Lab Assignment: Network Forensics

Siwei Zhang, Yanxiang Du, Yongjie Hao

May 21, 2024

1 Assignment 1: Introduction to Network Forensics

1.1 Evidence file: 1.pcap

1.1.1 What is/are the source(s) (IP address) of the suspicious traffic?

192.0.2.245, 192.0.2.196, 192.0.2.6, 192.0.2.25, 192.0.2.120, 192.0.2.83, 192.0.2.154, 192.0.2.253, 192.0.2.236

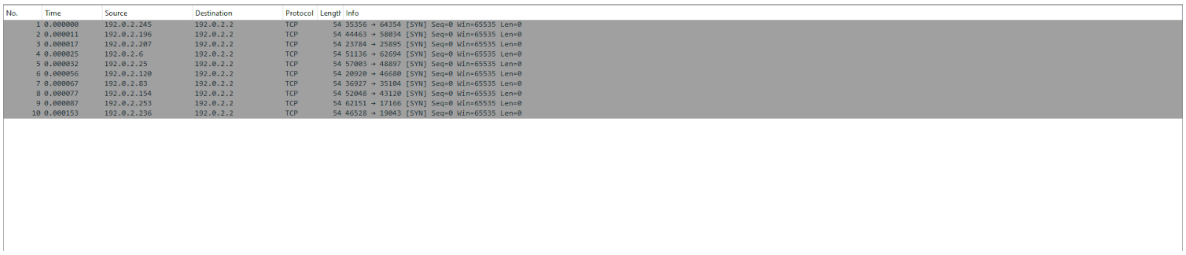


Figure 1: IP address of the suspicious traffic

1.1.2 What is the destination (IP address) of the suspicious traffic?

192.0.2.2

1.1.3 What is the transport layer protocol used?

TCP

1.1.4 What is/are the source port(s)?

35356, 44463, 23784, 51136, 57003

1.1.5 What is/are the destination port(s)?

64354, 58034, 25895, 62694, 48897, 35104, 43120, 17166, 19043

1.1.6 What conclusions can you draw from the type of the "attack"/activity illustrated by this pcap?

This pattern typically indicates a Port scanning attack and this would cause the SYN flood attack, which is a type of DDoS (Distributed Denial of Service) attack.

1.2 Evidence file: 2. pcap

1.2.1 What is the source(s) (MAC address) of the suspicious traffic?

00:11:22:33:44:55

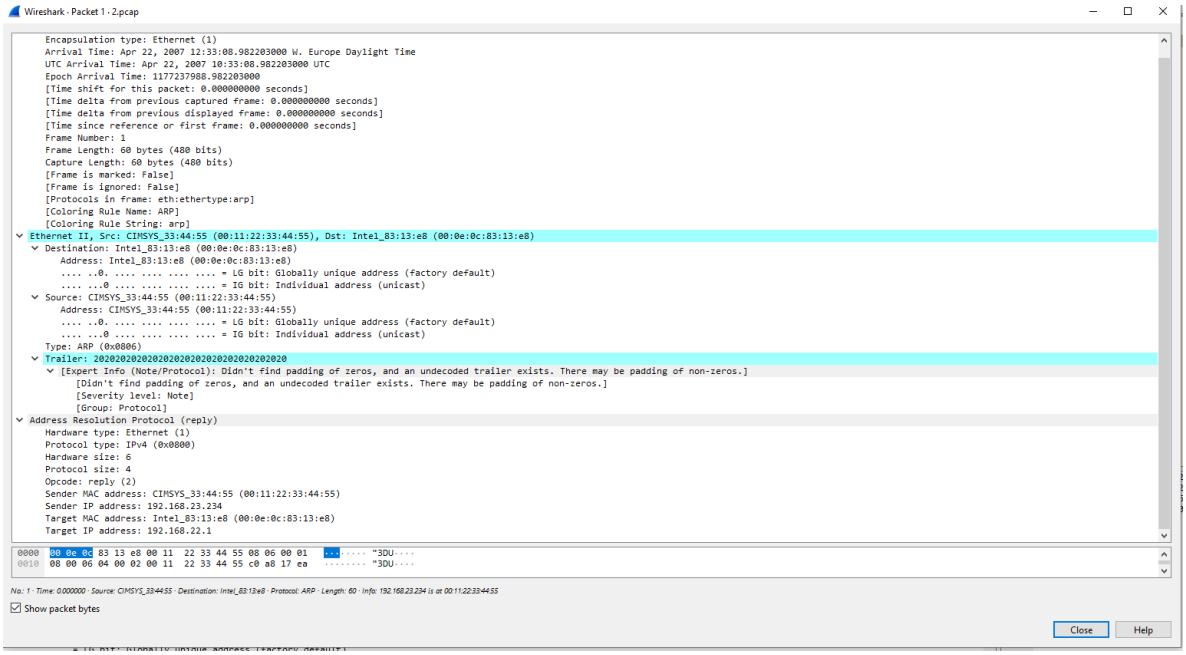


Figure 2: The details of the packet

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	CIMSYS_33:44:55	Intel_83:13:e8	ARP	60	192.168.23.234 is at 00:11:22:33:44:55
2	0.001909	CIMSYS_33:44:55	Broadcast	ARP	106	ARP Announcement for 192.168.23.234[Packet size limited during capture]
3	0.221830	CIMSYS_33:44:55	Intel_83:13:e8	ARP	60	192.168.22.238 is at 00:11:22:33:44:55
4	0.847522	CIMSYS_33:44:55	Intel_83:13:e8	ARP	60	192.168.22.238 is at 00:11:22:33:44:55
5	0.847529	CIMSYS_33:44:55	Broadcast	ARP	60	ARP Announcement for 192.168.22.238
6	1.131450	CIMSYS_33:44:55	Intel_83:13:e8	ARP	60	192.168.23.215 is at 00:11:22:33:44:55
7	1.691241	CIMSYS_33:44:55	Intel_83:13:e8	ARP	60	192.168.22.218 is at 00:11:22:33:44:55
8	1.692203	CIMSYS_33:44:55	Broadcast	ARP	60	ARP Announcement for 192.168.22.218
9	2.054293	CIMSYS_33:44:55	Intel_83:13:e8	ARP	60	192.168.22.113 is at 00:11:22:33:44:55
10	2.068087	CIMSYS_33:44:55	Intel_83:13:e8	ARP	60	192.168.22.250 is at 00:11:22:33:44:55
11	2.878756	CIMSYS_33:44:55	Intel_83:13:e8	ARP	60	192.168.23.234 is at 00:11:22:33:44:55
12	2.879755	CIMSYS_33:44:55	Broadcast	ARP	106	ARP Announcement for 192.168.23.234[Packet size limited during capture]
13	2.979726	CIMSYS_33:44:55	Intel_83:13:e8	ARP	60	192.168.22.250 is at 00:11:22:33:44:55
14	2.979732	CIMSYS_33:44:55	Broadcast	ARP	60	ARP Announcement for 192.168.22.250
15	3.338581	CIMSYS_33:44:55	Intel_83:13:e8	ARP	60	192.168.22.113 is at 00:11:22:33:44:55
16	3.830399	CIMSYS_33:44:55	Intel_83:13:e8	ARP	60	192.168.23.169 is at 00:11:22:33:44:55
17	3.830466	CIMSYS_33:44:55	Broadcast	ARP	60	ARP Announcement for 192.168.23.169
18	4.838000	CIMSYS_33:44:55	Intel_83:13:e8	ARP	60	192.168.22.113 is at 00:11:22:33:44:55
19	5.109974	CIMSYS_33:44:55	Intel_83:13:e8	ARP	60	192.168.23.234 is at 00:11:22:33:44:55
20	5.110876	CIMSYS_33:44:55	Broadcast	ARP	106	ARP Announcement for 192.168.23.234[Packet size limited during capture]
21	5.305808	CIMSYS_33:44:55	Intel_83:13:e8	ARP	60	192.168.22.237 is at 00:11:22:33:44:55
22	5.306774	CIMSYS_33:44:55	Broadcast	ARP	60	ARP Announcement for 192.168.22.237
23	6.225424	CIMSYS_33:44:55	Intel_83:13:e8	ARP	60	192.168.23.234 is at 00:11:22:33:44:55
24	6.226406	CIMSYS_33:44:55	Broadcast	ARP	106	ARP Announcement for 192.168.23.234[Packet size limited during capture]
25	6.337430	CIMSYS_33:44:55	Intel_83:13:e8	ARP	60	192.168.22.113 is at 00:11:22:33:44:55
26	6.536388	CIMSYS_33:44:55	Intel_83:13:e8	ARP	60	192.168.22.170 is at 00:11:22:33:44:55
27	6.536394	CIMSYS_33:44:55	Broadcast	ARP	60	ARP Announcement for 192.168.22.170
28	7.312028	CIMSYS_33:44:55	Intel_83:13:e8	ARP	60	192.168.22.190 is at 00:11:22:33:44:55
29	7.312994	CIMSYS_33:44:55	Broadcast	ARP	60	ARP Announcement for 192.168.22.190
30	7.341001	CIMSYS_33:44:55	Intel_83:13:e8	ARP	60	192.168.23.234 is at 00:11:22:33:44:55
31	7.341985	CIMSYS_33:44:55	Broadcast	ARP	106	ARP Announcement for 192.168.23.234[Packet size limited during capture]
32	7.515938	CIMSYS_33:44:55	Intel_83:13:e8	ARP	60	192.168.22.17 is at 00:11:22:33:44:55
33	7.516922	CIMSYS_33:44:55	Broadcast	ARP	60	ARP Announcement for 192.168.22.17
34	7.837785	CIMSYS_33:44:55	Intel_83:13:e8	ARP	60	192.168.22.113 is at 00:11:22:33:44:55
35	8.554472	CIMSYS_33:44:55	Intel_83:13:e8	ARP	60	192.168.23.11 is at 00:11:22:33:44:55

Figure 3: ARP protocol

1.2.2 What is/are the destination (MAC address[es]) of where the suspicious traffic is mostly directed towards?

00:0e:0c:83:13:e8

1.2.3 What is the link layer protocol used?

ARP Protocol

1.2.4 What is the purpose of this protocol?

The purpose of this protocol is to map MAC addresses with the IP addresses

1.2.5 What conclusions can you draw from the type of attack illustrated by this pcap? How can this attack be used for launching other kinds of attacks?

If all ARP packets come from the same source MAC address and go to the same destination MAC address, it's typically uncommon and may signal ARP spoofing or abnormal network communication. ARP spoofing attacks can lead to Man-in-the-Middle attacks, session hijacking, DNS spoofing, and denial of Service (DoS) Attacks: Overloading the network with ARP packets, causing congestion and disrupting communication, potentially leading to network instability and crashes.

1.3 Evidence file : 3. pcap

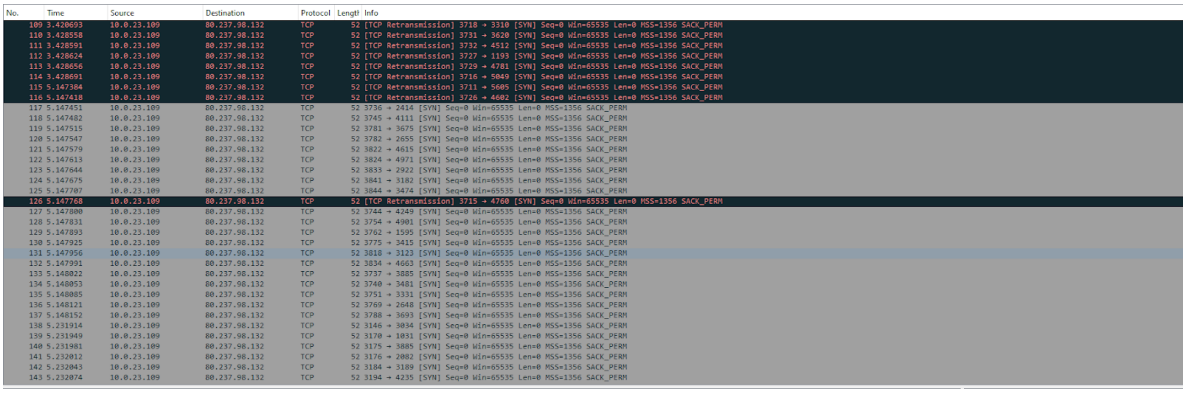


Figure 4: TCP protocol

1.3.1 What is the source (IP address) of the suspicious traffic?

10.0.23.109

1.3.2 What is the destination (IP address) of the suspicious traffic?

80.237.98.132

1.3.3 What is the transport layer protocol used?

TCP

1.3.4 This may be considered as not a direct attack but as a preparation step before an attack. Name the technique used and its purpose.

The technique is port scanning. The orderliness of SYN packets may indicate that a port scan is in progress, where the source IP address attempts to establish a connection to multiple ports on the target host. Port scanning can find which services are running, which can help attackers find potential security holes and attack

2 Assignment 2: Suspicious Wireless Traffic

2.1 Introduction

The network administrator of a small business environment has reported a suspected unauthorized bank account access incident. Traffic sensors deployed across the network have captured three sets of packet data, labeled A.pcap, B.pcap, and C.pcap. These captures originate from different network links and may contain evidence relevant to the incident. The report aims to analyze the captured data, identify any suspicious activities, and provide recommendations for enhancing network security.

2.2 Methods

2.2.1 General tools and methods

Packet analysis was conducted using Wireshark to examine traffic patterns and anomalies. Traffic sensors were deployed strategically across the network to capture packet data. Investigation of network traffic between various network entities, including access points, firewalls, and user devices, was performed to identify any unusual behavior or security breaches.

2.2.2 A.pcap

To examine A.pcap and extract pertinent information about WLAN traffic, encryption details, and IP communications, a systematic approach was employed. First, Wireshark was utilized to access the "Statistics" menu, where the "Conversations" option was selected. By filtering conversations using IEEE 802.11, the focus was directed to WLAN traffic. Subsequently, analysis of source and destination MAC addresses facilitated the identification of the network with the highest traffic. Further examination of packet details enabled the retrieval of crucial information, including the BSSID and SSID of the network.

Moving on to encryption details, the Packet Details pane in Wireshark was scrutinized to locate the "WEP Parameters" section. Relevant details such as Initialization Vector, Key Index, and WEP ICV were extracted, leading to the determination that the encryption used was WEP. Detailed parameters of WEP encryption were retrieved for further analysis.

For WEP key extraction, the Aircrack-ng tool in the Kali VM was employed. Aircrack-ng was used to launch an attack on the WEP-encrypted network using captured IVs. This process resulted in the successful extraction of a matching key from the captured data.

Following the extraction of the WEP key, Wireshark was configured to decrypt captured packets using the added key. Decrypted packets were then analyzed to gain insights into IP communications and additional packet information, allowing for a deeper understanding of network activities.

To identify IP communications with the most traffic, Wireshark's conversation feature was utilized. Conversations were sorted by protocol (IPv4 and IPv6), and data exchange was analyzed. External IP addresses involved in the communication were identified, along with additional details such as data transfer rate and duration.

2.2.3 B.pcap

To identify the IP and the MAC addresses of the firewall and the access point.

