DOI: 10.2507/34th.daaam.proceedings.xxx

DENIAL OF SERVICE ATTACKS USING THE EXAMPLE OF CROATIAN HOSTERS

Kristijan Dizdar, Zlatan Morić*, Vedran Dakić & Matej Bašić

This Publication has to be referred as: Dizdar, K[ristijan]; Moric, Z[latan]; Dakic, V[edran] & Basic, M[atej] (2023). Title of Paper, Proceedings of the 34th DAAAM International Symposium, pp.xxxx-xxxx, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-xx-x, ISSN 1726-9679, Vienna, Austria
DOI: 10.2507/34th.daaam.proceedings.xxx

Abstract

This research delves into Distributed Denial of Service (DDoS) attacks, investigating their types, tactics, and mitigation. DDoS attacks are categorized into volumetric, TCP-based, and application layer attacks, revealing their mechanics and accessibility through stressors and booters. Additionally, the study explores attack methods, such as UDP flood and amplification attacks, emphasizing their impact and challenges. It highlights the need for proactive cybersecurity measures. The research also discusses booters and stressors, tools used by cybercriminals, and mitigation techniques involving scrubbers, firewalls, and load balancers.
In conclusion, this research contributes to understanding the evolving DDoS threat landscape, emphasizing the importance of advanced detection and mitigation. Collaborative defence strategies and economic implications are also discussed, promoting network security in an interconnected world.

Keywords: Denial-of-Service; Distributed Denial-of-Service; penetration testing; cyber-attacks.

1. Introduction

The study under scrutiny immerses itself in the complex realm of Distributed Denial of Service (DDoS) attacks[1], a profound concern within the domain of cybersecurity. With a multifaceted focus, the primary objective of this research is to provide a comprehensive examination of DDoS attacks, shedding light on their diverse manifestations, repercussions, and countermeasures. By doing so, it aims to make a significant contribution to the understanding and mitigation of these disruptive cyber threats, which continue to pose substantial challenges to network security.

In pursuit of its overarching goal, this study tackles a series of fundamental questions. It delves into the different types of DDoS attacks, dissecting how each exploits specific vulnerabilities inherent in network protocols. It proceeds to unravel the tactics employed by malicious actors in launching DDoS attacks and examines the profound impact of these tactics on targeted systems. Furthermore, the study scrutinizes the existing mitigation techniques against DDoS attacks, carefully assessing their limitations and challenges. Ultimately, it strives to offer valuable recommendations to fortify cybersecurity measures in the face of ever-evolving DDoS threats.

*corresponding author

To provide a solid foundation for its exploration, this study makes extensive reference to seminal works within the field. It notably draws upon key references such as server's response to UDP packets[2], and an in-depth analysis of NTP amplification attacks[3]. Additionally, the study casts its spotlight on DNS amplification attacks[4].

In this endeavor, the study categorizes DDoS attacks into three primary classifications, namely volumetric-based attacks, TCP-based attacks, and application layer attacks[5]. It unearths the underlying mechanics of these attacks and their impact on network infrastructures. Furthermore, it underscores the disconcerting reality that DDoS attacks have become commoditized, available as services categorized as stressors and booters, showcasing their widespread accessibility and potency. This research sets the stage for a comprehensive exploration of DDoS threats and countermeasures, signifying their profound significance in the ever-evolving landscape of cybersecurity.

2. Types of DDoS attacks

Although quite similar, not all DDoS attacks are the same. The main differentiating factor between types of DDoS attacks is the protocol that is being exploited. One of the more popular types of DDoS attacks is a UDP flood. By design, the User Datagram Protocol is session less and connectionless. This means that for the application or service to work over UDP, the server and client do not need to establish a connection for the service to work. One example of such a service would be DNS, which primarily uses UDP port 53.

One design pattern of UDP that can be easily exploited to carry out DDoS attacks is the algorithm that each network device goes through when receiving a UDP packet. Under normal conditions, when a server receives a UDP packet, there are two steps, according to conditions[6]:

- Server checks if there is a service running on a particular port, in a listening state for new incoming requests.
- If the first condition is not fulfilled, the server will return ICMP (or ping) a packet back to the sender, informing him that the destination is unreachable.

