Zudin Rodion

TRANSPORT LAYER DDOS ATTACK TYPES AND MITIGATION METHODS IN NETWORKS

JYVÄSKYLÄN YLIOPISTO
TIETOJENKÄSITTELYTIETEIDEN LAITOS
2015

ABSTRACT

Zudin, Rodion
Transport Layer DDoS Attack Types and Mitigation Methods In Networks
Jyväskylä: University of Jyväskylä, 2015, 70p.
Information Systems, Master's Thesis
Supervisors: Hämäläinen, Timo; Siponen, Mikko

Distributed Denial of Service attacks have been a growing threat to businesses and organizations utilizing information systems with network elements in their activity. With not only financial, but political entities being targeted by the DDoS attacks it is increasingly important to grasp the current situation in this vibrant field of information security. With new attack methods and countermeasures being constantly developed and implemented, the need for the contemporary research is clear.

Five different attack types were found out to be the most popular DDoS attacks in the past year. These attack types were SYN, DNS Amplification, NTP Amplification, DNS and UDP flood attacks. SYN attacks were discovered to make up more than a half of all DDoS attack occurrences, while amplification and multi-vectoring could be seen as a rising trend in attack technologies.

According to the result of literature overview SYN Intercept was found out to be the most efficient mitigation method against TCP SYN, Response Rate Limiting was the most effective against typical DNS Amplification attacks, however leaving to be desired in the mitigation of attacks using varying queries. Modifying NTP servers themselves by removing MONLIST and VERSION functionality was proven to be successful in mitigation of NTP Amplification attacks. As for the DNS attacks go, a combination of three technologies TTL Refresh, TTL Renewal and Long-TTL was deemed superior in mitigating the attacks on DNS servers themselves.

DNS amplification and TCP SYN DoS impact on the web server was measured and analysed in the empirical part of the thesis. Activating SYN Cookies on the web server was deemed to be effective mitigation method against TCP SYN Flood. However, a mitigation technique against DNS or NTP amplification attack to be implemented on a simple small-scale web server without the involvement of ISP or CDN was not discovered.

Keywords: DDoS, information security, networks

TIIVISTELMÄ

Zudin, Rodion
Kuljetuskerroksen hajatetut palvelunestohyökkäystyypit ja niiden lieventämismenetelmät tietoverkoissa
Jyväskylä: Jyväskylän yliopisto, 2015, 70s.
Tietojärjestelmätiede, pro-gradu tutkielma
Ohjaajat: Hämäläinen, Timo; Siponen, Mikko

Hajautetut palvelunestohyökkäykset ovat olleet kasvava uhka yrityksille jotka käyttävät tietoverkkoihin perustuvia elementtejä tietojärjestelmissään. Viime aikoina eivät pelkästään liikeyritykset, vaan myös poliittiset organisaatiot ovat olleet hajautettujen palvelunestohyökkäysten kohteina. Tämän takia on erittäin tärkeää hahmoittaa nykyinen tilanne tässä tietoturvan jatkuvasti muuttuvalla alalla. Hyökkäysmenetelmien ja vastatoimenpiteiden uusiutuessa jatkuvasti, tarve ajankohtaiselle tutkimukselle on selkeä.

Viiden erilaisen hyökkäystyypin on havaittu koostavan suuremman osan hajautetuista palvelunestohyökkäyksistä vuonna 2014. Nämä olvat SYN, DNS vahvistus, NTP vahvistus, DNS hyökkäykset, sekä UDP. SYN-hyökkäysten on havaittu koostavan leijonaosan kaikista hyökkäksistä, kuin taas vahvistuksen ja multi-vektoroinnin on havaittu olevan trendeinä hyökkäysteknologioissa.

Kirjallisuuskatsauksen perusteella SYN Väliintulon on havaittu olevan tehokkain vastatoimenpide TCP SYN hyökkäyksiä vastaan. Vastausvauhdin rajoittaminen (RRL) oli paras vaihtoehto tyypillisiä DNS vahvistushyökkäyksiä vastaan, mutta sen suorituskyky hyökkäyksiä vastaan jotka käyttävät vaihtelevia hakutapoja jätti toivoimisen varaa. MONLIST ja VERSION ominaisuuksien poistamisen NTP palvelimista on havaittu olevan tehokas tapa NTP vahvistus hyökkäyksien vähentämisessä, ja se onkin ehdotettu pääasialliseksi strategiaksi kyseisen tyyppisen hajautetun palvelunestohyökkäyksen kanssa kamppailemiseksi. DNS hyökkäyksiä vastaan yhdistelemällä TTL Päivitystä, TTL Uudistusta ja Pitkää TTL:ää on todettu saavuttavan parhaat lieventämistulokset.

DNS vahvitushökkäysten sekä TCP SYN tulvien suorituskyky verkkopalvelinta vastaan on mitattu ja analysoitu tutkielman empiirisessä osuudessa. SYN Cookies metodin on todettu olevan tehokas keino suojautua TCP SYN palvelunestohyökkäystä vastaan, kuin taas DNS vahvistushyökkäyksen torjumiseksi ei havaittu keinoa yksinkertaisille verkkopalvelimille.

Asiasanat: DDoS, tietoturva, tietoverkot

FIGURES

TABLES

CONTENTS

CONCEPT INDEX

ACK = Acknowledgement

ACL = Access List

ANs = Authoritative Name server

CS = Caching Server

DDoS = Distributed Denial of Service

DNS = Domain Name System

FTP = File Transfer Protocol

HTTP = Hypertext Transfer Protocol

ICMP = Internet Control Message Protocol

IDS = Intrusion Detection System

IGP = Inferior Gateway Protocol

IRC = Internet Relay Chat

IP = Internet Protocol

IPS = Intrusion Prevention System

IRR = Infrastructure Resource Records

NTP = Network Time Protocol

OSI Model = Open Systems Interconnections Model

RAM = Random-access Memory

RTT = Round-Trip Time

RPF = Reverse Path Forwarding

RRL = Response Rate Limiting

SSDP = Simple Service Discovery Protocol

SR = Stub Resolver

SYN = Synchronize

TCP = Transmission Connection Protocol

TFTP = Trivial File Transfer Protocol

TLD = Top Level Domain

TMG = Threat Management Gateway

TTL = Time To Live

UDP = User Datagram Protocol

UFW = Uncomplicated Firewall

1 INTRODUCTION

A world of computers and communications has experienced a revolution with the advent of internet. The internet has become increasingly important to our society, changing the way of communication, business models as well as making all information accessible quickly and easily from almost anywhere, anytime. (Kolahi et. al., 2015).

The internet offers it's users fast, easy and cheap communication mechanisms, enforced with various protocols which make the reliable and timely delivery of messages possible to some extent with certain quality of service (Hussain & Beigh, 2013). However, the internet was not made with a security in mind. With numerous advantages, it, however, can not be considered a safe platform. Technically, internet design can be seen to follow an end-to-end paradigm. The end hosts employ numerous complex functionalities for achieving a desired service guarantee, while the intermediate network full of resources provides a bare-minimum, best effort service. (Hussain & Beigh, 2013).

Distributed Denial of Service (DDoS) attacks are only a one amongst numerous types of threats aiming to compromise the security criterion of information assets defined by Dubojs et. al. (2010). Security criterion (security property) is a property or constraint on business asset that characterizes their security needs. Security criteria act as indicators to assess the significance of a risk. Assets are subject to risks and risks should be evaluated with respect to the security properties that could be damaged.

