

Lab Assignment: Network Forensics

Cyber Forensics (CYFO) Stockholm University

Department of Computer and Systems Sciences

Team Members:

Samir Hossain Santo Bepu-(samirhossain924@yahoo.com)

Pavan Gupta - (pagu5230@su.se)

April 2024

Assignment 1

First, we checked the integrity of the 1.pcap, 2.pcap, and 3.pcap files by running the command of md5sum and sha1sum on the terminal and found the exact hash code mentioned in the assignment lab instructions. Therefore the integrity of the file was preserved.

Evidence file : 1.pcap

MD5 Hash : 47451679a42fc2a5a637886e97fd7283

SHA-1 Hash : 4623636b88b6293888a3ebcb75cffb767bd11094

```
(kali㉿kali)-[~/Downloads]
$ sha1sum 1.pcap
4623636b88b6293888a3ebcb75cffb767bd11094 1.pcap

(kali㉿kali)-[~/Downloads]
$ md5sum 1.pcap
47451679a42fc2a5a637886e97fd7283 1.pcap

(kali㉿kali)-[~/Downloads]
$
```

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

The source ip of the suspicious traffic is 192.0.2.245, 192.0.2.196, 192.0.2.6, 192.0.2.120, 192.0.2.154, and 192.0.2.236.

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

The destination IP of the suspicious traffic was 192.0.2.2

3. What is the transport layer protocol used?

The transport layer protocol used is TCP because all the packets are marked with the SYN flag, which is essential for the TCP three-way handshake. This handshake is a distinctive process used to initiate a connection between two hosts.

4. What is/are the source port(s)?

The source ports are: 44463, 51136, 20920, 52048, and 46528

5. What is/are the destination port(s)?

The destination ports are: 58034, 62694, 46680, 43120, and 19043

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

The given attack is a SYN flood attack. All the packets we're seeing are TCP packets with the SYN flag activated. This flag is crucial in the TCP three-way handshake, signifying the start of a connection attempt. However, we're not observing any ACK packets following the SYN ones. In a SYN flood attack, perpetrators flood a server with numerous SYN packets, consuming its resources. The server, expecting

ACK responses, holds these connections open, ultimately leading to a denial of service for legitimate users. As there are multiple IP addresses found sending requests to the destination address IP 192.0.2.2 within a short period of time, we could conclude that this is a kind of DOS attack that makes the server unavailable for legitimate requests.

Evidence file : 2.pcap

MD5 Hash : 19633e3a2a3d4c315994fddc3ce7090f

SHA-1 Hash : f9d5be156ca124b46450910d2b7b1e79f2f6825c

```
(kali㉿kali)-[~/Downloads]
$ sha1sum 2.pcap
f9d5be156ca124b46450910d2b7b1e79f2f6825c 2.pcap

(kali㉿kali)-[~/Downloads]
$ md5sum 2.pcap
19633e3a2a3d4c315994fddc3ce7090f 2.pcap
```

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

According to our observation, the suspicious source MAC address is 00:11:22:33:44:55

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

The destination MAC address of where the suspicious traffic is mostly redirected towards was Intel 83 : 13 : e8(00 : 0e : 0c : 83 : 13 : e8)

3. What is the link layer protocol used?

The Address Resolution Protocol (ARP) was used.

4. What is the purpose of this protocol?

ARP is used in the link layer for each computer to build a table of what IP addresses are associated with what MAC addresses. The entire purpose of this protocol is that data sent over the link goes to the correct MAC address.

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

As the network was flooded by lots of ARP requests and duplicating IP addresses it could be an ARP poisoning attack. This attack could be used to launch attacks like MITM(man in the middle), DOS(denail of services), and session hijacking.

Source:

Grimmick, R. (2022, August 4). *ARP Poisoning: What it is & How to Prevent ARP Spoofing Attacks*.

<https://www.varonis.com>. <https://www.varonis.com/blog/arp-poisoning>

Evidence file : 3.pcap

MD5 Hash : 0944977919541d4ee176450b7ce36f9d

SHA-1 Hash : 7349e1fea8e6ed6b4dce3f89898b1c6492f3a610

```

(kali㉿kali)-[~/Downloads]
$ sha1sum 3.pcap
7349e1fea8e6ed6b4dce3f89898b1c6492f3a610 3.pcap

(kali㉿kali)-[~/Downloads]
$ md5sum 3.pcap
0944977919541d4ee176450b7ce36f9d 3.pcap

(kali㉿kali)-[~/Downloads]

```

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

The source ip of the suspicious traffic was 10.0.23.109

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

The destination ip of the suspicious traffic was **80.237.98.132**

3. What is the transport layer protocol used?

TCP protocol was used.

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.

Based on the provided packet, this activity seems like a port scanning technique called SYN scanning. The aim of SYN scanning is to find out which ports on the target host are open. If the port is open, the target host sends back a SYN-ACK packet in response to a SYN packet. The host replies with an RST (reset) message if the port is closed. Through the analysis of the responses, or lack thereof, the attacker can determine which ports are open and which services might be susceptible. The packets consistently have the syn flag set in the TCP header indicating an attempt to initiate a connection. Moreover, there is no prove of a TCP handshake only the initial syn packet was sent. The packets target multiple destination ports, indicating scanning behavior for the identification of open ports. These characteristics align with the behavior of SYN scanning, where the attacker sends SYN packets to probe for open ports without completing the TCP handshake.

