Amplified Reflection Attacks Over TCP Exploiting Middleboxes

*Presented at the Workshop on Communication Networks and Power Systems (WCNPS'25)

Paulo Victor França de Souza
Department of Computer Science
University of Brasília
Brasília, Brazil
paulovictorfs@outlook.com

João J. C. Gondim
Department of Electrical Engineering
University of Brasília
Brasília, Brazil
gondim@unb.br

***Abstract*—Amplified reflection attacks over TCP exploiting middleboxes, first described in 2021 and already observed in real incidents, represent an emerging threat to cybersecurity. This technique abuses intermediate devices, such as firewalls and proxies, to amplify traffic in distributed denial-of-service (DDoS) campaigns. Laboratory experiments were conducted with pfSense and FortiGate firewalls, evaluating the technique's effectiveness and the devices' behavior under attack. Additionally, a scan of the Brazilian IP space was performed with the ZMap tool and custom TCP packets, identifying vulnerable middleboxes.**

***Index Terms*—amplified reflection attacks, middleboxes, DoS, DDoS, denial of service, networks, information security.**

I. Introduction

In recent years, cybersecurity professionals involved in mitigating and responding to denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks have faced increasingly complex challenges. These challenges arise from the rapid evolution of sophisticated attack techniques, including new forms of reflection and amplification-based DDoS [1]. While such attacks are traditionally associated with the UDP protocol [2], recent studies show they can also be executed via TCP. One form of exploitation of this protocol was demonstrated by Bock et al. [3], who showed that TCP reflection and amplification attacks can occur by exploiting improper behavior of middleboxes, intermediate devices that inspect, filter, and modify data packets [3] [4].

Middleboxes, such as firewalls, proxies, and intrusion prevention systems (IPS) [5], are widely used in corporate networks. However, when misconfigured, they can become attack vectors. In these cases, an attacker sends TCP packets with a spoofed source IP address, corresponding to the victim. The middlebox, when processing these packets, generates amplified responses directly to the target, hiding the true origin of the attack. Even advanced security solutions, such as Next Generation Firewalls (NGFWs), may have vulnerabilities in some versions, allowing these devices to act as amplifiers. The impact of this type of exploitation can be severe, resulting in significant financial losses and damage to the organization's reputation.

Several attacks exploiting middleboxes have already been documented. In March 2022, Akamai identified DDoS campaigns with peaks of up to 11 Gbps, confirming the use of the middlebox reflection technique [4]. Although still on a smaller scale than other DDoS vectors, the frequency and intensity of these attacks are growing, indicating that the technique is becoming a common resource in attackers' arsenals [4].

The objective of this work is to demonstrate and analyze, through practical experiments, how amplified reflection attacks over TCP can be exploited in middleboxes. First, an attack algorithm was applied and evaluated in three distinct controlled environments to analyze the behavior of the middleboxes. Next, the study was expanded through a large-scale scan of the Brazilian IP space to identify the vulnerability of middleboxes in real networks in the country and assess their proportion in a broader scenario.

II. Amplified Reflection Attack Over TCP Exploiting Middleboxes

An amplified reflection attack over TCP using middleboxes occurs when intermediate devices are manipulated to generate large volumes of traffic against a victim. This exploits the incorrect implementation of the TCP protocol, which allows middleboxes to respond to out-of-state packets without a complete handshake [6] [7]. This attack has a significant amplification rate, as per Equation 1.

$$\text{Amplification Rate} = \frac{\text{Data received by the victim}}{\text{Data sent by the attacker}} \quad (1)$$

In this type of attack, the attacker sends forged TCP packets, inserting the victim's IP address as the source. These packets contain HTTP requests directed to domains considered prohibited, such as pornography, weapons, or other blocked content sites. The middlebox, upon inspection, interprets the traffic as legitimate and applies its blocking or filtering rules, generating automatic responses that are forwarded directly to the victim's address, including responses such as block pages (with HTML content, images, or scripts), HTTP headers, or

TCP packets with extra data. As these responses can be larger than the initial request, an amplification effect occurs, resulting in a high volume of malicious traffic directed at the target [4] [3].

