



**Analysis of L4 DoS/DDoS Attacks and Mitigation
Techniques for DNS Reflection Attack**

by

Zaid Al-Jarrah

Master's Thesis 2018

Supervisor: Bahman Sarrafpour

Associate Supervisor: Dr Iman Ardekani

Abstract

Cybersecurity is a very important area that needs to be worked on and improved. Day by day technology becomes closer to human life in many ways. In recent years, especially after the emergence of IoT and cloud computing, technology has started to control a big part of our assets. These assets could be data medical assets, financial assets, etc. For example, today we see technology being involved in sensitive medical operations, so human life has become related to technology and any failure could cause risk to human life. There are many types of cyber threat and different types of cyber-attacks: this study discusses DNS Distributed Denial of Service (DDoS) attacks and focuses on DNS reflection attacks, which are one of the most common kinds of attack. These attacks depend on exploiting a DNS service that relies on User Datagram Protocol (UDP), which is one of the essential services that is working in the background to support internet services.

To be able to analyse a DNS reflection attack, I designed and built a testbed network which represented the environment that runs the attack. The testbed included Cisco routers, Cisco switches, and servers. These routers played the main role in demonstrating attack stages and factors, and by analysing the results, I built a mitigation technique to reduce or eliminate the possibility of those factors.

All the results and readings presented in this study are generated and collected by the author while creating an actual attack at lab using cisco routers, switches and servers rather than using simulation or emulation software. Using those devices make the testbed similar to a commercial environment that are exposed or targeted by real attacks. This process leads to achieve the desired practical outcomes.

The results showed that applying separate mitigation techniques in different stages and in more than one place worked perfectly to reduce attack load by using uRPF, Unicast Reverse Path Forward, is a standard security feature work to prevent spoofed packets. Using Separate technique in mitigation methodology gives very positive results without exhausting router resources, such as CPU, temperature and RAM. Then, I use Zone-based firewall, which is a cisco security feature, allow the Cisco router to behave like a Firewall, by separate the router to Zones to control incoming packets that are coming from outside the router (Internet). All stages together, work as an integrated solution. The suggested mitigation techniques are perfect for SMB organisations in terms of protecting their own network and DNS servers.

Acknowledgements

Thank you, New Zealand. Getting my Master's Degree was a dream, but today it becomes a reality.

Firstly, I would like to thank my Supervisor, Bahman Sarrafpour, and my Associate Supervisor, Dr Iman Ardekani, for their great support from the first day I enrolled in my study. They were the real light that showed me how I could finish my study in a better way; guiding me with their valuable advice and correcting me whenever I made a wrong choice – thanks a lot.

I would like to thank my parents and my wife for their support, and all my other family members who encouraged me and supported me in finishing my study.

I would also to thank all my friends in New Zealand, especially Mr Nathaniel Flick and Mrs Alice Rae-Flick for everything, plus the Red Cross team, especially Mr Dawit Arshak, who supported me and my family from the first day I started my life in New Zealand. Thank you, too, to my proofreader, Brigette Watkin.

Lastly, I would like to thank all the people that supported or encouraged me, even with only one word, who I have not mentioned above. The years of study have not been easy but all of you standing beside me have made a big difference to me achieving success.

Table of Contents

Chapter 1 Introduction	
Abstract	i
Acknowledgements	ii
Table of Contents	iii
List of Figures	vi
List of Tables	viii
List of Acronyms	ix
Chapter 1: Introduction	x
1.1 Introduction	1
1.2 Internet Services and DNS concepts	3
1.3 Open DNS Resolver	5
1.4 DoS/DDoS attack concepts	6
1.5 Related work	7
1.6 Business Impact	10
1.7 Security reports and statistics	12
1.8 Motivation	14
1.9 Research contribution	16
1.10 Thesis structure	16
1.11 Summary	17
Chapter 2: Classification of DoS/DDoS Attacks	18
2.1 DoS/DDoS terms and facts	19
2.2 L3 or Network DoS/DDoS Attack	21
2.2.1 Ping of Death DoS/DDoS attack	21
2.2.2 Smurf attack.....	21
2.3 L4 DoS/DDoS attacks	22
2.3.1 L4 SYN DoS/DDoS attack.....	22
2.3.2 L4 DoS/DDoS SYN-ACK Flood attack	23
2.3.3 NTP attack.....	23
2.4 L7 DoS/DDoS attacks	24
2.4.1 HTTP Flood attack	24
2.4.2 WordPress DoS/DDoS attack	26
2.4.3 SQL Injection Attack.....	27

2.5 Summary	30
Chapter 3: Classification of DNS DoS/DDoS Attacks.....	31
3.1 DNS Concepts and terminology	32
3.2 DNS attacks - methodology descriptions.....	34
3.2.1 Distributed Reflection DNS DoS attack	34
3.2.2 Cache Poisoning	35
3.2.3 DNS Tunnelling attack	36
3.2.4 DNS Hijacking attack	37
3.2.5 Basic NXDOMAIN attack	38
3.2.6 Domain Lock-up attack	38
3.3 Summary	40
Chapter 4: Thesis Methodology	41
4.1 Research area and thesis methodology	42
4.2 Attack Methodology representation.....	42
4.3 Research hypothesis	43
4.4 Research Methodology classifications	44
4.5 Research hypothesis test methodology	46
4.6 Research questions discussion.....	47
4.7 Research data gathering and measurements	48
4.8 Summary	48
Chapter 5: Mitigation techniques and verification process	49
5.1 Network Diagram (Testbed)	50
5.2 Monitoring tools	52
5.2.1 PRTG.....	52
5.2.2 Wireshark.....	52
5.2.3 CISCO Show Commands.....	52
5.3 Attacking tools.....	53
5.3.1 Hping3	53
5.3.2 Nping.....	53
5.4 Mitigation technique stages	53
5.4.1 Stage one - Shown in the network diagram as	54
5.4.2 Stage one verifications.....	55
5.4.3 Stage two - shown in the network diagram as	56
5.4.4 Stage two verification	58
5.4.5 Stage three - shown in network diagram as	61

5.4.6 Stage three - Verification	64
5.5 Launch Brute Force attack	67
5.6 Verification process conclusion	73
5.7 Summary	73
Chapter 6: Evaluation and Data Analysis	74
6.1 Mitigation technique mechanism and performance.....	75
6.2 Evaluate mitigation technique performance	76
6.3 Illustrate evaluation mechanism	77
6.3.1 Stage 1 Evaluation: Represented by uRPF in strict mode.....	78
6.3.2 Stage 2 Evaluation: Represented by uRPF in loose mode	80
6.3.3 Stage 3 Evaluation: Represented by Zone firewall	81
6.4 Evaluation process conclusion	83
6.5 Summary	84
Chapter 7: Discussion and Future Work.....	85
7.1 Evaluation results	86
7.2 Comparison with other solutions.....	87
7.3 Summary	89
Appendices.....	90
References	91

List of Figures

Figure (1) DNS Recursive mechanism diagram	4
Figure (2) DNS Iterative mechanism diagram	4
Figure (3) Service Provider Experience Threats	11
Figure (4) Ratio of different types of DDOS Attack	13
Figure (5) DDoS attack types in 2015 by Solarwinds	14
Figure (6) Protocols that are used in Reflection attacks	15
Figure (7) Targets of Application Layer attacks	15
Figure (8) Percentage of HTTP DoS attack types	25
Figure (9) Percentage of Application Layer attack types	26
Figure (10) SQL Injection attack idea and stages	29
Figure (11) Percentage of SQL Injection attack.....	30
Figure (12) Domain levels represent DNS hierarchical sample.....	33
Figure (13) Cache poisoning stages.....	35
Figure (14) DNS tunnel idea with illustration comments.....	36
Figure (15) DNS Hijacking attack	37
Figure (16) Attacker never completed stage 3 to establish a formal connection	39
Figure (17) DDoS reflection mechanism	42
Figure (18) The Waterfall Research Model that has been followed	45
Figure (19) Testbed diagram	51
Figure (20) A sample of NetFlow V9 configuration.....	54
Figure (21) Shows sending the attack from random IP	55
Figure (22) Sending the attack from random IP.....	56
Figure (23) Traffic OUT from the attacker router	56
Figure (24) Defender and SP router table.....	58
Figure (25) Sent packets on all sessions and dropped packets at Defender	59
Figure (26) Traffic out from attacker router	60
Figure (27) Traffic IN to the Defender.....	60
Figure (28) Traffic in to DNS server	61
Figure (29) Traffic classifications.....	62
Figure (30) Policies inside defender router.....	62
Figure (31) Zones and Zone pairs with their associated policies	63
Figure (32) Illustration of router zones and how they work.....	64
Figure (33) Sending data from an IP that has entry in the router table	65
Figure (34) All sensors receiving data	65
Figure (35) Traffic passing from ISP router to Defender router.....	66
Figure (36) All sensors receiving data except DNS server	66
Figure (37) Options are used in this attack.....	68
Figure (38) Sent 3000 packets per second and how ping reply time reacted	69
Figure (39) PRTG reading for whole installed sensors	69
Figure (40) System temperature sensor	70

Figure (41) Memory usage sensor	70
Figure (42) Power supply health sensor	70
Figure (43) Sending the same size of packet with less number of packets	71
Figure (44) Whole sensors	72
Figure (45) Matching the reply time	72
Figure (46) Diagram representing mitigation actions.....	77
Figure (47) Practical part of reading collection.....	79
Figure (48) Stage 1 Evaluation chart	79
Figure (49) Practical part of reading collection.....	80
Figure (50) Stage 2 Evaluation chart	81
Figure (51) Practical part of reading collection.....	82
Figure (52) Stage 3 Evaluation Chart.....	83

List of Tables

Table (1) How DDoS attacks impact the business (Incapsula, 2014)	12
Table (2) Devices that were been used in the lab.....	50
Table (3) Stage 1 Evaluation table	78
Table (4) Stage 2 Evaluation table	80
Table (5) Stage 3 Evaluation table	82

List of Acronyms

DoS	Denial of Service
DDoS	Distributed Denial of Service
SMB	Small and Medium Businesses
DNS	Domain Name System
TLD	Top Level Domain
tDNS	Top Domain Name System
RR	Resource Record
NXDOMAIN	Non-Existent Domain
IDS	Intrusion Detection System
CPU	Central Processing Unite
RAM	Random Access Memory
NTP	Network Time Protocol
UDP	User Datagram Protocol
SYN	Synchronisation
ACK	Acknowledgement
TCP	Transmission Control Protocol
OSI	Open System Interconnection
IT	Information Technology
ICMP	Internet Control Message Protocol
SQL	Structure Query Language
HTTP	Hypertext Transfer Protocol
WWW	World Wide Web
DNSSEC	DNS Security
CSP	Connection Service Provider
ISP	Internet Service Provider
uRPF	Unicast Reverse Path Forward
DMZ	Demilitarised Zone
SNMP	Simple Network Management Protocol

Chapter 1

Introduction

1.1 Introduction

In 1995, anyone who had a computer with a Pentium processor and Windows 95 was considered a lucky person. Later, technology gave us the option to send an email instead of sending a letter on paper - this was very useful and efficient for businesses, and a big addition to our community. With the introduction of Windows NT4 (and later Windows 2000, 2008, 2012 and today, 2016), onsite networks started to attract a lot of attention. This, however, has been succeeded by emerging online services, such as cloud and other services now produced by data centres.

Today, even very advanced networks with leading edge routers and switches are becoming a less compelling option to most SMB organisations when compared to cloud options and moving everything online, which supports the concept of pay-as-you-use. This gives organisations the option to start a service when they want and to stop it when they want, and recognises that different options suit different levels of clients, professionals, developers, and organisations. The most important point is that whatever services an organisation needs can be obtained cheaper online than if they buy their own devices and equipment and hire specialists to run and maintain the network. While this can be a great option, security is one of the biggest challenges that online services face.

The first Denial of Service (DoS) attack, conducted by Khan C. Smith in Las Vegas in 1997, caused internet service to shut down for an hour. An attack of this kind could cause huge damage by stopping legitimate users from being able to access services. According to previous experience, a DoS or Distributed Denial of Service (DDoS) attack is the first step, and in the next step the attacker might breach the victim's assets and destroy or steal sensitive information (Lanlan Pan, Xuebiao Yuchi, & Yong Chen, 2016). Therefore, cybersecurity, or internet security, is one of the most important aspects of online services today.

The Government and the private sector must take care as most government, commercial and public/private financial services are online today. Some of these online services are very obvious, but there are other services consumers use every day that they do not even notice. One example would be Domain Name System (DNS). DNS is a service that simplified the use of the internet, by using a 'friendly' name instead of using confusing numbers. DNS is responsible for converting numbers to names and names to numbers (Keyu Lu, Zhengmin Li, Zhaoxin Zhang, & Jiantao Shi, 2016). Some attackers exploit this hidden service to run different types of attacks using old and new approaches to avoid being detected by security appliances (Bassil et al., 2012).

Attackers develop their techniques day by day, so government and private agency security teams need to continuously develop their techniques to mitigate threats. There are different types of threats and they are going to become more and more sophisticated, so the responses need to follow new strategies and use of proactive methodologies to protect assets and to cover all possibilities, as discussed by Qin, Li, Shi, and Yu (2017).

Cybercrime is not just an attack to disrupt a service, it is actually a 'job' for a most attackers. According to statistics, cybercriminals gain millions, and even billions, of dollars each year. As such, they are able to invest a huge amount of money in their attacks and are prepared to do anything to get at their target.

With so much at risk, it is important to highlight a critical vulnerability for cybersecurity: human error. For example, an uneducated employee who is a little careless and makes simple mistakes or ignores security instructions, could allow attackers to pass through security lines. All of the efforts that security teams put into securing the organisation will not count for anything if an employee clicks on a link in an email that turns out to be a phishing link; visits an untrusted website which may have new malware or malicious software; or uses a memory stick from an untrusted computer. While there are many elements that need to be considered, the most vulnerable one is end-user behaviour, and security engineers and teams need to factor this into their risk management methodology.

This thesis includes seven chapters:

- Chapter one covers general ideas around cybersecurity and DNS mechanisms, statistics, motivation, and literature review.
- Chapter two discusses DoS/DDoS attack classification.
- Chapter three more specifically discusses the most common DNS DDoS attacks.
- Chapter four discusses thesis hypotheses and thesis methodology, including types of research.
- Chapter five covers the practical part and verification process through conducting a real attack on the thesis testbed.
- Chapter six discusses and conducts an evaluation process regarding different levels of attack load.
- Chapter seven discusses the evaluation results and future work.

1.2 Internet Services and DNS concepts

The Internet as an online service has a lot of support services underneath, that make it look like what we see on our computers and smartphones. One of the support services allows us to use an easy, 'friendly name' instead of numbers, which makes using the internet easier for most people. A human being has in his nature the ability to remember a friendly name much easier than random numbers. The system that is responsible for this conversion is called a Domain Name System (DNS). All internet users use it every day whenever they need to search for a website or specific information through the internet.

This system made our life easier, and there are more issues solved by using names instead of numbers. For example, let us assume that you have a server and you decide to move to another location or another country, or you need to change your service provider. With DNS name services, nothing will change for your customers who will still access your online services under the same name. However, if IP numbers were used instead of DNS, many issues would arise. For example, if you need to change place, country or Internet Service Provider (ISP), you also need to change your IP. This means advising all of your customers of the new details, resulting in a real possibility of losing customers.

As this thesis focusses on DNS attacks, this section clarifies some terms that the reader needs to know to help them understand the DNS service.

There are two main types of DNS services: Recursive DNS Service and Iterative DNS service. These two types of DNS services give us exactly the same answers and results and achieve the same aims but each one follows different mechanisms. The following section illustrates some of the main terms.

DNS server: a high-specification computer which runs DNS software; for instance, BIND, DNS Blast, and `gdnsd`. A DNS server saves a real information and database about DNS structure and DNS 'trees'. When a user sends a request about, for example, `www.google.com`, the DNS server tries to resolve the request by looking into its database for answers (which is usually looking for a record to find out what the IP address for the website is), then replies to the request based on the information that is found in the related record. If the server could not find the answer, it will try asking other servers, which will mostly be top-level domain servers. The mechanism for finding the answer will have occurred on behalf of the user until the server finds the answer and then replies - this is how Recursive DNS servers work. When an Iterative DNS server receives a query and cannot find the answer, the server will reply to the query with a referral to a higher tier server and does not ask on behalf of the client. The user does not realise all of the queries and replies that are happening in the background - all that the client knows is he gets

the website or information that he is searching for. Figures (1) and (2) illustrate both types of DNS mechanism.

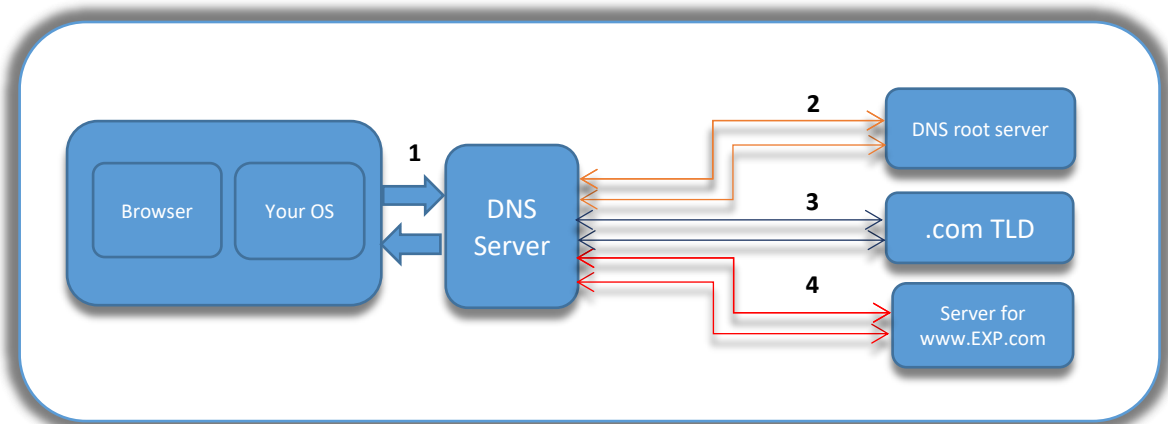


Figure (1) DNS Recursive mechanism diagram

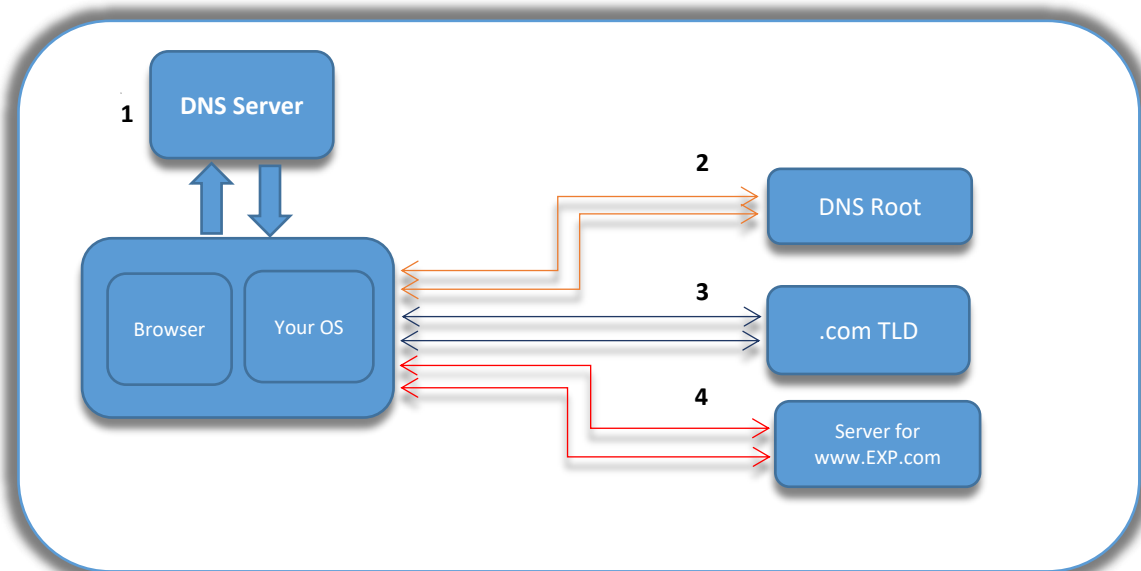


Figure (2) DNS Iterative mechanism diagram

DNS resolver: a kind of software or computer that has special software that responds to a client's request and processes the request by finding out the answer from the DNS server or servers. The resolver can deal with different types of DNS server, locally or remotely. This is usually a built-in utility.

DNS record: a file which has specific information that will tell the DNS server which IP is related to the request that the DNS server has received, i.e. a list of IP addresses and the names that are associated with those IPs.

There are different types of records, for instance:

- A record: Responsible for mapping name for IPv4
- AAAA Record: The same as A record, but specifically for IPv6
- CNAME (Canonical Name) Record: Works as a pointer to denote that this domain name is an alias name for another domain
- PTR Record: A record specifically designed for reverse lookup, and each PTR record related to a specific A record
- MX (Mail Exchanger): Specifies the mail server that is will receive the mail in this domain

There are more records; these are the most commons ones ("CNAME record," 2017).

DNS Zone: a part, or contiguous parts of namespace within a DNS, represented by specific records saved in order (or in sequential methodology) for an administrative purpose, to show which of those domains belong to which records, including which is the high tier domain and which is the lower one. A DNS zone is usually represented by a text file.

1.3 Open DNS Resolver

In general, a DNS resolver is software or a specific server responsible for responding to client's requests. In most standard enterprises, a DNS resolver is placed in the internal network to serve requests for internal users, but there is also another scenario where the enterprise has mobile teams or more than one branch which makes them put the DNS resolver facing the internet. To understand real behaviour and how the attacker exploits it, we can refer to (Matsubara, Musashi, Sugitani, & Moriyama, 2015).

If an organisation installs their own DNS server, they have to configure the server to avoid it being exploited. This can be done in different ways, for example, denying all unknown IPs, being aware of DNS ANY queries and put restrictions in to prevent them, not accepting any requests from untrusted sources and only from a specific range of IP, and filtering all requests that come from the internet, etc.

There are a lot of DNS servers that serve incoming requests in different ways, some publicly and some privately. There is another type that works as semi-public, such as those that are existing in ISP networks to serve a wide range of specific subnets and clients. Furthermore, there are other types configured to accept any request from any source: these servers are call Open Resolvers. Open Resolvers work without specific conditions, and could either be misconfigured, or could be deliberately configured in this way to run specific types of attacks. When the server accepts any type of request from any source, the attacker can easily exploit it to execute DNS Reflection and Amplification attacks, simply because there

are no security criteria or control permissions to check the request source. So, any request with a spoofed IP can pass and, in this case, the open resolver servers work as a bridge for these types of attacks. Also, it can be exploited to execute other types of DNS attack, such as DNS poisoning, DNS tunnelling and DNS hijacking.

1.4 DoS/DDoS attack concepts

DoS attacks are very common in cyberspace. In these attacks, the attackers launch a flood or start to exhaust target resources with rogue requests to prevent legitimate clients from accessing their services. As a result, the targeted server could be crashed or be unresponsive. There are a wide range of vulnerabilities the attacker could exploit to launch their attack (Nur & Tozal, 2016).

However, preventing and controlling, or even dealing with, those types of attacks is very hard for many reasons. There are a lot of scenarios that attackers could follow, which makes it very hard to deal with the attacks if an organisation and its staff are not prepared to face such situations. Being ready to mitigate or prevent DoS/DDoS attacks and defend against them needs a lot of preparation, such as a well-tested plan and a good monitoring system that can indicate the changes in services, such as degradations as a proactive indication, and having a backup of data. This backup could be on-site or off-site and it is important to have direct contact with the service provider for support when the organisation is under attack. They can work to support the organisation by mitigating attack factors, such as, for example, a flooding attack. In a flooding attack, organisations cannot do a lot and the ISP team that the organisation connects to should play their role to control malicious traffic as much as possible. There is a lot of deliberation about this, as Fu discusses in his paper (Fu, 2011).

In 1988, Robert Morris's worm launched the first DoS buffer overflow attack in the USA. At that time, DoS attacks were not too complex and were limited to sending a high amount of data, therefore detection systems were built based on anomaly traffic detection. Now DDoS attacks are constructed in a distributed methodology through multi-stages, as preparation before a real attack is run. The most common first stage is by developing malware or code that has the ability to run different types of attacks with more than one approach. The next stage represents recruiting zombie computers, which can be done in many ways, such as; using an internet worm to scan and recruit vulnerable computers and distribute malicious code in these systems; using a legitimate download but through malicious websites; or exploiting advertisements. The attackers consider the easiest way, and through a lack of knowledge, users enable the attackers without even being aware. The attackers use different protocols to communicate with their agents, but most of them use well-known protocols, such as HTTP, according

to the attack layer. Using these well-known protocols avoids the attacker communication being detected, while using their own protocols or an unknown one could more be easily detected by security systems.

1.5 Related work

A lot of papers have been written that cover DoS/DDoS attacks in general, and the different approaches of DNS DoS/DDoS attacks specifically. Each paper uses different methodology to analyse the attack and design mitigation techniques. While some of these approaches and ideas have been adopted by security companies and included to mitigate attacks, these types of attacks are still active and negatively impact our cyberspace and online services.

In general, from the location perspective, there are two approaches security teams could follow to prevent or mitigate DoS/DDoS attacks. The first one discusses applying their mitigation technique near to the attack's source, and the other discusses applying their solution near to the attack destination. This paper discuss TCP & UDP DoS aim to over-utilise targeted CPU attack and where the mitigation stage should be for better results. Also Analysis the response of Adaptive Security Appliance (ASA), Threat Management Gateway (TMG) and Next Generation Firewall as well. Each defending approach has positive and negative aspects and can be applied I different scenarion. To understand the reasons, it is a good idea to analyse the behaviour of the security appliance and decide from there (Maraj, Jakupi, Rogova, & Grajqevci, 2017).

Most IT security specialists consider applying the solutions near to the attacker's source as a big challenge for many reasons. For example, how to discern who the next attacker will be. Attackers can use legitimate requests that cannot be denied, and most attacks have a distributed nature which makes it impossible to apply anything near to any one source. For these reasons, most solutions will be near to the target or assets that are under attack. However, there are a lot of challenges in this scenario, too. By increasing attack complexity, it becomes very hard to detect the attack packets, especially when attackers use a legitimate request from victims' computers, i.e. hacking a computer to send attack packets. This makes detection more difficult, like a fight against a hidden enemy.

There are some kinds of attacks, such as a flooding attack, that cannot be defended against from the targeted side after it launches because the devices will be flooded with attack packets, and at this stage it is too late for anything to be done (Wang, Dunlap, Cho, & Qu, 2017).

However, defending response should combine all possible features and detection mechanism which can help in recognise malicious traffic, like, Time to Live (TTL), Spoofed Packets Address, IP Packets Size, reflection packets etc. then deal with legitimate and illegitimate traffic in different approaches and criteria. In my opinion, the best way is to consider a solution that reduces attack factors, and I address this with my mitigation technique. DoS/DDoS attacks come in different shapes and faces, and also from different sources, so by analysing previous attacks we can ascertain the factors that the attacker depends on in his attack, and the key points that the attacker uses to run a successful attack.

Malicious traffic should be controlled as close to the source(s) as possible (Jose & A., 2014). Most of today's solutions are deployed on at organisation's edge routers, which means that the threat is already at their door. While there are tools and mechanism could help in as a firewall supportive mechanism like Hadoop; which Java framework can be used to store and process large packets, Chukwa; which is distributed log and collection system, those tool can play a supportive role by updating security appliance in term of increasing detection approaches. However, the question of how effective their solutions can be when the ISP's router passes the flooding or volume attack traffic to the victim's router. The answer is they cannot. The link is already flooded and normal traffic does not get a chance to pass, so the aim of the attack is already achieved. Most recent statistics confirm that despite the various solutions, attacks still threaten and are highly effective. The ISP must take some responsibility for preventing malicious traffic and be a part of the solution or defence, rather than a bridge to pass on an attack.

There are other approaches, such as each group of routers, or clusters (called a "club". i.e. clusters that forward traffic between other clusters), have backbone routers which do the job of forwarding transit traffic (Fu, Papatriantafidou, & Tsigas, 2011). Each cluster has a centre router called a coordinator. The coordinator is responsible for allowing or denying passing traffic between the clusters according to authentication codes swapped between them. The traffic is classified into three types; Outbound, Inbound and Transit, and each cluster coordinator deals with each traffic type according to related criteria.

Another proposed method, which is depend on statistics methodology for specific features, packets flags and bit, like, SYN, FIN, REST, Source IP and DST IP. Proposed solution received traffic and processed it in parallel parts called "child parts". The results regarding suspicious traffic are then transferred to another part called an "aggregation part" (Hasanifard & Ladani, 2014). The proposed system works to recognise the incoming traffic and to increase the speed it uses at parallel stages. To recognise and categorize incoming traffic features and flags a thresholds have been set by depending on Holt-winters

Algorithm. Furthermore, the proposed solution implemented in Snort open source IDS to evaluate processing speed performance and it show using parallel methodology increase the performance. Practical results show each child can handle 1Gbs and with 8 childes it can have processed 8Gbs instantly. In conclusion, the proposed solution is good but including more features will increase its accuracy. The proposed system works similarly to existing IDS systems and depends on the same criteria, by counting the number of instances that have the same IP and header information, in a specific period.

Some researchers propose an anomaly-based detection system that monitors the outgoing traffic at the border router of a source network (Malliga, Tamilarasi, & Janani, 2008). This research team proposes a monitor system following calculation methodology at the edge router to collect statistics about the number of connections from each host and calculate the data received. Our approach proposes a Multiple Flow Monitor (MAM) that monitors the number of connections opened by each of the hosts and calculates the information entropy for each source address received (Malliga, Tamilarasi, & Janani, 2008). This monitor is located near or at the edge router and monitors traffic trends.

Some researchers also try to monitor the DNS ANY requests, as some IP hosts send a lot of the ANY Resource Record (RR) based DNS query request packets to the tDNS server, the researcher proposed a detection model called DNS ANY Request Cannon (DARC), and this model based on Euclidian distance between source IP address, then a suggested threshold have been set for detecting processes. (Tritilanunt, Sivakorn, Juengjincharoen, & Siripornpisan, 2010). The researchers statistically investigated the total ANY (RR) based DNS query request packet traffic from the internet to the top domain DNS server. There are a lot of discussions about ANY requests, as these types of requests can be easily exploited to exhaust server resources in different ways, such as using NXDOMAIN, random sub-domains, to exhaust server resources and make it unable to reply to normal requests, thus achieving the aim of the attack.