Traditionally, security properties include *confidentiality, integrity, availability, authenticity, non-repudiation and accountability*. Out of these, the most essential properties are *confidentiality, integrity* and *availability*. The *non-repudiation, authenticity and accountability* can be added if context requires, but they are generally deemed secondary. The security objectives of an information system are defined using security criteria on business assets.

While different types of information threats aim to compromise different security criteria of information assets, the main target of DDoS attacks is the availability. Availability can be defined as the property of being accessible and usable upon demand by an authorized entity (Dubojs et. al., 2010).

1.1 Motivation

DDoS attacks have become the daunting problem for businesses, systems administrators and computer system users. Prevention and detection of a DDoS attack is a major research topic in the information technology. As new counter-measures are developed to prevent or mitigate DDoS attacks, new methods to circumvent these new procedures are developed by the attackers. (Rawal et al., 2013).

Zargar et. al. (2013), classify the incentives of DDoS attackers into five different groups. First one is financial gain, which for example include attacks executed against a web business by attackers recruited by the web business' competitor. Second group is revenge, which include attacks conducted by frustrated individuals in reaction to injustice percepted by them. DDoS attacks from the third group are done based on the ideologcal belief of the attackers, in particual on political agenda. One of the brightest examples from the recent past was a DDoS attack conducted against Estonian government entities in 2007 in a response to removal of Soviet-era memorial statue from the capitals center (Greenmeier, 2007). Intellectual challenge is the name of the fourth group of attack incentives. They are usually conducted by a young computing enthusiasts in order to test their skills. A final group of incentives is cyberwarfare, which include attacks orchestrated by a military or terrorist organizations of the country with the purpose of disrupting the services of another country potentially incurring significant impact on economy and infrastructure. And as in some countries, most of the infrastructure is owned by a private organizations, the effect of DDoS attacks van be truly crippling. The brightest example is United States, where as much as 85% of infrastructure is owned by a private sector which does not willingly spend resources into system protection but rather uses it on business expansion instead, making the systems and the infrastructure vulnerable (Greenmeier, 2007).

There have been numerous DDoS attacks launched against different organizations since the summer of 1999 up until now (Criscuolo, 2000). As one example of the impact of the attacks, in February 2000 Yahoo! Was a target of a major DDoS attack, which kept its services out of the internet for a period of 2 hours resulting in a sifnificant loss of advertising revenue (Wired,com, 2000). In

the most recent example, a hacker activist group called ”Anonymous” executed multiple DDoS attacks against finaancial organizations Mastercard.com, Paypal, PostFinance and Visa.com resulting in those organizations' websites becoming unaccessible (Guardian, 2010).

In a Denial of Service (DoS) attack, an intruder penetrates and depletes a computer system's resources, preventing genuine users from using network's services, such as computer system, web server or a website (Koutepas et al., 2004). As Information System (IS) is a system composed of people and computers that process or interpret information, continuous availability of network's services is crucial to many kind of information systems.

DDoS attack is a synchronized, multiple DoS attack that is launched through multiple compromised machines. The ultimate target for the attack is termed the ”primary victim”, while the cooperated systems participating in the attack are referred to as the ”secondary victims”. The gist of DDoS attacks is that adding many secondary victims in a DDoS attack makes it possible for an attacker to launch a larger and more devastating attack while remaining concealed since the actual attack is launched by a secondary victim. (Rawal et al., 2013).

DDoS attack continues to be a prominent threat to cyber infrastructure of of information systems. It involves multiple DoS agents configured to send attack traffic to a single victim to exhaust it's resources. DDoS is a deliberate act that significantly degrades the quality and availability of services offered by a computer system by consuming its bandwidth and computing resources. As a result, the legitimate users are unable to have full quality access to web services. (Kumar, 2007).

A Denial of Service attack consumes a victim's system resources such as network bandwidth, CPU time and memory. Because the typical DDoS attack aims to deplete available bandwidth and computer resources, the degree of resource depletion depends on the traffic type, volume of the attack traffic and the processing power of the victims system. (Kumar, 2007).

For a long time, DDoS attacks were hard to tackle due to their semantic nature. It means that it is difficult to distinguish an actual attack from a rapid rise of popularity for a given service. (Kuhrer et. al., 2014).

In recent times, businesses utilizing information systems have been targeted by DoS/DDoS attacks. Common targets are gateways, web servers, electronic commerce applications, DNS servers and Voice-over IP servers (Rawal et al., 2013). In a semi-recent report by Arbor Networks (2012), it was concluded that 48% of all cyber threats are DDoS. A number today is potentially over 50%.

1.2 Research problem

The goal of this study is finding out what are the types of DDoS attack types being popular in the past year and how do they work. The research question of the literature overview part is

- How do contemporary widely used DDoS attacks work and how to efficiently mitigate them?

In order to answer to the question answering to the sub-questions of this study is also necessary.

- What are the widely used DDoS attack types in the past year?
- What are the mitigation methods proposed against them?
- How do the mitigation methods work and compare to each other?

As information security is rapidly changing field with new attack technologies and counter measures being discovered and implemented on daily basis, the literature overview is going to be relying not only on the academic papers and releases written and published by the scientific community but also on reports and findings of organizations working in the information security field. Especially in the case of recent technologies and trends, it proves to be extremely difficult to find academic papers related to the subject.

The sources used for data collection for literature overview are IEEExplore, Google Scholar as well as AIS Electronic Library, the database for information systems-related publications. Only the data from most recent research reports, mainly from the last two years was attempted to be included in the study. One reason for that is that there were many DDoS mitigation related publications to be found from the period lasting from 2000 to 2009, but after closer examination the information presented in those studies was deemed to be outdated, as some improved methods based on the older ones were found out to being developed recently and published in more recent publications.

In the empirical part of this thesis, least examined mitigation methods against a single DDoS attack type are going to be analysed using virtual computer network, DoS attacks are going to be simulated, mitigation effectiveness of selected methods is going to be measured and analysed as well as alternative methods will be proposed.

While only technical mitigation methods are examined in this thesis, it should be noticed that one of the basic methods to prevent the occurence of the attacks in general is lessening the attacker's interest in attacking. For example a study of attacker's incentives could help in development of policies to prevent

attacks by causing a loss of interest of attackers by making them face potential financial losses or imprisonment. (Zargar et. al., 2013).

2 ATTACK TYPES

In order to understand the field in which DDoS attacks take place, a basic introduction to networks is done in this study. The Open Systems Interconnections (OSI) model is one of the main models in the sphere. OSI model is a conceptual model characterizing the internal functions of communication system, in this case a network. OSI model has seven layers, each one capable of having several sub-layers (ISO/EIC 1994). The model is displayed below (Figure 1):

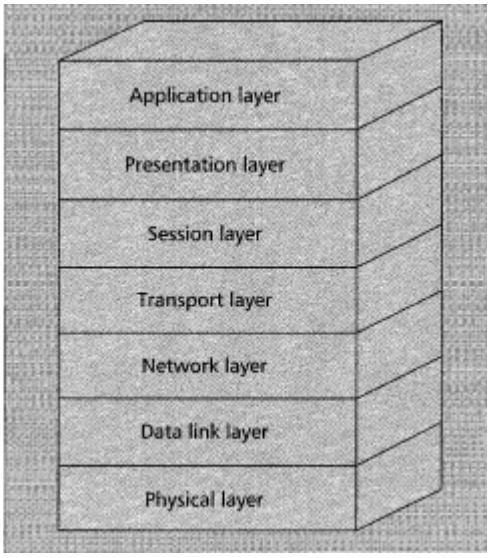

Figure 1: OSI model (ISO/EIC 1994)

The lowest level of OSI model is called physical layer. This layer is comprised of physical networking media and has several major functions. Physical layer defines electrical and physical specifications of the data connection, a relationship between a device and a physical transmission medium as well as a protocol to establish and terminate a connection. Network physical problems,

such as broken wires will affect the physical layer. A physical-connection may involve intermediate open systems, each relaying bit transmission within the layer. (ISO/EIC 1994).

