

Event Analysis

LAB01_IS3523_DR. MUÑOZ

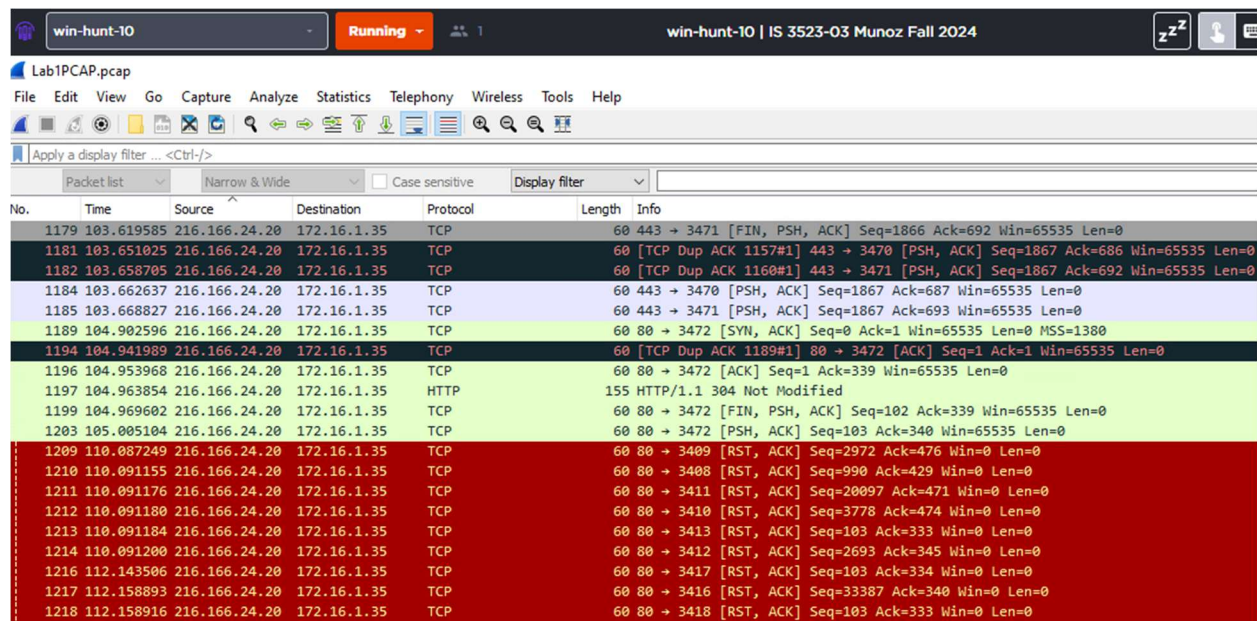
DAVID MURILLO SANTIAGO

INTRODUCTION

In this lab, we will conduct a thorough analysis of network traffic to determine whether any significant or malicious events occurred on the local network. The network traffic was captured using Wireshark, and the packet capture file will be analyzed using a variety of security tools, including Network Miner, Wireshark, and SNORT.

PROCESS

Initial observations from the captured traffic reveal the presence of the TiVoConnect protocol, multiple connections to RBFCU, and activity from a device named KAUFMANUPSTAIRS. Notably, there are signs of duplicate MAC addresses, which raises suspicion of ARP spoofing or other malicious activities. The focus of this analysis will be to validate whether these anomalies, particularly the duplicate MAC addresses, suggest a deliberate attempt to hijack or intercept traffic.



No.	Time	Source	Destination	Protocol	Length	Info
1179	103.619585	216.166.24.20	172.16.1.35	TCP	60	443 → 3471 [FIN, PSH, ACK] Seq=1866 Ack=692 Win=65535 Len=0
1181	103.651025	216.166.24.20	172.16.1.35	TCP	60	[TCP Dup ACK 1157#1] 443 → 3470 [PSH, ACK] Seq=1867 Ack=686 Win=65535 Len=0
1182	103.658705	216.166.24.20	172.16.1.35	TCP	60	[TCP Dup ACK 1160#1] 443 → 3471 [PSH, ACK] Seq=1867 Ack=692 Win=65535 Len=0
1184	103.662637	216.166.24.20	172.16.1.35	TCP	60	443 → 3470 [PSH, ACK] Seq=1867 Ack=687 Win=65535 Len=0
1185	103.668827	216.166.24.20	172.16.1.35	TCP	60	443 → 3471 [PSH, ACK] Seq=1867 Ack=693 Win=65535 Len=0
1189	104.902596	216.166.24.20	172.16.1.35	TCP	60	80 → 3472 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1380
1194	104.941989	216.166.24.20	172.16.1.35	TCP	60	[TCP Dup ACK 1189#1] 80 → 3472 [ACK] Seq=1 Ack=1 Win=65535 Len=0
1196	104.953968	216.166.24.20	172.16.1.35	TCP	60	80 → 3472 [ACK] Seq=1 Ack=339 Win=65535 Len=0
1197	104.963854	216.166.24.20	172.16.1.35	HTTP	155	HTTP/1.1 304 Not Modified
1199	104.969602	216.166.24.20	172.16.1.35	TCP	60	80 → 3472 [FIN, PSH, ACK] Seq=102 Ack=339 Win=65535 Len=0
1203	105.005104	216.166.24.20	172.16.1.35	TCP	60	80 → 3472 [PSH, ACK] Seq=103 Ack=340 Win=65535 Len=0
1209	110.087249	216.166.24.20	172.16.1.35	TCP	60	80 → 3409 [RST, ACK] Seq=2972 Ack=476 Win=0 Len=0
1210	110.091155	216.166.24.20	172.16.1.35	TCP	60	80 → 3408 [RST, ACK] Seq=990 Ack=429 Win=0 Len=0
1211	110.091176	216.166.24.20	172.16.1.35	TCP	60	80 → 3411 [RST, ACK] Seq=20097 Ack=471 Win=0 Len=0
1212	110.091180	216.166.24.20	172.16.1.35	TCP	60	80 → 3410 [RST, ACK] Seq=3778 Ack=474 Win=0 Len=0
1213	110.091184	216.166.24.20	172.16.1.35	TCP	60	80 → 3413 [RST, ACK] Seq=103 Ack=333 Win=0 Len=0
1214	110.091200	216.166.24.20	172.16.1.35	TCP	60	80 → 3412 [RST, ACK] Seq=2693 Ack=345 Win=0 Len=0
1216	112.143506	216.166.24.20	172.16.1.35	TCP	60	80 → 3417 [RST, ACK] Seq=103 Ack=334 Win=0 Len=0
1217	112.158893	216.166.24.20	172.16.1.35	TCP	60	80 → 3416 [RST, ACK] Seq=33387 Ack=340 Win=0 Len=0
1218	112.158916	216.166.24.20	172.16.1.35	TCP	60	80 → 3418 [RST, ACK] Seq=103 Ack=333 Win=0 Len=0

Figure 1: Wireshark Traffic.

a. How long did the session capture last?

