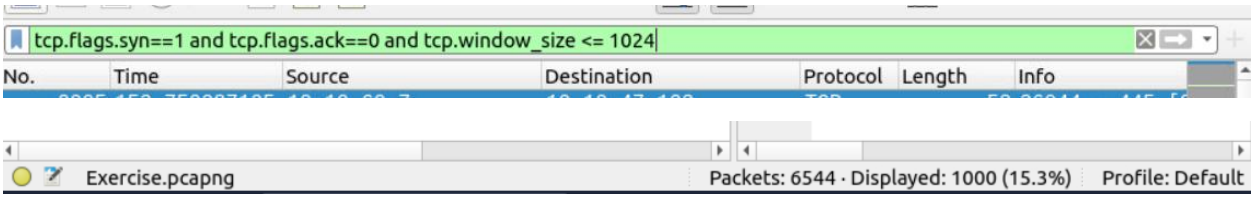
Wireshark: Traffic Analysis 2026/01/17

NMAP Scans

Detecting suspicious activities in chunked files is easy and a great way to learn how to focus on the details. **Now use the "~/Desktop/exercise-pcaps/nmap/Exercise.pcapng" file to put your skills into practice against a single capture file and answer the questions below!**

Step 1: What is the total number of the "TCP Connect" scans?



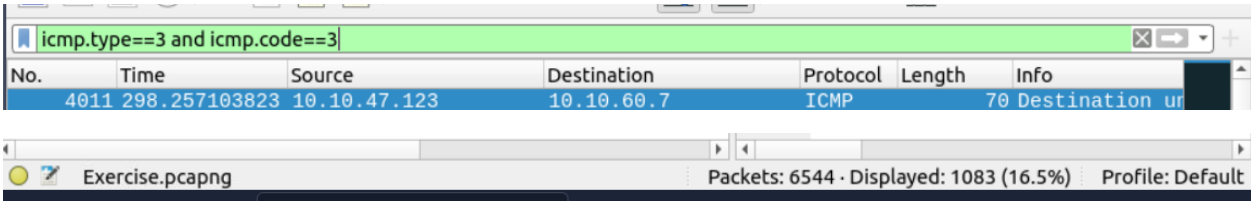
I used the following filter - **tcp.flags.syn==1 and tcp.flags.ack==0 and tcp.window_size <= 1024**. It validated the search and I identified a total of **1000** tcp connections.

Step 2: Which scan type is used to scan the TCP port 80?

- **TCP Connect**

- **Why use it?**
    - **Permissions:** It doesn't require "raw packet" privileges (sudo).
    - **Accuracy:** It is the most "honest" way to see if a port is open because it uses the same method as a web browser.
- **Downside:** It is much easier for a defender to detect because the full connection is logged by the web server.

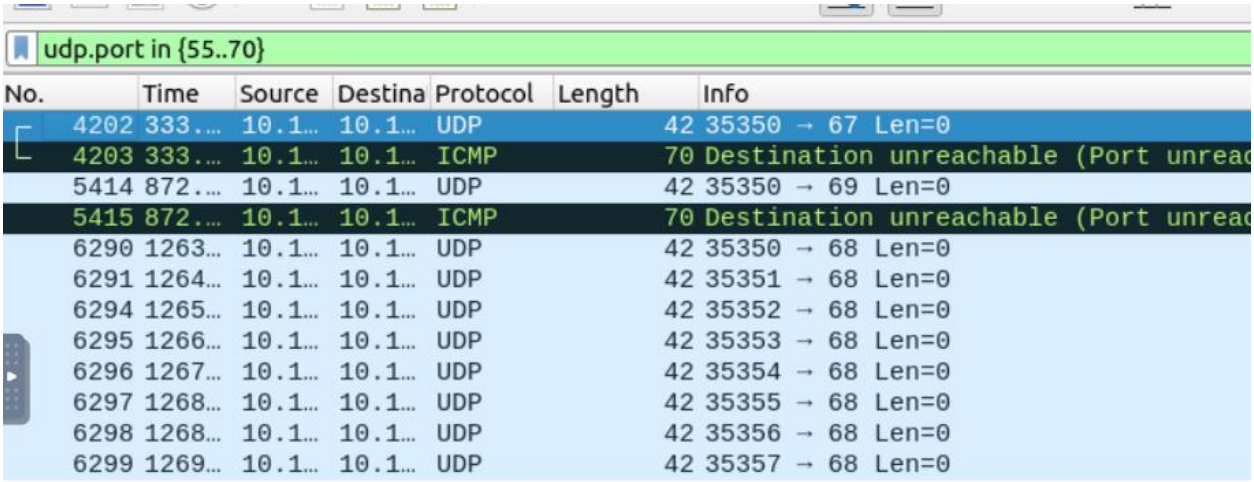
Step 3: How many "UDP close port" messages are there?



The filter – **icmp.type==3 and icmp.code==3** shows the UDP scan patterns in a capture file.

The number displayed – **1083**.

Step 4: Which UDP port in the 55-70 port range is open?

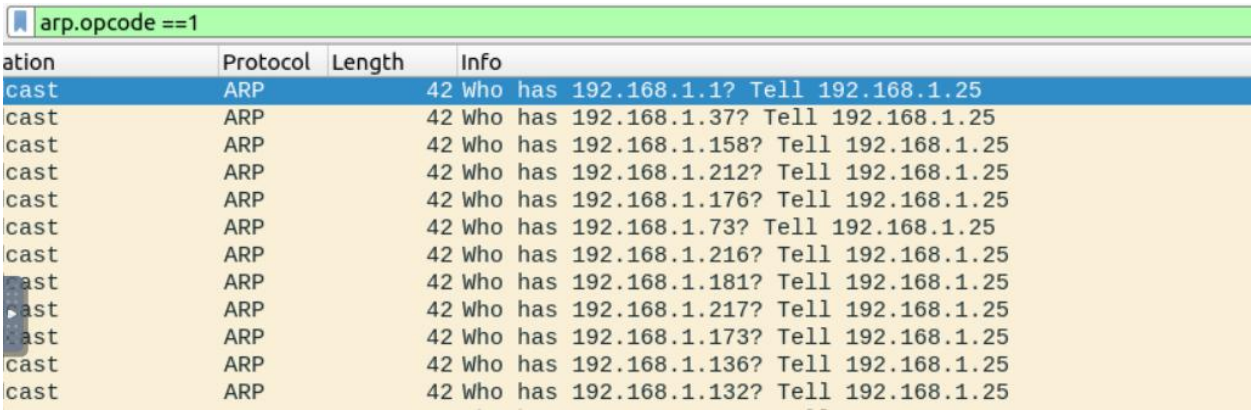


I used the filter – **udp.port in {55..70}**. The udp port identified – **68**.