Source:

Hanna, K. T. (2021, July 22). *SYN scanning*. Networking.

<https://www.techtarget.com/searchnetworking/definition/SYN-scanning#:~:text=SYN%20scanning%20is%20a%20tactic,known%20as%20half%20open%20scanning>.

Assignment 2

Introduction

Network traffic was examined at various points following a reported incident involving potential login credential theft. The objective was to confirm or refute suspicions of a malicious actor conducting Man-In-The-Middle (MITM) attacks within the private network. The prevailing hypothesis suggests that the attacker gained control of the firewall and access point during the attack. Three packet captures within the company's internal network were analyzed to investigate this matter.

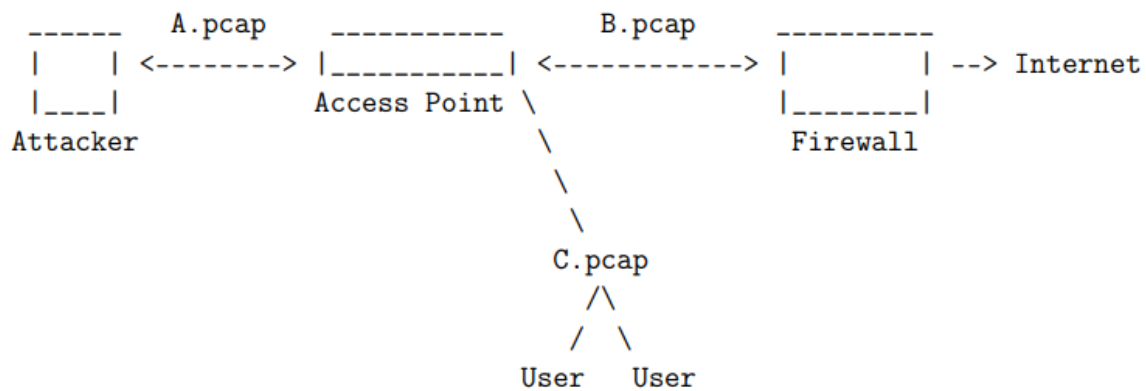


Fig. 1 - Depiction of network topology for A.pcap, B.pcap, and C.pcap.

File Name: A.pcap

SHA1: 8d5fa66a0a32c3dc6769205a26a922cd5e4ef0e6

MD5: 16dd44ba4d8842a5ffde82ba743f5c9c

```
(kali㉿kali)-[~/Downloads]
$ sha1sum A.pcap
8d5fa66a0a32c3dc6769205a26a922cd5e4ef0e6  A.pcap

(kali㉿kali)-[~/Downloads]
$ md5sum A.pcap
16dd44ba4d8842a5ffde82ba743f5c9c  A.pcap
```

File Name: B.pcap

SHA1: 9689f1283ea8e31b6d8a99eec957d9e4fc9deb67

MD5: 66ba2bb4b2cd73144ea1066d3bd2de8b

```
(kali㉿kali)-[~/Downloads]
$ sha1sum B.pcap
9689f1283ea8e31b6d8a99eec957d9e4fc9deb67  B.pcap

(kali㉿kali)-[~/Downloads]
$ md5sum B.pcap
66ba2bb4b2cd73144ea1066d3bd2de8b  B.pcap

(kali㉿kali)-[~/Downloads]
```

File Name: C.pcap

SHA1: 34a9a2422c6eca664a172c191a4c9c74af9fcef1

MD5: 9f89be02cf88ef530df2d2930df1553d

```
(kali㉿kali)-[~/Downloads]
$ sha1sum C.pcap
34a9a2422c6eca664a172c191a4c9c74af9fcef1  C.pcap

(kali㉿kali)-[~/Downloads]
$ md5sum C.pcap
9f89be02cf88ef530df2d2930df1553d  C.pcap
```

Figure 2.

Methodology-

First, we need to ensure the integrity of the pcap files by verifying their hashes. Assuming a Man-In-The-Middle (MITM) attack occurred, with both the attacker and victim connected to the same network, we'll examine three pcap files: B.pcap (between an access point and firewall), A.pcap (attacker), and C.pcap (victim). Using Wireshark, we'll search for connections between the attacker and victim, focusing on potential ARP poisoning instances. The investigation methodically checked file integrity, SSIDs, encryption, and decryption keys, identifying IP/MAC addresses of the attacker, access points, firewalls, and potential HTTP intrusions. C.pcap was used to determine the victim's IP/MAC addresses and analyse ARP poisoning. Financial website access and source of responses were investigated to reconstruct the website. Finally, an incident timeline was compiled for a comprehensive understanding.