Comparing the performance of existing solutions is also an excellent idea by investigating the performance of the three IDS under different traffic rates and attacks, and using the number of packets lost, the number of alerts, and the CPU usage as the metrics (UzmaSattar, Naqash, Zafar, Razzaq, & Bin Ubaid, 2013). Evaluation process run against DoS attacks, DNS attack, FTP attack, Scan port attack and SNMP attack by using attacking tools such as Network Mapper (NMAP), Ostinato, High Orbit Ion Cannon (HOIC) and Low Orbit Ion Cannon (LOIC). When testing three open source IDS software (Snort, Suricata, and Bro) under a different type of attacks each IDS generate its own alerts according to match rules and thresholds. However, they found that Bro performed better than the others, but these IDS's are

developing very quickly and showed differing results according to the number of rules and the attack type when last updated.

Also of interest is how to investigate packet headers, i.e. examining the header field of every incoming packet in order to detect DoS and DDoS packets. Checking each packet header will add a number of challenges, such as delay and high CPU and RAM usage. But, this technique fulfils the requirement of detecting DoS/DDoS attacks that have a small traffic volume. In normal traffic, the packet should not be identical in a specific period, but on the other hand most attacks have some kind of similarity, for example, in packet size, and this can help to detect a small packet which is not detected by a larger threshold.

Bloom filter: "A Bloom filter is a space-efficient probabilistic data structure, conceived by Burton Howard Bloom in 1970, that is used to test whether an element is a member of a set." ("Bloom filter - Wikipedia," n.d.). This kind of filter can be used to check each request with related responses and this can be done in both directions; in and out. The filter stores the request and waits for the response. If the response is within the given interval of time, the server will allow the traffic to pass through (Fu, Papatriantafidou, & Tsigas, 2011).

1.6 Business Impact

In 2014, according to many security statistics reports, DoS/DDoS attacks targeted a wide range of business. Take McAfee as one of the trusted security reports: McAfee stated in its report in June 2014, that the impact of cybercrime on economics across the globe was estimated to be more than \$400 billion dollars, and according to reports it could be in the range of \$375 billion minimum, to a maximum of \$575 billion dollars (McAfee, 2014). In 2017 cybercrime cost the global economy \$600 billion (McAfee, 2017).

Such numbers are alarming and could have the devastating outcome of destroying the targeted companies' businesses forever.

In addition to massive financial losses, there are millions of people who have also lost their personal and sensitive information, and this could have very dangerous impacts, too, should attackers decide to use some of the personal information in forged identity crimes. Let us take some statistics from McAfee reports for people affected and their personal information stolen in 2013: in USA the number was 40 million; Turkey, 54 million; Korea, 20 million; Germany, 16 million; and China, more than 20 million.

At a company level in the USA, in 2013 alone, 3000 SMBs were affected. In the Middle East (Gulf region), two banks were affected and the total loss was estimated at \$45 million dollars. It took this short attack only a couple of hours to cause these devastating results. In England, one company declared that it lost \$1.3 million in just one attack. Other companies do not declare this kind of information as not only do they lose money but they also risk losing their reputation and the confidence of their customers. If customers feel that services are unsafe or unstable, they will change service provider, and what was built through hard work across many years can be destroyed in a couple of hours.

Figure (3) shows service provider experience threats and table (1) illustrates different aspects of impacts that these attacks could cause.

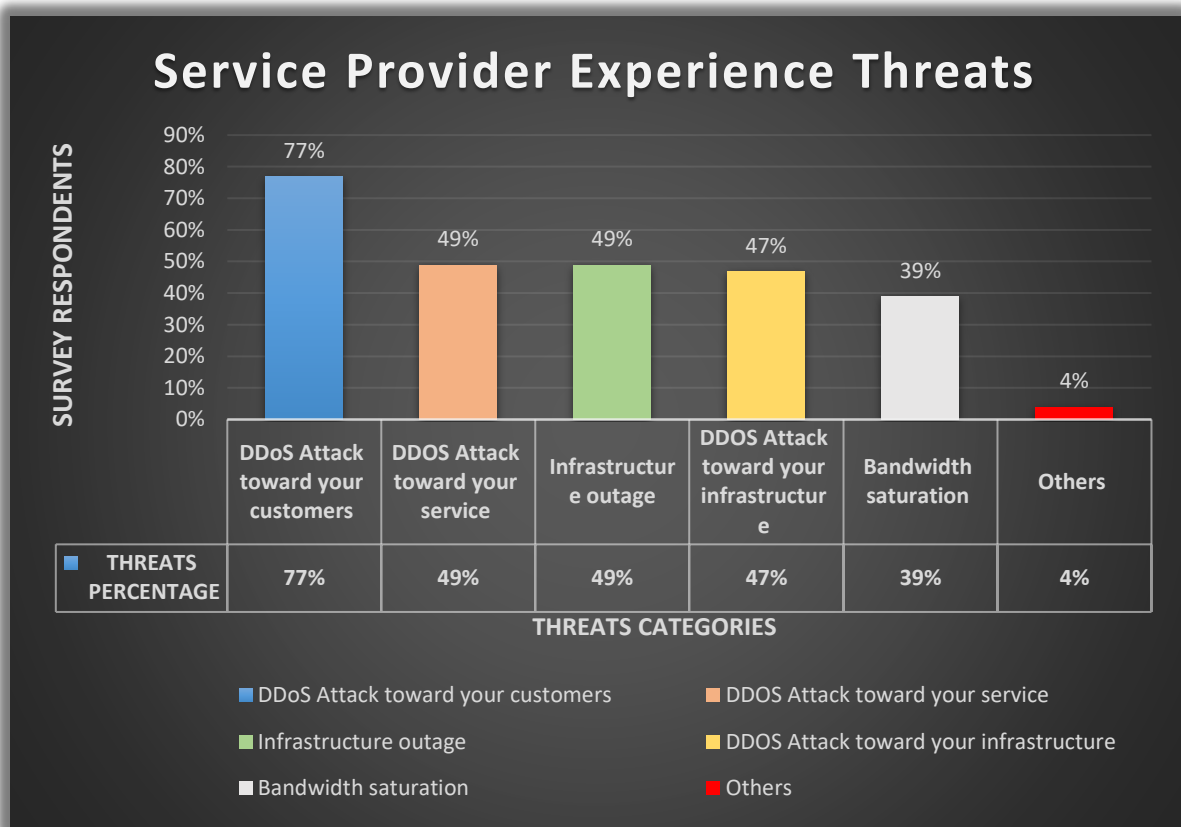


Figure (3) Service Provider Experience Threats (Anstee et al., 2016)

In 2014, the cost of DDoS attack per hour was \$40,000				In 2014, 36% of businesses were not confident about their current protection technology			
1- DDoS Widespread	Org's attacked in general	Org's attacked more than once	Org's attacked in the last month	Org's attacked on weekly basis			
Statistics in 2014	45%	74%	91%	10%			
2- DDoS Attack Length	0-6 hours	6-24 hours	1-7 days	7 days and above			
Statistics in 2014	37%	49%	8%	4%			
3- DDoS Attacks Impacts	Software\ Hardware Replacement	Reduction in Revenue	Loss of customer trust	Customer data theft	Financial theft	Loss of intellectual property	
Statistics in 2014	52%	51%	43%	33%	26%	19%	
4- Business Targeted	IT groups	Sales	Security	Customer's services	Marketing	Legal	Others
Statistics in 2014	35%	23%	22%	12%	5%	2%	2%

Table (1) How DDoS attacks impact the business (Incapsula, 2014)

1.7 Security reports and statistics

In any study, one of the main factors that the researcher depends on is the statistics that show the recent trends that the study is following. Cyber-attacks are increasing day by day, as recent statistics show, as explained later in this section, along with the statistics mentioned in the previous section. If we take into account all considerations, we can see how these types of attacks can seriously affect our lives. All of these statistics together could be a big motivation factor for IT security specialists to stand against these attacks which breach our organisations and the privacy of peoples' lives in such an aggressive way.

Recent trends show that most of technology's assets are moving to cloud and data centres, which offer a wide range of reliable services, but also add a lot of security challenges. There is a necessity to review protection methodologies and try to follow proactive methodologies to protect information and assets,

rather than reactive defence mechanisms. Because using cloud services means sensitive information is stored online and there is the possibility that our information is shared with other customers, these centres become a strategic target for attackers. There are different types of DDoS attacks shown in the following statistics and we can see that DNS attacks are the second highest. Figure (4) shows attack vector frequency and figure (5) illustrates DDoS attack statistics – statistics by Kaspersky and Solarwinds.

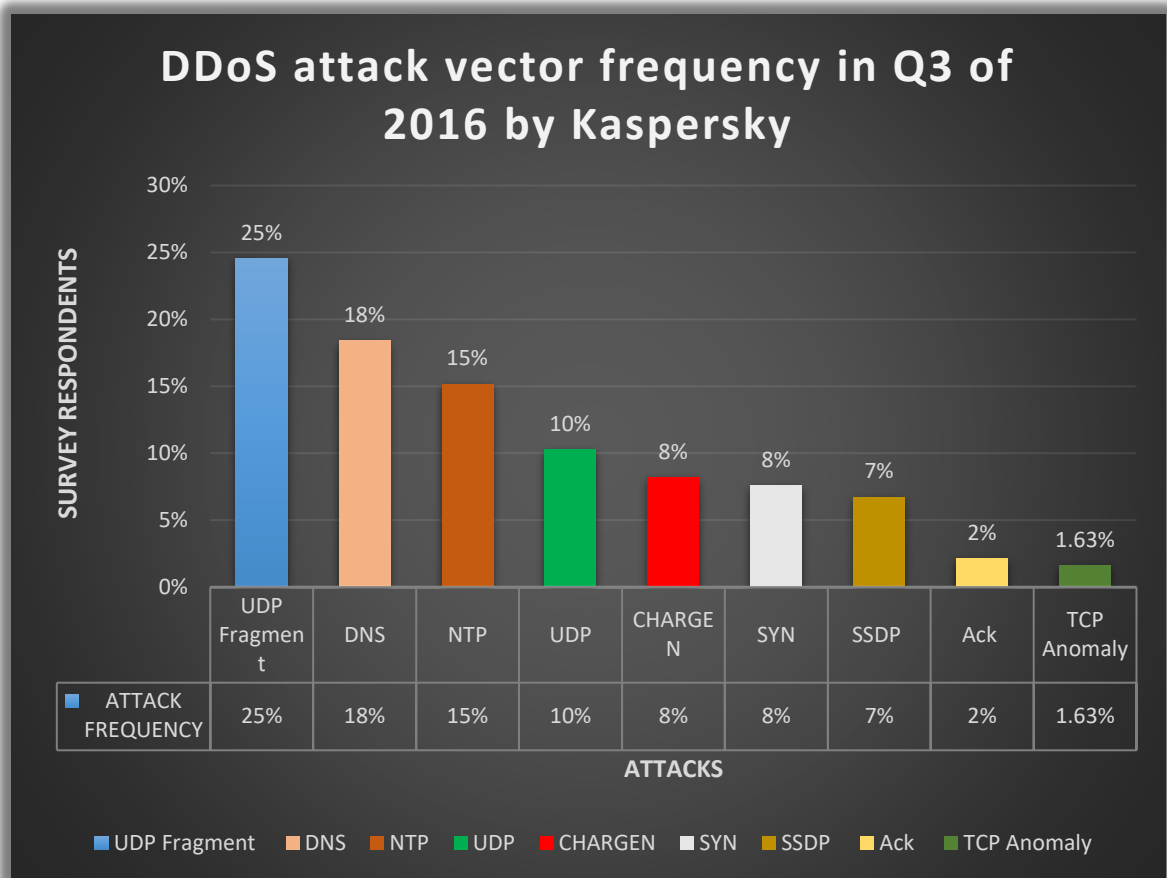


Figure (4) Ratio of different types of DDOS Attack (Akamai, 2016)

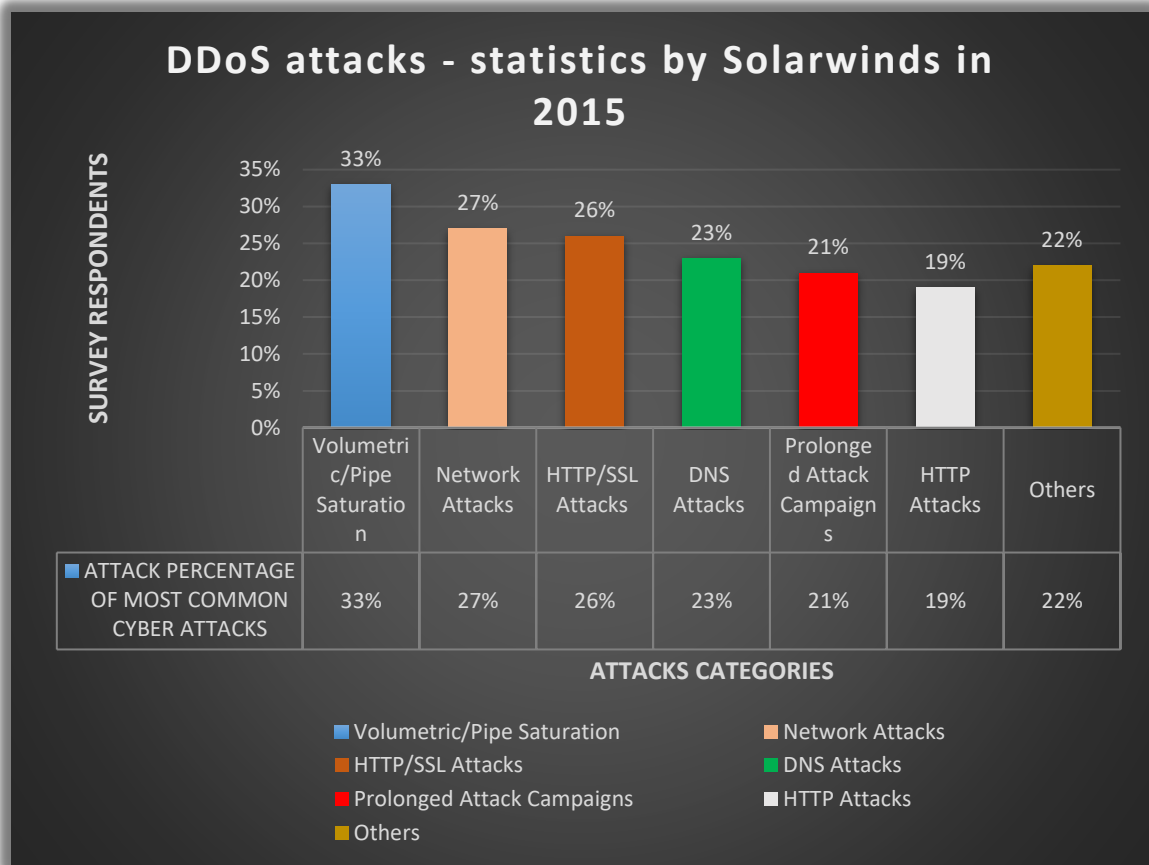


Figure (5) DDoS attack types in 2015 by Solarwinds (Ratio of DDoS attack types, 2015)

1.8 Motivation

This section discusses the scientific reasons for any type of research. The most important factors that motivate a researcher to complete his study in a particular area is the formal statistics that confirm that there is a need or lack of solutions about this specific area. Sometimes development and increasing complexity can also be good reasons to provide new solutions that can support existing legacy solutions. Usually researchers focus on how their research's results could help others or serve the community and improve the solutions they already have.

In this study I cover DNS reflection attacks, and figure (6) illustrates that DNS is the highest protocol or highest service that is exploited in reflection attacks compared to other protocols, at 84%. Figure (7) also illustrates that DNS is considered the highest targeted service when compared with other services in layer 7.

From the statistics shown in the figures below, it is obvious that there is a need to work on improving solutions and give SMBs more options to suit the different needs of the organisations that face these types of attacks.

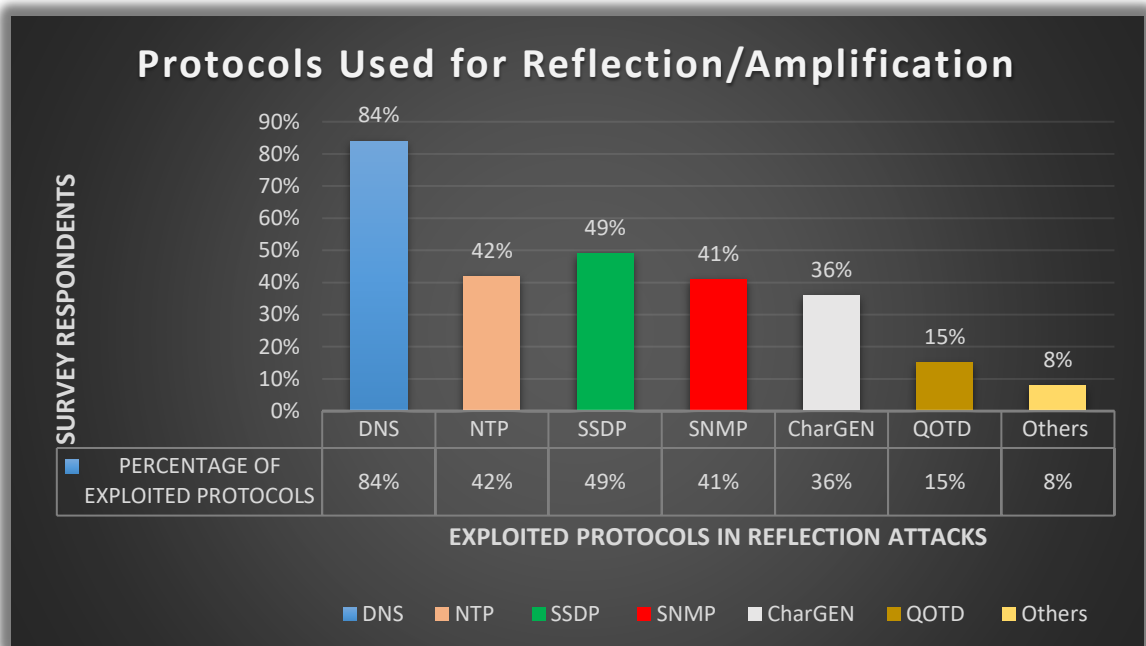


Figure (6) Protocols that are used in Reflection attacks (Anstee et al., 2016)

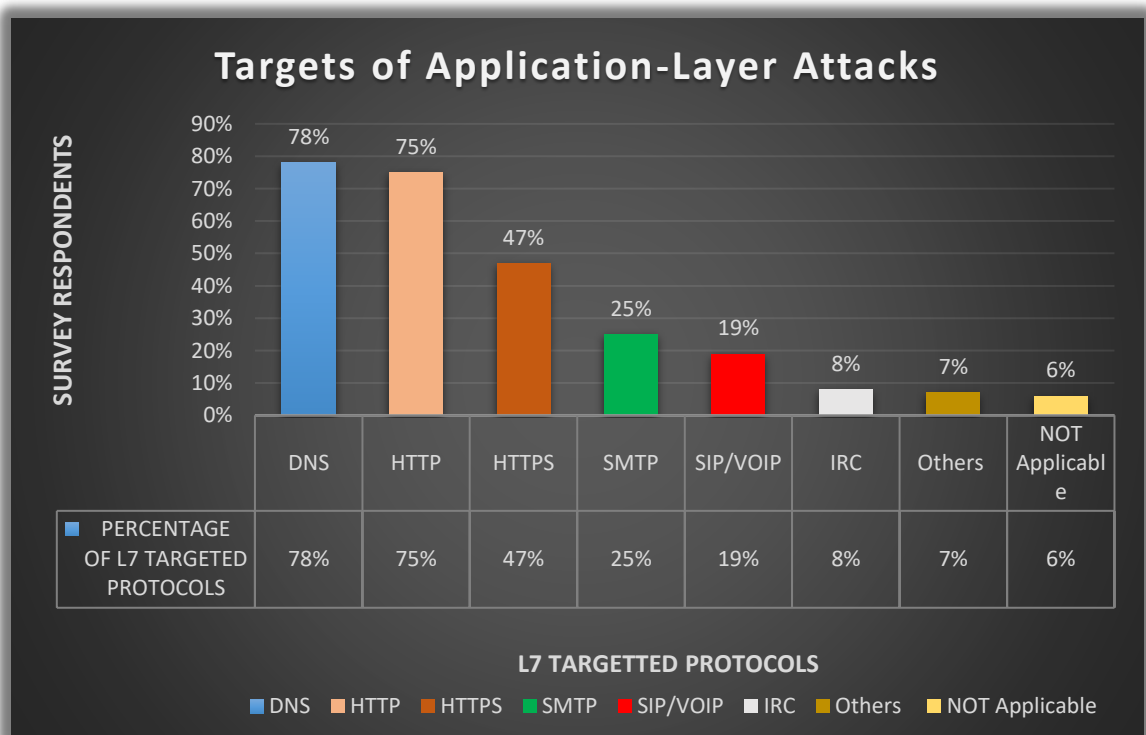


Figure (7) Targets of Application Layer attacks (Anstee et al., 2016)

1.9 Research contribution

In this study, I focus on mitigating Reflection DNS DDoS attacks to help SMB organisations protect their networks with easy, simple, cheap, compatible solutions against these types of attacks. To achieve this, mitigation techniques can be divided into three stages; attempting to stop attack traffic at each stage according to the role of each stage. The suggested techniques include reusing existing features in new ways to build an integrated mitigation solution and deal with attack traffic in different sequence stages. I chose to build these stages after I conducted attack traffic analysis, to cover all possibilities that an attacker could use to send attack packets.

When comparing the suggested solution with the solutions suggested by other researchers, it is important to consider solution compatibilities (vendors), complexity, practical results and solution cost (financially and resources).

In this study, all of the results were collected from practical experiences, which will help give people who choose to adopt the suggested solution confidence to apply it.

1.10 Thesis structure

This thesis includes seven chapters which cover sequentially all of the aspects of the research area. Chapter one includes the introduction and a brief discussion about DNS service with required statistics, literature review and motivation factors, ending with research contribution. Chapter two covers DoS/DDoS attacks according to the OSI model, by giving a brief description for one or two attack examples at each layer of the model. Chapter three focuses on one type of DDoS attack i.e. DNS DoS/DDoS attacks, and discusses the most common attack types that exploit DNS services. Chapter four covers the thesis hypothesis and research methodology, includes definitions and discusses research types, in particular qualitative and quantitative research, and the studies that are a mix of both. Chapter five represents the launch of the practical part of this study by discussing the testbed and applying the mitigation technique, then verifies the mitigation technique stages by running a real attack and collecting verification results. Chapter six covers the evaluation process for the mitigation technique under different conditions by running different levels of attack, represented by sending packets in different numbers and size to compare how much load the suggested technique can cope with at each stage. Finally, chapter seven discusses the results collected from the evaluation process conducted in chapter six, and discusses the possibilities of recommended future work.

1.11 Summary

This chapter includes an introduction which shows most aspects of the study. Study aspects include internet service terms and concepts, DNS service concepts and mechanisms, and open resolvers and their impacts, plus related work which covers most of the ideas and techniques in this area that are used by other researchers, and statistics that show the impacts and the ratio of attacks in 2015 and 2016. The next section represents research which shows how this study contributes by suggesting a suitable solution for SMB organisations. The final section has the thesis layout which shows what each chapter in this study represents.

Chapter 2

Classification of DoS/DDoS Attacks

2.1 DoS/DDoS terms and facts

DoS/DDoS attacks take different approaches and methodologies according to the targeted victim's assets and the attacker's plan in targeting a specific victim. DoS is a cyber-attack in which the attacker targets a server, network resources or any IT assets, with the aim of stopping or denying legitimate clients from accessing services from the service's providers (Fu, 2011).

However, with security specialists adding security systems to recognise and prevent these attacks, attackers have had to develop advanced methodologies to avoid being detected, and have also started using distributed attack methodologies. Initiating the attacks from different IPs and from different locations makes it hard to detect attack packets due to the difficulties distinguishing between attacker packets and legitimate packets, especially when attackers send normal-sized packets from thousands of distributed IPs and locations.

Attackers can run distributed attacks, without the need to use hundreds of computers. All they need to do is scan for computers, networks or any online assets that have vulnerable systems and recruit those systems under the attacker's control command servers. Whenever this has been accomplished, those controlled computers, or recruited or infected computers, will be ready to participate and play the main role in the next DDoS attack. There are different names for the infected computers, such as, bot, zombie or botnet (Pijpker & Vranken, 2016).

"Bot" is an abbreviation for robot. Any computer infected with malicious software, traditionally known as a Trojan horse, will be called a bot or a zombie, and it works to exhaust a machine's resources. If an attacker runs an attack through one computer, there are many ways for it to be recognised and blocked, but what makes the attack more effective and sophisticated, and live for longer hours, is when the attacker recruits hundreds of machines, network resources and online services to serve his attack in distributed approaches. All of these resources together are called a botnet, and it is difficult to detect all those that are participating in the attack in a short time.

A Control and Command centre (C & C centre), is a machine or server that works as a master or commander to slaves (computers) or recruiters, which are represented by bots or botnets (which are computers that are already infected with malicious software), with specific scripts for one or different types of DDoS attack. C & C centres use different methodologies to communicate with their botnets, for example, IRC (Internet Relay Chat), Tor browser, Facebook, and Twitter.

C & C centres are designed, run and communicate in different methodologies. C & C star design is where one centralised server controls the botnet in an easy and reliable way and all workstations are directly connected to each other. C & C topology design is where there is more than one server connected together and work as a redundant or backup to each other. Topology design adds more complexity with more features and also avoids depending on one server. If, for any reason, one of the servers gets detected, the attacker will not lose complete control, however it is also hard to implement and manage.

Levels or hierarchical servers will be connected in a tiered way, which means that when the main server wants to send an order, it will not send it directly to bots, instead it will, for example, send it to another server which will then send it to bots. The attacker protects themselves in this way and if for any reason the server which sends the order or command to the bots is detected, the attacker will not lose complete control as I said, and later on can recruit a new server and begin again.

The last piece of this puzzle is called a backdoor, which is a programme or feature that allows the computer administrator to log in to the computer system to run troubleshooting tasks. From the attacker's perspective, it provides an easy way to access and exploit a victim's computer, with the aim of controlling it. Sometimes these backdoors are system vulnerabilities known by the system programmer but not known by the system buyers. There is also another type of backdoor represented by malicious software installed by attackers for malicious purposes. When the attacker compromises any computer, it goes through multiple breaching stages until it gets full access and full control, so if the attacker loses his connection for any reason, or if a genuine user changes a setting, the attacker will lose his access if he did not create or install a backdoor. A backdoor plays the important role of supporting attack stages by giving the attacker quick and easy access to a victim's computer, saving the attacker a lot of time.

There are many types of backdoors and the most common one is called a Trojan. This name was derived from the historical Trojan horse, which was a wooden horse in which soldiers were hidden in order to gain access to the city of Troy. Nowadays, there are many types of backdoors, like a Trojan backdoor, which allow an attacker to access a computer system and steal user data, and download and install malicious software to create his own backdoor, which allows the attacker to gain full remote control of the victim's computer(s).

DoS/DDoS attacks are classified according to the targeted OSI layer (in the information technology field any service or device is classified according to standard OSI layer). Classifying attacks according to OSI

layer helps us to understand the attack's nature and recognise what is required to stop or mitigate these types of attacks.

2.2 L3 or Network DoS/DDoS Attack

In this category of DoS/DDoS attack, attackers target the network layer. There are many types of attack in this layer, but this section will cover two of most common attacks to clarify the general idea: “Ping of Death” and “Smurf” attacks.

2.2.1 Ping of Death DoS/DDoS attack

In the ping of death DoS/DDoS attack, attackers send informal packets larger than 65,535 bytes in size, which is the standard size allowed for IP packets. The TCP/IP suite has a feature that allows it to fragment the packets into smaller pieces at the sending edge and then merge them at the receiving edge. The attacker exploits this feature to change the size and execute their attacks, as there are a lot of systems that cannot deal with packets of that size and will crash or freeze if they receive anything over the standard size of packet. Different servers, firewalls and network devices deal with these types of packets using different methodologies (Arunwan, Laong, & Atthayuwat, 2016).

Most DoS/DDoS attackers need to know a lot of information about the target to discover how the system's owners built their system and the features the system has. However, with ping of death, the attacker does not need to know a lot; all they need to know is the IP address of their target. Additional to that, the attacker can easily have spoofed any IP to run his attack, which makes it very hard to trace or discover.

2.2.2 Smurf attack

With Smurf DoS/DDoS attacks, the attackers depend on Internet Control Message Protocol (ICMP) to run their attack. In a standard situation, the client sends an echo request to other clients to check the connectivity or the delay or service jitter. In the Smurf attack, attackers use a malware to generate their own request and send it with a spoofed IP (i.e. the IP address of the victim) to the broadcast IP address of the attacked network (Zargar & Kabiri, 2009). In other words, the attacker will send echo requests to all hosts and IT assets in a network. Those IT assets could be

servers or devices connected to the working network. Using this methodology will amplify the attack size by the number of clients that will receive the request. When the hosts start to reply to the requests, they look for the source IP address of the packet (which will usually be the IP address of victim's server), so all of those who receive the echo request will reply to the victim with an echo reply, thus flooding the victim.

In this attack, the attacker uses the amplification factor multiplied by the number of the hosts in the network, making it huge and difficult to trace or discover.

2.3 L4 DoS/DDoS attacks

DoS/DDoS attacks at this layer will mostly target a specific service rather than targeted host or computer. This section covers the most common attacks at this layer.

2.3.1 L4 SYN DoS/DDoS attack

In SYN attacks, the attackers exploit the “three-way hand shake”, which is the standard TCP protocol mechanism to establish the connection between two computers or two clients. The perpetrator will send TCP SYN requests to all ports in the servers, usually with a spoofed IP address. The server replies with the SYN-ACK and waits for the last reply to establish a connection that will never come. By sending hundreds of these, the attackers exhaust server resources (Nissanke & Sun, 2008).

In effect, the attacker will try to exhaust the victim's resources by waiting for the TCP timeout period to send another one, to keep the server busy and buffer full of requests. In this case, the server cannot accept any other normal, legitimate requests. Usually, the attackers use small scripts and configure the sending request time according to request time-out, to keep the victim's server's buffer almost full, and runs this automatically to exhaust server resources. Usually in this type of attack, attackers do not use a spoof address as well: in this attack scenario the attacker aims to keep the server or router so busy that it cannot accept any more requests from legitimate clients. Detecting this type of attack could be by using in a different approach, for example, by analysing CPU, memory, link and network resource usage in a specific frame time, as discussed in (Aborujilah, Ismail, & Musa, 2014).