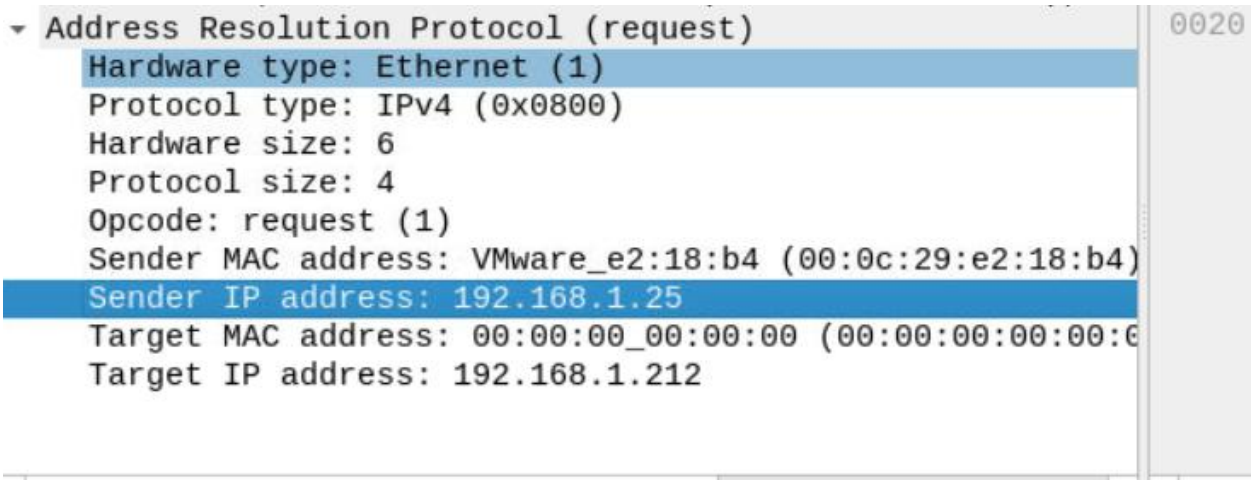
ARP Poisoning & Man In The Middle!

Detecting suspicious activities in chunked files is easy and a great way to learn how to focus on the details. **Now use the "~ /Desktop/exercise-pcaps/arp/Exercise"file to put your skills into practice against a single capture file and answer the questions below!**

Step 1: What is the number of ARP requests crafted by the attacker?

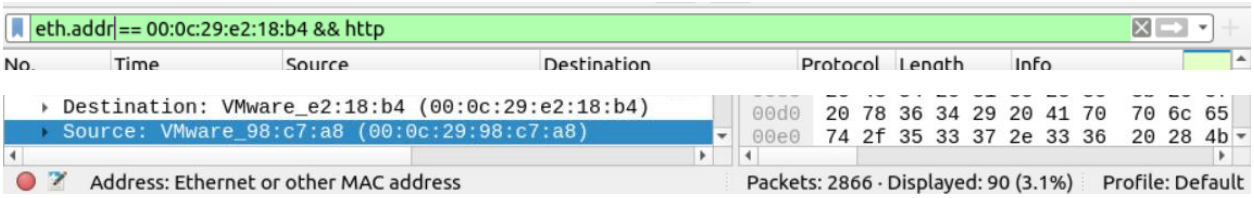


I applied the filter – **arp.opcode == 1** and searched. I identified an IP address **192.168.1.25**.



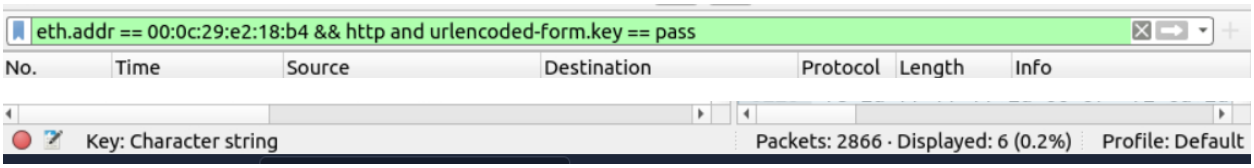
In the pane on the left at the bottom. I clicked **Address resolution Protocol (request)** then right clicked the Sender IP address and applied as a filter which gave the following filter - **arp.src.proto_ipv4 == 192.168.1.25**. I then hit search and got a total of **284**. Requests from the attacker.

Step 2: What is the number of HTTP packets received by the attacker?



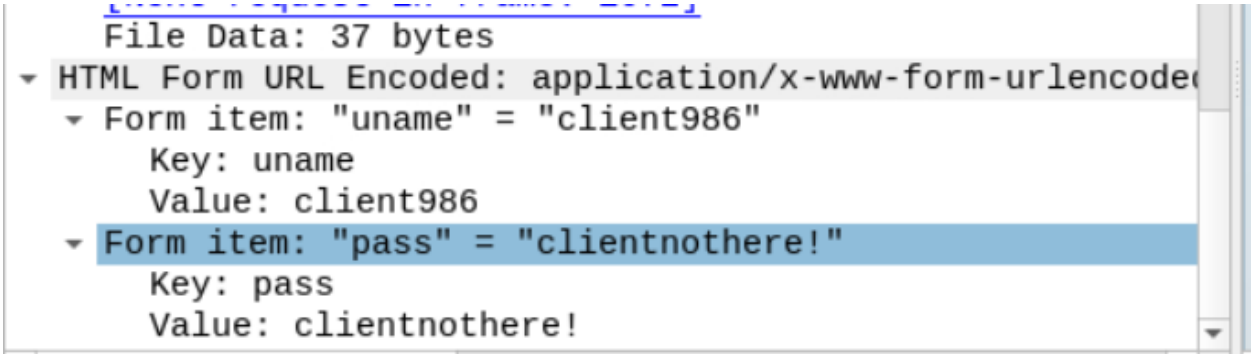
I used the filter **eth.addr == 00:0c:29:e2:18:b4 && http**. The number of http packets by received by the attacker – **90**.

Step 3: What is the number of sniffed username&password entries?



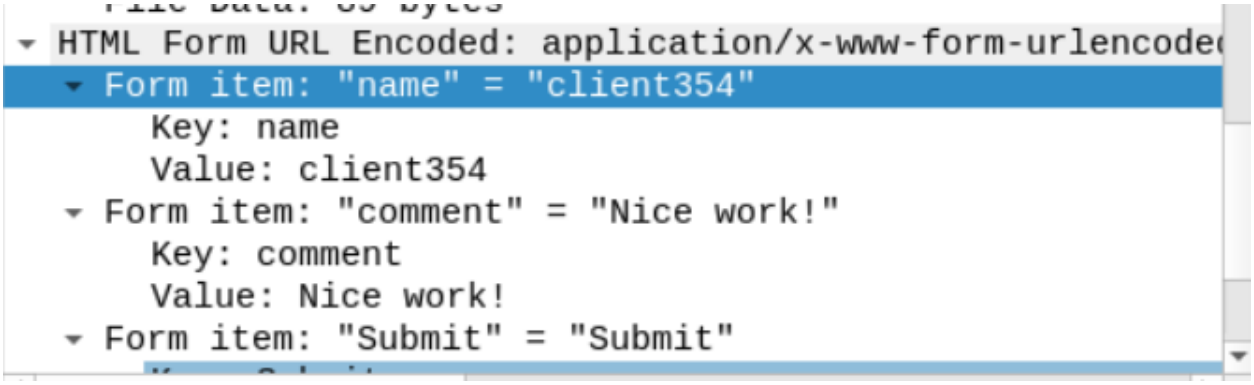
I used the filter - **eth.addr == 00:0c:29:e2:18:b4 && http and urlencoded-form.key == pass**. The number of sniffed entries was **6**.