The attack was first described by Kevin Bock and his research team at the University of Maryland and the University of Colorado Boulder, in the scientific article "Weaponizing Middleboxes for TCP Reflected Amplification" [3]. In the study, the authors performed a scan of the entire IPv4 internet to identify vulnerable middleboxes, finding that many offer amplification factors greater than 100 times. Furthermore, they observed that this type of attack occurs mainly in environments associated with censorship regimes, where such devices are widely used for traffic inspection and blocking.

Defense against this type of attack is made difficult by the use of common ports (such as TCP 80) and the fact that the packets contain valid HTTP requests, which makes signature-based detection unfeasible. Therefore, mitigation requires modifications to the middleboxes' default behavior, as well as firmware updates to reduce the exploitation surface [6].

III. Mitigation Strategies for Amplified Reflection Attacks Over TCP Exploiting Middleboxes

Amplified reflection attacks over TCP that exploit middleboxes represent a complex problem affecting various network implementations and devices, with no single solution applicable to all contexts [3]. One of the main strategies is to require the middlebox to observe both directions of communication and validate the complete establishment of the TCP handshake before injecting any response, preventing responses to forged packets [3].

SYN packets with a payload are indicative of malicious traffic, as they rarely contain data in legitimate contexts [4]. It is recommended to block these packets and discard connections with payloads coming from ports 80 or 443. An example of a practical ACL rule:

```
deny tcp any eq 80 host x.x.x.x match-all +SYN -ack
    packet-length gt 100
```

Another approach is to limit the size of responses issued by middleboxes, using simple RST packets, minimal HTTP redirects, or simple block pages, reducing the amplification factor [3] [8]. The simplification of block pages and the restriction of responses to internal traffic and specific regions are also recommended.

It is also recommended to disable unnecessary HTTP responses, prioritizing secure HTTPS, and send RST packets when the source IP does not respond, ending indefinite sessions [9]. The TCP-middlebox reflection technique exploits SYN packets with a payload and requests on multiple ports, making traditional scans ineffective [10]. This reinforces the need for proactive and integrated mitigation strategies that combine complete handshake validation, response limitation, and simple responses to suspicious requests.

IV. Methodology for Code and Laboratory Environment Usage

The code used for the attacks was obtained from a public GitHub repository, authored by the user `moloch54`. The project, titled "Ddos-TCP-Middlebox-Reflection-Attack" [11], includes the `mra.py` file, responsible for executing the attack. After a detailed investigation, it was found that the author of the code is Sébastien Meniere, residing in Nancy, in the Grand Est region, France, according to information available on his LinkedIn profile. In direct contact with the author, permission was obtained to use the code, in addition to the confirmation that it had not yet been tested in a laboratory environment, highlighting the importance of the experiments conducted in this study.

This public code was chosen due to its clear structure and approval of the Bock et al. methodology. [3], which ensured compliance with the attack and greater accuracy in the results. Using an already established tool allowed us to focus on experimental analysis, avoiding development from scratch and reducing the risk of inconsistencies.

The laboratory environment consisted of three distinct scenarios, all based on the same topology illustrated in Figure 1 and using the same IP addresses. Each scenario had three main elements: a target, an attacker, and a middlebox, represented by a firewall. In the first scenario, the firewall was implemented with pfSense and the pfBlockerNG add-on. In the second, the combination used was pfSense with the Squid and SquidGuard software. Finally, in the third scenario, the firewall used was FortiGate.

It should be noted that the firewalls were intentionally configured to simulate both typical corporate environment behaviors and incorrect configurations, with the aim of showing how certain configuration choices can make these devices vulnerable to amplified reflection attacks over TCP, even though they are widely used solutions in companies.

