National College of Ireland

Detection and mitigation of DNS laundering DDoS attacks

MSc Research Project
CyberSecurity

Kevin Salvador Garza Ruiz
Student ID: X22203788

School of Computing
National College of Ireland

Supervisor: Dr. Vanessa Ayala-Rivera

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Kevin Salvador Garza Ruiz
Student ID:	X22203788
Programme:	CyberSecurity
Year:	2023
Module:	MSc Research Project
Supervisor:	Dr. Vanessa Ayala-Rivera
Submission Due Date:	14/12/2023
Project Title:	Detection and mitigation of DNS laundering DDoS attacks
Word Count:	5736
Page Count:	23

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	**Kevin Salvador Garza Ruiz**
Date:	30th January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	✓
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	✓
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	✓

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Detection and mitigation of DNS laundering DDoS attacks

Kevin Salvador Garza Ruiz
X22203788

Abstract

Distributed Denial-of-Service (DDoS) attacks has raised new concerns during the first half of the current year (2023) since longer and more intense attacks has been detected. A recently observed raise of DDoS attack related to Domain Name System (DNS) has been identified as DNS laundering DDoS attack, where the attacker manages to send DNS request to the target through a DNS resolver making the requests appear to be legit. This paper is presenting a solution for DNS laundering DDoS attack and giving a comparison of the effectiveness given by existing methods such as black-hole and rate limit. The proposed solution implies a device that acts as DNS controller which can detect a DDoS DNS laundering attack, block the attack, and keep the access to the targeted domain from users' perspective. In this work the comparison between different methods and the proposed solution is given by experimentation, where in all cases when the proposed solution is applied, more than 99.6 percent of the load sent by the attacker is dropped, a reduction of 11.54 percent on memory utilization against "under attack" state is achieved, and finally 45.71 percent of swap memory utilization is reduced against "under attack" state. All these results are given under the main approach of the proposed solution that is giving access to legit users during the attack. The comparison given by the experimentation shows that blackhole solution accomplish the task of block the attack but failing on give legit users access to the targeted domain, on the other hand rate limit is successful on give legit users access to victim´s domain but only blocking partially the attack. Finally, the proposed domain is successful on blocking the attack and give legit users access to the victim´s domain.

1 Introduction

DDoS attacks are based on a series of machines, bots, or botnets that floods a target that could be a website or a web service with network traffic where the goal is to push real users out, the result of the attack is to delay a service or even make it fail for a period of time. One of the risks that a web service face during a DDoS attack is that security vulnerabilities can be exploited and eventually gaining access to databases and getting control over sensitive information [1]. Also DDoS attacks usually try to inundate with queries the resources of the victim to affect the functionality of the victim´s resources and avoid genuine users to access to the victim´s service [2].

The quantity of DDoS attacks has growth in 30.5 percent in the first half of 2023 compared with the first half of 2022 with a total of 7,858,705 attacks, this represents a daily number of attacks of 44,000 where the most targeted area was EMEA (Europe,

Middle East and Africa) with around 15,000 attacks, and the second one is APAC (Asia-pacific) with around 9,000 attacks [3].

1.1 DNS laundering DDoS attacks

The Domain Name System (DNS) is a name resolution system that allows the translation of IP addresses to domain names, this system is widely used since for humans is easier to remember a domain name (for example google.com) that remember an IP address which is composed by numbers (for example 192.168.100.109) [4].

DDoS DNS attacks has evolved since there are many types of them where some examples are DNS flood and DNS amplification, in the first one a series of boots present on internet of things devices target a DNS resolver in order to make legitimate users unable to access to provider´s DNS services [5]. On the other hand, DNS amplification is based on request with spoofed IP address where the source IP address has been substituted with the target address, as a result, the victim receives a big amount of DNS requests coming from DNS resolvers [6].

DNS laundering DDoS attacks is a type of DNS DDoS attack where the attacker make the traffic appear legit by a laundering process using recursive DNS resolvers [7]. In this type of attacks, the hacker sends queries containing subdomains of a domain managed by the victim´s DNS server. The prefix of the domains is randomized and used only once. Due to the randomization of the subdomains, the DNS resolver would never have a response on the cache, then DNS resolver would forward the queries to the authoritative DNS server of the victim´s domain. In consequence the authoritative DNS server of the victim would be bombarded by legit queries [7].

The detailed steps presented on this type of attack are explained upcoming where I clarify how DNS laundering attacks are being performed step by step with purpose to have a clear understanding of the behaviour of the attacker, recursive DNS resolver and authoritative DNS server under this particular type of attack.

Step 1: Figure 1 is showing the attacker sends randomized subdomain queries to the DNS resolver

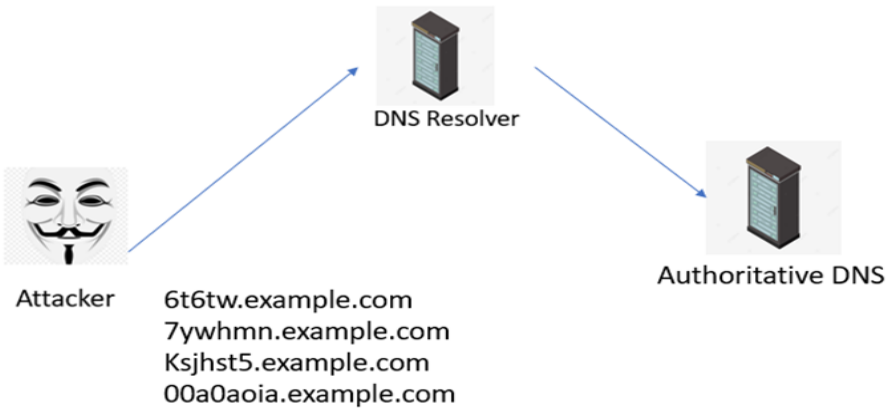


Figure 1: Randomized subdomain queries sent by the attacker

Step 2: The recursive DNS resolver does not have any cached entrance for the subdomains present during the attack, as showing in figure 2.

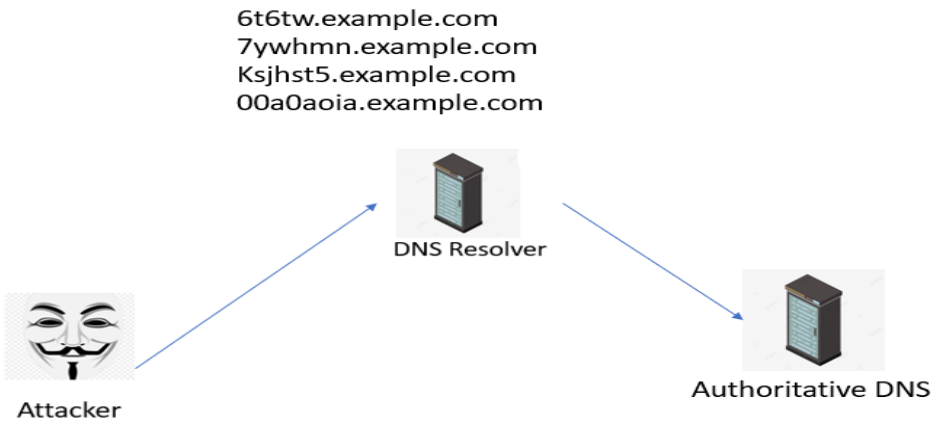


Figure 2: Recursive DNS resolver do not have cached entrance for the subdomains received

Step 3: The DNS resolver forward the randomized queries to the authoritative DNS server of the victim, as showing in figure 3.

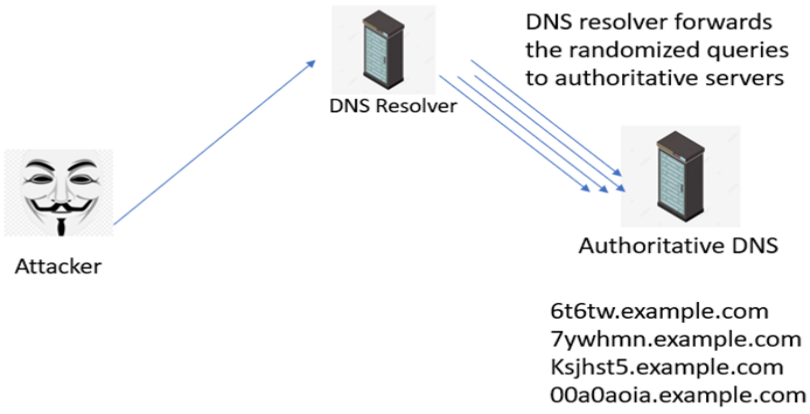


Figure 3: The DNS resolver forward the queries of the attack

Using this methodology attackers can perform DDoS attacks to authoritative DNS server by making their request genuine, since they are coming from a valid recursive DNS resolver.