Step 4: What is the password of the "Client986"?



I clicked each packet and identified the user name – **client986** as well as the pass = **clientnothere!**.

Step 5: What is the comment provided by the "Client354"?

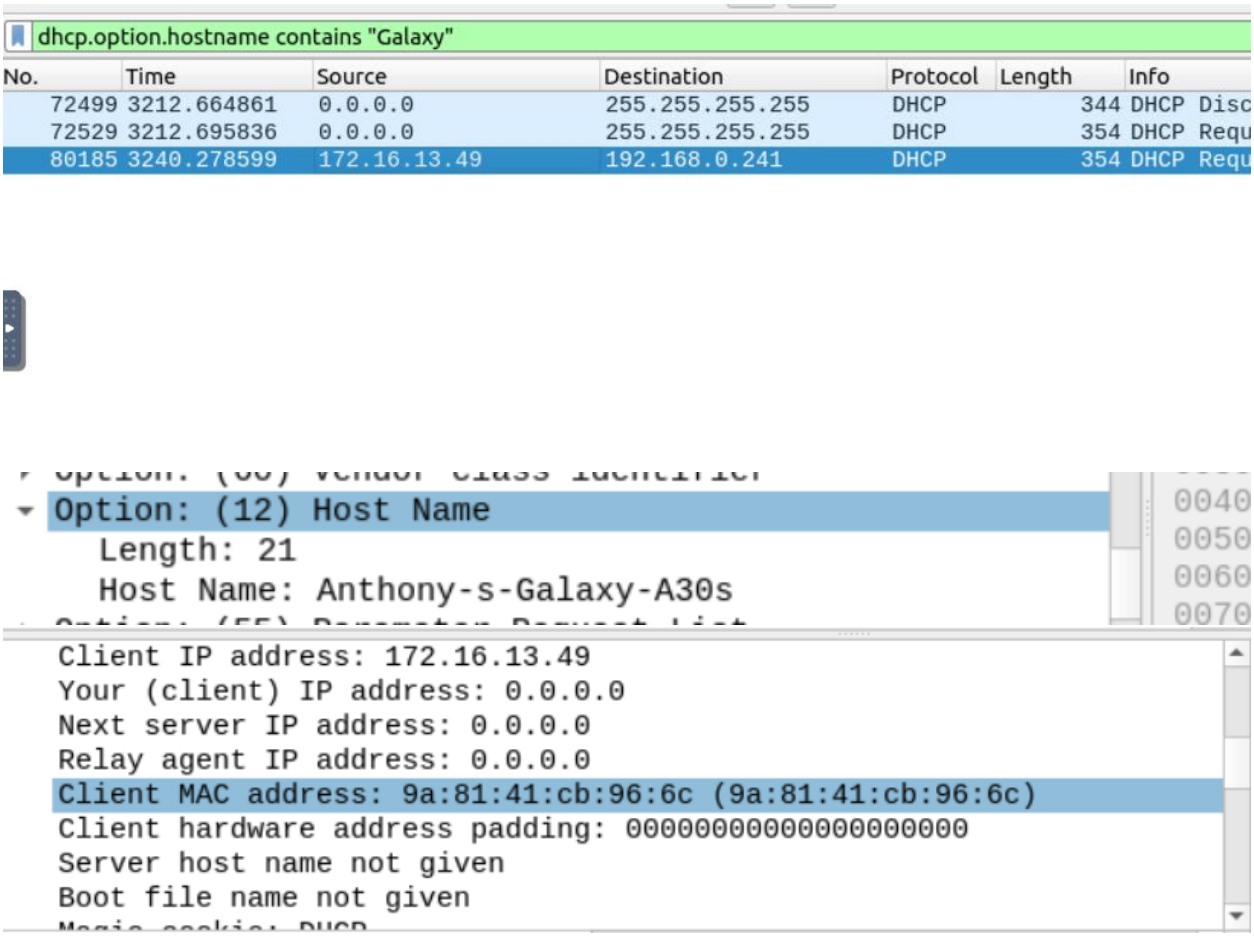


I clicked ctrl + f and entered Client354, cleared the main filter and clicked search. The comment found in the pane from the user – **Nice work!**.

Identifying Hosts: DHCP, NetBIOS and Kerberos

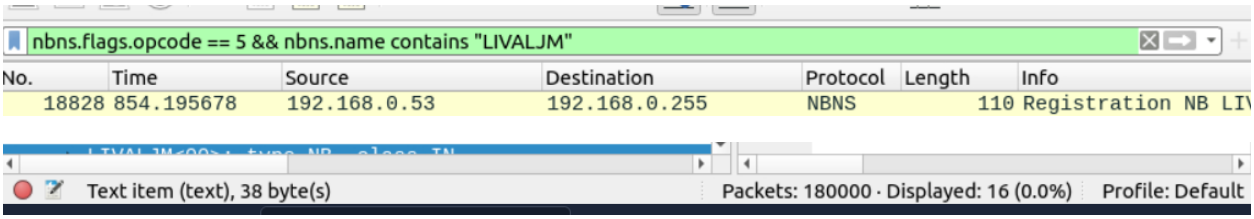
Detecting suspicious activities in chunked files is easy and a great way to learn how to focus on the details. **Now use the "~/Desktop/exercise-pcaps/dhcp-netbios-kerberos/dhcp-netbios.pcap" file to solve the questions 1 through 3 and the "~/Desktop/exercise-pcaps/dhcp-netbios-kerberos/**kerberos.pcap" to solve questions 4 and 5.**

Step 1: What is the MAC address of the host "Galaxy A30"?



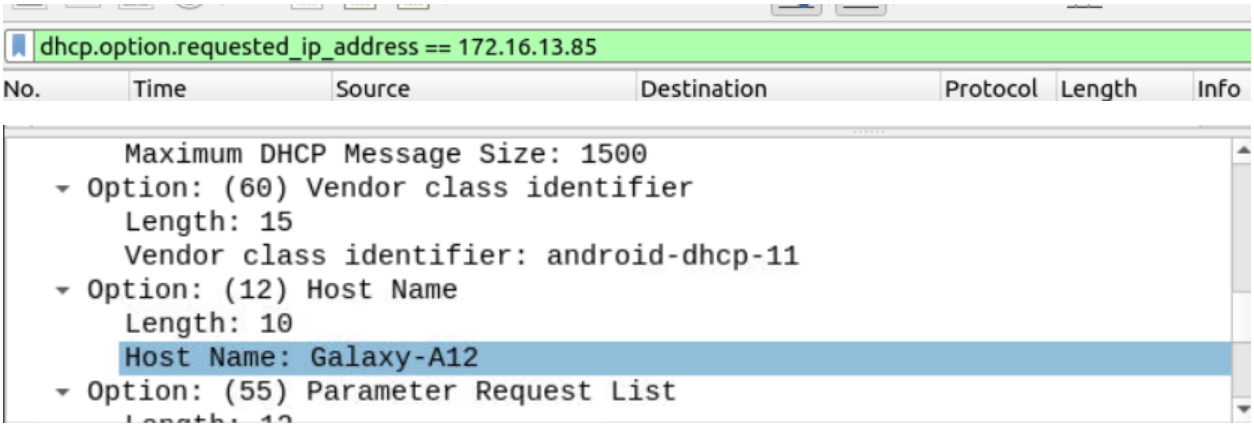
I added a the filter – **dhcp.option.hostname contains "Galaxy"**. I investigated the 3 packets that appeared, but the one that stood out to me was the one that didn't have an IP source of – **0.0.0.0**, it showed a device that had been a part of the network for some time. I clicked on it, validated the host name within the packet, and then investigated the DHCP section and identified the source MAC address – **9a:81:41:cb:96:6c**.