The second layer of OSI model is called data link layer, which provides node-to-node reliable data transfer by detecting and correcting errors occurring in physical layer. It provides functional and procedural means for connectionless-mode among network-entities, and for connection mode for the establishment, maintenance and release of data-link connections among network entities as well as for the transfer of data-link-service-data-units. (ISO/EIC 1994).

Third layer is a network layer, which is responsible for transferring variable length data sequences called datagrams. It also translates logical network address into physical machine address as well as provides transport entities with independence from routing and switching considerations. (ISO/EIC 1994).

Fourth layer is a transport layer which provides the means of transferring variable-length data sequences from a source to a destination host via one or more networks. Some of the features of a transport layer are flow control, multiplexing, virtual circuit management as well as error connection and recovery. (ISO/EIC 1994).

The fifth, session layer defines how to start, control and end a connection between the local and remote application while the sixth, presentation layer established the context and semantics for application-layer entities. The main function of the sixth layer is encryption and decryption of data. (ISO/EIC 1994).

The seventh and final layer of the OSI model is an application layer, with a main function of providing an interface to allow programs to use internet services. (ISO/EIC 1994).

DDoS attacks almost without exception utilize botnets due to their distributed nature. The term bot itself, derived from a work ”ro-bot” is a term used to describe a script or a set of scripts designed to perform some predefined functions recursively and automatically after being triggered intentionally by an attacker or through a system infection (Banday et. al., 2009). While there are two types of bots, benevolent, which are being used to execute legitimate activities automatically and malicious, which are meant for harming purposes, botnets utilized by DDoS attackers belong to the latter group.

Botnet can be defined as a network of infected machines, which are controlled by a human operator, botmaster (Rodriguez-Gomez et. al., 2013). There are some IRC channels which offer specialized training programs for creation and utilization of botnets (Lannelli & Hackworth, 2006).

While code may be developed or modified by an attacker in order to create a personal bot, ready-made, highly tailorable bots with easy-to-understand instructions as well as simple character and graphical interfaces are

being sold on the internet. After creation, the bot must be propagated to multiple vulnerable systems in order to create a bot network. There are several ways to do that, including infection using direct and indirect techniques. These techniques include abusing software vulnerabilities, social engineering using email, instant messaging as well propagation utilizing peer to peer networks, file sharing among other methods. FTP, HTTP and TFTP protocol based services are mostly used by attackers to infect computers in order to empower the botnet until sufficient strength is achieved. (Banday, et. al., 2009).

After infecting and discovering compromised systems, the victim machines have to be controlled by a botmaster using some kind of communication in order to carry out malicious operations. Several organized command languages and control protocols called Command and Control (C&C) techniques are utilized in order to operate botnets remotely. (Banday, et. al., 2009).

Botnet lifecycle defined by Rodriguez-Gomez et. al. (2013), consists of six phases, which are important to know in order to understand the underlying workings of what is considered the driving force of DDoS attacks.

First phase is botnet conception. The main characteristics of the botnet are conceived in this phase influenced by an ultimate intented purpose of the botnet. Motivation, design and implementation are the three cornerstones of the botnet conception phase. Motivation, more often than not financial, acts as a igniting spark of botnet creation. (Rodriguez-Gomez et. al., 2013).

Design of the botnet architecture can be centralized, distributed or hybrid. In a centralized model, bots communicate with C&C server with the purpose of receiving information from the botmaster. The quickness of the communications can be considered as a major advantage of centralized model. In a distributed architecture, all the bots have a status of both a server and a client. With no single point of failure, this kind of solution is stronger than a centralized one, but also much slower. The hybrid botnets combine the strong points of two previous solutions by implementing multiple distributed networks with multiple centralized servers, removing a single vital point of failure while upkeeping fast operation. After the botnet has been conceived and designed, it can be implemented using any of the software development processes. (Rodriguez-Gomez et. al., 2013).

Second phase is botnet recruitment, which consists of recruiting individual bots. According to Provos et. al. (2009), recruiting is based on remotely abusing servers' vulnerabilities as well as spreading of trojan and other malware.

Next is the phase called botnet interaction, which includes registering the bots into botnet and creating the C&C network for controlling and managing the bots. Interaction processes can be divided into internal and external. (Rodriguez-Gomez et. al., 2013).

Internal interactions consists of the messaging between the botnets and botmaster only and have two different types. First one is registration process, through which a compromised host becomes an effective part of the botnet. Second one is called C&C Communications, which consists all the communications after the registration process is finished. External interactions are the communications between a member of a botnet and a noncompromised system. (Rodriguez-Gomez et. al., 2013).

Fourth phase of botnet lifecycle is botnet marketing, during which the botnet is publicized in order to attract potential customers and users. Marketing is usually done by either selling the botnet code or more commonly renting the botnet services on the internet. (Rodriguez-Gomez et. al., 2013).

During the fifth phase, after defining the users of the botnet, the DDoS attack itself is executed and is potentially successful. It should be noted, that the botnet can be used not only for DDoS attacks but for other malicious purposes such as spamming, phishing, data stealing and click fraud. (Rodriguez-Gomez et. al., 2013).

Obviously, the phases of the life cycle can be occuring without a specific order. After a successful attack, a botmaster can return to the fifth phase in order to execute a new attack while continuing the process of the second phase by continuous botnet recruitment.

There has been important changes in the nature of DDoS attacks occurring in the past year. One of them is a multi-vector approach. While traditionally, DDoS attack campaigns used a single attack type, or vector, recently there is a rise of DDoS attacks using multiple vectors. Called multi-vector attacks, they are a combination of the volumetric attacks, state-exhaustion attacks and application layer attacks. This approach is very appealing to the attacker, since the tactic can cause the most collateral damage to a target. Typically several different network resources are targeted or one attack vector is used as a decoy while another, more powerful one is used as a main weapon. (Imperva, 2015).

Thus, vector can be seen as a single DDoS attack type. For example application-layer attack such as HTTP GET can be seen as a one vector in multi-vectored DDoS attack. As HTTP GET attack acts as a decoy in order to distract the defender, the more powerful network-layer DNS Amplification attack is a second vector, and is executed as a main weapon against the target.

Arbor Networks, a software company selling network security and network monitoring software provides information about occurrence rate of attack types classified by vector in its Worldwide Infrastructure Security Report (Figure 2). Information represents the data gathered using surveys during 2014 gathered from 287 organizations from around the world ranging from internet access providers to content services.

