Network Security (CS4430): Project Forensic Investigation

Mahira Ali
Delft University of Technology
M.Ali-10@student.tudelft.nl

Murtuza Ali
Delft University of Technology
M.A.Mohammed-1@student.tudelft.nl

Zohar Cochavi
Delft University of Technology
Z.Cochavi@student.tudelft.nl

Giorgos Koursiounis
Delft University of Technology
G.Koursiounis@student.tudelft.nl

Vyshnavi Molakala Narasimhalu
Delft University of Technology
V.MolakalaNarasimhalu@student.tudelft.nl

Kamal Nour
Delft University of Technology
K.nour@student.tudelft.nl

ABSTRACT

In this paper, we investigate three different traffic captures with suspicious network activity, aiming to identify potential attacks and provide mitigation strategies to protect these networks. After an exploratory stage, we perform network forensic analysis using tools such Python, Tshark, and tcpdump. For the first file, this analysis showed an enormous number of identical DNS response packets in a short amount of time, indicating a DNS flood attack. The next file shows an abnormally large number of CharGen connections from various IPs, and many fragmented IP packets. The first piece of evidence provides solid proof for a CharGen flood attack, but we cannot prove the latter to be a deliberate attempt at further clogging the network. Lastly, analysis on the last file shows proof of a man-in-the-middle attack through ARP cache poisoning, and an ACK flood attack. There are also signs of a possible teardrop-attack, but, similar to the last traffic capture, this could also be a side-effect of the sheer amount of traffic running through the network. It is hard to completely exhaust all possible attacks, simply because of the sheer amount of data, but we can confidently say all files contain different D/DoS techniques.

1 INTRODUCTION

This forensic investigation aims to analyze three PCAP files for any suspicious behavior that may threaten the security of the network. The analysis will include examining the packets captured in each file and identifying any patterns or anomalies that could indicate the presence of any malicious activity. The results of this investigation will provide insights into the nature and extent of the threat, allowing for the development of effective countermeasures to mitigate any potential damage to the network.

The initial approach for analyzing the PCAP files involved observing the captured packets using various traffic analysis tools (e.g. WireShark). Then the Pyshark packet analysis library in Python was utilized to extract relevant information from the pcap files, such as packet arrival time, source, destination, protocols, and payloads. After retrieving this information, statistical analyses were conducted to identify patterns and anomalies in the packet traffic. Python's Seaborn library was used to create visualizations of the statistics, which allowed for a more intuitive understanding of the data. This approach allowed for a thorough analysis of the pcap files and provided insights into any potential malicious behavior in the network traffic.

2 BACKGROUND

Because of the exploratory nature of this kind of analysis, the necessary background is hard to identify without knowing what the threat is in the first place. To mitigate this *chicken-and-egg* problem, we will cover the most common types of attacks on larger networks, and then cover the type of threat identified in the PCAPs in more depth.

2.1 Common Attacks

There are a variety of cyber-attacks an organization, or individual, can be subject to. For this report, only network-based attacks are of significance. Meaning that attacks such as phishing, although relevant and commonplace [9], will probably not be present in these data-sets after having a first look at the data. Currently, the most common types of cyber attacks are as shown in Figure 1.

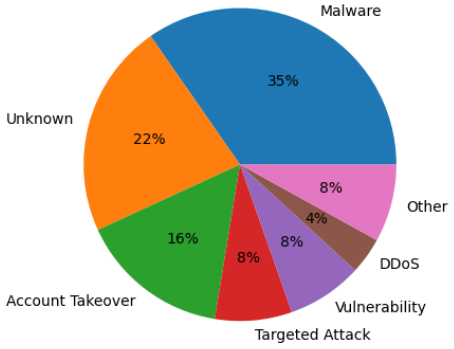

Figure 1: Proportions of different types of cyber attacks (2022) [23]. Others include attacks such as 'Malicious Script Injection', 'Defacement', 'Business Email Compromise', etc.

Malware and account takeover include a significant portion of the executed attacks over the last year. Specifically, they indicate a threat actor being able to access a network through either stolen credentials (e.g. phishing campaigns) or the installation of malware

by the target network users. While characteristics of the execution of this kind of attack might be present in the data, they are not usually associated with a network-based attack.

Targeted attacks, Vulnerability exploitation, and D/DoS attacks are more in line with the expectations for this analysis. They are prevalent and, though various characteristics, identifiable by monitoring network traffic[22]. A notable example is that of the *Log4j* (or *Log4Shell* incident [2], where HTTP requests to endpoints could be constructed to exploit a vulnerability found in the Apache Log4j library [16]. Because these exploits normally try to abuse behavior that is unaccounted for (and thus unintended), they are often very identifiable.

2.2 Foreshadowing: Denial of Service

After having analyzed the PCAP files, they all seem to indicate different types of D/DoS attacks. Some more complex than others, but all attempting to shut down the target service by overloading resources. While the list of different D/DoS attacks is barely exhaustable, different techniques all share similar characteristics [11].

Various techniques exist for detecting DoS attacks, but recently machine learning algorithms and other statistical methods have proved to be very effective [21]. But as with any ML method, guarantees in outcomes are hard to come by and therefore some human supervision is currently still required, especially when considering the sensitive nature of information security.

3 DATASETS

3.1 PCAP File 1

Considering the first PCAP file, 1.5 million packets were captured over a duration of 195 seconds. Table 1 outlines the basic statistic measurements of the captured data.

Table 1: Basic statistics PCAP file 1

Measurement	Captured size
Average pps	7729.4
Average packet size, B	125
Bytes	187,761,029
Average bytes/s	967 K
Average bits/s	7740 K

Upon initial inspection, it is evident that the DNS protocol is prevalent in the captured traffic, as a significant number of packets contain DNS as the application-layer protocol. Specifically, it is observed that the DNS packets primarily consist of response packets with a destination address of "ddostheinter.net."

Further analysis reveals that an alarmingly high number of DNS query response packets, around 7500 in total, were transmitted within a single second. Moreover, these packets' source Ethernet addresses, destination Ethernet addresses, and IP destination addresses seem to be largely identical. However, the IP source addresses of the packets appear to vary, suggesting that the source of these packets may be utilizing a botnet or other means of IP address spoofing.