1.2 Proposed solution

The primary goal of this paper is to simulate a potential solution for DNS laundering DDoS attacks where malicious traffic is successfully blocked, and genuine users do not lose access to the targeted domain. Also, another objective of this paper is to compare the proposed solution with other types of solutions seen before and to contrast the main advantages and disadvantages of the proposed method. As research question I am proposing: Is the proposed method effective on detecting and mitigating a DNS laundering DDoS attack while let the genuine traffic reach the targeted domain?

To accomplish the task of blocking a laundering DDoS attack and let genuine traffic reach the targeted domain, I am proposing a method where a DNS traffic controller placed between the authoritative DNS server and recursive DNS resolver can discriminate the traffic that is going through by implementing an algorithm, this algorithm would analyse the traffic and determine if the authoritative DNS server is under a DNS laundering DDoS attack and act in consequence by blocking the attack while letting genuine traffic go through.

From the protection perspective, the attack can´t be blocked by blocking the source of the attack since it comes from a legit reputable recursive DNS server like google DNS. This factor makes this type of attack very challenging to distinguish between valid queries from malicious ones [7].

2 Related Work

In this section I discuss the different ways of mitigating a DDoS attack based on network layer, I analyse their positive and negative aspects as a way of make a comparison to the proposed method on this paper. Different methods and solutions have been approached in past papers that give solution to some types of DDoS attacks. However, in this paper I am studding the effectiveness of the proposed method against DNS laundering DDoS attacks and comparing it to the solutions given by past works.

2.1 DDoS detection

Several approaches have been achieved to detect and mitigate a variety of types of DDoS attacks in the past, e.g., Antonakakis [8] has developed a mechanism to classify domain generation algorithms (DGA) based domains using non-existent domain responses (NXDOMAIN). On that work he was able to classify and detect different types of DGA-based bots based on unsuccessfully resolutions However this method was no focused on target randomly generated subdomains as DNS laundering DDoS attack is based on.

Also, Sommese [9] mentioned that "Distributed anycast deployment is generally the most effective method to operationally mitigate the effects of DDoS attacks on end users". Both previous mentioned works [8] [9] was developed on ISP (Internet Service Provider) side. On this paper the proposed solution is given on the victim´s side, since DNS laundering DDoS attack is focused to reach an authoritative server, it is a very focused attack that can hardly be dissipated through different servers.

Soliman [10] has proposed a method to detect attacks coming from clouds through a method called Hypervisor Ingress Filtering (HIF), which is useful to detect reflection amplification DDoS attacks.

Saharan [11] has proposed a solution for DNS amplification attacks where as mentioned on his work "The solution presented here involves a set of geographically distributed routers, called a Barrier of Routers (BoR)". the barrier of routers mentioned by Saharan [11] have the task of drop the packets that comes from DNS amplification attacks, by scanning the incoming an outgoing traffic.

Hasegawa [12]presented a solution for water torture DNS attack (which is another name for DNS laundering DDoS attack) where applying a whitelist to incoming queries, Hasegawa [12] accomplish to mitigate water torture DNS attack but certainly not to ensure legit users to have access to the victim´s domain as mentioned on its report "In other words, it is also highly possible that the FQDNs that users frequently resolve are registered in the whitelist. Therefore, we hypothesize that legitimate users will not be significantly affected by the false dropping of legitimate DNS queries even during an attack." [12]. Also the solution given on Hasegawa paper is implemented on DNS resolver server, meanwhile in this paper is implemented on a DNS traffic controller next to authoritative server.

2.2 Rate limit solutions

Rate limit solutions are given by determining a limit for the packets or information per second that go through a system. As the conclusion given by Deccio [13] around 16 percent of authoritative servers implement a sort of rate limitation to the traffic when they are under a DDoS attack, letting DNS server with connection to internet but with the inconvenience that the bandwidth would be used for both genuine and malicious traffic.

Wong [14] analysed a modification of Ganger's NIC-based DNS detection scheme [15], where the traffic is limited depending on previous DNS translation for the destination IP. As Wong [14] mentioned "for every outgoing TCP SYN, the rate limiting scheme permits it through if there exists a prior DNS translation for the destination IP, otherwise the SYN packet is rate limited".

Rozekrans [16] Analysed a response rate limit method to limit the effectiveness of DNS amplification attacks by dropping packet responses that overpass the rate limit established.

Also, as mentioned by Rozekrans [16] and Wong [14] rate limiting is an effective way to partially mitigate a DDoS attack but it comes with side effects like the actual DDoS attack is never stopped just avoiding the server to be vulnerated by limiting the amount of queries processed.

2.3 Black-hole Solutions

Black-hole solutions are given by deviating inbound traffic to a non-existent IP address, by the this way DDoS attacks can be mitigated with a high effectiveness as implemented by Jonker [17] using the BGP routing protocol.

Also, Ishibashi [18] implemented a method to mitigate a DDoS attack from ISP side by blocking the source of the attack. Ishibashi [18] mentioned "To mitigate the increase in the number of queries, the operators of the DNS cache servers intercepted the negative answers from the authoritative server, and returned a blackhole IP address to the users. The blackhole address is an address that is not assigned to any users, and was configured to be dropped at the edge router."

The negative side of this countermeasure is that source addresses are being blocked which means that genuine users could be blocked as well, and for the specific case of DDoS DNS laundering blocking the source IP address of the attack is not a viable solution without blocking genuine traffic.

2.4 Traffic filtering

Traffic filtering systems has been developed by Rizvi [19], which as approach to mitigate DDoS attacks, a system of constant observation of the DNS server to then apply a mitigation filter depending on the behaviour of the DNS server resources with a system called DDiDD that stands for DDoS defense in Depth. Rizvi [19] mentioned "The full DDiDD automatically chooses the best filter or combination of filters for each attack, always achieving 93 percent or higher controlled load and at most 1.7 percent collateral damage." On the other hand, a DNS filter was implemented by Datta [20] where an embedded devices is constantly sniffing DNS packets looking if the traffic overpass the rate limit established to then apply a filter and ban permanently the source of the DNS packets that overwhelm the network. In both cases a filter is applied which blocks the source of the traffic, which in the case of DDoS DNS laundering is non-viable without restrict the access to the targeted domain by genuine users, since on DDoS DNS laundering attacks the malicious traffic is coming from a valid recursive DNS resolver.

2.5 Summary of previous works

Analyzing previous works we can see that effective methods against some kind of DDoS attacks has been found in the past. However, facing this specific type of attack, those methods of mitigation are not robust enough to neutralize DNS laundering DDoS attacks. On figure 4 we can see clearly previous mentioned methods against the proposed method.

Method vs Aspect	Detect DDoS DNS laundering attack	Block source of the attack	Allow genuine traffic to go through	Server´s resources are being protected from the attack	Papers that proposed those approaches
Rate limit	No	No	Yes	No	Wong[14], Granger[15], and Rozekrans[16].
Black-hole	Yes	Yes	No	Yes	Jonker[17] and Ishibashi [18].
Traffic filtering	Yes	Yes	No	Yes	Rizvi[19] and Datta [20].
Hasegawa solution	Yes	Yes	No	Yes	Hasegawa[12]
Proposed solution	Yes	Yes	Yes	Yes	

Figure 4: Comparisons of previous solutions and proposed solutions under DNS laundering DDoS attack.

As we can see on figure4, the main aspect of the proposed solution compared to black-hole and traffic filtering methods is that the proposed solution let genuine traffic go through during the DDoS DNS laundering attack and the difference between the proposed solution and rate limit method is that this last one does not block the source of the attack. Also, the difference of the Hasegawa solution against the proposed method is the allowance of legit users to reach the victim´s domain.

3 Methodology

The proposed solution offers a comprehensive DNS traffic monitoring in order to achieve the goal of block the source of the attack meanwhile keeps the targeted domain reachable by legit users.

NXDOMAIN stand for Non-Existent Domain, and it is one the error responses given by DNS servers to indicate that the pointed domain does not exist [21]. In this paper I am using this type of response packets to identify if the authoritative DNS server is under attack by implementing a DNS traffic controller that acts as a man-in-the-middle, as previously implemented by [22].

The methodology implemented on this paper can be described in four phases: DNS traffic monitoring, DNS laundering DDoS attack detection, blocking attack, and keep targeted domain accessible by legit users.