With every packet sent, the server needs to use its resources to process requests. Every UDP packet has the IP address of a source machine. When the attack is launched from a well–organized botnet, attackers will usually spoof the originating IP address so that the returning traffic does not affect the botnet. As there are no checks such as when using TCP, the server would usually return a response to the source IP address.

The way this is leveraged in a DDoS attack is that an attacker would use an entire botnet of infected servers that would then send UDP packets with a spoofed IP address towards the victim. The victim would then try to process these requests and send out ICMP error packets to the spoofed IP sent in the initial request header.

Most modern systems implement some type of rate limit, but assuming the length of each packet is 1500 bytes, the compute and network resources on the victim machine are overwhelmed for the entire duration of the attack. Using this method, the attacker first overwhelms the victim's network bandwidth and later saturates the server resources.

In an example of a DDoS attack including 20 20,000 devices, which is nothing out of the ordinary, that send 10 packets per second, the target server needs to process 200 thousand requests per second. Even the most efficient computing resources today are not able to process this amount of sustained traffic.

In UDP-based attacks, a single server could, in theory, generate gigabits per second of traffic. However, this way, the attacker's IP addresses could be tracked. For this reason, attackers usually use methods of amplification to not only amplify the amount of traffic being sent to the victim but also create a scenario in which the original attacker's IP address is not visible from the victim's machine. The most popular methods of amplification are DNS amplification and NTP amplification.

Network Time Protocol is a service used to synchronize time on different devices. NTP has been a part of every network for decades and is one of the oldest Internet protocols still in use. One specific command, monlist, sent to an NTP server is the dominant method of executing an NTP amplification attack. Monlist was designed as a control message command that prints traffic counts collected and maintained by the monitoring facility of an NTP daemon. The monitoring facility of an NTP daemon collects and maintains data of the Most Recently Used hosts or clients that have communicated with the NTP daemon. Monlist command can display a maximum list of 600 entries. As the initial NTP request sent from the client to the server is relatively small, in the case where the monlist command returns a list of 600 servers, the response that the server sends will be 206 times larger than the initial request.[3]

DNS amplification is another variation of amplified DDoS attacks. Domain Name System is the most used protocol on the internet and uses both UDP and TCP port 53. One thing that makes DNS a great protocol for DDoS attacks is the fact that a DNS response packet is significantly larger than a DNS query packet. The method of exploiting DNS for a

DDoS attack is the same as with NTP. The attacker sends many DNS queries with a spoofed IP address that points to the victim to several DNS resolvers that then start sending responses to the victim. The average size of a DNS request is 60 bytes. The average size of a DNS response is 3 kilobytes or 3200 bytes, which makes the amplification factor to be around 50x, or even 100x in rare cases.[7]

UDP flood attacks are volumetric-based attacks. The goal of these attacks is to exploit a well-known protocol that most devices use to saturate the network connection on the victim side. Most of the time, completely preventing these types of attacks is not possible, but certain remediation methods are available that can help in scenarios when their infrastructure is a target of a DDoS attack.

Alongside UDP flood attacks, several attacks are exploiting various protocols that use TCP. Transmission Control Protocol, unlike UDP, requires a connection to be established between the client and a server for the communication to be successful. The way the server and a client establish this connection is by performing a 3–way handshake, a 3–step procedure that is used to prepare both sides.

The three-step procedure is as follows:

- Client sends a SYN packet to initiate a connection.
- Server returns a SYN/ACK packet as a response.
- Client sends an ACK packet.

Once this procedure is completed, the TCP connection will be open, making the client available to send and receive data. Attackers can use SYN flood attacks by exploiting this design.