Given the observations, it appears that the captured traffic in the first PCAP file indicates a distributed denial-of-service (DDoS) attack. The high volume of DNS packets suggests that either a DNS flood or DNS amplification attack is taking place. However, since there is a considerable number of DNS response packets, a DNS flood attack is the most likely scenario. Additionally, the sheer amount of packets transmitted within a second, combined with the similarities in destination and Ethernet addresses, strongly implies that the traffic is part of a coordinated effort to overwhelm the target system or network with a flood of malicious packets.

3.2 PCAP File 2

PCAP file 2 contains 1 million packets captured within approximately 16 seconds. Table 2 illustrates basic statistical measurements of the captured data. In particular, we note an average number of 59241 packets per second (pps) and an average packet size of around 1 KB each. In total, we have more than 1.2 GB of received data.

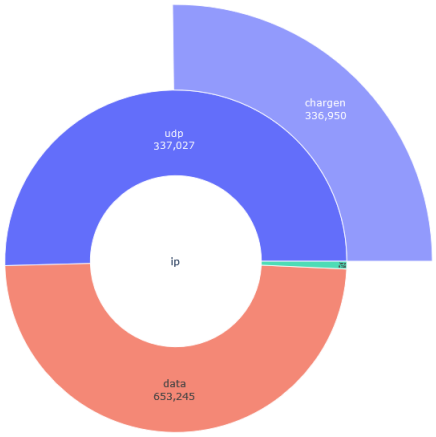

Figure 2: Protocol Hierarchy Statistics

Further inspection denotes the majority of packets are fragmented CharGen packets and a small number of ICMP segments. Figure 2 shows that the file contains 653,245 fragmented IPv4 packets that are re-assembled to 336,950 CharGen packets. This leads us to believe that an attack relating to the CharGen protocol might be occurring as they make up the bulk of the received data and also the connections. Also there are 77 UDP packets that are do not include any CharGen data. Thus, we have totally 337,027 UDP packets. Additionally, the file contains 9,728 ICMP packets (green slice). The Data (fragmented IPv4 packets), UDP packets and ICMP packets sum up to 1 million.

In the case of Log4j, this was a very specific type of string found in the user agent of the request, followed by some unusual network traffic [16].

Table 2: Basic statistics PCAP file 2

Measurement	Captured size
Average pps	59241.1
Average packet size, B	1203
Bytes	1,203,168,191
Average bytes/s	71 M
Average bits/s	570 M

However these seem to be directed to the victim without any outgoing connections seemingly correlated to them, this would suggest some sort of source IP spoofing, which would give more weight to the idea that this is a DDOS attack based on the CharGen protocol

Fragmentation occurs since the packets exceed the Maximum Transmission Unit (MTU), which is indicated by several packets being of size 1514, which is the maximum Ethernet frame size, and most of them are reassembled properly by the receiver. We also identified a small portion of fragmented CharGen packets that are not reassembled, which we will address as well. Secondly, we observed that the ICMP segments contained in the file are of type 3 and contain the following codes:

- Destination unreachable (Communication administratively filtered)
- Destination unreachable (Port unreachable)
- Destination unreachable (Host unreachable)
- Destination unreachable (Network unreachable)

There are some ICMP packets that contain a single character encapsulated inside CharGen. Lastly, all the packets have only one destination, 227.213.154.245.

3.3 PCAP File 3

A first glance at pcap file 3 shows that 3.4 Million packets have been captured in the span of approximately 354 seconds. Table 3 demonstrates basic statistics of the aforementioned file. Upon initial investigation, it is seen that TCP (1,809,005 packets) and IPv4 (1,156,957 packets) are the two dominant protocols. UDP follows with 477,650 packets and lastly ARP with 502 packets. Almost 1.1 million IPv4 fragmented packets have an overlapping offset. Further analysis by following the TCP conversations in Wireshark reveals that the captured byte length of the TCP packets are different from that mentioned in the TCP header. This in turn leads us to look into TCP-oriented attacks.

Table 3: Basic statistics PCAP file 3

Measurement	Captured size
Average pps	9730.1
Average packet size, B	1100
Bytes	37,899,584,611
Average bytes/s	10 M
Average bits/s	85 M

Additionally, the expert security information revealed a hint of ARP Poisoning. We realized that the analysis of ARP packets could be useful in detecting attacks on a network, as ARP-based attacks can be used to redirect traffic to a malicious host, which can be identified by analyzing ARP packets alongside TCP traffic. Considering the above observations, there is more than one attack happening in the captured PCAP file. Some of the suspected attacks are ARP Poisoning, Teardrop attack, TCP ACK Flood attack, and a small TCP SYN Flood attack. Evidence to support the claims is described in detail in the methodology and results section.

4 METHODOLOGY

In this section, we describe our methodology for performing network forensic analysis in our study. The high-level approach to this research involved four steps (i) data exploration & preliminary analysis of attack vectors (ii) formal analysis of attack vectors evidence collection (iii) evidence logging & interpretation.

More precisely, We start the investigation process with an exploration phase over the datasets to study the data and perform a preliminary analysis of potential attack vectors. Then, a formal analysis follows to examine in-depth potential attacks and formalize the preliminary analysis of the previous phase. During formal analysis, we use Wireshark, Tshark and Python. Wireshark is the de-facto standard for traffic analysis and has a Terminal User Interface (TUI) version along with a Python wrapper. It allows us examine the packets and aggregate the results. Next, we perform evidence collection and data post-analysis using Python and frameworks such as Pandas, Matplotlib, Pyshark, Pickle etc. The results are explained thoroughly in the next section.

5 RESULTS

5.1 PCAP File 1

The results of the analysis provide further evidence to support the initial observations made regarding the suspicious traffic in the captured packets. Figure 3 illustrates that the DNS protocol is the dominant protocol in the first PCAP file existing in 93% of the packets. This serves as a strong indicator of a DNS flood or amplification attack [17]. The ICMP packets are all of type 3 (Destination unreachable). This could indicate the occurrence of an ICMP flood [5]. However, we believe this to be unlikely as the ICMP protocol forms only 5% of the traffic. Furthermore, less frequent protocols such as TCP and IPv6 (column: Other) account for only a small percentage of the traffic. Table 4 in Appendix A.1 provides a comprehensive overview of the remaining identified protocols.

