2024

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.

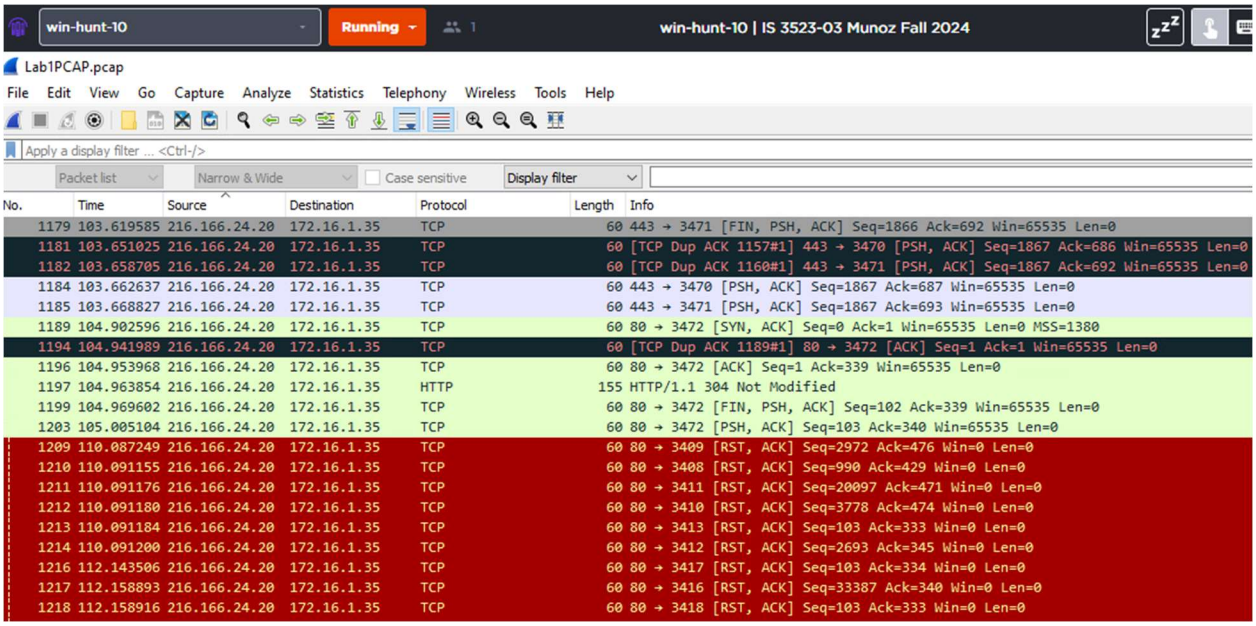


Figure 1: Wireshark Traffic.

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.