An SYN flood attack, also known as a “half-open” attack, is a type of DDoS attack targeting all systems, with services using TCP protocol such as web servers, mail servers, etc.[8]

The way a SYN flood attack is implemented is by exploiting the last step in a 3–way handshake. The attacker will send many SYN packets to the target server, again usually by using a botnet of infected devices. In general, there are 3 main vectors in which a SYN flood attack can be done:

- Direct attack – In this case, attackers do not spoof their IP address, but rather use various routing and filtering rules on their side so that their machine cannot respond to SYN/ACK packets.
- Spoofed attack – As is the case with most UDP flood attacks, in spoofed SYN flood attacks, attackers will also spoof their IP address to me the process of tracing the traffic back to them more difficult.
- Distributed attack – the most common type of a SYN flood attack in which an entire botnet of infected devices attacks the same target, making it almost impossible to trace traffic back to the original attacker.

The reason SYN flood attacks work is that the server will allocate resources upon sending a SYN/ACK packet back to the client. One of the most popular web servers today, Apache, schedules a new thread for each new incoming connection. Although this design is great for normal operations, it can be easily exploited by opening a bunch of connections to the server and never closing them. By doing this, we can schedule a bunch of threads and exhaust memory on the target server without ever compromising our Internet connection. That very reason is the main reason this method of implementing a DDoS attack is the most popular one. As it is not volumetric, it does not require a lot of traffic being sent to the target server which means it can be done using a smaller number of infected devices.

The final category of DDoS attacks we will cover are application layer or Layer 7 attacks. Alongside volumetric and protocol–based attacks, another popular method for implementing a DDoS attack is an HTTP flood attack. Hypertext Transfer Protocol is used for exchanging data between a server and a client, mostly used for serving websites on the server side and rendering websites on the client side. [6]

HTTP flood attack is a specific type of attack where attackers send requests as legitimate users but with a difference in the amount of potentially open connections. HTTP protocol defines different types of requests, with the 2 most used being GET and POST. A GET request is usually used to request something from the server, while a POST request is used to send some data from the client to the server. By exploiting a GET request, an attacker can perform a GET flood. A GET request is what a typical web browser sends to a web server to fetch web content. In this case, attackers impersonate real clients. The key to this attack is that it requests legitimate content served on the web server. Additional configuration pieces that make this attack even more complicated to remediate are the following:

- Using residential IP addresses – modern DDoS attacks are performed by leveraging large botnets of infected devices. These botnets are comprised of IoT devices that are notorious for having subpar security standards. Given that IoT

devices are connected to home networks, the originating IP address that the victim will see will be the public IP address that most of the time is not static
- User agents – By configuring User agent fields in HTTP requests, an attacker can impersonate a web browser client, such as Google Chrome, even though the requests might originate from an IoT device.
- Three–way handshake – connections must be open between 2 sides during the entire time of the attack.

On the other hand, a POST flood attack exploits the design of a POST request. As already mentioned, POST requests are used to send data from the client machine to the server, which means that to exploit this request, the attacker must keep the connection alive from the client side and try to exhaust server resources. The idea behind POST flood attacks is to send a small amount of data over a period just to keep the connection open on the server side.

Aside from classic GET and POST flood attacks, a more modern approach is something named Slow Client Attacks. Slow Client Attacks behave like classic GET and POST flood attacks, but the patterns and execution times are different. Slow client attacks are assigned to send specific requests to make the traffic flow slower. This is done because of the design of the protocol itself – the server must keep the connection open until a signal is sent. Denial of service happens when a web server has too many open connections and cannot open new ones for legitimate users. The most common implementations of slow HTTP attacks are:

- SlowLoris[9]
- Slow HTTP Post[10]
- Slow Read attack[11]

All these categories are known as “low and slow” attacks. They have a mechanism in common – sending a small amount of traffic and maintaining slow connections. Unlike traditional attack types, they do not require a lot of bandwidth volume to take down an average web server[12]. Some of the techniques for performing DDoS attacks may seem complicated. In reality, a DDoS attack can be bought as a service online. These services are separated into 2 categories – stressers and booters.