2.3.2 L4 DoS/DDoS SYN-ACK Flood attack

In this type of attack, the attacker sends an SYN request with spoofed IP address to the open servers, proxies and other resources that are already available on the internet, or even misconfigures some of them, to reply to any request. This could be very obvious by analysing TCP mechanism, SYN, SYN-ACK attacks (Mohamed & Kandil, 2009). These simple and small requests can easily pass any security appliance because the size, number and other criteria look normal.

Then the open resources and assets that receive the request will look to the source IP of the and it will reply to that source which in this case is the victim, and cause flooding of the victim's computer. Sending requests to these open online servers could create thousands of replies to the targeted victim, making the victim too busy to handle legitimate requests. There are many approaches to detect the attack, such as combining some thresholds to trigger the alarm detection (Aqil et al., 2015).

2.3.3 NTP attack

Network Time Protocol (NTP) is one of the oldest protocols used by different types of interconnected appliances through an internet network to synchronise their clocks. Some NTP versions also have the ability to participate in monitoring activities, so they can provide the administrators with a list of the last 600 hosts that are connected to the specific online server. In order for the attackers to exploit this protocol to execute a DoS attack, the attack could be launched in two fashions - standard DoS attack and DDoS attack. What makes it interesting is that a launch of a distributed version of attack does not need to breach computers (Sassani, Abarro, Pitton, Young, & Mehdipour, 2016).

This protocol is one of the UDP set, so it is a connectionless protocol that has a built-in command called a "monlist", which is used for many purposes, such as monitoring tasks. Attackers can exploit this to execute an amplification attack, by sending this command to the NTP server, aiming to get the server's lists that are synchronised with the NTP server. By sending this command the NTP server will reply with a list of IP servers and devices which could be a huge number of IP addresses that are synchronised by this NTP server to targeted victim.

The following is an example of a command:

```
ntpdc -c monlist xxx.xxx.xxx.xxx (Server IP)
```

When the attacker frequently sends this type of request to an NTP server with a spoofed IP address, the server will reply to the source IP address, which is the victim in this case. These requests flood the victim with the UDP packets which are much bigger than the sent requests. The ratio of amplification in NTP DoS attack ranges from 20:1 to around 200:1, and this ratio, along with frequent sending, exhausts any server resources.

2.4 L7 DoS/DDoS attacks

DoS/DDoS attacks at this layer target application layers with a different approach, such as targeting a specific application or Website architecture to perform malicious behaviour.

2.4.1 HTTP Flood attack

In this type of attack, the attackers target the upper layer, which it is Layer L7, or the applications layer, and target Web API (Application Programming Interface) to get at web servers. In this type of attack, the attackers do not send packets with spoofed IP or even send informal packets. The attacker's methodology here focuses on exhausting server resources with a different approach. In this attack, the attacker exploits formal HTTP requests, GET and POST, to craft requests that exhaust server resources. This type of attack is considered a "volumetric attack". Figure (8) illustrates some statistic percentages for different types of HTTP attack. The attackers study the victim's assets to understand how the victim's web server or applications work, which language is used, and how the internal commands work, to build their strategy to target architecture, and they exploit this information and knowledge to design the attack methodology.

After most websites moved to cloud service, attackers could not fully execute denial of service but they could cause other types of damage, as discussed in (Lin, Liu, Huang, Lee, & Chen, 2010).

After the perpetrator understands how the website works, they attempt to send GET or POST requests from their own or from recruited bot computers. A GET request is generally used for getting standard and static simple content and if the attacker exploits this kind of request it could

be detected by applying detection algorithms observing sent requests (Yatagai, Isohara, & Sasase, 2007), while a POST request is used to get dynamic and more processing requests. The attacker will try to control bot or zombie machines using Trojan horse software to send implicit legitimate POST requests which include complex parameters and force the server to allocate as much resources as it can to keep the server busy and prevent it from accepting legitimate requests, thus achieving the aim of attack; denial of service to legitimate clients. This type of attack aims to amplify the use of web server resources; CPU, RAM and any other server resources, such as internal OS library calls.

L7 DoS/DDoS attacks can be classified according to attack complexity, i.e. has the attacker launched his attack from a single or limited number of IPs, or has he used a wide range of different geographical distributed IPs (i.e. a botnet).

Figure (8) illustrates the percentage of the different types of L7 DoS attacks, and from the statistics shown in the figure (9), we can see that HTTP, in general, is the highest percentage compared with other types.

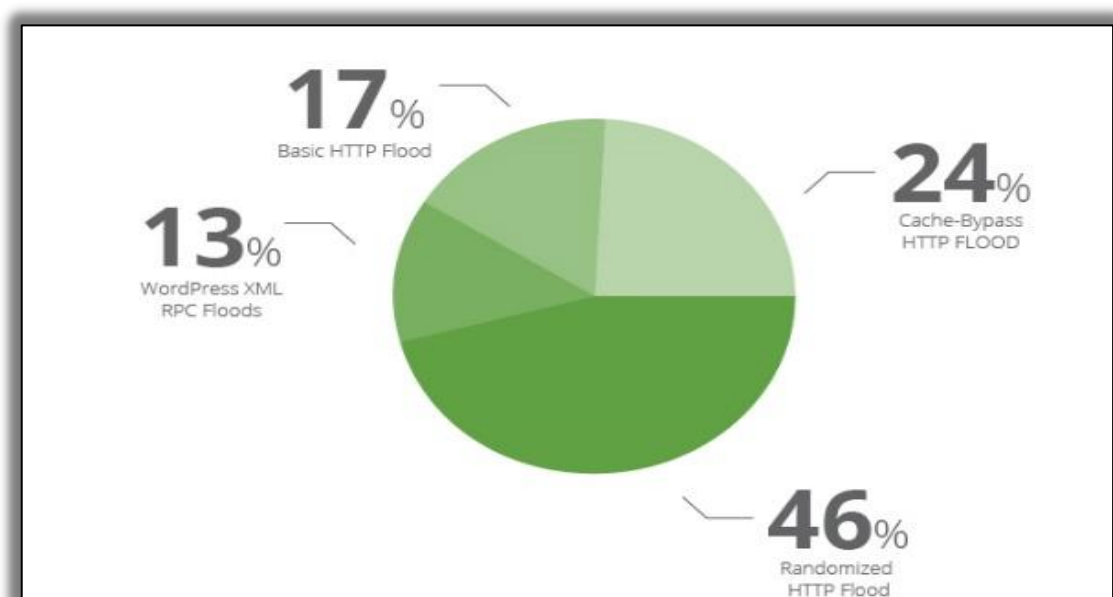


Figure (8) Percentage of HTTP DoS attack types (Daniel CID, 2015)

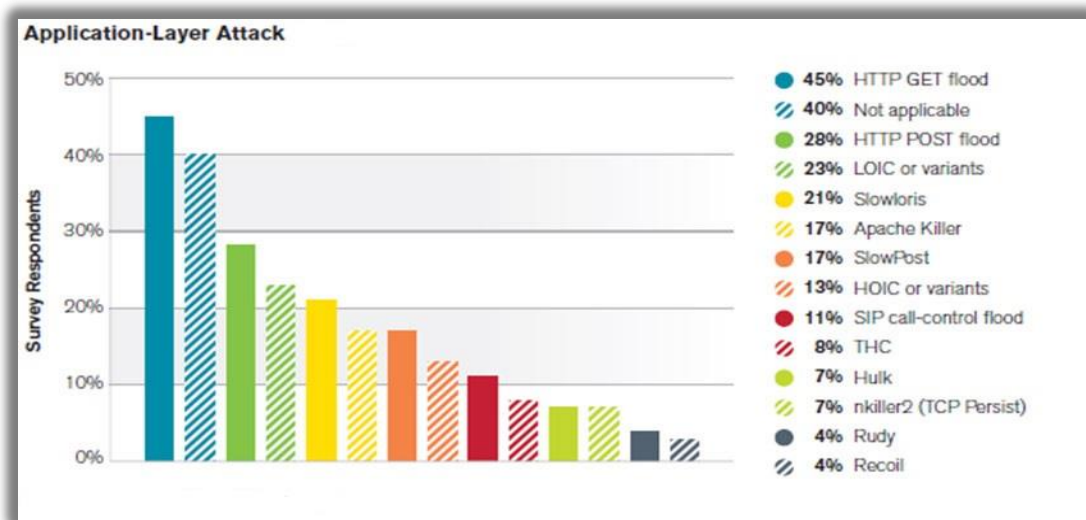


Figure (9) Percentage of Application Layer attack types (Calyptix, 2015)

2.4.2 WordPress DoS/DDoS attack

There are many languages and tools used to build websites. Around 2003, an open source project was launched by a group of interested people, which was dependant on Hypertext Processor language (PHP). This has grown to become one of the largest open source blogging tools, WordPress.

In general, WordPress is an open online project that supports two types of online services. The first one is by going to WordPress.org and downloading WordPress's script software to write and build whatever is needed. This is a simple, open community tool. The second one is by going to WordPress.com, which supports bloggers to start new blogs in a couple of clicks and seconds.

The following discusses one of the WordPress features that attackers exploit to run their attacks. This feature is called XML-RPC (Remote Procedure Call), as discussed in (Silaen & Lim, 2016). There are a lot of benefits to using this feature. At the advent of internet services, internet speed was not like what we have today. Today's internet speed would have been a dream in the 1990's. Bloggers and writers used a dial-up connection, so it was very hard to write a blog online because of low speed. Therefore, articles were written on a computer and then copy/pasted online. This did not solve the issue completely, because when their posts had codes or graphics bloggers faced another problem in that they could not load images and codes with a low-speed dial-up connection as they had before WordPress. From here came the idea of XML-RPC, which allowed the blogger to write and create his blog offline and then connect it.

Later, when smart devices appeared, and new WordPress apps, users were able to use their username and password to login to any WordPress, with the same privileges from wherever they logged in. If anyone discovers those login details, they have full access to all of the blogs. Plus, there are very useful toolkits used to customise websites, such as Jetpack, but as these also require XML-RPC, they too could be exploited.

XML-RPC use a “pingback” feature: pingback is one of the comment categories or features that are used for connecting two or more blogs. The attacker can exploit this feature to launch his attack, and this is what happened in 2013, according to Incapsula. The attacker exploited the pingback feature by using one computer to create thousands of links through both well-known and unknown websites and exploited them, without the need to even breach these websites. It is very obvious how these requests can exhaust server resources then easily take down any website with huge numbers of links.

The last thing I would like to say, is that this feature was enabled by default from the first release of WordPress and still exists, even with latest release of WordPress 4.8.2, Security and Maintenance where some filters have been added. This feature can also be exploited to launch brute force attacks because XML-RPC's mechanism allows the attacker to repeat-guess tens of passwords with one command, called “system.multicall”, by using a very small number of HTTP requests.

2.4.3 SQL Injection Attack

In general, a Structure Query Language (SQL) Injection type of attack is run by executing malicious SQL statements (code) to retrieve a web database. With malicious code, also sometimes called “payloads”, attackers will look for web vulnerability which is normally represented by user entry and requires an SQL query. Attackers inject their payload or malicious code as part of the SQL query, intending to access web data which will lead to theft or modification of sensitive data (Karuparthi & Zhou, 2016).

This is a brief explanation of how the attacker runs SQL Injection attack but is not the only approach this type of attack follows. An SQL Injection can be run as a DDoS attack or run through the DDoS attack. To simplify, any web query or data request sent to retrieve data from a web database and cause a high load on web server resources like CPU and RAM, and make it too busy to handle any legitimate queries, can be classified as SQL Injection in DoS attack specification.

The attacker, in this case, will recruit bots or botnets to send multiple tasks at the same time to a website to exhaust web server resources and also try to use complex POST requests rather than the simpler GET requests. There are detection methodologies that depend on monitoring and checking approaches to detect these attack requests, as discussed in (Dubey & Gupta, 2016).

Looking more deeply at this, the attacker could also try to exploit the methodology that the web architecture was built with. One of those methods is how database search engines are designed and work, what the search box entry accepts and what different types of entry mean. All of these together will be crucial factors to the success of the attack when the attacker tries to exploit the search engine or entry field to launch the attack.

The following is a simple example to clarify what the attacker tries to do with different approaches.

Case 1: What will the normal user enter in a job search website?

Case 1: For instance
the real user will enter:

OR

Network engineer

Security engineer

Case 2: What will the attacker enter in a job search website?

Case 2: For instance
The attacker will enter

OR

#\$googleahste OR &#\$NETWORK%\$#

OKYahooo127^%\$

Consider how long the server will take to return the results for what the search engine is looking for in these two cases: in case one it could take a couple of seconds, but case two will take much longer if the server did not configure correctly. Now, if hundreds of these requests are sent frequently and sequentially at the same time, it is easy to take down any website, especially if the website is not configured properly.

Furthermore, this could also have been deployed through scripts to redirect the search to specific sites. Before they run their attacks, all attackers collect a lot of information about their victim and choose the right attack according to the victim's website architecture and which language the web developers used. Figure (10) illustrates an example of how this attack could be done through scripts. Figure (11) shows percentage statistics about SQL Injection attacks compared with other types of attacks.

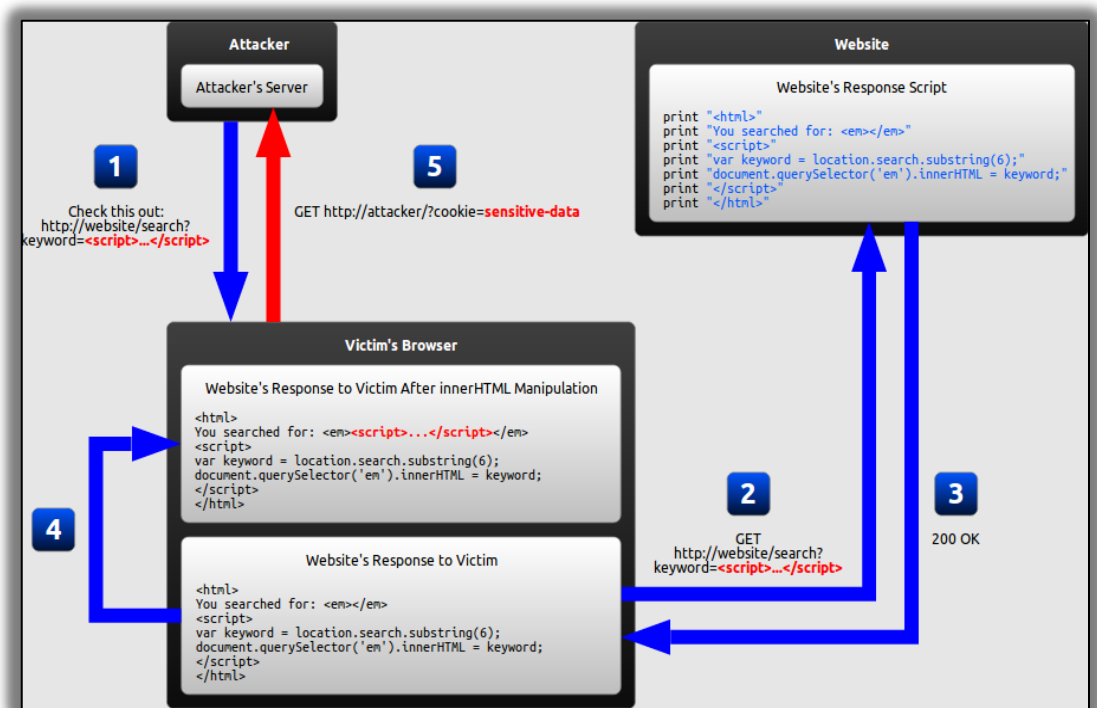


Figure (10) SQL Injection attack idea and stages (Wikipedia, n.d.)

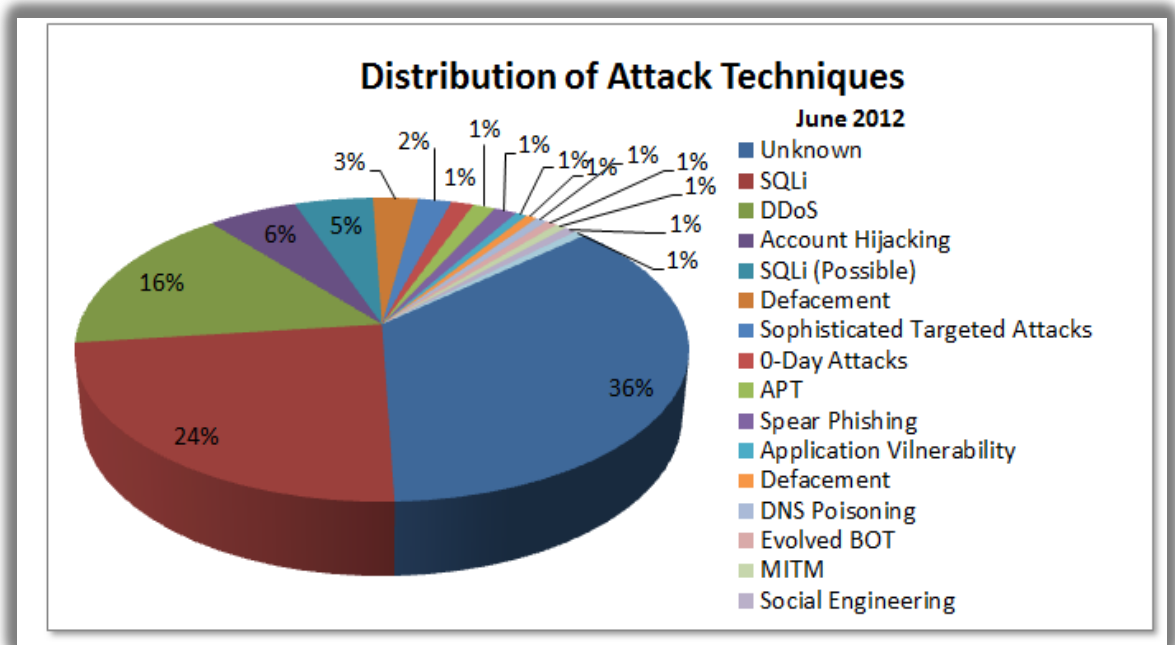


Figure (11) Percentage of SQL Injection attack (Kaspersky, 2012)

2.5 Summary

This chapter discussed DoS/DDoS attacks according to their OSI model. At each layer of the OSI model, two or three types of the most common attacks are discussed to illustrate the attack ideas. Discussing DoS attacks in this methodology helps the reader understand the nature of those attacks and how they work.

Chapter 3

Classification of DNS

DoS/DDoS Attacks

3.1 DNS Concepts and Terminology

The DNS is the huge database that contains all computer names and IPs. Records are designed as an index with which IP is assigned to which name. It is one of the most important internet services that is working in the background. All internet users use it even if they do not know about it or do not realise what this service provides them. Simply put, this service allows internet users to use common, friendly names instead of huge, confusing numbers.

This chapter covers the most common DNS attacks and explains some DNS terms and gives definitions for some DNS features.

DNS can be represented by the process or queries that have been made for specific records or IPs. When the user writes a name in their internet explorer to locate to a specific website, the translation process occurs between the common human-friendly name and the IPs related to or assigned to that website.

To explain more fully, most commonly each normal DNS server represents a domain. Usually, a server receives a query from inside its domain for information in his database. The server will directly reply and this is considered an authoritative reply because the server which receives the query has the answer already in its database. If the server receives a query for information that it does not have, or is out of its domain, then there is a need to ask another authoritative server to get the answer and pass it to the client who made the query - this is called a Non-Authority Server. By understanding the whole idea and analysing it, we can recognise the nature of the security concepts and the risks involved (Bassil et al., 2012).

Recursive DNS server: These are servers that work as a pointer to refer to other DNS servers. These servers are commonly deployed and used by ISP companies to work as recursive and DNS cache servers to increase network efficiency, and these servers need to be configured properly to serve clients or they can be easily used by attackers who exploit the nature of a query to run a different types of attack, and the server could be a victim or a part of an attack against other servers (Lanlan Pan, Xuebiao Yuchi, & Yong Chen, 2016).

DNS Zone: DNS Zones are created for organising purposes. Each DNS Zone represents specific parts of a DNS system to be controlled and managed by the administrators of the zone. Figure (12) illustrates an example of DNS hierarchical design.

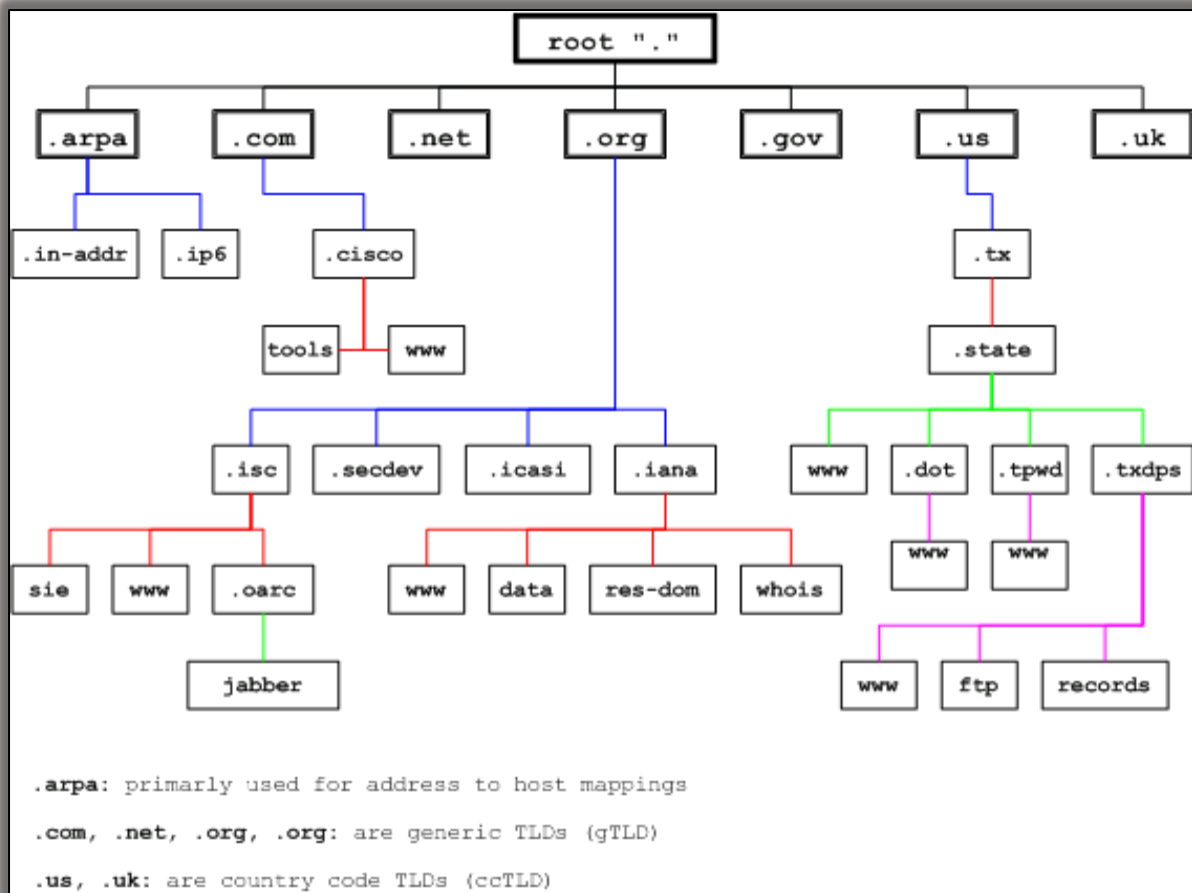


Figure (12) Domain levels represent DNS hierarchical sample (Cisco, n.d.)

World Wide Web (WWW): The WWW is a communication methodology that gives the ability to access and exchange information on the internet. It is dependent on hypertext transfer protocol in its communication to link to and access information. This is the very simple meaning of the internet, which is to access and exchange data in different distributed locations over all internet networks. To access any website, an “address” is required - this address is called an IP. Keeping or saving huge numbers of IPs is too difficult for human memory, so DNS was created to make it easier and more familiar for human memory.

DNS Resource Record: This is a type of record that contains information related to a database in the DNS server. These records work to increase service efficiency and it could be a little different from one DNS software to another, according to software design and architecture.

3.2 DNS attacks - Methodology Descriptions

This section discusses the most common DNS attack methodologies.

3.2.1 Distributed Reflection DNS DoS attack

This type of attack is the focus of this study. I deployed a mitigation model that focused on the factors that the attackers depend on to achieve their aim, this mitigation model worked on eliminating or reducing the possibilities of the attack factors to make this mitigation model a proactive solution rather than a defending solution.

In this type of attack, the attackers exploit two points to successfully run their attacks. These two points, or vulnerabilities, are represented by the ability to send packets with a spoofed IP address, and the existence of thousands of open DNS resolvers, which accept and serve any query from any source. These open resolvers reflect the response to targeted victim and floods them, and can be made even worse by the use of amplification methodology, which is very common (Jose & A., 2014).

Usually attackers send DNS queries with spoofed IP addresses, sent out from different servers. They could be open DNS servers, and some of them could also be from hijacked well-known servers. When the servers receive these queries, which come from different clients, they look for an address to send the reply to, which, in this case, is the address of the targeted victim. Then that attacker tries to amplify their attack, aiming to take down the site or to make the attack more complex, so they try using DNS security extension features, EDNS0 and DNSSEC (Keyu Lu, Zhengmin Li, Zhaoxin Zhang, & Jiantao Shi, 2016), which lets the attacker create a reply much larger than the query that has been sent. The attacker may also use query type ANY, which will make the targeted server look for all possible information that the server can find and send it to the requester (Takeda, Musashi, Sugitani, & Moriyama, 2013). The attacker can exploit these formal features to amplify their attack and make it more effective against their victim. By using these amplification factors the attacker can send a query of 500 bytes and get a response around 3000 – 4000 bytes, so it is worthwhile from attacker's point of view to use these features to escalate the effectiveness of the attack.

3.2.2 Cache Poisoning

A cache poisoning attack is one of the DNS DOS attack categories where attackers aim to redirect victim queries from a legitimate website to a malicious website that looks exactly like the real one.

This could be happening when the attacker somehow gains access to the DNS server. When the attackers breach the server, having access means they can modify the DNS server's cache to redirect the query to a malicious website. The attacker may also build an open DNS server, which is created specifically for the purposes of the attack, that connects to well-known servers in an attempt to spread fake malicious links and websites to attract more victims to use those websites and launch different types of attacks (Wu, Dang, Zhang, & Wang, 2015).

Attackers also use another approach where they build a malicious website then send a query to the well-known DNS server about that website, aiming to link this website to a well-known DNS cache server. If the attacker takes a sustained approach, the malicious website will spread to other DNS servers which is considered as an important step in the preparation plan. After that, the attacker will remap the IP address of the malicious website with a well-known website; for example, online trading where victims use their credential information to make purchases, and once the victims use their credentials, the attacker has that information, which can be used to steal the victim's money or identity (Naqash, Ubaid, Ishfaq, & Fazal-e-Hadi, 2012). Figure (13) illustrates an attack diagram and the steps from the beginning to the hacked stage.

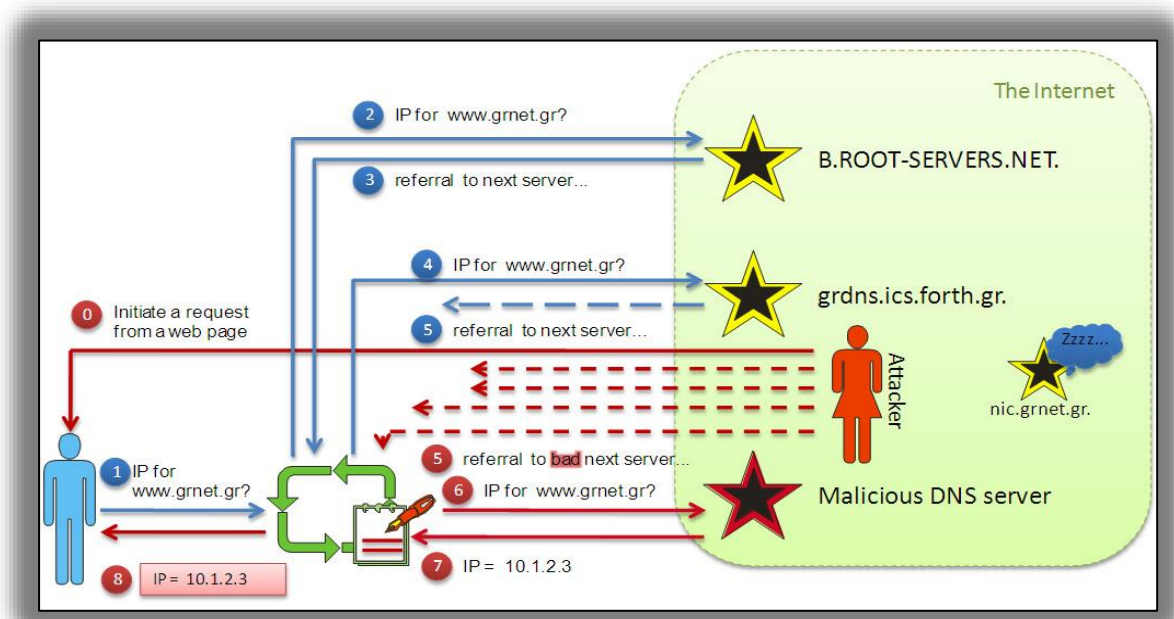


Figure (13) Cache poisoning stages (Saragiotis, 2009)

3.2.3 DNS Tunnelling attack

DNS Tunnelling is an approach to embed data or protocols inside a DNS query to get access to a specific computer, which is considered step one of malicious activity. Usually, DNS Tunnelling is used to bypass the restricted rules in proxies. The DNS traffic passes the proxies without inspection and is allowed to pass through the firewall as an infrastructure service (Aiello, Mongelli, & Papaleo, 2013). Most network administrators have a lack of knowledge about this topic and attackers exploit this lack to bypass the security appliance and firewall.

There are a lot of utilities such as Lodine, DNS2TCP and DNScat, used to create this type of tunnel. In this type of attack, the attackers create a tunnel, aiming to access the internal host as its target or as a stepping stone to a more advanced attack. Attackers use these tunnels as a cover to pass other protocols or scripts through the security appliance to attack their victims, as shown figure (14). Each one of these tunnel creation utilities uses different encoding methods to code and modify the data or payload embedded inside the DNS query. For instance, Base32 Encoding (5 Bit) to encode, Base64 Encoding. There are also other types of encoding but they do not work with all available DNS software.