Results-

Examine A.pcap:

From our observation, we saw that D-link has the most traffic. On investigating further it reveals that the SSID was DSI_DSV and the BSSID was 00:13:46:48:b0:f9. When we further investigated the packet in depth we found that it's using WEP encryption. It was specified that it had an Initialization Vector: 0x6d1a00 in the WEP parameters unique RC4 key is created for every packet by utilizing an Initialization Vector (IV)., according to (Borisov et al). WEB encryption protects the data by scrambling data into a secret code that can only be unlocked with a single digital key. To extract the WEP key we use the tool called "aircrack-ng". By using this toolkit, we able to decrypt the key, which was 44:53:49:4c:41 (ASCII: DSILA). After decrypting the key, we entered it in wireshark-preferences/protocols/802.11/decrypt key/key. As a result of the decryption key being stored in Wireshark, the packets were now automatically decrypting. When the description was complete, we were able to read the content of every single packet which was not possible due to encryption. Before decryption, all packets were categorised under the protocol column as 802.11. However, after decryption, the protocol columns exhibited the specific protocols, such as TCP, associated with the packets.

Both IPs 192.168.1.201 and 74.125.15.19 have most of the traffic. The external IP address involved was 74.125.15.19, which was using port 80, transmitted address was DLink_48:b0:f9 and acts as a web server.

Examine B.pcap:

The analysis of the .pcap file revealed potential security threats targeting the network's firewall and access point. The firewall was identified with an IP address of 192.168.1.100 and a MAC address of 08:00:27:d2:f8:61, while the access point's IP was 192.168.1.201, with a MAC address of 00:1f:3c:6e:49:24. Examining the packet data indicated suspicious activity, suggesting a UDP port scan followed by a TCP Connect() scan, often signaling an attack.

Further inspection of HTTP traffic showed a pattern of unauthorized login attempts on the firewall, with attackers using common dictionary-based brute force methods. Attempts with simple credentials like "admin:admin" eventually led to a successful login with "admin:DSILA." This suggests that a dictionary attack was employed to breach the network's security, providing unauthorized access to sensitive systems.

Examine C.pcap:

After opening the C.pcap file it was easy to identify the website the client attempted to visit was www.nordea.se. This is evident from packet number 29, where the DNS query for "www.nordea.se" is made. The response that the system got was "HewlettPacka_32:a9:13" MAC- 18:a9:05:32:a9:13 and IP- 192.168.1.98. We were able to re-construct the page, but only partially. Although the text and link sections were recovered, the picture, CSS, and JS elements could not be successfully reconstructed.

The timeline for the attack is as follows.

Timeline

Firewall penetrated by the attacker: Mar 2, 2011, 19:22:46.018518000 CET on frame 51 of B.pcap
Access point penetrated by the attacker: Mar 2, 2011, 19:23:56.596112000 CET on frame 1628 of B.pcap
Arp poisoning starts: Mar 2, 2011, 18:58:52.505853000 CET on frame 11 of C.pcap
The user tries to log in to Nordea:
DNS query: Mar 2, 2011, 18:58:53.946952000 CET on frame 29 of C.pcap
HTTP request: Mar 2, 2011, 18:58:53.953637000 CET on frame 39 of C.pcap

Explanation of the timeline

On March 2, 2011, at 18:58:52.505853 CET (frame 11 of C.pcap), an attacker begins ARP (Address Resolution Protocol) poisoning, a tactic that involves sending deceptive ARP messages to a local network, potentially letting the attacker intercept or disrupt communications. Right after this, at 18:58:53.946952 CET (frame 29 of C.pcap), a DNS (Domain Name System) query is made as a user attempts to access the Nordea website, converting the domain name to an IP address. Almost at the same time, an HTTP (HyperText Transfer Protocol) request to the Nordea website is captured at 18:58:53.953637 CET (frame 39 of C.pcap), suggesting that the user is trying to connect and possibly log in.

Later in the day, at 19:22:46.018518 CET (frame 51 of B.pcap), the attacker successfully penetrates the network's firewall, marking a critical security breach that allows unauthorized access to the internal network. Shortly thereafter, at 19:23:56.596112 CET (frame 1628 of B.pcap), the attacker infiltrates the access point, which likely represents the attack's culmination, granting broader access to the network and its connected devices.

Discussion-

Upon analysing the acquired data, it's evident that the network fell victim to an attack. The attacker, connected to SSID DSI_DSV (IP: 192.168.1.201, MAC: 00:1f:3c:6e:49:24), executed ARP poisoning, assuming the role of the Man in the Middle. Subsequently, a user (IP: 192.168.1.98, MAC: 18:a9:05:32:a9:13) attempted to access nordea.se but received a manipulated login page from the attacker. When the user logged in, the attacker intercepted the credentials. The attacker breached the firewall and access point, likely accessing nordea.se independently and manipulating funds. To mitigate risks, the user should change their password promptly. For network administrators, it's imperative to update firewall policies and strengthen passwords to thwart future attacks.

Assignment 3: Intrusion Analysis

Introduction-

From the subject of the email from CERT it can be observed that some unusual activity from one or more computers within the IT department of DSV. Therefore, it's possible that the student management and database system Haisy could have been targeted in an attack, potentially leading to unauthorised access to personal student information, compromising the integrity of the data, or both. To determine whether a breach or loss of sensitive data has occurred, this investigation will analyse two key sources: the pcap trace file and the Haisy server image.

IP: 193.10.9.5

Computer name: haisy.cs2lab.dsv.su.se

Attack type: Possible intrusion attempt

Time, from circa: 2023-04-27 23:50:25 CET

As we can see, the location of the Haisy server and the IT network of DSV are in the given instruction document.