3. Booters and stressors

Booters and stressers are often offered by hackers, or cybercriminals, who make botnets using cloud virtual machines and infected devices all over the world. All stressors have the same methodology in common, using proxy servers and various other techniques for spoofing IP addresses of attack machines to protect their botnet and make it more difficult to trace the attack back to them.[13]

The way These services are usually presented as a toolset used to stress test your infrastructure to make sure that you could handle a DDoS attack if you were ever a victim of one. Both terms have the same meaning in practice. On the front end, hackers usually offer a web page where you can sign up for their services. These web pages are often hosted in non–EU countries to avoid being banned from the Internet. In most countries, it is illegal to create and sell such services. Today, you can buy a subscription and start performing a DDoS attack after just a few minutes of searching the web. Once you land on one of these pages you are immediately presented with the price and the payment method. Prices vary from one provider to the other, and based on their reputation, the prices can be as low as 20 USD or go up to as high as 1000 USD for a single attack.

4. Tests and results

Tests were performed on 3 independent hosting providers available in Croatia. Tests were carried out using “DDoS for hire” services explained earlier in the paper. Tests were announced upfront, and written consent was received from all 3 hosting providers. Results are separated into 3 tables, one for each hosting provider. Test results for the first provider are in the table below:

#	Attack Vector	Target	Rate	Result
1	Advanced HTTPS Flood (without protection)	Website	90K R/S	FAIL
2	Advanced HTTPS Flood (with scrubber on)	Website	90K R/S	FAIL
3	VSE Flood @ 443 port – with protection disabled	Website	120KPS	Partial pass
4	VSE Flood @ 443 port – with protection enabled	Website	120KPS	Partial pass
5	SYN Flood @ 443 port – with protection	Website	1M PPS	FAIL
6	SYN Flood @ 443 port – without protection	Website	1M PPS	FAIL
7	UDP-MIX - against 22 ports – protected	Host	20Gbps	FAIL
8	UDP-MIX against 22 ports - unprotected	Host	20Gbps	FAIL
9	HTTP get flood – without protection	Website	80K R/S	FAIL
10	HTTP get flood – with protection	Website	80K R/S	FAIL

11	HTTP raw get – with protection	Website	50K R/S	FAIL
12	HTTP raw get – without protection	Website	50K R/S	FAIL
13	HTTP null – with protection	Website	50K R/S	Pass
14	HTTP null – without protection	Website	50K R/S	Pass

Table 1. First provider results

Result table for the second hosting provider:

#	Attack Vector	Target	Rate	Result
1	Advanced HTTPS Flood (without protection)	Website	90K R/S	FAIL
2	Advanced HTTPS Flood (with scrubber on)	Website	90K R/S	FAIL
5	SYN Flood @ 443 port – with protection	Website	1M PPS	PASS
6	SYN Flood @ 443 port – without protection	Website	1M PPS	FAIL
7	UDP-MIX - against 22 ports – protected	Host	20Gbps	PASS
8	UDP-MIX against 22 ports - unprotected	Host	20Gbps	FAIL
9	HTTP get flood – without protection	Website	80K R/S	PASS
10	HTTP get flood – with protection	Website	80K R/S	PASS
11	HTTP raw get – with protection	Website	50K R/S	FAIL
12	HTTP raw get – without protection	Website	50K R/S	FAIL
13	DNS-AMP against 53 port - protected	Host	300K R/S	PASS
14	DNS-AMP against 53 port - unprotected	Host	300K R/S	FAIL

Table 2. Second provider results

Result table for the third hosting provider:

#	Attack Vector	Target	Rate	Result
1	Advanced HTTPS Flood – protected subnet #	Website	90K R/S	FAIL
2	Advanced HTTPS Flood – protected subnet #2	Website	90K R/S	FAIL
3	VSE Flood @ 443 port – protected subnet #1	Website	120KPS	PASS
4	VSE Flood @ 443 port – protected subnet #2	Website	120KPS	PASS
5	SYN Flood @ 443 port – protected subnet #1	Website	1M PPS	PASS
6	SYN Flood @ 443 port – protected subnet #2	Website	1M PPS	PASS
7	UDP-MIX - against 22 port – protected subnet #1	Host	20Gbps	PASS
8	UDP-MIX against 22 port–protected subnet #2	Host	20Gbps	PASS
9	HTTP get flood – protected subnet #1	Website	80K R/S	FAIL
10	HTTP get flood – protected subnet #2	Website	80K R/S	FAIL
11	HTTP raw get – protected subnet #1	Website	50K R/S	Partial pass
12	HTTP raw get – protected subnet #2	Website	50K R/S	FAIL
13	HTTP null – protected subnet #1	Website	50K R/S	Pass
14	HTTP null – protected subnet #2	Website	50K R/S	Pass

Table 3. Third provider results

The DDoS testing was a basic analysis that included seven attack vectors. The network attack (UDP flood and SYN flood) was not blocked. While protection was on, it required about five minutes of detection time in terms of UDP flood, only to lower packet loss. Services behind the host were not available during that time.

Only one vector had a partial pass, and that was the VSE vector. Five vectors were against the website itself, and none of them were detected by DDoS scrubbing services, as the results were the same.

5. Methods of remediation

Hosting your applications on your infrastructure has many advantages, but handling DDoS attacks surely is not one of them. Apart from the already mentioned method of adding more devices, one common method companies use to remediate this threat is using one of the online services that claim to be able to filter and handle traffic for you, often called scrubbers. One of the most popular companies that provide such services is CloudFlare[14]. CloudFlare offers a wide range of services, all designed to make your infrastructure and your services more resilient and less vulnerable to DDoS attacks.

Based on the types of attacks mentioned earlier, we define different methods for mitigation and prevention of these attacks. Prevention and mitigation solutions for amplified attacks are limited. The main reason behind this is the amount of traffic generated from these attacks, where the surrounding networks usually have problems with processing this type of traffic. Usually, ISPs blackhole all that traffic, but the effects for the site owner are practically the same since, while an attack is ongoing, the server receives packets the whole time.[15]

On the other hand, many scripts available have recognizable patterns that can be easily blocked on edge routers. Along with filtering by pattern, some of the techniques used can be to limit the NTP packet ratio or verify the source IP address.

When it comes to TCP-based attacks, there are more advanced methods of mitigation, both on the server itself and once designed to be placed "in front" of the server. Firewall filtering is a method that mitigates a large portion of SYN flood attacks. Many modern firewalls come with predefined patterns for detecting SYN attacks. They can fully prevent or limit the surface of incoming attacks. A popular example of such systems would be a FortiGate firewall which has a threshold for TCP SYN flood. In general, it intercepts TCP connections before they come to the server itself, and a numeric limit of incoming connections can be implemented along with a waiting time. All stages of a 3–way handshake will be carried out between the client and a firewall, after which the firewall itself makes a "replay" to the original destination. If a threshold is reached, all further packets will be blocked, but with the exception that SYN packets will be allowed if clients send another SYN packet – simply put if the client performs a retry, a full connection will be opened. This is the main benefit of using a firewall as a separate appliance. Something similar could be implemented on the firewall on our server itself on the OS level, but having a firewall on the same machine as the web server means that the same server must process both real and malicious network traffic which could highly increase compute resource consumption on the target server.

The second method we can use to mitigate against SYN flood is half–open connection recycling. Half–open connection recycling means that time for leaving half–open connections can be lowered along with cleaning a certain number of recent connections, but this approach does not have any effects if the attack has a higher volume.

As for HTTP flood prevention techniques, scrubbing services still barely manage to mitigate attacks, except for a few industry leading CDN services with additional protection layers in the background. Because of that, many hosting providers decide to invest money and effort in on–premise solutions, which are usually open–source and easier to maintain and configure. Mitigation techniques include:

- Using a CDN
- Web application firewalls
- Web server hardening
- Proxy as a mitigation mechanism

Web application firewalls are usually software suites installed on an endpoint, and they reside between a web server and clients connecting to it. The primary goal of WAFs is to monitor and filter traffic between the two sides.