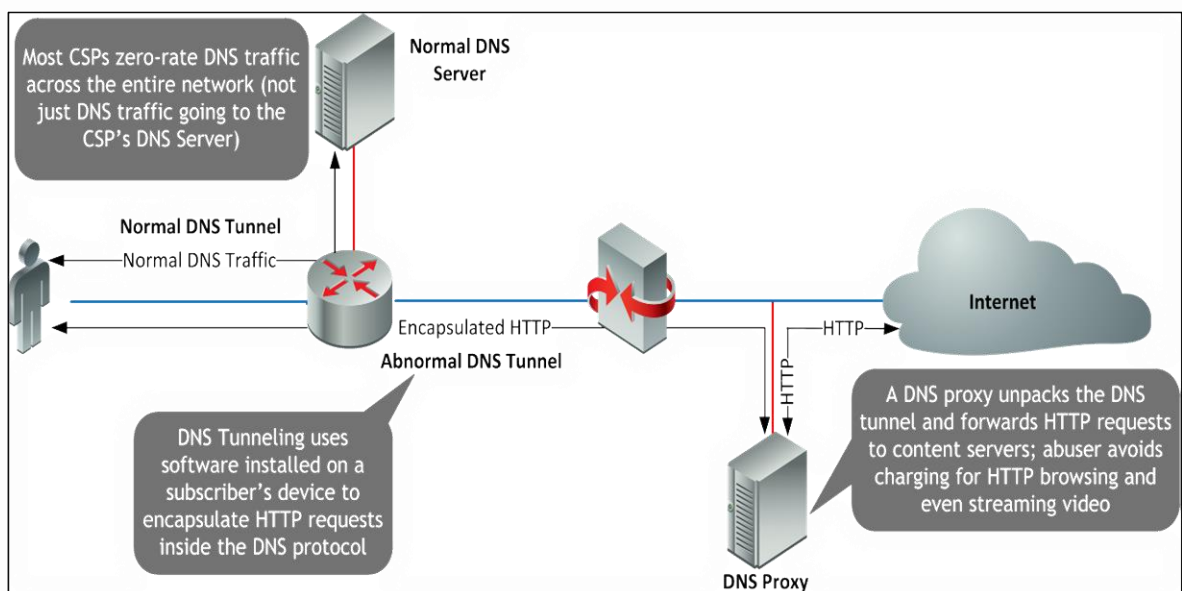


Figure (14) DNS tunnel idea with illustration comments (Phuonglm, 2014)

3.2.4 DNS Hijacking attack

This type of attack is similar to a cache poisoning attack but the attacker compromises the victim's computers with malware, aiming to change the TCP/IP settings to redirect the victim's internet traffic or DNS traffic to the attacker's server rather than the real DNS (Janbeglou, Zamani, & Ibrahim, 2010). After the attacker changes the DNS, they can easily learn a lot about the victim's traffic, which could lead to devastating results.

This can also be done by changing the DNS record in an attempt to redirect the client's traffic (Sakurai & Ushirozawa, 2010). Redirecting client traffic from a generic website to a malicious website that belongs to the attacker will reveal sensitive data - see figure (15) which illustrates how DNS hijacking works. In this scenario, the user enters "Wordpress.org" in their browser, so the browser needs to know the IP of Wordpress.org, then sends a request to ask for the website. The rogue DNS replies with the fraudulent website, as shown in this scenario.

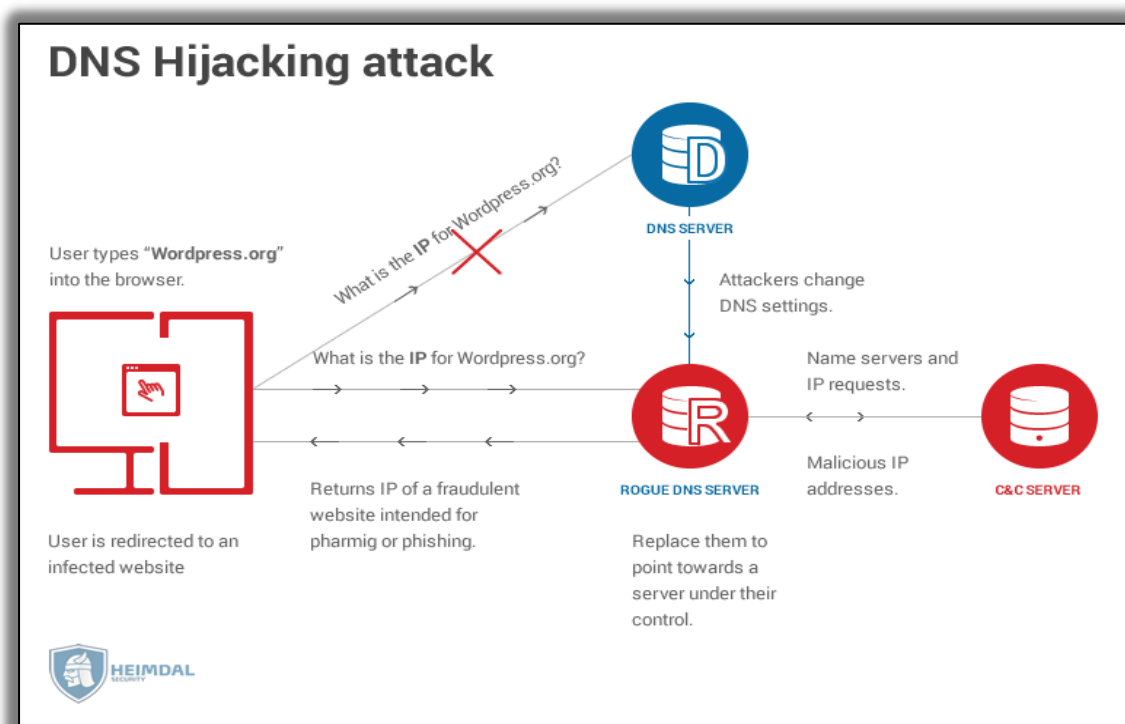


Figure (15) DNS Hijacking attack (Heimdahl ,2016)

3.2.5 Basic NXDOMAIN attack

NXDOMAIN, is a response methodology represented by a non-existent message from a DNS server. When the client sends a query to a DNS server about a domain name, the server will try to find it, and attackers exploit search mechanisms to exhaust server resources by asking servers to search for non-existent websites.

In this type of attack, the attackers exhaust a server's resources by sending hundreds of queries for non-existing domains and flooding the server with requests. The server stays busy looking for the non-existing domains, during which time the server cannot handle or receive requests from legitimate clients. In this way, the attackers achieve their aim to take down the server, or to at least prevent the server from responding efficiently.

There are also other approaches that could be used to hijack and exploit this feature to launch different types of attacks. This could occur when an ISP, third party service provider or open DNS server tries to exploit this feature and send, for example, some advertisements instead of sending a typical message saying the domain does not exist. These advertisements may be hacked or hijacked and running malicious scripts to reveal sensitive information or install malicious malware to change the DNS, aiming to redirect traffic towards the attacker's server.

Another approach is, these long non-existent domain queries could be used by the attacker to pass embedded encoded messages inside a long query. These types of queries could easily pass through firewalls without deep inspection.

There is also another complex approach of building software responsible for sending a huge number of random non-existent domain queries and inside this software embedded Domain Generating Algorithms (DGA) will work to build a botnet communication network between bots and the C&C.

3.2.6 Domain Lock-up attack

Domain Lock-up is a DDoS attack which exploits TCP mechanisms to run malicious behaviour. Attack methodology relies on either a half-open TCP connection with the targeted DNS server to keep it engaged for a long time, or sending random junk packets. In the half-open TCP connection

scenario, the attacker will either never complete the request or deliberately respond very slowly to cause long delays with the aim of exhausting DNS server resources, and as a result keeping the server busy and unable to accept legitimate requests (Ghafir & Prenosil, 2015).

In the second scenario, in which the attacker sends random junk packets, the attackers may set up a server especially for running this type of attack. When the targeted DNS server receives the query from the attacker's server, it will normally send a reply, the attacker's server will continue sending random junk packets, aiming to keep the targeted server busy either waiting to complete the connections or processing high number of these junks packets. So it will be slow to respond, and this methodology is repeated in a random and slow way.

Figure (16) illustrates how the attacker exploits a three-way handshake TCP mechanism. As can be seen in the figure below, at stage #3 the DNS server will expect an ACK from the supposed clients, and it is at this point the attacker tricks the server to keep it waiting for the ACK that will never come.

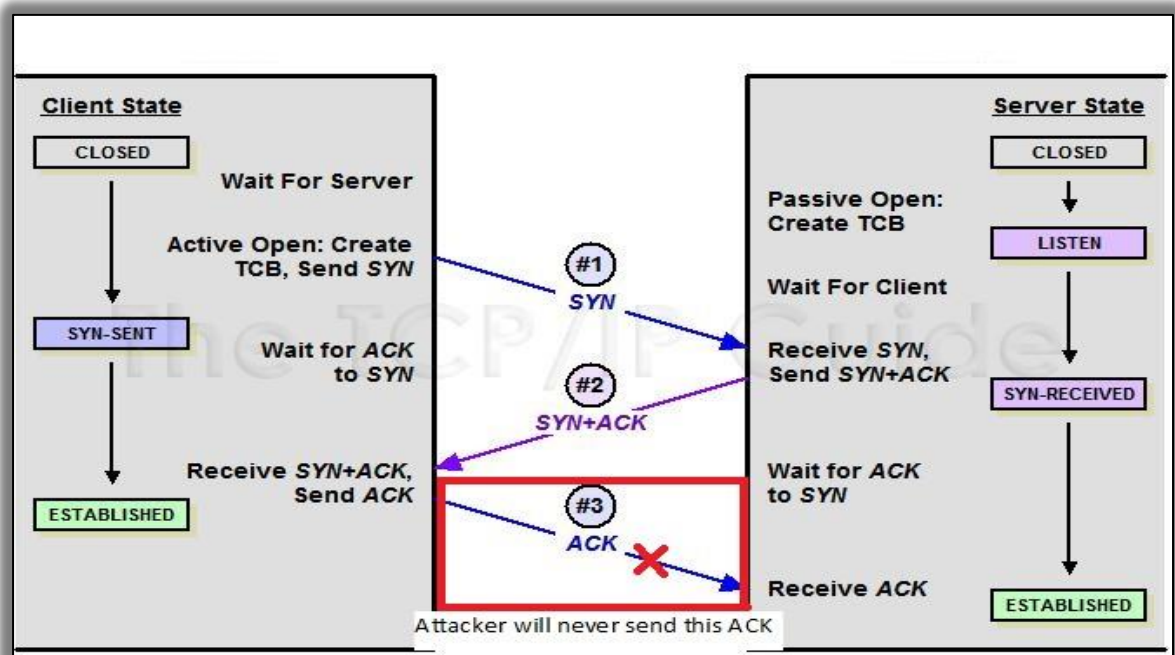


Figure (16) Attacker never completed stage 3 to establish a formal connection (Arbor, 2012)

3.3 Summary

This chapter discusses some concepts and terms of DNS services and the hierarchical architecture of the DNS domain level. It also discusses most of the common DNS attacks, and illustrates the mechanisms of those attacks, which help the reader to understand how attackers exploit public internet services to run malicious attacks aimed at revealing sensitive information or stealing users' data. From this chapter we can understand that not only unknown websites can be dangerous, but even public internet services can be exploited to run malicious attacks.

Chapter 4

Thesis Methodology

4.1 Research area and thesis methodology

Through the literature review and according to many security reports and statistics, we can see that a DDoS reflection/amplification attack is one of the common attacks that effects and disrupts online services. Attackers use their knowledge to exploit some open online services to abuse and disrupt public services. In the past, money was not necessarily the first motivation factor, but in recent times money has become one of the biggest motivations in the different types of attacks. Although many good solutions have been adopted, the ratio of attacks is still high and very effective. Figure (17) below shows the DNS Reflection DDoS attack mechanism and attack stages that attackers follow on to run this type of attack.

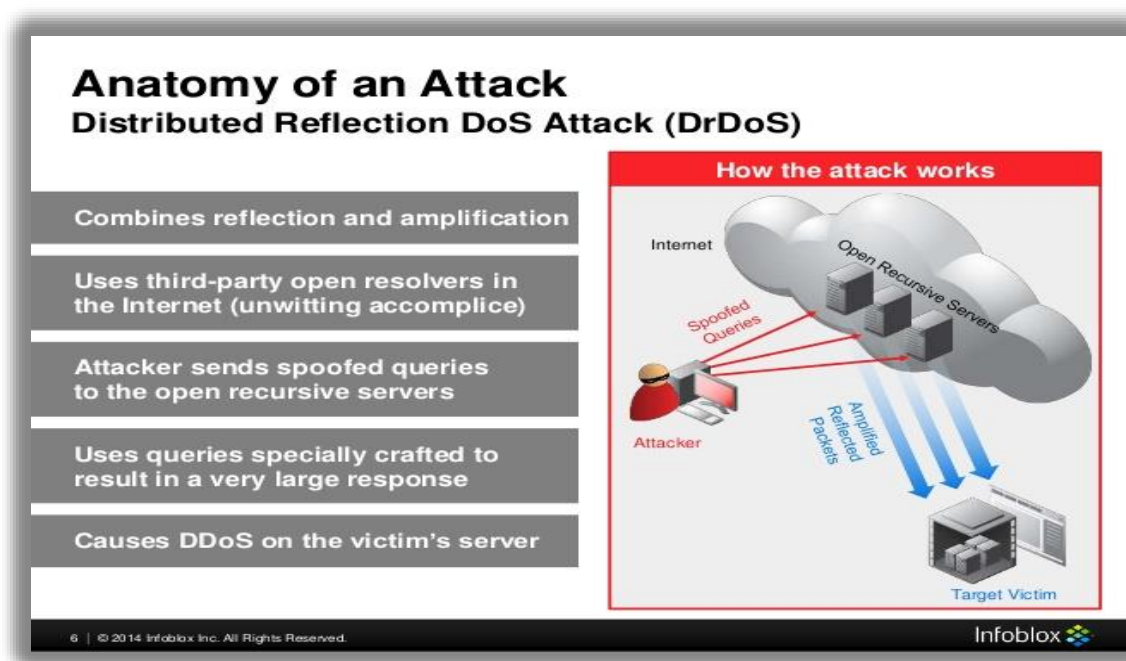


Figure (17) DDoS reflection mechanism (Infoblox, 2014)

4.2 Attack Methodology Representation

DNS Reflection attack depend in two factor in his methodology, those two factors that the attacker depends to run the attack represent the key points that make the adversary run a success attack, and I discuss those factors as following:

1. The ability to send packets with spoofed IP address

This issue is crucial because most cyber-attacks exploit the ability to send spoofed packets. To access the internet, any computer should be connected through an ISP. ISP's are responsible for passing clients' traffic, so are a gateway to anyone, and as such, issues should be solved at this

point as much as they can. Any attacker can either exploit a victim's computer by using malicious software, such as a Trojan horse, or use his own computer under disguise (although this is rare). In both situations, when the attacker runs a reflection attack, he will send a spoofed packet. These packets will normally pass the first checkpoint, which is ISP router, at which stage the packets should be prevented from passing further through the internet network. This would add pressure on the attackers to either stop the attack or send the packets with their real IP, which would risk their identity being disclosed.

2. The ability to exploit open DNS resolvers

In normal situations or well-configured servers there should be a strict policy around how these servers handle the incoming query and which query should be accepted, including the source the query is coming from. While leaving open servers without limitation policy makes them prone to attacks. If this problem is solved, and DNS servers organised in a way that only registered and authorised queries can pass through specific servers, this will help a lot in mitigating these and other types of attacks, too.

4.3 Research hypothesis

Research hypothesis represents a researcher's ideas and scope that they are willing to follow to achieve the research's goal. This could be either in theory methodology by following other researchers' experiments and reading results or even using some simulation software, or by practical methodology by executing an experiment either in a real environment or in a lab using the same environment and devices. In some cases, there are some restrictions preventing the researcher from conducting his experiment live in the real environment, such as in this study, in which I could not run a live DDoS attack against an ISP or an organisation, so I used the same routers, switches, servers, tools and applications that are used in a real network, to get as close to a real result as possible.

In this study I worked on a DNS reflection DoS/DDoS attack, and I propose a proactive solution to prevent foreign packets or unwanted packets from reaching or passing to my DNS server, which is, for the purposes of the study, the targeted server. In this study, the research hypothesis is represented by a defense plan or multi-tier defense plan. The following steps will explain the research hypothesis represented by each defense line and the expected role for each one of them. Details will be discussed more fully in following chapters:

1. **Stage one:** This stage considers how we can stop the attacker from using a spoofed IP address, which makes the attacker safer. I used uRPF in strict mode at the ISP router to stop the spoofed packets and force the attacker to use his real IP.
2. **Stage two:** The attacker hacks a group of computers, servers or open DNS servers and exploits them as victims to run his attack. The queries that are sent by those victims are considered as legitimate requests, so will pass the first line of defense represented by stage one. I used uRPF loose mode, which is different to the strict mode that I used in the ISP router, and I used it at my edge router to prevent any foreign packets that do not belong to my inside network or even the outside network, from gaining access. By “inside network” I mean my private network, and “outside network” represents other organisation branches, trusted DNS servers, customers, and trusted servers. For example, I wanted to let my DNS server communicate with the Google DNS server or one of my trusted customers, so I let the packets that source an address from the Google DNS Server or any trusted customers pass. Otherwise, even if the packet is a legitimate one, it will not pass.
3. **Stage three:** This stage is represented by splitting my edge router into three zones. These zones worked to secure and control the movement of allowed incoming traffic between zones. In other words, which traffic was allowed to access which zone. The zones are as follows:
 - A. Inside Zone: represented the private network part of the organisation network.
 - B. Outside Zone: represented the public network or the ISP router that faces my edge router and passes internet traffic from and to my edge router.
 - C. DMZ Zone (i.e. demilitarised zone): represented the middle zone between the Inside and the Outside; both zones of which need to access to the DMZ.These three zones had control over passing traffic between all zones through classifying the traffic. These classifications were associated with the zones to achieve a better security plan.

4.4 Research Methodology classifications

In general, there are two research classes and each one has different criteria and uses: qualitative research, quantitative research or a mix of both.

Quantitative research is a quantity research for a specific topic or area, and its focus is on getting real measurements, counting, or statistical results of what the research discovers through practical experiments, and then it is saved in a readable format as a statistical document.

Qualitative research is a broad exploratory research to describe the topic in detail and it is very useful if it is conducted at the beginning because it will give the researcher a good understanding of the whole picture about the research topic, and other work that has been done in the same area, along with others' opinions. An example of this type of research is a literature review for a specific topic (Explorable, 2009).

This study has used a mix of both qualitative and quantitative research, although the practical experiments and the results collected mean the study tends to be more quantitative. I used qualitative research in my literature review of all DNS DoS/DDoS attacks, and then quantitative research by carrying out the practical part at Unitec's labs. Real DoS/DDoS attacks have been run and real results gathered by using different tools for the attacks, such as NMAP and HPING3 from Kali Linux OS. PRTG software was used for monitoring purposes to track attack traffic and Wireshark was also used to get more specific results from the lab environment, which was the same as a real environment in our networks today.

The Waterfall Model has been followed in this study as a guideline to show the necessary steps that should be conducted to pass from the beginning to the end and realise the results that this study aimed to achieve. See figure (18).

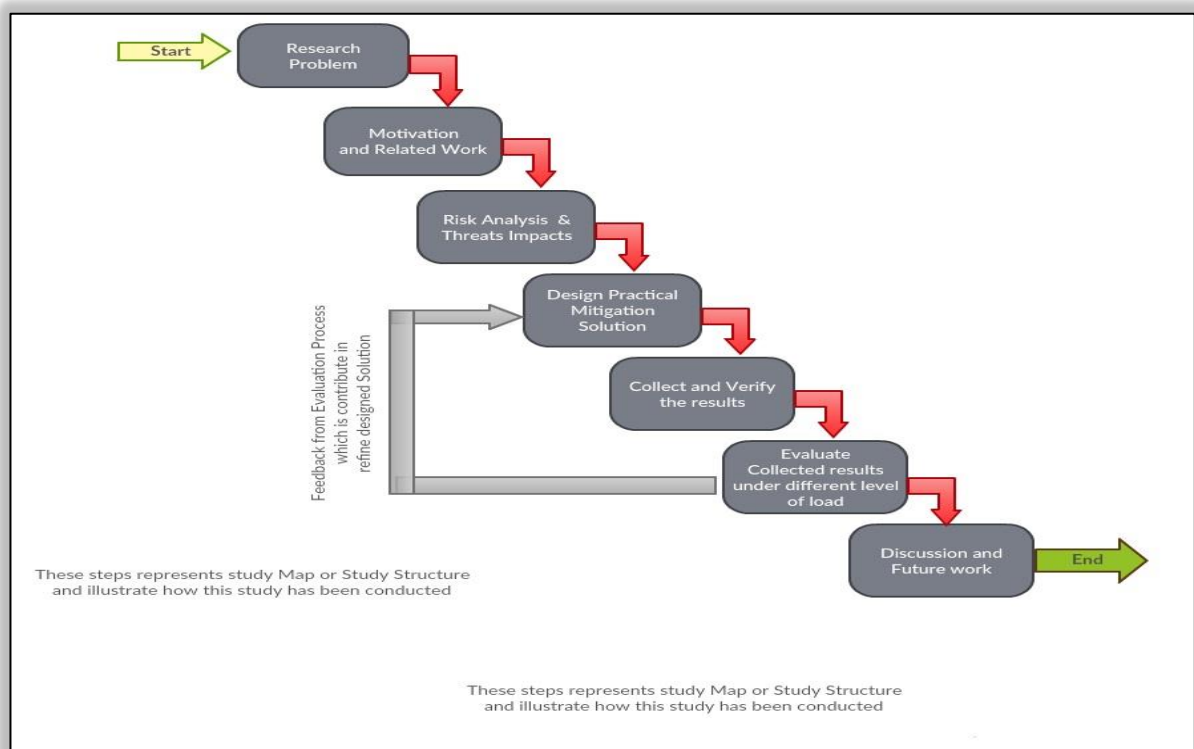


Figure (18) The Waterfall Research Model that has been followed

The following stages represent how the study was conducted, and each is briefly defined:

Stage 1: Research problem - includes recognising research gaps or research problems which will need to be worked on.

Stage 2: Motivation and related work - covers motivation factors for working on the recognised gap in stage one, and included considering related work done in this area.

Stage 3: Risk Analysis & threats Impacts, include analysis attack severity and also size of damage impacts severity too and also include attack factors

Stage 4: Design practical environment - included designing and building a lab environment as near as possible to a real environment to represent the problem.

Stage 5: Collect and verify the results – carrying out of tests, verification processes and modifications, and collating results.

Stage 6: Evaluate collected results – after practical results had been collected they were evaluated. The aim of gathering more trusted results is gained by exposing the results to different levels of loads and conditions.

Stage 7: Discussion and future work - in this final stage, the results are discussed along with what future work could be undertaken to support and enhance the suggested solution.

4.5 Research hypothesis test methodology

There is more than one approach to test researcher hypotheses but the most recommended one is the use of a testbed, which provides a real result. Using real devices on a testbed as a practical methodology gives the researcher more realistic results than depending on theory or simulation. Using a real practical environment also provides the chance to conduct an evaluation process to test the results, which means there is the chance to verify and evaluate the hypothesis under different load levels and conditions. This helps to not just get an accurate result but also to understand design performance and changes under varying conditions.

In this study, I used a testbed consisting of the following:

1. Cisco Routers and Switches
2. Microsoft Windows server 2012 as a DNS server
3. An attacker machine with Kali Linux OS with all attack tools
4. Windows 10 with PRTG software installed to monitor the attack and record traffic patterns and measurements; adding different types of sensors supported by SNMP to get as much of a reading possible, to support the mitigation idea.
5. Normal computers, representing legitimate users sending normal traffic.

4.6 Research questions discussion

Research questions are key concepts of any research, representing the problem or the gap in the research area. Although it may be an area already worked on by another researcher, an attempt is made to solve the problem via another approach to get another practical solution which produces more options which could suit more categories of clients. Research questions could be one question or could be divided into more than one question to cover the research area more broadly. In this study, I divided the research question into four questions to help to give a better understanding and cover all the factors that the attacker depends on to achieve his attack. We cannot stop the attacker from launching his attacks but we can work on reducing the availability of attack factors and add more difficulty and obstacles to mitigate the attacks. This is what we aim to achieve when working on cyber-attack mitigations.

The possible questions that are related to the research area in this study and could support and be a guide to analyse and face the attack, are highlighted below:

- **How can DNS reflection attacks be mitigated?**
Focus on attack mechanism itself.
- **How could the attackers exploit Open DNS Resolver?**
Focus on open DNS configuration and restrict policy.
- **How can we recognise attack traffic from legitimate traffic?**
How the targeted organisation can isolate and deny malicious traffic.
- **Where will be the most effective mitigation technique implemented?**
Focus on mitigation model compatibility and location that each mitigation model should be deploy.

4.7 Research data gathering and measurements

In this study, I have depended on real networks and real devices. I installed everything in the labs and launched a real attack. Through each stage, I collected the readings and data by using the following approaches:

1. Using show commands from inside CISCO routers and switches for verifying
2. Using SNMP protocols and NetFlow V9 service inside CISCO devices to collect and send the data to PRTG monitoring software
3. Using ping commands from inside Kali Linux for test purposes
4. Using Wireshark software from inside Kali Linux
5. Using PRTG software to collect the following:
 - A. Traffic patterns before and during the attack
 - B. Delays in the attack
 - C. How mitigation technique controlled the attack by showing the reduction in traffic percentage at each stage from mitigation plan
 - D. CPU usage
 - E. Memory usage
 - F. Temperature

Therefore, all the results and readings are actual data gathered from the testbed network which I suggest represented a real network scenario, and I used BGP protocol between the routers because it is the preferred protocol that is used between companies and their ISP through connecting to the ISP network.

4.8 Summary

This chapter discusses study methodology and types of research methodologies. Quantitative and qualitative research are discussed, plus a mix of both types, which was used in my study. My study hypothesis is discussed in this section to illustrate what has been assumed and then proven through the practical part of the study.

Chapter 5

Mitigation techniques and verification process

5.1 Network Diagram (Testbed)

In this study, the verification approach depends on a real network that has been connected in a lab and the launch of a real attack. The network diagram shown in figure (14) shows three main routers which routers represent the attacker, the ISP and the edge router for the targeted organisation. Those routers are connected using BGP protocol, as used in real networks.

Windows Server 2012 with the latest update was installed on the organisation side to represent the targeted DNS server. There was a need for additional servers to represent two organisation branches; one for open DNS resolver and the other for a trusted DNS server, such as Google and Yahoo. These additional servers represent the real scenario and their IPs have been used to send packets to the DNS server to check accessibility, who can access the server, and who is not according to the security plan. The table below represents the devices that I used in my testbed:

Device	Model	Quantity	Notes
Routers	Cisco 2811	3	Representing Attacker, ISP and Defender
DNS servers	Server OS 2012	2	DNS server
Attacker	Laptop with Kali Linux	1	Computer with Kali Linux installed to run real attack
Monitoring server	Normal	1	Server with free PRTG software support up to 100 sensor

Table (2) Devices that were been used in the lab

One last server worked to monitor network traffic and collect data by PRTG software, which is a professional software for monitoring network traffic through multiple sensors.

I configured Netflow v9 inside the router as a sensor to send traffic data to the PRTG and I configured the PRTG and added Netflow v9 to receive the sending data. There were four sensors that read and sent the live traffic, as follows:

1. Sensor read the traffic OUT from attacker router
2. Sensor read the traffic OUT from the ISP router
3. Sensor read the traffic IN to the organisation's edge router (Defender)
4. Sensor read the traffic IN to the DNS Server

By monitoring the traffic at these stages, we can create a clear picture how the network works. I created my scenario of an SMB organisation called “ABC” that had been under DNS Reflection attack before, so the organisation's CEO asked the IT team to suggest a proactive solution within existing devices, due to limited budget, to protect the organisation from these types of attacks in the future.

The challenge was to find a cheap, simple solution that could be applied on old and new equipment, and support different vendors, on a limited budget, to protect the organisation. Furthermore, the organisation has two branches connected to Head office through the internet, as shown in the figure (19) below.

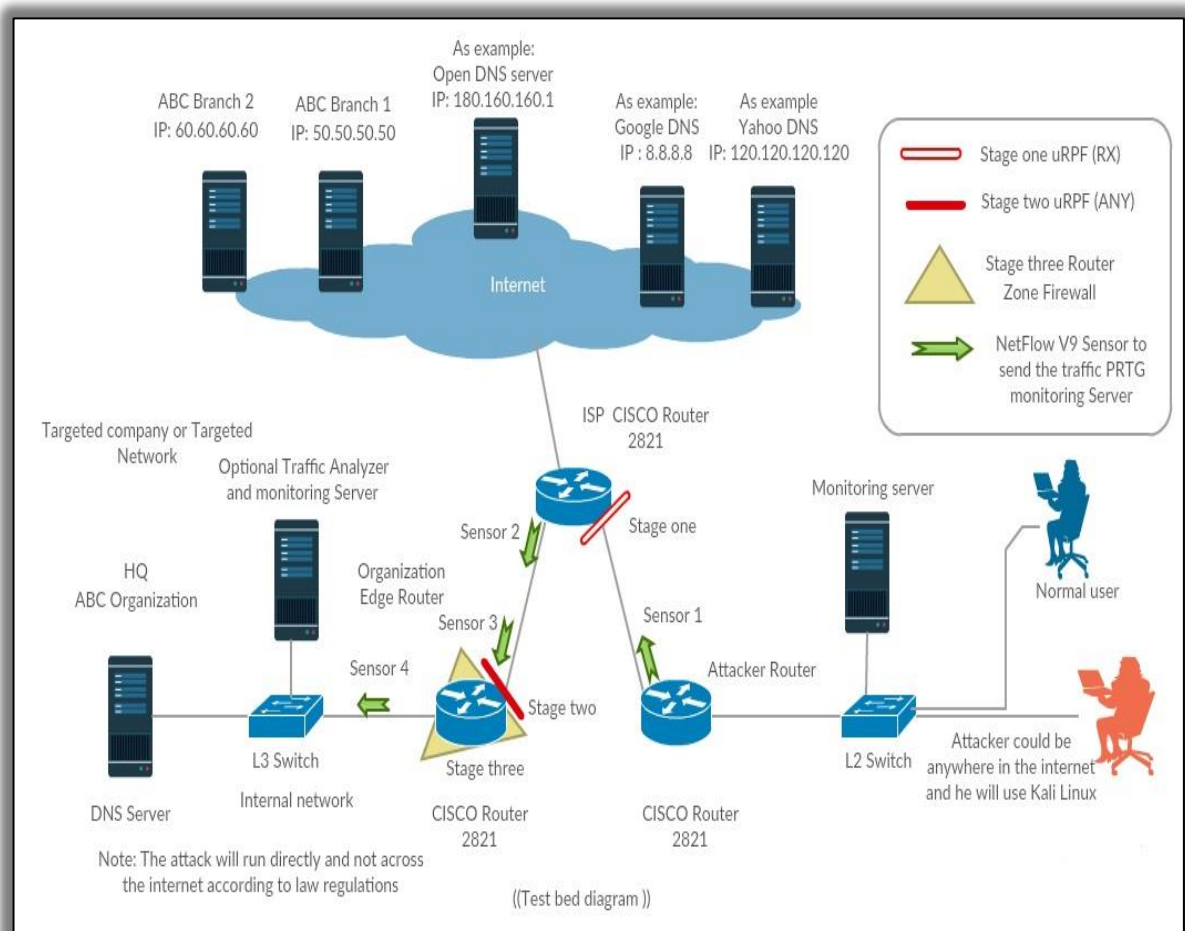


Figure (19) Testbed diagram

5.2 Monitoring tools

The tools that were used to support my study were:

5.2.1 PRTG

This is a powerful monitoring software that is used to monitor and collect network data statistics which help the network administrator monitor their network, and also monitor the services that are running on that network, by providing nearly real-time recordings for traffic that passes through the network. Results are shown to the network administrator graphically.