Internet <-> Router <-> Reverse proxy <-> haisy.cs2lab.dsv.su.se

Router: 193.10.9.5

Reverse Proxy: 10.11.2.22

haisy.cs2lab.dsv.su.se: 10.11.8.18

Methodology-

The investigation will be primarily based on these two files: haisy_400.pcap and haisy.raw. Firstly, it's crucial to verify the integrity of the files by checking their hashes. Then, examine the IPv4 addresses contained in the pcap file, prioritizing those with the highest traffic, which may indicate a brute force attack. Validate the geolocation of non-internal IPv4 addresses to identify any notable origins. Next, investigate the IPv4 addresses and ports involved in communications, looking for signs of port monitoring. Review the network traffic file for unusual activity like DNS or HTTP traffic. Moving on to the server image, ensure the integrity of the image file by checking its hash sums. Examine user-created accounts to assess access to the server and review command history for any signs of infiltration. Finally, analyze system log files for evidence of compromised user accounts and utilise Autopsy's timeline function to track visited files during the incident.

Results

1. A forensic process should always begin with an identification of the evidence. What is the SHA1 hash sum of the file? What is the exact byte size of the file?

With the help of the basic command in Linux as shown in the image given below, we found the SHA1 hash sum of the file haisy_400.pcap, and then we compared it with the hash given in the lab instruction file, and it matches. To find the exact byte size of the file, we ran the command stat -c %s haisy_400.pcap and we got 35116793 bytes.

```
(kali㉿kali)-[~/Downloads]
└─$ sha1sum haisy_4000.pcap
5d50246cd8ed94b9d39d60b4008a2ead1e3cba50  haisy_4000.pcap

(kali㉿kali)-[~/Downloads]
└─$ md5sum haisy_4000.pcap
8f7f17adf4de26e88dd2841dca174b02  haisy_4000.pcap

(kali㉿kali)-[~/Downloads]
└─$
```

Figure 3.4.1

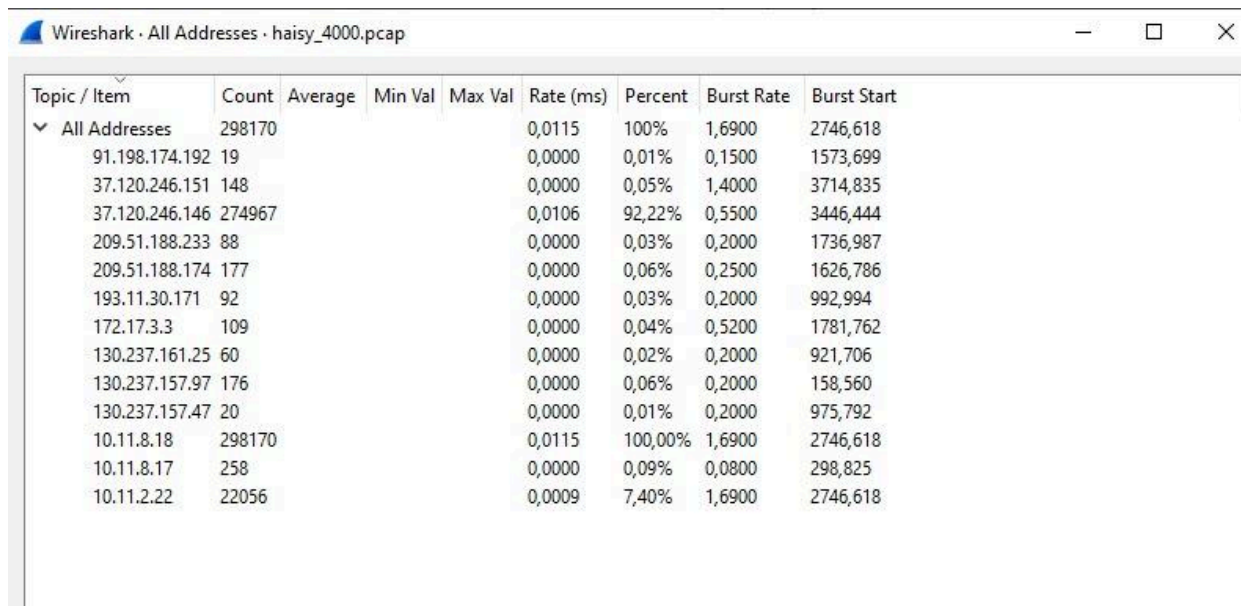
File Name: haisy_4000.pcap
SHA1: 5d50246cd8ed94b9d39d60b4008a2ead1e3cba50
MD5: 8f7f17adf4de26e88dd2841dca174b02

Router: 193.10.9.5

Reverse Proxy: 10.11.2.22

haisy.cs2lab.dsv.su.se: 10.11.8.18

2. How many IPv4 addresses are present in the .pcap file. Tip: In Wireshark, go to the Statistics tab.



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

Figure 3.4.2

From our observation, there were a total of 13 IP addresses present in the given haisy_4000.pcap file. Later we analyzed based on the “count” section and then we determined the top 5 Ip’s are:

1. 37.120.246.146 (92.22%)
2. 10.11.8.18 (100%)
3. 10.11.2.22 (7.40%)
4. 10.11.8.17 (0.09%)
5. 209.51.188.174 (0.06%)

4. Which ports were the most common ones (top five ports)?

The most five common ports are:

- 60836
- 84
- 443
- 59808
- 39140

5. Where do the top-three most occurring IP addresses reside? Geo-locate the IP addresses. Useful tools can be found in section 7.