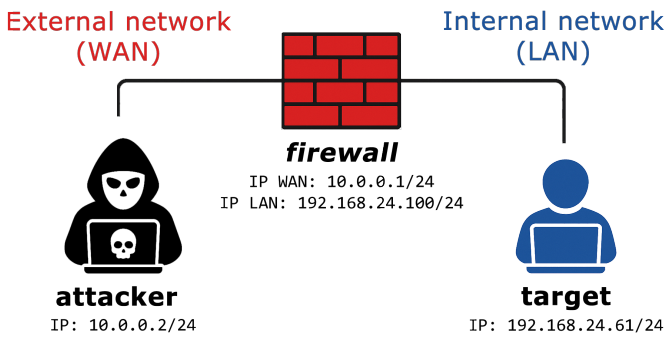

Fig. 1: Laboratory network topology used in the three experimental attack scenarios.

Furthermore, the complete configuration of the experimental environment used in the analysis of amplified reflection attacks over TCP, including source code documentation, virtual elements, network topology, and laboratory configurations, is detailed in a public repository available at: https://github.com/paulovictorbrines/TCP_Middlebox_Reflection_Attack_Lab.

A. Results of the pfSense + pfBlockerNG Firewall

In the tests performed with pfSense in conjunction with pfBlockerNG, a tool responsible for blocking based on domains and IP lists, displaying a block page to the user, it was found that the system only redirects requests destined for prohibited domains to the local block page of pfBlockerNG itself. This behavior is characterized only as reflection, since no external HTML content is sent. Thus, the resulting amplification rate is equal to 1.00x, as defined in Equation 1. The record of this reflection was identified in the pfSense logs, as illustrated in Figure 3.

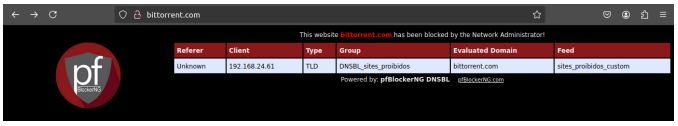

Fig. 2: Block page of the pfBlockerNG package in pfSense.

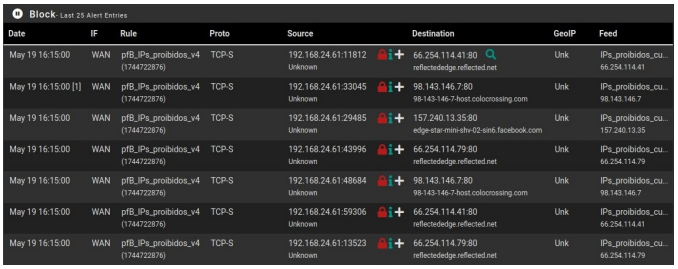

Fig. 3: pfBlockerNG logs in pfSense, showing the successful blocking of domains classified as prohibited during the attack execution.

B. Results of the pfSense + Squid + SquidGuard Firewall

As an alternative to the first scenario (IV-A) and with the aim of evaluating the possibility of traffic amplification, the combination of the pfSense firewall with Squid (proxy) and SquidGuard (content filter) was used. Unlike pfBlockerNG, which only redirects prohibited domains, SquidGuard effectively sends an HTML block page to the user, as shown in Figure 4. This difference motivated the choice of this configuration for the second experimental scenario.

Fig. 4: Block page of the SquidGuard package in pfSense.

During manual access tests to prohibited domains via a browser, the blocking was correctly applied. However, in attack scenarios with forged packets, the traffic is released without any interception or response. This failure occurs because, apparently, SquidGuard does not inspect packets that are not part of the TCP handshake process, thus allowing traffic to pass through.

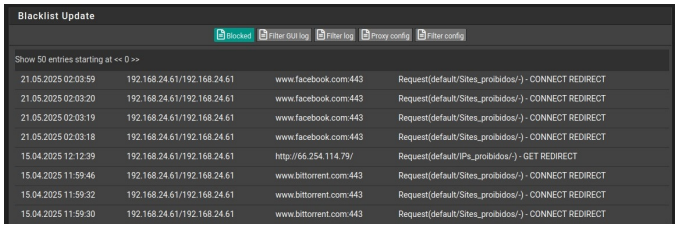

Fig. 5: SquidGuard logs, integrated into Squid in pfSense, showing the successful blocking of domains classified as prohibited during manual access via a browser.