First, filter the HTTP data in Wireshark. Check the firewall first. If the firewall is not 192.168.1.100, it should be 192.168.1.201 or 192.168.1.150. Because 192.168.1.x are all internal IPs (So the firewall and access point could be 192.168.1.201, 192.168.1.100, and 192.168.1.150). Find the packet labeled text/HTML in Wireshark's Info column and export the text data in HTML format. Opening the exported text data in a web browser reveals the gateway's address, which is exactly the address of the firewall. In 'A file', we've found the BSSID: 00:13:46:48:b0:f9. Then we filtered the mac address in b.cap and found that the corresponding IP. We can confirm this is the access point IP because BSSID is a unique identifier assigned to each wireless access point in a network.

There is an attack we can identify from this .pcap file. Given that we have identified the firewall's IP and the access point's IP, it indicates that the other IP could belong to the attacker. We observed that the attacker modified the access point's IP.

2.2.4 C.pcap

To reconstruct the webpage from the captured packet data in the C.pcap file, we follow a systematic approach: Firstly, we need to identify the HTTP response packets containing the relevant webpage content. This can be achieved in Wireshark by filtering for HTTP response packets using the filter 'http. response'.

Once the HTTP response packets are identified, we extract the relevant packets containing HTML content, CSS, JavaScript, GIF, and JPG files. These packets constitute the essential components required for reconstructing the webpage.

After extracting the relevant packets, we save the content by right-clicking on each packet and selecting "export packet bytes." This action saves the content as files with the same name as the original, preserving their integrity and structure.

To reconstruct the webpage, all saved files, including the HTML file and relevant resource files (such as CSS, JavaScript, GIF, and JPG files), need to be organized within the same directory. This ensures that the webpage can be properly rendered with all its associated resources.

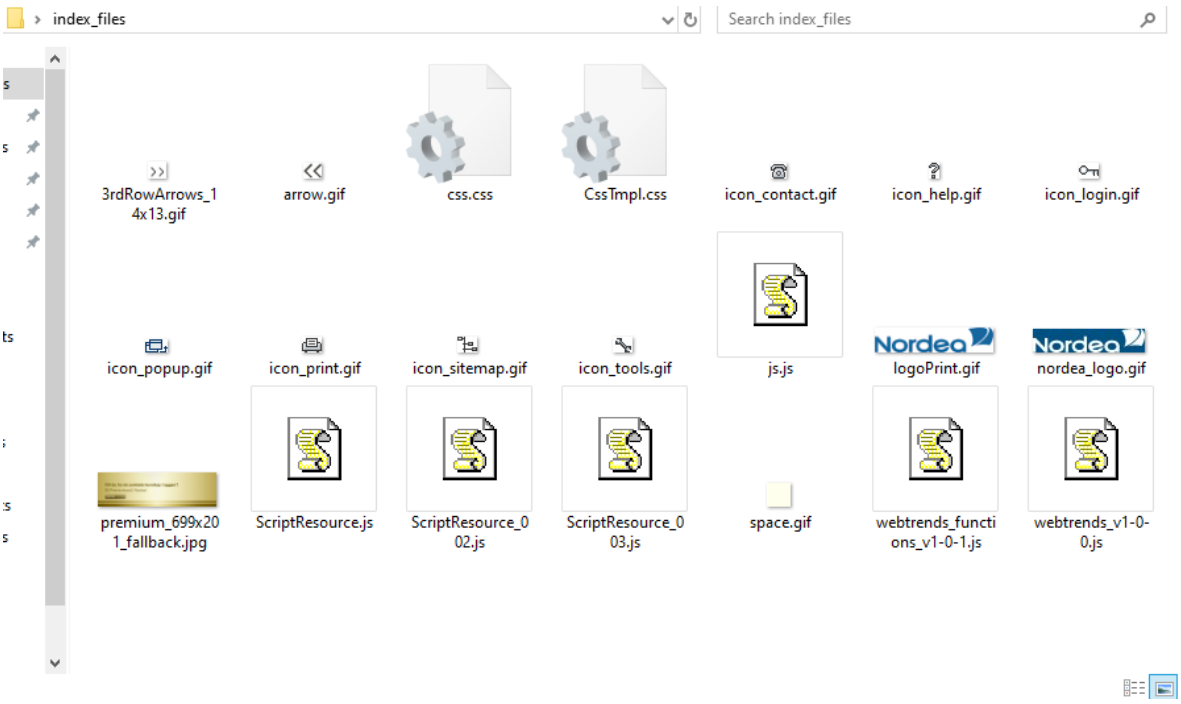


Figure 5: The reconstruction of the webpage

Finally, by reassembling all the files within the same directory, we reconstruct the webpage, allowing for a faithful representation of the original webpage as captured in the packet data.

By following this systematic process, we can accurately reconstruct the webpage from the captured packet data, enabling further analysis and investigation into the user's online activities.

2.3 Results

In A.pcap, the network with the most traffic is between MAC address 08:00:27:d2:f8:61 and 00:1f:3b:2f:73:f1. These two MAC addresses refer to the IP address "Intel_2f:73:f1" and "PCSSystemtec_d2:f8:61". Therefore, this pair of source and destination addresses have the highest data transmission volume.

The BSSID of the network is DLink_48:b0:f9 (00:13:46:48:b0:f9), which is shown on the packet info. The SSID is DSI_DSV.

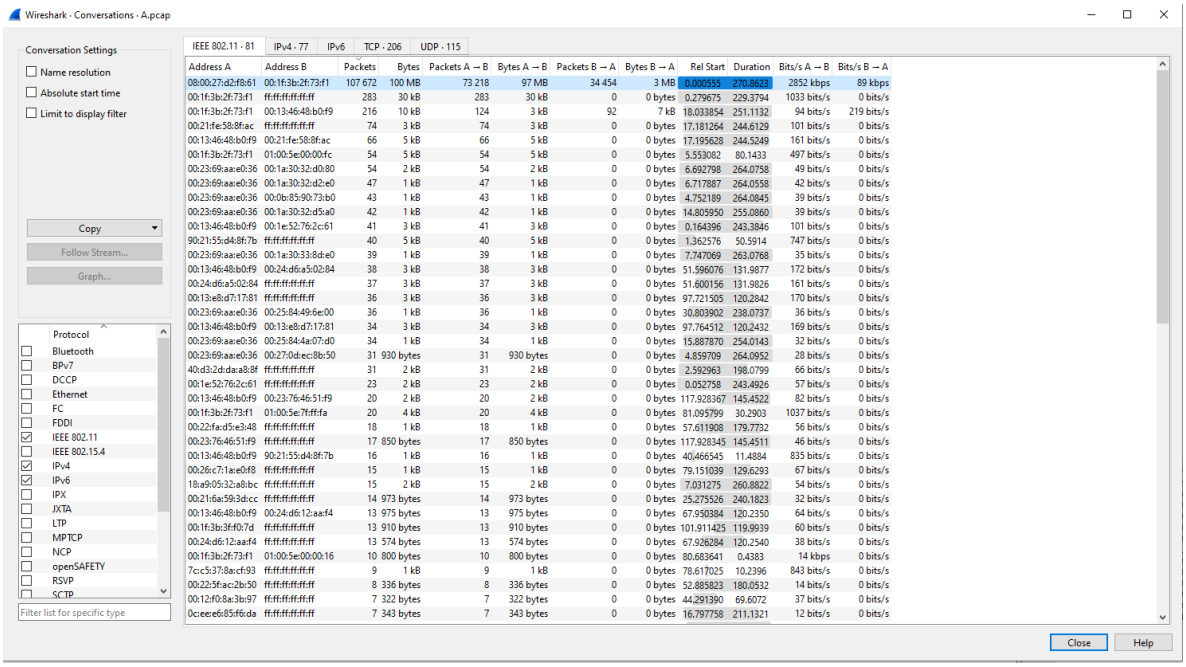


Figure 6: The network with the most traffic

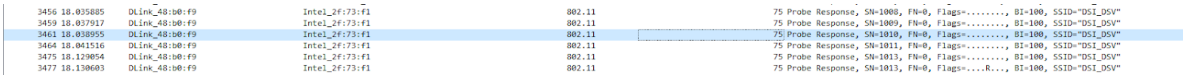


Figure 7: The BSSID and SSID

The Initialization Vector was identified as 0x6d1a00, while the Key Index was determined to be 0. Additionally, the WEP ICV is denoted as 0xcd4c3aac.

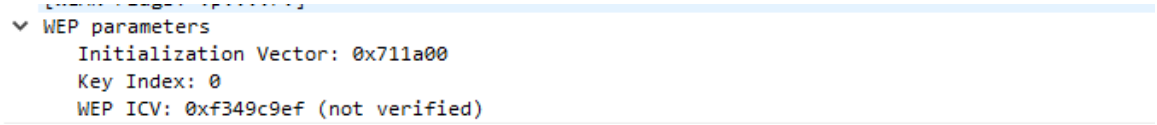


Figure 8: The parameters of WEP key

'A.pcap' was captured in Kali Linux using a tool like 'airodump-ng', with 210499 packets successfully read. During this attack, a matching key was successfully found, with the discovered key being '44:53:49:4C:41', representing the ASCII character 'DSILA'.

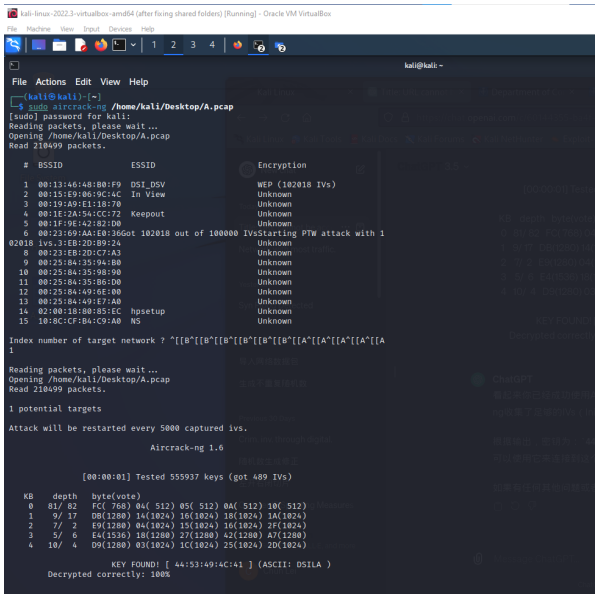
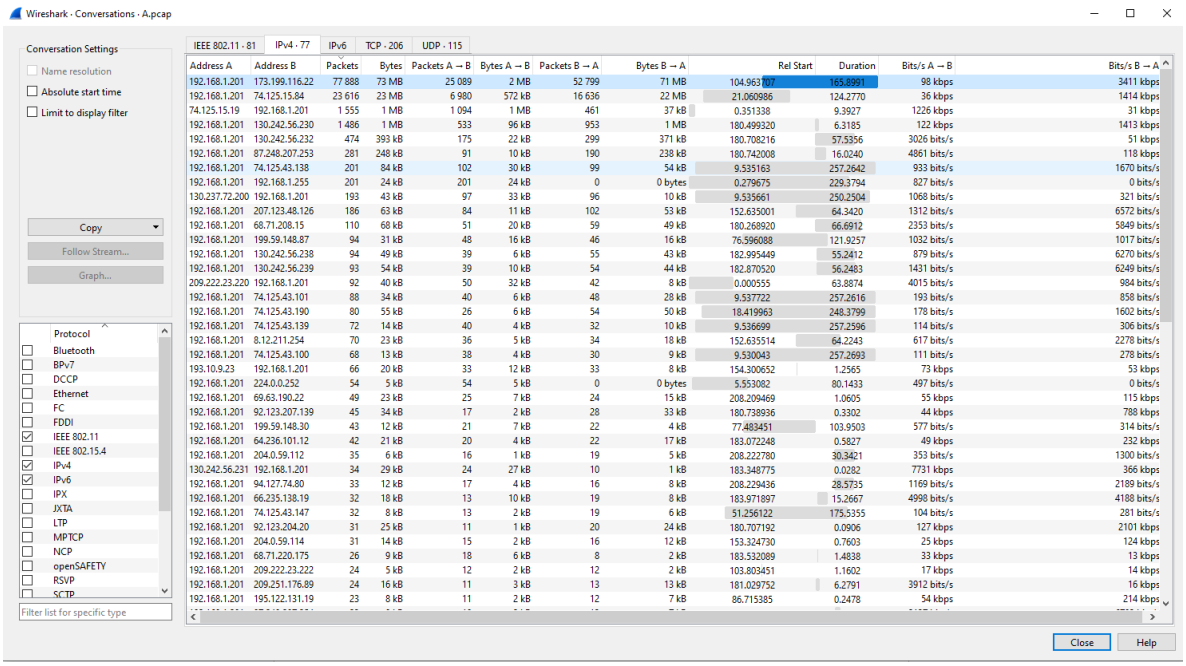


Figure 9: The capturing of A.pcap

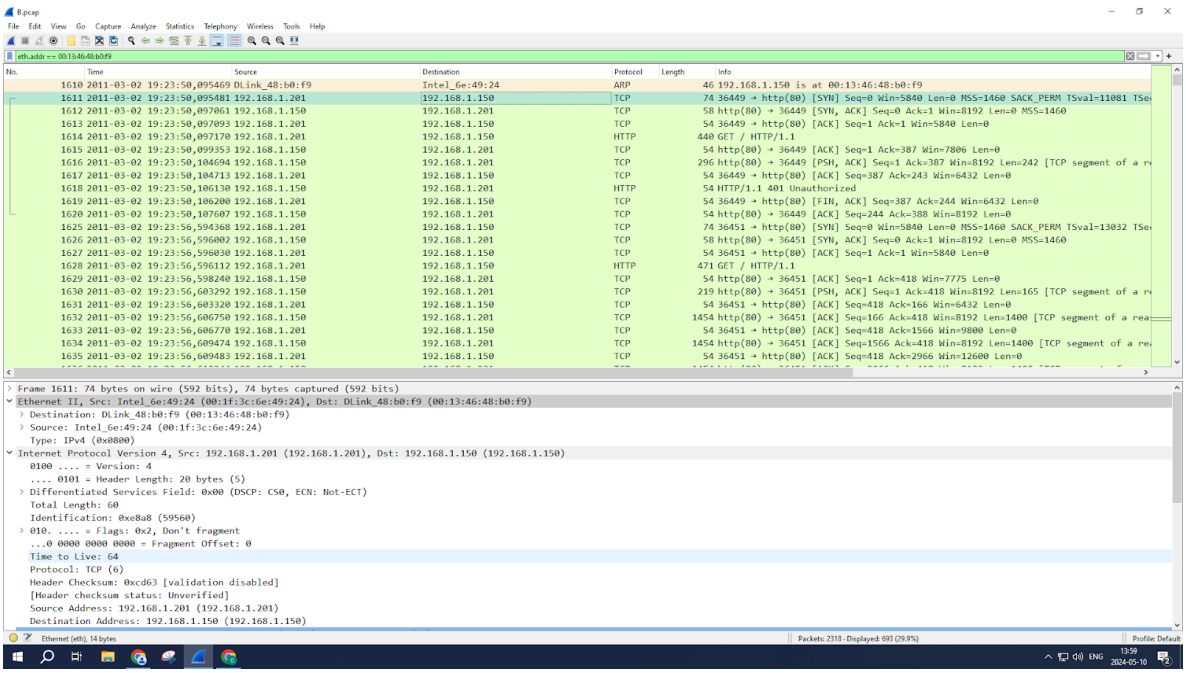
Following the successful key cracking, 'aircrack-ng' proceeded to decrypt the captured packets, achieving a decryption success rate of 100%. This confirms the effective use of the discovered key 'DSILA' to decrypt the WEP-encrypted packets. Following the successful decryption of data packets captured by Wireshark using WEP, pertinent details such as the IP source address, destination address, protocol, and additional packet information become discernible within the Wireshark interface.