- 37.120.246.146 - Romania
- 10.11.8.18-Sweden
- 10.11.2.22-Sweden

Based on the top 3 ips the first ip was from Romania and the remaining two was of the Haisy server and Reverse proxy. This activity is raising suspicion because it's highly unlikely for a Swedish student to access the DSV test grade server from Bucharest at this particular time. Another red flag is the low traffic received by the IP address 37.120.246.151 in Bucharest, Romania, despite its apparent involvement in the attack. Therefore, both foreign IP addresses should be closely examined for any suspicious behavior. Upon investigating the ports used in the communication initiated by 37.120.246.146, it's evident that a port-scanning activity took place. This involved communication attempts across a wide range of TCP ports, with only a few attempts at UDP ports. Most of the traffic originated from the endpoint 37.120.246.146:60836, indicating a clear case of port scanning. Although the host PC blocked most contact attempts due to restricted ports, a scan of port 59808 yielded promising results. Consequently, the subsequent message to 10.11.8.18:59808 is being thoroughly analyzed. During this investigation, TCP injections were detected in the conversation, as observed in the packet bytes displayed in the Wireshark frames.

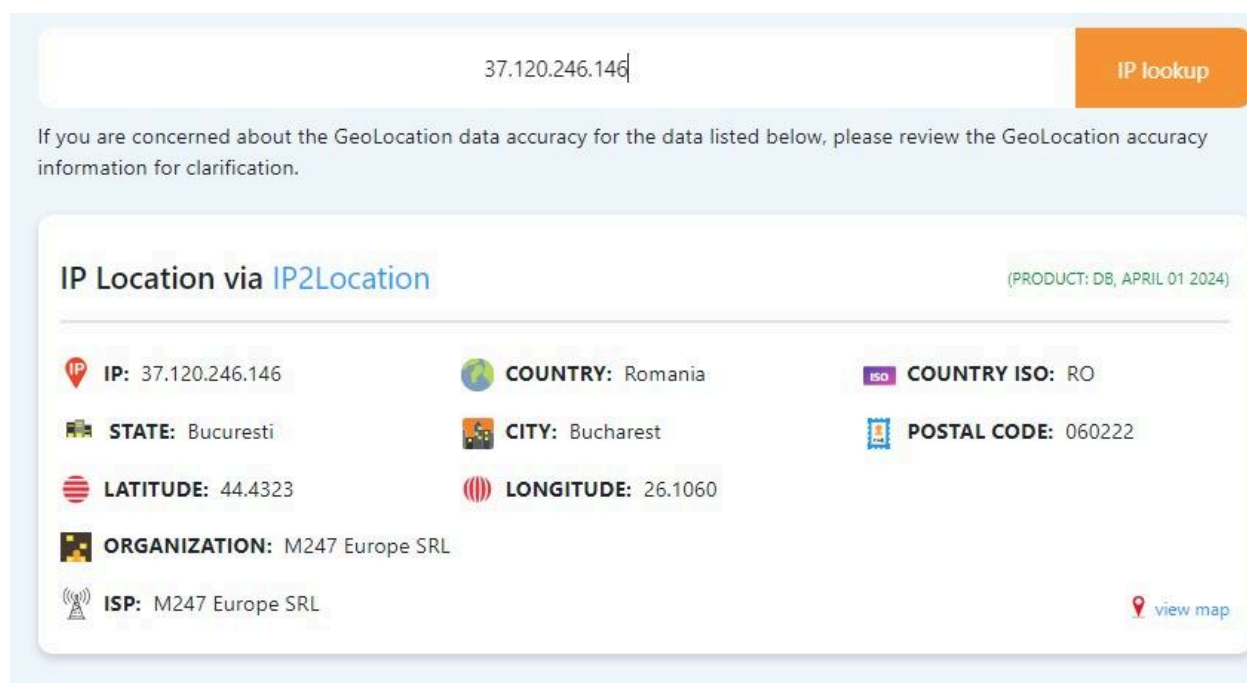


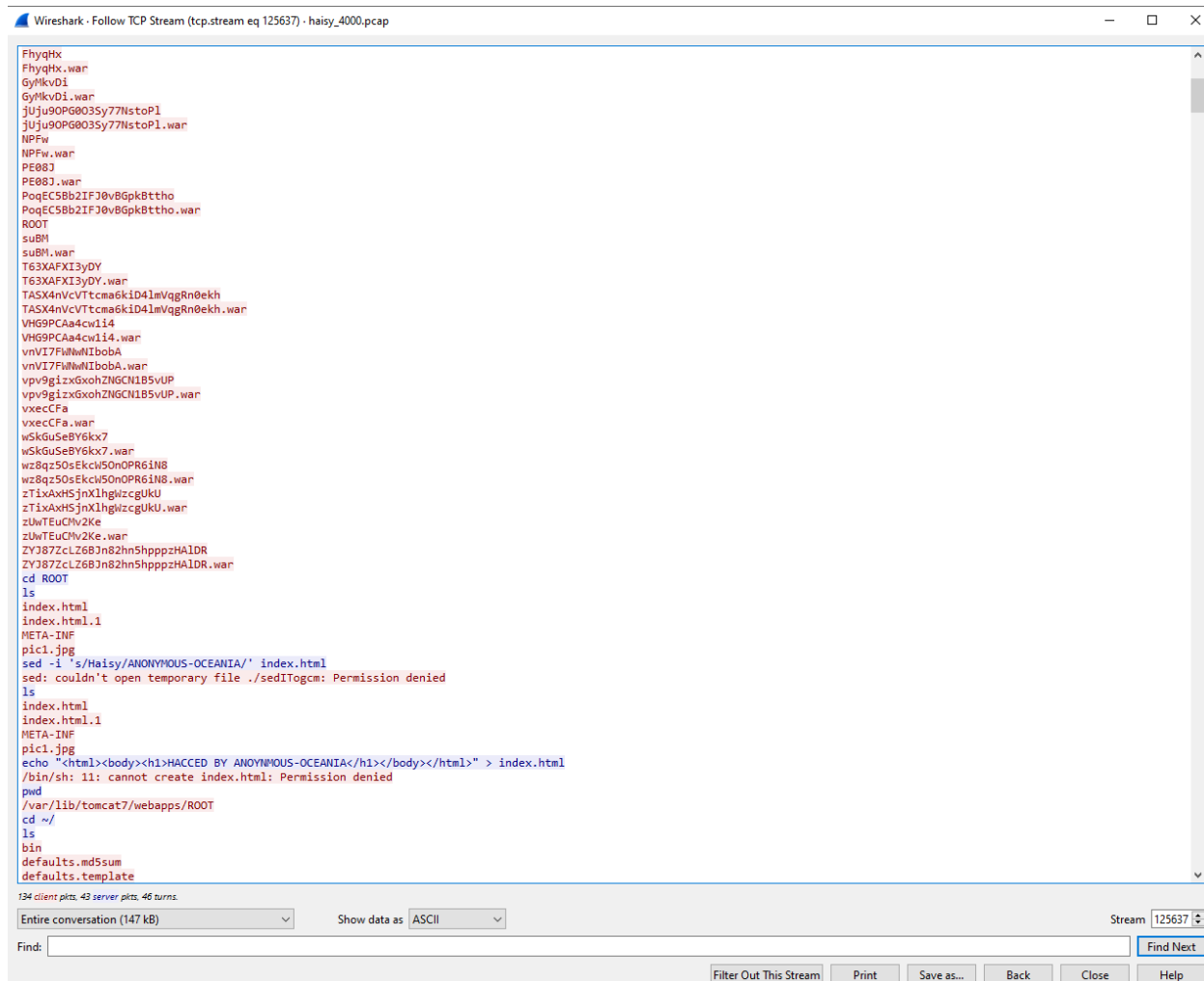
Figure 3.4.3

6. Use the filter "DNS" to find the DNS related packets. Describe any interesting DNS related traffic. Are there any suspicious DNS queries?

We looked into the traffic, and from our analysis, we did not find anything that could be marked as suspicious. The traffic was showing communication between Daisy/DSV and the web page. Now when we use the HTTP filter in Wireshark we find that there are lots of requests asking for resources on the web server that are not found therefore it seems suspicious. As the user sends requests one after another it might be a DGA