND THE IP ADDRESS WAS 142.250.1.139

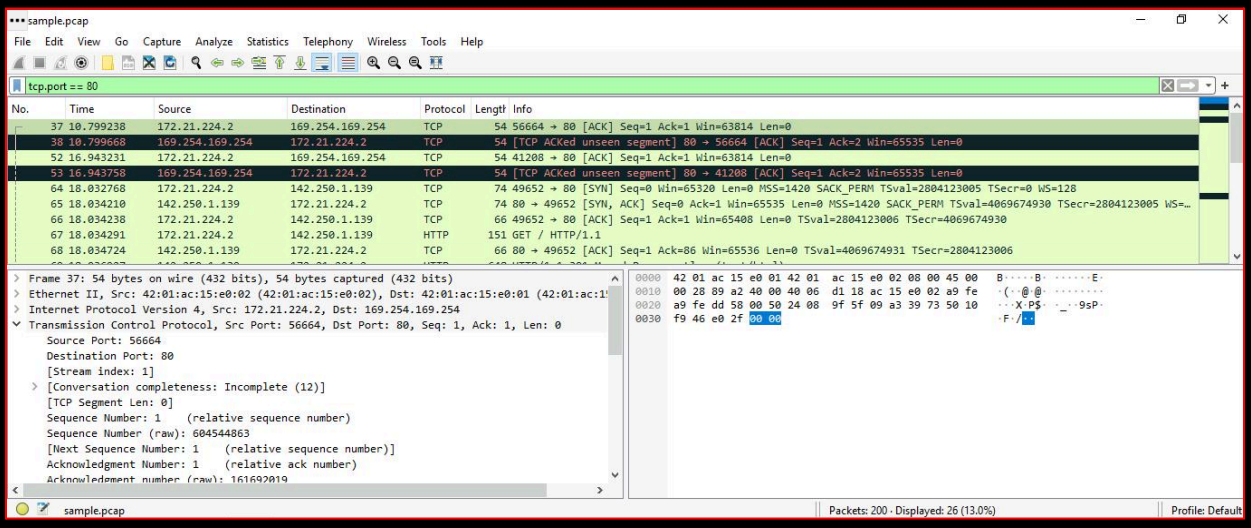




[TCP EXPLORATION]

THAT DNS DATA DIDN'T SIT RIGHT WITH ME. SO I DID MORE DIGGING AND FILTERED THROUGH TCP PACKETS AS WELL. MY LEAD GAVE ME THIS TO SEARCH FOR:

TCP.PORT == 80



A FEW PACKETS WERE MADE AFTER THE USER ACCESSED THE TARGET SITE. I CLICKED THE FIRST PACKET TO EXAMINE THE COMPONENTS OF IT.

- THE DESTINATION IP: [169.254.169.254]
- THE SOURCE IP: [172.21.224.2]
- THE TIME TO LIVE VALUE: [64]
- HEADER LENGTH: [20 BYTES]