Step 2: How many NetBIOS registration requests does the "LIVALJM" workstation have?



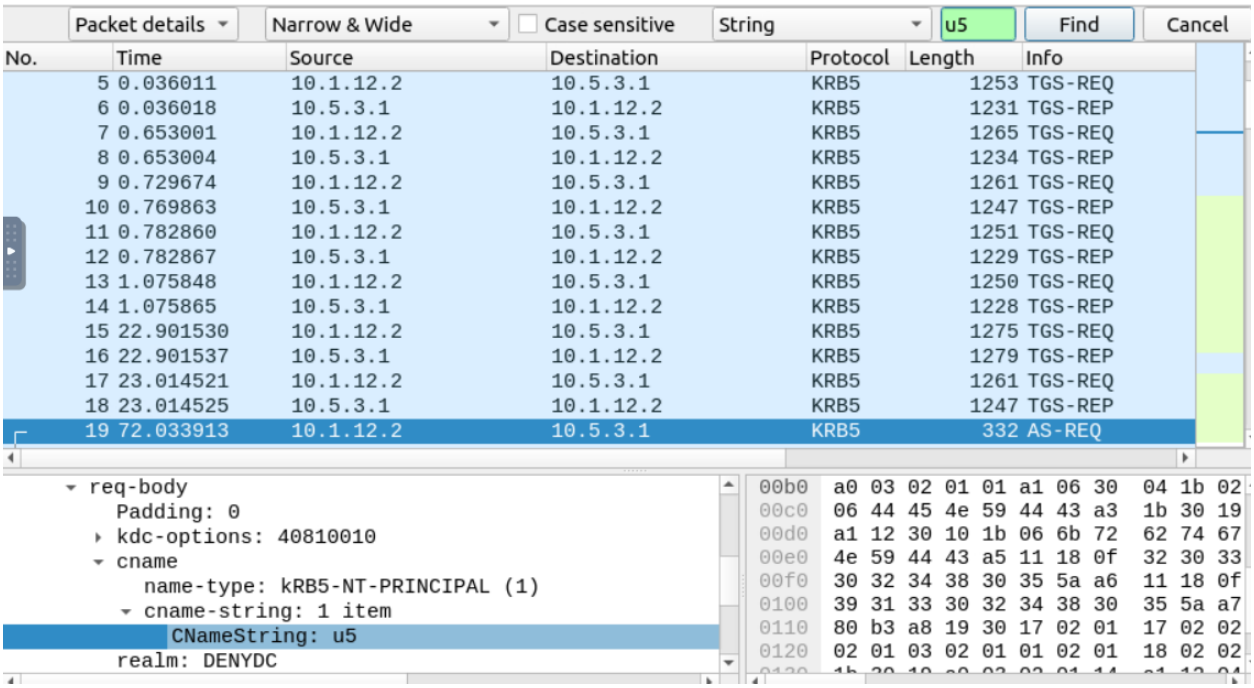
I used the relevant filter and it produced a number of **16** packets.

Step 3: Which host requested the IP address "172.16.13.85"?



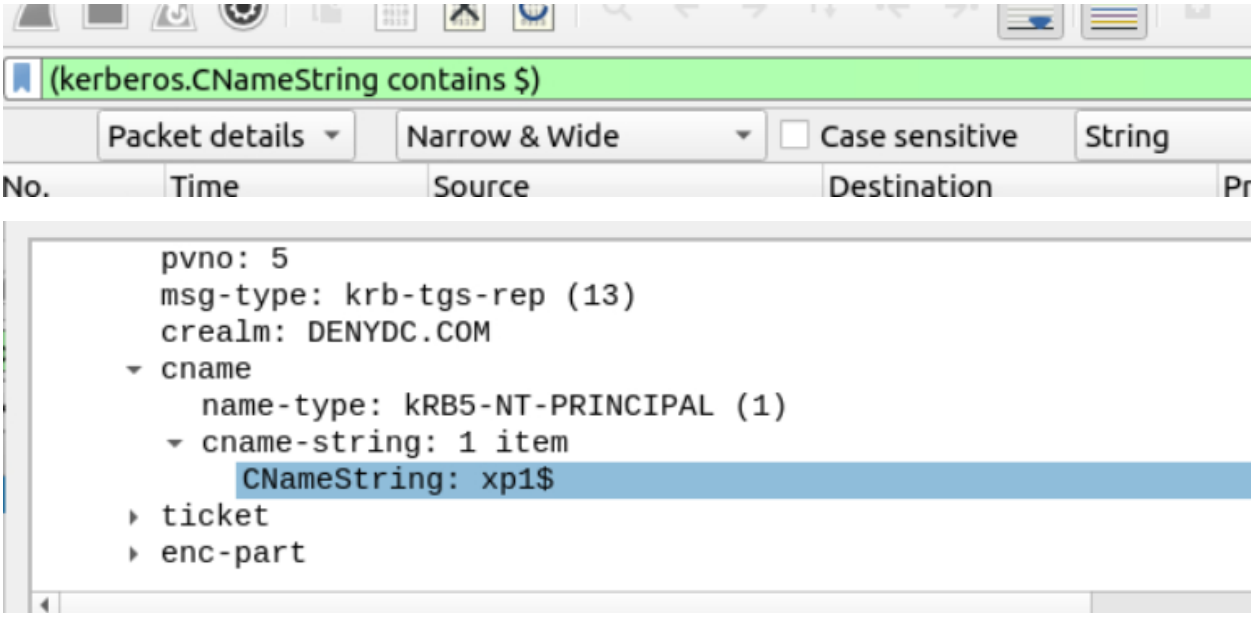
I applied the relevant filter and hit search. I then proceeded to click on the 1 packet that appeared and investigate it. I went to Hostname within the DHCP protocol section and identified the host name – **Galaxy-A12**.

Step 4: What is the IP address of the user "u5"? (Enter the address in defanged format.)



I clicked ctrl + f and searched **u5**, I clicked to further validate the user name – **u5** the packet and identified the relevant defanged IP address – **10[.]1[.]12[.]2[.]**.