C. Results of the FortiGate Firewall

To overcome the limitations observed in the first two scenarios (IV-A and IV-B), the FortiGate firewall was used. This choice is justified by it being a more robust alternative and the version used (7.2.0) being vulnerable to the attack according to CVE-2022-27491 [12]–[14]. Unlike pfBlockerNG and similar to Squid in conjunction with SquidGuard, FortiGate sends an HTML block page to the user (Figure 6) and is widely used in corporate and government institutions.

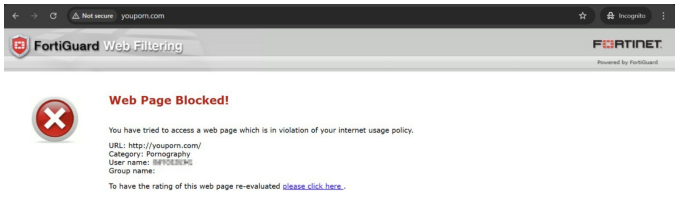

Fig. 6: FortiGate block page.

The observed behavior was similar to the first scenario (IV-A), generating reflection, but without amplification, as evidenced in the logs in Figure 7, resulting in an amplification rate of 0.68x, according to Equation 1. FortiGate has a more rigorous behavior in verifying the TCP state and requires the complete establishment of the connection (three-way handshake) to display the block page, which reduces the number of reflected packets due to the rigorous verification of the TCP state.

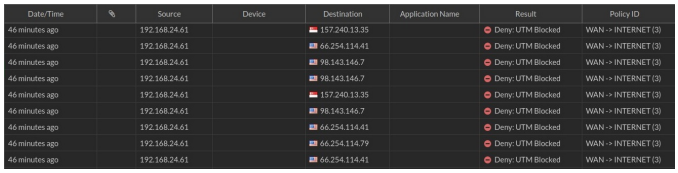

Fig. 7: FortiGate logs showing the successful blocking of domains classified as prohibited during the attack execution.

D. Comparison of the Three Firewall Results

In Figure 8, the sending of forged packets by the attacker in scenarios IV-A and IV-C is observed, with spoofed source IP addresses (corresponding to the victim) and containing `SYN` flags, simulating legitimate communication with services potentially blocked by middleboxes. These packets trigger the middleboxes to generate reflected responses.

Figure 9 presents the direct effect of this manipulation: packets sent by the middleboxes in response to the forged requests, received by the victim's host. These packets constitute the reflected traffic, demonstrating the practical exploitation of middleboxes as attack vectors.

Table I summarizes the results, indicating the volume of packets sent by the attacker, the volume of reflected packets and received by the target, the percentage received in relation to the total sent, and the amplification rate for each middlebox tested. The results of Squid with SquidGuard were excluded due to the absence of representative data: although the blocking via browser occurred, the traffic generated by the attack code was allowed, showing inconsistent proxy behavior in the face of different types of requests.

In the other scenarios, pfSense with pfBlockerNG reflected almost all packets (almost 100%), with a unitary amplification (1.00x). FortiGate presented a reflection rate of approximately 67.7%, resulting in an amplification of less than 1 (0.68x). These results demonstrate that, even without significant content injection, that is, without amplification, there was middlebox reflection to the target, confirming the effectiveness of the technique.

No.	Time	Source	Destination	Protocol	Length	Info
17	0.001815	192.168.24.61	66.254.114.41	TCP	54	30544 → 80 [SYN] Seq=0 Win=8192 Len=0
18	0.001905	192.168.24.61	98.143.146.7	TCP	54	45783 → 80 [SYN] Seq=0 Win=8192 Len=0
19	0.002032	192.168.24.61	66.254.114.79	TCP	54	8618 → 80 [SYN] Seq=0 Win=8192 Len=0
20	0.002120	192.168.24.61	157.240.13.35	TCP	54	42221 → 80 [SYN] Seq=0 Win=8192 Len=0
21	0.002247	192.168.24.61	157.240.13.35	TCP	54	24639 → 80 [SYN] Seq=0 Win=8192 Len=0
22	0.002336	192.168.24.61	98.143.146.7	TCP	54	38563 → 80 [SYN] Seq=0 Win=8192 Len=0
23	0.002474	192.168.24.61	157.240.13.35	TCP	54	5355 → 80 [SYN] Seq=0 Win=8192 Len=0
24	0.002562	192.168.24.61	98.143.146.7	TCP	54	42691 → 80 [SYN] Seq=0 Win=8192 Len=0
25	0.002673	192.168.24.61	98.143.146.7	TCP	54	12898 → 80 [SYN] Seq=0 Win=8192 Len=0
26	0.002763	192.168.24.61	66.254.114.41	TCP	54	45431 → 80 [SYN] Seq=0 Win=8192 Len=0
27	0.002892	192.168.24.61	66.254.114.79	TCP	54	17521 → 80 [SYN] Seq=0 Win=8192 Len=0
28	0.002980	192.168.24.61	66.254.114.41	TCP	54	31791 → 80 [SYN] Seq=0 Win=8192 Len=0