PRTG software supports a lot of vendors and a lot of protocols that help an administrator to monitor their network in a very efficient way. It also offers different types of sensors with the ability to report events in different methodologies and also send a notification via email or SMS. Most of the people who have experience with this software are happy with performance and find it very helpful.

5.2.2 Wireshark

This is a powerful packet analyser which can be used to analyse hidden network issues and troubleshoot normal daily network issues. It provides many options and filters which help to analyse network performance and also recognises that it is real traffic that has passed through the network. Wireshark provides a wide range of services, such as packet captures, packet filters for different types of protocol, port monitoring and network taps. The versions that are after 1.4 also have the ability to put a wireless interface controller in monitoring mode.

5.2.3 CISCO Show Commands

CISCO is one of the best vendors for network devices and equipment in the world, so some show commands have been used, which helped to get results from inside the router and compare them with results that I collected from PRTG and Wireshark. Having more than one source for collecting statistics ensures the reliability of the results.

5.3 Attacking tools

This section defines the attacking tools:

5.3.1 Hping3

Hping3 is a great TCP/IP packet generator tool, which has the ability to create different types of packets, like ICMP, UDP etc, which are mostly used for penetration tests or for educational purposes to analyse attack factors. It can also be used for a scan, in listen mode.

Hping3 as a tool is a good reliable option, making it a great security testing tool. It has the ability to use spoofed IP addresses, random destination, random source, modify MTU and many other options, such as supporting packet fragmentation.

5.3.2 Nping

Nping is also a very good packet generator, used as a troubleshooting tool to recognise network issues and measure network response time. It supports a broad range of protocols, with the advantage of giving strong full control on generated packet headers.

It is also used in real attacks for educational purposes and as a penetration test for assessment purposes.

5.4 Mitigation technique stages

In this section, I will explain the suggested mitigation technique; how it works and what the task or role is for each stage. Each stage is responsible for denying or mitigating a specific part of the attack because, as we know, the attacker could come in different guises.

I configured a Netflow V9 service as sensors in all the devices to record and send the traffic to the monitoring software PRTG. These readings were used to verify suggested mitigation techniques. Figure (20) shows a sample of this configuration. The same sensor is configured with the same version of Netflow V9 in PRTG, to be able to understand the information the software received.

There was also a need to configure the SNMP protocol in the Defender Router, so other types of information could be monitored and collected, such as CPU, RAM, temperature, fan and power supply, which helped to evaluate the defender performance, both in its normal situation and when it was under attack. The differences were then compared to ascertain how much load the router could manage.

```
ATTACKER#sh run
Building configuration...
CUTTED .....!
ip flow-cache timeout active 1
ip flow-export source FastEthernet0/0
ip flow-export version 9
ip flow-export destination 10.10.10.2 9999
```

Figure (20) A sample of NetFlow V9 configuration

5.4.1 Stage one - Shown in the network diagram as

This part of the solution was applied at the ISP router, and could be deployed as uRPF in strict mode to stop sending any spoofed IP. This should be applied at the edge or the interface that faces the customer's edge or routers. Applying uRPF will not stop only DNS Reflection attacks, it will stop many other types of cyber-attacks that are dependent on sending spoofed packets.

ISP companies need to take responsibility and play an active role in mitigating spoofing packets because they are the first gate that these packets pass through. After the packets bypass the ISP's router the damage has occurred and they are no longer easy to detect. With many types of attacks, such as flooding attacks, once their packets reach the victim edge router most solutions will be ineffective because the attacker has already flooded the link, server, or router and not a lot can be done at this stage.

uRPF in strict mode will work at this stage to check if the source of upcoming packets is registered to come from this source and specifically from this interface. If yes, the packets will pass, if not, the packets will drop. The practical experiment in the lab worked 100% as it should, without any exceptions, in relation to the number and size of the packets. I sent different number and size of spoofed packets and it dropped all, again regardless of the size of the packet.

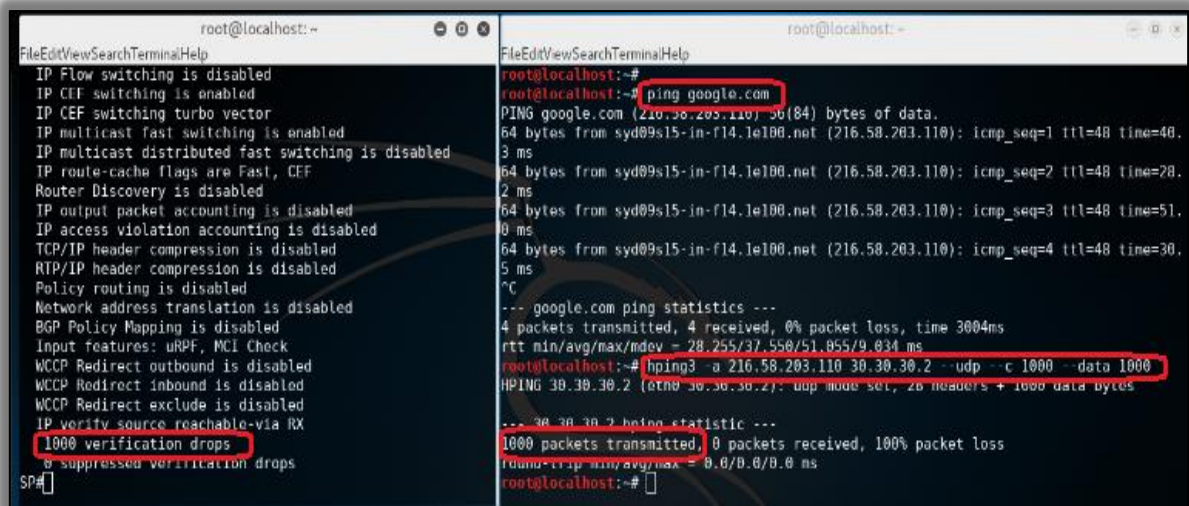


Figure (22) Sending the attack from random IP

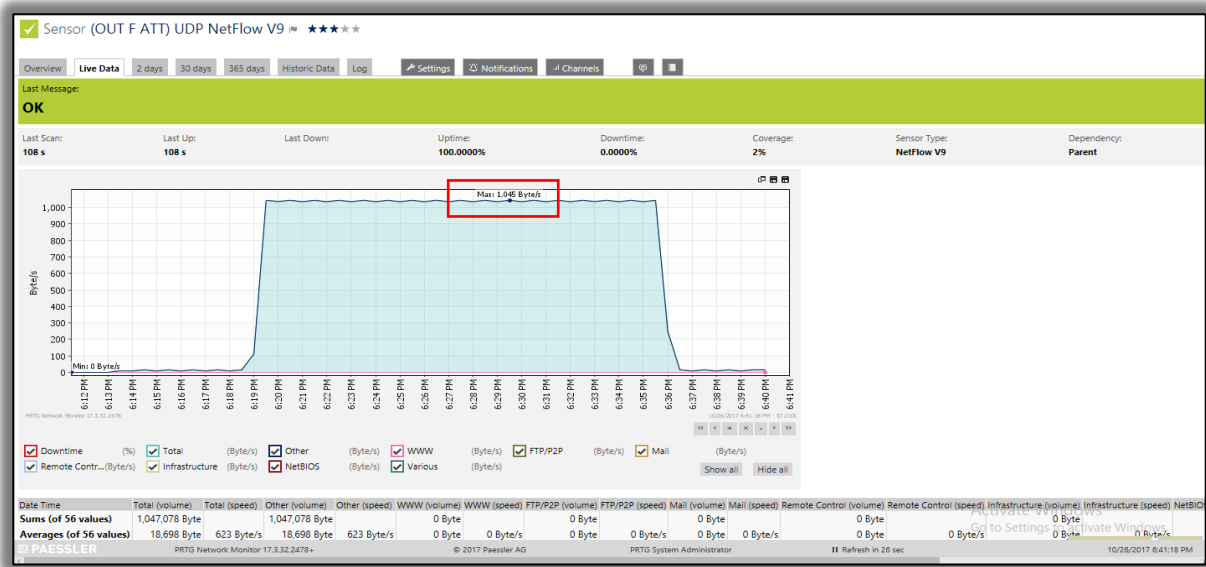


Figure (23) Traffic OUT from the attacker router

Also, as we can see from the figure (22) above, the router efficiently dropped all spoofed packets by using uRPF in strict mode and will prevent the attacker from using or sending spoofed address.

5.4.3 Stage two - shown in the network diagram as

At this stage, there are many solutions that could be applied. At the beginning, I tried different types of solutions, such as different types of access lists, but this did not give me the solution that I needed because by using access-list there are a need to manually add a lot of permission and deny

sentences, and from previous experience, there is a high possibility of mistakes by adding, deleting or following wrong sequencing as the router brain engine checks access list lines sequentially. Also, many other academic researchers have used it before in their suggested solutions, so I wanted to try using something different. After trying many solutions, I had the idea to use uRPF in loose mode, and after I ran some tests, it was working absolutely fine and compared with long access lists sentences, uRPF gave me the solution that I was looking for to achieve the goal efficiently in one line at the interface level.

To more explain more fully, I had a DNS server in the DMZ area and it should only be accessed by authorised people and employees in HQ and other branches, and also deal with and accept packets from other trusted DNS servers, such as Google or Yahoo.

Instead of creating a long list of access list lines to permit 1, 2, 3, etc, it is more efficient to deny all packets that are not related or do not exist in my network - this is what uRPF in loose mode does.

In a normal situation, when the router receives a packet, its behaviour is to check the destination address - if it exists, the router passes or route the packet. When I add uRPF in loose mode, it checks further and more closely inspects the source address for if it matches any entry in the routing table. If yes, it passes, but if not, it is dropped. Activating uRPF can be done by adding one sentence at the interface level at the edge router, as we can be see later in the router configuration.

The main role of the uRPF at this stage is to stop all foreign packets, as follows:

- Stop all misconfigured open DNS servers, even the good ones if, according to our security plan, those DNS servers are not authorised to connect to our DNS server. Also, it is safer for our network as a kind of proactive procedure because if those servers are hacked in any way, the attackers will try to hack or attack our server too, so it is a good idea to leave our server to deal with and connect to specific servers, which are trusted. It is also easier to monitor specific trusted servers.
- Simply stop all types of foreign packets from any sources from accessing our network. This is a good safety step, and allows only those packets that have an entry in the router.
- Also, with further configuration and by adding more options, such as “allow-default” for uRPF, then supporting it by matching the access list with the uRPF, this feature works as a security filter to prevent organisations’ employees from accessing any untrusted website and any other website they are not authorised to access. However, we must take into

consideration BGP protocol configuration because there are many scenarios that could be affected by this.

The CEO of ABC then raised the question, “What about trusted DNS servers and trusted servers that we need to connect out servers? How can we let our DNS server communicate with those servers, as they play a crucial role in supporting our DNS server?” The answer is by adding an entry in the edge router, which could be a static entry, for example, one for Google and one for Yahoo, so any packets coming from those trusted servers passes normally and foreign traffic drops. Any additional customers, trusted servers or partners, can easily be added by one or two static entries to let them pass without any issues, or even using another approach like matching access lists or any other approach.

5.4.4 Stage two verification

In this stage, I used the Google IP address to launch an attack. Figure (24) shows a basic routing table of both SP and Edge router, which I called, “Defender”. You can see, the routing table before I add an entry for another branch’s server or other servers which I want to let her access my server.

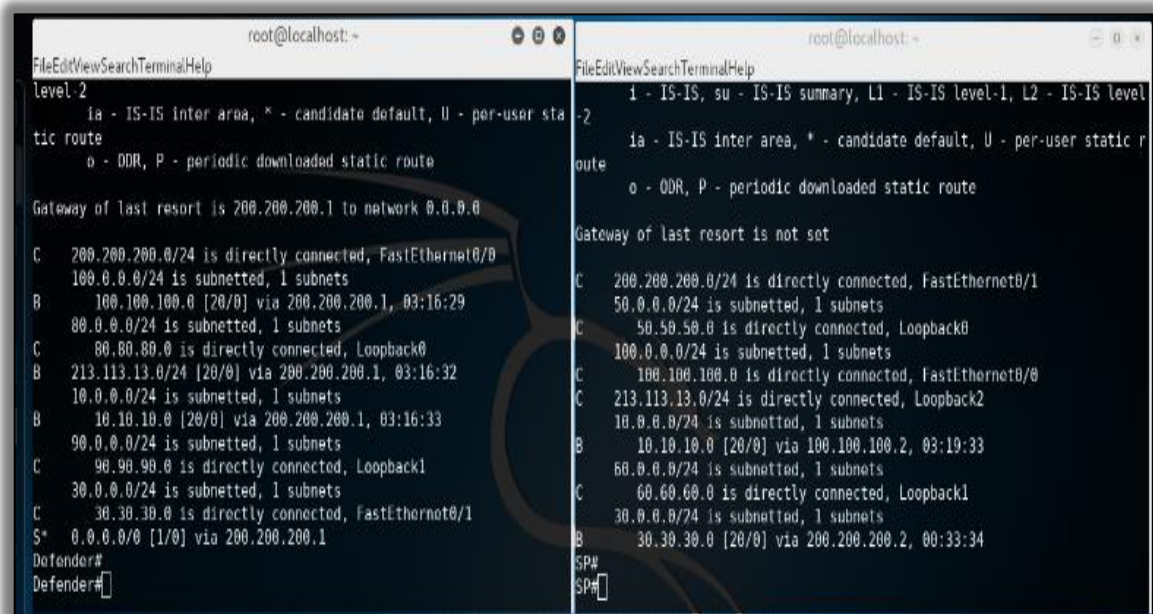


Figure (24) Defender and SP router table

The attack will launch with four sessions: each one sends 500 packets with a size of 1000 bytes for each packet.

Figure (25) shows the attacker using Hping3 to launch his attack by using the following command:

```
Hping3 -a 216.58.203.110 30.30.30.2 -- UDP -- c 500 -- data 1000
```

1. -a 216.58.203.110: represents the source address
2. 30.30.30.2: represents the targeted address
3. -- UDP: represents sending the UDP packets
4. -- c 500: represents sending 500 packets
5. -- data: represents packet size of 1000 bytes

To monitor the traffic by PRTG monitoring software, figure (26) shows the traffic out from the attacker router. Figure (27) show us the traffic into the defender router and figure (28) show us that no traffic passed to the DNS server.

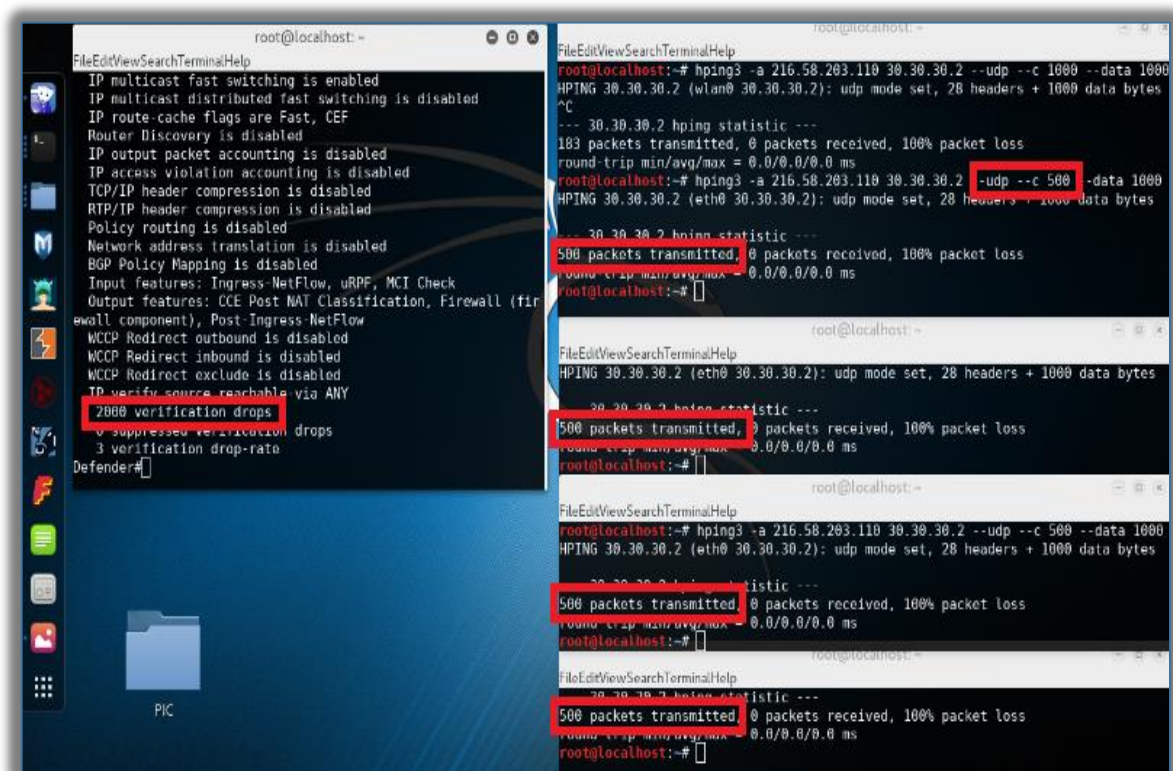


Figure (25) Sent packets on all sessions and dropped packets at Defender

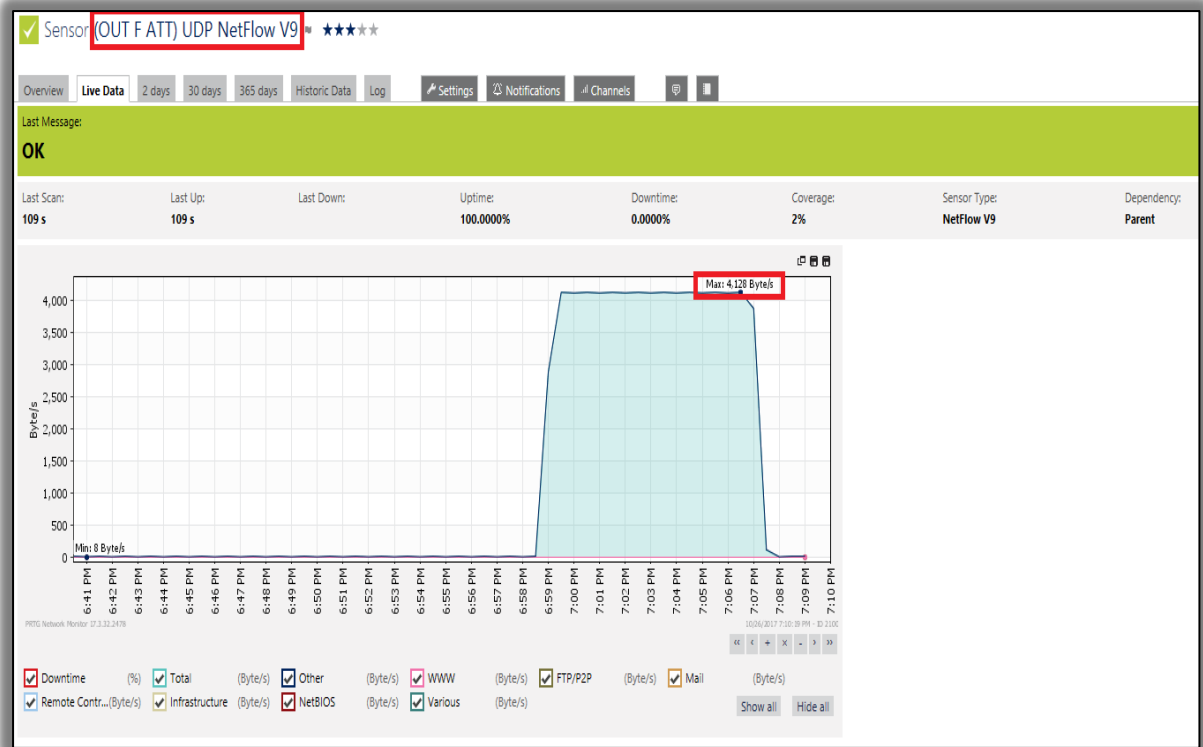


Figure (26) Traffic out from attacker router

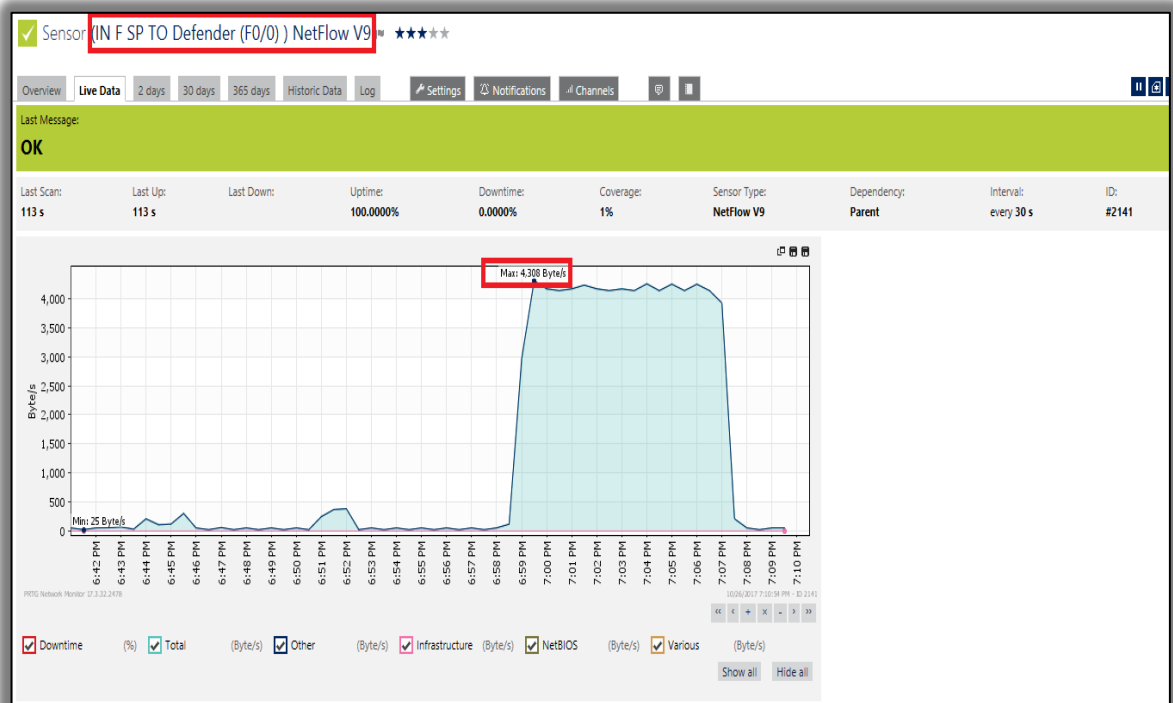


Figure (27) Traffic IN to the Defender

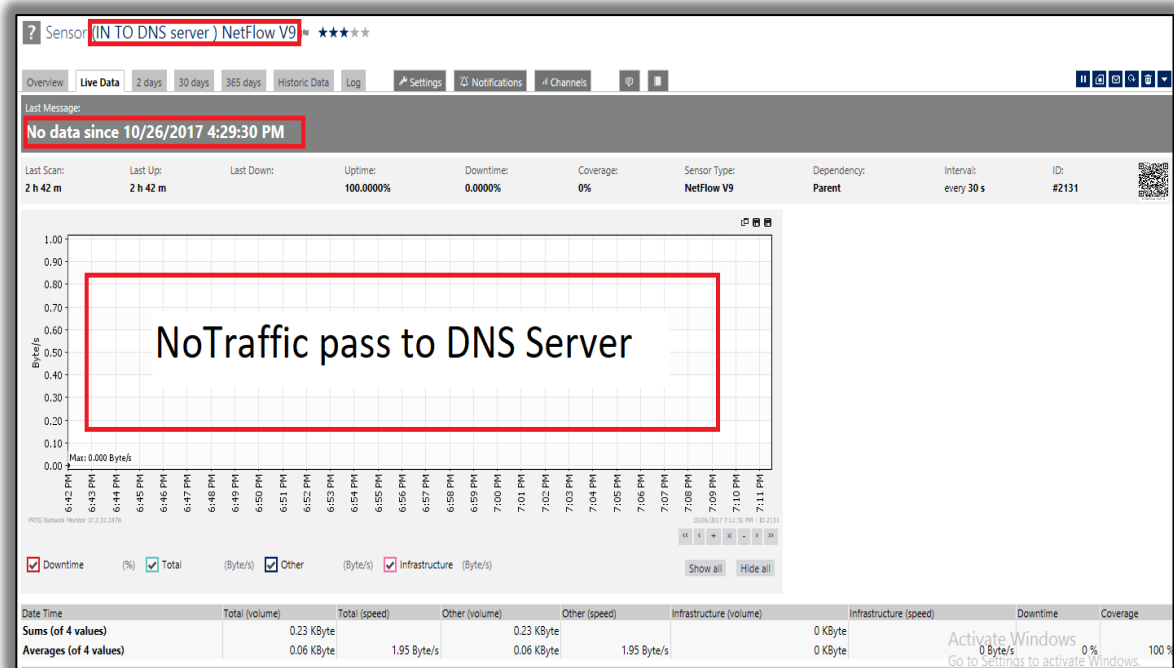


Figure (28) Traffic in to DNS server

5.4.5 Stage three - shown in network diagram as



At this stage, I used a Zone-based firewall to work together beside stage two to secure the organisation's edge router and as result, the network overall. There is a lot of detail about stage mechanism related to who can access what, and which protocol can pass. More options could be deployed to enhance performance but there was not enough time to cover all possible options.

The main idea in this stage is represented by splitting the router into zones to control and inspect all the traffic that passes between the zones. These inspections applied policies to permit and deny, according to the security strategy.

At the same time, all internal traffic passed forward to the internet and returned safely without delay. Below, I have detailed sequential steps to explain the mechanisms of this stage clearly:

1. Classify the traffic and create a specific class for each traffic type, as shown in figure (29).
2. Take action by creating a reaction policy, as shown in figure (30).
3. Create the zones and put the associated interface into their zone, as shown in figure (31).
4. Create zone pair and call the appropriate policy to be applied between those zone pairs, as also shown in figure (31).

```

class-map type inspect match-any ICMP
  match protocol icmp
  match access-group name SUPP
class-map type inspect match-any IP
  match access-group name SUPP
class-map type inspect match-any INTERNET
  match protocol http
class-map type inspect match-any UDP
  match protocol udp
  match access-group name SUPP
class-map type inspect match-any TCP
  match protocol tcp
class-map type inspect match-all ICMP-DNS
  match protocol icmp
  match protocol udp
!

```

Figure (29) Traffic classifications

```

!
policy-map type inspect OUTSIDE-DMZ
  class type inspect INTERNET
    inspect
  class type inspect UDP
    inspect
  class type inspect ICMP
    inspect
  class type inspect TCP
    inspect
  class class-default
    drop
policy-map type inspect DMZ-OUTSIDE
  class type inspect INTERNET
    inspect
  class class-default
    drop
policy-map type inspect INSIDE-DMZ
  class type inspect INTERNET
    inspect
  class class-default
    drop
policy-map type inspect DMZ-INSIDE
  class class-default
    drop
policy-map type inspect INSIDE-OUTSIDE
  class type inspect TCP
    inspect
  class type inspect UDP
    inspect
  class type inspect INTERNET
    inspect
  class type inspect ICMP
    inspect
  class type inspect ICMP-DNS
    inspect
  class class-default
    drop
policy-map type inspect OUTSIDE-INSIDE
  class class-default
    drop
!

```

Figure (30) Policies inside defender router

```

!
zone security INSIDE
zone security OUTSIDE
zone security DMZ
zone-pair security F-IN-OUT source INSIDE destination OUTSIDE
  service-policy type inspect INSIDE-OUTSIDE
zone-pair security F-IN-DMZ source INSIDE destination DMZ
  service-policy type inspect INSIDE-DMZ
zone-pair security F-OUT-IN source OUTSIDE destination INSIDE
  service-policy type inspect OUTSIDE-INSIDE
zone-pair security F-OUT-DMZ source OUTSIDE destination DMZ
  service-policy type inspect OUTSIDE-DMZ
zone-pair security F-DMZ-IN source DMZ destination INSIDE
  service-policy type inspect DMZ-INSIDE
zone-pair security F-DMZ-OUT source DMZ destination OUTSIDE
  service-policy type inspect DMZ-OUTSIDE
!

```

Figure (31) Zones and Zone pairs with their associated policies

When one of the employees from inside our network wants to access the internet, the packet comes to the router at the inside zone and the router creates a state full database and saves a copy of the packet within the database, then passes the packet to the internet across the outside zone. When the packet returns, the router engine checks its database at the outside zone edge for whether there is a copy from the returning packets. If there is, the packet passes, if not, the router drops it.

Other traffic which comes from the outside zone will be treated according to the policy rules; to either pass, deny or inspect. The figures above show the idea through the router configuration, and I have drawn a diagram to illustrate and show the concepts visually. See figure (32).