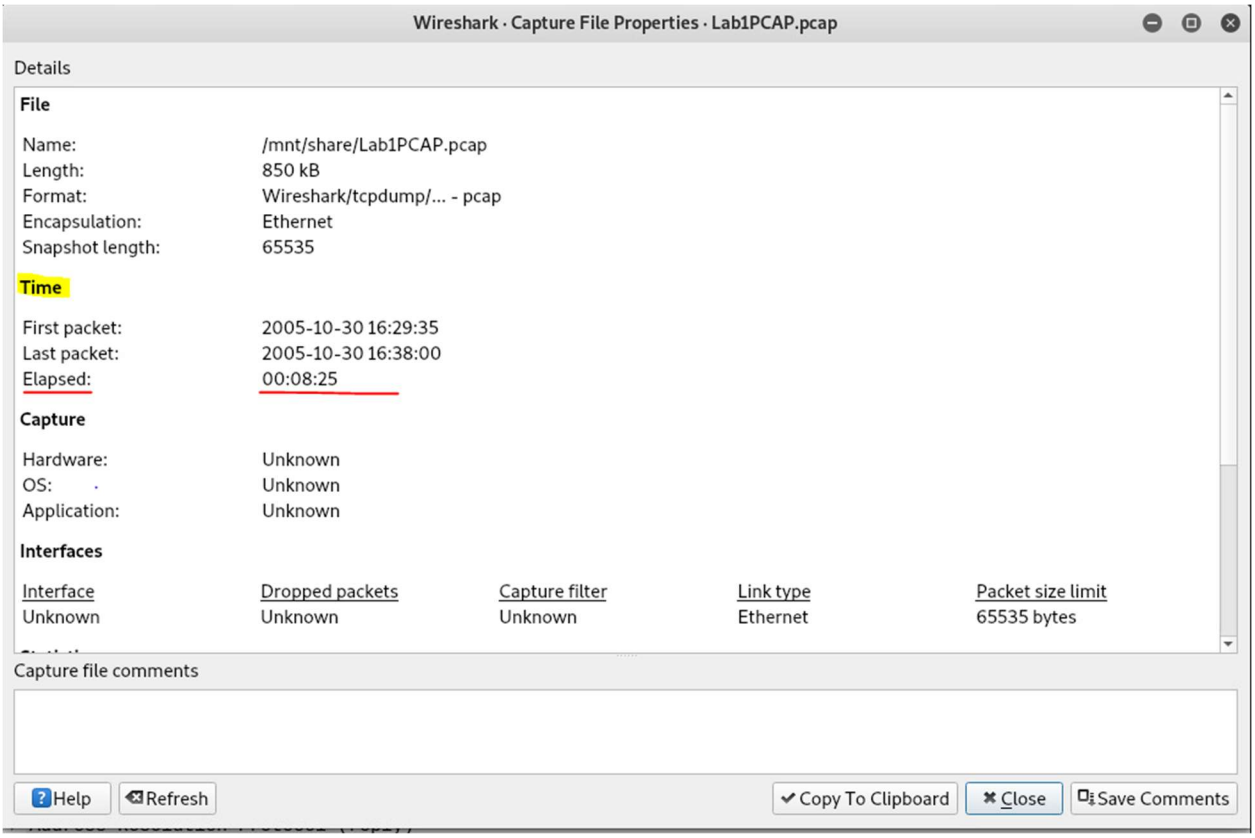


Figure 2: Duration of Packet Capture.

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.

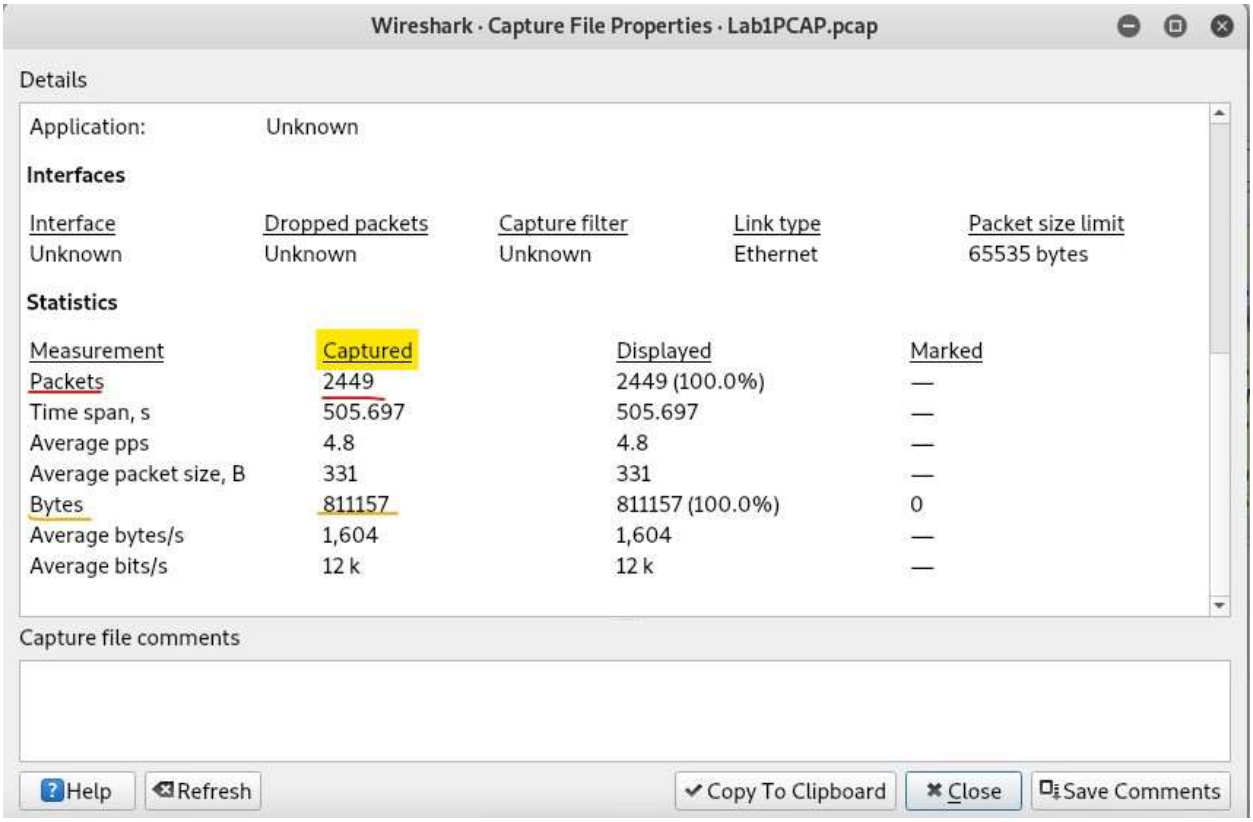


Figure 3: Bytes Captured.

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%).