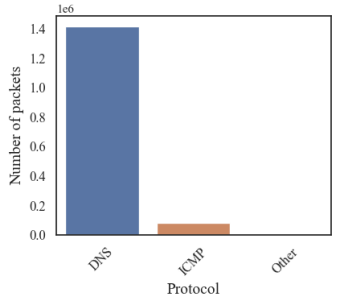

Figure 3: The number of packets per protocol

Figure 4 shows that the DNS packets primarily consist of response packets with the majority of the packets directed towards the "ddostheinter.net" domain. We provide a detailed overview of all the domain names in Appendix A.2. The large number of DNS packets with the single domain "ddostheinter.net" strongly suggests a coordinated DDoS attack which aims to disrupt the targeted system [7].

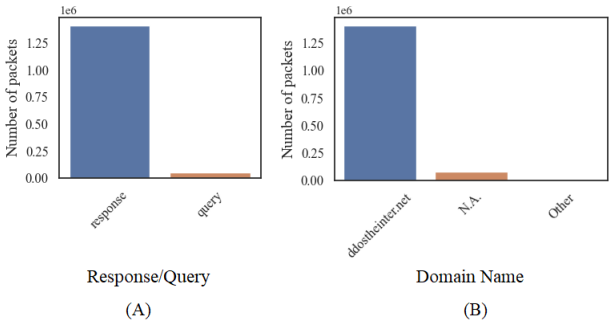

Figure 4: (A) The number of response and query DNS packets
(B) The occurrences of the domain names in the DNS packets

In Figure 5 we observe a packet transmission at a rate of more than 7500 packets per second for a duration of 195 seconds. This further supports the argument for a coordinated DDoS attack. Ethernet source (00:1b:c0:e6:93:3c), Ethernet destination (44:d3:ca:5f:61:40), and IP destination addresses (227.213.154.241) are identical in all packets as illustrated in Figure 6. Moreover, the analysis identified 8208 unique IP source addresses which participate in the DDoS attack [25].

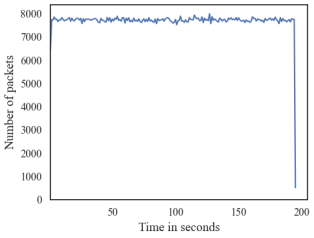

Figure 5: The number of packets per second

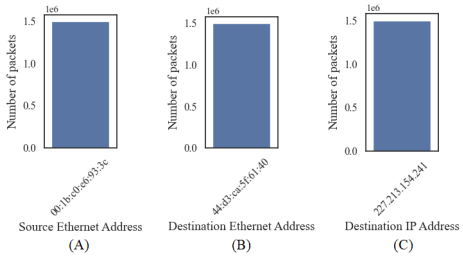

Figure 6: (A) Ethernet Source Address occurrences
(B) Ethernet Destination Address occurrences
(C) IP Destination Address occurrences

5.2 PCAP File 2

Having noted the abnormally large number of CharGen connections from a variety of IP addresses and the large bandwidth consumed by these connections, we can classify this as a DDoS attack using CharGen, or a CharGen flood attack. We calculate the throughput to approximate 74.44 MBps. Thus, the intensity of the attack can generally overwhelm most consumer internet connections which usually range from 20-100 Mbps or 2.5-12.5 MBps. Table 7 demonstrates the attack's intensity over time, which fluctuates on average between 70 and 120 MBps.

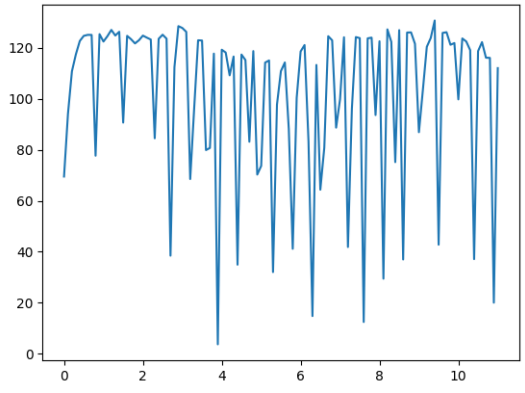

Figure 7: CharGen flood attack intensity

This traffic is received from 274 distinct IP addresses. However, considering that some of these might be legitimate connections to a CharGen service, we introduce a threshold of minimum 200 packets from each host to examine if we suffer an attack by them. We have 254 final IP addresses that send an average number of 3935 packets over 16 seconds and, therefore, we can safely classify this traffic as part of the CharGen attack (Figure 8).

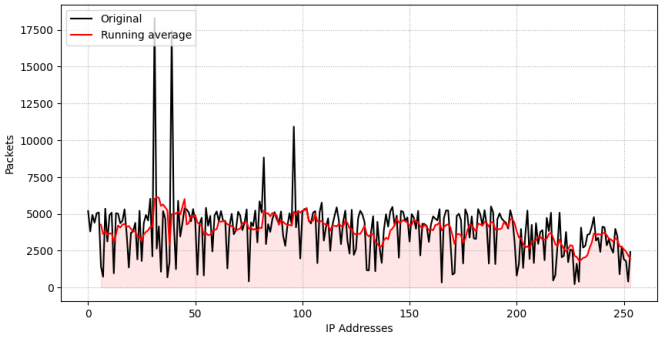

Figure 8: Running average of packets received from each host

5.2.1 ICMP Packets. Apart from fragmented CharGen packets, the PCAP file also contains a number of ICMP packets. In section 3.2, we discovered that all ICMP packets are of type 3 (Destination unreachable) and sent from multiple IP addresses to the single recipient, 27.213.154.245. We do not process outgoing traffic to check

if the recipient actually sent ICMP requests to all the aforementioned hosts. However, literature [6] [15] suggests that this might be an ICMP flood attack attempt spoofing the source address as 27.213.154.245. It is our belief that there is no such attack here. There are 9728 ICMP packets in total which constitute 1% of the total traffic, a low percentage to consider an ICMP flood attack. Instead, it appears that the CharGen flood attackers started pinging and sending data randomly to CharGen servers spoofing the IP source address. Some networks/hosts were unreachable or their port (19) was closed or a router dropped the packet (communication administratively filtered). Hence, these hosts bounced back to the spoofed recipient with ICMP segments. We filtered the traffic to find the intersection of CharGen traffic and ICMP pings source addresses. We got an empty set indicating that ICMP pings are received only by servers that do not send CharGen traffic as well. These servers are either down or reject attackers' packets. As a result, ICMP traffic is considered a side effect to the CharGen attack.