First, I utilized the 'statistics' > 'capture file properties' ribbon to view high level information on the packet capture. One notable piece of information I discovered was that the packet capture lasted 8 minutes and 25 seconds. The first packet captured was from October 10th, 2005, at 4:29 and the last packet captured was that same day at 4:38. Doing the math I was able to verify the integrity of the data, but in the Time/elapsed section it automatically calculated the duration of the packet capture.

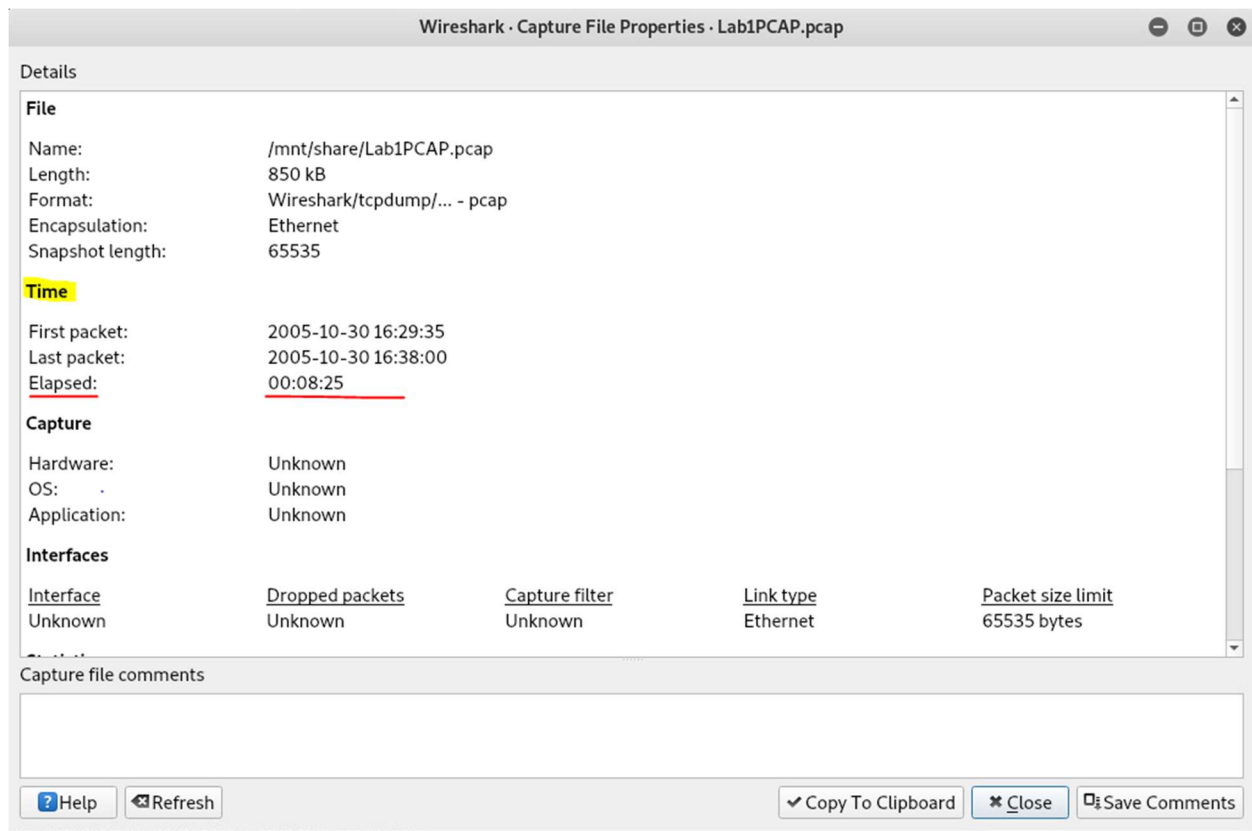


Figure 2: Duration of Packet Capture.

a. How many packets were captured? How many bytes were captured?

As shown in the PCAP properties, the number of packets captured was 2,449 packets. The number of bytes captured was 81,1157; with the average packet size being 331 bytes long. We could also find the duration in seconds for the packet capture. The capture lasted 505 seconds, matching the previous duration in minutes.

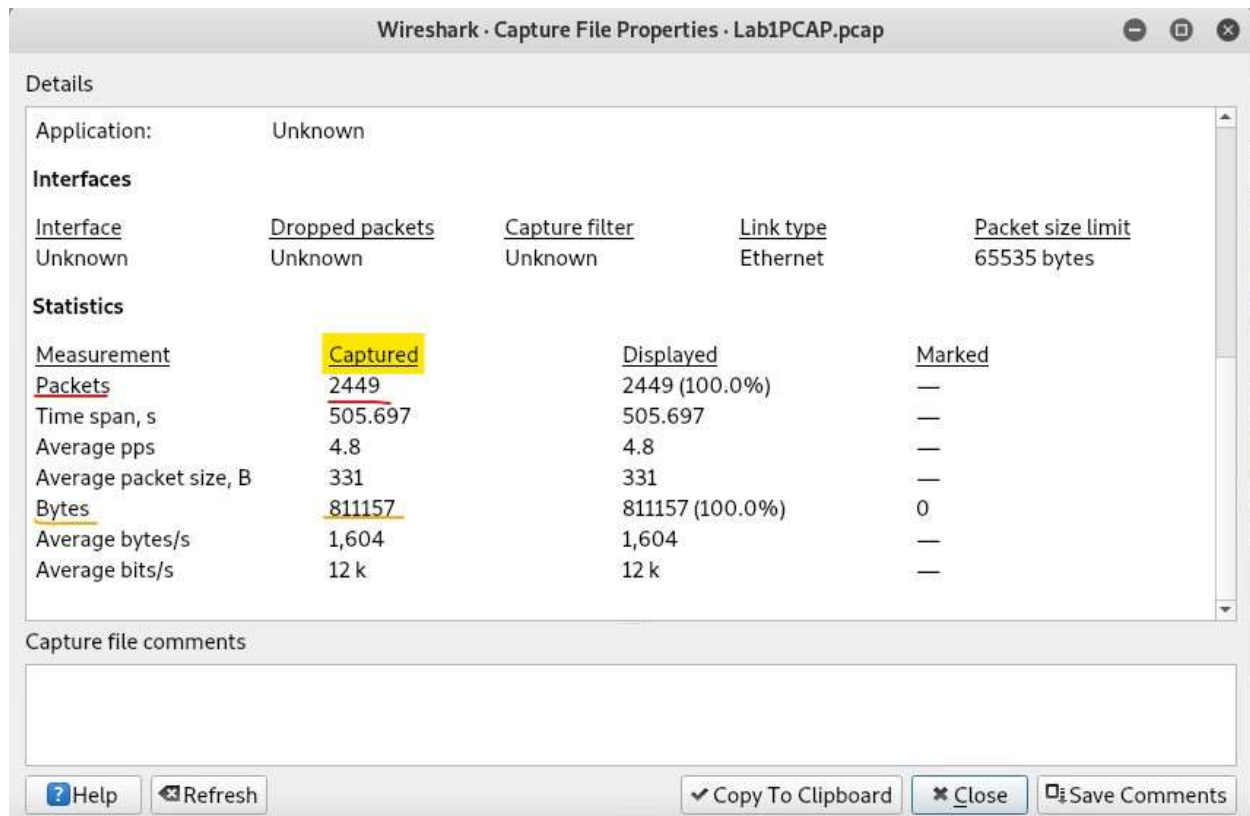


Figure 3: Bytes Captured.

b. What protocols were observed?