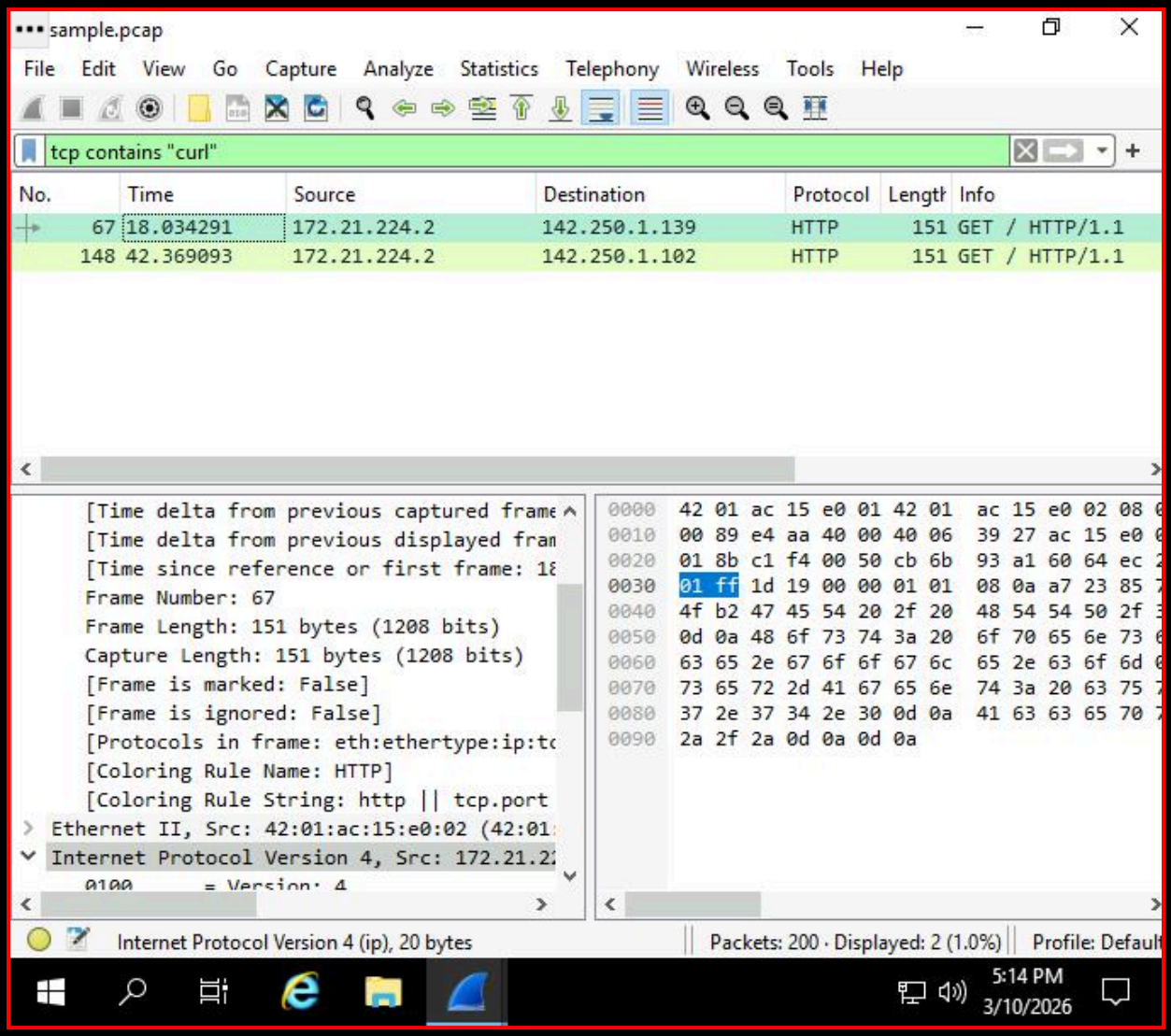
```
Internet Protocol Version 4, Src: 172.21.224.2, Dst: 169.254.169.254
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 40
    Identification: 0x89a2 (35234)
  > 010. .... = Flags: 0x2, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: TCP (6)
    Header Checksum: 0xd118 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 172.21.224.2
    Destination Address: 169.254.169.254
> Transmission Control Protocol, Src Port: 56664, Dst Port: 80, Seq: 1, Ack: 1, Len: 0
```

```
0000  42 01 ac 15 e0 01 42 01  ac 15 e0 02 08 00 45 00   B·····B· ····  ··E·
0010  00 28 89 a2 40 00 40 06  d1 18 ac 15 e0 02 a9 fe   ·(··@·@· ·····  ····
0020  a9 fe dd 58 00 50 24 08  9f 5f 09 a3 39 73 50 10   ···X·P$·  ·_··9sP·
0030  f9 46 e0 2f 00 00                                 ·F·/··
```

• THE FRAME LENGTH: [54 BYTES]

```
Frame 37: 54 bytes on wire (432 bits), 54 bytes captured (432 bits)
    Encapsulation type: Ethernet (1)
    Arrival Time: Nov 23, 2022 12:38:27.387163000 Greenwich Standard Time
    UTC Arrival Time: Nov 23, 2022 12:38:27.387163000 UTC
    Epoch Arrival Time: 1669207107.387163000
    [Time shift for this packet: 0.000000000 seconds]
    [Time delta from previous captured frame: 0.151417000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 10.799238000 seconds]
    Frame Number: 37
    Frame Length: 54 bytes (432 bits)
    Capture Length: 54 bytes (432 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp]
    [Coloring Rule Name: HTTP]
    [Coloring Rule String: http || tcp.port == 80 || http2]
```

AFTER GIVING THIS INFORMATION TO THE LEAD, THEY HAD ONE MORE TASK FOR ME. THEY WANTED ME TO SEARCH FOR A SPECIFIC PACKET THAT CONTAINED THE CURL COMMAND. IT CAME BACK WITH THE TWO RESULTS BELOW.



[CONCLUSION]

AFTER GIVING ALL THE INFORMATION TO THE LEAD, THEY WERE QUITE IMPRESSED. THEY SAID PACKET SNIFFING IS A QUALITY SKILL THAT MANY SECURITY ANALYSTS NEED IN DIRE TIMES OF EXPLORATION AND DATA ANALYSIS. THE PURPOSE OF PACKET SNIFFING IS TO MONITOR TRAFFIC AND SEARCH FOR ANYTHING SUSPICIOUS AS WELL.

IF YOU ENJOYED THIS LAB, CLICK THE LINK AND SHOOT ME A FOLLOW OR A MESSAGE ON LINKEDIN. I'M HAPPY TO COLLABORATE AND WORK. THANK YOU FOR READING.