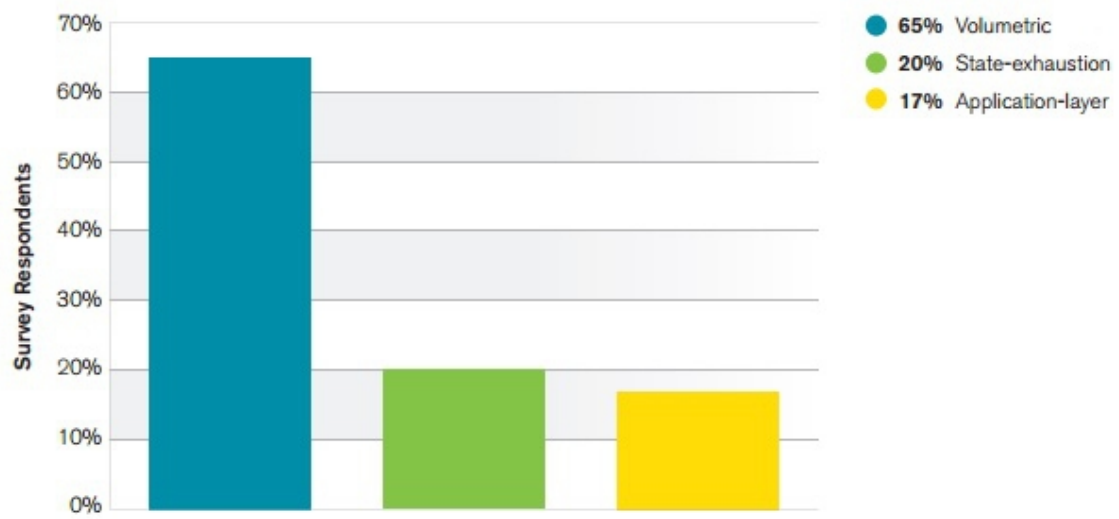

Figure 2: DDoS attack types by vector (Arbor Networks, 2015)

As we can see, 65% of survey respondents report being targeted by volumetric attacks, while only 17% have experienced application layer DDoS. According to Arbor Networks (2015), the proportion of volumetric attacks has slightly risen in expense of the drops in state-exhaustion and application layer attacks. The percentages represent the proportion of the total number of the attacks experienced by the survey respondents.

In essence, if classified by the damage type, there are two main types of DDoS damage types. One is bandwidth depletion, while another is resource depletion. A bandwidth depletion attack is designed to flood the victim's network with unwanted traffic, that prevents the legitimate traffic from reaching the victim's system. A resource depletion attack on the other hand, is an attack that is designed to tie up the resource of a victim's system. This type of attack targets a server or process on a victim's system, making it unable to process legitimate requests for service. (Hussain & Beigh, 2013)

The recently released white paper for the last quarter of 2014 by Akamai, a cloud service provider, called the state of the internet, can not be overlooked when accumulating data about DDoS occurrences in the recent past (Figure 3).

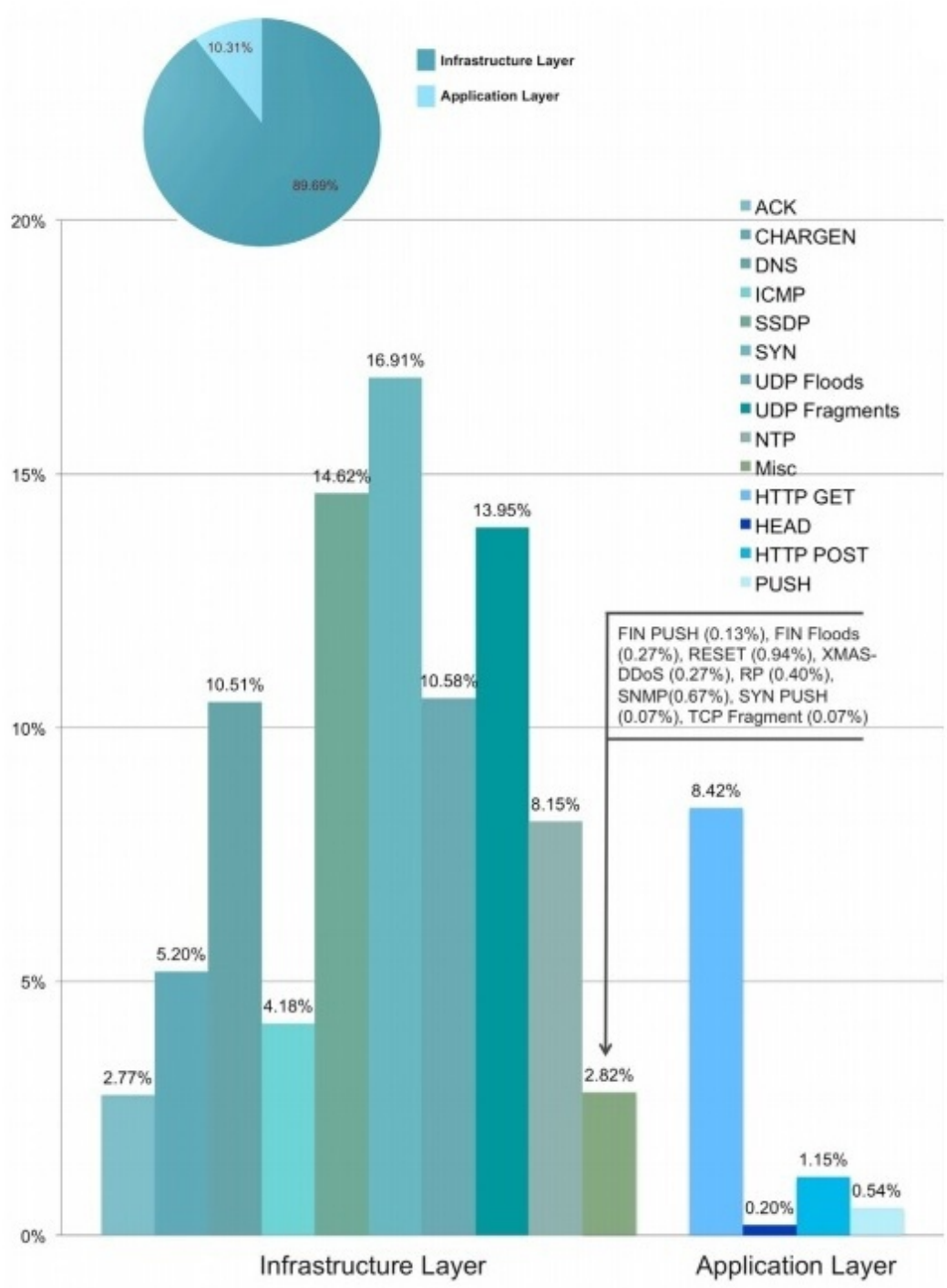

Figure 3: Types of DDoS attacks and their relative distribution in Q4 2014 (Akamai, 2015)

It can be seen that the data related to attack distribution by vectors from Akamai report is in line with information provided by Arbor Networks. The occurrence of application layer attacks is noticeably smaller compared to the volumetric and state-exhaustion vector attacks occurring in the infrastructure layer. In the same way as Arbor Networks present their vector distribution data, Akamai's graph also present the values as the percentages of the total

number of attacks, but while Arbor Networks base their data on the surveys filled by service providers and various businesses, Akamai make their graphs based on the instances recorded on the Akamai's PLXrouted network, deployed to serve multiple customers from all around the world.

It is also worthwhile to take under inspection a recent research by Imperva (2015), a provider of cyber and data security products, which provides a graph separating DDoS attack types by type and occurrence (Figure 4).