1. The green arrow represents internal traffic
2. The green-yellow arrow represents customer traffic authorised to access the DMZ Zone
3. The red arrow represents foreign traffic that does not have permission to access

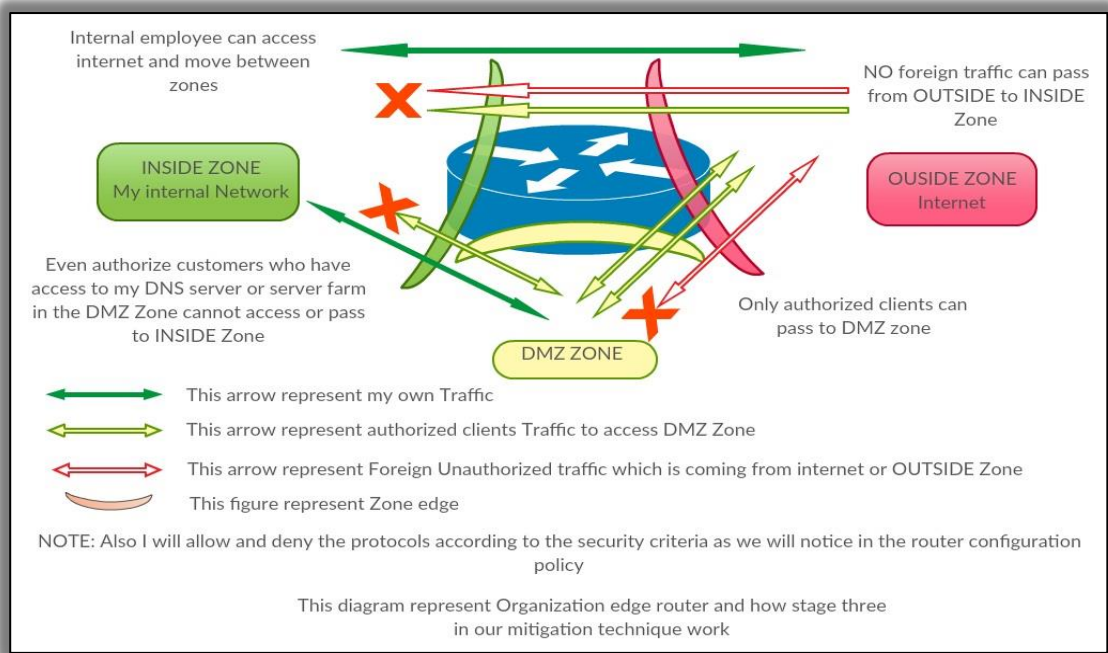


Figure (32) Illustration of router zones and how they work

5.4.6 Stage three - Verification

At this stage, when I applied a drop policy for any type of traffic, it dropped. So, for now, I sent ICMP traffic and the traffic showed in the PRTG software. This traffic can pass if there is an inspection or pass policy. "Inspection" means the packet has a copy in the zone database, otherwise packets are dropped because they do not meet passing criteria. So, this stage and stage two will support each other to produce an integrated solution at the edge router of the organisation's networks.

Figure (33) shows how I launched an attack using an IP to gain entry to the Defender router. It passed stage two security and at the same time I allowed the traffic to pass through the zone by removing the policy rule, to let the traffic reach the DNS server to demonstrate how all sensors worked and started receiving data, as proof that the traffic reached our DNS server. See figure (34). Figure (35) shows the traffic that is out from the ISP towards the Defender.

Then I returned the policy, to show how the zone and policy prevented the traffic - see figure (36). As you can see, the DNS server sensor does not receive any data, which is proof that the policy is working to stop the attack.

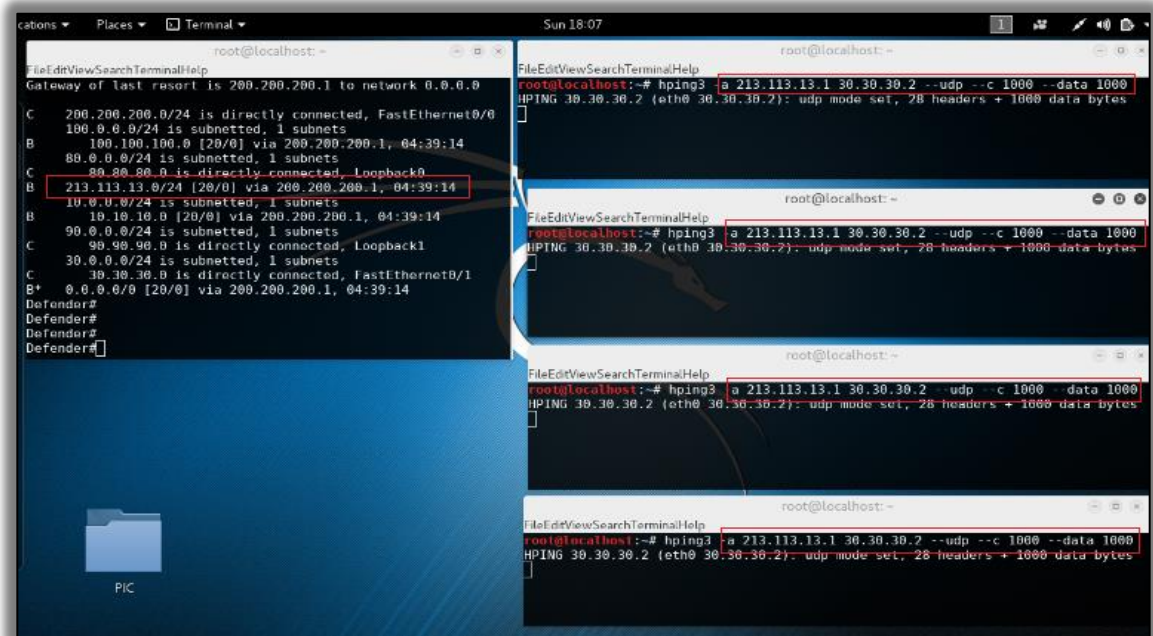


Figure (33) Sending data from an IP that has entry in the router table

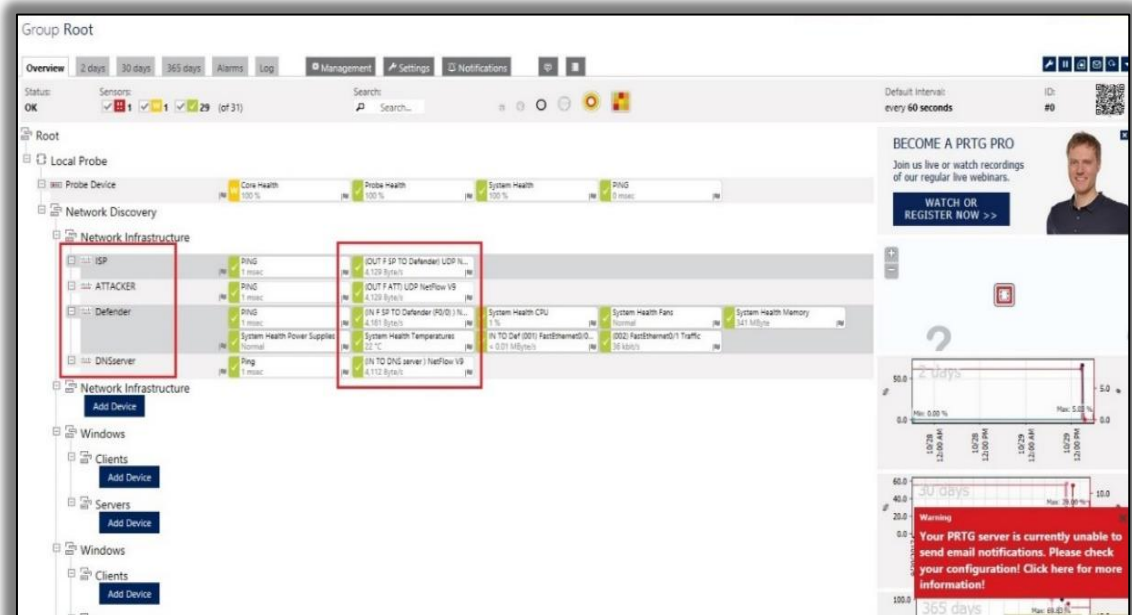


Figure (34) All sensors receiving data

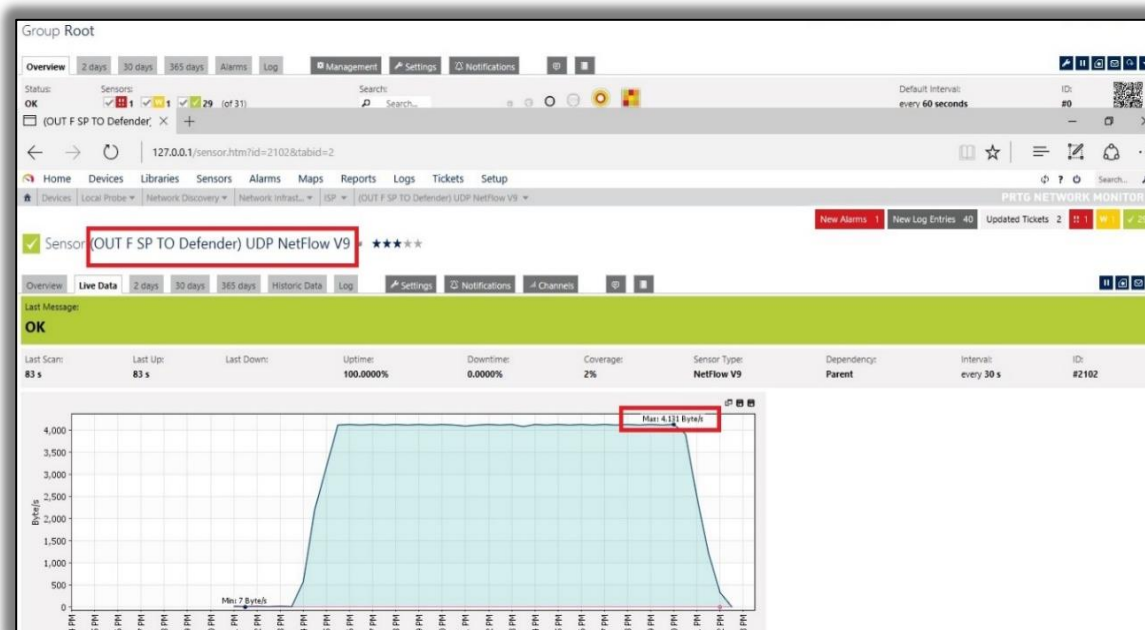


Figure (35) Traffic passing from ISP router to Defender router

As shown in these pictures, the traffic passes to the server normally, but if I change the policy in the router to stop the traffic from passing to the server to check the policy's reaction, the router stops the traffic from passing to the DMZ by dropping it, as we see in the figures below. The last sensor, which is the one responsible for reading the data on the DNS server is turned off because no data will have passed to the server.

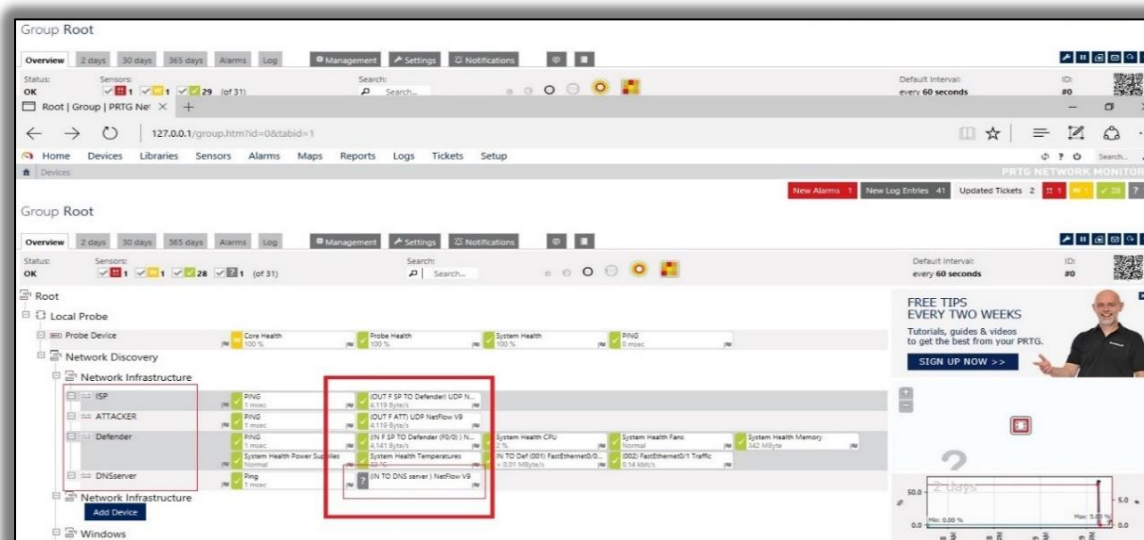


Figure (36) All sensors receiving data except DNS server

From the figures above, I am able to verify all three stages. While this is demonstrated stage by stage to illustrate how the stages work, in reality they would work together as an integrated solution.

5.5 Launch Brute Force attack

At this point, after the three earlier stages had been verified, I launched a DDoS attack with a large number of packets to check the router performance and how much the router could handle (but also considering sensor readings to avoid damaging the router) and took note of other affected resources, such as:

1. RAM Usage
2. CPU Usage
3. Temperature
4. Power supply

Those resources give us an idea about how much load and pressure the adopted solution, represented by the three-stage mitigation technique, put on the router. In this test, I assumed that the attack was launched from the hacked computer (which means hijacked packets) so it will pass stage one security and come directly to the edge router (defender) where stage two and three work as an integrated solution. I recorded all possible readings to illustrate how the router status is when the router is under the attack. I assumed two scenarios as follows:

- A.** Scenario one: in this scenario the attacker sends a large number of packets – 3000 packets per second. This is a large number and unlikely to happen in a real attack because sending this number would be easily recognised by any security appliance, so this is just for the purposes of checking the parameters that I mentioned above.

Figure (37) shows the option that I used with Hping3 to launch the attack:

- -i u10000 will send 10 packets/second
- -i u1000 will send 100 Packets/Second
- -i u100 will send more than around 1000, as shown in the figure (37) below

```
FileEditViewSearchTerminalHelp
root@localhost:~#
root@localhost:~#
root@localhost:~# hping3 -help
usage: hping3 host [options]
  -h --help          show this help
  -v --version       show version
  -c --count         packet count
  -i --interval      wait (uX for X microseconds, for example -i u1000)
  --fast            alias for -i u10000 (10 packets for second)
  --faster          alias for -i u1000 (100 packets for second)
  --flood           send packets as fast as possible. Don't show replies.
  -n --numeric      numeric output
  -q --quiet        quiet
  -I --interface    interface name (otherwise default routing interface)
  -V --verbose      verbose mode
  -D --debug        debugging info
  -z --bind         bind ctrl+z to ttl (default to dst port)
  -Z --unbind       unbind ctrl+z
  --beep           beep for every matching packet received
Mode
  default mode     TCP
  -0 --rawip       RAW IP mode
  -1 --icmp        ICMP mode
  -2 --udp         UDP mode
  -8 --scan        SCAN mode.
                   Example: hping --scan 1-30,70-90 -S www.target.host
  -9 --listen      listen mode
IP
  -a --spoo        spoof source address
  --rand-dest      random destination address mode. see the man.
```

Figure (37) Options are used in this attack

I sent the command as follows:

```
Hping3 -a 120.120.120.1 --udp -i u100 -data 600 30.30.30.2
```

As shown in figure (38) below, I used three sessions to send the UDP packets. With each command or session, I sent around 1000 packets per second.

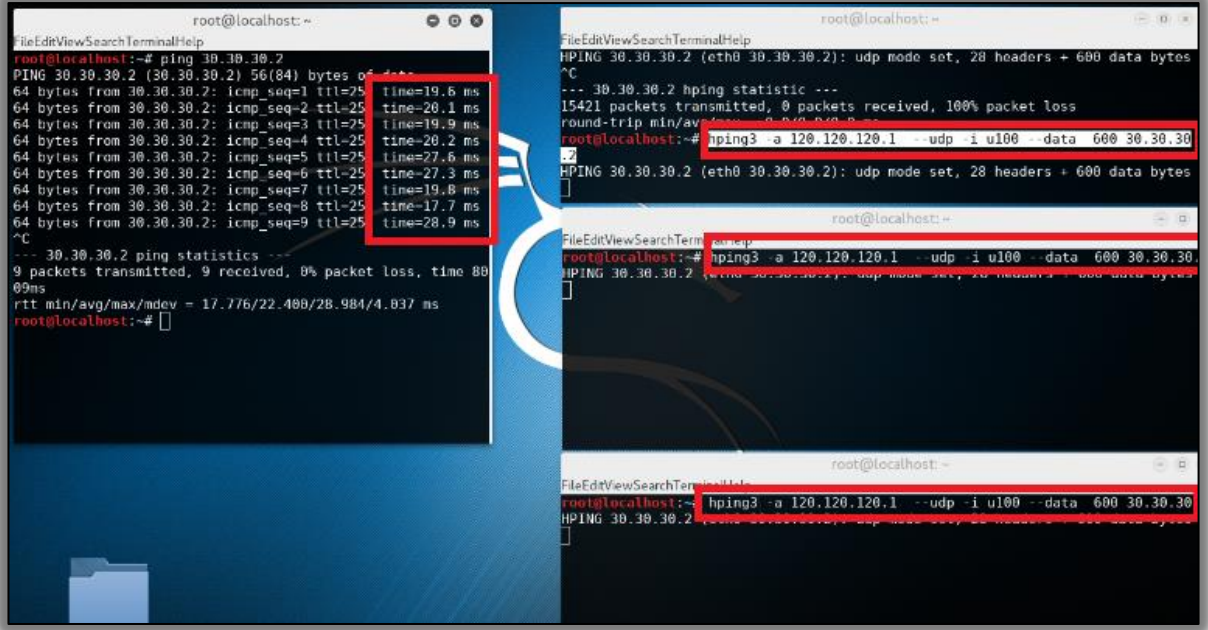


Figure (38) Sent 3000 packets per second and how ping reply time reacted

By looking at the PRTG sensors in figure (39), we notice the traffic does not pass to the DNS server. By following figures (40), (41) and (42), the PRTG software shows us the temperature, memory and power supply respectively.

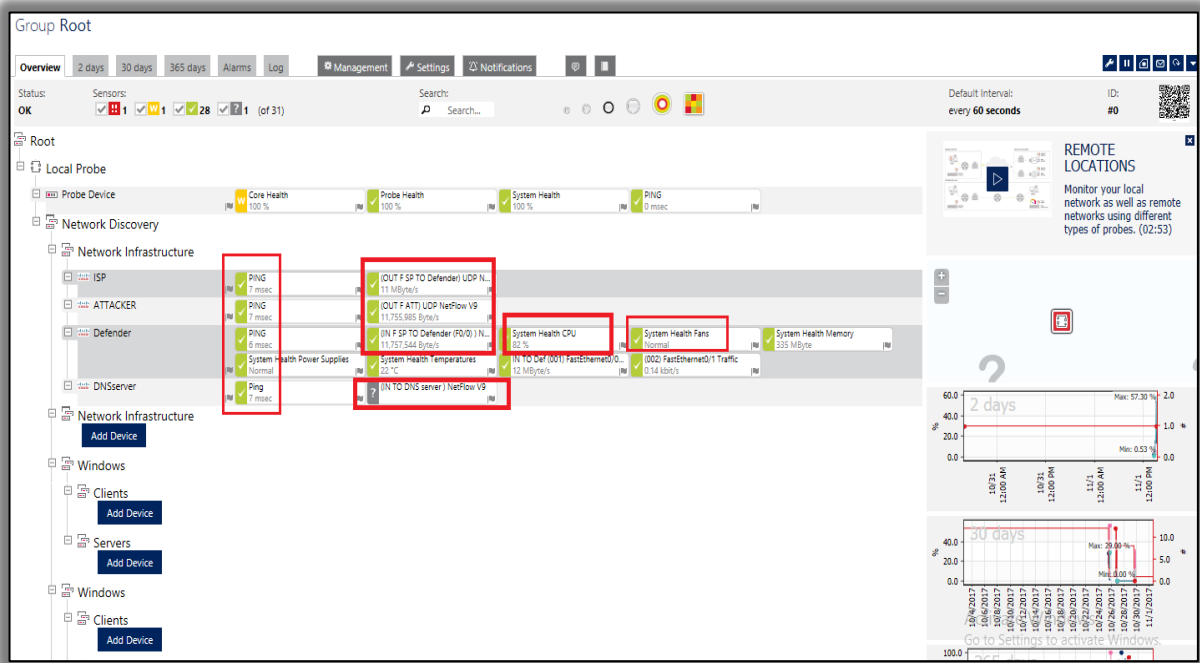


Figure (39) PRTG reading for whole installed sensors

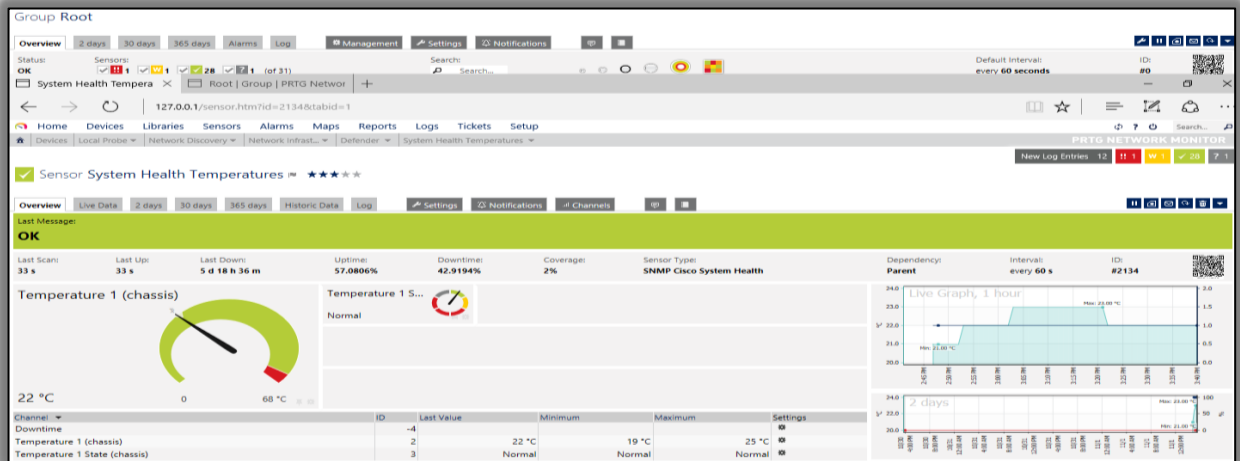


Figure (40) System temperature sensor

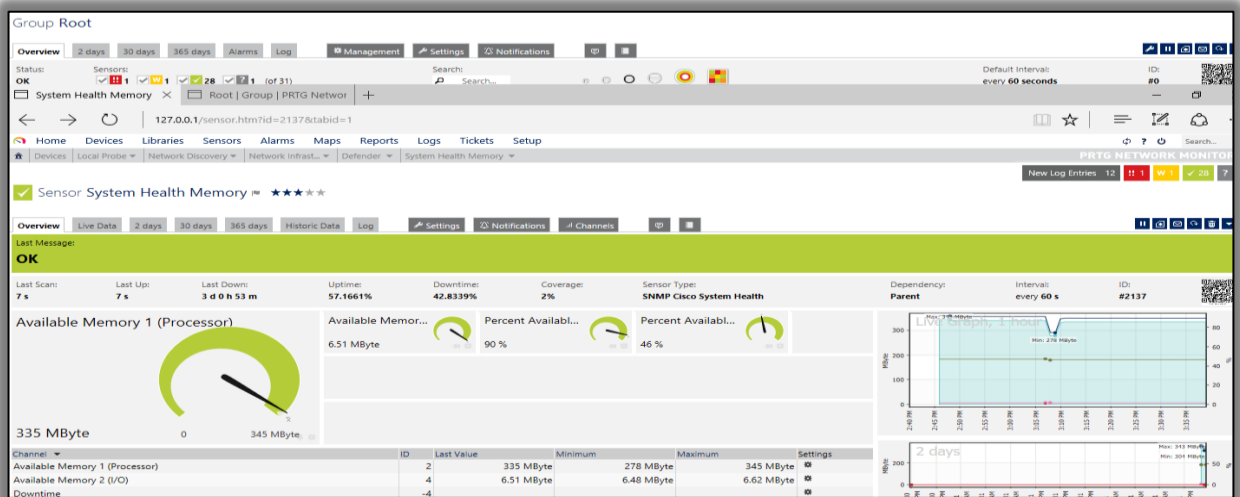


Figure (41) Memory usage sensor

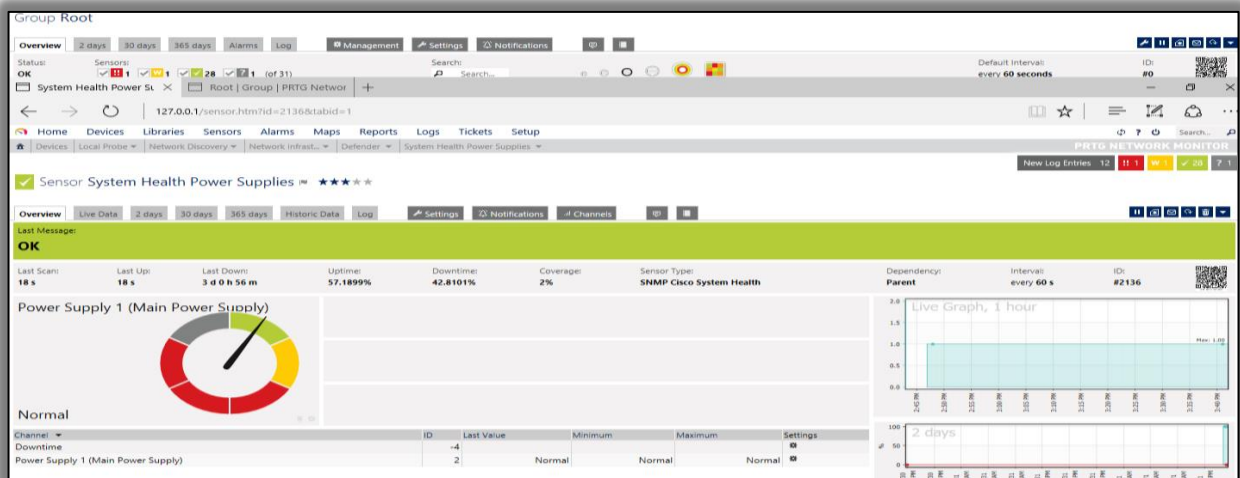


Figure (42) Power supply health sensor

From the figures above, I concluded that only the CPU usage was affected and increased due to the high packet drop, but the other router resources were not affected much. This is a good indication, and even the increase in CPU usage was expected because of the high number of packets dropped.

- B. Scenario Two: I tried sending the same amount of traffic by reducing the number of packets and increasing the size of the packets themselves. The differences can be seen from the command noted below. I tried sending traffic size around 11Mbps to 12Mbps, which is the same size of UDP traffic I used in the first scenario that caused the increase in CPU usage, so we could observe the difference.

This is the command I used:

```
Hping3 -a 120.120.120.1 --udp -i u1000 --data 5000 30.30.30.2
```

- -i u1000 means sending 100 packets per second
- --data 5000 means sending 5000 bytes for each packet

I sent the same size of traffic but with less packets to compare the CPU usage in both cases. See figure (43). As we can see in figure (44), the whole sensors working and the CPU usage went down to half or less than half with the same size of traffic. This was also expected due to the smaller number of packets dropped. It is interesting to note that memory usage was not affected in all cases, as shown in all figures below.

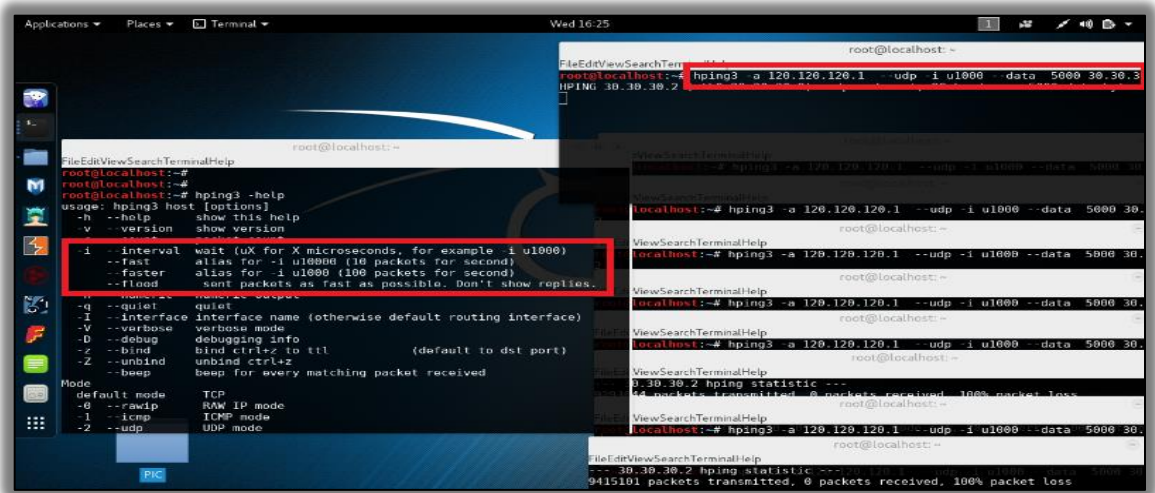


Figure (43) Sending the same size of packet with less number of packets

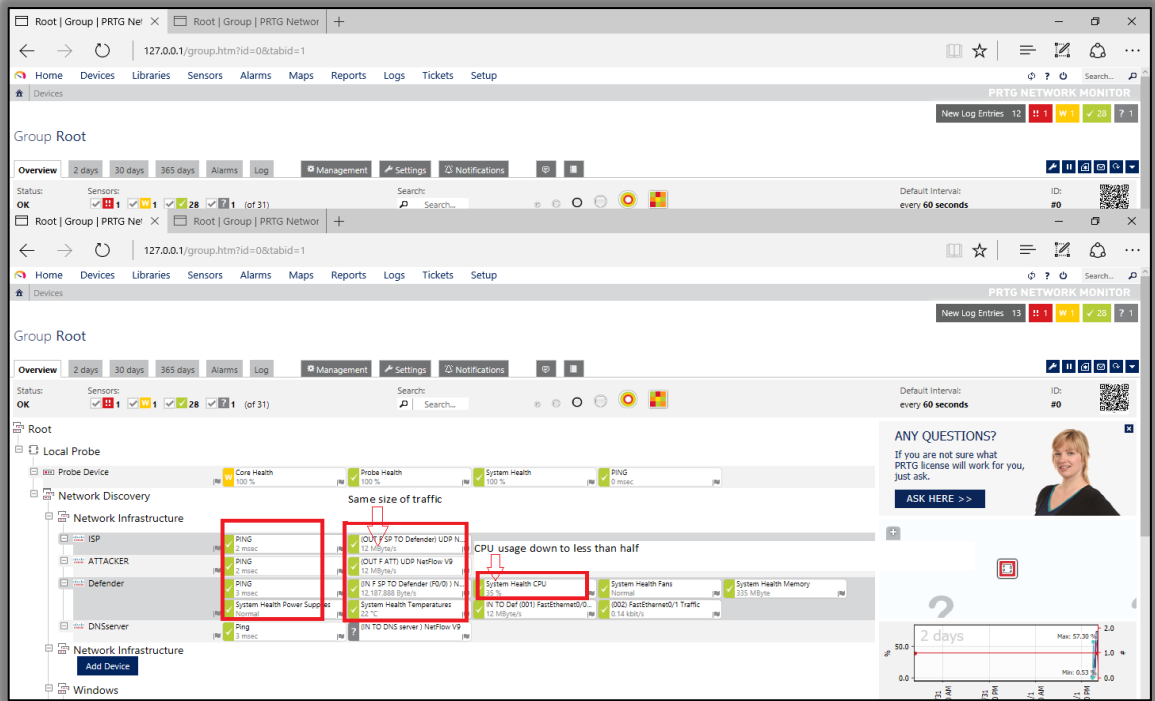


Figure (44) Whole sensors

To check the time response that is shown in the PRTG sensor page, I sent ICMP packets by using ping command to check the response time and it was exactly the same and compatible with PRTG, 2ms-3ms delay time, as we can see in figure (45), which gives us an indication that the result is really accurate.