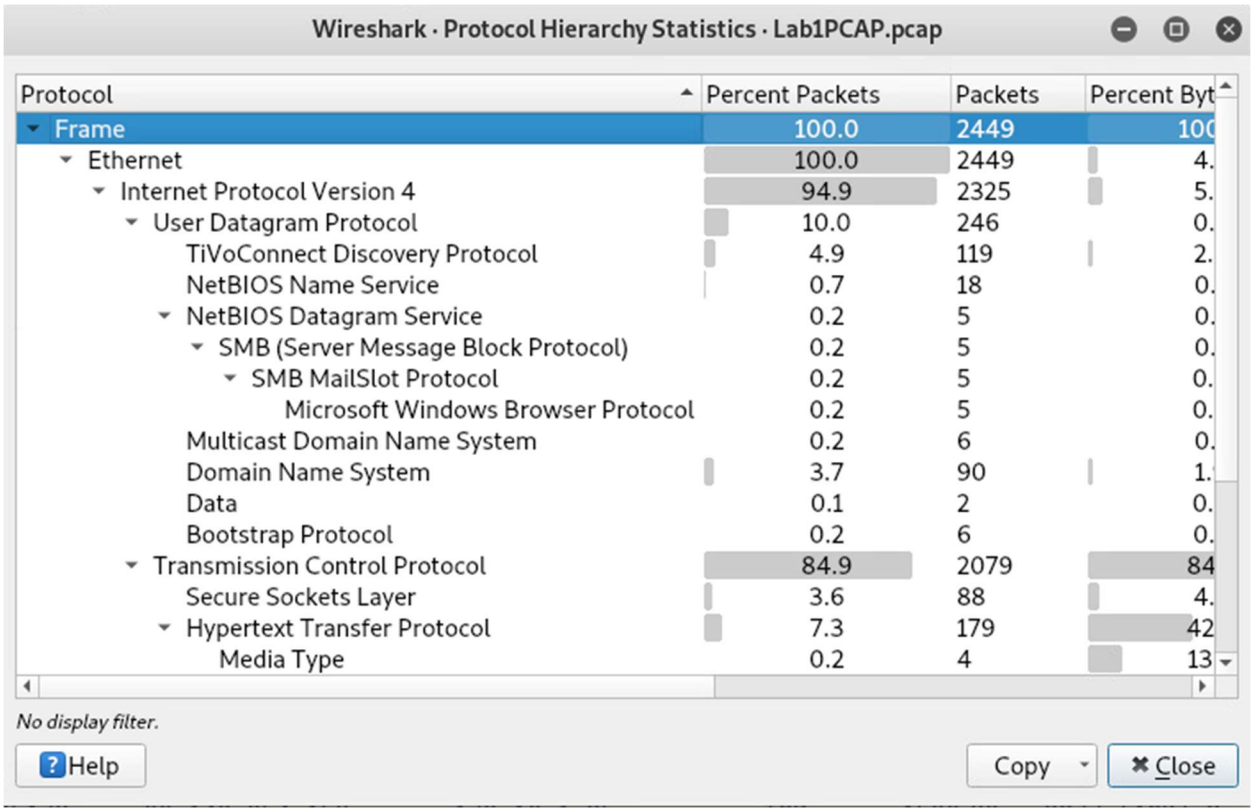


Figure 4: Protocols Observed.

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.

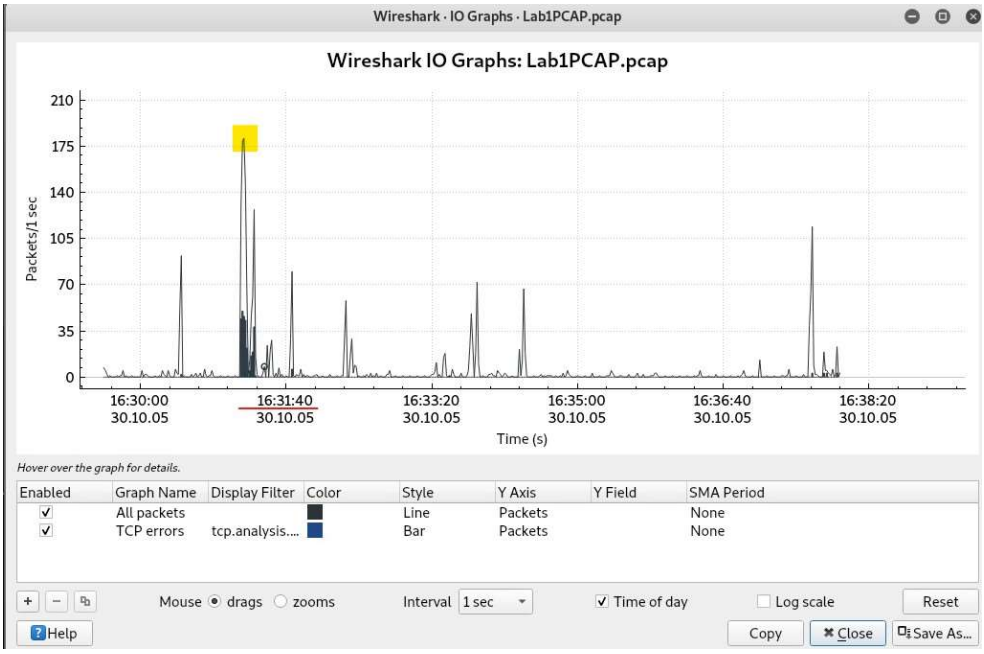


Figure 5: Bulk of data transmission.

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.