If a condition is fulfilled in terms of server resources, a WAF can be an efficient tool in mitigating not only DDoS attacks but other types of malicious traffic such as SQL injections, cross–site scripting, etc. In terms of mitigating an attack, a firewall is effective against low–volume attacks, but strong attacks reach the server. A WAF works like any other firewall, it has a set of policies protecting against vulnerabilities. As a DDoS attack arrives, a WAF is capable of rate–limiting addresses, blocking them from further attacks. However, attackers often use proxies to hide IP addresses, which can leave a server struggling to block all IP addresses in a timely manner.

Web application firewalls operate in two ways:

- Blacklist – usually by IP ranges pulled from known vendors like *SpamHaus* or by pattern–specific options.
- Whitelist – only specific IP ranges can access the server, or only specific browser agents or specific patterns should be fulfilled to open a webpage.

The cheapest mitigation technique that can be implemented on a customer's premises is to utilize additional publicly available modules, along with web server packages, nginx, and Apache. *Mod_evasive* module is the most popular solution among shared hosting providers. This addon monitors connections for DDoS attacks and brute force attacks. This works similarly to WAFs, and its main feature is that in case of excessive connections to the resource the IP gets banned for a period. This module works only if the client experiences GET and POST flood attacks with higher volume, while low – and – slow attacks go undetected.

Another open–source solution that could be utilized is *HAProxy*.[16] *HAProxy* is a high-availability load balancer and reverse proxy for TCP and HTTP applications. This tool balances traffic to two or more servers in the backend after health checks are performed. *HAProxy* is used not only for web applications but it can be used for all TCP–based applications, such as SQL databases. Companies would usually set up *HAProxy* with an external IP address. From the Internet, the application hosted on our server is available over ports 80 and 443. The request first ends up on the load balancer, behind which multiple backend servers are up and running.

6. Further recommendations for mitigation

DDoS attacks are primarily used for disrupting services and causing business damage through every second of downtime with a huge load of illegitimate traffic. As a countermeasure to these attacks, companies need to utilize a solution that will amortize the effects of received attacks.

DDoS mitigation is a process where a protection service of any kind, such as a scrubbing service, firewall, or appliance, protects resources from DDoS attacks, thus protecting resources from being unable to process legitimate requests. As DDoS attacks are developing with new methods and growing each day in terms of their strength, mitigation services are improving at the same pace.

Mitigation can be achieved with three models:

- On-premises
- Cloud-based
- Hybrid environment

There are pros and cons of using various cloud-based and/or hybrid solutions. However, an additional problem is the public or hybrid cloud computing model, where a single physical server can contain multiple virtual machines[17] that could suffer noisy-neighbour problems (influence each other's performance) or have availability problems. This is why these types of simulations are a powerful tool used in testing organizations' vulnerability and exposure to cyberattacks[18]. Furthermore, there's a problem of digital footprint, which is what attackers will try to use to extract useful information to be even more malignant when initiating cyberattacks[19]. The same recommendations that we have for employees of any company apply to our scenarios here, as well – the digital footprint of a service provider needs to be tightly managed and as small as possible.

The mitigation process is divided into four steps: detection, diversion, filtering, and analysis. The first phase is detection, where the environment should be aware of incoming traffic if the service and enterprise using it expect effective mitigation. The primary goal of DDoS is to flood the destination with a lot of traffic, In the detection phase a particular service should detect patterns of incoming attacks and if there is a match, block the traffic.

Patterns must be prioritized instead of source IP addresses, as botnets consist of numerous machines with different source IP addresses. The reason behind this is that attacks cannot be blocked by banning IP addresses, as there is a possibility that behind these addresses are legitimate users, and such users would not be able to access the content, especially if addresses are CGNAT–ed.

The process of detection consists of receiving and identifying anomalies in network traffic reaching the resource. To be successful in resolving anomalies promptly, owners or providers should decide how they will face these attacks by blackholing, enabling a BGP speaker, or using a scrubber service.