Attack: Malware often uses DGAs to generate a multitude of domain names dynamically, attempting to connect to them to establish communication with its command and control (C&C) server or reconnaissance: It could also be a reconnaissance attempt, where the attacker is probing your network's DNS infrastructure to gather information for a potential future attack.

8. Use the follow protocol stream tool. Describe any interesting TCP streams. Can you find any suspicious activities? (It might be best to look at streams close to a particular time, such as after an attack or close to an event observed in haisy.raw)

When we analyzed the TCP stream in Wireshark, it could be seen in the figure 3.4.4 that that attacker was looking for something. The attacker was just opening navigating itself to multiple file to see what was inside that file.



```
FhyqHx
FhyqHx.war
GyMkvD1
GyMkvD1.war
jUju9OPG0035y77NstoP1
jUju9OPG0035y77NstoP1.war
NPFW
NPFW.war
PE08J
PE08J.war
PoqEC5BbZIF70v8Gpk8ttho
PoqEC5BbZIF70v8Gpk8ttho.war
ROOT
suBM
suBM.war
T63XAFXI3yDY
T63XAFXI3yDY.war
TASX4nVcVTtcmakId4mVqgRn0ekkh
TASX4nVcVTtcmakId4mVqgRn0ekkh.war
VHG9PCA4cwIi4
VHG9PCA4cwIi4.war
vnV17FwNwNIboba
vnV17FwNwNIboba.war
vpv9giz6xohZN6CN185vUP
vpv9giz6xohZN6CN185vUP.war
vxecCfa
vxecCfa.war
wSkGuSeBY6kx7
wSkGuSeBY6kx7.war
wz8qz50sEkck50nOPR6iN8
wz8qz50sEkck50nOPR6iN8.war
zTixAxHSJnXlghVzcgUkU
zTixAxHSJnXlghVzcgUkU.war
zUwTeuChv2Ke
zUwTeuChv2Ke.war
ZYJ87ZcLZ68Jn82hn5hpppzHA1DR
ZYJ87ZcLZ68Jn82hn5hpppzHA1DR.war
cd ROOT
ls
index.html
index.html.i
META-INF
pic1.jpg
sed -i 's/Haisy/ANONYMOUS-OCEANIA/' index.html
sed: couldn't open temporary file ./sedITogcm: Permission denied
ls
index.html
index.html.i
META-INF
pic1.jpg
echo "<html><body><h1>HACCED BY ANONYMOUS-OCEANIA</h1></body></html>" > index.html
/bin/sh: 11: cannot create index.html: Permission denied
pwd
/var/lib/tomcat7/webapps/ROOT
cd ~/
ls
bin
defaults.md5sum
defaults.template
```

194 client pkts, 43 server pkts, 46 turns.