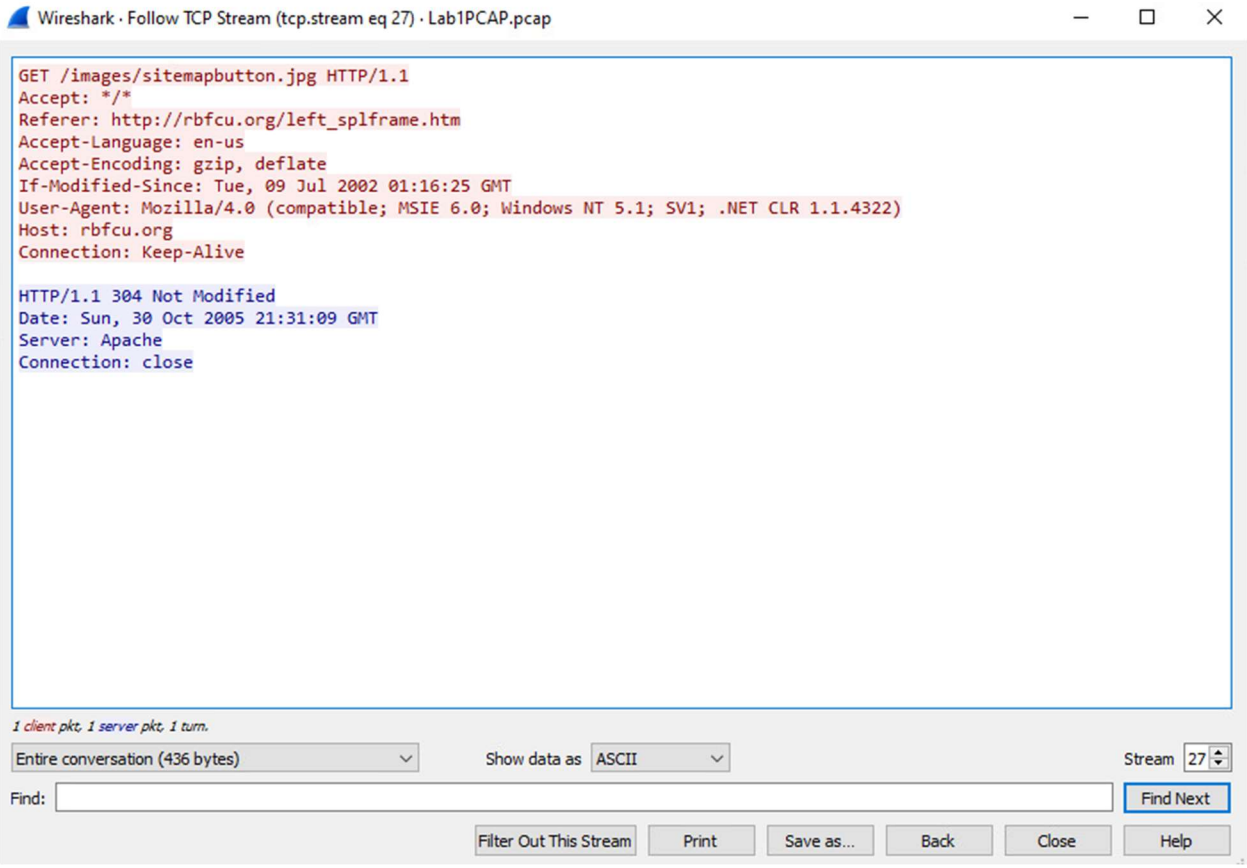


Figure 6: RBFCU Accessed (Frame 545).

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.