No.	Time	Source	Destination	Protocol	Length	Info
28	0.352360	74.125.15.19	192.168.1.201	TCP	1340	80 → 27239 [ACK] Seq=2521 Ack=1 Win=180 Len=1260
29	0.352379		DLink_48:b0:f9 (00:…	802.11	10	Acknowledgement, Flags=........
30	0.353914		DLink_48:b0:f9 (00:…	802.11	10	Acknowledgement, Flags=........
31	0.354411	74.125.15.19	192.168.1.201	TCP	1340	[TCP Previous segment not captured] 80 → 27239 [ACK] Seq=7561 Ack=1 Win=180 Len=1260
32	0.354427		DLink_48:b0:f9 (00:…	802.11	10	Acknowledgement, Flags=........
33	0.354939	192.168.1.201	74.125.15.19	TCP	80	27239 → 80 [ACK] Seq=1 Ack=8821 Win=65520 Len=0
34	0.354922		Intel_2f:73:f1 (00:…	802.11	10	Acknowledgement, Flags=........
35	0.355434	74.125.15.19	192.168.1.201	TCP	1340	80 → 27239 [ACK] Seq=8821 Ack=1 Win=180 Len=1260
36	0.355451		DLink_48:b0:f9 (00:…	802.11	10	Acknowledgement, Flags=........
37	0.355452	192.168.1.201	74.125.15.19	TCP	80	[TCP Dup ACK 33#1] 27239 → 80 [ACK] Seq=1 Ack=8821 Win=65520 Len=0
38	0.355434		Intel_2f:73:f1 (00:…	802.11	10	Acknowledgement, Flags=........
39	0.355962		DLink_48:b0:f9 (00:…	802.11	10	Acknowledgement, Flags=........
40	0.356458	74.125.15.19	192.168.1.201	TCP	1340	[TCP Previous segment not captured] 80 → 27239 [ACK] Seq=11341 Ack=1 Win=180 Len=1260
41	0.356475		DLink_48:b0:f9 (00:…	802.11	10	Acknowledgement, Flags=........
42	0.357481	74.125.15.19	192.168.1.201	TCP	1340	80 → 27239 [ACK] Seq=12601 Ack=1 Win=180 Len=1260
43	0.357499		DLink_48:b0:f9 (00:…	802.11	10	Acknowledgement, Flags=........
44	0.357991	74.125.15.19	192.168.1.201	TCP	1340	80 → 27239 [ACK] Seq=13861 Ack=1 Win=180 Len=1260
45	0.358011		DLink_48:b0:f9 (00:…	802.11	10	Acknowledgement, Flags=........
46	0.357993	74.125.15.19	192.168.1.201	TCP	1340	80 → 27239 [ACK] Seq=15121 Ack=1 Win=180 Len=1260
47	0.358011		DLink_48:b0:f9 (00:…	802.11	10	Acknowledgement, Flags=........
48	0.358506	74.125.15.19	192.168.1.201	TCP	1340	80 → 27239 [ACK] Seq=16381 Ack=1 Win=180 Len=1260
49	0.358523		DLink_48:b0:f9 (00:…	802.11	10	Acknowledgement, Flags=........
50	0.358505	74.125.15.19	192.168.1.201	TCP	1340	80 → 27239 [ACK] Seq=17641 Ack=1 Win=180 Len=1260
51	0.359035		DLink_48:b0:f9 (00:…	802.11	10	Acknowledgement, Flags=........
52	0.359018	74.125.15.19	192.168.1.201	TCP	1340	80 → 27239 [ACK] Seq=18901 Ack=1 Win=180 Len=1260
53	0.359035		DLink_48:b0:f9 (00:…	802.11	10	Acknowledgement, Flags=........
54	0.359529	74.125.15.19	192.168.1.201	TCP	1340	80 → 27239 [ACK] Seq=20161 Ack=1 Win=180 Len=1260
55	0.359547		DLink_48:b0:f9 (00:…	802.11	10	Acknowledgement, Flags=........
56	0.359528	74.125.15.19	192.168.1.201	TCP	1340	80 → 27239 [ACK] Seq=21421 Ack=1 Win=180 Len=1260
57	0.359547		DLink_48:b0:f9 (00:…	802.11	10	Acknowledgement, Flags=........
58	0.360059	192.168.1.201	74.125.15.19	TCP	80	[TCP ACKed unseen segment] 27239 → 80 [ACK] Seq=1 Ack=11341 Win=65520 Len=0
59	0.360042		Intel_2f:73:f1 (00:…	802.11	10	Acknowledgement, Flags=........
60	0.360556	74.125.15.19	192.168.1.201	TCP	1340	80 → 27239 [ACK] Seq=22681 Ack=1 Win=180 Len=1260
61	0.360572		DLink_48:b0:f9 (00:…	802.11	10	Acknowledgement, Flags=........
62	0.360555	74.125.15.19	192.168.1.201	TCP	1340	80 → 27239 [ACK] Seq=23941 Ack=1 Win=180 Len=1260

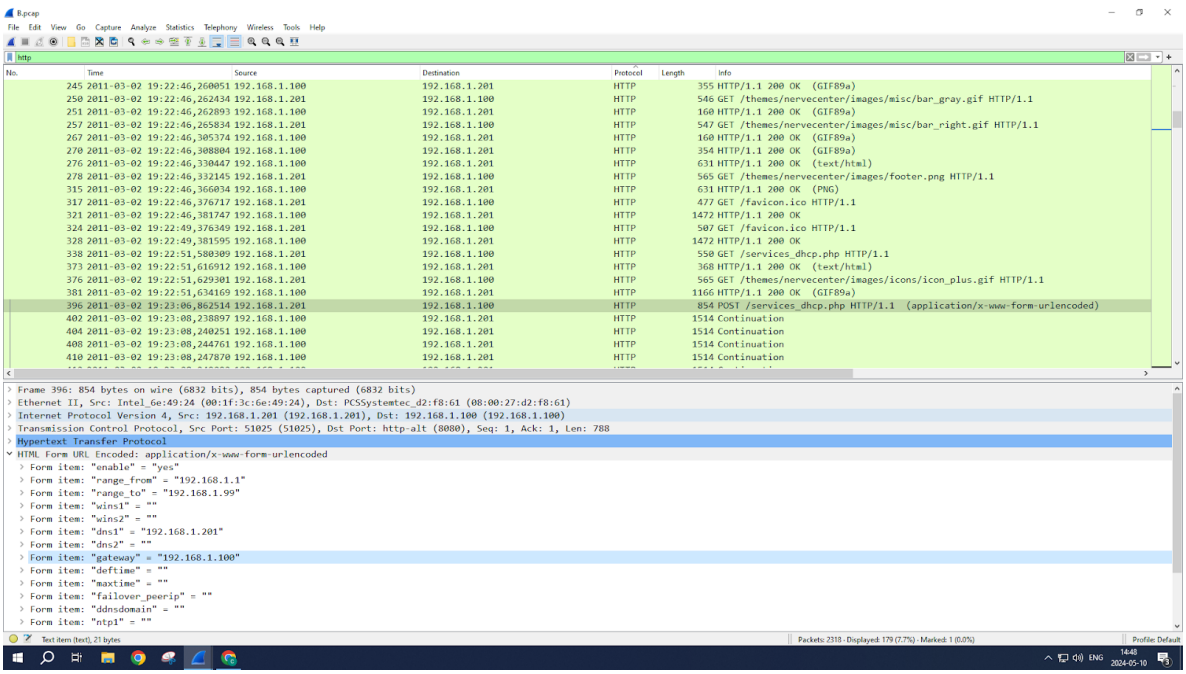
The IP communication with the most traffic involves the IP addresses 192.168.1.201 and 173.199.116.22, exchanging a total of 73 MB of data. Additional external IP addresses involved in the communication include 74.125.15.19, 74.125.43.121, 130.242.56.230, and 199.59.148.12. The data exchange appears to be bidirectional, with a duration of approximately 165.8991 seconds. The bitrate (Bits/s) indicates varying data transfer speeds throughout the communication.



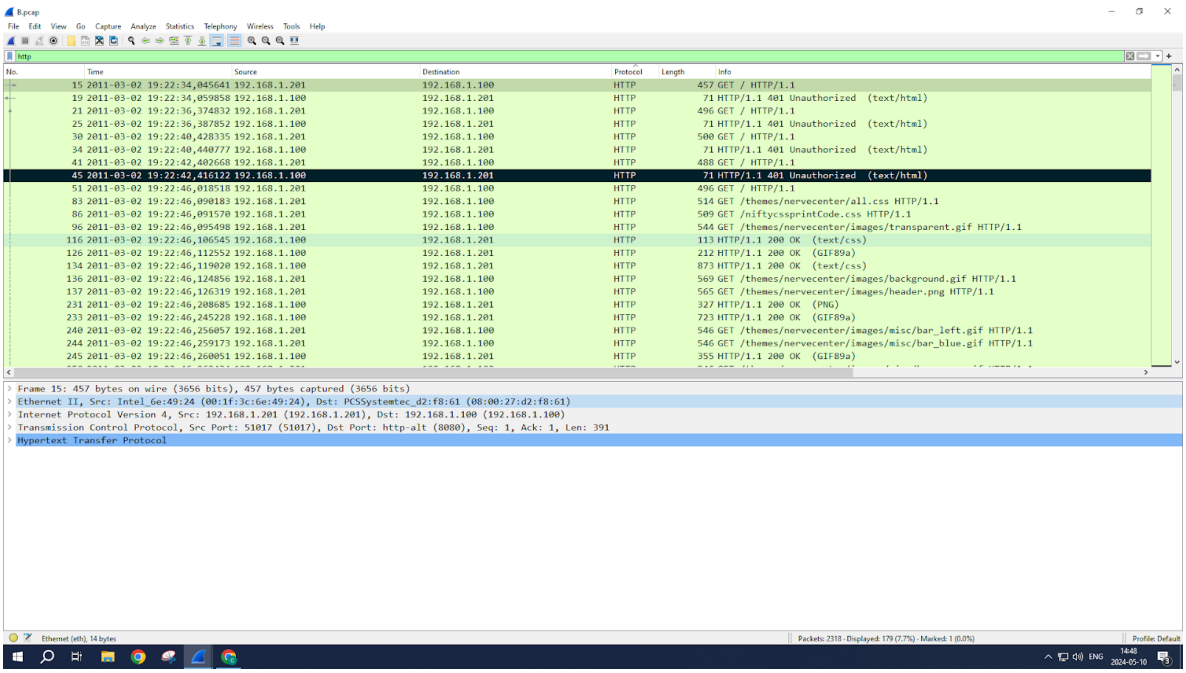
In B.pcap we know that the IP address of the firewall is 192.168.1.100 (Mac address: 08:00:27:d2:f8:61). We can also see that the IP address of the access point is 192.168.1.150 (Mac address: 00:13:46:48:b0:f9).



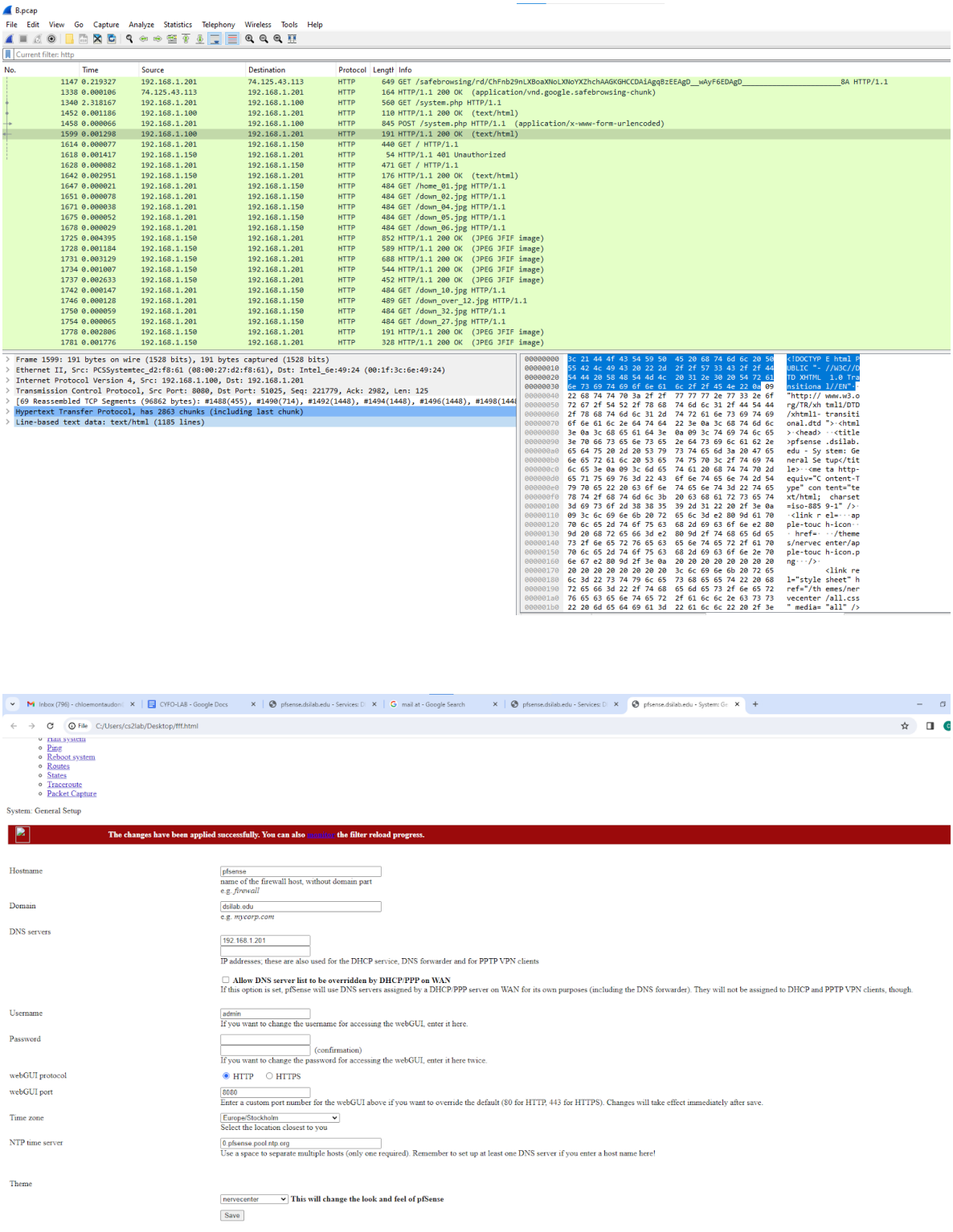
Bcap is supposed to be the wireless traffic of the firewall and access point as the picture shows in the instruction. But we can only see the connection of 192.168.1.201 and 192.168.1.100.