To observe the protocols found in the packet capture, I used the 'Protocol Hierarchy Statistics' feature, which allowed me to view the captured protocols and their relative percentage of packets in the capture. From this, I determined that the top protocols were TCP (84.9%), UDP (10%), HTTP (7.3%), ARP (5.1%), and TiVOConnect Discovery Protocol (4.9%).

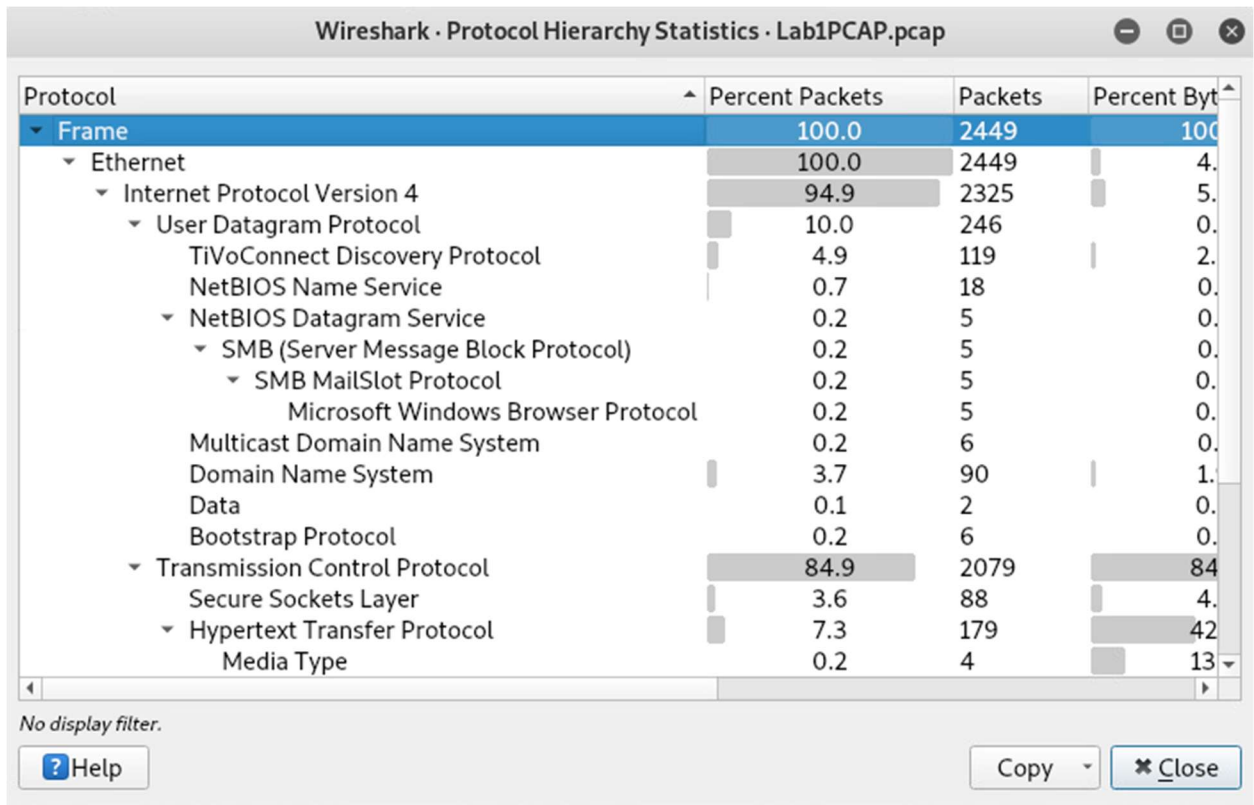


Figure 4: Protocols Observed.

c. When did the bulk of the data get transmitted?

I navigated to the IO Graph feature under the "Statistics" menu to visualize the traffic patterns over time. The bulk of the data got transmitted right at 4:30 pm, as revealed by the spike on the IO graph.

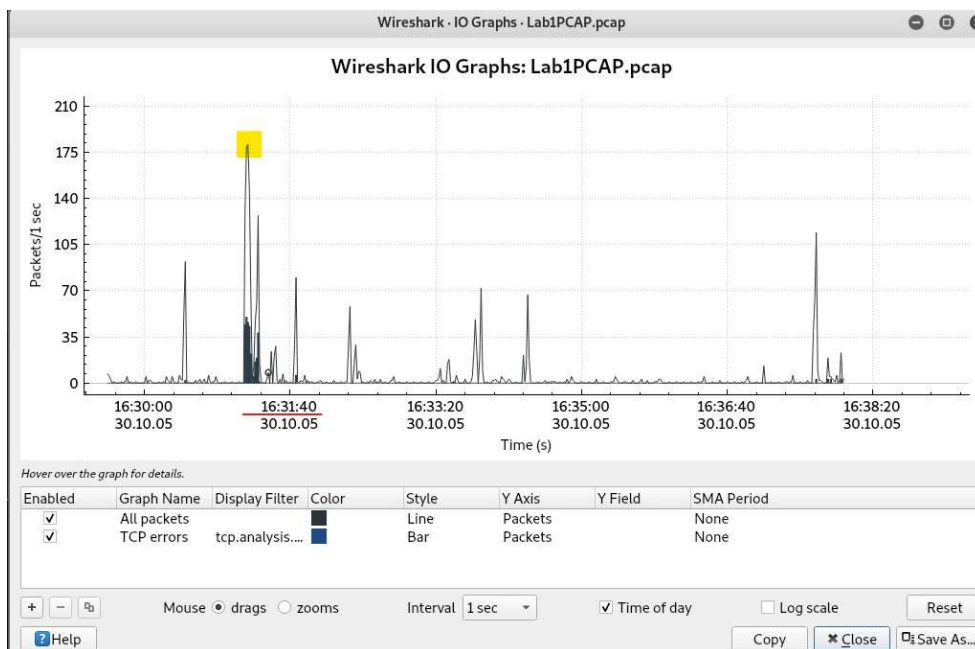


Figure 5: Bulk of data transmission.

d. What caused this transmission spike?