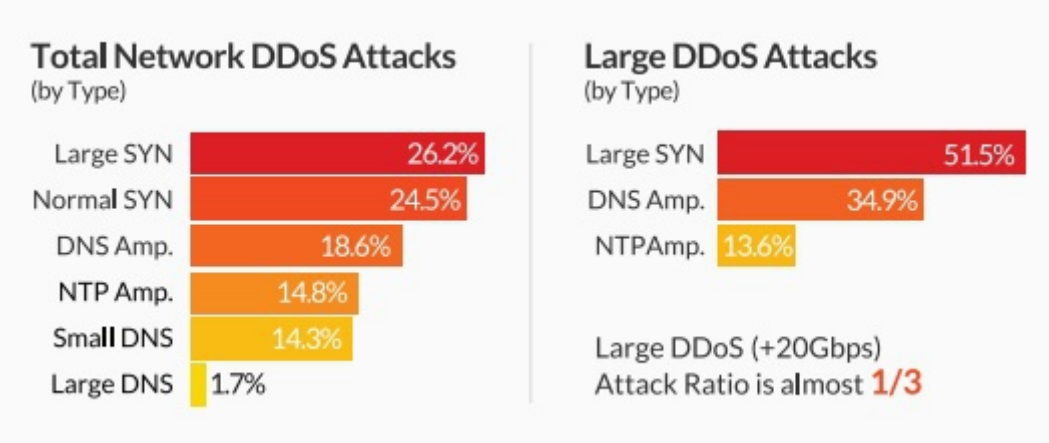

Figure 4: Network DDoS Attacks by type (Imperva, 2015)

It can be observed that according to Imperva, Large SYN, Normal SYN, DNS Amplification, NTP Amplification, Small and Large DNS attacks constitute over 90% of all DDoS attacks in the recent past. Akamai, however claims that at least in the last quarter of 2014, SSDP, SYN, UDP Floods and DNS attacks take the lion share of the attack occurrences.

CDNetworks (2015), a full-service content delivery network business provides their own input to the topic, with similar findings and additionally high reported number of occurred UDP flood attacks. While both charts are based on a differing data collected by the respective organizations, the attack types seemingly prevalent according to the three graphs are examined closer in this study. With CDNetworks graph also providing the information about change from year 2013 to 2014, it can serve as a reference for recent trends in the sphere of attack types (Figure 5).

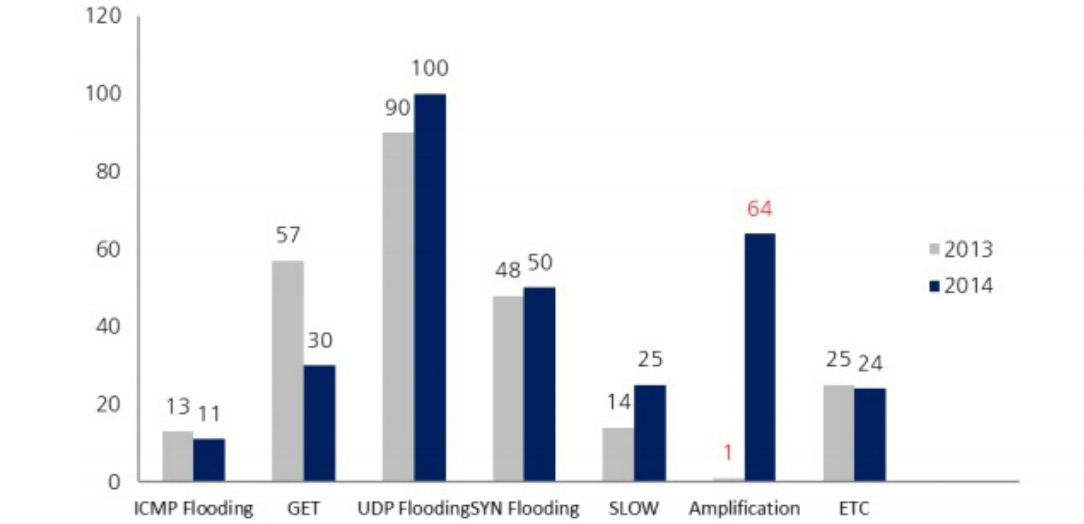

Figure 5: DDoS attack analysis by type (CDNetworks, 2015)

The lack of DNS attacks in the chart by CDNetworks must be because the attack data was measured only from the CDNetworks customers, which are mostly small and medium businesses, which probably do not include any DNS server administrators.

It should be noticed, that the statistics provided by the three organizations mentioned above consists of information collected from their customers to whom they provide DDoS mitigation among other services. While separately they can not serve as an objective overview representing the whole world, their graphs were compared with each other in an attempt to define the attack types being common in recent years.

Attack types taken under closer inspection are SYN Flood, DNS Amplification, NTP Amplification and DNS attacks. Amplification attacks have been reported to be the trend of 2014, with little to no occurrences in 2013. SYN flood was chosen because of it's popularity as it has been mentioned in all the examined reports from 2014. DNS attacks were mentioned by both Imperva (2015) and Akamai (2015) to have a big share of all the attack cases, which was the reason for including them in the study.

While application layer DDoS attacks are becoming more and more popular, this thesis concentrates on transport layer attacks, which according to Mirkovic & Reiher (2004), can be classified into four different types.

Flooding attacks, which aim to disrupt legitimate user's connectivity by exhausting target systems network's bandwidth. Out of the attacks studied in this thesis, UDP Flood and DNS Flood belong to this type.

Protocol explotation flooding attacks abuse some of the specific features of the victim's protocols in order to consume a lot of victim's resources. SYN Flood examined in this thesis belong to this type.

Reflection-based flooding attacks cover attacks executed by attacker sending forged requests to the reflectors with the reflectors replying to the victim exhausting their resources. DNS Amplification and NTP Amplification belong to this type, as they are executed by sending forged requests.

Amplification-based flooding attacks are a fourth type of transport layer DDoS attacks. They are executed in a fashion where attackers exploit services to amplify a traffic they redirect to the victim. As amplification-based flooding attacks usually require a forged source IP addresses, they are commonly executed in a tandem with reflection based flooding attacks. Thus, DNS Amplification DDOS and NTP Amplification DDOS can be considered to belong to both reflection-based and amplification-based attack types.

2.1 SYN Attacks

A TCP SYN flood attack is a type of DoS attack in which an attacker sends a huge quantity of SYN requests to targeted system in order to consume sufficient amount of server resources and bandwidth to make a system unavailable to legitimate traffic (Eddy, 2006). A SYN request is a part of a three-way handshake of connection establishment used by a Transmission Connection Protocol (TCP). TCP is the protocol that major internet applications rely on for reliable data stream service in a transmission layer of OSI Model. TCP/IP protocol suite is the most widely used protocol suite for data communication (Kavisankar & Chellappan, 2011).

All mega DDoS attacks with traffic of over 100 GBPS measured by AKAMAI in Q1 2015 included TCP SYN flood as an attack type, making TCP SYN flood responsible for big attacks against gaming sites and services during the past year (Akamai, 2015).

The internet today is driven by machines that communicate using services layered on top of the TCP/IP protocols of the transmission layer. These protocols include HTTP, FTP and SSH, among others. The accessibility of these services is dependent on how well the underlying transport protocol performs, which in the sphere of TCP SYN flood attacks is TCP. If TCP is unable to deliver the layered service to a remote machine, the user perceives the site as being dead or inaccessible. While this may have been merely a small inconvenience in the past, this is becoming much more serious problem today as machines are being used for commerce and business. (Lemon, 2002).

By a generic design of a TCP protocol any application is required to complete a three-way handshake before data transfer is possible. As the name suggest, there are three stages in a TCP three-way hand shake.

First TCP client initiates a connection request to TCP server with a SYN bit set in the flags in the TCP header. In the second step a TCP server responds with a TCP with a TCP segment with TCP SYN and ACK bit set in the flags after receiving a TCP SYN segment from a client. In the third phase a client responds with a TCP segment with ACK bit set in the flags. (Samad et. al., 2014).