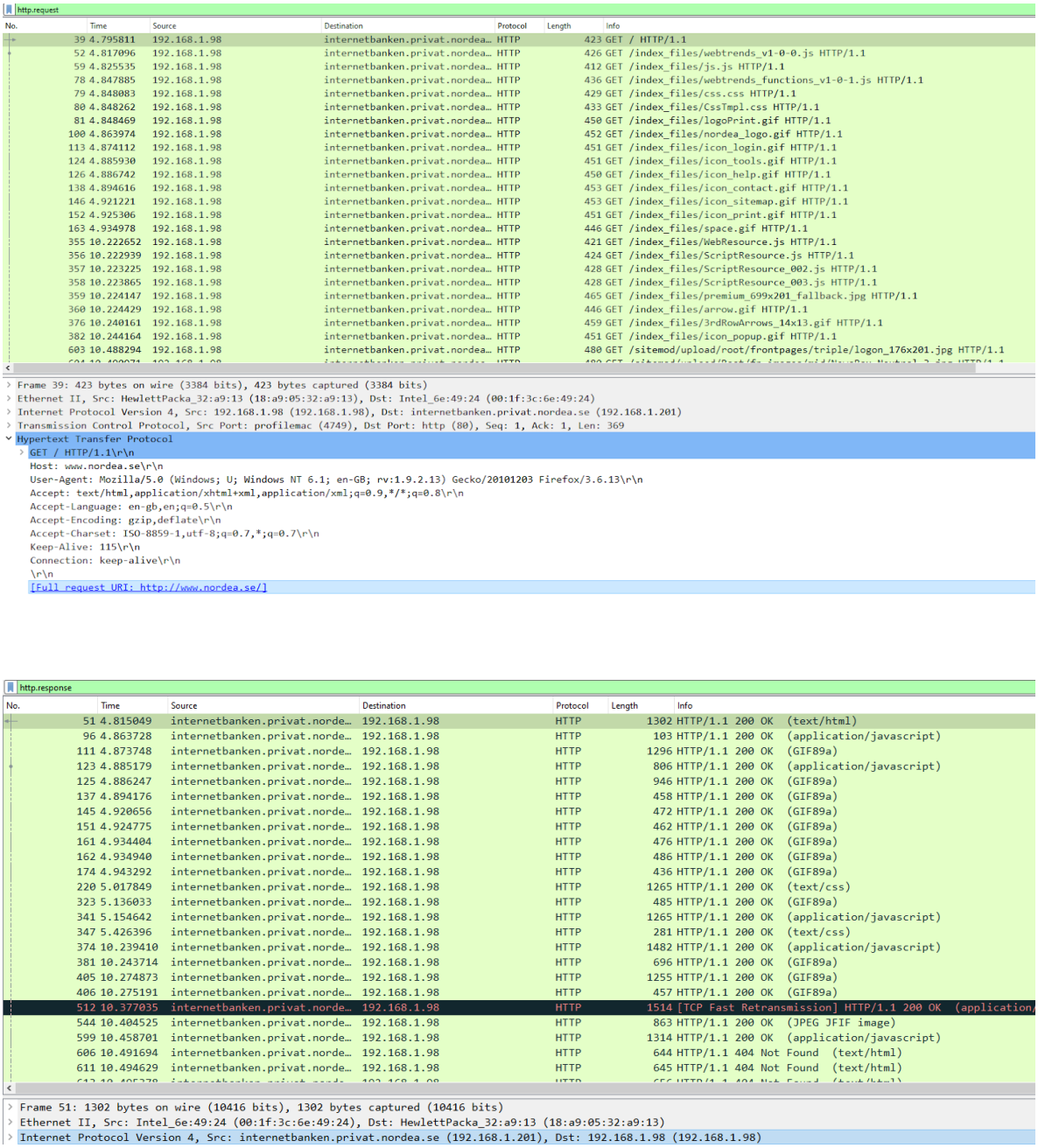
Because 192.168.1.100 (mac address: 08:00:27:d2:f8:61) is the firewall's IP, it indicates 192.168.1.201 (mac address: 00:1f:3c:6e:49:24) could be the attacker IP. The access point IP 192.168.1.150 is changed by the attacker.



So, there is a Man-in-the-middle attack and DNS spoofing taking place: We discovered that the DNS server became 192.168.1.201. This indicates that the attacker tampered with DNS requests and responses during the communication process.



We can see the info shows 'GET / HTTP/1.1'. Click it and we can see the full request URL: http://www.nordea.se/ and Internetbanken.privat.nordea.se replied to the HTTP request from the client 192.168.1.98. The victim's computer is not accessing Nordea's information through the IP address of the access point. Instead, it receives information from the attacker, who is using the IP address 192.168.1.201. At the start of the C file, ARP spoofing occurs to facilitate an MITM attack. This allows the attacker to intercept communications between the victim and Nordea's web page, potentially stealing sensitive information.



The original page around 2011, March can be found on the Wayback Machine website. Compared to the reconstructed one, there is no difference.

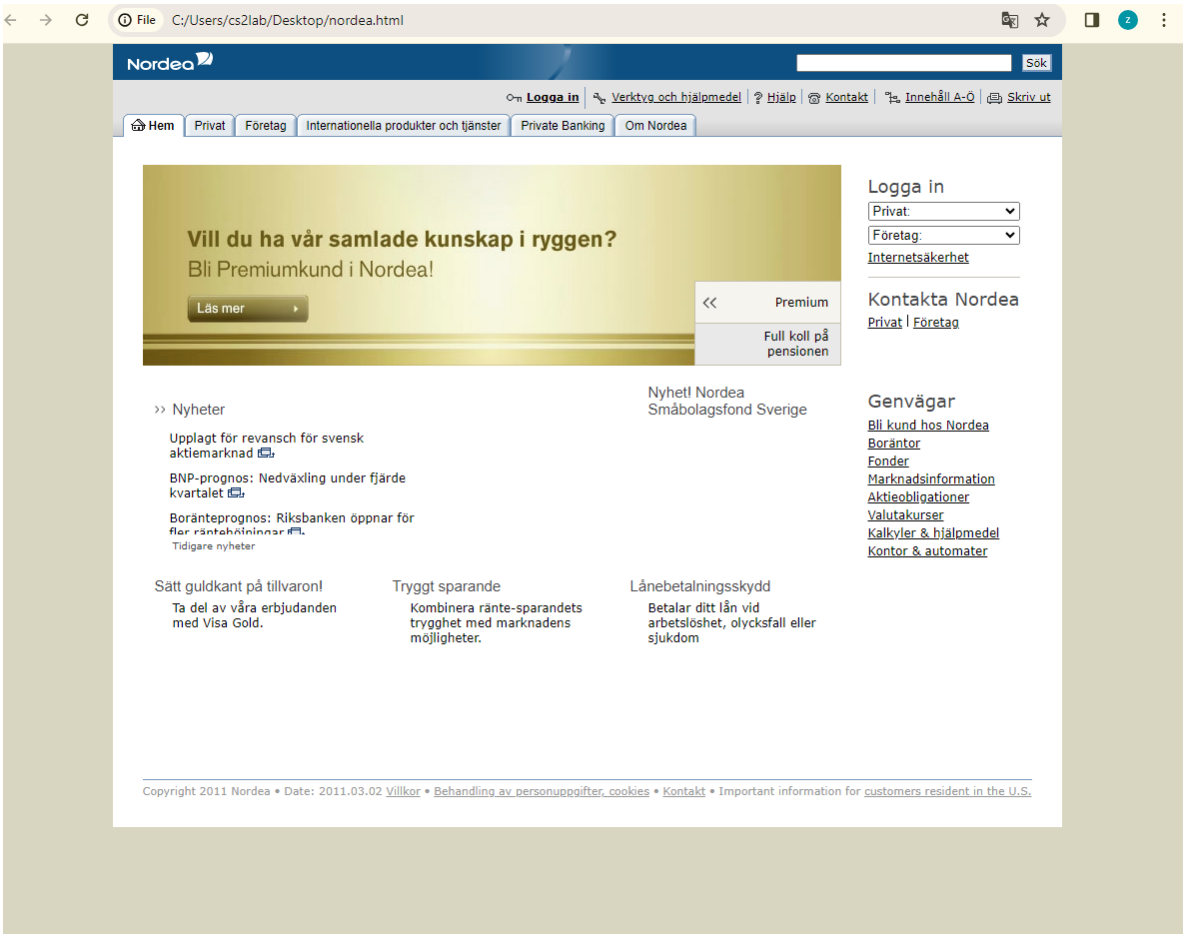


Figure 10: The reconstructed webpage

2.4 Discussion

Beginning with the user's activity, represented by the capture file C.pcap, the IP address 192.168.1.98 initiates an HTTP request to internetbanken.privat.nordea. Subsequently, the response to this request is routed back to the access point at 192.168.1.201. This sequence suggests typical user behavior, where a user engages with an online banking service.

Moving to the intermediary point between the access point and the firewall, captured in file B.pcap, data packets are exchanged between the access point (192.168.1.201) and the firewall (192.168.1.100). This communication encompasses the transmission of data from various devices linked to the access point towards the external network, which is managed and secured by the firewall. Such traffic includes legitimate requests from users like the one to the Nordea bank website.

Lastly, in the capture file A.pcap representing the attacker's perspective, unauthorized access to the access point is achieved. Consequently, all traffic flowing through the access point becomes susceptible to interception and capture by the attacker. This opens up the possibility for malicious activities, such as redirecting users attempting to access the Nordea bank website to a counterfeit page designed to collect sensitive information.

2.5 Conclusion

The investigation into the suspected unauthorized bank account access incident has revealed potential security vulnerabilities and suspicious activities within the network environment.

Recommendations for enhancing network security include upgrading encryption protocols from WEP to more secure alternatives like WPA2 or WPA3[1], deploying intrusion detection systems to detect and mitigate potential DoS attacks, conducting regular security audits and updates to address emerg-

ing threats and vulnerabilities, and providing ongoing user education and training on safe browsing practices and password security measures.

A proactive approach to network security is essential for protecting against unauthorized access and mitigating potential threats.

References

[1] "WEP, WPA, WPA2 and WPA3: Differences and explanation." Section: Resource Center.

3 Assignment 3: Intrusion Analysis

3.1 Introduction

In response to an alert from CERT-SE indicating possible intrusion attempts on our network, an investigation has been launched into potential security breaches affecting the Haisy student management system. This report presents the findings of a digital forensic investigation conducted on two pieces of digital evidence: a network traffic capture file (.pcap) and a Linux server disk image. The investigation aims to analyze network activities, potential intrusions, and system compromises based on the evidence collected. The investigation focuses on identifying the attack IP addresses, understanding the timeline of the attack, and determining the network where the attack occurred. Additionally, the report aims to uncover the nature of the attack, including its methods, timing, origin, and possible motives. By thoroughly examining the evidence, we aim to assess the extent of the intrusion and recommend appropriate measures to enhance the security posture of the Haisy system and prevent future incidents.

3.2 Methods

3.2.1 General tools and methods

Digital evidence for the investigation into potential security breaches affecting the Haisy student management system includes network traffic capture data in the form of a .pcap file and a disk image of the Linux server. To analyze these sources, a combination of forensic analysis tools such as Wireshark for network traffic analysis, Autopsy for disk image examination, and command-line utilities for hash calculation and metadata inspection were employed. Following established forensic analysis procedures, including hash verification, network traffic inspection, log file examination, and system artifact analysis, aims to uncover any potential security threats or intrusion attempts within the network or on the Haisy server.

3.2.2 Network Traffic Analysis

Calculate the SHA1 hash sum of the file and determine the exact byte size of the file

To calculate the SHA1 hash of a file in the Windows terminal, use the 'CertUtil' command-line tool. This allows you to get the hash value of the file and determine the exact bytes of the file.

Traffic Analysis by IP Addresses

Go to the Statistics tab in Wireshark to find all the IPv4 addresses in the .pcap file. To recognize traffic that generated the most traffic we check the "percent" and "count" columns to put them in order.

Port Analysis

To find the most common ports, we go to the endpoint tab in Wireshark and filter out the packets in descending order.

Geolocation of IP Addresses

We go to Geo-locate to find the top three most occurring IP addresses, and we eventually only find one of those addresses on the Geo-locate.

DNS Traffic Analysis

After filtering DNS-related packets we can see many DNS queries are being made to various domains. And there are a few queries that could be considered suspicious.

HTTP Traffic Analysis

In the filtered HTTP traffic, there are interesting requests (all this information is displayed in the browser's address column and info column) and suspicious requests (we can find those in the info

column as well, we can see various status codes, some unusual resources and so on).

TCP Stream Analysis

After filtering TCP traffic, we can see the pattern of sending duplicate SYN-ACK packets from one address to another address is consistent, which is quite interesting. We find suspicious activities based on the repeated transmission time from one address to another address.

3.2.3 Linux Server Disk Image Analysis

SHA1 Hash and Byte Size of The haisy.raw File

The forensic investigation commenced with the determination of crucial file attributes for 'haisy. raw'. Using the 'sha1sum' command with sudo privileges, the SHA1 hash sum was computed, and the 'stat' command with the '-c %s' option revealed the exact byte size. Authentication was required for sudo access, ensuring secure execution. These commands, executed in the Linux terminal, provided essential insights into the file's integrity and size.

Finding of Hard Drive Partitions Operating System

After importing the `haisy.raw` file into Autopsy, the left-hand pane displayed the `haisy_raw_1 Host` section, revealing five volumes under `haisy.raw`. Among these volumes, two were allocated, indicating the presence of two hard drive partitions. Additionally, in the `Operating System Information` section of the left-hand pane, the exact name and version of the operating system were identified as Linux (Debian) and Linux (Ubuntu).

Finding of User Accounts

The investigation involved navigating to the 'etc/passwd' file within the file explorer in Autopsy. Upon accessing this file, details about each user account were analyzed, including the username, UID, GID, user information, home directory, and shell. Each line in the 'passwd' file provided attributes corresponding to a user account.

Investigate the Command History And System Log Files for Indications of Suspicious Activitiess

Utilized Autopsy to navigate through the file system and identify log files such as `.bash_history`, `syslog`, and `auth.log`. Focused on the `error.log` file within the Apache2 logs located in the `/var/log/apache2` directory as an example, the error log records errors that occurred while the Apache2 server was processing requests. The error log contains messages like "file not found" or "unable to access file," revealing attempts to reach non-existent PHP scripts or use invalid request methods. These entries suggest possible hacker activities targeting server weaknesses. Analyzing the log for repeated access attempts to specific scripts or unusual request methods helps identify potential intrusion attempts. Look for patterns like PHP configuration warnings or efforts to access sensitive scripts, indicating possible security breaches.

Finding Other Running Services

Explored the '/etc/init.d/' directory to investigate the presence of additional services running on the server. Examined file metadata including modification, access, and change times to understand if these services were active or configured on the server.

Identifying Potential Attacks on Tomcat and Apache Services in Network Traffic Analysis

Analyzed network traffic stored in the "hairy.pcap" file using Wireshark to identify potential attacks targeting the Tomcat or Apache services. Filtered TCP protocol records to focus on relevant traffic, specifically using the filter 'tcp.port == 59808'. Reviewed TCP streams to detect any suspicious activities, such as attempts to modify files within the Tomcat web server directory (e.g., 'index.html').