Following the peak of the IO graph, I was led to frame 545, a TCP packet whose TCP stream I followed. I was able to gather from the TCP stream that the user was accessing the RBFCU website, specifically requesting resources such as /images/sitemapbutton.jpg. The host header clearly indicates that the traffic was directed to rbfcu.org, confirming that the RBFCU site was actively being accessed during the session.

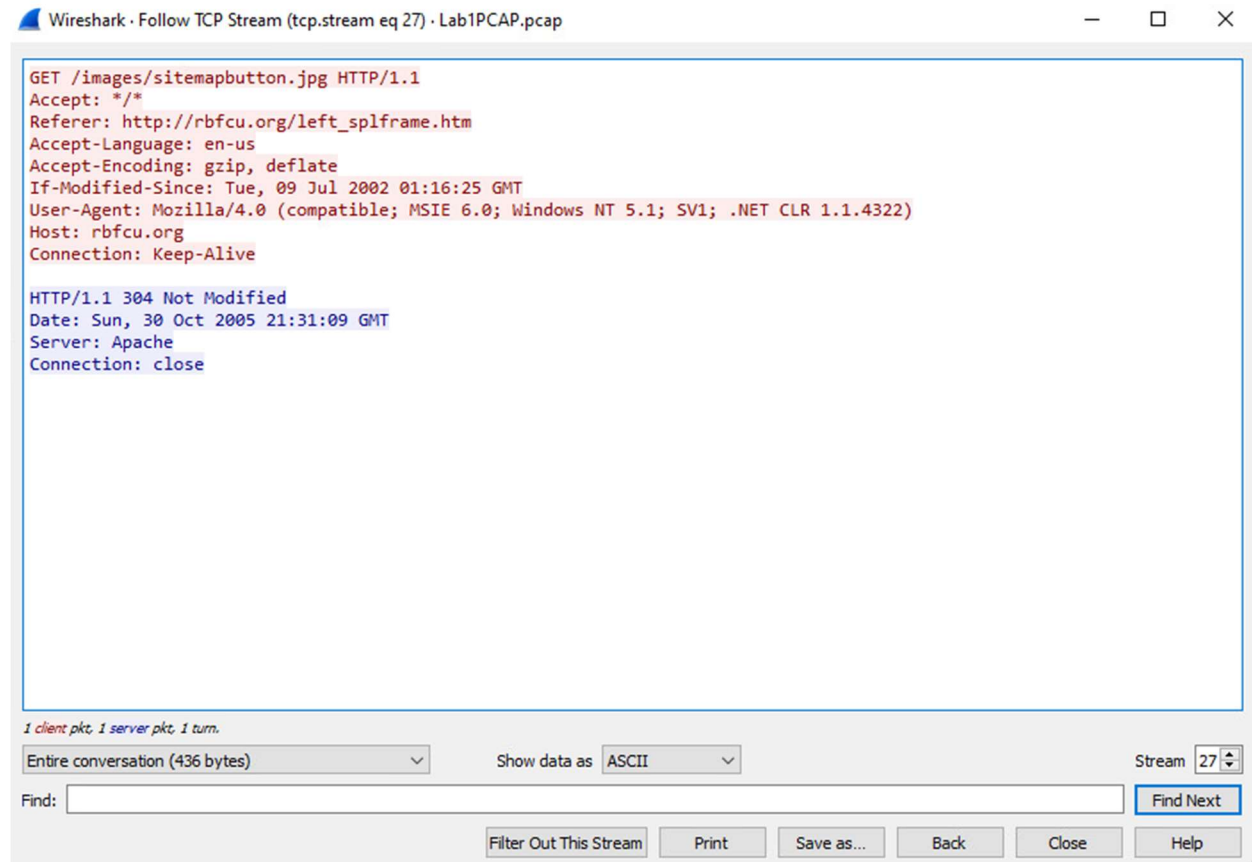


Figure 6: RBFCU Accessed (Frame 545).

Upon further analysis of the TCP stream, I was able to identify both the host IP and MAC address. In frame 1236, it was revealed that the host machine with IP address 172.16.1.25 was communicating with 216.166.24.20 to access rbfcu.org. This was further confirmed by Network Miner, which identified the Windows device 'Kaufman Upstairs' as the host with the IP address 172.16.1.35, while rbfcu.org uses 216.166.24.20.



Figure 7: Network Miner: Host and RBFCU IP Addresses.