After completing the three-step process described above, TCP connection is established. However, in a TCP SYN Flood attack, the attacker exploits this behaviour of the TCP protocol. First the attacker crafts a TCP segment with a SYN bit set and sends it to the target server. As per three-way handshake, the server on receipt sends a response with a SYN and ACK bit set to the attacker. The corresponding state of the TCP connection in the TCP state table of the server would now progress to the SYN-RECEIVED state. Now, according to the three-way hand shake of the TCP protocol, server would be waiting for the receipt of the TCP segment with the ACK bit set from the attacker in order to complete the three-way handshake and progress to the ESTABLISHED state. In the TCP SYN Flood attack however, the ACK response never comes. (Samad et. al., 2014).

The ACK response never comes, as the attacker's machine can be configured to ignore the SYN-ACK packets from the target. Each half-open connection will remain on the memory stack until it times out. SYN-ACK is commonly re-transmitted by the server 5 times, doubling the time-out value after each retransmission. In the default case of time-out value being 3 seconds, half open connections are kept open 96 seconds, which results in the accumulating SYN requests filling up the memory stack and crippling the services of the system. (Kavisankar & Chellappan, 2011).

In TCP SYN Flood attack, the goal of the attacker is to fill up the TCP half open states which are allowed for the target system. When the maximum allowed number of half open states is filled up in the memory, the connection requests from the legitimate users are dropped and the server runs out of resources crashing, creating a Denial of Service for the application of valid users. (Ohsita et. al., 2012).

From an attacker]s point of view, there are multiple benefits to using TCP as an attack protocol. The benefits include facts that providers cannot easily block or filter TCP traffic related to well-known protocols as they are widely in use. It is also difficult to distinguish attacks from normal traffic in a stream of TCP control segments and there are millions of potential TCP amplifiers, so fixing them is an unfeasible option. (Kuhrer et. al., 2014).

Another fact worthwhile noticing is that according to Lemon (2002), the attacker does not have to be on fast machine or network to execute a TCP SYN

flood attack. Standard TCP will not time out connections until a certain number have been made, which usually is a total of 511 seconds (Wesley, 1993).

Under assumption that a machine permits a maximum of 1024 incomplete connections per sockets attacker needs to send only 2 connection attempts per second to exhaust all allocated resources. While this by itself does not form a DoS attack as existing incomplete connections are dropped when a new SYN request is received, by forcing the server to drop incomplete connection state at a rate larger than the round trip time (RTT), an attacker is able to insure that no connections are able to be established completely. RTT stands for the time required for the server to send a SYN, ACK and have the client reply. (Lemon, 2002).

In his study, Lemon (2002) elaborates further on the practical implementation of the attack. According to him, each connection is dropped with the probability of 1/N, and if the goal of the attacker is to recycle every connection before the average RTT, machine would be needed to be flooded with a rate of N/RTT packets per second. If we assume the size of the listen queue to be 2048, and RTT to be 100 millisecond, 20480 packets per second would have to be sent. As a minimal size of TCP packet is 64 bytes, the total bandwidth used would be 1.25Mb/second, which is totally achievable.

It is worthwhile to notice that an attacker can also launch a DDoS attack on the target victim server using the Spoofed IP address. During the attack, the attacker sends SYN packets with source IP addresses that do not exist or are not active. In the similar way as in the SYN Flood attacks not using the spoofed addresses, the server will not receive confirmation packets for requests created by the SYN flood attack. IP address spoofing is the main case for amplification attacks, as it makes it possible for an attacker to specify arbitrary targets to be flooded (Kuhrer et. al., 2014).

An alarming new types of SYN Flood attacks have been detected in the recent past. Radware Emergency Response Team (2014) has classified a new type of SYN Flood attack called Tsunami. While in a common SYN Flood attack the TCP SYN packets sent by an attacker are empty containing no other data except the connection request, in the Tsunami SYN Flood attack packets are not empty. In the two instances observed in the final months of 2014, each of the SYN packets contained as much as 1000 bytes of data per packet, making the bandwidth footprint of the attacks gigantic. This kind of an attack is more likely to saturate the internet pipe of the victim. (Radware ERT, 2014).

While an ordinary TCP SYN packet only contains 40-60 bytes of data, TSUNAMI sends around 20 times more data per handshake, causing network saturation of the target. The regular SYN Flood attack with the small packet sizes is capable of crippling target's server resources such as CPU, but is not

intended for burdening the network itself. Tsunami is essentially a large SYN Flood attack.

An attack type targeting the target's network as well as server resources is called Combo SYN Flood attack. According to Imperva (2015), a combo SYN flood comprises two types of SYN attacks, one uses the regular SYN packets with data size of around 50 bytes and the other uses large SYN packets with size topping 250 bytes per packet. A SYN Flood attack can thus be considered a multi-vector approach.

Gupta et. al. (2010) claim that TCP SYN flooding has remained one of the most destructive attack techniques since September 1996.

2.2 DNS Amplification attacks

The Domain Name System (DNS) is a naming system for resources connected to the internet or a private network. It is essential in the functionality of the most internet, because it is the Internet's primary directory service. DNS functions in the application layer of the OSI model.

The functionality of DNS servers, the core of the DNS is as follows. When DNS server receives a DNS query, it tries to respond by searching the DNS data in the cache. A cache is a set of domain-name records separately associated with a time-to-live (TTL) value. A domain name is removed from the cache if its TTL expires. If a matching record for the DNS query is found in the cache, the server responds with it. If matching record is not found, the server searches for the closest zone in the hierarchy that encloses the query and caches the information. After that, starting from the closest enclosing zone, the DNS server travels down the DNS zone hierarchy tree by querying subsequent sub-zones. This continues until the zone responsible for the domain-name is reached and included in the answer to the query, a traversal can not go on and error is responded or server fails to get response from any relevant zones during the traversal sending the ”server failure” answer to the query. (Li et. al., 2010).

Fachkha et al. (2014) state, that in order to have as high impact as possible, the attackers use DNS requests of type *ANY* to return all possible known information to the victim increasing the amplification of the attack.

According to Rozekrans & de Koning (2014), originally resolvers were utilized for traffic reflection, but recently amplification attacks relying on Authoritative Name Servers (ANs) for amplification have been increasing. One of the reasons is speculated to be that more and more resolver operators are

following the access restriction guidelines provided by RFC5358 (2008), which significantly reduce the chance of their servers being used in a reflection attack. On the other hand authoritative name servers can not follow the guidelines presented in the RFC5358 and are being used more and more for amplification attacks.

DNS Amplification DDoS attacks are bandwidth exhaustion attacks, which utilize the connectionless User Datagram Protocol (UDP), which is a part of a transport layer of OSI model. The first step of the preparation for the DNS Amplification DDoS attack is spoofing the IP address of the target. After acquiring the address, a multitude of queries are sent to the name servers across the internet. The name servers respond with instigated large responses up to 4096 bytes to the spoofed address of the attacker.

Typically, attackers will submit a request with as much zone information as possible to maximize the amplification effect. Because the size of the response is considerably larger than the size of the request, the attacker is able to increase the amount of traffic directed at the victim. By utilizing a botnet to produce a large number of spoofed DNS queries, the attack size can be amplified with ease. (US Cert, 2013).

Rozekrans & de Koning (2014), state that DNS amplification attacks can be divided into three types, which are repeating queries, varying queries and a distributed attack.

A repeating query attack is an attack, which requests the same record over and over again. As mentioned previously, usual query to use is ANY, as it returns all the records for a specific domain name resulting in a massive amplification. (Rozekrans & de Koning, 2014).

Varying query attack is a tool for an attackers to use if their simple approach of repeating query is mitigated. The varying query attack sends queries for varying domain names to the DNS server. This makes an attack less obvious as unique responses are getting extracted. (Rozekrans & de Koning, 2014).