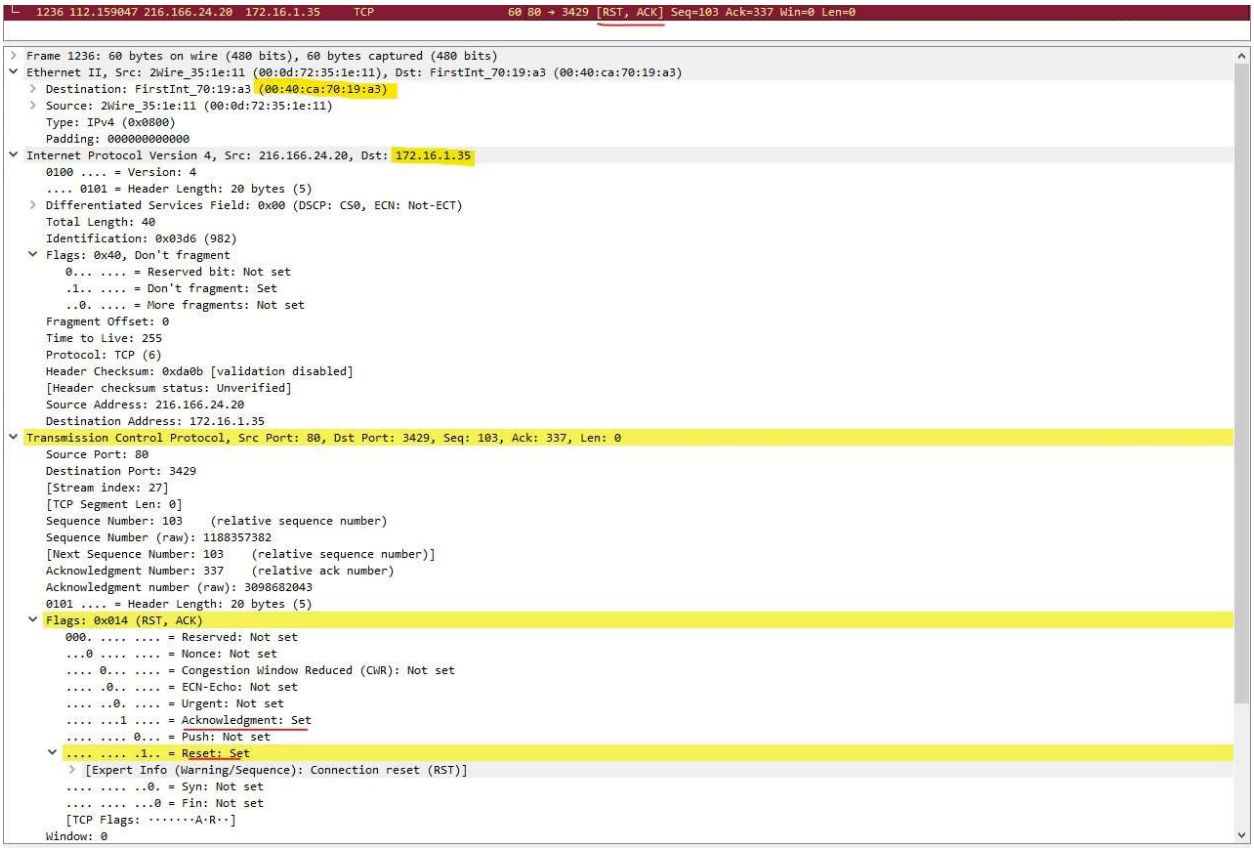


Figure 8: Wireshark: Host IP and MAC address (Frame 1236).

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.

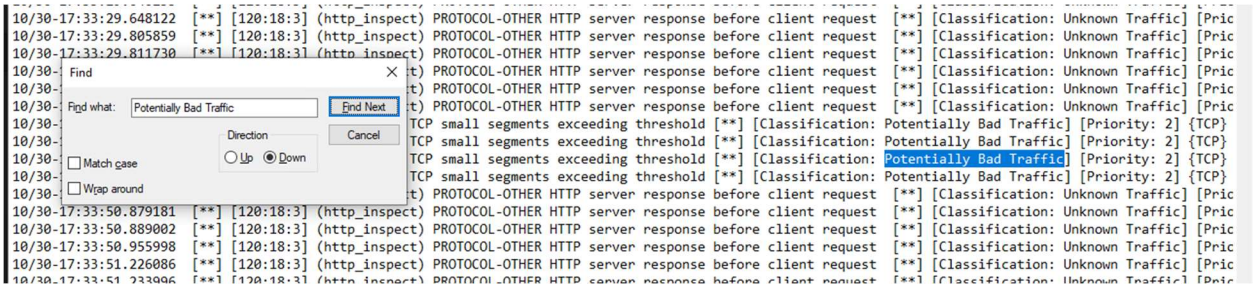


Figure 9: Snort: Potentially Bad Traffic.

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.

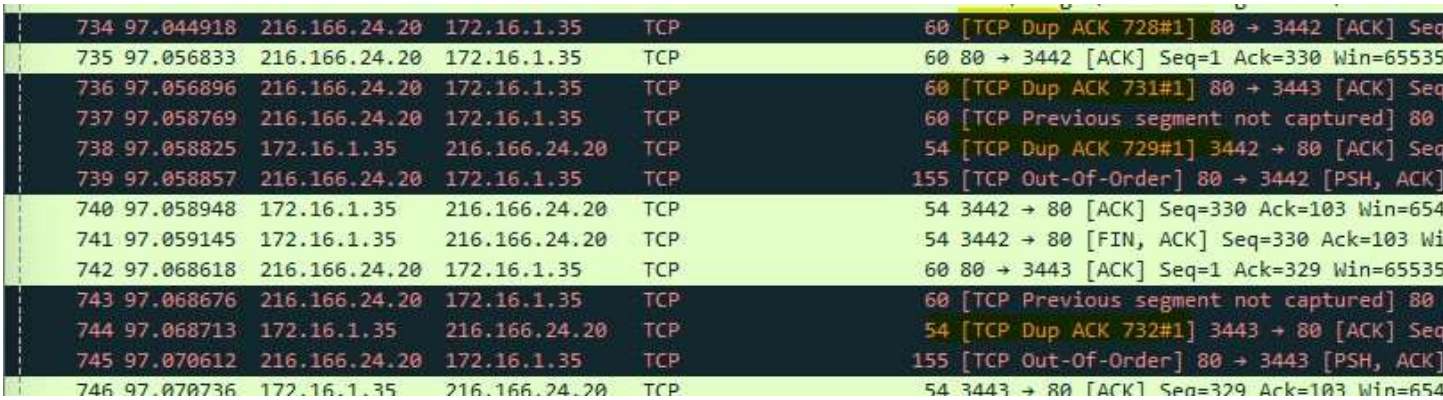


Figure 11: Wireshark: DUP ACK packets from associated IP addresses.

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'.