The first stage, DNS traffic monitoring is given by the implementation of the previous mentioned DNS traffic controller, which, with the execution of an algorithm developed for this particular task, first the algorithm is detecting if the authoritative DNS server is under attack by measuring the NXDOMAIN responses from the authoritative DNS server to the recursive DNS resolver server. To perform this task a limitation for NXDOMAIN packets throughput has been set. By this way The traffic that goes through authoritative DNS server and recursive DNS resolver is being constantly analysed to discriminate DNS traffic. In order to intercept traffic and be able to drop it if necessary, since man-in-the-middle devices has the capacity of intercept, read and modify packets [23], the device for DNS controller that will act as a man in the middle device would remain on passive mode sniffing to detect NXDOMAIN packets [24] [25].

The second stage DNS laundering DDoS attack detection, is given when the limit on throughput for NXDOMAIN packets has been over passed, once this point is reached means that the authoritative DNS server is under DNS laundering DDoS attack.

The third stage blocking attack, is given by the algorithm of the DNS traffic controller stopping the forwarding of the traffic between both DNS servers, once reached this point, the attack is been blocked and the authoritative DNS server is not more under attack.

The last stage keep targeted domain accessible by legit users, is given by taking advantage of the recursive DNS resolvers' cache where they save the responses for a period of time [26]. In order to keep genuine traffic go through the algorithm is keeping the targeted domain alive on the recursive DNS resolver server by sending DNS requests, allowing by this way the genuine traffic to reach the victim´s domain.

— As part of the proposed solution, I am taking advantage of the recursive DNS resolvers' cache where they save the responses for a period of time [26]. The methodology proposed implemented two scripts written on python that ideally should be run on the nearest router to the authoritative DNS server.

In order to keep genuine traffic go through the algorithm is keeping the targeted domain alive on the recursive DNS resolver server by sending DNS requests, allowing by this way the genuine traffic to reach the victim´s domain. This process allows the malicious traffic to reach the targeted domain for a short period of time (for this work that configuration is taken to three seconds), this with the goal of placing DNS entrances on the recursive DNS resolver server.

After the initial detection of the attack, the legit traffic has always access to the targeted domain, since DNS resolver has all the time the entrance for the targeted domain in its cache.

4 Design Specification

The proposed solution is given by two phases, the first one is given by implementing DNS traffic controller device that acts as a man-in-the-middle between recursive DNS resolver and authoritative DNS server, which implements an algorithm that is performing a constant scanning of the network traffic between recursive DNS resolver server and authoritative DNS server to measure the throughput of NXDOMAIN packets sent from authoritative DNS server. When the throughput exceeds the limit, set as 1 Kilo bits for test purposes, a traffic filter is implemented, this filter drops every DNS packet but allows the rest of the traffic going through.

The second phase consists of keeping the targeted domain from authoritative DNS server alive on recursive DNS resolver´s cache. The algorithm accomplishes this task sending periodically DNS requests pointing the targeted domain to the recursive DNS resolver. To carry out this task the traffic between the DNS servers gets allowed by a short period of time, set on 10 seconds for test purposes, during that period, the algorithm performs a scanning of the traffic to measure the NXDOMAIN packets again and implement the filter if the throughput exceeds the limit.

This entire process is being in a loop during the attack since once the second phase is done, and an exceeded throughput of NXDOMAIN packets is detected, the algorithm goes back to the phase number 1. By implementing this and after the first time the second phase is implemented, the targeted domain of the attack is always accessible by legit users through the recursive DNS resolver.

I have created a topology to develop the proposed solution where the DNS traffic controller is placed between the recursive DNS resolver and the authoritative DNS server. This device can sniff the traffic that comes from both DNS servers and act like a man-in-the-middle, by this way the device has the capacity of stop the DNS traffic between both servers [27].

4.1 Step by step on DNS laundering DDoS attack proposed solution

In this section I will explain the details of a DNS laundering DDoS attack proposed solution as a way to have a better understanding of the steps taken to mitigate the DNS laundering DDoS attack.

Step 1: The attacker sends randomized subdomain queries to the recursive DNS resolver as seen on figure 5.

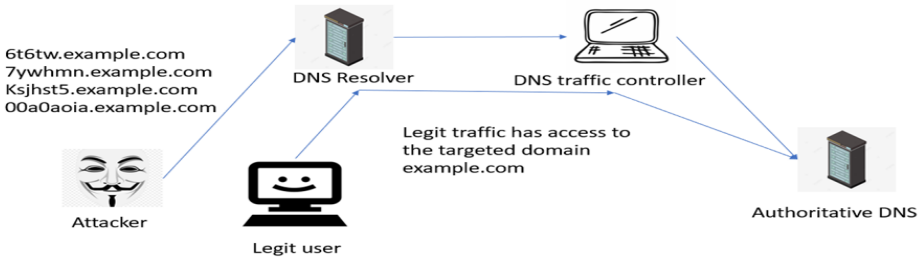


Figure 5: Attacker send queries to DNS resolver.

Step 2: The DNS resolver does not have any cached entrance and forward the queries to the DNS authoritative server as seen on figure 6.

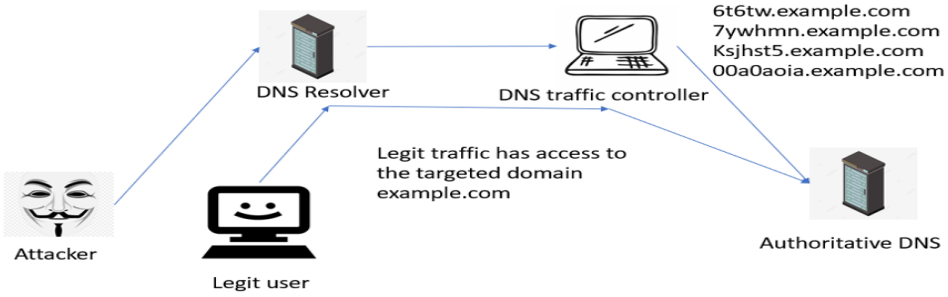


Figure 6: DNS resolver forward the queries.

Step 3: The authoritative server does not have the subdomain requested by the queries, then it answers with NXDOMAIN packets to the recursive DNS resolver as seen on figure 7.

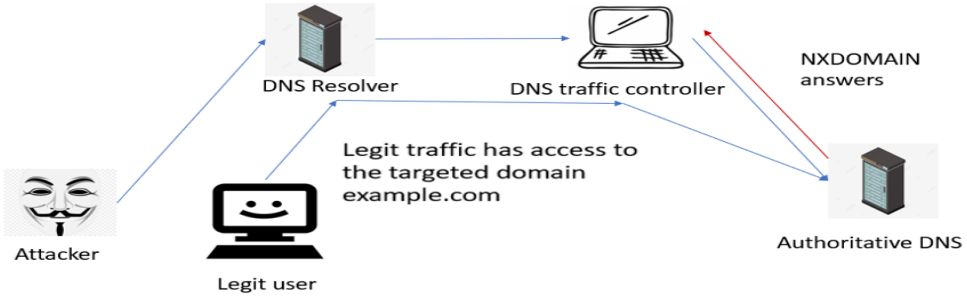


Figure 7: The authoritative server answers with NXDOMAIN packets.

Step 4: The DNS traffic controller detects overflow of NXDOMAIN packets and stop the DNS communication between Authoritative DNS server and DNS resolver as seen on figure 8.

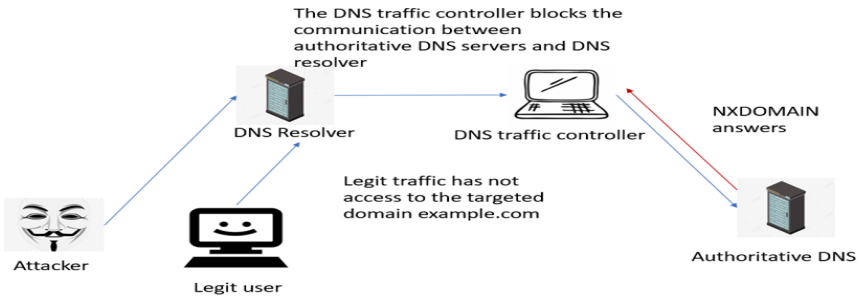


Figure 8: DNS traffic controller detects over flow of NXDOMAIN traffic

.

At this point the legit user has no access to the targeted domain “example.com” since the communication between the authoritative DNS server and DNS resolvers is blocked as seen on figure 9.

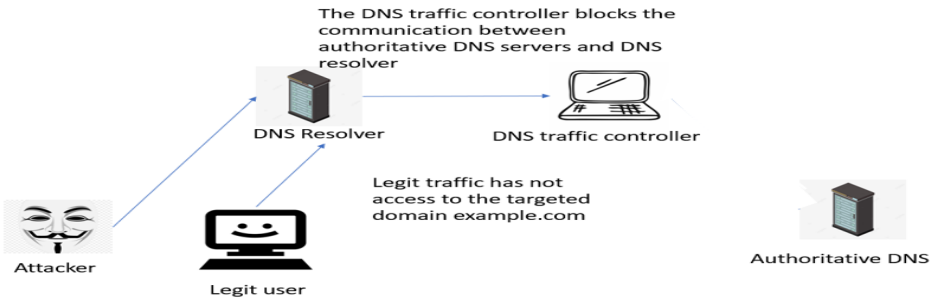


Figure 9: DNS traffic controller blocks DNS traffic

.