The next phases are interpreted differently by various DDoS protection services. After detecting a DDoS attack, a protection service should be able to drop malicious traffic incoming from a botnet, while legitimate traffic should reach the destination. This is implemented by using routing schemes, where traffic patterns are identified and dropped without blocking the IP address itself, because of possible blocking of further legitimate traffic from the same IP.

Building upon the insights gained from this research, several promising avenues for future investigations emerge. Firstly, researchers should delve deeper into the behavioural analysis of DDoS attacks, exploring advanced machine learning and AI-based techniques to enhance real-time detection and mitigation. Additionally, there is a need to focus on zero-day DDoS attacks, understanding how attackers exploit previously undocumented vulnerabilities in network protocols, and developing proactive defences against such emerging threats.

Furthermore, the intersection of blockchain technology and DDoS mitigation presents an intriguing research area, with potential for decentralized and resilient solutions. As the Internet of Things (IoT) expands, it becomes crucial to explore the vulnerabilities and threats posed by IoT devices in the context of DDoS attacks and develop strategies for securing IoT networks. Additionally, researchers can delve into the impact of quantum computing on DDoS defence,

investigating post-quantum cryptography techniques to secure network communication. A comprehensive study of the global DDoS threat landscape, region-specific trends, and emerging patterns can inform adaptive mitigation strategies. Research into human factors, cybersecurity awareness, and incident response training is essential, as is the improvement of DDoS attack attribution methods. Finally, exploring the economic and policy implications of DDoS attacks and collaborative defence strategies that involve information sharing among organizations and service providers are also promising areas for future inquiry.

7. Conclusion

Nowadays many issues cannot be easily addressed. The number of botnets is growing daily, and these threats consist of devices that people did not even expect to cause issues, meaning IoT is the biggest problem.
In this paper, we have used a DDoS-for-hire service, consisting of dedicated servers just for that purpose and a small number of infected devices. DDoS-for-hire services are powerful these days and can be made with minimal effort and investment, thanks to revealing methods and a constantly growing community on public repositories, such as GitHub, along with so-called "underground" forums, which can be accessed from a single Google search.

We have selected three independent "DDoS-for-hire" services, popular as "stressors", which can be found on the same resources, meaning that they are practically publicly advertised with one "security" measure: they can only be paid via cryptocurrencies, as they consider it the safest way to hide their identity. These tools were selected by "word of mouth", as sellers often scam people, selling services as the "most powerful" which in practice cannot reach the target with the required strength.

The primary goal of this work was to carry out a stress test on several local hosting companies. All of them had subscriptions to scrubbing services, claiming that they could filter the most known attack types.

Testing results:

- None of the providers passed all tests
- One provider failed to mitigate Layer 3 and 4 attacks, even with the scrubber enabled
- Layer 7 attacks were not filtered in most tests

In the analysis, the author did not have full insight into the other side of the test (incoming bandwidth, incoming requests). The only true result that can be seen and is realistic for a reader is load time from the browser, tested from two independent carriers (home line subscription and mobile carrier).
Further, after testing, several mitigation techniques were analysed to find a nearly fully working solution. All of them have a single point of failure, except CloudFlare at this point, as it combines all possible mitigation methods with proprietary technologies (like Under Attack Mode).

Overall, these results are unexpected, as the services they are using are known worldwide, without any recently reported issues. In the appendices, several recommendations should be tried to address these issues.

On the other side, there were no recent attacks on hosting providers in Croatia with significant impacts, but this work showed that almost all of them should consider changing their approaches to defending against DDoS attacks. Unfortunately, in most cases, prevention techniques are not cost-effective, especially for the Croatian market. Small businesses will still use these providers as they are placed locally and the simplicity of using these has more advantages over security, but the market is unfortunately not ready for bigger-scale projects and mission-critical web pages, which are already behind CDN providers like CloudFlare or Akamai.