A TLD name server usually include a single large zone file which contains a large collection of domain names. Before performing a varying query attack on a server, attacker has to have some information about the domain names which are inside the zone. If a dictionary attack or a webcrawler is used, in a matter of several attacks a wide selection of attack scenarios is simulated resulting in attacker getting 100% resolvable domain names as an answer after only 5 attacks. Another method is the abuse of Next-Secure (NSEC) records to gather information about target zone. NSEC records are designed to be used to prove a name does not exist by pointing to the previous and the next record. (Rozekrans & de Koning, 2014).

Rozekrans & de Koning (2014) finish by stating that the repeating query attacks and varying query attacks can be enhanced by distributing the attack traffic over multiple DNS servers.

2.3 NTP Amplification attacks

Network Time Protocol (NTP) is a networking protocol for clock synchronization between computer systems over data networks, which is a part of application layer of OSI model . According to Rossow (2014), NTP is only one of at least 14 UDP protocols vulnerable to amplification abuse. The attacks were the most imminent in the beginning of 2014, with number of NTP amplification attacks surging up by 371% (Reading, 2014).

After running extensive fingerprinting tests, Kuhrer et. al. (2014), managed to classify an OS distribution for NTP amplifiers. According to the results, over 40% of the vulnerable NTP hosts ran Cisco IOS, which is an OS that is deployed on Cisco devices including business routers and switcher. Over 17% of the amplifiers ran Linux on MIPS and around 5% were running Linux on PowerPC. The last two are stated to be common combinations for consumer devices such as routers and modems. It can be concluded that majority of NTP amplifiers run on networking equipment.

NTP amplification attacks are bandwidth exhaustion attacks, which work in a similar way as DNS amplification attacks. After target's address is spoofed, a feature called MONLIST on NTP servers is being exploited (Minerva, 2015). MONLIST is a command that requests a list of the last 600 hosts connected to that server. In a similar way to DNS amplification attacks, a small query can be amplified into a large amount of data in the response redirected to the spoofed target's address. According to Minerva (2015), there are more than 400,000 NTP servers around the world that can potentially be used in an NTP amplification attack. Some are capable of amplification factors up to 700 times, which is massive. Rossow (2014) comes up with even a higher number, stating that in the worst case MONLIST is capable of amplification factor of 4670.

It should be noticed, however, that while some of the NTP servers may have a large amount of traffic, there are servers with less than 600 hosts ever connected, which will result in a lower degree of amplification compared to the high-traffic, vulnerable NTP servers.

It is also worth noticing, that both NTP amplification attacks and DNS amplification attacks use UDP as their transport protocol. While TCP has a three-way handshake procedure to start a connection, UDP does not, which makes it impossible to know if the UDP packet indeed comes from an address

the packet's source address indicates. This makes the spoof attacks, which are prevented by the three-way handshake process of the TCP protocol, possible.

NTP amplification was used in one of the biggest DDoS attacks in history. In February 2014, there was a 400 Gbps attack against a French hosting provider. It has been speculated, that if the attacker had even more resources to send spoofed MONLIST requests, the impact would have been even higher. (Prince, 2014).

2.4 DNS Attacks

While DNS Amplification attacks only abuse DNS servers to amplify the attack traffic into the spoofed address, DNS attacks target the DNS servers themselves.

According to the Fanglu et. al. (2006), there is one main DDoS attack strategy against DNS servers. It consists of simply sending a large number of DNS requests to the server in order to overload it. As the standard DNS server cannot distinguish between a spoofed and non-spoofed requests, the only choice is to handle all of them and indiscriminately start dropping requests after becoming overloaded. With legitimate requesters interpreting drops of requests as a sign of congestion backing off their timer for retransmission, the amount of legitimate requests served by overloaded servers are drastically decreased.

Due to its hierarchical structure, the DNS availability depends on a small number of servers that serve the root and other important top level domains (Vasileios et. al., 2007). A number of DDoS attacks have been directed against those top-level DNS name-servers, with two most noticeable being conducted in October 2002 and February 2007. While according to Vasileios et. Al (2007) the impact on the overall DNS availability was debatable, some attacks did succeed in disabling the targeted DNS servers resulting in some parts of the internet suffering from severe name resolution problems.

Essentially, a DNS name space is divided into a large number of zones. Each zone is authoritative for the names that share the same suffix with the zone's name, while a zone can also delegate a part of it's name to another zone, referred as a child zone. Generic top-level domains (gTLD) and country top-level domains (ccTLD) appear directly below the root. (Vasileios et. al., 2007).

A DNS name space structure can be imagined as a tree, with the top-level domains being at the top, and names using the suffixes of the top domains being below them. For example a source *ieeexplore.ieee.org* is in a DNS zone *ieee.org* which is under a top-level DNS zone *.org*. If a *ieee.org* DNS zone would

be rendered inaccessible by a DDoS attack, all resources in DNS zones and subnets under that zone will be as well. A success of the attack depends on resources of the attacker and defender. DDoS attacks can easily succeed if a zone is served by a small number of servers.

Vasileios et. al. (2007) state that there are mainly three factors that affect the end-user experience of a successful DDoS attack against DNS. First is position of a target zone. If the zone is *stub*, meaning not used in order to access the name servers of other zones, the attack will only naturally affect the names defined in the targeted zone. Second is the popularity of the target zone, i.e. the number of referrals provided by the target zone. The third factor is resource record caching. Even if some zone becomes unavailable due to DDoS attack, the record of these zones may be cached in some caching servers and still be accessible.

2.5 UDP Flood attacks

A User Datagram Protocol (UDP) flood attack is done by attacker crafting numerous packets to random destination ports on the victim's computer. On the receipt of the UDP packet requests, the victim system would respond with the appropriate Internet Control Message Protocol (ICMP) packets, in the case the port is closed (Singh & Junefa, 2010). A large number of these packet responses would slow down the system or cause a crash, making the resource unreachable for other clients. (Kolahi et. al., 2015).

In order to hide the identity of the attacker, the attacker often spoofs the source IP address of the attacking packets. UDP flood attacks may also deplete the bandwidth of network around victim's system, impacting other systems around the victim. (Sejdini et. al., 2006).

3 MITIGATION METHODS

For the examining of mitigation methods against attack methods described in the previous chapters, TCP Syn, DNS Amplification, NTP Amplification and DNS attacks were chosen for a closer inspection. DNS attacks have been one of the most prevalent attack types for many years continuously making it impossible to ignore. DNS and NTP Amplification were chosen for their extreme relevance as they can be seen as the main trend of DDoS attacks in 2014. Provided that TCP Syn is the most widespread and used DDoS attack type according to numerous sources, choosing it for closer inspection in this chapter is natural.

Mirkovic et. al., 2004, classify DDoS attack countermeasures into two categories: proactive techniques and reactive techniques.

DDoS attack detection is a vital part of reactive DDoS mitigation. Mainly, there are two methods to detect the attack traffic via intrusion detection systems (IDS) and intrusion prevention systems (IPS) which are are signature-based technique and anomaly-based technique (Purvanto et. al., 2014).

Signature-based detection technique consists of matching the packet signature with existing attack signatures in a database. If a database is adequately populated, the technique has a strong point of having low false positive, but is unable to detect attacks that are not in the database. This is an enormous weakness considering the possibility of a new or modified attack. (Purvanto et. al., 2014).

Anomaly-based detection technique is based on detecting changes from normal patterns. However, there are some challenges distinguishing DDoS attacks from recenlty widespread phenomena called flashcrowd.