Entire conversation (147 kB) Show data as ASCII Stream 125637

Find: Find Next

Filter Out This Stream Print Save as... Back Close Help

Figure 3.4.4

Moreover from observation, it seems like attacker was attempting to transfer the content of a SQL file (haisy_students_2023.sql) to a remote machine using TCP and UDP connections, possibly using the nc netcat utility and also the attacker reading the data which was inserted in the sql file, it could be seen in beblow figure that attacker get student name and email address.

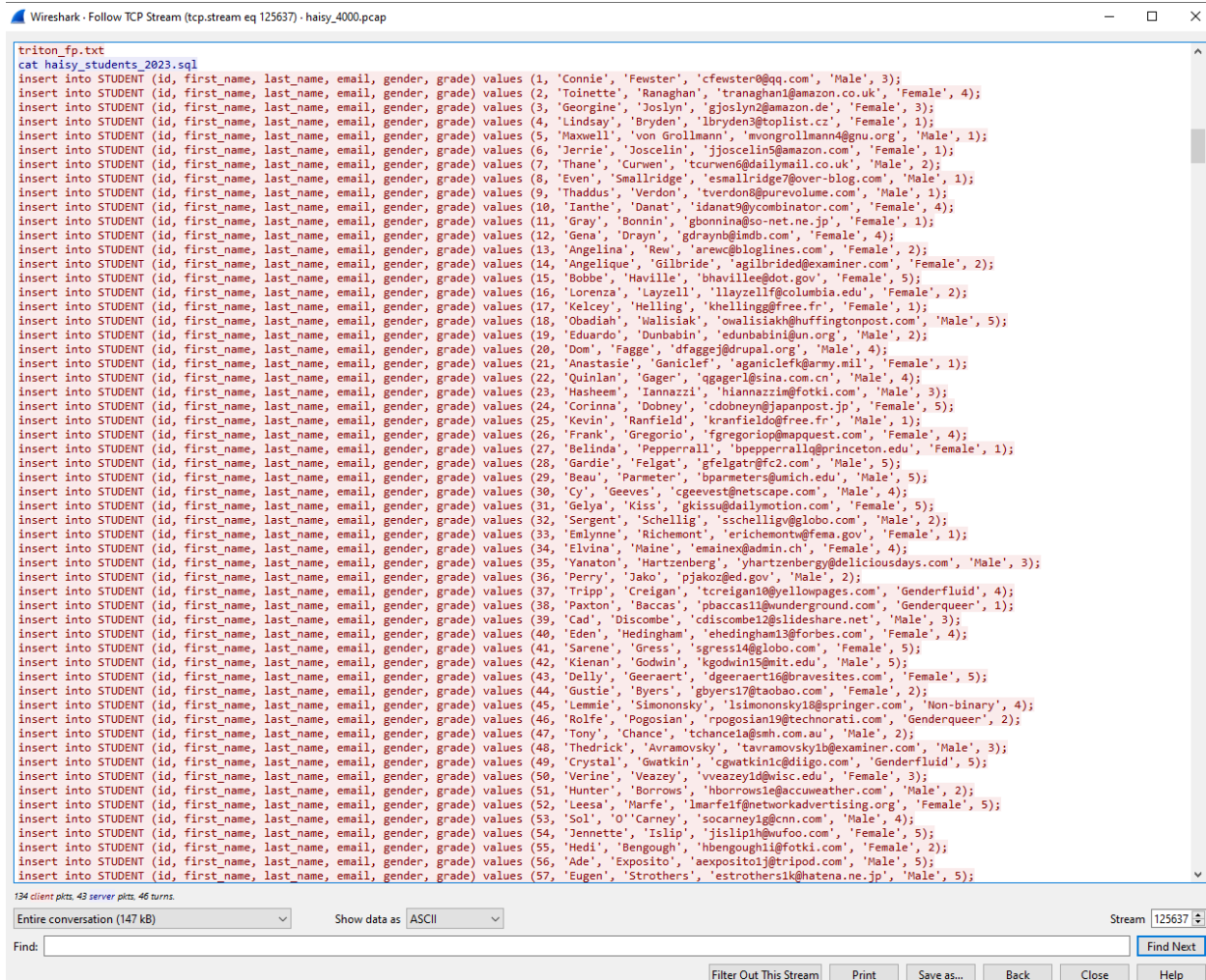


Figure 3.4.5

Source:

Communication with suspicious random domain name (Preview). (n.d.).

TECHCOMMUNITY.MICROSOFT.COM.