Fig. 8: Excerpt from the `attacker.pcap` packet capture in Wireshark, showing the forged packets sent by the attacker with the objective of initiating the TCP reflection attack.

No.	Time	Source	Destination	Protocol	Length	Info
17	9.890023	157.240.13.35	192.168.24.61	TCP	60	80 → 5014 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
18	9.890189	157.240.13.35	192.168.24.61	TCP	60	80 → 50629 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
19	9.890749	66.254.114.79	192.168.24.61	TCP	60	80 → 8618 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
20	9.890983	98.143.146.7	192.168.24.61	TCP	60	80 → 49538 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
21	9.891132	66.254.114.41	192.168.24.61	TCP	60	80 → 46733 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
22	9.891284	98.143.146.7	192.168.24.61	TCP	60	80 → 3933 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
23	9.891425	98.143.146.7	192.168.24.61	TCP	60	80 → 25510 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
24	9.891568	66.254.114.41	192.168.24.61	TCP	60	80 → 30544 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
25	9.891722	98.143.146.7	192.168.24.61	TCP	60	80 → 45783 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
26	9.891897	157.240.13.35	192.168.24.61	TCP	60	80 → 42221 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
27	9.892956	157.240.13.35	192.168.24.61	TCP	60	80 → 24639 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
28	9.892957	157.240.13.35	192.168.24.61	TCP	60	80 → 5355 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

Fig. 9: Excerpt from the `target.pcap` packet capture in Wireshark, showing the responses sent by the firewall to the target host as a result of the TCP reflection attack during the attack

V. Methodology of the Scan in the Brazilian IPv4 Space

This study used a methodology inspired by tests conducted by organizations such as ShadowServer [10] (https://www.shadowserver.org/news/over-18-8-million-ips-vulnerable-to-middlebox-tcp-reflection-ddos-attacks) and Akamai [4] (https://www.akamai.com/blog/security/tcp-middlebox-reflection). Additionally, it sought to validate and deepen the results presented by Bock et al. [3].

The main objective was to scan the Brazilian IPv4 address space to identify vulnerable middleboxes capable of reflecting HTTP responses from malformed SYN packets. The motivation was to assess the presence and distribution of vulnerabilities, as well as the potential for exploitation in Brazilian territory, considering expected responses such as SYN-ACK, RST with payloads, or block pages, among others.

A. Scanning Tools and Execution

The scan was performed using a customized version of ZMap, a fast and modular network scanner designed for large-scale research. This version incorporated the `forbidden_scan` module, developed by Bock et al. [3], which is not part of the official ZMap. The module sends TCP SYN packets with an HTTP payload, an atypical pattern that aims to provoke responses from middleboxes that inspect packets before the completion of the TCP handshake. The module's code is available at: https://github.com/Kkevsterr/zmap/blob/master/src/probe_modules/module_forbidden_scan.c.