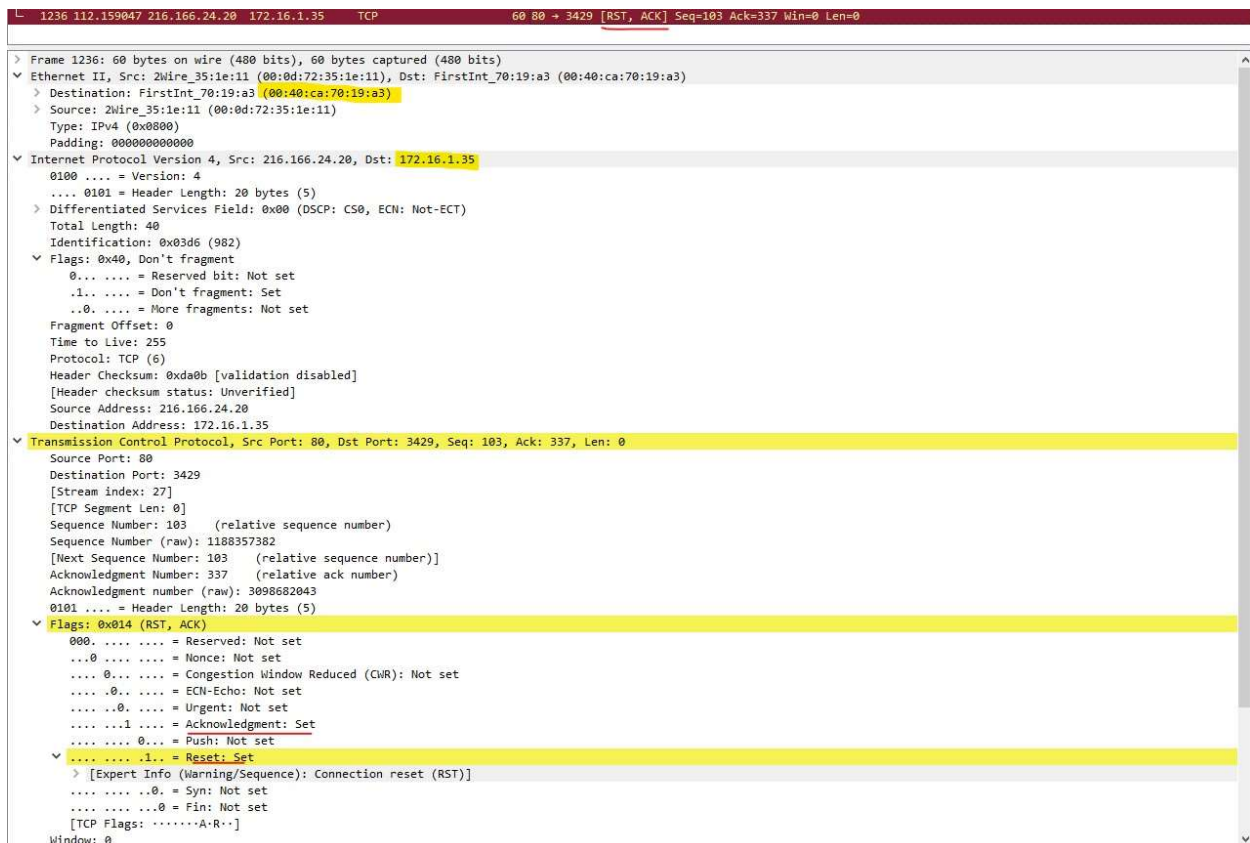


Figure 8: Wireshark: Host IP and MAC address (Frame 1236).

Continuing my investigation, I analyzed the packet capture in Snort. I sent the output to a text file in order to facilitate my investigation. Immediately I noticed there were 'Potentially Bad Traffic' flags throughout the packet capture. Next, I highlighted all the potentially malicious and priority 2 packets, to pull their IP addresses to analyze in Wireshark.

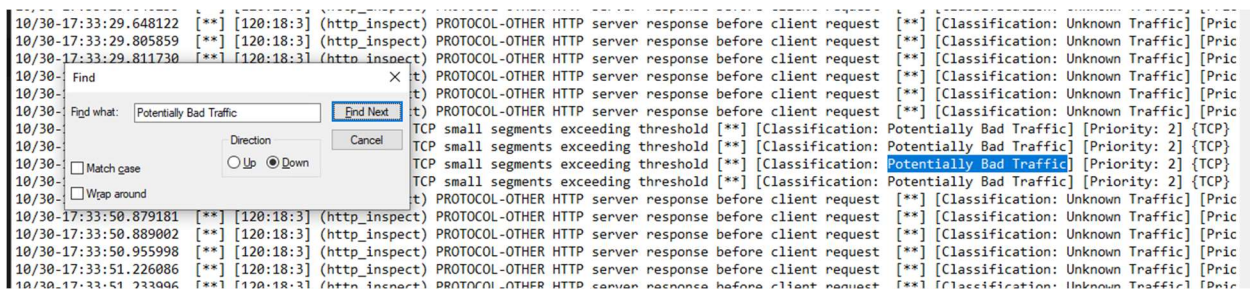


Figure 9: Snort: Potentially Bad Traffic.

Source IP	Destination IP
66.39.22.157	172.16.1.35
66.142.254.158	172.16.1.35
207.68.172.246	172.16.1.35
66.39.22.157	172.16.1.35
172.16.1.35	65.54.140.158
172.16.1.35	216.166.24.20
172.16.1.35	209.3.40.190
129.115.102.173	172.16.1.35
207.68.172.246	172.16.1.35
64.12.15.121	172.16.1.35

Figure 10: Table of IP addresses associated with possibly malicious packets.

Looking into the captured traffic associated with the listed IP's, I found several 'TCP DUP ACK' flags. This flag is sent when a receiver receives out of order packets but could also be a symptom of malicious activity.

734	97.044918	216.166.24.20	172.16.1.35	TCP	60 [TCP Dup ACK 728#1] 80 → 3442 [ACK] Seq=1 Ack=330 Win=65535
735	97.056833	216.166.24.20	172.16.1.35	TCP	60 80 → 3442 [ACK] Seq=1 Ack=330 Win=65535
736	97.056896	216.166.24.20	172.16.1.35	TCP	60 [TCP Dup ACK 731#1] 80 → 3443 [ACK] Seq=1 Ack=329 Win=65535
737	97.058769	216.166.24.20	172.16.1.35	TCP	60 [TCP Previous segment not captured] 80 → 3443 [ACK] Seq=1 Ack=329 Win=65535
738	97.058825	172.16.1.35	216.166.24.20	TCP	54 [TCP Dup ACK 729#1] 3442 → 80 [ACK] Seq=330 Ack=103 Win=654
739	97.058857	216.166.24.20	172.16.1.35	TCP	155 [TCP Out-Of-Order] 80 → 3442 [PSH, ACK] Seq=329 Ack=103 Win=654
740	97.058948	172.16.1.35	216.166.24.20	TCP	54 3442 → 80 [ACK] Seq=330 Ack=103 Win=654
741	97.059145	172.16.1.35	216.166.24.20	TCP	54 3442 → 80 [FIN, ACK] Seq=330 Ack=103 Win=654
742	97.068618	216.166.24.20	172.16.1.35	TCP	60 80 → 3443 [ACK] Seq=1 Ack=329 Win=65535
743	97.068676	216.166.24.20	172.16.1.35	TCP	60 [TCP Previous segment not captured] 80 → 3443 [ACK] Seq=1 Ack=329 Win=65535
744	97.068713	172.16.1.35	216.166.24.20	TCP	54 [TCP Dup ACK 732#1] 3443 → 80 [ACK] Seq=330 Ack=103 Win=654
745	97.070612	216.166.24.20	172.16.1.35	TCP	155 [TCP Out-Of-Order] 80 → 3443 [PSH, ACK] Seq=329 Ack=103 Win=654
746	97.070736	172.16.1.35	216.166.24.20	TCP	54 3443 → 80 [ACK] Seq=329 Ack=103 Win=654

Figure 11: Wireshark: DUP ACK packets from associated IP addresses.

ARP spoofing involves an attacker sending falsified ARP messages to a network, causing a victim's machine to associate the attacker's MAC address with the IP address of another device, typically the gateway. This misdirection allows the attacker to intercept and manipulate network traffic, which can then be exploited to hijack TCP sessions and gain unauthorized access to communications between the victim and other network services. Additionally, duplicate acknowledgments (ACKs) generated during TCP hijacking can indicate communication disruption. I found further proof of ARP spoofing in Network Miner under 'Anomalies'.