5.2.2 Non-reassembled Fragmented Packets. Lastly, we review the non-reassembled fragmented CharGen packets. Despite the majority of the received fragmented CharGen packets that are reassembled, we identify specific packets that were not reassembled. This could potentially point out to an IPv4 fragmentation attack, where the memory of the router is filled waiting for the remaining fragments to arrive. Then, the receiver either denies legitimate traffic or suffers system crash. We analyzed this traffic and managed to see there is approximately 88 MB worth of fragmented packets created over a span of 16 seconds. lthough this is quite considerable in terms of router RAM size, nonetheless most modern routers have a higher storage capacity and would not suffer from an attack of this size. These types of attacks can be easily mitigated, by having a maximum queue size or timeouts for any fragmented packet waiting for reassembly. We could wait for the queue to fill or some time to pass to drop the earliest or oldest packet.

5.3 PCAP File 3

5.3.1 ARP Poisoning and Man-in-the-Middle-Attack. Figure 9 gives an idea of the types of ARP packets encountered in the pcap file. There are a total of 502 ARP packets out of which 165 are requests and 337 are replies. Further evaluation of these packets reveals that there are 148 fake ARP request packets. Thus, the ratio of legitimate requests to replies is 1:20. This shows that the attacker sends spoofed ARP replies at regular intervals to poison the ARP cache entry of the victim with the attacker's MAC address. As a result, all network traffic sent by the victim is sent to the attacker instead of the intended recipient. Further, two types of fake request packets were identified:

(1) 55 ARP requests only from one source and doesn't receive any reply. The sender with IP Address 128.3.23.117 requests a MAC address of 128.3.23.141.

(2) ARP requests with a target MAC address set to 00:00:00:00:00:00. It is unusual because the target MAC address in an ARP broadcast request should be set to all 1's (ff:ff:ff:ff:ff:ff). This can be an attempt to trick other devices on the network into revealing their MAC addresses. A deeper analysis of the ARP packets confirms that the attacker with IP address 128.3.23.100 and MAC address 7c:d1:c3:94:9e:b8 tries to poison the ARP cache of victims with IP addresses 128.3.23.103 and 123.3.23.1. The attacker here is performing a man-in-the-middle attack. This is shown in Figure 10.

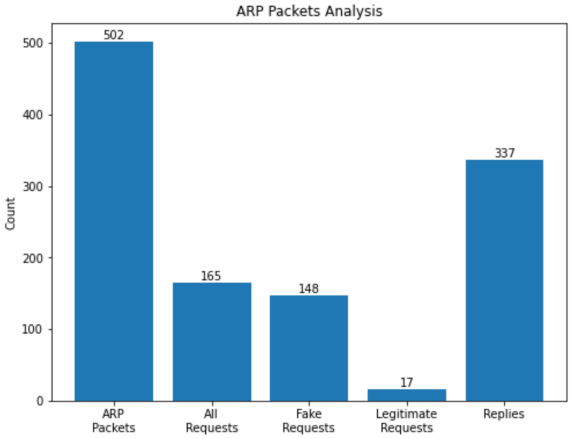

Figure 9: ARP Packets Statistics

In addition, there are 100 ICMP redirect for host packets in the file. Since ICMP redirect messages can only be sent by a router, the attacker here is a malicious router trying to intercept traffic between the two devices on the same network. An ICMP redirect for host packet contains the IP address of the gateway to which the message is being redirected. From source, destination, and gateway addresses in Figure 11, we can notice the traffic being redirected, which provides further evidence of the fact that 128.3.23.100 is the attacker.

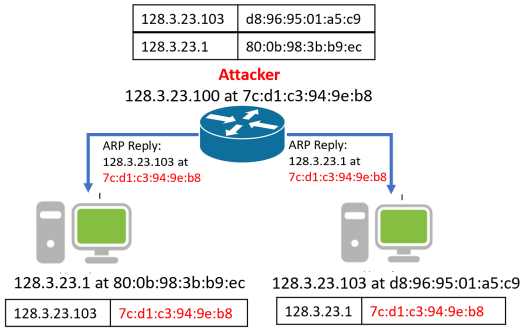

Figure 10: ARP poisoning and Man-in-the-middle

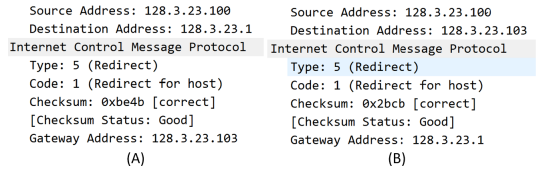

Figure 11: (A) Attacker redirects traffic from 128.3.23.1 to 128.3.23.103 (B) Attacker redirects traffic from 128.3.23.103 to 128.3.23.1

5.3.2 IP Spoofing. Upon closer examination of ARP Packets, it is evident that the attacker IntelCor_c2:5e with MAC address 14:4f:8a:ed:c2:5e sends ARP replies with spoofed IP addresses to the victim LiteonTe_46:5a with MAC address 20:68:9d:41:46:5a. Out of 337 ARP replies in the pcap file, 189 replies (56 percent) are being sent by this attacker with spoofed IPs. The spoofed IP addresses used are shown in Figure 12. We can also notice that the time difference between each packet is almost 10 milliseconds.