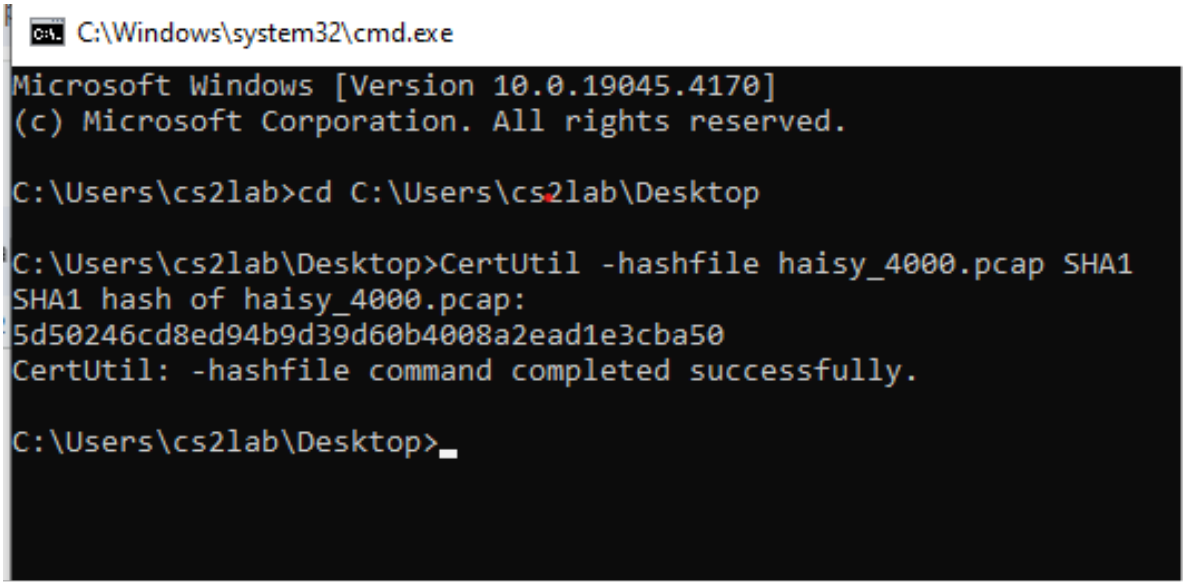
Timeline Analysis of Network Attack on IP Address 10.11.8.18

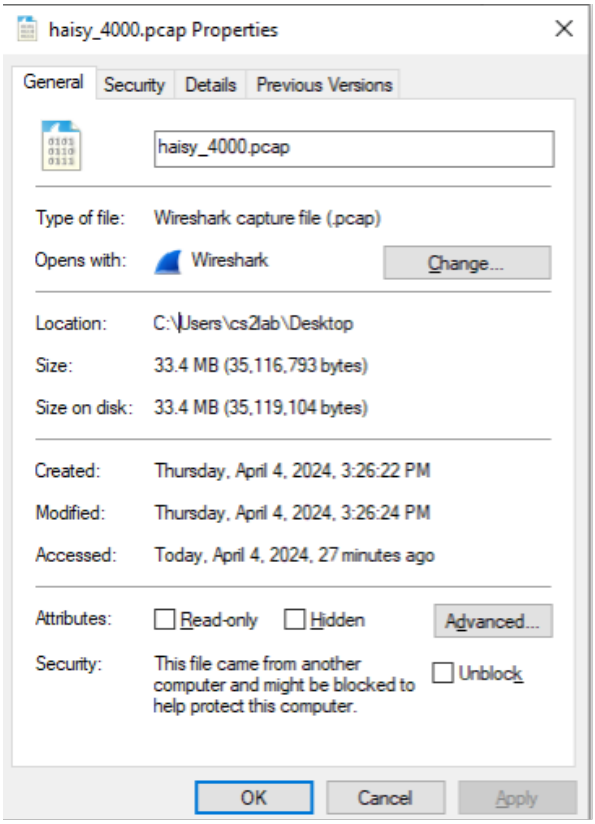
Reviewed network traffic captured in the `"haisy_4000.pcap"` file using Wireshark to identify events correlating with log entries on the seized Linux system. Focused on packets involving the IP address `37.120.246.146` sending TCP packets to the IP address `10.11.8.18`, marked with the `[RST, ACK]` flags, indicating reset packets with acknowledgment. Using the filter `tcp.port == 59808` can expedite the identification of SYN-ACK packet establishment time. Recognized the repeated occurrence of these packets as potential signs of an attack on the IP address `10.11.8.18` originating from Bucharest since November 15, 2011, at 12:24:39.

3.3 Results

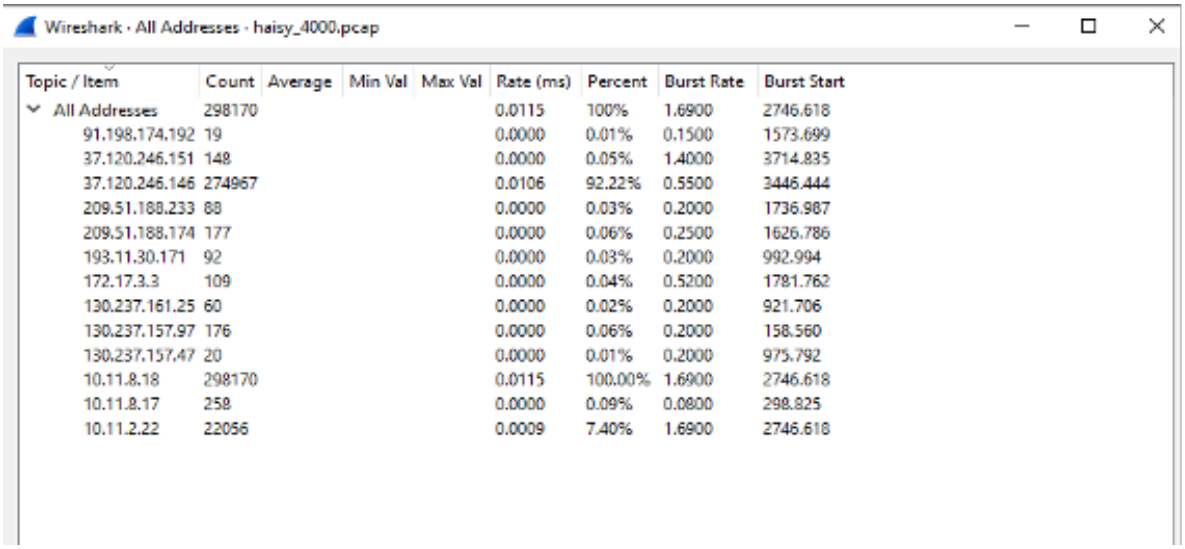
3.3.1 Network Traffic

After running the 'CertUtil' command, we can get the hash value of the file. The hash value is 5d50246cd8ed94b9d39d60b4008a2ead1e3cba50, which is different from the one given in the instructions. The exact bytes of the file are 33.4 MB



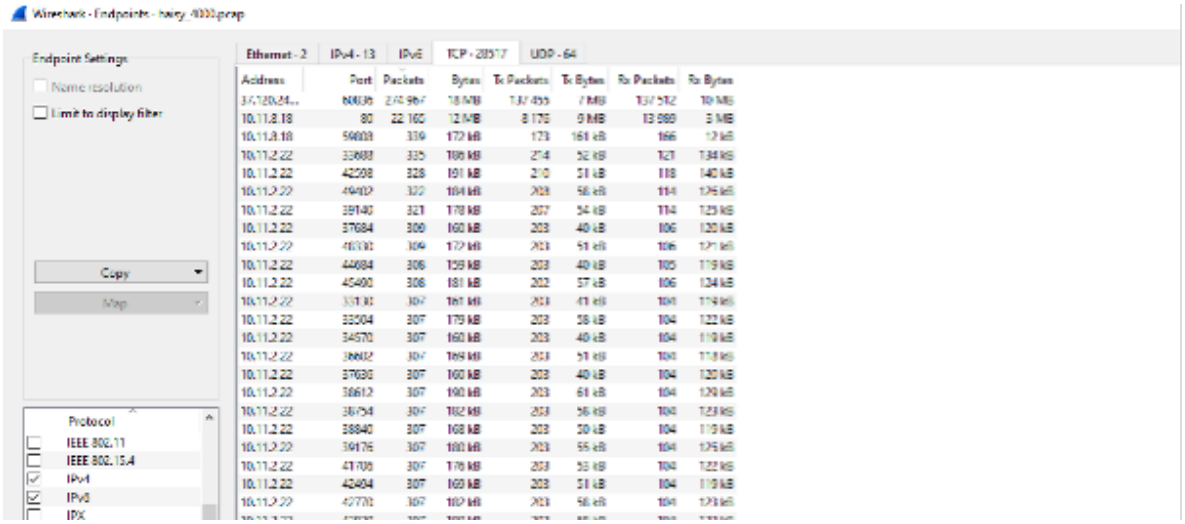


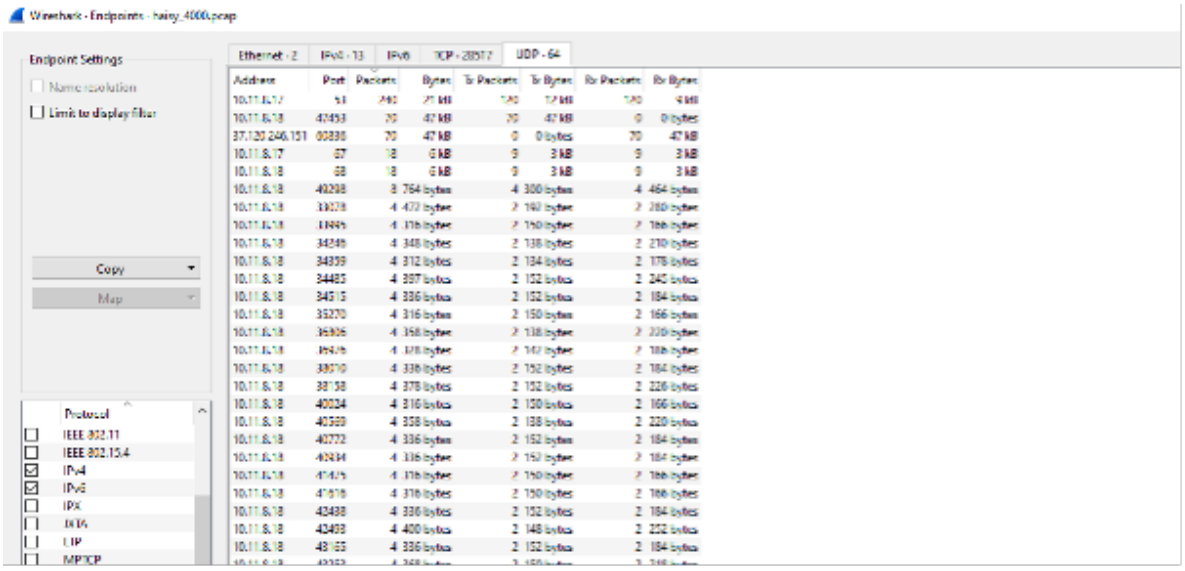
There are 13 IPv4 addresses present in the file.



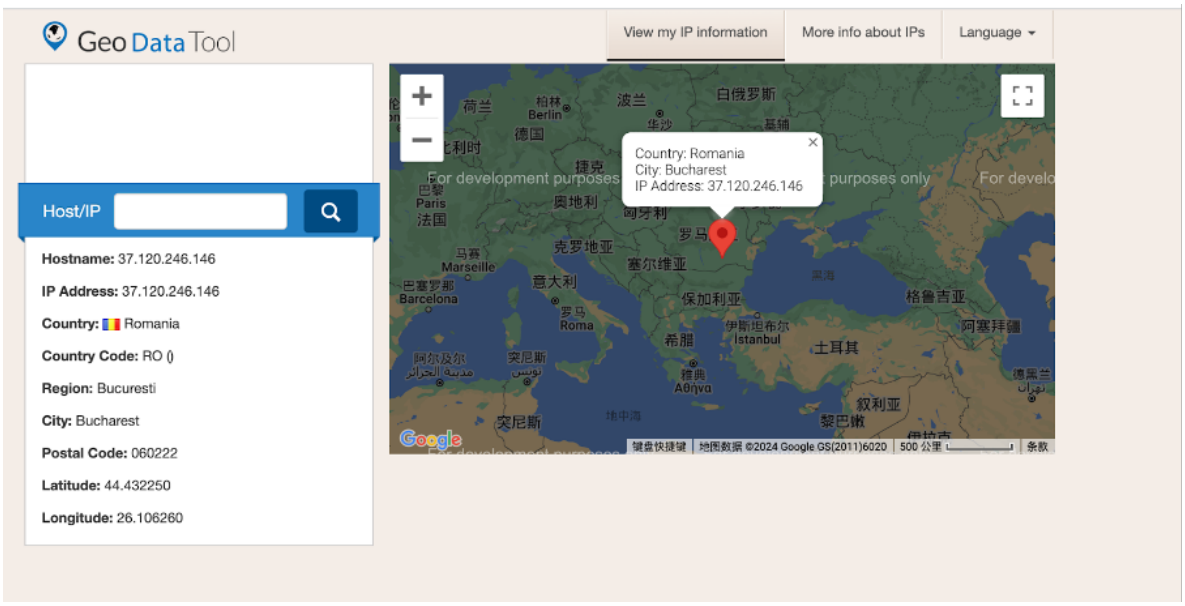
Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
All Addresses	298170				0.0115	100%	1.6900	2746.618
91.198.174.192	19				0.0000	0.01%	0.1500	1573.699
37.120.246.151	148				0.0000	0.05%	1.4000	3714.835
37.120.246.146	274967				0.0106	92.22%	0.5500	3446.444
209.51.188.233	88				0.0000	0.03%	0.2000	1736.987
209.51.188.174	177				0.0000	0.06%	0.2500	1626.786
193.11.30.171	92				0.0000	0.03%	0.2000	992.994
172.17.3.3	109				0.0000	0.04%	0.5200	1781.762
130.237.161.25	60				0.0000	0.02%	0.2000	921.706
130.237.157.97	176				0.0000	0.06%	0.2000	158.560
130.237.157.47	20				0.0000	0.01%	0.2000	975.792
10.11.8.18	298170				0.0115	100.00%	1.6900	2746.618
10.11.8.17	258				0.0000	0.09%	0.0800	298.825
10.11.2.22	22056				0.0009	7.40%	1.6900	2746.618

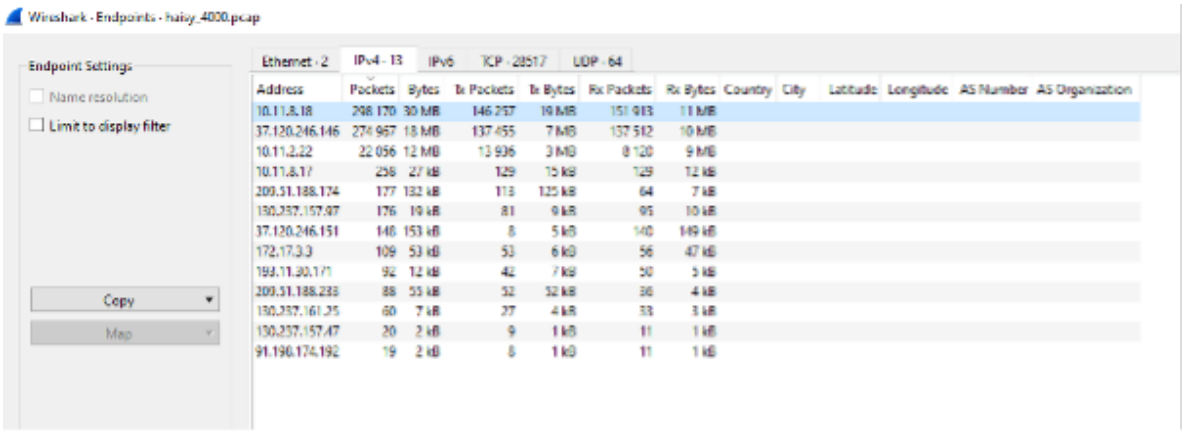
We can see that 10.11.8.18, 37.120.246.146, 10.11.2.22, 10.11.8.17,209.51.188.174 are the top five addresses (IP addresses generated the most traffic).