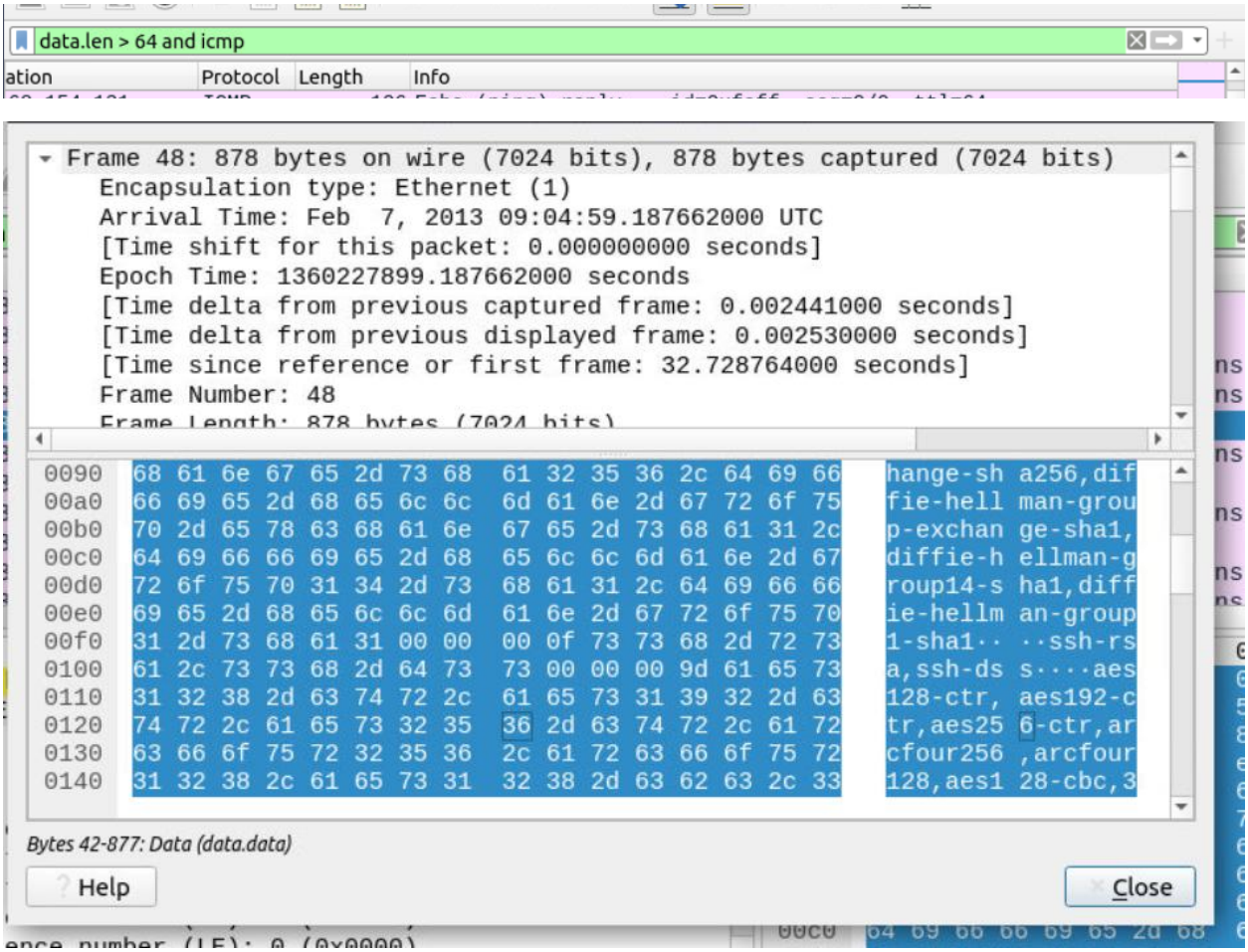
Step 5: What is the hostname of the available host in the Kerberos packets?



I applied the relevant filter and hit search. I then proceeded to investigate the contents of the packet and identified the name – **xp1$**.

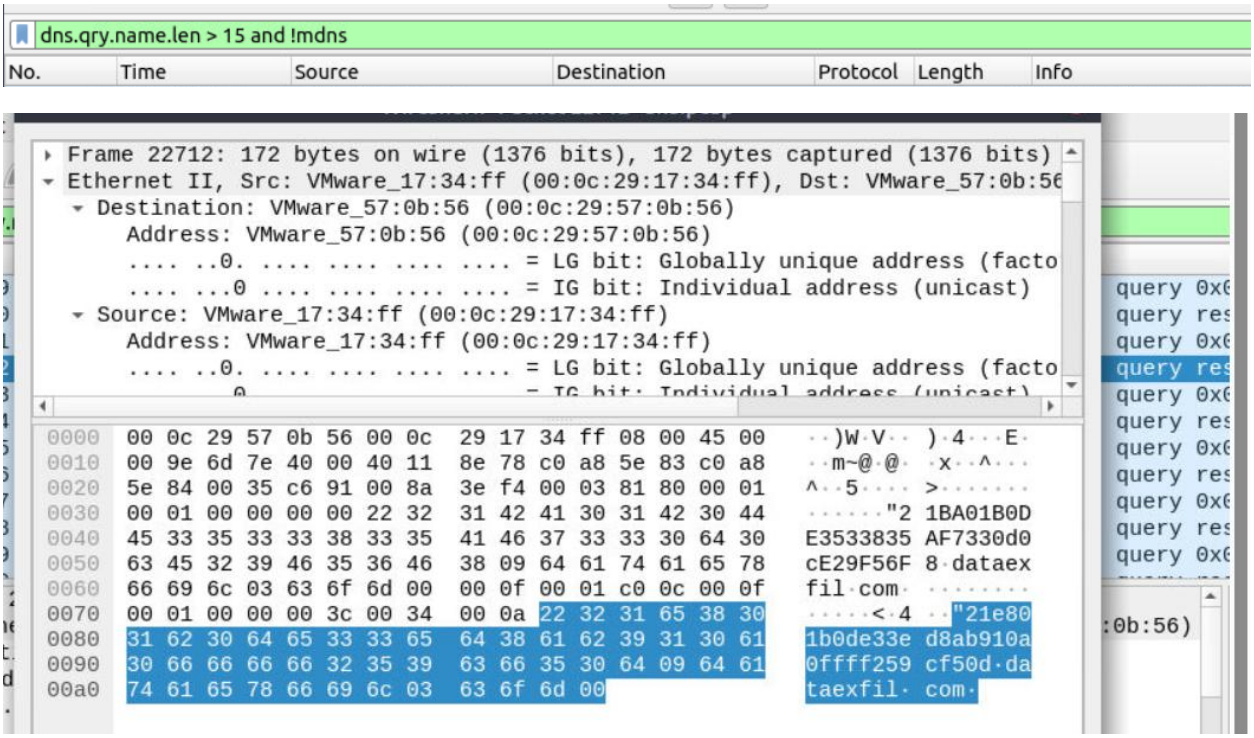
Tunneling Traffic: DNS and ICMP

Step 1: Use the "Desktop/exercise-pcaps/dns-icmp/icmp-tunnel.pcap" file. Investigate the anomalous packets. Which protocol is used in ICMP tunnelling?



I used the relevant filter, I then proceeded to inspect later packets as they'd help by giving more information. I investigated 1 of the packets and identified the protocol used – **ssh** in the payload data in the packet bytes section.