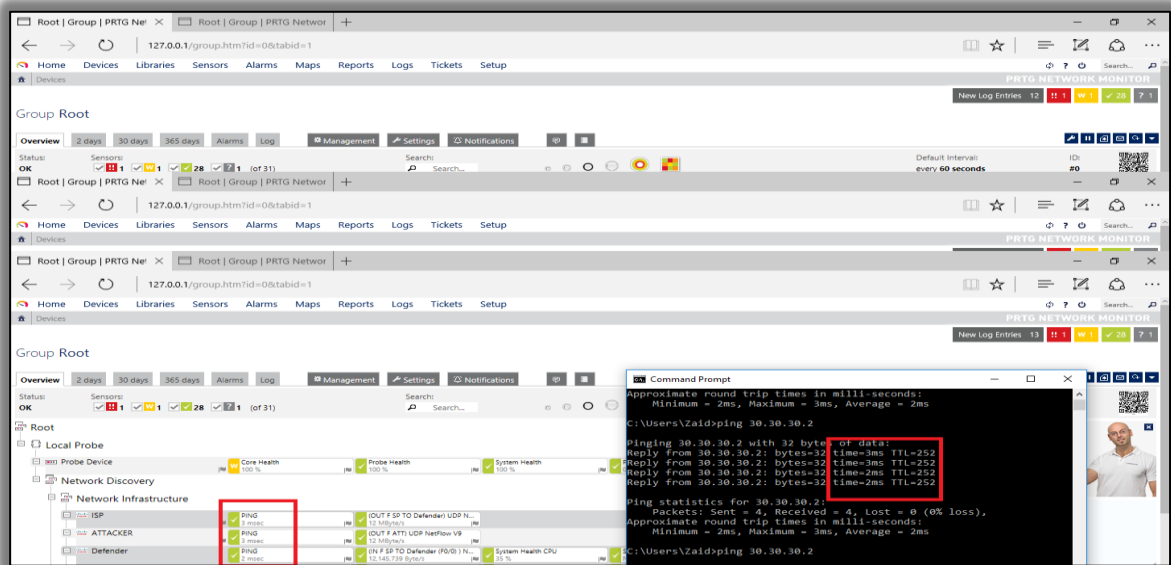


Figure (45) Matching the reply time

5.6 Verification process conclusion

The verification process shows that the mitigation technique worked perfectly. The suggested technique distributes the attack load in stages which lead to an increase in mitigation performance. Instead of doing this all on one router, I separated it into two routers. Also, in the edge router, I split the load again at stage two and three, instead of considering all of the load at the outside interface, which is the interface that faces the internet. By focussing on the readings collected, we can see that the CPU usage was affected and raised a lot when I increased the number of attack packets. While when I reduced the number, and increased the size of the packets, the CPU usage went down. From this we can conclude that the number of packets has more effect on the mitigation technique performance and router performance as a result. This indication helps us to understand router behaviour when it is under attack, and why attackers choose to send a high number of packets of normal or non-noticeable size.

5.7 Summary

This chapter includes all of the practical experiments that were conducted to verify the suggested solution. The suggested solution includes three stages and each stage was responsible for a specific role to mitigate attack traffic. Stage one was responsible for stopping spoofed packets; stage two was responsible for stopping unauthorised packets and foreign hijacked packets; and stage three was responsible for controlling the movements of permitted packets inside the organisation's edge router, by apply policies and rules that are embedded inside the zone policy to allow internal traffic to go forward to the internet and return normally, whilst restricting other traffic and defining which protocol can access which zone or service. All results are verification that each stage achieves the role that it should through the mitigation technique as an integrated technique.

Chapter 6

Evaluation and Data Analysis

6.1 Mitigation technique mechanism and performance

The results that were collected in the previous chapter show the mitigation technique success in mitigating this type of attack, as well as all other attacks that depend on sending spoofed IP addresses and those that send hijacked packets.

At this stage, I would like to illustrate how the mitigation technique works as an integrated solution in term of evaluation stage. Stage one and two worked in very high performance with less load on router resources as we mentioned in chapter five. Stage three had different performance because it depends on more than one option, such as, if dropping process drop against IP, Protocol or Service, what is really different is, the inspection methodology and time response when the inspection process is carried against IP, Protocol or service. In general stage three allows internal traffic to travel through the router by saving a copy of out traffic packets, then when the packet returns it is checked to see if it matches the saved copy. If it matches, it will pass, otherwise it will be dropped. This is how stage three deals with internal traffic. When internal traffic travels across the router towards the internet then returns, the first checking point that all packets should pass is the uRPF in loose mode. Here we can see there are some concept similarities between uRPF loose mode (stage two) and the zone firewall deployment (stage three).

My idea here focusses on splitting a load of incoming packets into two stages to produce efficient deployment to get better performance and a safer reliable solution. Stage two is responsible for checking all incoming packets and dropping all foreign, legitimate hijacked packets, unauthorised packets, and all type of packets that do not belong to my network, and also checking and passing my internal traffic, authorised customers, authorised DNS servers and all packets belonging to my router in any way. This represents around 50% of the traffic load.

Stage three, as shown in the results, is responsible for controlling which authorised packets are allowed to access which zone, and which packets are not allowed to access specific zones but can still access others. My internal packet can return to inside the network but nothing else. Authorised packets can access the DMZ but cannot pass to the inside zone. This can be deployed according to protocols, and also can be done by adding an access list for matching criteria in the IP level at the network level. In my case, I focussed on the DNS service, so I was focusing on working at the protocol level.

The results that I achieved were successful according to each stage role, and the deployment approach achieved the target (or the aim of the design). No one has carried out this integrated technique as I

have. At this point I would like to address the question about what if one of my customers is hacked and the attacker in this case can access my DMZ zone. Although it will access my DMZ, it will get no further. The best way to solve this issue would be to install IDS software to monitor the traffic coming in and out of the DMZ zone. By monitoring traffic anomaly behaviour, I can detect any malicious behaviour, and this constitutes my future work to improve mitigation performance.

The solution works perfectly to protect the inside network and assets and also prevents the attacker from using my servers to reflect and attack other networks or servers.

6.2 Evaluate mitigation technique performance

The evaluation of the mitigation technique was conducted by applying different levels of attack size – i.e. different numbers of packets per second, with different sized packets, at each level of the evaluation process. This is explained in more detail in the following bullet points.

Let us consider the brute force attack that is shown in figure (53):

- This mitigation technique split attack factors into three stages and reduced attack factors sequentially according to mitigation stages. Splitting the load raised the technique performance.
- Evaluation processes should include a range of values to establish the difference in performance. In order to conduct an accurate evaluation process, packet numbers sent out ranged from 300 p/s to 3000 p/s, and each packet was 500 bytes. By doing this it was possible to record how much the mitigation stage can mitigate and also monitor and record router performance. Under these conditions the router stopped the attack 100%, and the CPU usage raised at a reasonable percentage at each stage according to the verification process. This is still very good performance because it can stop and drop till 3000 p/s which is a large number of packets.
- Monitored router's RAM and consider the effects of the number of packets at each stage.
- Monitored router's temperature and the effect of the number of packets at each stage.
- Monitored power supply and the effect of the number of packets at each stage.
- Monitored stage three with CISCO devices and how stage three controls all incoming traffic according to protocol, IP and service port, so that different types of routing policies could be created to support the mitigation technique.

All of the points above were dependent on verification steps which were conducted at each stage in the previous chapter. At this point, a new evaluation process was launched for each stage alone to get as much accurate evaluation information as possible.

6.3 Illustrate evaluation mechanism

The evaluation process was conducted by sending a large number of attack packets to create a high load on the devices, and at the mitigation stage as well, with the standard size of the UDP DNS packets. Figure (46) represents the concepts of the evaluation process.

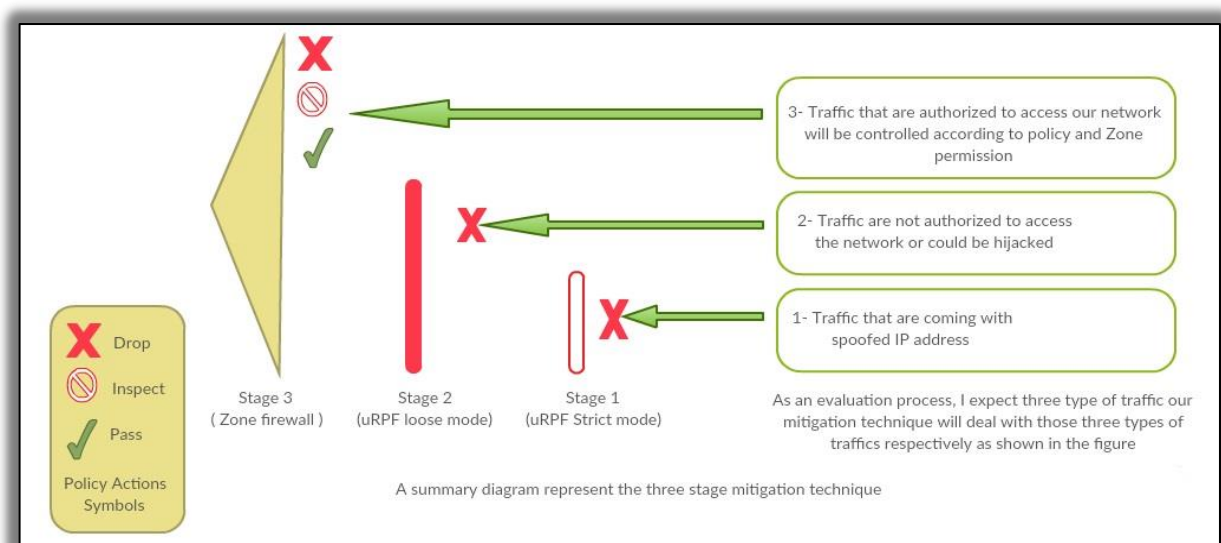


Figure (46) Diagram representing mitigation actions

Wireshark was used to sniff UDP packets and Cisco Show Commands to check drops, temperature, and CPU usage from inside the router, which gives an accurate reading. Also using Wireshark and the Show Commands gives us an opportunity to compare the results with the results that were collected in the previous chapter. This helped us to check how performances changed at each mitigation stage according to the change in the number of packets.

At each evaluation stage a diagram is included to demonstrate the practical part of the evaluation process. All evaluation reading steps and its results are recorded and listed in the evaluation stage table. A diagram of the last step or attack which is represented by Attack 5 for each stage has also been included because it represents the highest effects. The evaluation diagrams show how the evaluation process was conducted and where the evaluation results came from, to use for comparing studies. Also

shown is the Wireshark sniffing software filter that sniffed the packets sent through each step of the evaluation process, plus command results from inside the routers, which also matched with the PRTG reading.

6.3.1 Stage 1 Evaluation: Represented by uRPF in strict mode.

At this stage, packets were sent with a spoofed source IP address, as shown in the Evaluation Table – see table (3).

Attacks	NOP	SOP	SOT	NOD	CPU Usage	System Temp.	Memory
Attack1	300	500	1.5Mb	100%	10%	21c	91% Available
Attack2	500	500	2.4Mb	100%	16%	22c	Not effected
Attack3	1000	500	4.7Mb	100%	26%	22c	Not effected
Attack4	2000	500	9.4Mb	100%	48%	22c	Not effected
Attack5	3000	500	14.3Mb	100%	57%	22c	Not effected

Table (3) Stage 1 Evaluation table

NOP: Number of packets sent

SOP: Size of packets sent

SOT: Size of traffic sent

NOD: Number of dropped packets, i.e. mitigation success

CPU Usage: CPU usage due to dropping large number of packets

System temp.: System temperature and fans

Memory: Memory usage

All of the results above are actual readings, and as can be seen, the most effective factors are NOP, SOT and CPU, as shown in figure (47) and in the evaluation chart in figure (48).

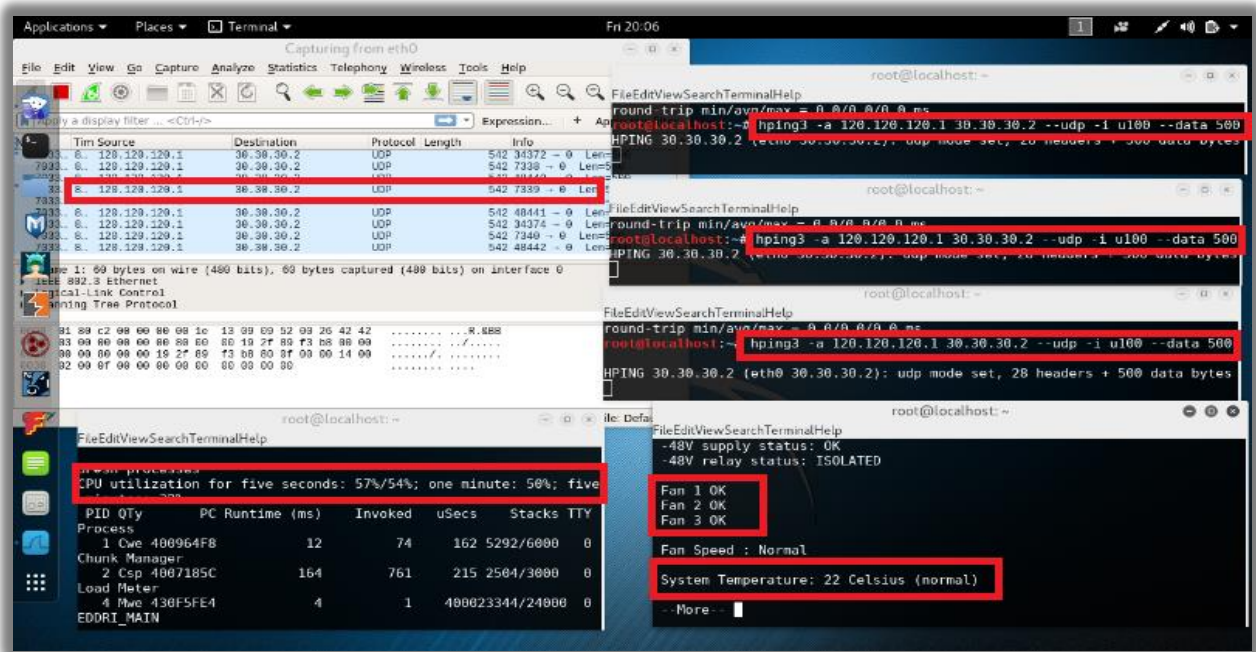


Figure (47) Practical part of reading collection for Attack5

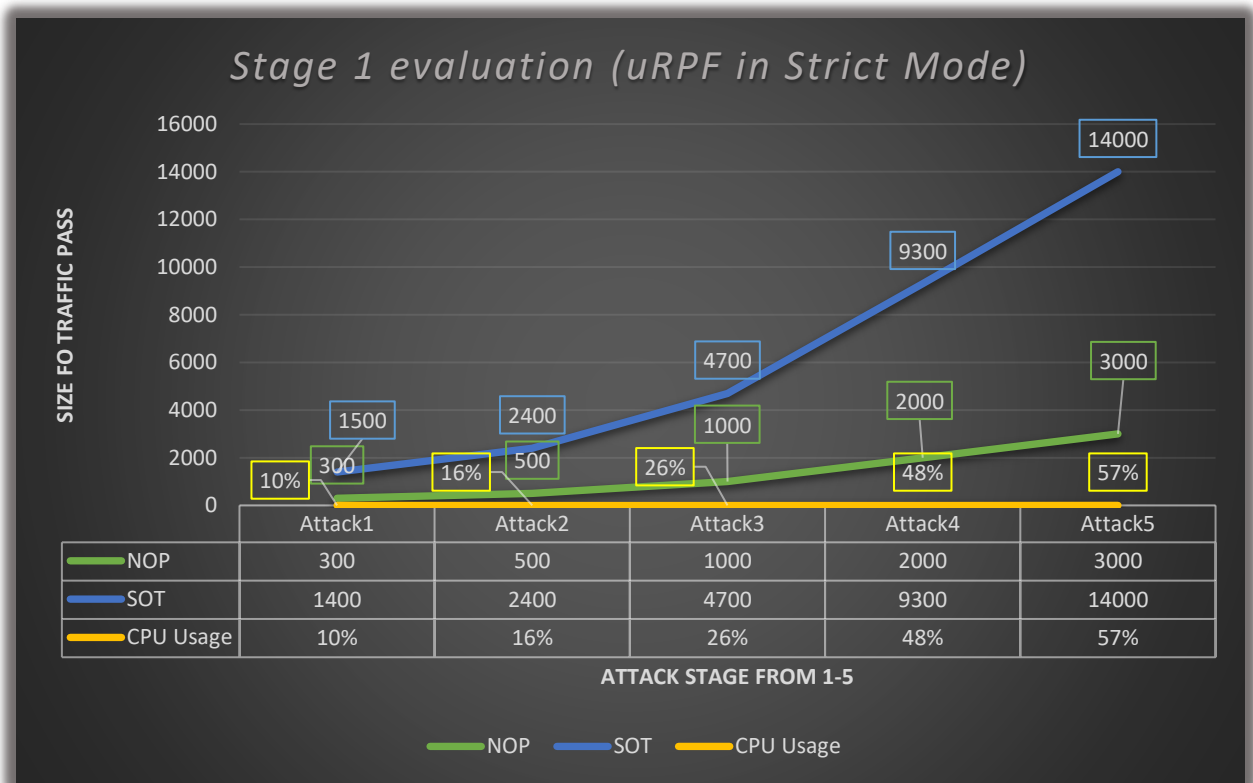


Figure (48) Stage 1 Evaluation chart

As shown in the figures above, all of the readings and results were collected by Wireshark and Cisco devices through a real attack, and that mitigation success was 100% at different load levels (which is represented by different size and number of sent packets). The chart above shows the differences and the relationships through the evaluation steps represented by the attacks conducted from Attack1 until Attack5.

6.3.2 Stage 2 Evaluation: Represented by uRPF in loose mode

At this stage, packets were sent with a source IP that did not exist in the router, which meant it could be normal unauthorised packet, or hijacked packets, or even an open DNS resolver. The evaluation process was conducted according to table (4) below, with figure (49) showing the practical part, and figure (50) showing the Stage 2 Evaluation Chart.

Attacks	NOP	SOP	SOT	NOD	CPU	System Temperature	Memory
Attack1	300	500	1.5Mb	100%	19%	21 C	91% Available
Attack2	500	500	2.4Mb	100%	26%	22 C	Not effected
Attack3	1000	500	4.7Mb	100%	42%	22 C	Not effected
Attack4	2000	500	9.4Mb	100%	64%	22 C	Not effected
Attack5	3000	500	14.3Mb	100%	91%	22 C	Not effected

Table (4) Stage 2 Evaluation table

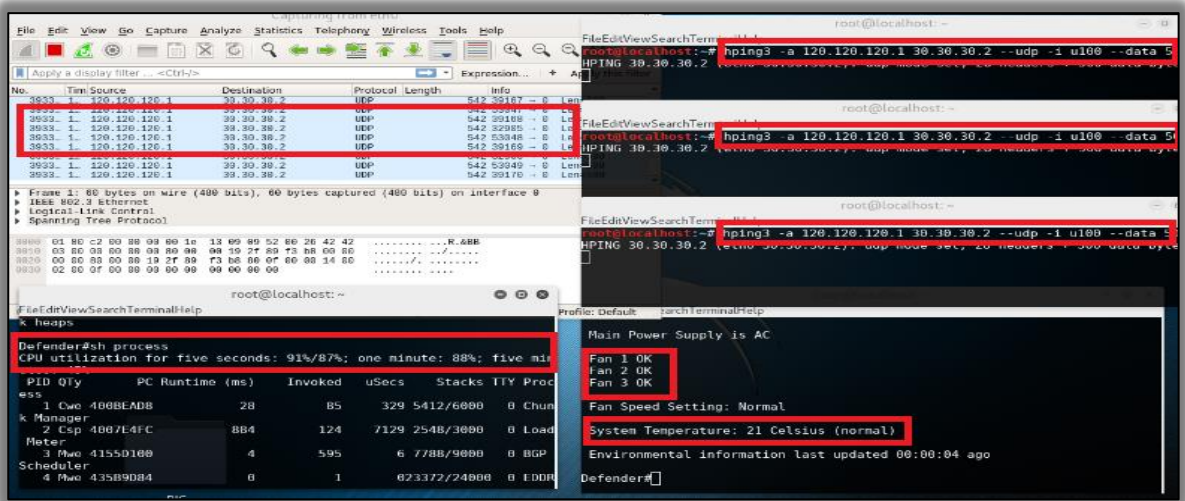


Figure (49) Practical part of reading collection for Attack5

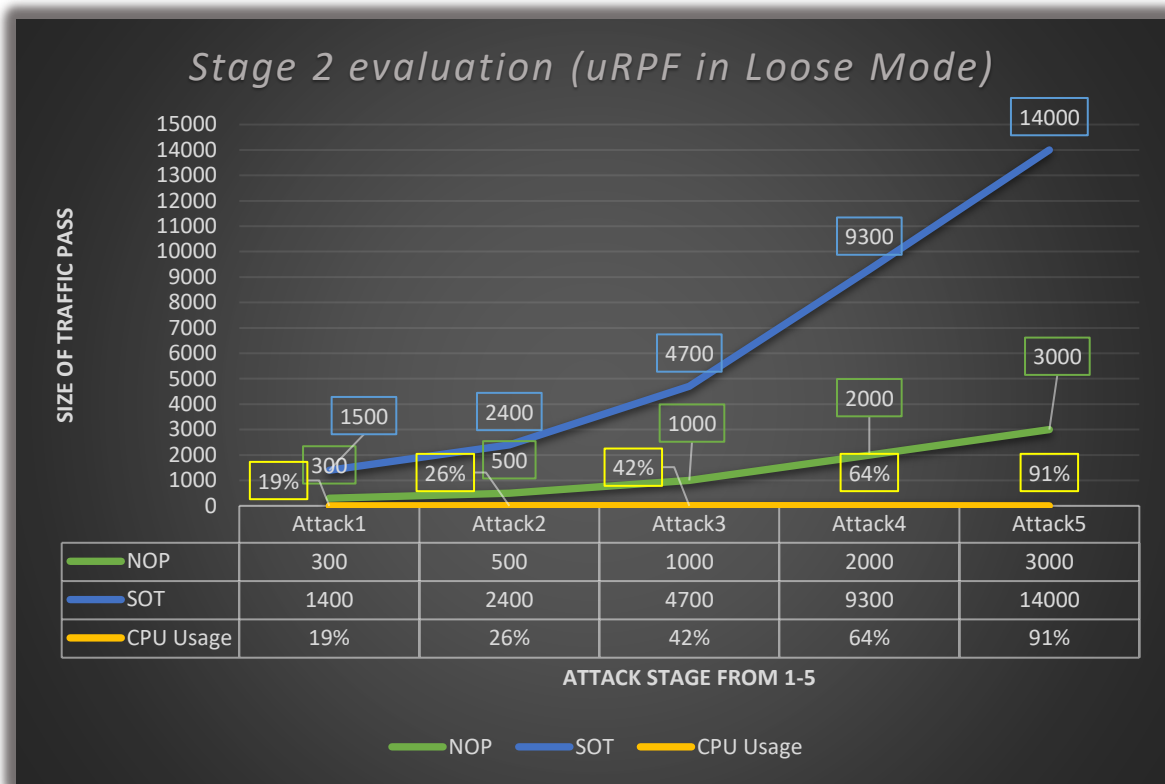


Figure (50) Stage 2 Evaluation chart

The reading above shows that CPU usage increased more in uRPF loose mode than compared with uRPF strict mode under the same conditions, which means with the same number of packets sent, uRPF has different performance, even when using the same size of packets with the same procedures and the same conditions.

6.3.3 Stage 3 Evaluation: Represented by Zone firewall

At this stage, authorised packets were sent, which means packets that had an entry in the router so they could pass stage two without being dropped, but I tried to stop the packets at the OUTSIDE zone for the aim of conducting the evaluation process at stage three to check how much the zone firewall could afford. The policy had to change to let the router drop the packets and check CPU usage, memory, and temperature.

The stage three evaluation process, shown in table (5) below, shows the details about the packets sent, with the change in the number and size of the packets to highlight the differences. See figure (51) and the evaluation chart in figure (52).

Attacks	NOP	SOP	SOT	NOD	CPU	System Temp.	Memory
Attack1	100	1000	~1000	100%	27%	21c	91% available
Attack2	200	1000	~1700	100%	43%	22c	Not affected
Attack3	300	500	~1500	100%	57%	22c	Not affected
Attack4	500	500	~2500	100%	89%	22c	Not affected
Attack5	700	500	~3500	100%	99%	22c	Not affected

Table (5) Stage 3 Evaluation table

As can be seen from readings above, the CPU usage is raised too much and is near to being halted, but it is interesting to note that it can still drop or stop 100% of DNS packets efficiently. Also, the number of packets that were sent is much less than at stages one and two, which means that stage three cannot manage the same load, but works efficiently to manage normal authorised traffic with permitted packets. Also, because it gets good support at stage two which stops a lot of unwanted traffic and at the same time gives an opportunity to stage three to manage authorised traffic with a good level of efficiency.

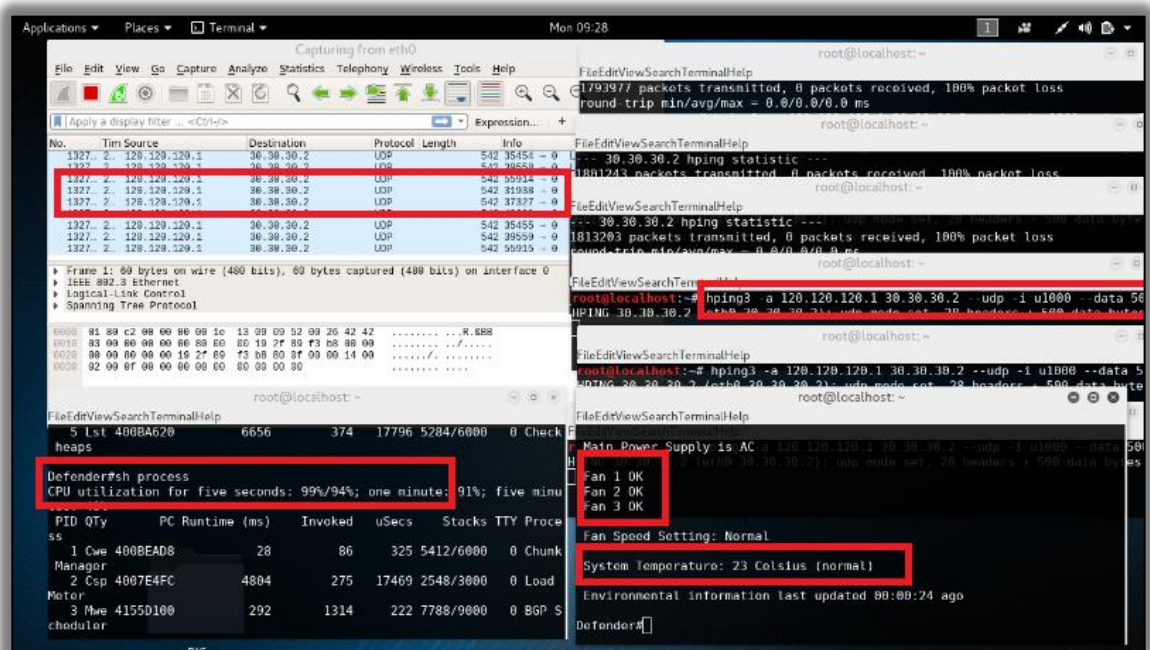


Figure (51) Practical part of reading collection for attacks

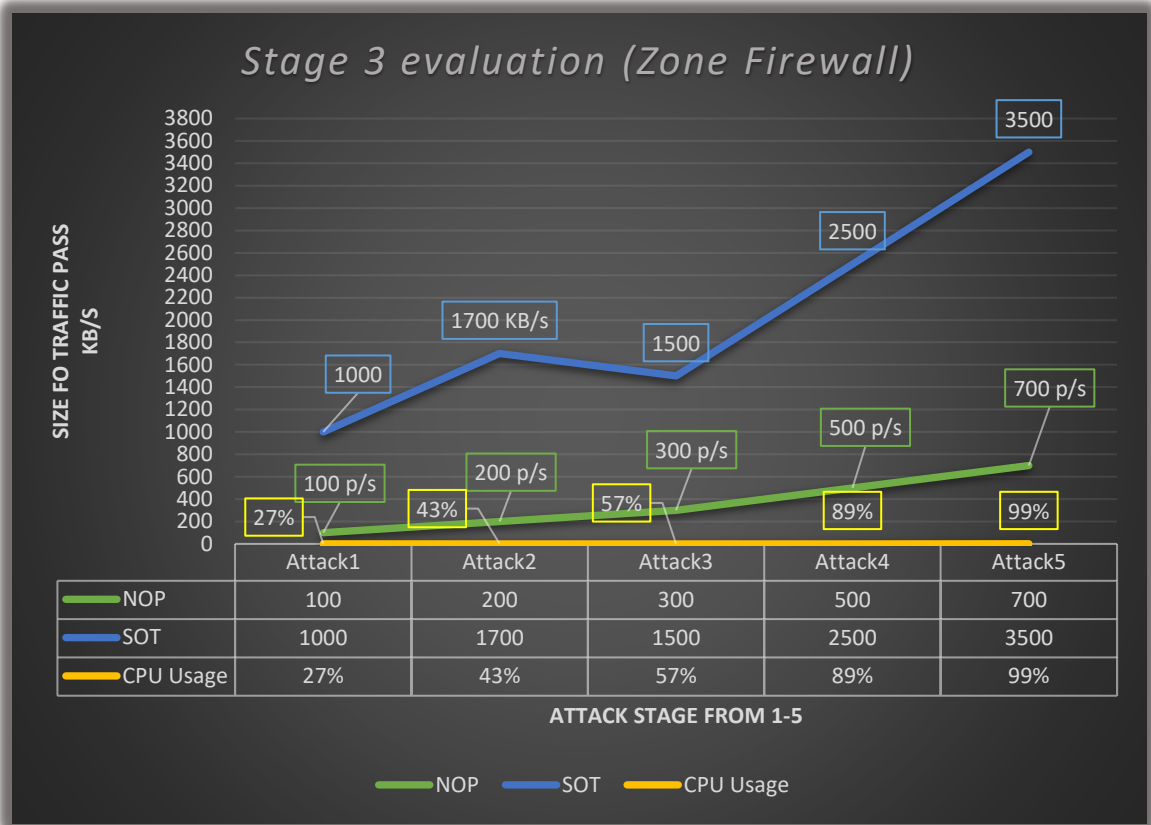


Figure (52) Stage 3 Evaluation Chart

6.4 Evaluation process conclusion

The evaluation of any technique should be conducted under a different load level with the same conditions. Sending 300 packets per second, and increasing the number to 3000 packets per second, tested the mitigation technique in order to know how much it could cope with and also record the number of dropped packets.