It is important to point that this is the only moment when the legit user has no access to the targeted domain, and this only happens the first round of the entire process.

Step 5: The DNS traffic controller keeps the DNS resolver´s cache updated by sending DNS request as seen on figure 10.

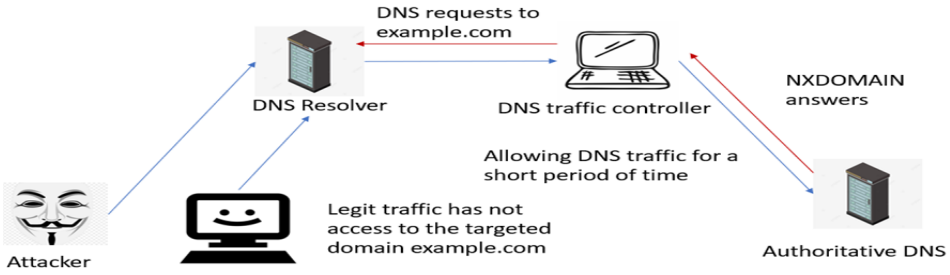


Figure 10: DNS traffic controller keeps the DNS resolver´s cache updated

.

Step 6: DNS traffic controller allows DNS traffic to go through for a short period of time to allow the DNS resolver to reach the authoritative DNS server. Also, this short period of time the DNS packets are going through is used to measure the throughput of the NXDOMAIN traffic and verify if the blocking of the DNS traffic is still necessary as seen on figure 11.

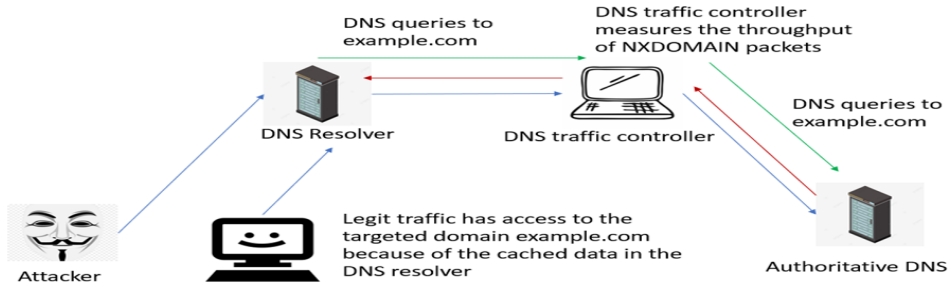


Figure 11: DNS traffic controller allows DNS traffic to go through

.

Step 7: DNS traffic controller would execute the same task of blocking the DNS traffic between authoritative DNS server and DNS resolver in case the throughput of NXDOMAIN traffic is high, with the only difference that the legit traffic would have always access to the targeted domain because of the cached entries on the DNS resolver as seen on figure 12.

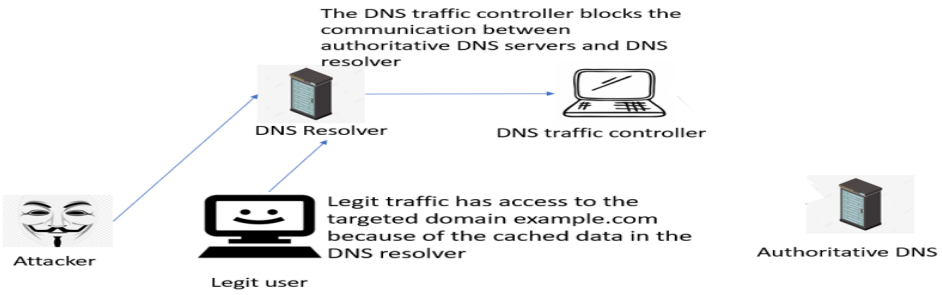


Figure 12: DNS traffic controller execute task of blocking the DNS traffic

.

Step 8: The DNS traffic controller would do the task of measure the NXDOMAIN traffic and keep the DNS resolver´s cache entrance alive for the targeted domain again. This would be a cycle until the attack finish and the DNS traffic controller let the DNS traffic go through as seen on figure 13.

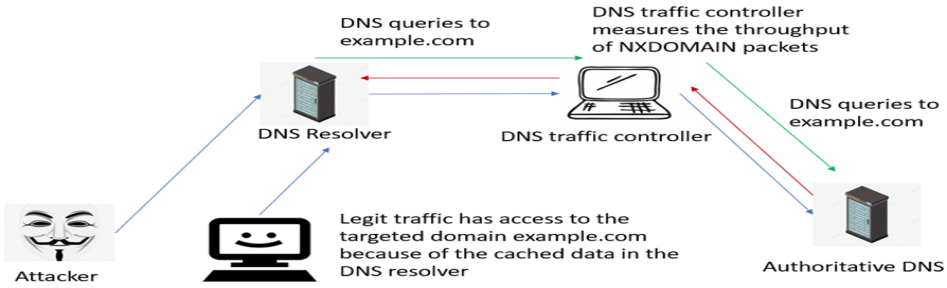


Figure 13: The DNS traffic controller measures the NXDOMAIN traffic and keep alive the cache of the recursive DNS resolver alive for the targeted domain

.

After the initial detection of the attack, the legit traffic has always access to the target “example.com” since DNS resolver has all the time the entrance for the targeted domain in its cache.

5 Implementation

In this section I am going to discuss the implementations I have done to perform in first instance the execution of an attack carried out with the same characteristics of DNS laundering DDoS attack. Also, the simulation of an environment properly created capable to emulate a DNS laundering DDoS attack and the proposed solution in an accurate way.

I have implemented a simulation using virtual machines based on Kali. I have chosen Kali due the tools that are already are installed in. Also, since Kali is capable to run

in low system requirements due the amount of virtual machines it is needed to run the simulation [28] [29].

The architecture diagram of the implementation it is seen on the figure 14. As the topology shows on the implementation made, the DNS traffic controller can control the communication between the two DNS servers, that means that even when the DNS traffic controller has blocked the traffic, DNS servers has still access to the router.

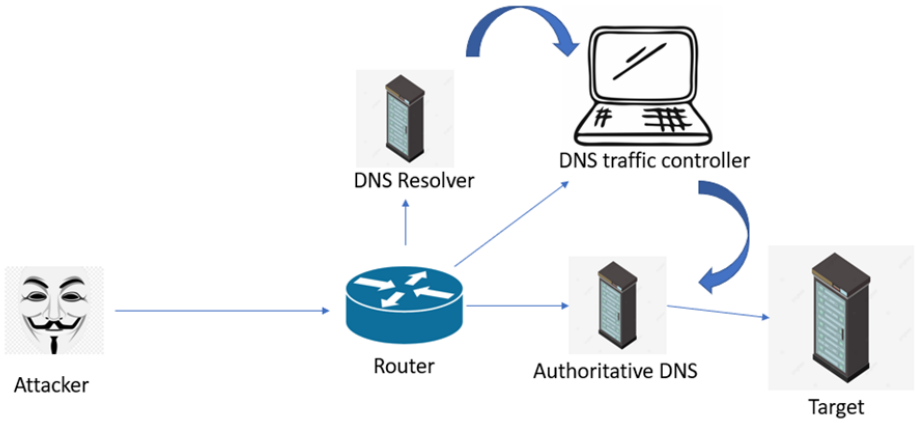
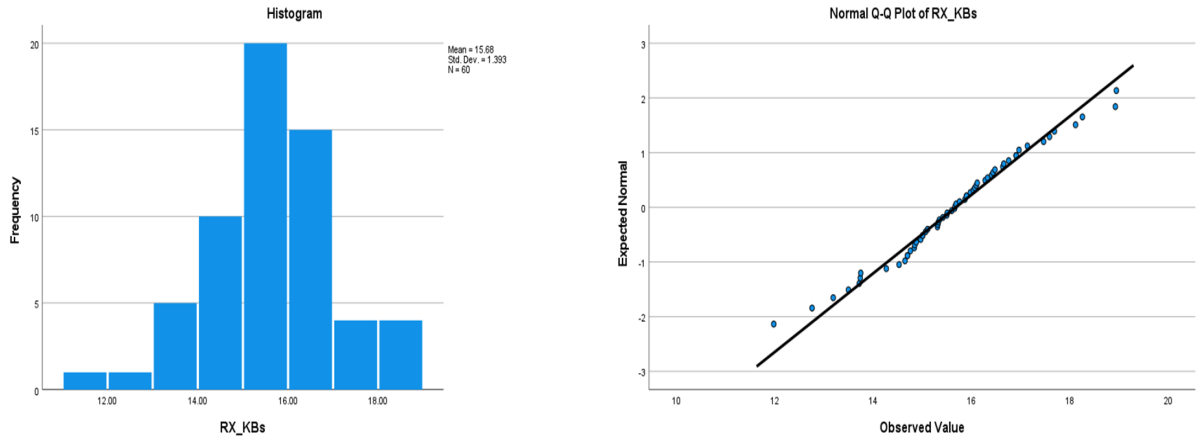


Figure 14: Topology of the proposed solution

I have firstly implemented a DNS laundering DDoS attack without accomplish any defense in order to compare the results of the attack to the proposed solution, black-hole solution, and rate limit solution

I have analysed the results of the attack with SPSS data analyser software, I have chosen this software due its reliability and effective data management.