Step 2: Use the "Desktop/exercise-pcaps/dns-icmp/dns.pcap" file.Investigate the anomalous packets. What is the suspicious main domain address that receives anomalous DNS queries? (Enter the address in defanged format.)

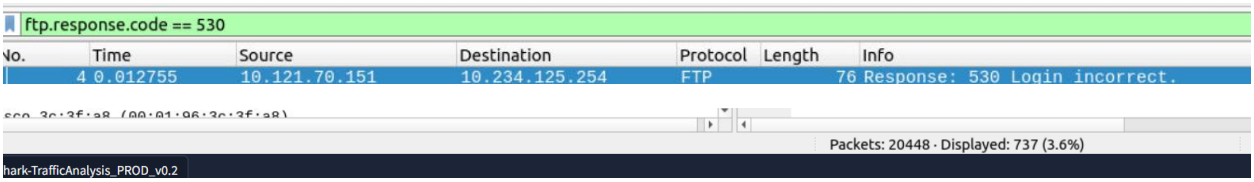


I entered the relevant filter and hit search. I then proceeded to investigate the packets and looked into the payload data in the packet bytes. I identified the main address and defanged it using Cyberchef – **dataexfil[.]com**.

Cleartext Protocol Analysis: FTP

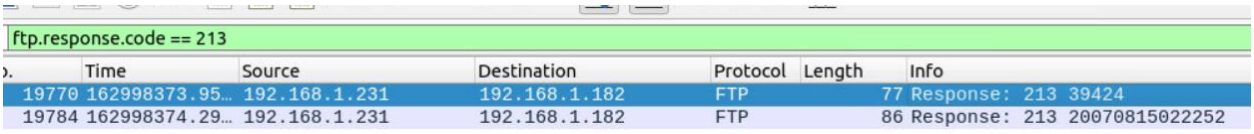
Detecting suspicious activities in chunked files is easy and a great way to learn how to focus on the details. **Now use the "~/Desktop/exercise-pcaps/ftp/ftp.pcap" file to put your skills into practice and answer the questions below!**

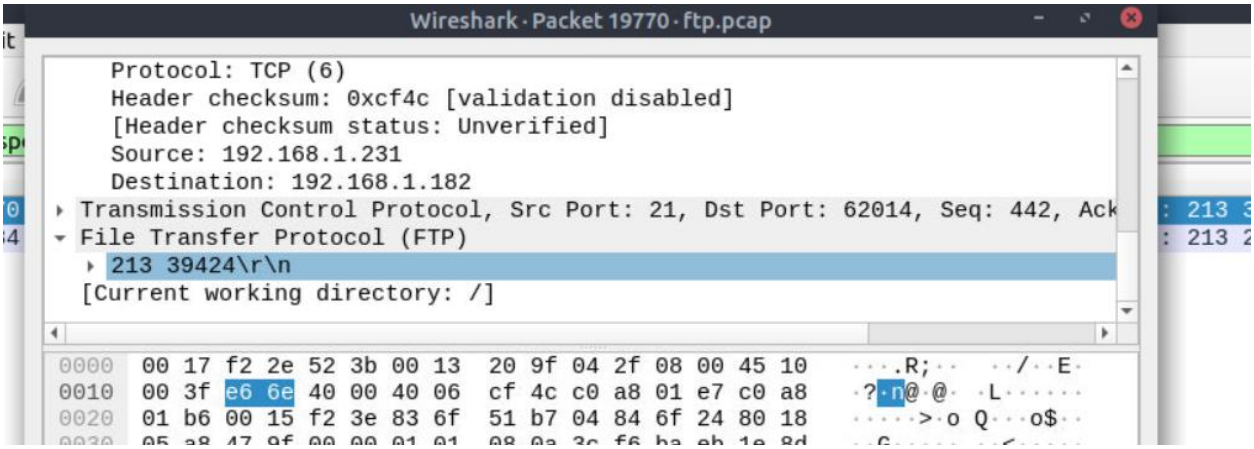
Step 1: How many incorrect login attempts are there?



I used the filter – **ftp.response.code ==530**. I identified the number of failed login attempts – **737**.

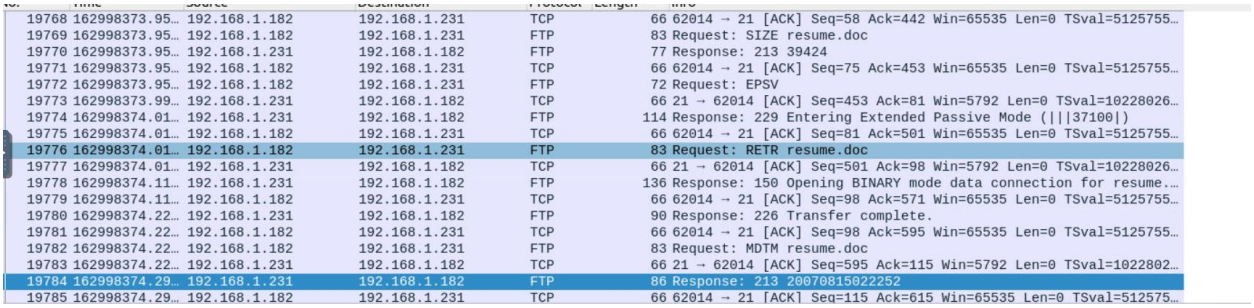
Step 2: What is the size of the file accessed by the "ftp" account?





I used the relevant filter. I then clicked on the first packet and investigated it. I navigated to the FTP section and identified the size – **39424**.

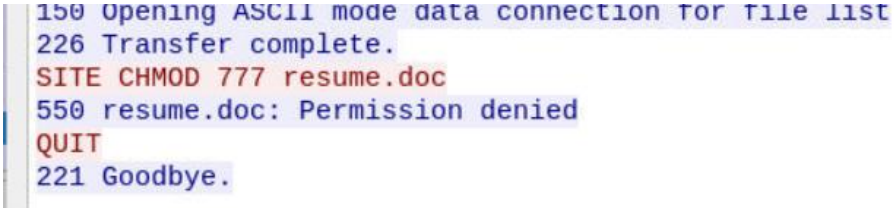
Step 3: The adversary uploaded a document to the FTP server. What is the filename?