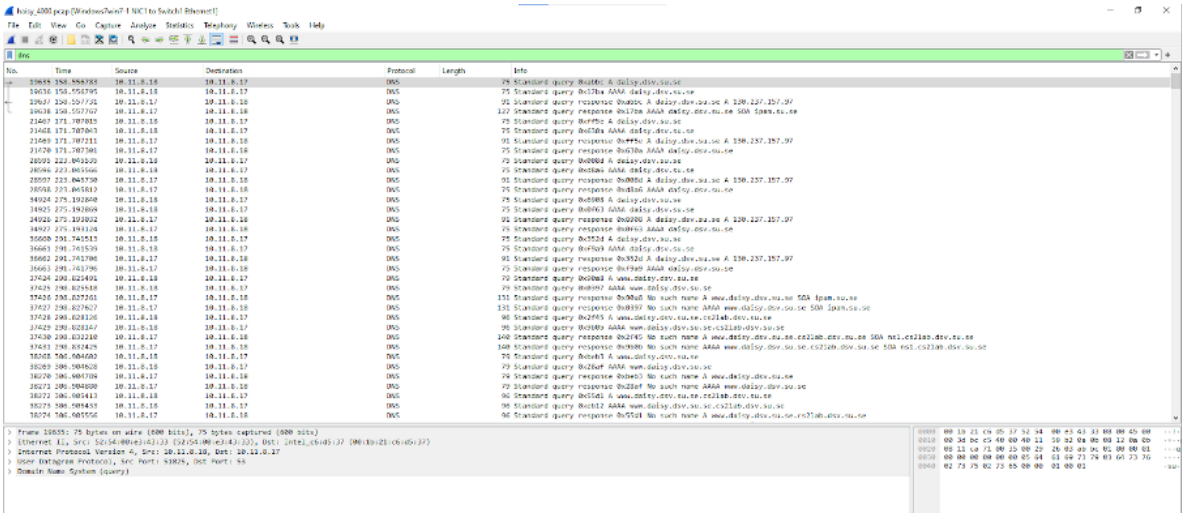
Based on the packets we can see that TCP protocol has the most common ports: 60836, 80, 59808, 33688, 42598. Top-three most occurring IP addresses- 10.11.8.18-can't be found in geo data tool, 37.120.246.146-located in Bucharest, Romania, 10.11.2.22-cannot found in geo data tool.

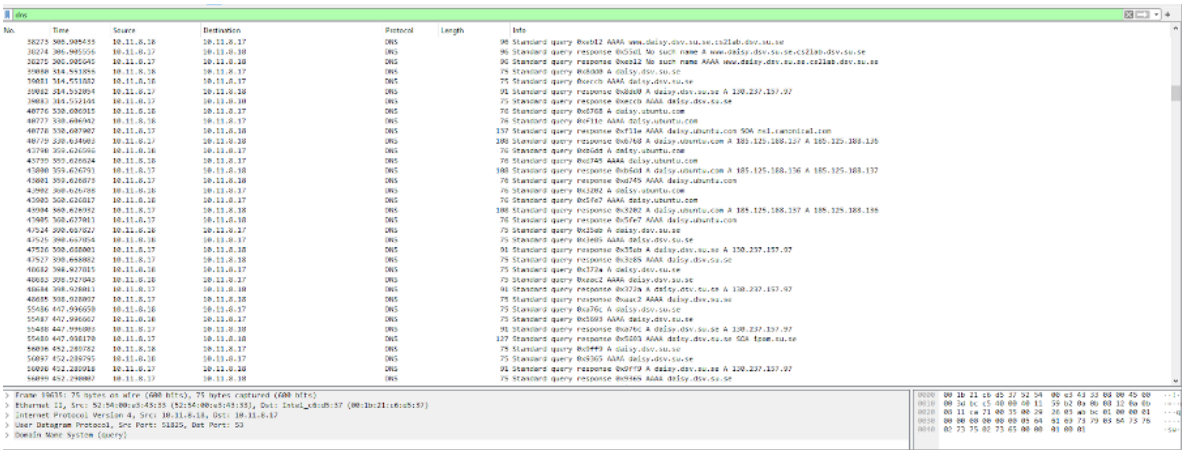


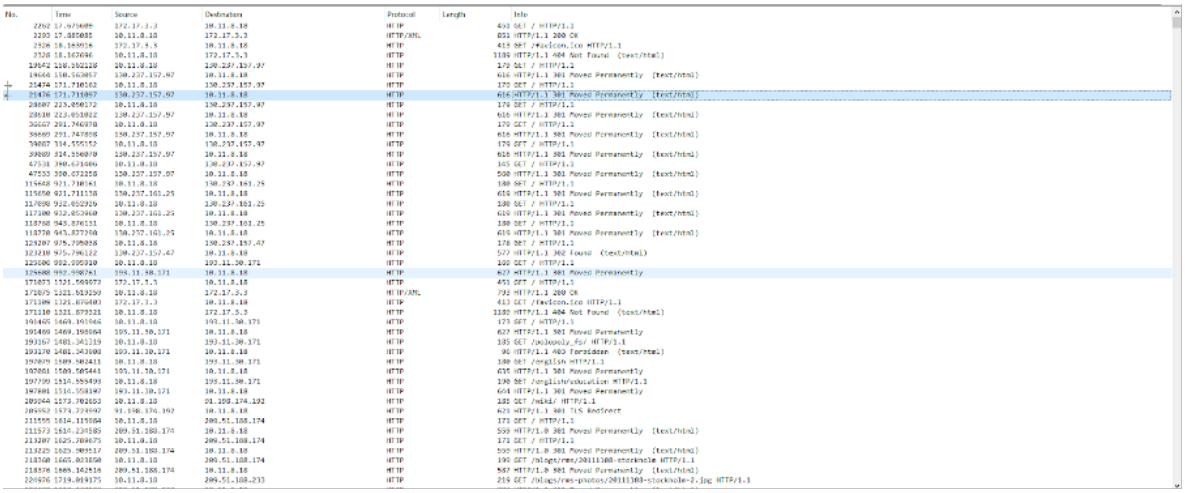


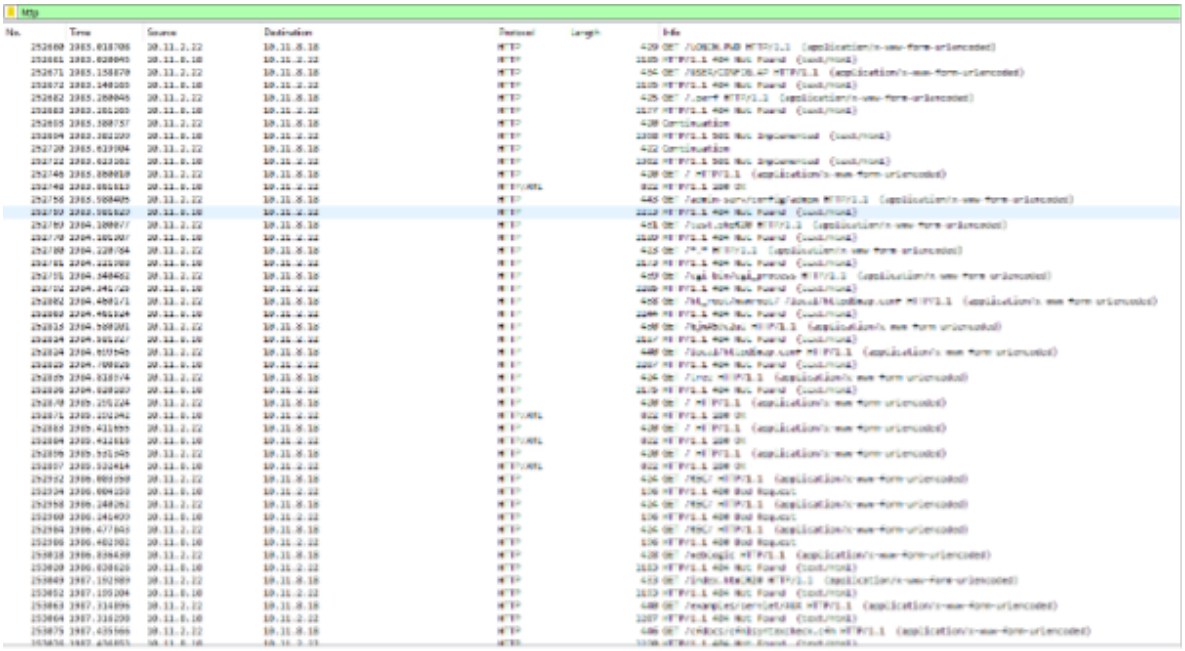
Interesting DNS-related queries: Many DNS queries are being made to various domains, such as "daisy.dsv.su.se" and "daisy.ubuntu.com" etc. These queries seem to be standard.

But there are a few rows where the response shows "No such name", "No such domain" "403 Forbidden", "404 Not Found", and "301 moved permanently". These could be considered suspicious, we assume they're trying to resolve non-existent domains.



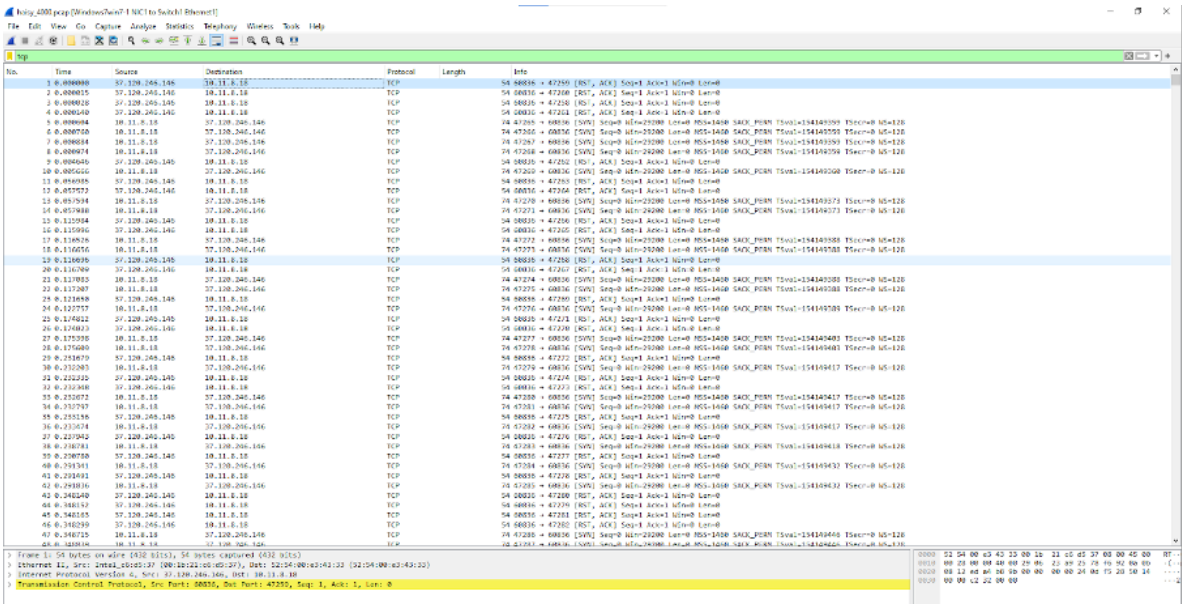


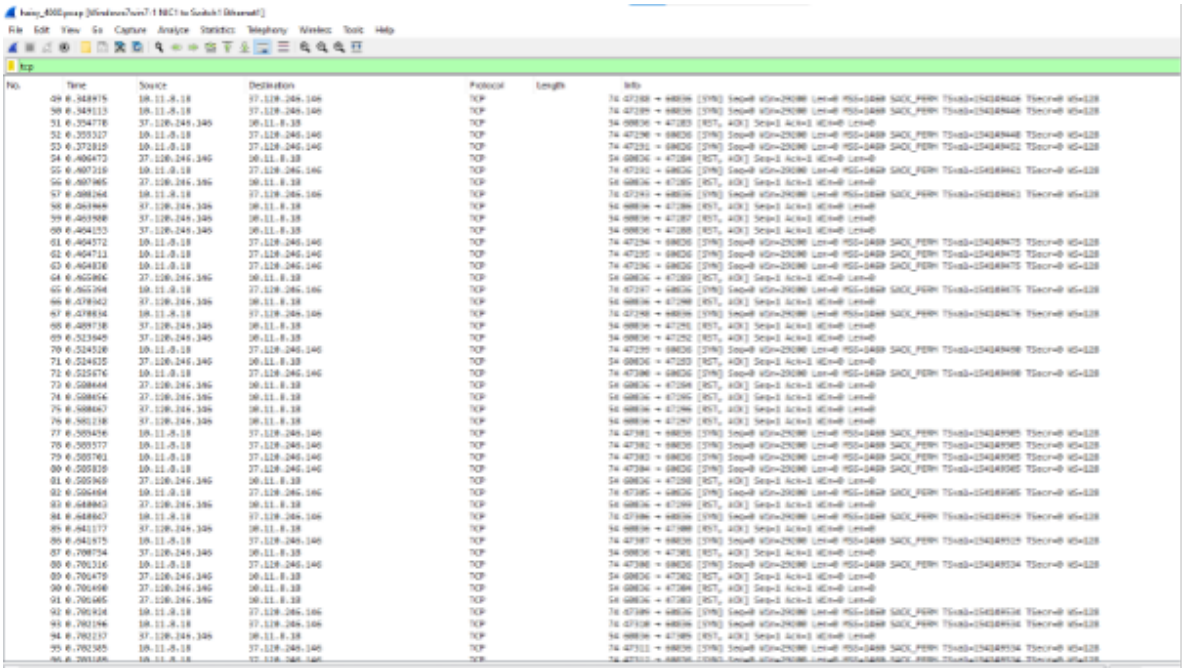


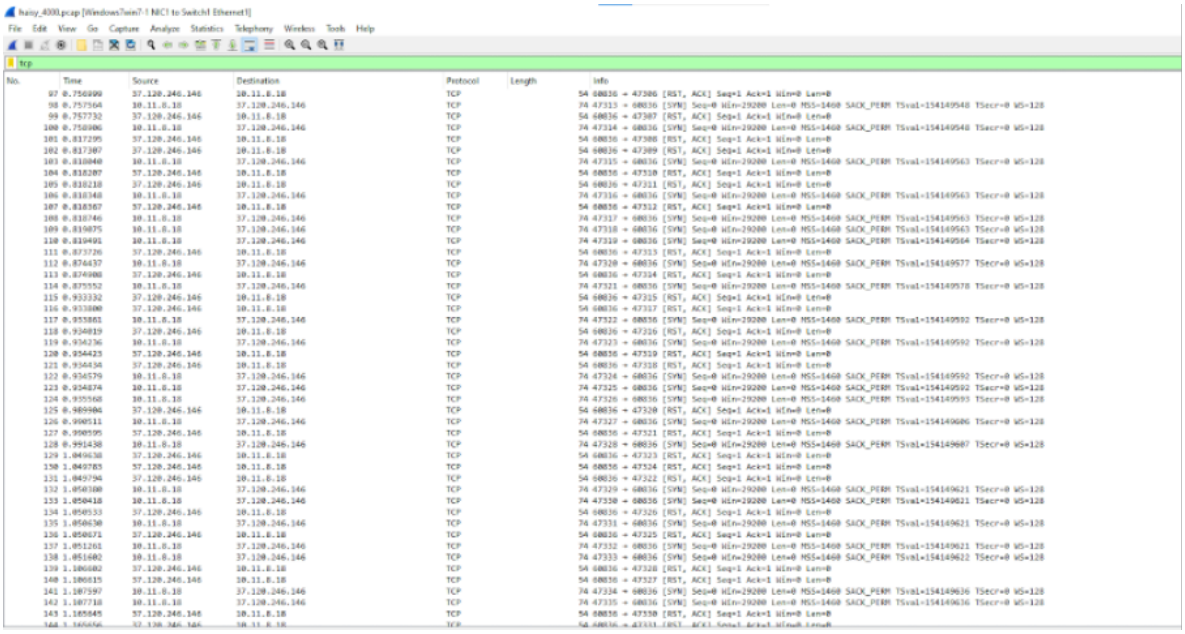


We can see the interesting and suspicious TCP streams from the pattern of sending duplicate SYN-ACK packets from 10.11.8.18 to 37.120.246.146, which is consistent. This may be the behavior of a flood attack. The attacker repeatedly sends SYN packets to initiate a TCP connection, and the target consumes resources responding to SYN-ACK packets.