Data coming from the data analyser software regarding DNS laundering attack with a load of 15KB/s without any defense are shown on the figure 15



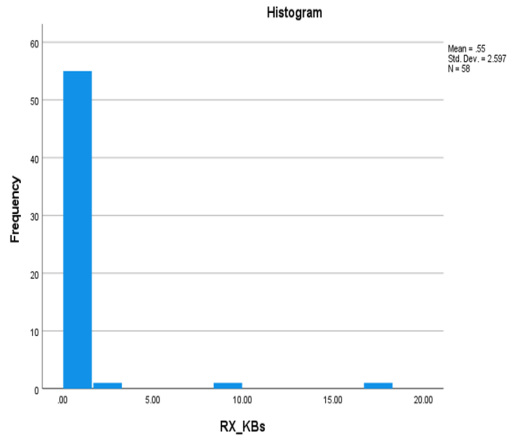
(a) 15KB/s normal load distribution

(b) 15KB/s normal plot load distribution

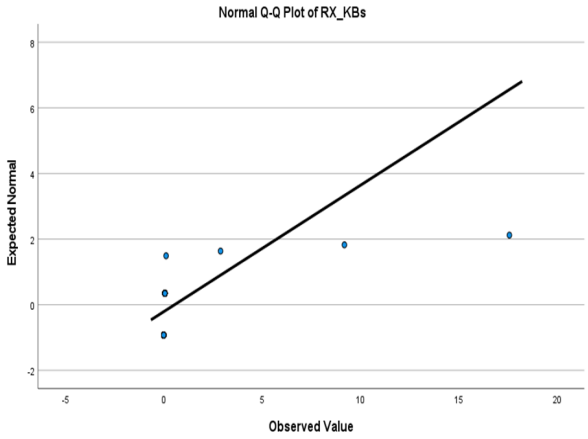
Figure 15: 15KB/s attack normal distribution

on the figure 15 it is shown that the mean load is 15.88KB/s with a standard deviation of 1.393 taking in account 60 samples.

I have analysed the results given by SPSS software after the implementation of the proposed defense when a load of 15KB/s is applied. The results from the software are shown in the figure 16



(a) 15KB/s attack when proposed solution normal load distribution

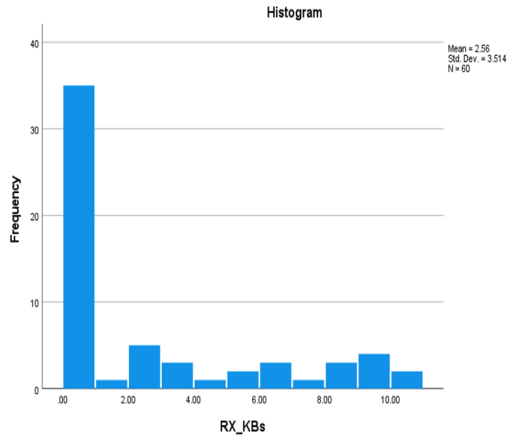


(b) 15KB/s attack when proposed solution normal plot load distribution

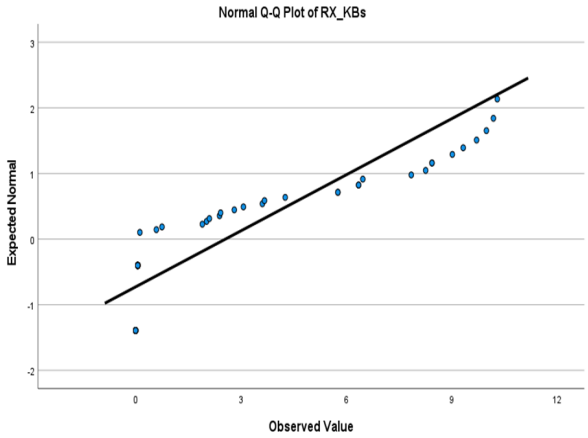
Figure 16: 15KB/s attack when proposed solution normal distribution

As can be seen on the figure 16, the values for DNS laundering DDoS attack are not properly distributed since the majority of the values are set on zero with some exception peaks.

For black-hole solution, the incoming traffic presented on authoritative DNS server analysed with SPSS software is given by the next figure 17.



(a) 15KB/s attack when black-hole solution normal load distribution

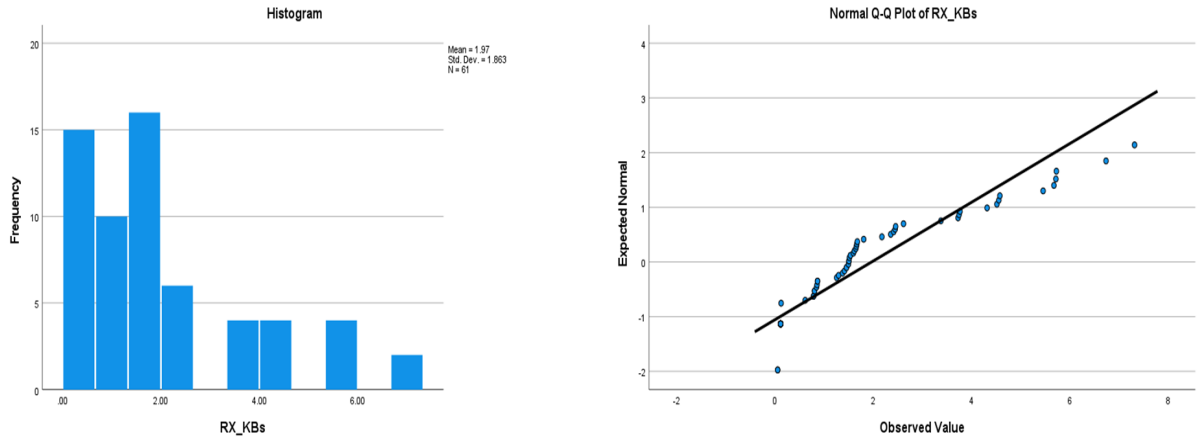


(b) 15KB/s attack when black-hole solution normal plot load distribution

Figure 17: 15KB/s attack when black-hole solution normal distribution

As seen on the last figure 17, the incoming traffic after applying black-hole solution is no normally distributed since the majority of the samples are set to zero, but we can observe some data that is still present, since the majority of the samples are zero this represents a drastic reduction of the incoming data due authoritative DNS server is not responding the queries from the attack.

For rate limit solution, the incoming traffic presented on authoritative DNS server analysed with SPSS software is given by the next figure 18 As seen on the figure 18, the



(a) 15KB/s attack when rate limit solution normal load distribution

(b) 15KB/s attack when rate limit solution normal plot load distribution

Figure 18: 15KB/s attack when rate limit solution normal distribution

incoming traffic after applying rate limit solution goes to low rates making the majority of the samples low values. This means that the incoming traffic from the attack is still present but in low rate values.

6 Experimental evaluation

In this section I am going to navigate through the results of the simulation where the authoritative DNS server that servers the targeted domain is facing a DNS laundering attack, In the simulations I got results from the behaviour of the DDoS attack, the authoritative DNS server and last but not least, the behaviour of queries from legitimate users.

It is important to mention that an extrapolation should be made to fit the results given by this study to real world DDoS attacks, since any simulation that can be performed by a study can extrapolate the data results to "predict peak traffic based on self-similar traffic" [30].

On the third quarter of 2023 a new record on DDoS attack request has been reached with 71 million requests per second [31], the largest amount of requests per second reached on this study is 203, then an extrapolation need to be done to fit this results on a real world scenario. However in this paper it is studied the effectiveness of the proposed solution to mitigate a DNS laundering attack meanwhile legit users still have access to the targeted domain and the main advantages and disadvantages against different mitigation methods.