8. References

[1] Osterweil, E.; Stavrou, A. & Zhang, L. (2020). 21 Years of Distributed Denial-of Service: Current State of Affairs, Computer (Long. Beach. Calif)., vol. 53, no. 7, pp. 88–92.
[2] Thomas, D. R. ;Clayton, R. & Beresford, A. R. (2017). 1000 days of UDP amplification DDoS attacks, 2017 APWG Symposium on Electronic Crime Research (eCrime), pp. 79–84.
[3] Rudman, L. & Irwin, B. (2015). Characterization and analysis of NTP amplification based DDoS attacks, 2015 Information Security for South Africa (ISSA), pp. 1–5.
[4] Gupta, V.; Kochar, A.; Saharan, S. & Kulshrestha, R. (2019). DNS Amplification Based DDoS Attacks in SDN Environment: Detection and Mitigation, 2019 IEEE 4th International Conference on Computer and

Communication Systems (ICCCS), pp. 473–478.
[5] Douligeris, C. & Mitrokotsa, A. (2003) 'DDoS attacks and defense mechanisms: a classification', Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology (IEEE Cat. No.03EX795), pp. 190–193.
[6] Bijalwan, A.; Wazid, M.; Pilli, E. S. & Joshi, R.C. (2015). Forensics of Random-UDP Flooding Attacks, J. Networks, vol. 10.
[7] https://www.cloudflare.com/learning/ddos/dns-amplification-ddos-attack (2022). DNS amplification DDoS attack | Cloudflare. Accessed: 12-Feb-2023.
[8] Yuan, D. & Zhong, J. (2008). A lab implementation of SYN flood attack and defense, Proceedings of the 9th ACM SIGITE conference on Information technology education, pp. 57–58.
[9] Shorey, T.; Subbaiah, D.; Goyal, A.; Sakxena, A. & Mishra, A. K. (2018). Performance Comparison and Analysis of Slowloris, GoldenEye and Xerxes DDoS Attack Tools', 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 318–322.
[10] Calvert, C.; Kemp, C.; Khoshgoftaar, T.M. & Najafabadi, M.M. (2019). Detecting slow http post dos attacks using netflow features, Proc. 32nd Int. Florida Artif. Intell. Res. Soc. Conf. FLAIRS 2019, pp. 387–390.
[11] Tayama, S. & Tanaka, H. (2017). Analysis of Effectiveness of Slow Read DoS Attack and Influence of Communication Environment, 2017 Fifth International Symposium on Computing and Networking (CANDAR), pp. 510–515.
[12] https://www.cloudflare.com/learning/ddos/ddos-attack-tools/slowloris (2022). Slowloris DDoS attack | Cloudflare, Accessed: 12-Feb-2023.
[13] Haria, S. (2019). The growth of the hide and seek botnet', Netw. Secur., vol. 2019, no. 3, pp. 14–17.
[14] Blankenship, J. (2017). The Forrester Wave: DDoS Mitigation Solutions, p. 16.
[15] Kushwah, G.S. & Ali, S.T. (2017). Detecting DDoS attacks in cloud computing using ANN and black hole optimization, 2017 2nd International Conference on Telecommunication and Networks (TEL-NET), pp. 1–5.
[16] Cruz, J.E.C. & Goyzueta, I.C.A.R. (2017). Design of a high availability system with HAProxy and domain name service for web services, 2017 IEEE XXIV International Conference on Electronics, Electrical Engineering and Computing (INTERCON), pp. 1–4.
[17] Cvitić, I.; Vujic, M. & Husnjak, S. (2016). Classification of Security Risks in the IoT Environment, 26th DAAAM International Symposium, Vienna, Austria, pp. 0731–0740.
[18] Kafol, C. & Bregar, A. (2017). Cyber Security – Building a Sustainable Protection, DAAAM International Scientific Book pp. 081–090.
[19] Matvej, E.; Moric, Z. & Papic, S. (2020). Croatian Bank Security Analysis by Publicly Available Data, Proceedings of the 31st International DAAAM Symposium, pp. 0184–0188.