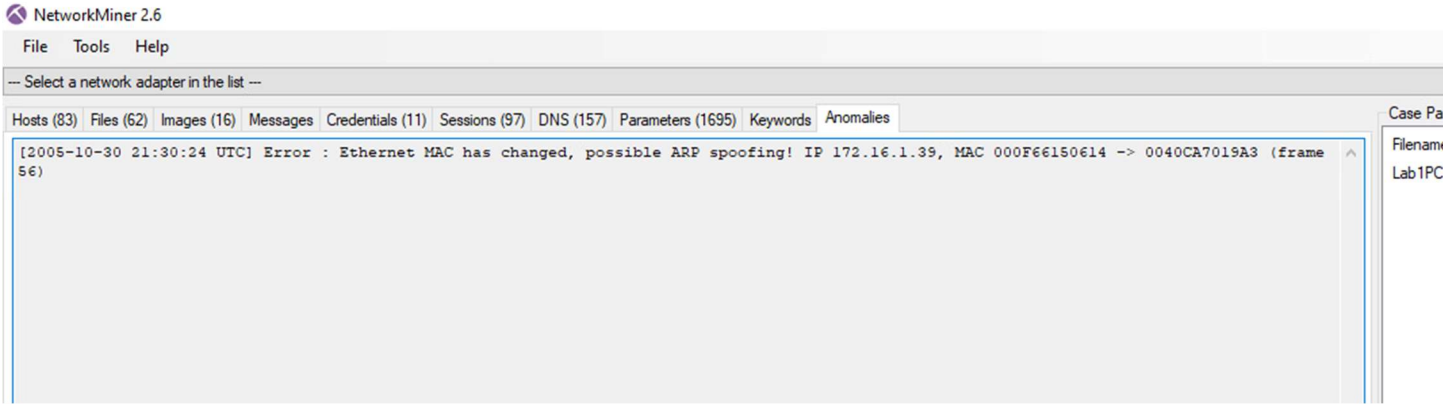


Figure 12: Network Miner: ARP Spoofing Anomaly.

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.

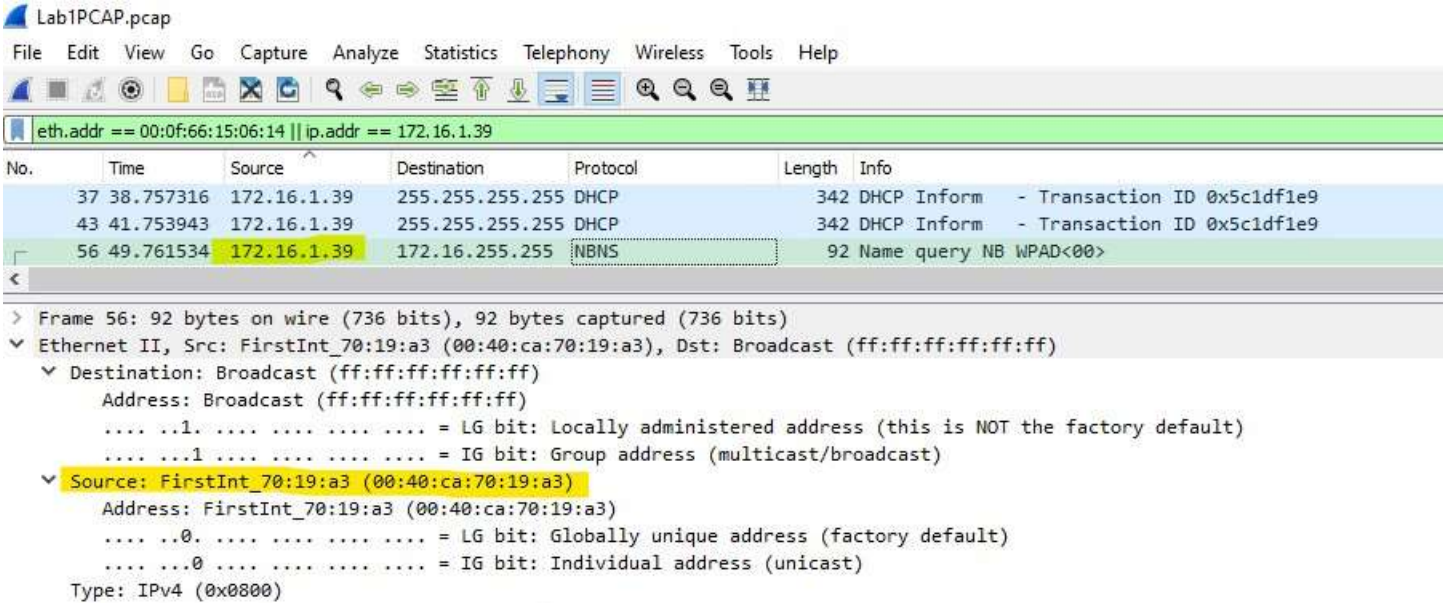


Figure 14: 172.16.1.39's MAC address 00:40:ca:70:19:a3.

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.