The list of Brazilian IPv4 addresses was obtained in CIDR format (ex.: 138.97.188.0/22) from the Country IP Blocks website (https://www.countryipblocks.net/acl.php). The scan was executed with the following command, after cloning and compiling the modified ZMap repository:

```
sudo src/zmap -i ens37 -M forbidden_scan -p 80 \
-w "blocos_ipv4_brasil.txt" \
-f "saddr,len,payloadlen,flags,validation_type" \
-o "scan/scan_brasil.csv" -O csv
```

The command scanned port 80, recording in the `scan_brasil.csv` file the source address (`saddr`), the payload length (`payloadlen`) and the TCP flags (`flags`) of the responses, essential information for analyzing traffic reflection and amplification.

B. Statistical Analysis and Graph Generation

The raw scan data, stored in CSV format (`scan_brasil.csv`), was processed by a Python script called `stats.py` (https://github.com/breakerspace/weaponizing-censors). The script analyzed the received responses, extracting relevant statistics to evaluate the potential for reflection and amplification of the middleboxes.

The script was executed with the command:

```
sudo python3 stats.py zmap/scan/scan_brasil.csv 149
```

The `149` parameter corresponds to the size of the probe packet sent. The script calculated metrics such as the number of IPs that generated amplification, the average amplification rate observed, and the distribution of TCP flags in the responses.

VI. Scan Results

To circumvent restrictions imposed by internet providers, the initial scan in the Brazilian IPv4 space was performed through a VPN with an exit in the USA. This approach made it possible to identify $13,299,793$ unique IPs that responded to the requests, from a total of $40,805,661$ blocks. The scan, conducted with ZMap and the `forbidden_scan` module, lasted approximately 2 hours and 26 minutes and was

TABLE I: Comparison of results obtained during five minutes of attack on the pfSense + pfBlockerNG and FortiGate middleboxes

Middlebox	Packets sent	Packets received	% received	Amplification
pfSense + pfBlockerNG	2,044,369	2,044,368	99.99%	1.00x
FortiGate	2,044,328	1,384,473	67.73%	0.68x

successfully completed. Of these responding IPs, $4,541,741$ (approximately 34%) were classified as amplifiers, indicating vulnerability to amplified reflection attacks via TCP (Table II).

The high number of amplifier IPs does not correspond to the actual number of distinct middleboxes, as many IPs respond from behind a few middleboxes. This concentration confirms the presence of vulnerable middleboxes distributed throughout the Brazilian space. Figures 11 and 12 illustrate, respectively, the quantity and proportion of IPs classified as amplifiers. Figure 10 presents the distribution of TCP flags in the response packets, providing information about the behavior of the respondents' network systems.

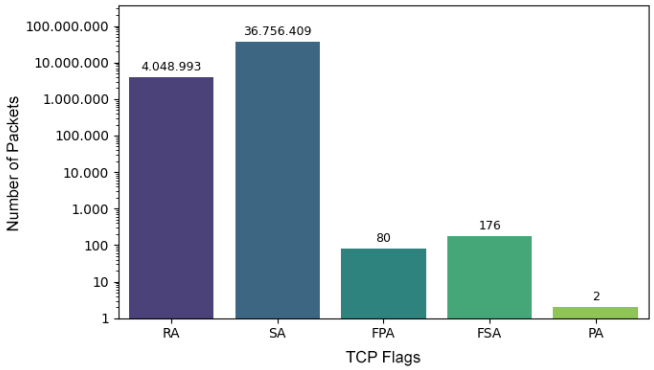

Fig. 10: Distribution of TCP flags in the response packets emitted by the IPs.

In the detailed analysis, the predominance of the **SYN-ACK** flag ($36,756,409$ packets) shows a behavior that favors amplification attacks, even without a complete handshake establishment. The behavior of the amplifiers was analyzed through Cumulative Distribution Functions (CDFs), presented in Figures 13, 14 and 15, which show, respectively, the distribution of packets, bytes, and amplification rates per IP. These results demonstrate that TCP amplification is feasible on a large scale, even in scenarios without a complete connection.

VII. Conclusions

This work demonstrated that the amplified reflection attack via middleboxes constitutes a real and viable threat, demanding attention from professionals and researchers. To this end, a detailed study of the topic was carried out, involving the development of a laboratory environment for controlled tests and a large-scale scan of the Brazilian IPv4 space.