19768	162998373.95…	192.168.1.182	192.168.1.231	TCP	66	62014 → 21 [ACK] Seq=58 Ack=442 Win=65535 Len=0 TSval=5125755…
19769	162998373.95…	192.168.1.182	192.168.1.231	FTP	83	Request: SIZE resume.doc
19770	162998373.95…	192.168.1.231	192.168.1.182	FTP	77	Response: 213 39424
19771	162998373.95…	192.168.1.182	192.168.1.231	TCP	66	62014 → 21 [ACK] Seq=75 Ack=453 Win=65535 Len=0 TSval=5125755…
19772	162998373.95…	192.168.1.182	192.168.1.231	FTP	72	Request: EPSV
19773	162998373.99…	192.168.1.231	192.168.1.182	TCP	66	21 → 62014 [ACK] Seq=453 Ack=81 Win=5792 Len=0 TSval=10228026…
19774	162998374.01…	192.168.1.231	192.168.1.182	FTP	114	Response: 229 Entering Extended Passive Mode (|||37100|)
19775	162998374.01…	192.168.1.182	192.168.1.231	TCP	66	62014 → 21 [ACK] Seq=81 Ack=501 Win=65535 Len=0 TSval=5125755…
19776	162998374.01…	192.168.1.182	192.168.1.231	FTP	83	Request: RETR resume.doc
19777	162998374.01…	192.168.1.231	192.168.1.182	TCP	66	21 → 62014 [ACK] Seq=501 Ack=98 Win=5792 Len=0 TSval=10228026…
19778	162998374.11…	192.168.1.231	192.168.1.182	FTP	136	Response: 150 Opening BINARY mode data connection for resume.…
19779	162998374.11…	192.168.1.182	192.168.1.231	TCP	66	62014 → 21 [ACK] Seq=98 Ack=571 Win=65535 Len=0 TSval=5125755…
19780	162998374.22…	192.168.1.231	192.168.1.182	FTP	90	Response: 226 Transfer complete.
19781	162998374.22…	192.168.1.182	192.168.1.231	TCP	66	62014 → 21 [ACK] Seq=98 Ack=595 Win=65535 Len=0 TSval=5125755…
19782	162998374.22…	192.168.1.182	192.168.1.231	FTP	83	Request: MDTM resume.doc
19783	162998374.22…	192.168.1.231	192.168.1.182	TCP	66	21 → 62014 [ACK] Seq=595 Ack=115 Win=5792 Len=0 TSval=1022802…
19784	162998374.29…	192.168.1.231	192.168.1.182	FTP	86	Response: 213 20070815022252
19785	162998374.29…	192.168.1.182	192.168.1.231	TCP	66	62014 → 21 [ACK] Seq=115 Ack=615 Win=65535 Len=0 TSval=512575…

I started by clearing the filter, investigate the packets, then identified the file – **resume.doc**.

Step 4: The adversary tried to assign special flags to change the executing permissions of the uploaded file. What is the command used by the adversary?



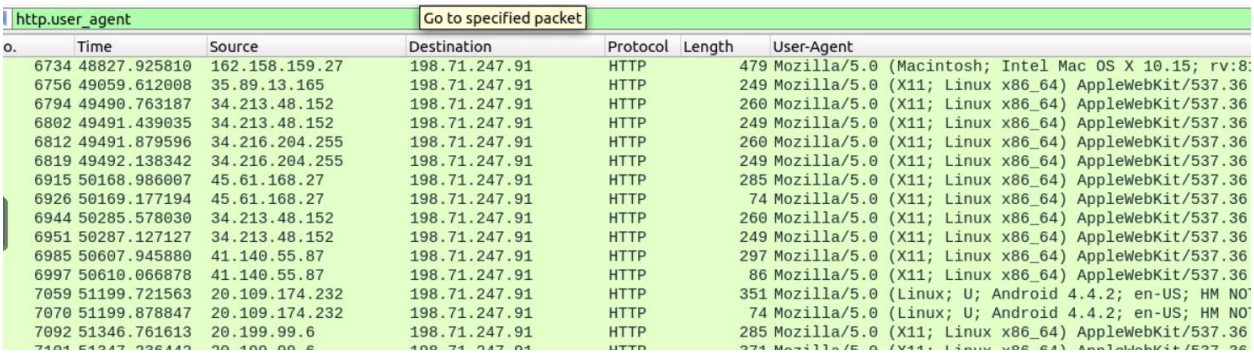
```
150 Opening ASCII mode data connection for file list
226 Transfer complete.
SITE CHMOD 777 resume.doc
550 resume.doc: Permission denied
QUIT
221 Goodbye.
```

I right clicked on the packet follo > tcp stream. I then skimmed through the lines and identified the command – **CHMOD 777**.

Cleartext Protocol Analysis: HTTP

Detecting suspicious activities in chunked files is easy and a great way to learn how to focus on the details. **Now use the Use the "~/Desktop/exercise-pcaps/http/user-agent.pcap" file to answer questions 1 to 2 and the "~/Desktop/exercise-pcaps/http/http.pcapng" file to answer questions 3 to 4.**

Step 1: Investigate the user agents. What is the number of anomalous "user-agent" types?

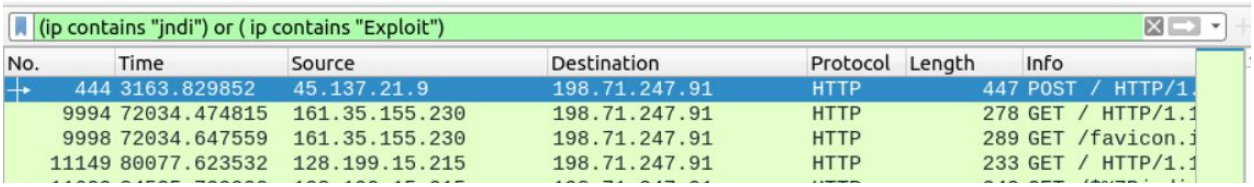


o.	Time	Source	Destination	Protocol	Length	User-Agent
6734	48827.925810	162.158.159.27	198.71.247.91	HTTP	479	Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:8
6756	49059.612008	35.89.13.165	198.71.247.91	HTTP	249	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
6794	49490.763187	34.213.48.152	198.71.247.91	HTTP	260	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
6802	49491.439035	34.213.48.152	198.71.247.91	HTTP	249	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
6812	49491.879596	34.216.204.255	198.71.247.91	HTTP	260	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
6819	49492.138342	34.216.204.255	198.71.247.91	HTTP	249	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
6915	50168.986007	45.61.168.27	198.71.247.91	HTTP	285	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
6926	50169.177194	45.61.168.27	198.71.247.91	HTTP	74	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
6944	50285.578030	34.213.48.152	198.71.247.91	HTTP	260	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
6951	50287.127127	34.213.48.152	198.71.247.91	HTTP	249	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
6985	50607.945880	41.140.55.87	198.71.247.91	HTTP	297	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
6997	50610.066878	41.140.55.87	198.71.247.91	HTTP	86	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
7059	51199.721563	20.109.174.232	198.71.247.91	HTTP	351	Mozilla/5.0 (Linux; U; Android 4.4.2; en-US; HM NO
7070	51199.878847	20.109.174.232	198.71.247.91	HTTP	74	Mozilla/5.0 (Linux; U; Android 4.4.2; en-US; HM NO
7092	51346.761613	20.199.99.6	198.71.247.91	HTTP	285	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36

I used the filter – **http.user_agent**. I then clicked on the packet. Right clicked on the user agent and hit apply as column to view the types – **6.**

Step 2: What is the packet number with a subtle spelling difference in the user agent field?