There are TCP retransmissions in the captured data packets. "10.11.8.18" repeatedly transmits information to "37.120.246.146" 3 times. This could indicate network congestion or bypassing network security protections by repeatedly sending the same information.







3.3.2 Linux Server Disk Image

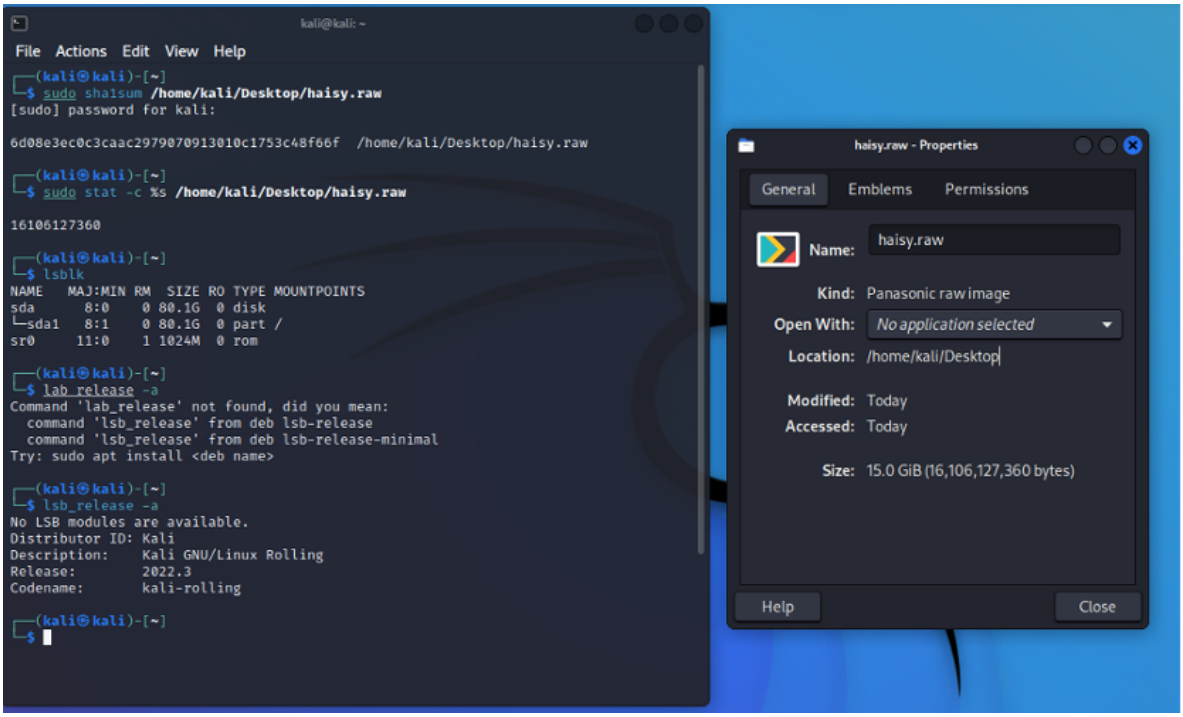
After running the 'CertUtil' command, we can get the hash value of the file. The hash value is 5d50246cd8ed94b9d39d60b4008a2ead1e3cba50, which is different from the one given in the instructions. The exact bytes of the file are 33.4 MB

Identification of The Evidence

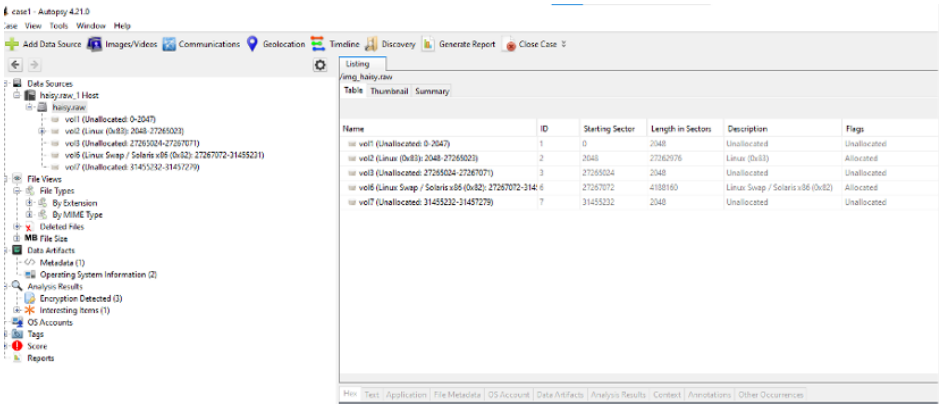
SHA1 hash sum: '6d08e3ebcf01c3c0a3cc9976eb80c54c0d22a73e'

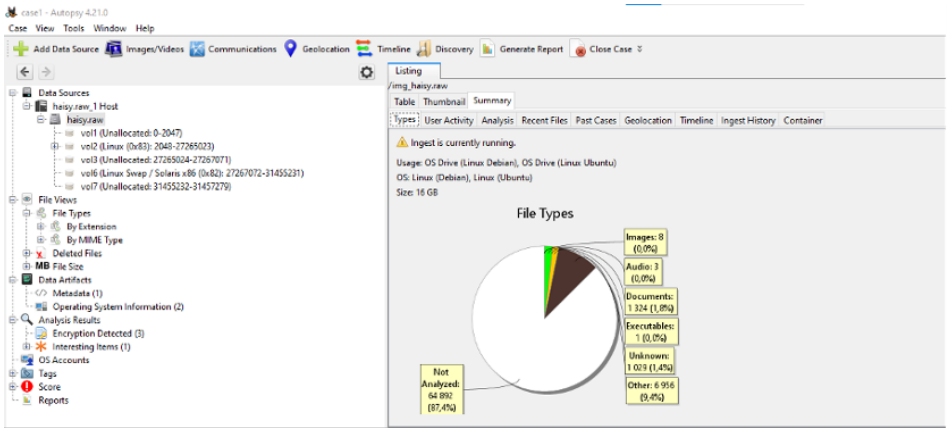
Exact byte size: '16,106,127,360 bytes' (approximately 15.0 GiB or 16GB)

The SHA1 hash sum and investigation results provided are consistent.



Number of hard drive partitions: Two Exact name and version of the operating system: Linux (Debian) and Linux (Ubuntu)





Listing

Operating System Information

Table | Thumbnail | Summary

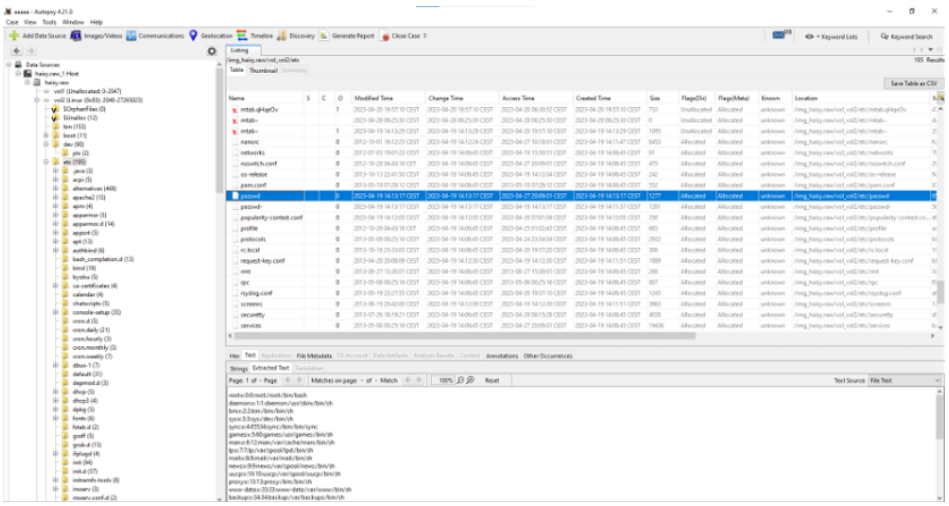
Source Name	S	C	O	Program Name	Data Source
debian_version				Linux (Debian)	haisy.raw
lsb-release				Linux (Ubuntu)	haisy.raw

Hex | Text | Application | Source File Metadata | OS Account | Data Artifacts | Analysis Results | Context | Annotations | Other

Metadata

Name:	/img_haisy.raw/vol_vol2/etc/debian_version
Type:	File System
MIME Type:	text/plain
Size:	11
File Name Allocation:	Allocated
Metadata Allocation:	Allocated
Modified:	2012-10-28 04:43:19 CET
Accessed:	2012-10-28 04:43:19 CET
Created:	2023-04-19 14:06:45 CEST
Changed:	2023-04-19 14:06:45 CEST
MD5:	931870fda5e3f942afc004db670b3cae
SHA-256:	4a45b84c771022c900f30a2da9c749176e27ecdb0e9f7bc81ce9c2cd29b84993
Hash Lookup Results:	UNKNOWN
Internal ID:	1584

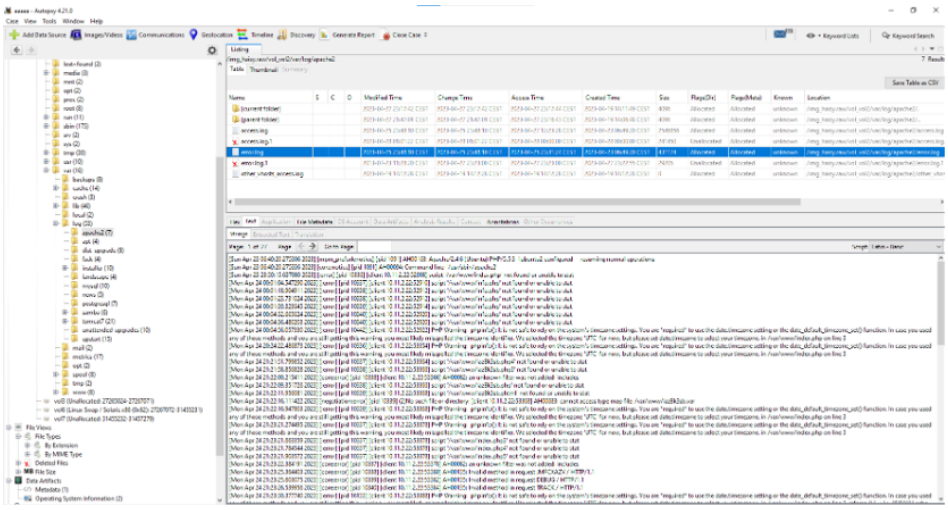
User Accounts



In the `/etc/passwd` file, each line represents information about a user account, containing various attributes:

- `root`: The superuser account with full access to the system.
- `daemon`: An account used by system processes.
- `bin`, `sys`, `sync`, `games`, `man`, `lp`, `mail`, `news`, `uucp`, `proxy`, `www-data`, `backup`, `list`, `irc`, `gnats`: These are accounts used by various system services.
- `nobody`: A special account for running system services with minimal privileges.
- `libuuid`, `syslog`, `mysql`, `messagebus`, `bind`, `landscape`, `sshd`, `postgres`, `whoopsie`, `tomcat7`, `erika`: These accounts are also used by system services or applications, each serving a specific purpose.

Suspicious Activities Found in Command History and System Log Files

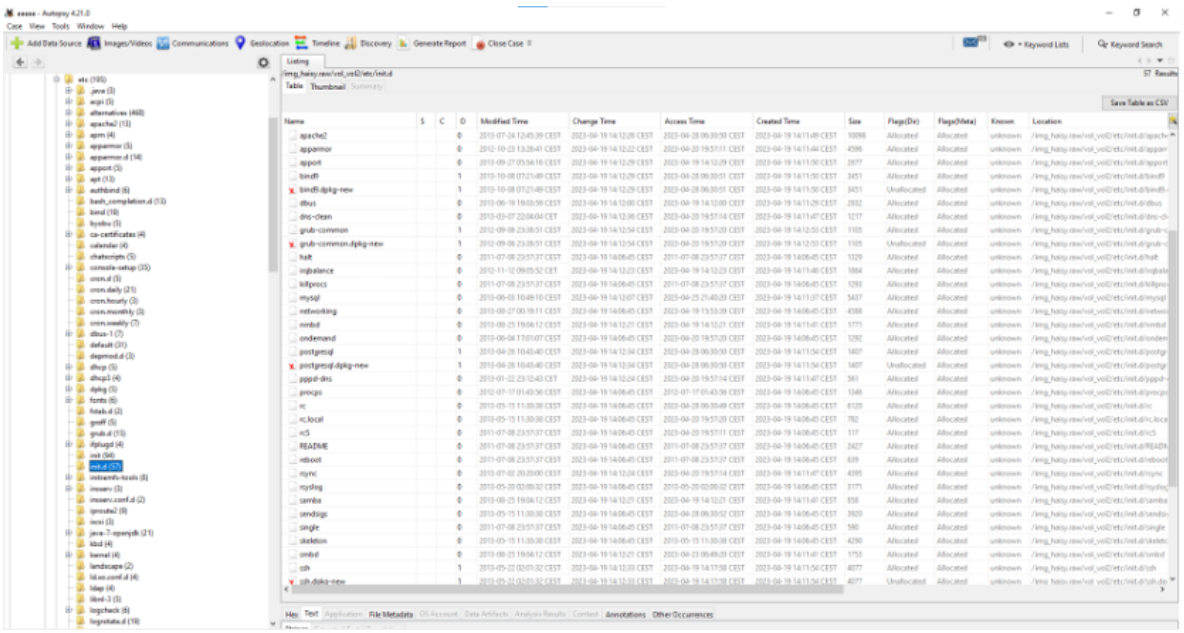


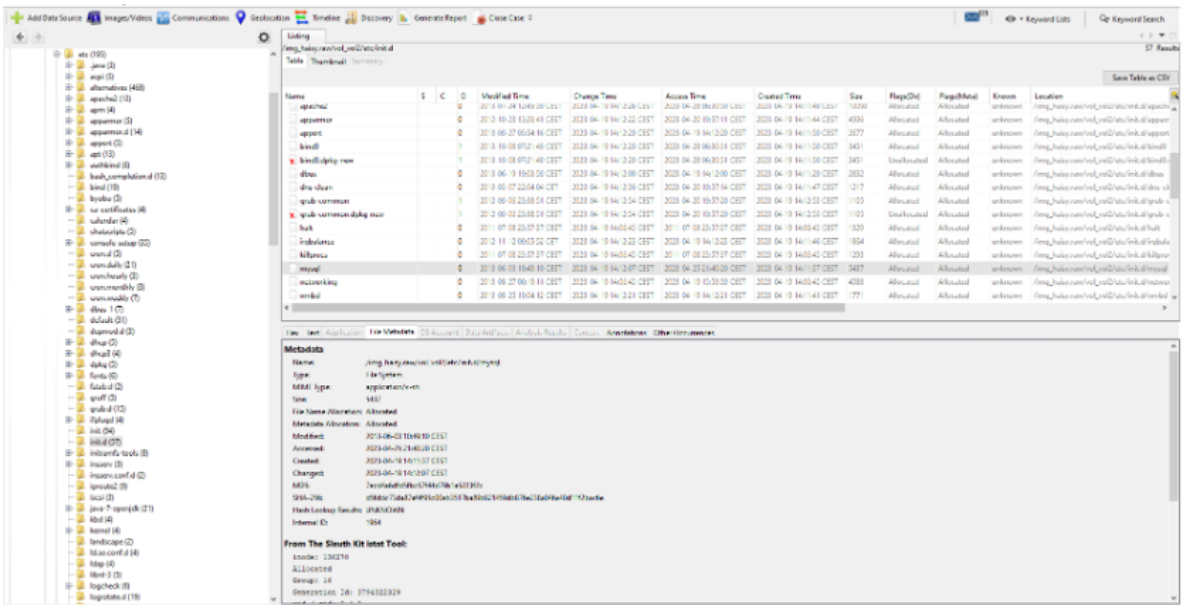
The Apache2 error log (`error.log`) contains multiple entries indicating attempts to access non-existent PHP scripts on the server, such as `index.php`, `info.php`, `login.php`, `server.php`, and `test_cgi.php`. These entries suggest potential exploitation attempts targeting vulnerabilities in the server's PHP environment.