The results are divided in sections depending on the mitigation technique, as previously mentioned, those techniques are black-hole DDoS mitigation, Rate limit mitigation, and the proposed solution.

The results are based on 4 factors: number packets received by the authoritative DNS server, load in Kilo Bytes per second that those packets represent, memory utilization and swap memory utilization. The CPU did not present relevant utilization during the attack or either during the mitigation process.

It is considered that the authoritative DNS server has a total RAM memory of 752 Mega Bytes and a swap memory of 1023 Mega Bytes, where in quiet conditions use 530MB from memory and 70MB from swap.

6.1 Behaviour of the targeted authoritative DNS server without any mitigation technique applied

I have performed a DNS laundering DDoS attack with three different loads: 5KB/s, 10 KB/s and 15KB/s. The figure 19 illustrates the main aspect that is affected on authoritative server when is under attack, memory compared to the load of the attack and also the peaks of the load.

	Peaks of load in KB/s	Difference in memory in percentage compared to not under attack state
5KB/s	9	13.20
10KB/s	6	14.52
15KB/s	8	13.4

Figure 19: Comparisons table for DNS laundering attack

Swap memory used 140MB representing an increment of 100 percent compared to the conditions of the server when is not under attack. this result was given on the three load scenarios.

6.2 Behaviour of the targeted authoritative DNS server when the proposed solution is applied

The results of the attack over the targeted server are shown on the figure 20

	Mean load in KB/s	Peaks of load in KB/s	Difference in percentage of load received compared to attacker load
5KB/s	0.02	0.08	-99.6
10KB/s	0.03	0.08	-99.7
15KB/s	0.025	0.08	-99.83

Figure 20: Comparisons table for proposed solution results

It is observed on the figure 20 that the load traffic decreased considerably compared to the original load of the attack

However is been observed that the low rate traffic detected is control traffic from the protocol ARP, which is used in networking to associate IP addresses to MAC addresses on every device.Also, during the three load scenarios the legit traffic could reach the targeted authoritative DNS server successfully.

In all scenarios the memory utilization was placed on 533MB during this attack, this represents a decrement of 11.6 percent compared to the mean of memory conditions of the server when is under attack. The swap memory was placed on 76 which represents an decrement of 45.71 percent compared to the mean of memory conditions of the server when is not under attack.

On the figure 21 we can observe the results for memory utilization for the proposed solution on the simulation done.

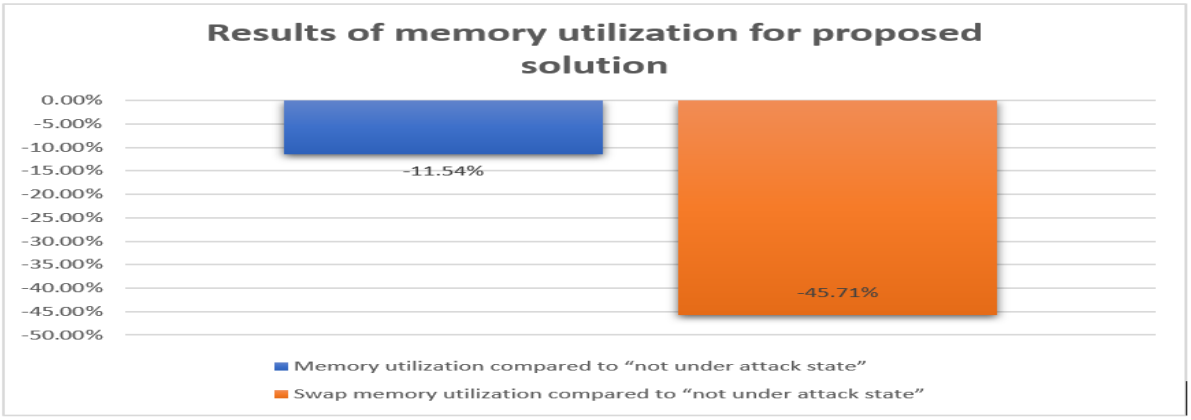


Figure 21: Results of memory utilization for proposed solution

The figure 21 shows that the memory decreased by 11.6 percent meanwhile the swap memory decreased in utilization by 45.71 percent compared to the mean of memory conditions of the server when is not under attack.

On the figure 22 we can observe the percentage of load received from the DNS laundering attack on the authoritative DNS server.

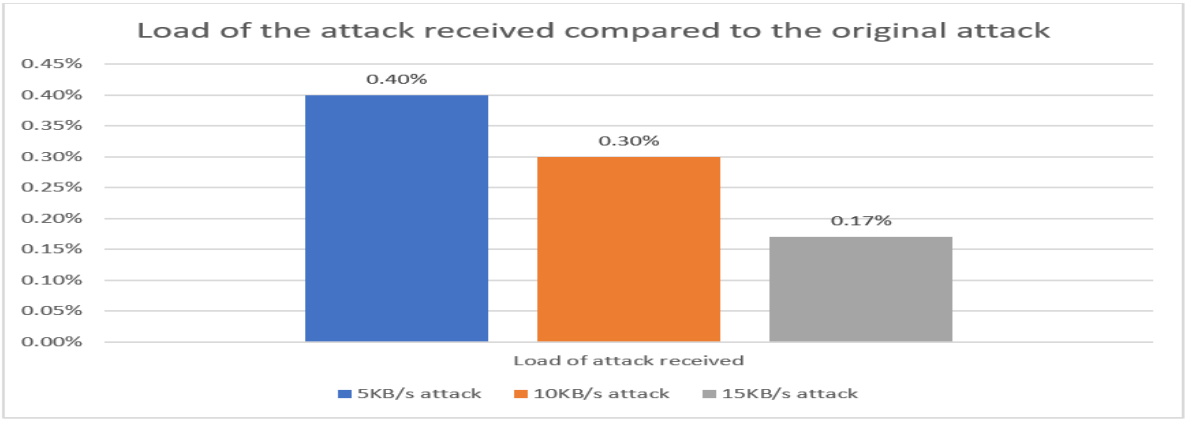


Figure 22: Results of simulation for proposed solution

As we can see on the the figure 22, the load received from the DNS laundering attack became almost zero, then we can say that the proposed solution for DNS laundering attack was successful in prevent that a big quantity of data reach the authoritative DNS server.

6.3 Behaviour of the targeted authoritative DNS server when the black-hole mitigation is applied

During the DNS laundering DDoS attack with the black-hole as method of mitigation was observed that the quantity of load received was reduced significantly, as shown on the figure 23.

	Mean load in KB/s	Peaks of load in KB/s	Difference in percentage of load received compared to attacker load
5KB/s	2.3	8	-54
10KB/s	2.4	10	-74.5
15KB/s	2.55	10	-83.66

Figure 23: Comparisons table for proposed solution results

The memory utilization for the three load scenarios was placed on 668MB, this represents an increment of 10.85 percent compared to the mean of memory conditions of the server when is under attack. Also, the swap memory was placed on 413 which represents an increment of 195 percent compared to mean of memory conditions of the server when is under attack.

On the figure 24 we can observe the results for memory utilization for black-hole solution on the simulation done.

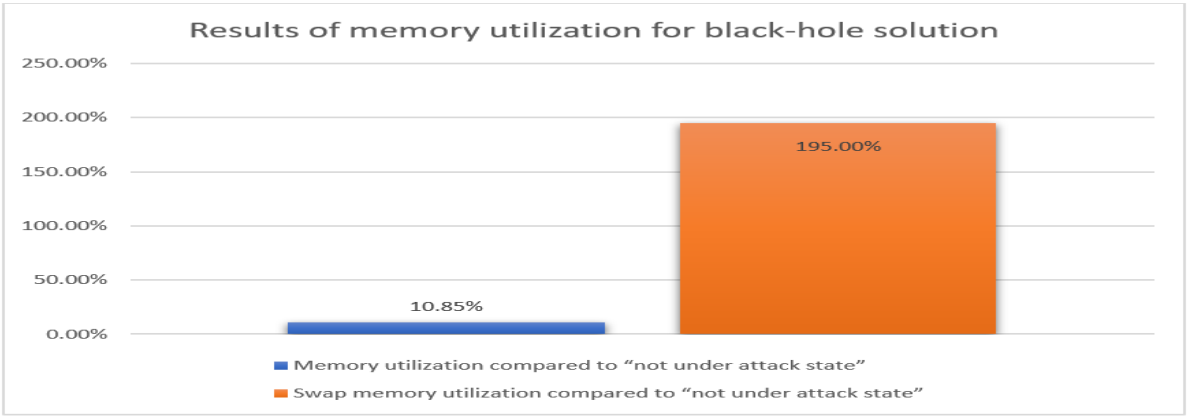


Figure 24: Results of memory utilization for black-hole solution

On figure 24 we can observe that the memory increased for the three scenarios by 10.85 percent, and finally, it is shown that the swap memory utilized increased in a considerable way, almost 3 times compared to the state of this swap memory when is under attack.