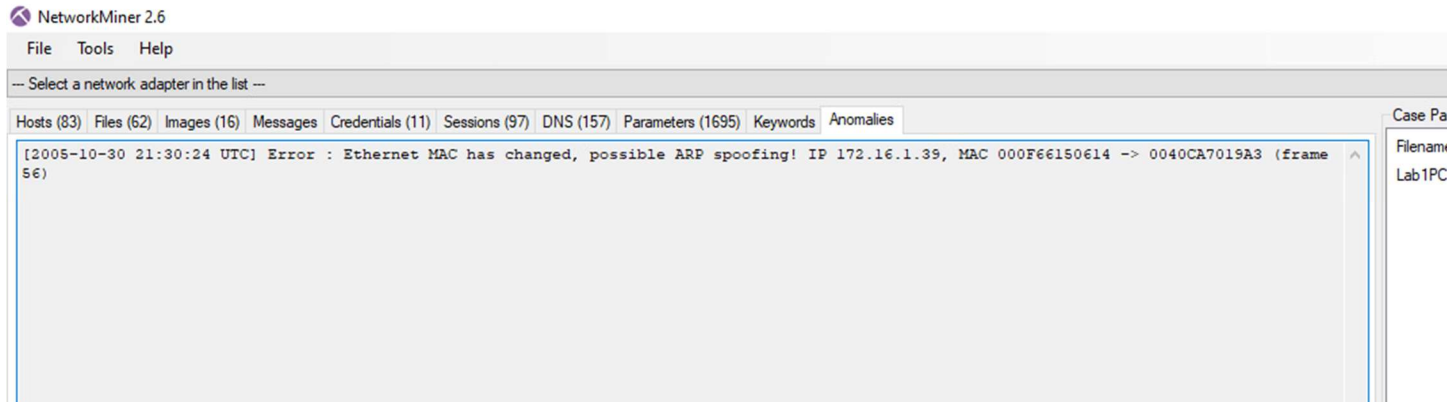


Figure 12: Network Miner: ARP Spoofing Anomaly.

The alert claims that on frame 56, the device whose IP address is 172.16.1.39 changed its MAC address from 00:0f:66:15:06:14 to 00:40:ca:70:19. To validate the alert, I further analyzed the traffic in Wireshark. I found that the device with IP 172.16.1.39 originally had a MAC address of 00:0f:66:15:06:14, but it later changed to 00:40:ca:70:19:a3. Additionally, the MAC address 00:40:ca:70:19:a3 is also associated with IP address 172.16.1.35, indicating a duplicate assignment. This is characteristic of ARP spoofing, a technique used to associate multiple IP addresses with a single MAC address, allowing an attacker to intercept or manipulate network traffic.

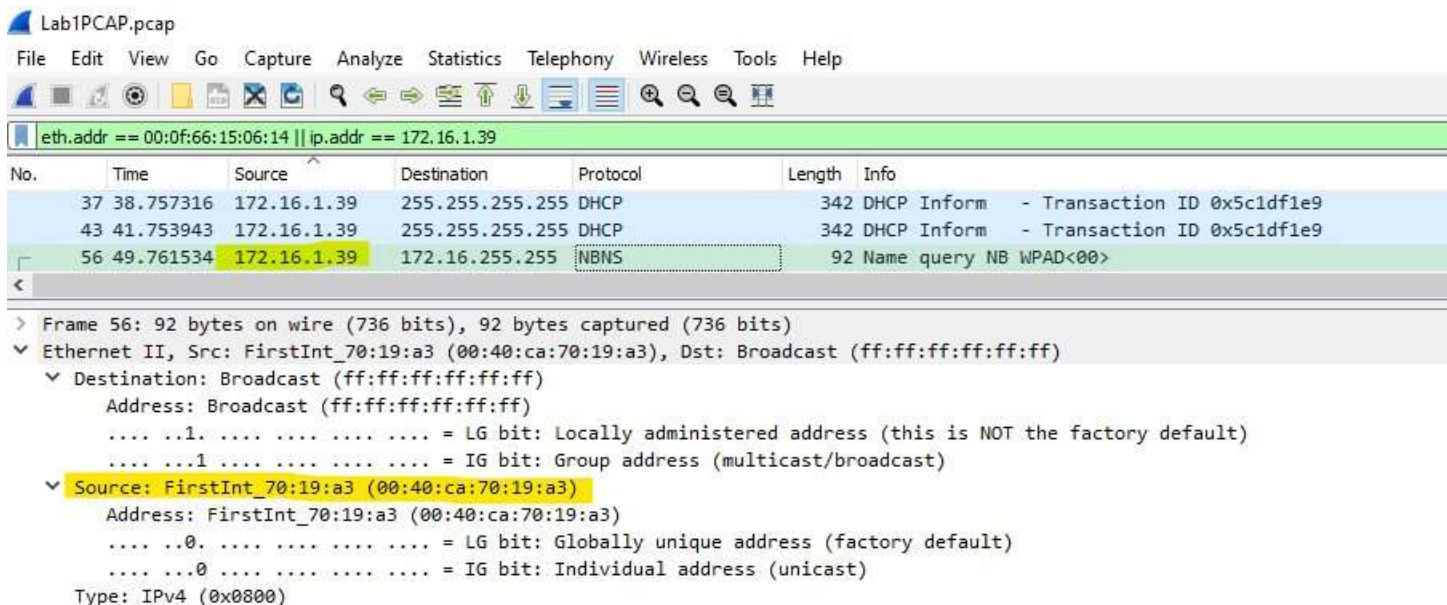


Figure 14: 172.16.1.39's MAC address 00:40:ca:70:19:a3.

From figure 14, you can see that device with the 172.16.1.39 address is tied to a MAC address of 00:40:ca:70:19:a3. This is the same MAC address used by 172.16.1.35 as shown by the ARP messages.