Additionally, there are entries in the Apache2 error log indicating "Invalid method in request" errors, with unexpected methods like `JMFCXAZV`, `DEBUG`, and `TRACK`. These errors imply that the server received requests with unusual or unrecognized HTTP methods, possibly indicating attempted attacks.

The server logs between 21:24:44 and 21:26:09 on April 24, 2023, reveal a significant volume of requests attempting to access non-existent PHP scripts sequentially. Examples include requests for `upload.php`, `soinfo.php`, and `modules.php`. These sequential requests indicate a systematic probing of the server for vulnerabilities, which is often a precursor to an intrusion attempt.

Other Running Services

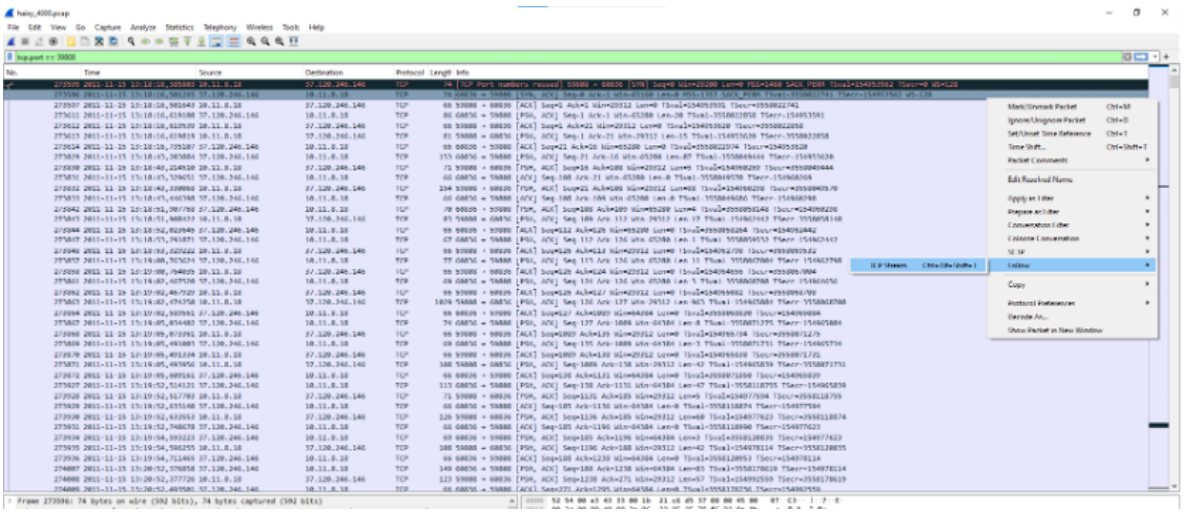




There are indications that other services were running on the server besides Tomcat and Apache. Several files in the `/etc/init.d/` directory suggests the presence of various services:

- `bind9`: Indicates the presence of a DNS server.
- `cron`: Indicates the presence of a time-based job scheduler commonly used in Unix-like operating systems.
- `grub-common`: Related to the GRand Unified Bootloader (GRUB), suggesting bootloader-related functionalities.
- `mysql`: Indicates the presence of MySQL database server.

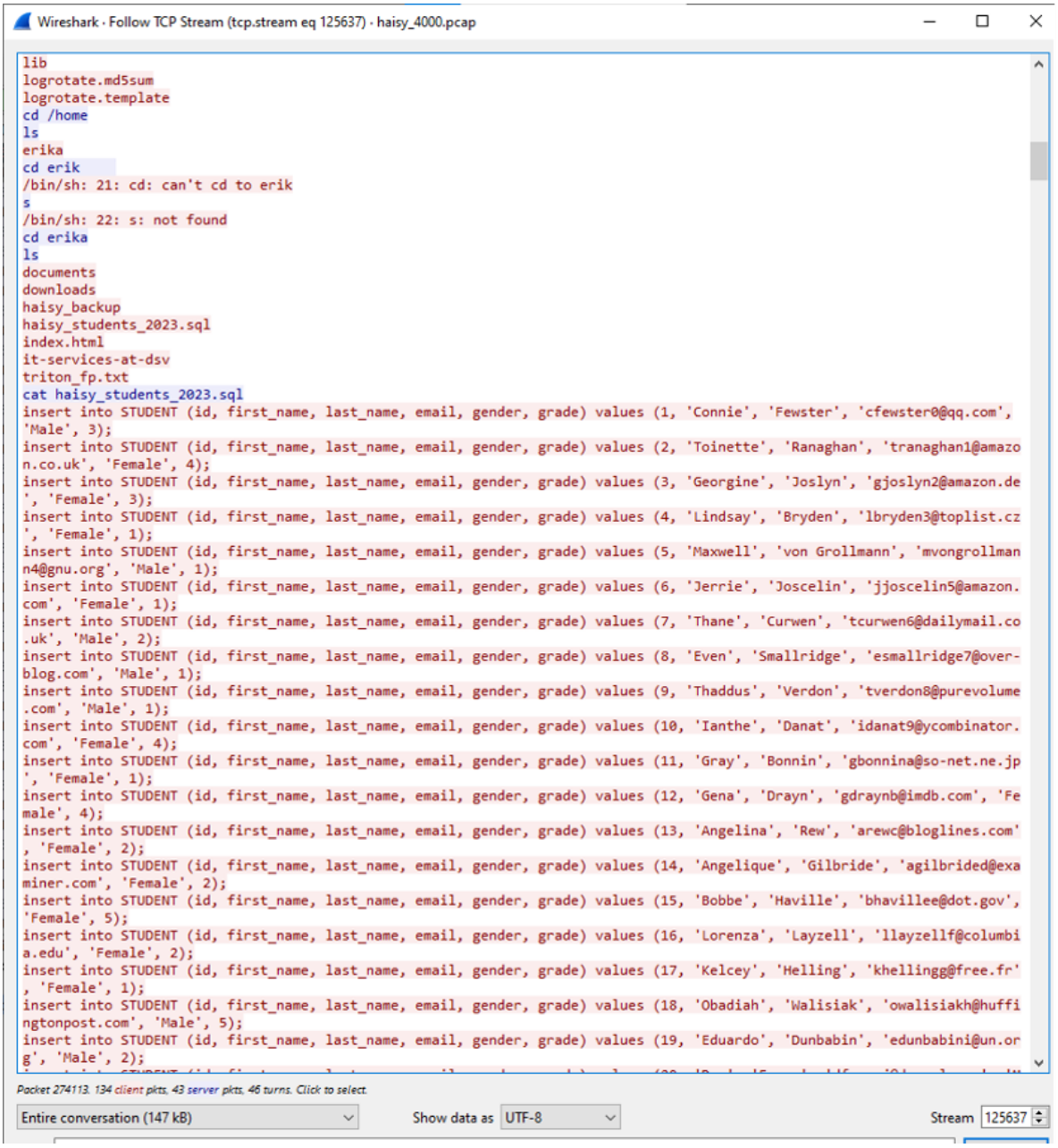
Potential Attacks on Tomcat and Apache Services in Network Traffic Analysis



Wireshark · Follow TCP Stream (tcp.stream eq 125637) · haisy_4000.pcap

```
echo EqeiWsh02lzk10
EqeiWsh02lzk10
sed -i 's/Haisy/H4kked by anonymous-oceania/' /var/lib/tomcat7/webapps/ROOT/index.html
sed: couldn't open temporary file /var/lib/tomcat7/webapps/ROOT/sedusS0RT: Permission denied
pwd
/var/lib/tomcat7

cd webapps
ls
0O5o6CNUcphQQeMkYjzDzNpbc
0O5o6CNUcphQQeMkYjzDzNpbc.war
1tkDtZQOnS5c7V
1tkDtZQOnS5c7V.war
2KciAgnNB
2KciAgnNB.war
48JIuvfBPAjb60Q
48JIuvfBPAjb60Q.war
5KXbJwkVILM4nYVHZWD
5KXbJwkVILM4nYVHZWD.war
5vJDbRFfeQqXFIzPZW16zxQWHSDs8z
5vJDbRFfeQqXFIzPZW16zxQWHSDs8z.war
6wlC
6wlC.war
8E2g2PCvnf
8E2g2PCvnf.war
CXMmw
CXMmw.war
FhyqHx
FhyqHx.war
GyMkvDi
GyMkvDi.war
jUju9OPG0O3Sy77NstoPl
jUju9OPG0O3Sy77NstoPl.war
NPFw
NPFw.war
PE08J
PE08J.war
PoqEC5Bb2IFJ0vBGpkBttho
PoqEC5Bb2IFJ0vBGpkBttho.war
ROOT
suBM
suBM.war
T63XAFXI3yDY
T63XAFXI3yDY.war
TASX4nVcVTtcma6kiD4lmVqgRn0ekh
TASX4nVcVTtcma6kiD4lmVqgRn0ekh.war
VHG9PCAa4cw1i4
VHG9PCAa4cw1i4.war
vnVI7FWNwNIbobA
vnVI7FWNwNIbobA.war
vpv9gizxGxohZNGCN1B5vUP
vpv9gizxGxohZNGCN1B5vUP.war
vxecCFa
vxecCFa.war
wSkGuSeBY6kx7
wSkGuSeBY6kx7.war
wz8qz5OsEkcW5OnOPR6iN8
wz8qz5OsEkcW5OnOPR6iN8.war
```



There are indications that the Tomcat server may have been attacked. Upon filtering TCP protocol records in Wireshark using the filter 'tcp.port == 59808', suspicious activities were observed in the TCP stream.

Specifically, there were attempts to modify files within the Tomcat web server directory ('/var/lib/-tomcat7/webapps/ROOT/'). The commands aimed to modify the 'index.html' file but encountered permission-denied errors. Additionally, there were attempts to change the directory to '/home/erika', resulting in errors due to incorrect directory names.

These errors suggest unauthorized access attempts to modify web files on the Tomcat server, indicating a potential attack.

Timeline of Network Attack on IP Address 10.11.8.18

- **2011.11.15 12:24:39:** Initial observation of [RST, ACK] packets from IP address 37.120.246.146

to IP address 10.11.8.18 indicates the start of a potential attack.

- **2011.11.15 13:18:16:** Attack successful. SYN-ACK packets exchanged between the client (IP address 37.120.246.146) and the server (IP address 10.11.8.18) confirm the establishment of a network connection.

3.4 Discussion

The forensic investigation conducted on the network traffic capture file and the Linux server disk image yielded significant insights into potential security breaches affecting the Haisy student management system. The analysis of these digital artifacts provided a comprehensive understanding of the nature and extent of the suspected intrusion attempts.

The examination of network traffic revealed several noteworthy observations. The presence of multiple IP addresses generating significant traffic, along with the identification of common ports used, indicated potential attackers' activities. Furthermore, geolocating some of these IP addresses provided valuable context, potentially attributing the origin of the attacks to specific geographic locations.

The analysis of DNS and HTTP traffic uncovered both interesting and suspicious queries and requests. While some queries appeared to be standard, others raised suspicion due to their nature or the responses received. This suggests that the attackers may have been probing for vulnerabilities or attempting to exploit known weaknesses in the system. The inspection of TCP streams revealed patterns of potentially malicious behavior, such as repeated SYN-ACK packets, indicating possible network attacks. These findings suggest that the attackers may have been attempting to establish unauthorized connections or disrupt normal network operations.

Parallel to the network traffic analysis, The examination of the Linux server disk image provided insights into the server's configuration, user accounts, system logs, and running services. Suspicious activities found in command history and system log files, such as repeated attempts to access non-existent PHP scripts or unexpected HTTP request methods, indicated potential intrusion attempts aimed at exploiting vulnerabilities in the server environment. The findings from the investigation underscore the importance of maintaining a vigilant approach to cybersecurity and implementing comprehensive security measures to safeguard critical systems and data from potential threats.

3.5 Conclusion

In light of these findings above, it is evident that proactive measures are imperative to bolster the security posture of the Haisy system. This necessitates the implementation of robust network security measures, including intrusion detection systems and firewalls, to actively monitor and prevent unauthorized access and malicious activities. Additionally, regular security audits and updates to server configurations and applications are paramount to mitigate vulnerabilities and minimize the risk of future attacks.

The findings from the forensic investigation underscore the critical importance of maintaining a vigilant approach to cybersecurity and implementing comprehensive security measures to safeguard critical systems and data from potential threats. Organizations must remain proactive in identifying and addressing security vulnerabilities to ensure the integrity and confidentiality of sensitive information.

References

1. M. Berry, "Use certutil to get file hash," Mar. 2017. [Online]. Available: https://www.mcbsys.com/blog/2017/03/use-certutil-to-get-file-hash/
2. A. Praseed and P. S. Thilagam, "DDoS attacks at the application layer: Challenges and research perspectives for safeguarding web applications," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 661-685, 2018.

3. D. Nashat, X. Jiang, and S. Horiguchi, "Detecting SYN Flooding Agents under Any Type of IP Spoofing," in *2008 IEEE International Conference on e-Business Engineering*, Xi'an, China, 2008, pp. 499-505, doi: 10.1109/ICEBE.2008.18.

4. M. Bishop, *Introduction to computer security.* Addison-Wesley Professional, 2004.

5. Linode, "Linux Users and Groups." [Online]. Available: https://www.linode.com/docs/guides/linux-users-and-groups/