On the figure 25 we can observe the percentage of load received from the DNS laundering attack on the authoritative DNS server with a black-hole defence.

As we can see on the figure 25, the load from the attack was significantly reduced using a black-hole solution. However an important amount of load is still being received.

6.4 Behaviour of the targeted authoritative DNS server when the rate limit mitigation is applied

During the DNS laundering DDoS attack with the rate limit as method of mitigation was observed that the load received was reduced significantly, as shown on the figure 26

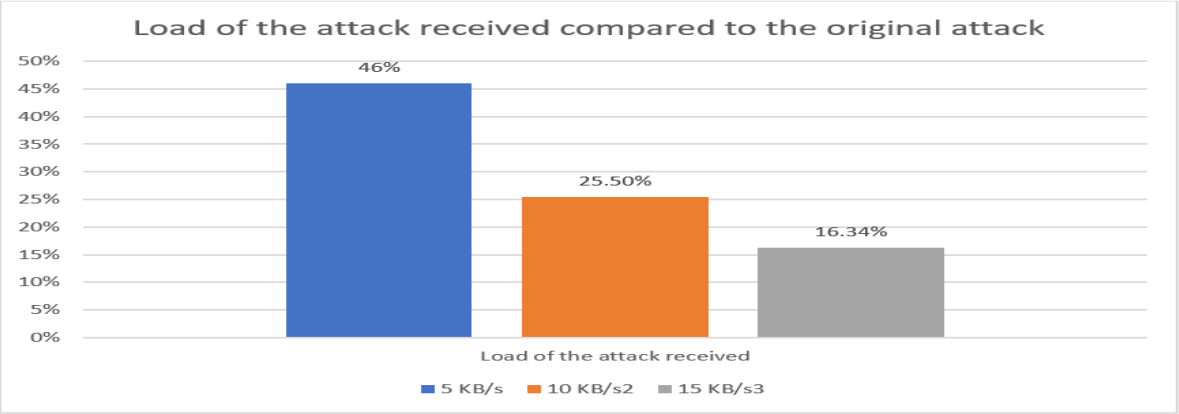


Figure 25: Comparisons of load received against load from the attack in percentage for black-hole solution

	Mean load in KB/s	Peaks of load in KB/s	Difference in percentage of load received compared to attacker load
5KB/s	2.38	8	-52.4
10KB/s	2.41	6	-75.9
15KB/s	1.97	8	-86.9

Figure 26: Comparisons table for rate limit solution results

The memory utilization for the three load scenarios was placed on 588MB, this represents an decrement of 2.42 percent compared to the mean of memory conditions of the server when is under attack. Also, the swap memory was placed on 394 which represents an increment of 181.42 percent compared to the conditions of the server when is under attack.

On the figure 27 we can observe the results for memory utilization for rate limit solution on the simulation done.

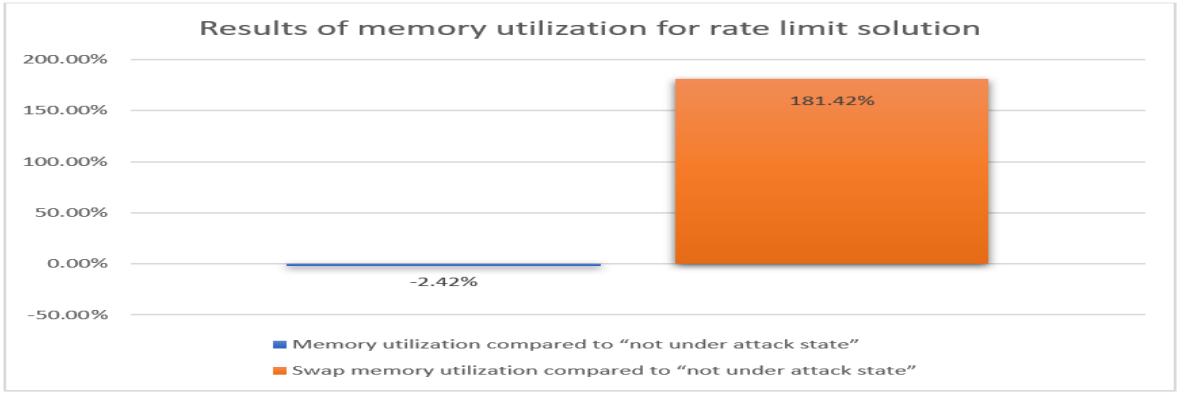


Figure 27: Results of memory utilization for rate limit solution in percentage

On the figure 27 we can observe that the memory decreased for the three scenarios by 2.42 percent compared to the mean of memory conditions of the server when is under attack, and finally, it is shown that the swap memory utilized increased in a considerable way, 2.8 times compared to the state of this swap memory when is under attack.

On the figure 28 we can observe the percentage of load received from the DNS laundering attack on the authoritative DNS server with a rate limit defence.

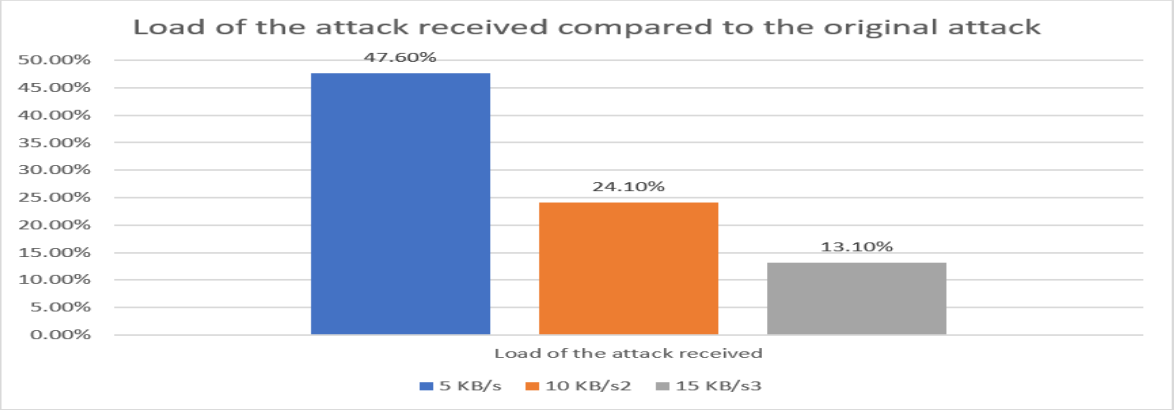


Figure 28: Comparisons of load received against load from the attack in percentage

As we can see on the last figure 28, the load from the attack was significantly reduced to levels near to the 50 percent in the case of the load of 5KB/s, and 10 percent in the case of the load of 15KB/s.

6.5 Discussion

Taking in account the results given by the simulation, as the load received on the authoritative DNS server side in comparison to the load generated by the attacker, memory utilization, and intrinsic characteristics of the nature of the different methods implemented as mitigators, I have developed the table 29 to have a better and clear understanding of the advantages and disadvantages of the methods compared.

Proposed defence / Characteristic	Black-hole	Rate limit	Proposed defence
Legit users can go through	NO	YES	YES
Source of the attack is blocked	PARTIALLY	NO	YES
Server resources get compromised (Memory)	PARTIALLY	NO	NO
Server resources get compromised (Swap memory)	YES	YES	NO
Avoid responding back the malicious queries	YES	NO	YES

Figure 29: Characteristics of the mitigation methods following the results from the simulation

As we can see on the table 29, black-hole solution blocks in a partial way the source of the attack, since even when dropping the queries received, this method still lets the queries to reach the authoritative DNS server, for this method, legit users are not allowed to go through while the attack is in process. On the other hand rate limit lets the legit users to go through but keeps the source the attack unblocked and keep responding the queries coming from the attacker, this is the only mitigation method that responds the malicious queries. Last but not least we have the proposed solution that let the legit traffic go through meanwhile blocks the source of the attack and does not respond the malicious queries.

For the server resources section from the table, we can observe that black-hole solution gives the worse approach in this section, since the swap memory presents a noticeable increment compared to the swap utilization when under attack, meanwhile the memory presents an increment as well but not so relevant, only of 10.85 percent. Rate limit solution on the other hand, presents even a decrement on memory utilization, small but still a decrement of 2.42 percent, but also presenting an increment of 2.8 times on swap memory utilization. Finally, the proposed solution has presented an important decrement on both memory and swap elements, when the first one is about 11.45 percent, the second one comes to 45.71 percent.