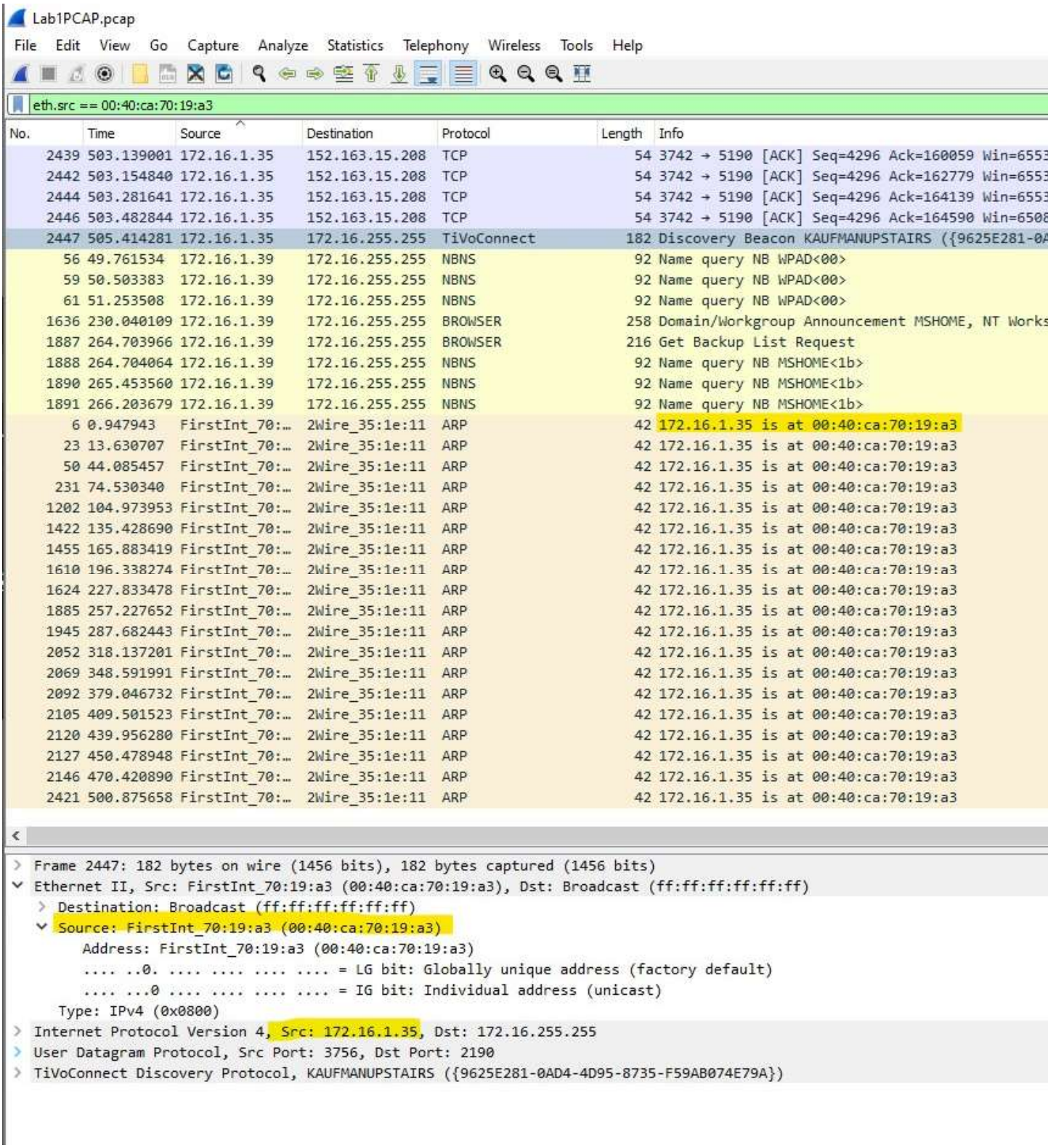


Figure 15: Duplicate MAC addresses associated with 172.16.1.25.

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.

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.

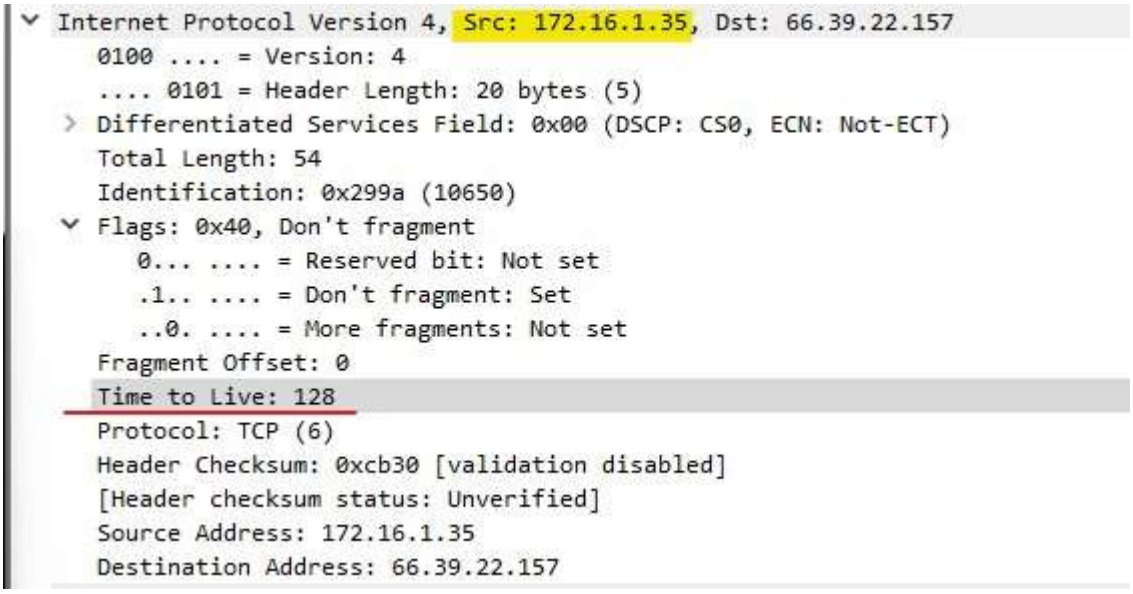


Figure 20: Windows TTL.