Lab1PCAP.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

eth.src == 00:40:ca:70:19:a3

No.	Time	Source	Destination	Protocol	Length	Info
2439	503.139001	172.16.1.35	152.163.15.208	TCP	54	3742 → 5190 [ACK] Seq=4296 Ack=160059 Win=6553
2442	503.154840	172.16.1.35	152.163.15.208	TCP	54	3742 → 5190 [ACK] Seq=4296 Ack=162779 Win=6553
2444	503.281641	172.16.1.35	152.163.15.208	TCP	54	3742 → 5190 [ACK] Seq=4296 Ack=164139 Win=6553
2446	503.482844	172.16.1.35	152.163.15.208	TCP	54	3742 → 5190 [ACK] Seq=4296 Ack=164590 Win=6508
2447	505.414281	172.16.1.35	172.16.255.255	TiVoConnect	182	Discovery Beacon KAUFMANUPSTAIRS ({9625E281-0A...
56	49.761534	172.16.1.39	172.16.255.255	NBNS	92	Name query NB WPAD<00>
59	50.503383	172.16.1.39	172.16.255.255	NBNS	92	Name query NB WPAD<00>
61	51.253508	172.16.1.39	172.16.255.255	NBNS	92	Name query NB WPAD<00>
1636	230.040109	172.16.1.39	172.16.255.255	BROWSER	258	Domain/Workgroup Announcement MSHOME, NT Works
1887	264.703966	172.16.1.39	172.16.255.255	BROWSER	216	Get Backup List Request
1888	264.704064	172.16.1.39	172.16.255.255	NBNS	92	Name query NB MSHOME<1b>
1890	265.453560	172.16.1.39	172.16.255.255	NBNS	92	Name query NB MSHOME<1b>
1891	266.203679	172.16.1.39	172.16.255.255	NBNS	92	Name query NB MSHOME<1b>
6	0.947943	FirstInt_70:...	2Wire_35:1e:11	ARP	42	172.16.1.35 is at 00:40:ca:70:19:a3
23	13.630707	FirstInt_70:...	2Wire_35:1e:11	ARP	42	172.16.1.35 is at 00:40:ca:70:19:a3
50	44.085457	FirstInt_70:...	2Wire_35:1e:11	ARP	42	172.16.1.35 is at 00:40:ca:70:19:a3
231	74.530340	FirstInt_70:...	2Wire_35:1e:11	ARP	42	172.16.1.35 is at 00:40:ca:70:19:a3
1202	104.973953	FirstInt_70:...	2Wire_35:1e:11	ARP	42	172.16.1.35 is at 00:40:ca:70:19:a3
1422	135.428690	FirstInt_70:...	2Wire_35:1e:11	ARP	42	172.16.1.35 is at 00:40:ca:70:19:a3
1455	165.883419	FirstInt_70:...	2Wire_35:1e:11	ARP	42	172.16.1.35 is at 00:40:ca:70:19:a3
1610	196.338274	FirstInt_70:...	2Wire_35:1e:11	ARP	42	172.16.1.35 is at 00:40:ca:70:19:a3
1624	227.833478	FirstInt_70:...	2Wire_35:1e:11	ARP	42	172.16.1.35 is at 00:40:ca:70:19:a3
1885	257.227652	FirstInt_70:...	2Wire_35:1e:11	ARP	42	172.16.1.35 is at 00:40:ca:70:19:a3
1945	287.682443	FirstInt_70:...	2Wire_35:1e:11	ARP	42	172.16.1.35 is at 00:40:ca:70:19:a3
2052	318.137201	FirstInt_70:...	2Wire_35:1e:11	ARP	42	172.16.1.35 is at 00:40:ca:70:19:a3
2069	348.591991	FirstInt_70:...	2Wire_35:1e:11	ARP	42	172.16.1.35 is at 00:40:ca:70:19:a3
2092	379.046732	FirstInt_70:...	2Wire_35:1e:11	ARP	42	172.16.1.35 is at 00:40:ca:70:19:a3
2105	409.501523	FirstInt_70:...	2Wire_35:1e:11	ARP	42	172.16.1.35 is at 00:40:ca:70:19:a3
2120	439.956280	FirstInt_70:...	2Wire_35:1e:11	ARP	42	172.16.1.35 is at 00:40:ca:70:19:a3
2127	450.478948	FirstInt_70:...	2Wire_35:1e:11	ARP	42	172.16.1.35 is at 00:40:ca:70:19:a3
2146	470.420890	FirstInt_70:...	2Wire_35:1e:11	ARP	42	172.16.1.35 is at 00:40:ca:70:19:a3
2421	500.875658	FirstInt_70:...	2Wire_35:1e:11	ARP	42	172.16.1.35 is at 00:40:ca:70:19:a3

<

- > Frame 2447: 182 bytes on wire (1456 bits), 182 bytes captured (1456 bits)
- ▼ Ethernet II, Src: FirstInt_70:19:a3 (00:40:ca:70:19:a3), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 - > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
 - ▼ Source: FirstInt_70:19:a3 (00:40:ca:70:19:a3)
 - Address: FirstInt_70:19:a3 (00:40:ca:70:19:a3)
 -0. = LG bit: Globally unique address (factory default)
 -0. = IG bit: Individual address (unicast)
 - Type: IPv4 (0x0800)
 - > Internet Protocol Version 4, Src: 172.16.1.35, Dst: 172.16.255.255
 - > User Datagram Protocol, Src Port: 3756, Dst Port: 2190
 - > TiVoConnect Discovery Protocol, KAUFMANUPSTAIRS ({9625E281-0AD4-4D95-8735-F59AB074E79A})

Figure 15: Duplicate MAC addresses associated with 172.16.1.25.

Therefore, it can be concluded that ARP spoofing is occurring on the network, as the device with IP address 172.16.1.39 has assumed the same MAC address as 172.16.1.35. Analyzing the traffic created by

172.16.1.35, I noticed a series of FTP packets and I followed their TCP stream. A device with an IP of 66.39.22.157 connected to the 172.16.1.35 IP and signed in anonymously to a Linux FTP server to transfer data between the devices.



Figure 16: Linux FTP.

```

Wireshark - Follow TCP Stream (tcp.stream eq 83) - Lab1PCAP.pcap
220 linux-wlan.org NcFTPd Server (licensed copy) ready.
USER anonymous
331 Guest login ok, send your complete e-mail address as password.
PASS IEUser@
230-You are user #6 of 32 simultaneous users allowed.
230-
230 Logged in anonymously.
opts utf8 on
501 Option not recognized.
syst
215 UNIX Type: L8
site help
211-The following SITE commands are recognized:
211-  BUFSIZE
211-  CHMOD
211-  DATE
211-  DF
211-  QUOTA
211-  RBUFSIZ
211-  RBUFSZ
211-  RETRBUFSIZE
211-  SBUFSIZ
211-  SBUFSZ
211-  STORBUFSIZE
211-  SYMLINK
211-  UMASK
211-  UTIME
211
PWD
257 "/" is cwd.
CWD /pub/linux-wlan-ng/
250 "/pub/linux-wlan-ng" is new cwd.
TYPE A
200 Type okay.
PASV
227 Entering Passive Mode (66,39,22,157,238,167)
LIST
150 Data connection accepted from 68.92.158.179:3603; transfer starting.
226 Listing completed.

```

Figure 17: Linux FTP TCP Stream, Data transfer.

Therefore, we can conclude that host 172.16.1.35 was signed into to intercept and record RBFCU traffic. The threat actor is likely using ARP spoofing to view their RBFCU information and thus creating the spike in network traffic shown on the I/O graph.

e. Were any Internet Service Provider sites accessed? If so, which ones? What accounts?

The ISPs accessed in the screenshot are AOL (dial.internet.aol.com), MSN (msn.com), and Yahoo (ssl.vip.scd.yahoo.com). Accounts related to AOL, MSN, and Yahoo were accessed.



Figure 18: Network Miner: Internet Service Providers.

f. What is the name of the host computer? Its IP address?

The name of the host computer is Kaufman Upstairs, and it has an IP address of 172.16.1.35. This is the machine that connected to a Linux FTP server.



Figure 19: Network Miner: Host Machine.

g. What operating system is it using?

The host machine is using Windows, as shown in figure 19. I also determined the operating system by analyzing the packet time to live. The host machine's packet time to live was 128, which is the default used by Windows.