Time Diff	Info
0.161013	128.3.23.251 is at 14:4f:8a:ed:c2:5e
0.010352	128.3.23.243 is at 14:4f:8a:ed:c2:5e
0.011156	128.3.23.239 is at 14:4f:8a:ed:c2:5e
0.009273	128.3.23.198 is at 14:4f:8a:ed:c2:5e
0.010384	128.3.23.135 is at 14:4f:8a:ed:c2:5e
0.010176	128.3.23.114 is at 14:4f:8a:ed:c2:5e
0.010274	128.3.23.109 is at 14:4f:8a:ed:c2:5e
0.010319	128.3.23.98 is at 14:4f:8a:ed:c2:5e
0.012101	128.3.23.93 is at 14:4f:8a:ed:c2:5e
0.038614	128.3.23.90 is at 14:4f:8a:ed:c2:5e
0.000002	128.3.23.84 is at 14:4f:8a:ed:c2:5e

Figure 12: Spoofed IP addresses sent by attacker to victim

5.3.3 Teardrop Attack. The Teardrop attack works by exploiting a vulnerability in the way that the target system handles fragmented packets. If the fragments are designed to overlap or have incorrect fragmentation offsets, it can cause the reassembly process to fail, resulting in a system crash or instability. This similar scenario is noticed in the pcap file. The statistics of the communication between A (128.3.70.97) and B(128.3.23.3) show that A sends 1,383,356 packets (85.09%) and B sends 242,452 packets (14.91%). In total, we have 1,625,808 packets. In Figure 13, the first packet starts with an offset of 1490 and is 1494 bytes in length (1514 - 20 = 1494). This means that the last byte of this packet is at offset 1490 + 1494 - 1 = 2983. The second packet starts with an offset of 2960. Since the last byte of the first packet (2983) is greater than the first byte of the second packet (2960), there is an overlap of 24 bytes (2983 - 2960 + 1) between the two packets. This indicates that the packets are not properly aligned and there is an overlapping fragment. Attacker A sends such overlapping fragmented IPv4 packets to B with a ratio of 6:1.

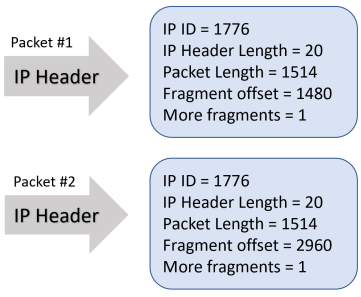

Figure 13: Packets with overlapping offset

Address A	Port A	Address B	Port B	Packets	Bytes	A → B	B → A	Duration
128.55.174.226	65073	128.3.23.123	7503	1,351,409	1.254 GiB	868,503	482,906	285.4692
128.55.22.235	35041	128.3.23.123	7501	173,393	166.395 MiB	112,691	60,702	130.8273
128.55.22.235	34502	128.3.23.123	7501	129,142	124.332 MiB	84,205	44,937	120.3572
128.55.22.235	35527	128.3.23.123	7501	60,512	57.494 MiB	38,905	21,607	95.2124
128.3.23.227	49636	128.3.164.15	143	50,967	45.089 MiB	17,530	33,437	133.1674
128.3.23.120	53409	128.55.248....	22	4,843	876.881 KiB	1,019	3,824	176.7061
128.3.23.245	4769	128.3.70.248	631	2,576	373.416 KiB	1,399	1,177	333.0478
128.55.174.226	65072	128.3.23.123	7502	2,108	213.000 KiB	909	1,199	285.3945
128.3.23.84	37765	128.3.70.248	631	1,991	340.067 KiB	1,004	987	303.7979
128.3.23.14	33454	128.3.70.248	631	1,773	325.615 KiB	897	876	303.0249
128.3.164.135	60568	128.3.23.245	80	1,618	1.499 MiB	542	1,076	0.4448
128.55.248.85	22	128.3.23.120	53563	1,399	211.600 KiB	983	416	295.6432
59.136.13.122	22	128.3.23.231	49957	1,225	339.135 KiB	564	661	150.6922
128.3.23.227	49640	128.3.164.15	143	1,185	359.826 KiB	559	626	133.1647
128.3.23.126	44780	128.3.70.248	631	1,055	163.745 KiB	532	523	323.9661
128.3.164.135	60659	128.3.23.245	80	1,020	951.933 KiB	352	668	3.1761

Figure 14: TCP Conversations with significant traffic

5.3.4 TCP ACK Flood Attack. Concerning the TCP packets' analysis, we first generated a graph of packets sent per second as shown in Figure 15. From this, we can see that an average of 5000 packets/s are sent and sometimes it reaches the peak of more than 10,000 packets/s. Secondly, we looked into the TCP conversation that had the most significant traffic. Figure 14 shows the TCP conversation streams with significant traffic. For analysis, we pick the top 4 conversations. In the first conversation, hosts with IP addresses A: 128.55.174.226 and B: 128.3.23.123 collectively exchange 1.35 million packets with 1.254 Giga Bytes of ongoing traffic in a duration of 285.46 seconds. In the second, third, and fourth conversations, the attacker is always the same with IP address 128.55.22.235 but sends packets with different ports. Two common things can we notices between all four conversations (1) The victim with IP address 128.3.23.123 and (2) The ratio of packets sent by attackers to the victim is 2:1.

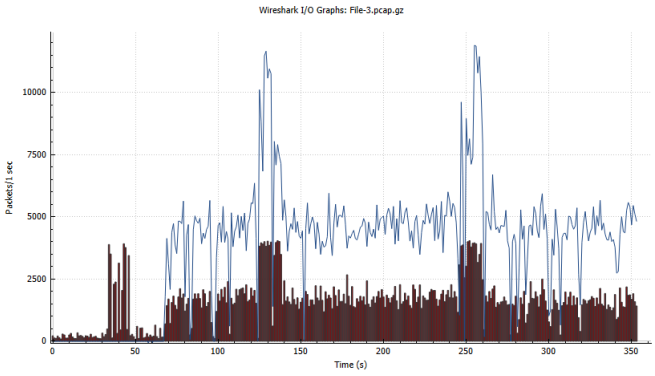

Figure 15: Packets per second

One of the characteristics of an ACK flood attack is that it floods the victim with ACK packets while not actually sending any data. We further inspect the TCP stream graph for window scaling and throughput. All four conversations show the same trend in graphs and are constant as shown in Figure 18 and Figure 19 respectively (subsection A.3). This indicates that no actual data is being transmitted over the network, but only a large number of ACK packets. Additionally, the constant sequence number graph as shown

in Figure 16 indicates that the packets are being generated by an automated tool such as a botnet rather than a legitimate connection. All the above-mentioned results support our ACK flooding speculations and that there is a Distributed Denial of Service attack happening.