In the laboratory environment, even without large volumes of amplification, the reflection of packets to the target proved the viability of the attack in a controlled scenario. This was the first recorded successful application of the test code in a real environment, validating the attack concept.

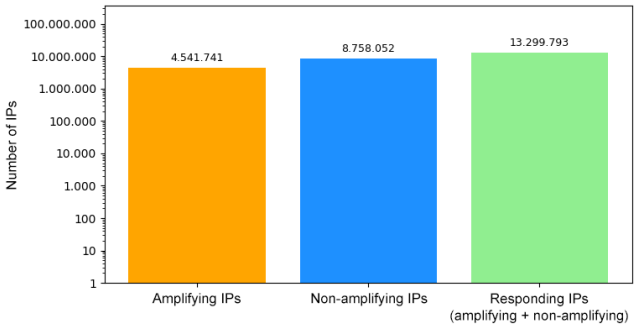

Fig. 11: Quantity of IP addresses that responded to the scan, categorized as amplifiers and non-amplifiers, including the total number of respondents.

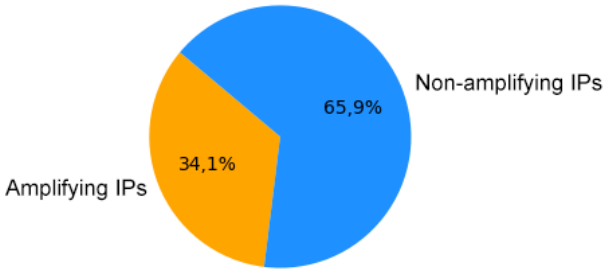

Fig. 12: Percentage of responding IPs classified as amplifiers and non-amplifiers.

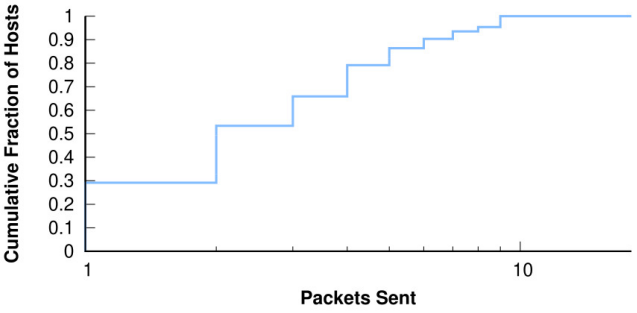

Fig. 13: Cumulative distribution of the fraction of amplifier hosts as a function of the number of packets sent by each IP.

The national scan revealed a number of vulnerable IPs much higher than expected. Millions of IPs responded to the probe packets, showing the presence of several middleboxes susceptible to traffic amplification. These results confirm the relevance and severity of the problem, highlighting the need for awareness of this attack vector in network infrastructures.

References

[1] E. J. A. Silva, G. R. L. Andrade, and R. L. S. Oliveira, "Ataques Negação de Serviço Distribuído (DDoS): o Que é e Como Prevenir," 2024.

TABLE II: Results of the scans in different IP address blocks.

Network	Responding IPs	Amplifier IPs	Non-amplifier IPs	Amplified Bytes	Average Amplification
Brazil	13,299,793	4,541,741	8,758,052	1,119,928,020	1.655x
A	12,481	8,819	3,662	2,334,588	1.777x
B	6,319	4,444	1,875	1,173,560	1.772x

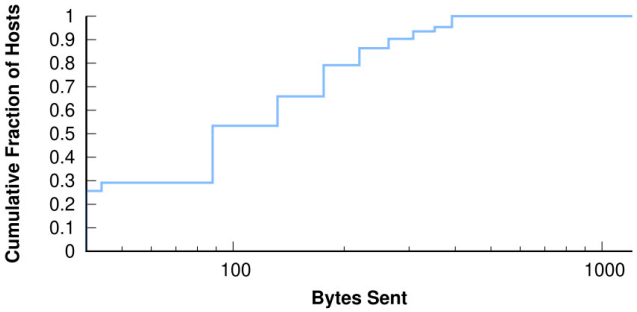

Fig. 14: Cumulative distribution of the fraction of amplifier hosts as a function of the number of bytes sent by each IP.