<https://techcommunity.microsoft.com/t5/microsoft-defender-for-cloud/communication-with-suspicious-random-domain-name-preview/m-p/2795653>

Chen, Z., & Szurdi, J. (2020, September 1). Cybersquatting: attackers mimicking domains of major

brands including Facebook, Apple, Amazon and Netflix to scam consumers. *Unit 42*.

<https://unit42.paloaltonetworks.com/cybersquatting/>

4.2

```
C:\Windows\system32>certutil -hashfile C:\CYFO\haisy.raw MD5
MD5 hash of C:\CYFO\haisy.raw:
89fd1b9b40f2b7793440a4a13d045837
CertUtil: -hashfile command completed successfully.

C:\Windows\system32>certutil -hashfile C:\CYFO\haisy.raw SHA1
SHA1 hash of C:\CYFO\haisy.raw:
6d08e3ec0c3caac2979070913010c1753c48f66f
CertUtil: -hashfile command completed successfully.
```

File Name: haisy.raw

MD5: 89fd1b9b40f2b7793440a4a13d045837

SHA1: 6d08e3ec0c3caac2979070913010c1753c48f66f

1. A forensic process should always begin with an identification of the evidence. What is the SHA1 hash sum of the file? What is the exact byte size of the file? How many hard drive partitions are there in the image? What was the exact name and version of the operating system?Tip: look for the lsb-release file.

The exact file size was 16106127360 bytes. There are a total of 5 hard drive partitions available in the image, but three of those partitions are unallocated(vol1, vol3, vol6), and allocated hard drives are vol2 and vol6. After opening haisy.pcap inside autopsy we checked out the summary of the haisy.pcap we found that the OS name Linux(Ubuntu) version 13.10. In “etc/passwd” we found that there are root users and another main user called “ Erika”.

After that we investigate the.bash_history,syslog, and auth.log files here are our findings.

File	Commnet	Description
.bash_history	Need further investigation	It seems like there are numerous commands related to managing services, networking, web servers, and MySQL databases, along with some basic Linux system administration tasks. Regarding suspicious activity, some commands like changing the network configuration and MySQL commands, could be considered suspicious depending on authorisation.

File	Comment	Description
syslog	Need further investigation	It seems like there are several DHCP requests and renewals, which is normal network activity. However, there are also several kernel messages indicating segfaults in various processes.
auth.log	No suspicious activity found	It shows various sessions being opened and closed for the root user, indicating the execution of scheduled tasks. Additionally, there are authentication attempts using the su command, some successful and some failed, along with some attempts to switch to the postgres user. Overall, it seems to be routine system activity and user authentication events

Table 1

In addition to Tomcat and Apache services, other services were also running like MySQL, ssh, bind9, AppArmor, PostgreSQL, and skeleton.

5. Are there any indications that the Tomcat or Apache services were attacked from their log files?

Tip: look in the /var/log/directory.

Investigation of Tomcat:

The high volume of 404 (Not Found) responses for these requests suggests that the attacker may be trying to identify vulnerable endpoints or directories on the server. Moreover, repeated access attempts from the same IP address within a short timeframe, especially with a wide range of file extensions, could indicate automated scanning or a targeted attack. We also found that on different log files HTTP requests are trying to exploit vulnerabilities in the web server or web application by including remote file inclusion attack (RFI). RFI attacks involve an attacker tricking a web application into including a remote file, usually containing malicious code, into a script that is being executed on the server.

Investigation of Apache:

The abnormality in the access log lies in the presence of requests associated with Nikto, indicating deliberate scanning activity aimed at identifying potential vulnerabilities in the target server.

6. Create a timeline (e.g. a bullet-point list of events with timestamps) based on the log entries on the seized Linux system. Correlate with any packets of interest in the haisy_4000.pcap

2023-04-28 00:12:21 -> Tcp connection established on port 59808(dest port)

2023-04-27 23:49:03 -> attacker send get request to haisy

Date	Time	Description
2023-04-27	23:18:18	Significant TCP flooding was observed. It is an indication of port scanning where the source ip was 37.120.246.146 and the destination IP: was 10.11.8.18

2023-04-27	23:49:03	Packet number 2262 shows the first suspicious GET request from 172.17.3.3 to 10.11.8.18 but the server consistently responds with a redirection message..
2023-04-27	23:49:03	The attacker sends get request to haisy
2023-04-28	00:16:10	At this moment the attacker accessed the database.
2023-04-28	00:00:00	These log entries suggest that host 10.11.2.22 is attempting to exploit a vulnerability related to FCK editor by trying to access specific PHP files

The screenshot displays two forensic analysis tools. On the left, Wireshark shows a list of files with columns for Name, Size, Modified Time, Change Time, and Access Time. The file 'haisy_students_2023.sql' is highlighted. On the right, Autopsy shows a detailed view of this file, displaying its contents in a text editor. The file contains a large number of INSERT statements for a database table named 'STUDENT'.

The above image is all about showing the correlation using wireshark and autopsy. Whatever entries the attacker executes can be found in autopsy and the folders he tried to access were also found. As shown in the above image.

Discussion:

Source:

Masas, R. (2023, December 20). *What is RFI | Remote File Inclusion Example & Mitigation Methods | Imperva*. Learning Center. <https://www.imperva.com/learn/application-security/rfi-remote-file-inclusion/>