By using an old Router Model 2811, and also using Hping3 as attack tools, stages one and two coped successfully, and dropped until 3000 packets per second, with a reasonable percentage of CPU usage and less affected temperature and RAM, as demonstrated in the results above. Stage three, managed a lesser number of packets and the CPU raised higher than at stage two and three.

According to the solution design, stage three would be responsible for dealing with trusted packets and may not face attack packets directly because it is protected and supported by stage one and two, so even if performance at stage three is less than at stages one and two, it can still efficiently control trusted packets, and drop any packet trying to pass any zone it should not. It is worth noting that this

test was run on an old router model and if the latest router models had been used at stage three, performance would have been much better. Most organisations use more modern router models than was used in this study.

6.5 Summary

The evaluation chapter includes exposing all mitigation technique stages to different load levels. These load levels are represented in the table, including attack specifications that were conducted at each stage and recording responses at each stage. The evaluation process produced robust results that give the reader a complete picture of how the suggested solution reacts under different levels of attack. At this stage of the study, it became very obvious that using a practical testbed environment is very important and how much more effective it is than using simulation or theory methodology in these types of academic studies.

Chapter 7

Discussion and Future Work

7.1 Evaluation results

According to the results collected at each stage respectively, stage one efficiently stopped all of the spoofed packets without missing even one packet. Stage two did the same, but as we can see from the results, there was a little difference in performance.

From the theory design, uRPF in strict mode should take higher CPU usage than loose mode, because the concept of strict mode has met two criteria before it lets the packet pass. The first one is the source IP address of incoming packets has an entry in the routing table, and the next one is whether the IP address comes from this interface specifically or if it is related to another interface. If it comes specifically from this interface, it will pass, but if it has an entry but is related to another interface, the second criteria has not been met so it will drop.

uRPF in loose mode just looks for the first condition, i.e. whether it has an entry in the routing table or not. If it has, it will pass, and if it does not, it will drop. This mode was found to solve the problem for organisations that have more than one link to the ISP, and the traffic could go out from an interface and return from another interface according to routing protocol metrics and measurements.

It was interesting to note the different performance between stages one and two, even by sending the same number of packets of the same size. Stage one, which works in strict mode, uses less CPU resources than stage two, which uses loose mode. This was very obvious in all five steps of the evaluation process which started with attack one with 300 p/s, to attack five with 3000 p/s, and this was one of the benefits of the evaluation process.

Stage three, represented by zone firewall, returned a different result, with less capability compared with stages one and two. However, by looking at the results closely, it is clear that the CPU router is exhausted with much fewer attack packets compared with stages one and two, and it is very important to know and understand feature capability. From the results above, we can see that the zone firewall cannot face the attack alone and could easily fall down. However, bearing in mind that these experiments were conducted on an old Cisco 2811 router, much better performance could be expected if a modern model were to be used. Despite that, it is clear from the results that stages one and two produce a solid result in comparison to stage three.

An interesting question could be raised around whether stage two could stop an attack covered by a tunnel. The answer is yes, because an organisation's router acts as a stub or peripheral router, so

applying that at this stage would work perfectly because all the tunnel covers should be taken off at this point. Simply put, the location or the place where the solution is applied is very important and could play a vital role in the success or otherwise of the suggested solution.

On the other hand, if the attacker uses a tunnel such as L2TP, PPTP or GRE to cover their packets, there is a possibility it would pass stage 1 without detection, and this is because an ISP router is not a peripheral router and this router is not the tunnel destination so there are still more encapsulations. At this stage, not all tunnel covers are stripped off and for that reason it could pass. Attention needs to be paid to this issue and there are a lot of possibilities, such as applying an access list ingress filter on the physical interface itself or on the tunnel interface to mitigate this tunnel. Also, if the tunnel run supported by keep alive feature which is one of tunnel features, as this could affect the results too, so you should be aware of this matter too.

Any solution has a positive side and negative side and the negative one needs to be focussed on in order to refine or retune the solution over time, so it can face new challenges. Also, the mitigation technique could be improved by adding more policy to stage three and adding access list ingress and egress filters to support it, too.

7.2 Comparison with other solutions

Looking at other suggested solutions, enables us to evaluate our work in better way. In (Fu, Papatriantafilou, & Tsigas, 2011) the research team suggested creating a cluster design and assigning a router for each checking task by depending on a specific algorithm. They used a router called an Egress Checking Router (ECR) and an Ingress Checking Router (ICR) to check and control the outgoing and incoming traffic. In addition to that, the key role of this solution was achieved by another router called a Coordinator, at the edge of each cluster. Coordinator router responsible for applying the security policy in deny, permit and all other policies, were applied to protect the cluster from outside threats. By applying the following equation, $\sum_{h=1}^6 P^{h+1} Ph$, for the evaluation process, and by considering related assumptions, the research team found that the probability of malicious traffic reaching the target victim was smaller than 10^{-18} . By comparing the solution with solution in this thesis, we find it depends on the same concepts but uses a different approach.

In (Tritilanunt, Sivakorn, Juengjincharoen, & Siripornpisan, 2010), the researchers suggested a solution to detect DoS/DDoS attacks. The solution depended on inspecting all packet headers that passed through the router, which could lead to detecting all DDoS attacks, even small ones. Some small packets

sneak quietly, but by inspecting packets one by one they can be detected. The detection mechanism depends on Shannon's function - $H(t) = -\sum_i \left(\frac{n_i}{S}\right) \log\left(\frac{n_i}{S}\right)$ and by considering all related assumptions. This solution has the ability to detect 99.48% of ICMP attacks, 99.40% of TCP SYN attacks, 99.52% of SMURF attacks, and can recognise 98.14% of legitimate traffic. The approach includes packet size and content, which would help to detect anomaly behaviour, while all other solutions that use the same technique were focussed on packet size only. The solution uses different methodology to that suggested in this thesis and the inspection approach used would create a high load which would affect overall performance, as the research team itself noted.

In (Thongkanchorn, Ngamsuriyaroj, & Visoottiviseth, 2013), the study research team installed and evaluated the performance of three software - IDS, Suricata, Snort, and Bro - against different types of DoS attacks, one of which was DNS DDoS attacks. The performance of the three IDS systems was measured by three criteria; the number of packets lost, the number of alerts and CPU usage metrics. The experiment results showed that the three IDS had a low rate for the three dependent metrics for TCP traffic, but when the traffic size increased, all the indication metrics increased significantly. It is notable that the Bro IDS results flattened CPU usage results when the traffic rate increased compared with the other two IDS systems which were much more highly affected, and this is crucial. According to the study, the performance of all the IDS's is different from one attack to another, but in general the IDS Bro gave better results. This study will be very helpful in my future work as I will analyse the study and choose one of the IDS's to include in my solution if one of my authorised customers gets hacked and the hacker exploits the customer's system to run the attack.

In (Malliga, Tamilarasi, & Janani, 2008), the research team focused on detecting spoofed DoS attack packets, even virtually unnoticeable ones, by relying on a statistics anomaly detection system, which also relies on Shannon's function, in their work. They modified the calculation equation to obtain more accurate and dependent results, using the following equation:

$$H(S) = - \sum S_i P_i(x_i) \log_2 (1/(P(x_i))) \dots\dots \text{Where } S_i P_i \text{ is source IP address.}$$

The team modified the function to get better results, such as creating multiple flow monitors to monitor how many connections were opened, with minimum and maximum limits by each client at the same time, and by monitoring these factors they calculated the anomaly factor.

The effectiveness of the suggested solution would depend on the measurement of the statistics gathered and deployed in the suggested solution, and is also different from victim to victim according

to the working nature of the organisation needing to be protected. The research team considered their solution superior to other solutions because it had the ability to recognise the packets that used an internal IP as the spoofed address, but my point of view is, this statistical solution works in mathematical reaction methodology rather than providing effective proactive solutions.

By comparing the suggested solutions with my suggested solution, I believe my solution could work as a proactive solution and stop a lot of attacks at stage one and two, and stage three would control and deal with the last part of allowed traffic, while the suggested solution would react to the number of packets that attackers send, so it is obvious each solution would depend on different methodology.

In (UzmaSattar, Naqash, Zafar, Razzaq, & Bin Ubaid, 2013), the study research team used two Bloom filters, one for outgoing traffic and one for incoming traffic, to detect the attack. The Bloom filter in this mechanism works well and could support my technique, too. A Bloom filter calculates the response time for the incoming packet, so when the client sends a request to the DNS server, a Bloom filter saves the request and waits for the response, which is expected to return in a specific period of time. If it does not return in the expected period, it will be blocked. They also added a threshold to reduce the likelihood of false positives.

In general, most of the solutions are not too dissimilar, but everyone takes a different approach and sometimes depend on different functions or different equations to achieve the same target. I cannot cover all studies, but have included a sample of those which tried to mitigate DNS attacks or those who used mitigation techniques that can help in mitigating DNS attacks.

7.3 Summary

This chapter discusses future work and some specific solutions that are similar to the solution suggested in this thesis, and highlights some aspects that could help the reader to understand the differences between the approaches that are discussed in those papers.

Appendices

Appendix A: Hardware Specifications

Routers

Router Cisco 2811	Specification
Interfaces	Fast Ethernet
Management Protocol	SNMP
Ports	2 L3 ports/9 L2 switch ports
RAM	DDR SDRAM 256MB/768MB (Maximal)
Flash Memory	64MB/256MB(Maximal)
Voltage Required	AC 120/230V
Features	Firewall protection, VPN support, hardware encryption, and Quality of Service (QoS)

Table 1 Router Specification

Server

DNS Server	Specification
CPU	AMD Ryzen 1700/1800
Mother board	ASUS PRIME B350M-A
RAM	32GB
HD	2TB
O/S	Win Server 2012
Voltage Required	AC 120/230V

Table 2 Server specification

References

- A. Herrero and E. Corchado. (2011). *A Comparative Performance Evaluation of DNS Tunneling Tools*. 2011 In Computational Intelligence in Security for Information Systems. Retrieved from <http://ai2-s2pdfs.s3.amazonaws.com/2c56/e5a5768f684f8378753ab0e426071ab893a5.pdf>
- Aborujilah, A., Ismail, M. N., & Musa, S. (2014). *Detecting TCP SYN Based Flooding Attacks by Analysing CPU and Network Resources Performance*. 2014 3rd International Conference on Advanced Computer Science Applications and Technologies. Retrieved from <http://ieeexplore.ieee.org/document/7076886/>
- Aiello, M., Mongelli, M., & Papaleo, G. (2013). *Basic classifiers for DNS tunneling detection*. 2013 IEEE Symposium on Computers and Communications (ISCC). Retrieved from <http://ieeexplore.ieee.org/document/6755060/>
- Alieyan, K., Kadhum, M. M., Anbar, M., Rehman, S. U., & Alajmi, N. K. (2016). *An overview of DDoS attacks based on DNS*. 2016 International Conference on Information and Communication Technology Convergence (ICTC). Retrieved from <http://ieeexplore.ieee.org/document/7763485/>
- Anstee, D., Bowen, P., Chui, C. & Sockrider, G. (2016). *Arbor Special Report: Worldwide Infrastructure Security Report*. Retrieved from https://www.arbornetworks.com/images/documents/WISR2016_EN_Web.pdf
- Aqil, A., Atya, A. O., Jaeger, T., Krishnamurthy, S. V., Levitt, K., McDaniel, P. D., ... Swami, A. (2015). *Detection of stealthy TCP-based DoS attacks*. MILCOM 2015 - 2015 IEEE Military Communications Conference. Retrieved from <http://ieeexplore.ieee.org/document/7357467/>
- Arunwan, M., Laong, T., & Atthayuwat, K. (2016). *Defensive performance comparison of firewall systems*. 2016 Management and Innovation Technology International Conference (MITicon). Retrieved from <http://ieeexplore.ieee.org/document/8025212/>
- Bassil, R., Hobeica, R., Itani, W., Ghali, C., Kayssi, A., & Chehab, A. (2012). *Security analysis and solution for thwarting cache poisoning attacks in the Domain Name System*. 2012 19th International Conference on Telecommunications (ICT). Retrieved from <http://ieeexplore.ieee.org/document/6221233/>
- Cambiaso, E., Aiello, M., Mongelli, M., & Papaleo, G. (2016). *Feature transformation and Mutual Information for DNS tunneling analysis*. 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN). Retrieved from <http://ieeexplore.ieee.org/document/7536939/>

- Cisco. (2013, November 1). Security Configuration Guide: Unicast Reverse Path Forwarding Cisco IOS XE Release 3S - Unicast Reverse Path Forwarding Loose Mode [Support]. Retrieved from https://www.Cisco.com/c/en/us/td/docs/ios-xml/ios/sec_data_urpf/configuration/xs-3s/sec-data-urpf-xe-3s-book/sec-unicast-rpf-loose-mode.html
- CNAME record. (2017, November 3). Retrieved from https://en.wikipedia.org/wiki/CNAME_record
- Compagno, A., Conti, M., Gasti, P., & Tsudik, G. (2013). *Poseidon: Mitigating interest flooding DDoS attacks in Named Data Networking*. 38th Annual IEEE Conference on Local Computer Networks. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6761300>
- Diovu, R. C., & Agee, J. T. (2017). *A cloud-based openflow firewall for mitigation against DDoS attacks in smart grid AMI networks*. 2017 IEEE PES PowerAfrica. Retrieved from <http://ieeexplore.ieee.org/document/7991195/>
- Dubey, R., & Gupta, H. (2016). *SQL filtering: An effective technique to prevent SQL injection attack*. 2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO). Retrieved from <http://ieeexplore.ieee.org/document/7784972/>
- Explorable. (2009, November 3). Quantitative and Qualitative Research - Objective or Subjective? Retrieved from <https://explorable.com/quantitative-and-qualitative-research>
- Fu, Z. (2011). *Mitigating Distributed Denial-of-Service Attacks: Application-Defense and Network-Defense Methods*. 2011 Seventh European Conference on Computer Network Defence. Retrieved from <http://ieeexplore.ieee.org/document/6377740/>
- Fu, Z., Papatriantafidou, M., & Tsigas, P. (2011). *CluB: A Cluster based framework for Mitigating Distributed denial Of Service attacks*. Retrieved from <http://www.syssec-project.eu/m/page-media/3/club-sac2011.pdf>
- Ghafir, I., & Prenosil, V. (2015). *DNS traffic analysis for malicious domains detection*. 2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN). Retrieved from <http://ieeexplore.ieee.org/document/7095337/>
- Gillman, D., Lin, Y., Maggs, B., & Sitaraman, R. K. (2015). *Protecting Websites from Attack with Secure Delivery Networks*. Retrieved from <http://ieeexplore.ieee.org/abstract/document/7085639/>
- Gunjan, V. K., Kumar, A., & Rao, A. A. (2014). *Present & Future Paradigms of Cyber Crime & Security Majors - Growth & Rising Trends*. 2014 4th International Conference on Artificial Intelligence with Applications in Engineering and Technology. Retrieved from <http://ieeexplore.ieee.org/document/7351818/>

- Hasanifard, M., & Ladani, B. T. (2014). *DoS and port scan attack detection in high speed networks*. 2014 11th International ISC Conference on Information Security and Cryptology. Retrieved from <http://ieeexplore.ieee.org/document/6994023/>
- Hawkins, B., & Demsky, B. (2017). *ZenIDS: Introspective Intrusion Detection for PHP Applications*. 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE). Retrieved from <http://ieeexplore.ieee.org/document/7985665/>
- hping3 | Penetration Testing Tools. (n.d.). Retrieved from <https://tools.kali.org/information-gathering/hping3>
- James, C., & Murthy, H. A. (2011). *Time series models and its relevance to modeling TCP SYN based DoS attacks*. 2011 7th EURO-NGI Conference on Next Generation Internet Networks. Retrieved from <http://ieeexplore.ieee.org/document/5985951/>
- Jamous, Z. E., Soltani, S., Sagduyu, Y., & Li, J. (2016). *RADAR: An automated system for near real-time detection and diversion of malicious network traffic*. 2016 IEEE Symposium on Technologies for Homeland Security (HST). Retrieved from <http://ieeexplore.ieee.org/document/7568889/>
- Janbeglou, M., Zamani, M., & Ibrahim, S. (2010). *Redirecting outgoing DNS requests toward a fake DNS server in a LAN*. 2010 IEEE International Conference on Software Engineering and Service Sciences. Retrieved from <http://ieeexplore.ieee.org/document/5552339/>
- Jin Wang, Min Zhang, Yang, X., Keping Long, & Chimin Zhou. (2013). *HTTP-sCAN: Detecting HTTP-flooding attack by modeling multi-features of web browsing behaviour from noisy dataset*. 2013 19th Asia-Pacific Conference on Communications (APCC). Retrieved from <http://ieeexplore.ieee.org/document/6766035/>
- Jin, Y., Ichise, H., & Iida, K. (2015). *Design of Detecting Botnet Communication by Monitoring Direct Outbound DNS Queries*. 2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing. Retrieved from <http://ieeexplore.ieee.org/document/7371456/>
- Jose, A. S., & A., B. (2014). *Automatic Detection and Rectification of DNS Reflection Amplification Attacks with Hadoop MapReduce and Chukwa*. 2014 Fourth International Conference on Advances in Computing and Communications. Retrieved from <http://ieeexplore.ieee.org/document/6906023/>
- Kambourakis, G., Moschos, T., Geneiatakis, D., & Gritzalis, S. (2007). *A Fair Solution to DNS Amplification Attacks*. Second International Workshop on Digital Forensics and Incident Analysis (WDFIA 2007). Retrieved from <http://ieeexplore.ieee.org/document/4299371/>
- Karuparthi, R. P., & Zhou, B. (2016). *Enhanced Approach to Detection of SQL Injection Attack*. 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA). Retrieved from <http://ieeexplore.ieee.org/document/7838186/>

- Keyu Lu, Zhengmin Li, Zhaoxin Zhang, & Jiantao Shi. (2016). *DNS recursive server health evaluation model*. 2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS). Retrieved from <http://ieeexplore.ieee.org/document/7737281/>
- Kuisheng Wang, & Yan Hou. (2016). *Detection method of SQL injection attack in cloud computing environment*. 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC). Retrieved from <http://ieeexplore.ieee.org/document/7867260/>
- Kumar, S. (2007). *Smurf-based Distributed Denial of Service (DDoS) Attack Amplification in Internet*. Second International Conference on Internet Monitoring and Protection (ICIMP 2007). Retrieved from <http://ieeexplore.ieee.org/document/4271771/>
- Kyaw, A. K., Sioquim, F., & Joseph, J. (2015). *Dictionary attack on Wordpress: Security and forensic analysis*. 2015 Second International Conference on Information Security and Cyber Forensics (InfoSec). Retrieved from <http://ieeexplore.ieee.org/document/7435522/>
- Lanlan Pan, Xuebiao Yuchi, & Yong Chen. (2016). *Mitigating DDoS attacks towards Top Level Domain name service*. 2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS). Retrieved from <http://ieeexplore.ieee.org/document/7737252/>
- Lin, C., Liu, J., Huang, S., Lee, C., & Chen, C. (2010). *A detection scheme for flooding attack on application layer based on semantic concept*. 2010 International Computer Symposium (ICS2010). Retrieved from <http://ieeexplore.ieee.org/document/5685483/>
- Liu, J., Li, S., Zhang, Y., Xiao, J., Chang, P., & Peng, C. (2017). *Detecting DNS Tunnel through Binary-Classification Based on Behaviour Features*. 2017 IEEE Trustcom/BigDataSE/ICSS. Retrieved from <http://ieeexplore.ieee.org/document/8029459/>
- Malliga, S., Tamilarasi, A., & Janani, M. (2008). *Filtering spoofed traffic at source end for defending against DoS / DDoS attacks*. 2008 International Conference on Computing, Communication and Networking. Retrieved from <http://ieeexplore.ieee.org/document/4787695/>
- Maraj, A., Jakupi, G., Rogova, E., & Grajqevci, X. (2017). *Testing of network security systems through DoS attacks*. 2017 6th Mediterranean Conference on Embedded Computing (MECO). Retrieved from <http://ieeexplore.ieee.org/document/7977239/>
- Matsubara, Y., Musashi, Y., Sugitani, K., & Moriyama, T. (2015). *Open DNS Resolver Activity in Campus Network System*. 2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS). Retrieved from <http://ieeexplore.ieee.org/document/7528907/>
- Maziar Janbeglou, Mazdak Zamani, & Suhaimi Ibrahim. (2010). *Redirecting network traffic toward a fake DNS server on a LAN*. 2010 3rd International Conference on Computer Science and Information Technology. Retrieved from <http://ieeexplore.ieee.org/document/5565196/>

- Mohamed, A. B., & Kandil, A. (2009). *Strengthening and securing the TCP/IP stack against SYN attacks*. Proceedings of the ITI 2009 31st International Conference on Information Technology Interfaces. Retrieved from <http://ieeexplore.ieee.org/document/5196159/>
- Mohan, J., Puranik, S., & Chandrasekaran, K. (2015). *Reducing DNS cache poisoning attacks*. 2015 International Conference on Advanced Computing and Communication Systems. Retrieved from <http://ieeexplore.ieee.org/document/7324091/>
- Naqash, T., Ubaid, F. B., Ishfaq, A., & Fazal-e-Hadi. (2012). *Protecting DNS from cache poisoning attack by using secure proxy*. 2012 International Conference on Emerging Technologies. Retrieved from <http://ieeexplore.ieee.org/document/6375486/>
- Networks, A. (2015). Network Security Infrastructure Report | Arbor Networks®. Retrieved from <https://www.arbornetworks.com/insight-into-the-global-threat-landscape>
- Nissanke, N., & Sun, J. (2008). *A Model for Analysis of SYN Flood DoS Attacks*. Networks 2008 - The 13th International Telecommunications Network Strategy and Planning Symposium. Retrieved from <http://ieeexplore.ieee.org/document/6231359/>
- Nmap: The Network Mapper - Free Security Scanner. (n.d.). Retrieved from <https://nmap.org/>
- Nur, A. Y., & Tozal, M. E. (2016). *Defending Cyber-Physical Systems against DoS Attacks*. 2016 IEEE International Conference on Smart Computing (SMARTCOMP). Retrieved from <http://ieeexplore.ieee.org/document/7501685/>
- Online Diagram Software to draw Flowcharts, UML & more | Creately. (n.d.). Retrieved from <https://creately.com>
- Pijpker, J., & Vranken, H. (2016). *The Role of Internet Service Providers in Botnet Mitigation*. 2016 European Intelligence and Security Informatics Conference (EISIC). Retrieved from <http://ieeexplore.ieee.org/document/7870186/>
- PRTG Network Monitoring Tool - Your All-in-One Solution. (2017, December 12). Retrieved from https://www.paessler.com/network_monitoring_tool
- Qin, J., Li, M., Shi, L., & Yu, X. (2017). *Optimal Denial-of-Service Attack Scheduling with Energy Constraint Over Packet-dropping Networks*. IEEE Transactions on Automatic Control, 1-1. Retrieved from <http://ieeexplore.ieee.org/document/8049303/>
- Sakurai, S., & Ushirozawa, S. (2010). *Input method against Trojan horse and replay attack*. 2010 IEEE International Conference on Information Theory and Information Security. Retrieved from <http://ieeexplore.ieee.org/document/5689592/>
- Sassani, B. A., Abarro, C., Pitton, I., Young, C., & Mehdipour, F. (2016). *Analysis of NTP DRDoS attacks' performance effects and mitigation techniques*. 2016 14th Annual Conference on Privacy, Security and Trust (PST). Retrieved from <http://ieeexplore.ieee.org/document/7906966/>

- Silaen, K. E., & Lim, C. (2016). *A novel countermeasure to prevent XMLRPC WordPress attack*. 2016 International Conference on Data and Software Engineering (ICoDSE). Retrieved from <http://ieeexplore.ieee.org/document/7936147/>
- Takeda, Y., Musashi, Y., Sugitani, K., & Moriyama, T. (2013). *DNS ANY Request Cannon Activity in DNS Query Packet Traffic*. 2013 6th International Conference on Intelligent Networks and Intelligent Systems. Retrieved from <http://ieeexplore.ieee.org/document/6754702/>
- Thongkanchorn, K., Ngamsuriyaraj, S., & Visoottiviseth, V. (2013). *Evaluation studies of three intrusion detection systems under various attacks and rule sets*. 2013 IEEE International Conference of IEEE Region 10 (TENCON 2013). Retrieved from <http://ieeexplore.ieee.org/document/6718975/>
- Tritilanunt, S., Sivakorn, S., Juengjincharnoen, C., & Siripornpisan, A. (2010). *Entropy-based input-output traffic mode detection scheme for DoS/DDoS attacks*. 2010 10th International Symposium on Communications and Information Technologies. Retrieved from <http://ieeexplore.ieee.org/document/5665097/>
- UzmaSattar, Naqash, T., Zafar, M. R., Razzaq, K., & Bin Ubaid, F. (2013). *Secure DNS from amplification attack by using modified bloom filters*. Eighth International Conference on Digital Information Management (ICDIM 2013). Retrieved from <http://ieeexplore.ieee.org/document/6694018/>
- Wang, Q., Dunlap, T., Cho, Y., & Qu, G. (2017). *DoS attacks and countermeasures on network devices*. 2017 26th Wireless and Optical Communication Conference (WOCC). Retrieved from <http://ieeexplore.ieee.org/document/7928974/>
- Williams, R. (2014, June 9). Cybercrime costs global economy \$445 bn annually. *The telegraph* [London]. Retrieved from <http://www.telegraph.co.uk/technology/internet-security/10886640/Cyber-crime-costs-global-economy-445-bn-annually.html>
- Wu, H., Dang, X., Zhang, L., & Wang, L. (2015). *Kalman filter based DNS cache poisoning attack detection*. 2015 IEEE International Conference on Automation Science and Engineering (CASE). Retrieved from <http://ieeexplore.ieee.org/document/7294328/>
- Xiao, L., Matsumoto, S., Ishikawa, T., & Sakurai, K. (2016). *SQL Injection Attack Detection Method Using Expectation Criterion*. 2016 Fourth International Symposium on Computing and Networking (CANDAR). Retrieved from <http://ieeexplore.ieee.org/document/7818686/>
- Yatagai, T., Isohara, T., & Sasase, I. (2007). *Detection of HTTP-GET flood Attack Based on Analysis of Page Access Behaviour*. 2007 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing. Retrieved from <http://ieeexplore.ieee.org/document/4313218/>
- Zargar, G. R., & Kabiri, P. (2009). *Identification of effective network features to detect Smurf attacks*. 2009 IEEE Student Conference on Research and Development (SCORED). Retrieved from <http://ieeexplore.ieee.org/document/5443345/>

Figures references

1. Figure (3): Service Provider Experience Threats. (2016). Retrieved from https://www.arbornetworks.com/images/documents/WISR2016_EN_Web.pdf
2. Figure (4): Infrastructure DDoS Attacks Total Percentage. (2016). Retrieved from <https://www.akamai.com/us/en/about/our-thinking/state-of-the-internet-report/>
3. Figure (5): The ratio of DDoS attack types. (2015). Retrieved from <https://www.slideshare.net/SolarWinds/solarwinds-federal-cybersecurity-survey-2015>
4. Figure (6): Protocols that are used in Reflection attacks. (2016). Retrieved from https://www.arbornetworks.com/images/documents/WISR2016_EN_Web.pdf
5. Figure (7): Targeted of Application Layer attacks. (2016). Retrieved from https://www.arbornetworks.com/images/documents/WISR2016_EN_Web.pdf
6. Figure (8): Daniel CID, D. (2015, September 3). *the percentage of HTTP DoS attack types and also targeted website*. Retrieved from <https://blog.sucuri.net/2015/09/analysing-popular-layer-7-application-ddos-attacks.html>
7. Figure (9): Calyptix. (2015, April 26). *Application layer attack percentage*. Retrieved from <https://www.calyptix.com/top-threats/ddos-attacks-101-types-targets-motivations/>
8. Figure (10): Wikipedia. (n.d.). *SQL Injection attack idea and attack stages*. Retrieved from <https://mn.wikipedia.org/wiki/XXX:Nurmukhamed>
9. Figure (11): Hackmageddon. (2012, Jul 13). *Cyber attack statistics*. Retrieved from <http://www.hackmageddon.com/2012/07/13/june-2012-cyber-attacks-statistics/>
10. Figure (12): CISCO. (n.d.). *Domain Name Space*. Retrieved from <https://www.Cisco.com/c/en/us/about/security-centre/dns-best-practices.html>
11. Figure (13): Saragiotis. (2009, February 27). *Ellaborating on the cache poisoning attacks*. Retrieved from <http://www.saragiotis.gr/posts/tag/dns/>
12. Figure (14): Phuonglm. (2014). *DNS Tunnel illustration*. Retrieved from <http://blog.phuonglm.net/2014/02/access-network-within-tiny-dns-packet.html>
13. Figure (15): DNS hijacking. (2016). Retrieved from <https://heimdalsecurity.com/blog/dns-security/>
14. Figure (16): Arbor. (2012, April 24). *TCP handshake effected by attacker action*. Retrieved from <https://www.arbornetworks.com/blog/asert/ddos-attacks-on-ssl-something-old-something-new/>
15. Figure (17): Slideshare. (2014, October 29). *How DNS reflection attack work*. Retrieved from <https://www.slideshare.net/srikrupa5/dns-security-presentation-issa>