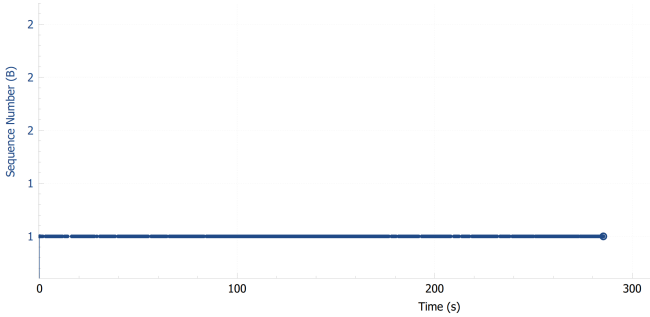

Figure 16: Time / Sequence number TCP Stream Graph

5.3.5 TCP Syn Flood Attack. Finally, there is also a small range of SYN Flooding attack happening in the pcap file. Figure 17 shows the number of SYN packets sent by the attacker without sending ACKs to the SYN/ACKs received, thus, remaining in a half-open state of TCP connection. We notice that most of the IP addresses sending SYN packets are in the subnet range of 128.3.23.0/24, in which ARP poisoning and man-in-the-middle attack was identified.

Attacker	Number of Victims	SYN Packets Count	Different Ports Used
128.3.23.150	12	99	27
128.3.164.229	1	58	58
128.3.23.210	16	33	33
128.3.23.117	3	26	26
128.3.23.227	8	25	25
128.3.23.103	9	25	13
128.3.161.22	4	24	24
128.3.23.5	1	19	3
128.3.23.85	7	17	17
128.3.23.74	7	16	16
128.3.23.49	3	14	13
128.3.23.81	6	13	13
128.3.23.216	2	13	11
128.3.23.158	1	12	4
128.3.23.32	7	11	10
128.3.23.2	1	10	2
128.3.161.252	2	10	10
128.3.164.135	1	7	7
128.3.23.168	1	7	7
128.3.23.239	1	6	6
128.3.23.42	2	4	4
128.55.22.235	1	4	4
128.3.23.98	3	4	4

Figure 17: Statistics of Attackers sending SYN Packets without sending ACKs

6 DISCUSSION

This section discusses mitigation strategies for the suspected attacks, and limitations of the research. In both cases, it is important to note that neither subsection is exhaustive. To adequately present the mitigation strategies, covering the limitations is in order.

6.1 Limitations

While confident in the conclusions, there is so much traffic that it is hard to rule out other potential threats. The sheer amount of data in the captured traffic makes it hard to rule out targeted attacks where the footprint might be smaller compared to that of a D/DoS attack. This would be one explanation for the results as found, as a similar argument can be used for the ARP spoofing attack. It also uses some form of flooding, which is simply easier to observe compared to something like Log4J [16]. This lack of completeness could also be attributed to a lack of time, but a guarantee is hardly possible, regardless of time.

6.2 DNS Flood Attack Mitigation

To mitigate the effects of a DNS flood attack, as seen in the first PCAP file, several strategies can be applied, including filtering, rate controlling, and deploying CAPTCHA.

The first potential strategy considered to mitigate the effects of a DNS flood attack is filtering. This involves blocking incoming traffic from known malicious sources or traffic exhibiting suspicious behavior using techniques such as firewalls and intrusion detection systems [8]. The blocking criteria can be based on various attributes, including IP addresses, port numbers, and application-level protocols [18]. However, relying solely on filtering is not sufficient as attackers can bypass filters by utilizing techniques such as IP spoofing to employ source obfuscation, making filtering an incomplete mitigation strategy on its own [4].

Another effective strategy is rate controlling, which can be achieved through three approaches: rate limiting, capacity enhancing, and load balancing. Rate limiting involves setting a threshold on the amount of incoming traffic to the DNS server to prevent overload. Upon exceeding the threshold, the traffic is either dropped or delayed. Capacity enhancing, on the other hand, involves adding more resources to the DNS server, such as CPUs, bandwidth, and memory, to improve its ability to cope with large amounts of traffic [19]. This approach can be expensive and may require significant changes to the network infrastructure [28]. Load balancing, however, can be a more cost-effective solution [20]. This approach distributes the load across multiple servers, reducing the load on individual servers.

Lastly, deploying CAPTCHA is a strategy that can help distinguish bots from real users [13]. The high variability of IP source addresses in the PCAP file suggests the use of bots during the execution of the performed DNS flood attack. By implementing CAPTCHA, the server can differentiate between legitimate users and bots and reduce the generation of illicit traffic, preserving the server bandwidth solely for legitimate users. However, CAPTCHA can increase the time and effort required for a legitimate user to access the server, which can result in a poor user experience [12]. Moreover, sophisticated methods exist to bypass CAPTCHA, hence making it an incomplete mitigation strategy on its own [10].

In short, filtering, rate controlling, and CAPTCHA can be used in conjunction to mitigate the effects of a DNS flood attack. However, it is important to note that these strategies are not foolproof and may need to be adapted to the specific circumstances of each attack.

6.3 CharGen Flood Attack Mitigation

CharGen attacks can be easily mitigated in one of two ways. First of all, antiquated services that still run CharGen can disable this functionality by making small changes to their configuration or registries on the host operating system. In fact, researchers doubt the installation and use of this protocol on end systems, since it is mostly abused for DDoS attacks [24]. Another method could be to restrict access to port 19 completely or by limiting it to trusted parties only [1]. Further prevention of service resources misuse is to monitor outbound traffic and to make sure that all outgoing traffic uses internal addresses [3].

End-users also need to take actions since we cannot rely on all such servers to monitor and update their systems. Specifically, one mitigation method is to drop all traffic that originates from port 19, either on arrival or as a firewall rule [24]. However, we should also consider the fact that some (legacy) systems might still use the CharGen protocol for debugging etc. In this case, we can use more generic methods to detect DDoS attacks, such as deploying Software Defined Networks (DNS) to catch and identify malicious traffic or flows [27]. Additionally, monitoring the traffic rate from certain flows and limiting offending flows could be a potential mitigation strategy. In this paper, we provide a Python script that monitors an interface for CharGen attacks and alerts the user if it detects too many incoming CharGen streams (10 connections).