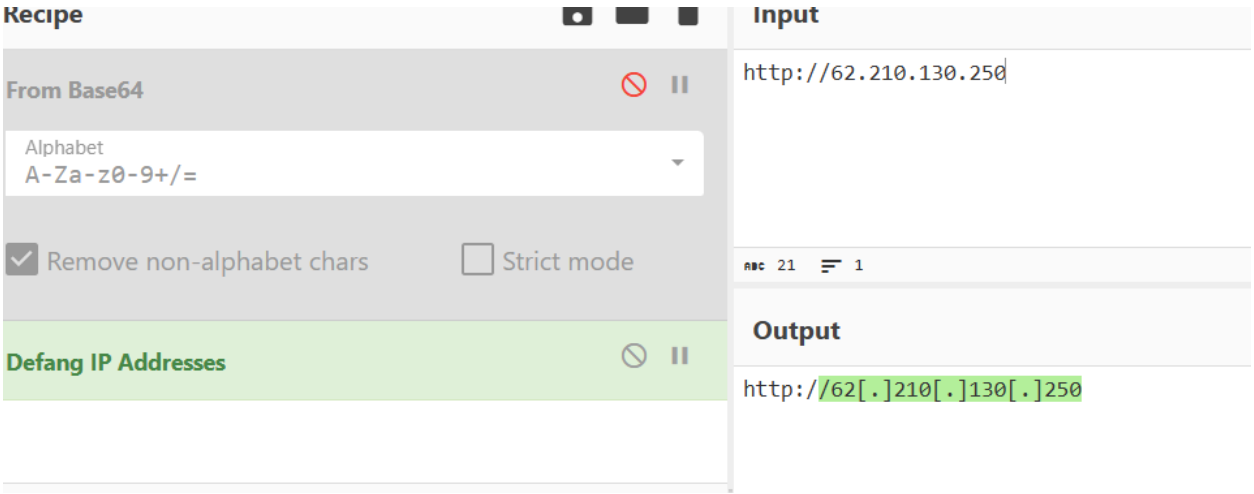
Step 3: Locate the "Log4j" attack starting phase. What is the packet number?



I used the following filter – **(ip contains "jndi") or (ip contains "Exploit")** then clicked search. The packet number – **444**.

Step 4: Locate the "Log4j" attack starting phase and decode the base64 command. What is the IP address contacted by the adversary? (Enter the address in defanged format and exclude "{}".)

I clicked on the packet, navigated to the user agent and copied the value and pasted it to Cyberchef. I then decoded it, copied the IP address and pasted it again in the input section.



I defanged the IP address and copied the value – **62[.]210[.]130[.]250**.

Encrypted Protocol Analysis: Decrypting HTTPS

Detecting suspicious activities in chunked files is easy and a great way to learn how to focus on the details. **Now use the "Desktop/exercise-pcaps/https/Exercise.pcap" file to put your skills into practice and answer the questions below!**

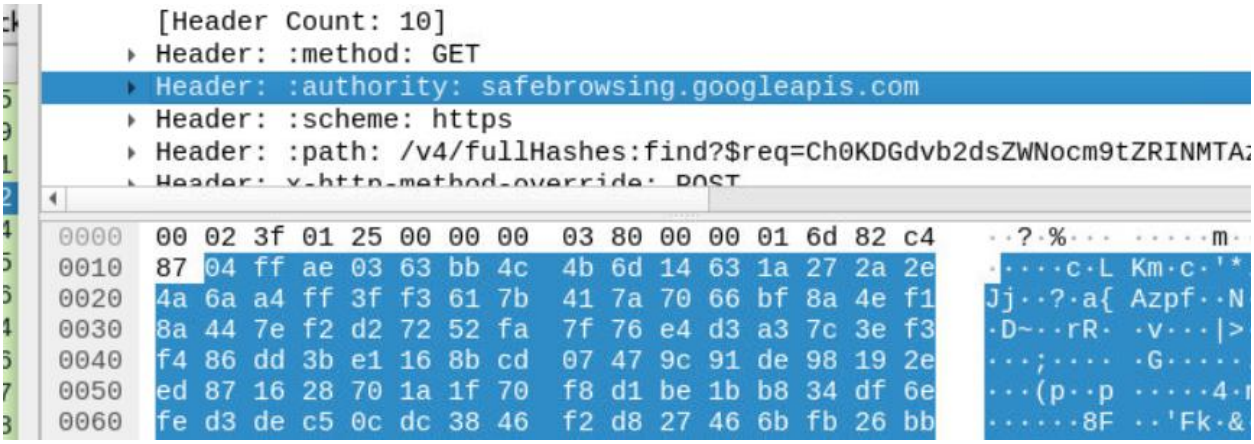
Step 1: What is the frame number of the "Client Hello" message sent to "accounts.google.com"?

As soon as I opened the file. I identified the packet number after doing a bit of investigation – **16**.

Step 2: Decrypt the traffic with the "KeysLogFile.txt" file. What is the number of HTTP2 packets?

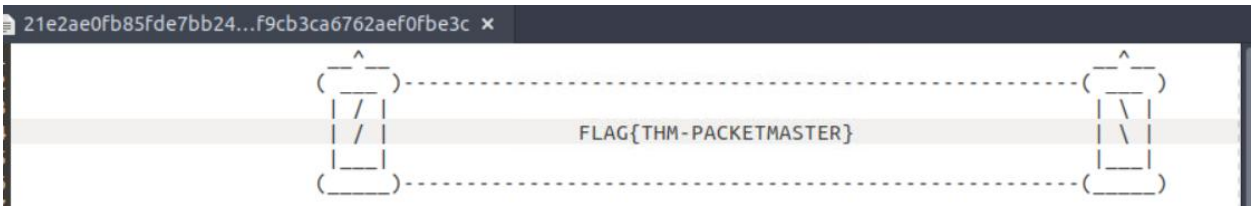
I clicked on Edit > Preferences > Protocols > TLS and browsed for the file. I then used the following filter – **http2**. The number of packets – 115.

Step 4: Go to Frame 322. What is the authority header of the HTTP2 packet? (Enter the address in defanged format.)



I clicked on the packet, opened its contents and navigated to the authority header and defanged it on Cyberchef - **safebrowsing[.]googleapis[.]com**.

Step 5: Investigate the decrypted packets and find the flag! What is the flag?



I searched flag.txt, found the packet and exported it as an object and saved it in the same folder as the file currently open in wireshark. I opened the file and got the flag.

Conclusion Summary

This Wireshark Traffic Analysis lab provided comprehensive, hands-on exposure to identifying and investigating malicious and anomalous network activity across both cleartext and encrypted protocols. Through systematic use of Wireshark display filters, protocol dissection, and payload inspection, I successfully identified multiple attack techniques including Nmap TCP and UDP scans, ARP poisoning and man-in-the-middle activity, credential sniffing over HTTP and FTP, DNS and ICMP tunneling, and exploitation attempts such as Log4j.

The exercise strengthened my ability to correlate packet-level evidence with attacker behaviour accurately distinguishing scan types, identifying compromised hosts, extracting credentials, reconstructing transferred files, and detecting covert channels used for data exfiltration. Additionally, decrypting HTTPS traffic using key log files reinforced practical understanding of TLS handshakes and post-decryption analysis of HTTP/2 traffic.

Overall, this lab reinforced real-world SOC analyst skills such as alert validation, attacker profiling, protocol analysis, and evidence extraction from PCAPs. It demonstrated how detailed traffic analysis can uncover the full attack chain, from reconnaissance to exploitation and exfiltration, and highlighted the critical role of network visibility in effective incident detection and response.