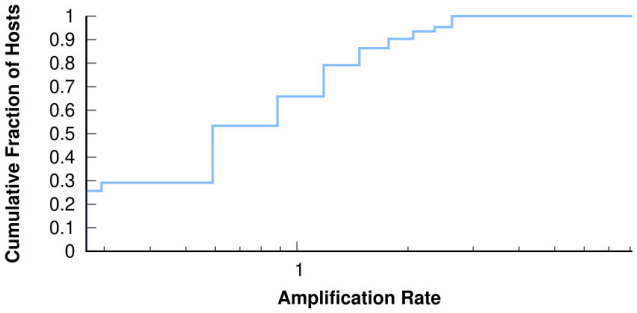

Fig. 15: Cumulative distribution of the fraction of hosts as a function of the observed amplification rate for each amplifier IP.

[2] Nexusguard, "DDoS Trend Report 2024," 2024. [Online]. Available: https://www.nexusguard.com/threat-report/ddos-trend-report-2024

[3] K. Bock, A. Alaraj, Y. Fax, K. Hurley, E. Wustrow, and D. Levin, "Weaponizing Middleboxes for TCP Reflected Amplification: Censors pose a threat to the entire Internet." Aug. 2021. [Online]. Available: https://geneva.cs.umd.edu/posts/usenix21-weaponizing-censors/

[4] Akamai Security Intelligence Response Team, "TCP Middlebox Reflection: Coming to a DDoS Near You," Mar. 2022. [Online]. Available: https://www.akamai.com/blog/security/tcp-middlebox-reflection

[5] S. Huang, F. Cuadrado, and S. Uhlig, "Middleboxes in the Internet: A HTTP perspective," *2017 Network Traffic Measurement and Analysis Conference (TMA)*, pp. 1–9, 2017.

[6] K. Bock, A. Alaraj, Y. Fax, K. Hurley, E. Wustrow, and D. Levin, "Weaponizing Middleboxes for TCP Reflected Amplification," *30th USENIX Security Symposium (USENIX Security 21)*, pp. 3345–3361, Aug. 2021. [Online]. Available: https://www.usenix.org/conference/usenixsecurity21/presentation/bock

[7] D. Pal, "A new DDoS attack vector: TCP Middlebox Reflection," Oct. 2022. [Online]. Available: https://blog.apnic.net/2022/10/18/a-new-ddos-attack-vector-tcp-middlebox-reflection/

[8] INCIBE, "TCP Middlebox Reflection: new DDoS attack vector," May 2022. [Online]. Available: https://www.incibe.es/en/incibe-cert/blog/tcp-middlebox-reflection-new-ddos-attack-vector

[9] J. Ji, "7 Gbps TCP-Middlebox-Reflection Incident Mitigated by NSFOCUS - NSFOCUS, Inc., a global network and cyber security leader, protects enterprises and carriers from advanced cyber attacks." Apr. 2022. [Online]. Available: https://nsfocusglobal.com/pt-br/7-gbps-tcp-middlebox-reflection-incident-mitigated-by-nsfocus/

[10] Shadowserver Foundation, "Over 18.8 million IPs vulnerable to Middlebox TCP reflection DDoS attacks," Apr. 2022. [Online]. Available: https://www.shadowserver.org/news/over-18-8-million-ips-vulnerable-to-middlebox-tcp-reflection-ddos-attacks/

[11] S. Meniere, "Ddos-TCP-Middlebox-Reflection-Attack," 2023. [Online]. Available: https://github.com/moloch54/Ddos-TCP-Middlebox-Reflection-Attack

[12] NIST, "CVE-2022-27491," Sep. 2022. [Online]. Available: https://nvd.nist.gov/vuln/detail/cve-2022-27491

[13] SecAlerts, "CVE-2022-2749: TCP Middlebox Reflection," Sep. 2022. [Online]. Available: https://secalerts.co/vulnerability/CVE-2022-27491

[14] FortiGuard Labs, "TCP Middlebox Reflection," Sep. 2022. [Online]. Available: https://www.fortiguard.com/psirt/FG-IR-22-073