6.4 ARP Spoofing Mitigation

To mitigate ARP spoofing attacks, there are several measures that can be implemented. One approach is to use static ARP entries on critical devices, which prevents unauthorized changes to the ARP cache by hardwiring MAC addresses to specific IP addresses. However, this solution may not be scalable for larger networks [14]. Another measure is to use VLANs to isolate different network segments and restrict the broadcast domain, making it more challenging for attackers to launch ARP spoofing attacks across the entire network. Implementing port security features on network switches can also help by limiting the number of MAC addresses that can be associated with a single port. Additionally, ARP spoofing detection tools or network security monitoring solutions can be used to detect and alert on any suspicious ARP activities. Finally, educating network users about ARP spoofing and promoting good security practices, such as avoiding clicking on suspicious links or downloading unknown files, can also contribute to preventing ARP spoofing attacks. Overall, a combination of technical measures and user awareness can effectively mitigate ARP spoofing attacks and enhance network security.

6.5 SYN Flooding Mitigation

SYN flood attacks may be reduced using a few different methods. Using rate-limiting on incoming traffic to limit the server's ability to process a set number of connection requests per second is one typical strategy. Using firewalls or intrusion detection systems (IDS) that can identify and stop traffic from unknown or known malicious IP addresses is an additional tactic. SYN cookies, a security feature of the TCP/IP protocol, can be used to distinguish between valid and invalid connection requests, assisting servers in thwarting SYN flood assaults. SYN flood attacks can also be avoided by proper network architecture and setup, as well as by applying security updates to software and systems.

6.6 TearDrop Attack Mitigation

Given the fact that around a third of organizations run on outdated OS, there are a couple ways of mitigating TearDrop attacks: the first would be using a firewall at network layer level in order to protect against Denial of Service caused by teardrop attacks [26]. Another solution would be to implement packet filtering at the network perimeter to detect and discard any malformed packets before they reach the targeted system. Finally, caching servers provides protection against DDos attacks since ensures preservation of continuously available data even if the network encounters failures. Additionally, proxy servers make it simple to keep an eye on incoming traffic and identify data fragments as soon as they appear [26].

7 CONCLUSION

All members of the team worked in teams of two to analyse the PCAP files, using visual tools, and some scripts. We used the insights obtained from these tools and related literature to understand and classify the attacks. All members then presented their findings to each other as a group to ensure that we could collaborate to understand and contribute to each others work collectively.

Overall, the results provide compelling evidence that the traffic captured in the first PCAP file is indicative of a DDoS attack, with the DNS protocol being the primary method of attack. The high flow rate and identical source and destination addresses suggest the use of a botnet or other means of IP address spoofing.

Looking at PCAP-2, the large amount of incoming CharGen packets from a large number of unique IP addresses suggests a distributed CharGen Flood attack. The ICMP type 3 packets received by the victim lend credence to the idea of an attacker spoofing the victims address and redirecting the attack.

In PCAP-3 we see indicators towards a multitude of attacks, there are a large number of ARP reply packets found in the PCAP, which on further investigation turned out to be fake, which would suggest an ARP poisoning leading into a MITM attack, we also see evidence of IP spoofing by examining the ARP packets sent by the attackers MAC address.

Some of the packets sent were also malformed to have overlapping fragmentation offsets, which could indicate a teardrop attack. The large amount of TCP ACK sent to a particular IP leads us to believe that a TCP ACK flood attack is occurring as well.

7.1 Future Work

As mentioned in the Background section, machine learning models are gaining popularity for identifying DDoS attacks as well as other cyber threats. An interesting question would be to check the conclusions with some machine learning-based tools. These might be able to identify more attacks than we have within our limited time-frame.

REFERENCES

[1] 2021. Internet Accessible CHARGEN Service. (2021). https://www.ncsc.gov.ie/emailsfrom/Shadowserver/DoS/Chargen/