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

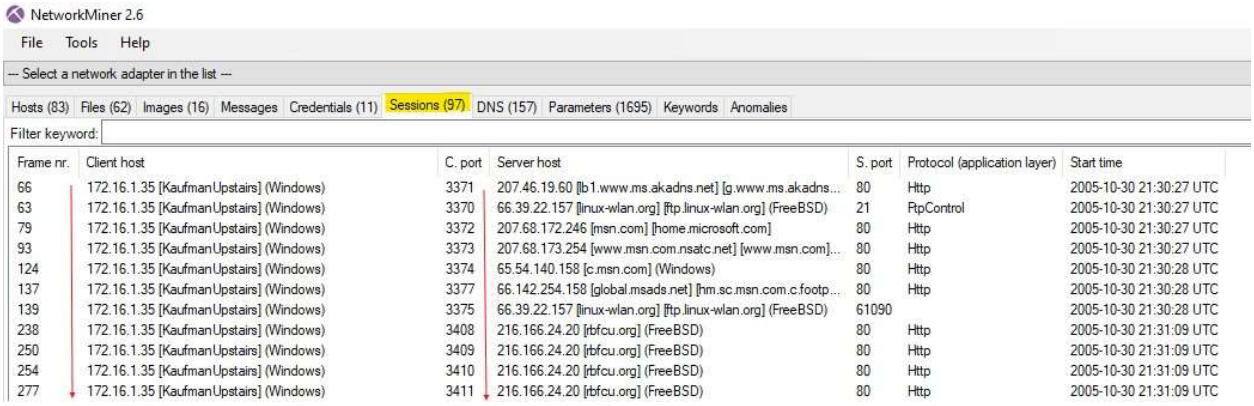
Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
LinksysG_26:d6:22	14	3172	14	3172	0	
2Wire_35:1e:11	2,297	781k	1,281	675k	1,016	
Cisco-Li_15:06:14	2	684	2	684	0	
FirstInt_70:19:a3	2,328	801k	1,152	131k	1,176	
IPv4mcast_fb	6	1308	0	0	6	
Broadcast	251	34k	0	0	251	

☑ 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.


NetworkMiner 2.6

File Tools Help

-- Select a network adapter in the list --

Hosts (83) Files (62) Images (16) Messages Credentials (11) Sessions (97) DNS (157) Parameters (1695) Keywords Anomalies

Filter keyword:

Frame nr.	Client host	C. port	Server host	S. port	Protocol (application layer)	Start time
66	172.16.1.35 [KaufmanUpstairs] (Windows)	3371	207.46.19.60 [lb1.www.ms.akadns.net] [g.www.ms.akadns...	80	Http	2005-10-30 21:30:27 UTC
63	172.16.1.35 [KaufmanUpstairs] (Windows)	3370	66.39.22.157 [linux-wlan.org] [ftp.linux-wlan.org] (FreeBSD)	21	RpcControl	2005-10-30 21:30:27 UTC
79	172.16.1.35 [KaufmanUpstairs] (Windows)	3372	207.68.172.246 [msn.com] [home.microsoft.com]	80	Http	2005-10-30 21:30:27 UTC
93	172.16.1.35 [KaufmanUpstairs] (Windows)	3373	207.68.173.254 [www.msn.com.nsatc.net] [www.msn.com]...	80	Http	2005-10-30 21:30:27 UTC
124	172.16.1.35 [KaufmanUpstairs] (Windows)	3374	65.54.140.158 [c.msn.com] (Windows)	80	Http	2005-10-30 21:30:28 UTC
137	172.16.1.35 [KaufmanUpstairs] (Windows)	3377	66.142.254.158 [global.msads.net] [hm.sc.msn.com.c.footp...	80	Http	2005-10-30 21:30:28 UTC
139	172.16.1.35 [KaufmanUpstairs] (Windows)	3375	66.39.22.157 [linux-wlan.org] [ftp.linux-wlan.org] (FreeBSD)	61090		2005-10-30 21:30:28 UTC
238	172.16.1.35 [KaufmanUpstairs] (Windows)	3408	216.166.24.20 [rbfcu.org] (FreeBSD)	80	Http	2005-10-30 21:31:09 UTC
250	172.16.1.35 [KaufmanUpstairs] (Windows)	3409	216.166.24.20 [rbfcu.org] (FreeBSD)	80	Http	2005-10-30 21:31:09 UTC
254	172.16.1.35 [KaufmanUpstairs] (Windows)	3410	216.166.24.20 [rbfcu.org] (FreeBSD)	80	Http	2005-10-30 21:31:09 UTC
277	172.16.1.35 [KaufmanUpstairs] (Windows)	3411	216.166.24.20 [rbfcu.org] (FreeBSD)	80	Http	2005-10-30 21:31:09 UTC

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.