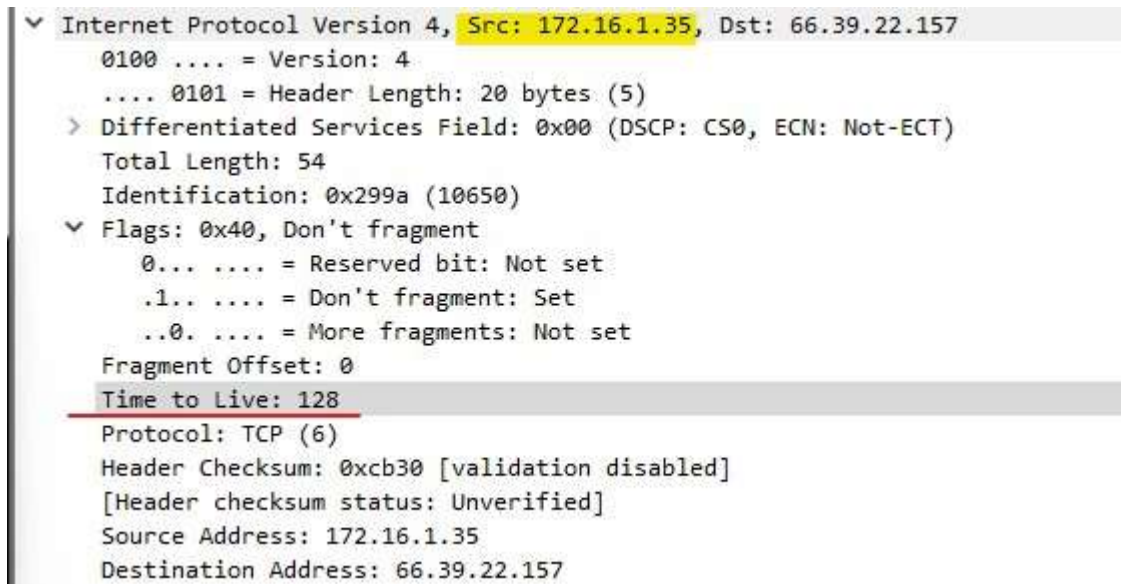


Figure 20: Windows TTL.

h. What device names are on the local network?

In Wireshark, I found the devices on the local network by going to 'Statistics' then "Endpoints' and viewing the captured MAC addresses. Next, I checked the 'name resolution' to view the name of the

devices on the network. On the network, the following devices are connected: Linksys, 2Wire, Cisco-Li, and FirstInt.

Wireshark · Endpoints · Lab1PCAP.pcap

Ethernet · 6	IPv4 · 28	IPv6	TCP · 124	UDP · 133		
Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
LinksysG_26:d6:22	14	3172	14	3172	0	
2Wire_35:1e:11	2,297	781k	1,281	675k	1,016	
Cisco-Li_15:06:14	2	684	2	684	0	
FirstInt_70:19:a3	2,328	801k	1,152	131k	1,176	
IPv4mcast_fb	6	1308	0	0	6	
Broadcast	251	34k	0	0	251	

☒ Name resolution ☐ Limit to display filter

Figure 21: Wireshark: Endpoints.

i. Did I access any other computers on the local area network?

No other computers were accessed on the local area network as revealed by the Sessions window in network Miner. The host machine connected outside of the network to IP's like 66.39.22.157, 207.68.172.246, 65.54.140.158, and other outside servers, but none within the local network.

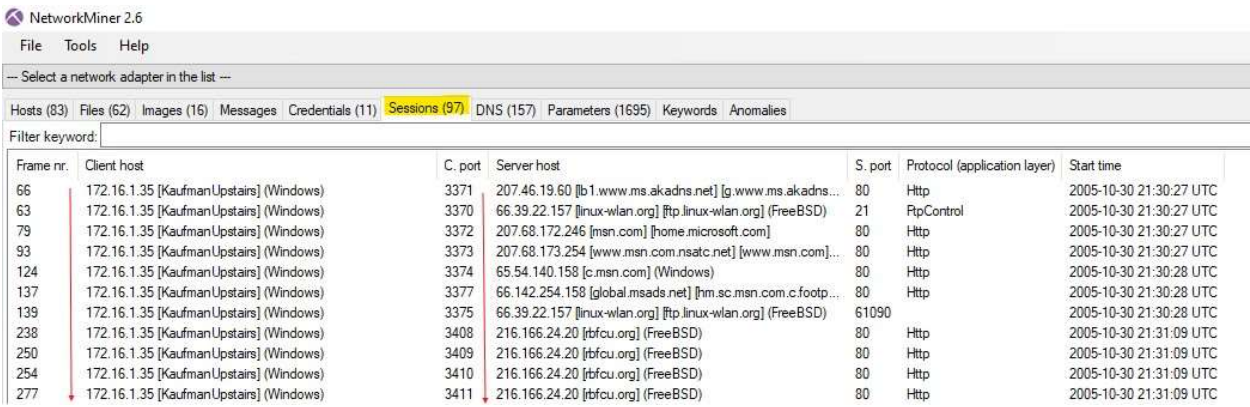


Figure 22: Network Miner: Session traffic.

CONCLUSION

The packet capture reveals a series of suspicious events on the local network, suggesting possible malicious activity. Over an 8-minute and 25-second span, the network traffic shows signs of ARP spoofing due to the presence of duplicate MAC addresses, raising concerns about traffic interception. A device named "Kaufman Upstairs" (IP 172.16.1.35) appeared to be involved in accessing the RBFCU website, generating a spike in data transmission. Further analysis using tools like Wireshark, Network Miner, and Snort confirmed ARP spoofing, where the attacker manipulated MAC addresses to hijack communications. The compromised host was connected to a Linux FTP server, and it seems the attacker used ARP spoofing to intercept and monitor traffic, particularly targeting the RBFCU traffic, while also accessing other services like AOL, MSN, and Yahoo. The findings point to deliberate efforts to exploit the network using ARP spoofing and session hijacking.

REFERENCES

"8.4. The 'Protocol Hierarchy' Window." *Wireshark User's Guide*, The Wireshark Foundation, https://www.wireshark.org/docs/wsug_html_chunked/ChStatHierarchy.html. Accessed 9 Sept. 2024.

"What Is Duplicate ACK? When Does It Occur?" *Stack Overflow*, <https://stackoverflow.com/questions/48148820/what-is-duplicate-ack-when-does-it-occur>. Accessed 15 Sept. 2024.

"Lab 5 - ARP Spoofing and TCP Hijacking." *EEE 466 Labs*, Knight Segfaults, <https://knight.segfaults.net/EEE466Labs/Lab%205/Lab%205%20-%20ARP%20spoofing%20and%20TCP%20hijacking.html>. Accessed 15 Sep. 2024.

"Expert Information." *Wireshark User's Guide*, The Wireshark Foundation,
https://www.wireshark.org/docs/wsug_html_chunked/ChAdvExpert.html. Accessed 14 Sept. 2024.