[2] 2021. NVD - CVE-2021-44228. (2021). https://nvd.nist.gov/vuln/detail/CVE-2021-44228
[3] John Bambenek. 2013. A Chargen-based DDoS? Chargen is still a thing? https://isc.sans.edu/diary/A+Chargenbased+DDoS+Chargen+is+still+a+thing/15647. (2013).
[4] Yuan Cao, Yuan Gao, Rongjun Tan, Qingbang Han, and Zhuotao Liu. 2018. Understanding internet DDoS mitigation from academic and industrial perspectives. *IEEE Access* 6 (2018), 66641–66648.
[5] Rocky KC Chang. 2002. Defending against flooding-based distributed denial-of-service attacks: A tutorial. *IEEE communications magazine* 40, 10 (2002), 42–51.
[6] Varun Chauhan and Pranav Saini. 2018. ICMP Flood Attacks: A Vulnerability Analysis. In *Cyber Security*, M. U. Bokhari, Namrata Agrawal, and Dharmendra Saini (Eds.). Springer Singapore, Singapore, 261–268.
[7] Yu Chen and Kai Hwang. 2006. Collaborative change detection of DDoS attacks on community and ISP networks. In *International Symposium on Collaborative Technologies and Systems (CTS'06)*. IEEE, 401–410.
[8] Ji-Ho Cho, Ji-Yong Shin, Han Lee, Jeong-Min Kim, and Geuk Lee. 2015. Ddos prevention system using multi-filtering method. In *International Conference on Chemical, Material and Food Engineering*. Atlantis Press, 774–778.
[9] Giuseppe Desolda, Lauren S. Ferro, Andrea Marrella, Tiziana Catarci, and Maria Francesca Costabile. 2021. Human Factors in Phishing Attacks: A Systematic Literature Review. 54, 8 (2021), 173:1–173:35. https://doi.org/10.1145/3469886
[10] Peter Djalaliev, Muhammad Jamshed, Nicholas Farnan, and José Brustoloni. 2008. Sentinel: hardware-accelerated mitigation of bot-based DDoS attacks. In *2008 Proceedings of 17th International Conference on Computer Communications and Networks*. IEEE, 1–8.
[11] Claude Fachkha, Elias Bou-Harb, and Mourad Debbabi. Fingerprinting Internet DNS Amplification DDoS Activities. In *2014 6th International Conference on New Technologies, Mobility and Security (NTMS)* (2014-03). 1–5. https://doi.org/10.1109/NTMS.2014.6814019 ISSN: 2157-4960.
[12] Ruti Gafni and Idan Nagar. 2016. CAPTCHA: Impact on user experience of users with learning disabilities. *Interdisciplinary Journal of e-Skills and Lifelong Learning* 12 (2016), 207–223.
[13] Shivangi Garg and RM Sharma. 2017. Anatomy of botnet on application layer: Mechanism and mitigation. In *2017 2nd International Conference for Convergence in Technology (I2CT)*. IEEE, 1024–1029.
[14] Robert Grimmick. 2022. ARP poisoning: What it is amp; how to prevent ARP spoofing attacks. (Aug 2022). https://www.varonis.com/blog/arp-poisoning
[15] Neha Gupta, Ankur Jain, Pranav Saini, and Vaibhav Gupta. 2016. DDoS attack algorithm using ICMP flood. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*. 4082–4084.
[16] Raphael Hiesgen, Marcin Nawrocki, Thomas C. Schmidt, and Matthias Wählisch. 2022. The Race to the Vulnerable: Measuring the Log4j Shell Incident. (2022). https://doi.org/10.48550/arXiv.2205.02544 arXiv:2205.02544 [cs]
[17] Adam Ali Zare Hudaib and EAZ Hudaib. 2014. DNS advanced attacks and analysis. *International Journal of Computer Science and Security (IJCSS)* 8, 2 (2014), 63.
[18] Kübra Kalkan, Gürkan Gür, and Fatih Alagöz. 2016. Filtering-based defense mechanisms against DDoS attacks: A survey. *IEEE Systems Journal* 11, 4 (2016), 2761–2773.
[19] Aapo Kalliola, Kiryong Lee, Heejo Lee, and Tuomas Aura. 2015. Flooding DDoS mitigation and traffic management with software defined networking. In *2015 IEEE 4th International Conference on Cloud Networking (CloudNet)*. IEEE, 248–254.
[20] Sarp Köksal, Yaser Dalveren, Bamoye Maiga, and Ali Kara. 2021. Distributed denial-of-service attack mitigation in network functions virtualization-based 5G networks using management and orchestration. *International Journal of Communication Systems* 34, 9 (2021), e4825.
[21] Meenakshi Mittal, Krishan Kumar, and Sunny Behal. 2022. Deep learning approaches for detecting DDoS attacks: a systematic review. (2022). https://doi.org/10.1007/s00500-021-06608-1
[22] Fabio Pasqualetti, Ruggero Carli, Antonio Bicchi, and Francesco Bullo. Identifying cyber attacks via local model information. In *49th IEEE Conference on Decision and Control (CDC)* (2010-12). 5961–5966. https://doi.org/10.1109/CDC.2010.5717914 ISSN: 0191-2216.
[23] Paolo Passeri. 2022. 2022 Cyber Attacks Statistics. (2022). https://www.hackmageddon.com/2023/01/24/2022-cyber-attacks-statistics/ Section: Cyber Attacks Statistics.
[24] José Jair Santanna, Roland van Rijswijk-Deij, Rick Hofstede, Anna Sperotto, Mark Wierbosch, Lisandro Zambenedetti Granville, and Aiko Pras. 2015. Booters An analysis of DDoS-as-a-service attacks. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. 243–251. https://doi.org/10.1109/INM.2015.7140298
[25] Krushang Sonar and Hardik Upadhyay. 2014. A survey: DDOS attack on Internet of Things. *International Journal of Engineering Research and Development* 10, 11 (2014), 58–63.
[26] Wallarm. 2023. What is a Teardrop Attack? Definition, Examples, Prevention. https://www.wallarm.com/what/teardrop-attack-what-is-it. (February 2023).
[27] Jin Wang and Liping Wang. 2022. SDN-Defend: A Lightweight Online Attack Detection and Mitigation System for DDoS Attacks in SDN. *Sensors* 22, 21 (2022). https://doi.org/10.3390/s22218287
[28] Alex Zemlianov and Gustavo De Veciana. 2005. Capacity of ad hoc wireless networks with infrastructure support. *IEEE Journal on selected areas in Communications* 23, 3 (2005), 657–667.

A PCAP 1

A.1 Overview Protocols

Table 4: Overview Protocols PCAP 1

NBNS	Chargen	XTACACS	XYPLEX	WTLS+WSP
TCP	TIME	SRVLOC	CIP I/O	CLDAP
ECHO	DAYTIME	MobileIP	BAT_VIS	IPv6
QUAKE3	KINK	NTP	LWARP	H.248

A.2 Overview Domain Names

- sdostheinter\\003net\\000\\000\\001\\000\\001\\000\\000\\002\\000\\001\\000
- hak4umz.net
- ddostheinter.net,ddostheinter.net,<Root>, >t\357\277\275\\022,<Root>
- \357\277\275dostheinter\\003net\\000\\000\\001\\000\\001\\275\\\000\\001\\000\\001

A.3 Throughput, Window Scaling for PCAP3

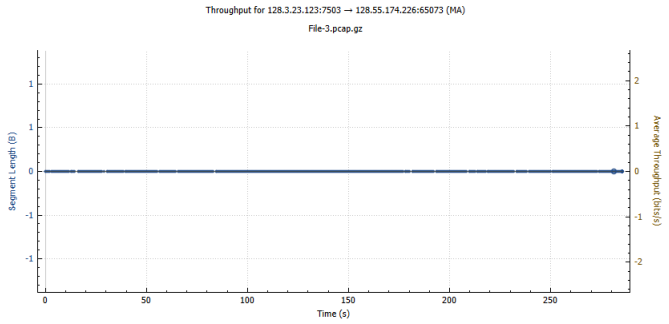

Figure 18: Throughput TCP Stream Graph

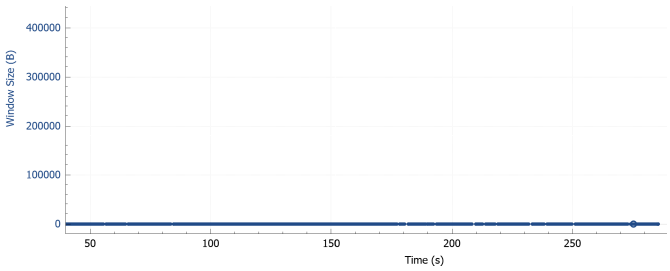

Figure 19: Window Scaling TCP Stream Graph