With the results given by the simulation, I can assume the proposed solution had a successful approach on mitigating DNS laundering DDoS attacks, since it blocked the source of the attack and let the legit traffic go through. Also was been observed that the proposed solution is more effective against DNS laundering DDoS attacks due the targeted DNS server did not even receive malicious queries because the DNS traffic controller blocked them before they could be able to find its way to the targeted server.

7 Conclusion and Future Work

I can conclude that the proposed solution is effective in detecting and mitigating DNS laundering DDoS attacks while letting genuine traffic reach the targeted domain. Comparing the proposed solution to the other two methods analysed in this paper, I assume it is a complete but narrowed solution to this specific type of attack. The principal contribution of the proposed solution is that let the legit users access to the victim´s domain through DNS resolver´s cache, which is being updated every minute to maintain the victim´s domain active.

Limitations of this work are given by the resources of the machine where the simulations is running, since it is necessary to run multiple virtual machines, resources of the host machine are quickly exhausted. For this same reason, assumptions has been made on results, since the data obtained need to be extrapolated to fit a real-world attack. Also, another limitation is given by the fact that I am using only one virtual machine that is performing the duty of attack, since in real world this task might be given by hundreds of machines.

For future works a good approach will be implementing this solution mounted in the traffic controller this time to a router, by this way the solution becomes more compacted and elegant, since the implementation on routers would not demand the implementation of an external device. Also, future simulations of experiments could be done implementing this model by increment the load and the number o attacker machines.

References

[1] M. Mittal, K. Kumar, and S. Behal, "Deep learning approaches for detecting ddos attacks: a systematic review," *Soft Computing*, vol. 27, p. 13039–13075, 2023.

[2] Y. Cui, Q. Qian, C. Guo, G. Shen, Y. Tian, H. Xing, and L. Yan, "Towards ddos detection mechanisms in software-defined networking," *Journal of Network and Computer Applications*, vol. 190, p. 103156, 2021.

[3] Netscout, "Ddos threat intelligence report," 2023. [Online]. Available: https://www.netscout.com/threatreport/internet-traffic-slipstreamed-threats

[4] O. van der Toorn, M. Müller, S. Dickinson, C. Hesselman, A. Sperotto, and R. van Rijswijk-Deij, "Addressing the challenges of modern dns a comprehensive tutorial," *Computer Science Review*, vol. 45, 2022.

[5] Ömer KASIM, "A robust dns flood attack detection with a hybrid deeper learning model," *Computers and Electrical Engineering*, vol. 100, 2022.

[6] E. Sagatov, S. Mayhoub, A. Sukhov, and P. Calyam, "Countering dns amplification attacks based on analysis of outgoing traffic," *Journal of Communications and Information Networks*, vol. 8, no. 2, pp. 111–121, 2023.

[7] J. P. Omer Yoachimik, "Ddos threat report for 2023 q2," 2023. [Online]. Available: https://blog.cloudflare.com/ddos-threat-report-2023-q2

[8] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, and D. Dagon, "From {Throw-Away} traffic to bots: Detecting the rise of {DGA-Based} malware," in *21st USENIX Security Symposium (USENIX Security 12)*, 2012, pp. 491–506.

[9] R. Sommese, K. Claffy, R. van Rijswijk-Deij, A. Chattopadhyay, A. Dainotti, A. Sperotto, and M. Jonker, "Investigating the impact of ddos attacks on dns infrastructure," in *Proceedings of the 22nd ACM Internet Measurement Conference*, ser. IMC '22. Association for Computing Machinery, 2022, p. 51–64.

[10] A. K. Soliman, C. Salama, and H. K. Mohamed, "Detecting dns reflection amplification ddos attack originating from the cloud," in *2018 13th International Conference on Computer Engineering and Systems (ICCES)*, 2018, pp. 145–150.

[11] S. Saharan and V. Gupta, "Prevention and mitigation of dns based ddos attacks in sdn environment," in *2019 11th International Conference on Communication Systems and Networks (COMSNETS)*, 2019, pp. 571–573.

[12] K. Hasegawa, D. Kondo, and H. Tode, "Fqdn-based whitelist filter on a dns cache server against the dns water torture attack," in *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 2021, pp. 628–632.

[13] C. Deccio, D. Argueta, and J. Demke, "A quantitative study of the deployment of dns rate limiting," in *2019 International Conference on Computing, Networking and Communications (ICNC)*, 2019, pp. 442–447.

[14] C. Wong, S. Bielski, A. Studer, and C. Wang, "Empirical analysis of rate limiting mechanisms," in *Recent Advances in Intrusion Detection*, A. Valdes and D. Zamboni, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 22–42.

[15] G. R. Ganger, G. Economou, and S. Bielski, "Self-securing network interfaces: What, why and how (cmu-cs-02-144)," in *Carnegie Mellon University. Journal contribution.*, 2002.

[16] T. Rozekrans and K. d. Javy, "Defending against dns reflection amplification attacks," University of Amsterdam, Tech. Rep., 2013.

[17] M. Jonker, A. Pras, A. Dainotti, and A. Sperotto, "A first joint look at dos attacks and bgp blackholing in the wild," in *Proceedings of the Internet Measurement Conference 2018*, ser. IMC '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 457–463.

[18] K. Ishibashi, T. Toyono, H. Matsuoka, K. Toyama, M. Ishino, C. Yoshimura, T. Ozaki, Y. Sakamoto, and I. Mizukoshi, "Measurement of dns traffic caused by ddos attacks," in *2005 Symposium on Applications and the Internet Workshops (SAINT 2005 Workshops)*, 2005, pp. 118–121.

[19] A. S. M. Rizvi, J. Mirkovic, J. Heidemann, W. Hardaker, and R. Story, "Defending root dns servers against ddos using layered defenses," in *2023 15th International Conference on COMmunication Systems & NETworkS (COMSNETS)*, 2023, pp. 513–521.

[20] S. Datta, A. Kotha, K. Manohar, and U. Venkanna, "Dnsguard: A raspberry pi-based ddos mitigation on dns server in iot networks," *IEEE Networking Letters*, vol. 4, no. 4, pp. 212–216, 2022.

[21] Y. Iuchi, Y. Jin, H. Ichise, K. Iida, and Y. Takai, "Detection and blocking of dga-based bot infected computers by monitoring nxdomain responses," in *2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, 2020, pp. 82–87.

[22] H. Choi, H. Lee, H. Lee, and H. Kim, "Botnet detection by monitoring group activities in dns traffic," in *7th IEEE International Conference on Computer and Information Technology (CIT 2007)*, 2007, pp. 715–720.

[23] G. Liu, L. Jin, S. Hao, Y. Zhang, D. Liu, A. Stavrou, and H. Wang, "Dial "n" for nxdomain: The scale, origin, and security implications of dns queries to non-existent domains," in *Proceedings of the 2023 ACM on Internet Measurement Conference*, ser. IMC '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 198–212.

[24] L. Hadjidemetriou, G. Tertytchny, H. Karbouj, C. Charalambous, M. K. Michael, M. Sazos, and M. Maniatakos, "Demonstration of man in the middle attack on a feeder power factor correction unit," in *2020 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*, 2020, pp. 126–130.

[25] A. Al-Hababi and S. C. Tokgoz, "Man-in-the-middle attacks to detect and identify services in encrypted network flows using machine learning," in *2020 3rd International Conference on Advanced Communication Technologies and Networking (CommNet)*, 2020, pp. 1–5.

[26] K. Man, Z. Qian, Z. Wang, X. Zheng, Y. Huang, and H. Duan, "Dns cache poisoning attack reloaded: Revolutions with side channels," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 1337–1350.

[27] D. Javeed, U. Mohammedbadamasi, C. Ndubuisi, F. Soomro, and M. Asif, "Man in the middle attacks: Analysis, motivation and prevention," *International Journal of Computer Networks and Communications Security*, 07 2020.

[28] S. Z. ul Hassan, Z. Muzaffar, and S. Z. Ahmad, "Operating systems for ethical hackers-a platform comparison of kali linux and parrot os," *International Journal*, vol. 10, no. 3, 2021.

[29] Kali.org, "Installing kali linux," 2023. [Online]. Available: https://www.kali.org/docs/installation/hard-disk-install

[30] A. B. de Neira, B. Kantarci, and M. Nogueira, "Distributed denial of service attack prediction: Challenges, open issues and opportunities," *Computer Networks*, vol. 222, p. 109553, 2023.

[31] J. P. Omer Yoachimik, "Ddos threat report for 2023 q3," 2023. [Online]. Available: https://blog.cloudflare.com/ddos-threat-report-2023-q3