Flashcrowd is an occurence, where the number of users of a web service increases significantly during a specific event. From the quality of service perspective, the increased amount of users should still be served. However, from an anomality detection point of view, it is different to distinguish the

flashcrowd from the DDoS attack, creating one challenge for the anomaly-based detection techniques. Li et. al. (2009), propose hybrid probability metrics to detect DDoS attacks and distinguish them from flashcrowds.

3.1 Against TCP SYN

Peng et. al. (2004) claim that all the efficient defences against SYN flooding attacks can be categorized into four: firewall-based, server-based, agent-based and router-based. Firewall-based defence mechanism acts on behalf of the services. The packet needs to be inspected before it goes to the desired server. Server-based defence mechanism is where server monitor keeps the table of incomplete queued connections resulting in removal of need for the server to watch half-open connections. Agent-based mechanism is a software developed the mitigation of SYN flooding attacks in mind. Its purpose is to continuously monitor the TCP-three way handshake messages before the server reply. The last defence mechanism, router-based distributed packet filtering (DPF) exploits routing information to determine if packet arriving at the router matches with its inscribed source and destination addresses.

In the sections below, mitigation methods presented in academic papers and publications in recent years are classified by the type following the classification proposed by Peng et. al. (2004). In the final analysis chapter results from any conducted comparisons and experiments are compiled and presented.

3.1.1 Server-based defence

SYN cache and SYN cookies are two examples of server-based defence introduced in the paper by Lemon (2002). As mentioned before, the point of the SYN flood attack is that the malicious host sends a large number of TCP open requests, which are known as SYN packets. When the server receives this packet, it is interpreted as a request by a remote host to initiate a TCP connection, at which point the machine allocates resources to track the TCP state. By sending large amount of these requests in a short period of time, attacker can exhaust the resources on the machine to the critical point where it becomes unresponsive or crashes.

Because there is a way for an attacker to forge their source IP address, a defence relying on filtering packets based on the source IP will not be effective. Another benefit of using a random source IP address for an attacker is that it

will cause more resources to be tied up on the server in a case per-IP route structure is allocated. (Lemon, 2002).

Lemon (2002), elaborates that usually it is impossible to distinguish attacks from real connection attempts, other than by observing the volume of SYNs that are arriving at the server. In order to defend against SYN attack, the amount of state that is allocated should be reduced, or even better eliminated by delaying allocation of resources until the connection is completed. Two ways are proposed in the study by Lemon (2002), SYN cache and SYN cookies.

SYN cache is a mitigation approach, where server allocates minimal state when the initial request is received, and only allocate all the resources required when the connection is completed. While the amount of allocated resources per connection is minimal, it is still possible to encounter resource exhaustion in a situation with many SYN requests arriving from an attacker. Modifications to the code in order to handle state overflows and prioritize the packets should be prepared, according to Lemon (2002).

SYN cookie is another mitigation approach, where the server allocates no state, instead sending a cryptographic secret with the SYN,ACK back to the originator, which is called a cookie. However, because if using this method, no state is stored on the machine, but all information carried by the initial SYN requests such as the desired MSS, requested window scaling, use of timestamps among other information is encoded and sent back to the client, all the TCP options of the initial request are not possible to be included into the cookie. The loss of possibility for certain TCP performance enhancements with the loss of these options can be considered as a drawback of SYN cookie method. (Lemon, 2002).

Lemon (2002), claims that there is also a secondary problem related to the SYN cookie method. The problem is that the TCP protocol requires unacknowledged data to be re-transmitted. As according to the protocol, the server is supposed to re-transmit the SYN,ACK before dropping the connection, ultimately sending a reset (RST) to the client to shutdown it. When SYN, ACK arrives at a client but the return ACK is lost, disparity about the established state between the client and the server occurs. While normally this case would be handled by server re-transmits, if SYN cookies are utilized there is no state kept on the server making a re-transmission not possible.

In his study, Lemon (2002), continues to elaborate in the issues of the cookies method. According to him, cookies have the property that the entire connection establishment is performed by the returning ACK, which is independent of the preceding SYN and SYN, ACK transmission. This fact makes it possible for the attacker to flood the server with ACK requests with random values, hoping that one of them will be correct allowing a connection to

be established. This also made it possible to bypass any firewalls being potentially utilized by the server side restricting external connections by filtering out incoming packets which have the SYN bit set, since only ACK is required to establish the connection.

Another paper written by Bo & Ruimin (2009), presents a novel SYN cookie method, which is claimed to be superior to the original one. While the authors agree that the SYN cookie is an effective way to prevent DoS attacks against TCP, there are some issues such as high computational complexity. And as Bo & Ruimin (2009) admit that there are some improved SYN cookie programs, they also criticize them for different users. While iterative algorithm proposed by Jianying et. al. (2007) is agreed to reduce the CPU utilization rate compared to the original approach described by Lemon (2002), its weakness is pointed out to be a consumption of the additional storage space. Method proposed by Di & Wensheng (2007) is praised for improving the defence system efficiency, but criticized for its limitations such as possibility of utilization only in setups, where the defence system is separated from the server. Lastly, the method described by Xiaochun et. al. (2008), while reducing the stress on the CPU, is claimed to require additional system resources to maintain a HASH table, increasing the waiting time for a normal TCP connections.

The method proposed by Bo & Ruimin (2009), includes a novel cookie calculation algorithm, which modifies the 32 bit sequence number field definition used to store TCP cookie. The whole method includes three main components, which are the controller, attack detector and attack responder. The controller plays the main control role in the system, while the other two components have only a single function respectively and are used for attack detection.

When the first detection component spots any abnormal flow of data, the second detection component uses high detection standard to determine if the abnormal flow of data is an attack or not. If it is, then the attack responder, a third component of the method will be called by a controller and start processing. The attack responder has to generate the cookie as described in the SYN Cookie method my Lemon (2002) before. However, as the computational complexity of the cookie directly affects the performance of the whole approach, the proposed method by Bo & Ruimin (2009) uses a 32-bit key Blowfish encryption algorithm as well as introduces random secret value to the algorithm. Because the calculation of the cookie is dependent on not only IP packet information in the appropriate fields, but also on the random secret values, if an attacker can not get a secret value from the system, he can not attack. The algorithm proposed also sets the expiry time for secret values. Once the secret value is timed out, the algorithm will use a new one which further increases the attacking difficulty.

Bo & Ruimin (2009) also claim, that the 8 bits used for time-out certification in the traditional 32-bit cookie field result in Hash value field being too small. To solve the issue, they propose using only 1 bit for the time-our certificate making it possible to utilize 31 bits for cookie value. When generating cookie, the proposed method uses the current secret value to calculate the hash value of IP packet information and then fill the current time number into the highest bit of cookie. In the validation phase, after restoring the original cookie value from the packet sequence confirmation number field, the time number is extracted from the one bit remaining. After doing that, the algorithm finds the secret value based on the time number and calculates 31 bit hash value of the packet information, ultimately comparing the new and the old cookie values for verification.

Some mitigation technologies have been proposed with TCP SYN attacks using IP spoofing in mind. One of them is proposed by Kavisankar & Chellappan (2011). The method uses TCP Probing for Reply Acknowledgement Packet, which crafts/appends TCP acknowledgement messages to provide another layer of protection. In this method, recipient host/server sends acknowledgement which states that the client should change the TCP window size or cause packet retransmission. If the supposed source does not change the window size or does not re-transmit the packet, it can be judged to be spoofed. The mitigation process is pictured below (Figure 6).

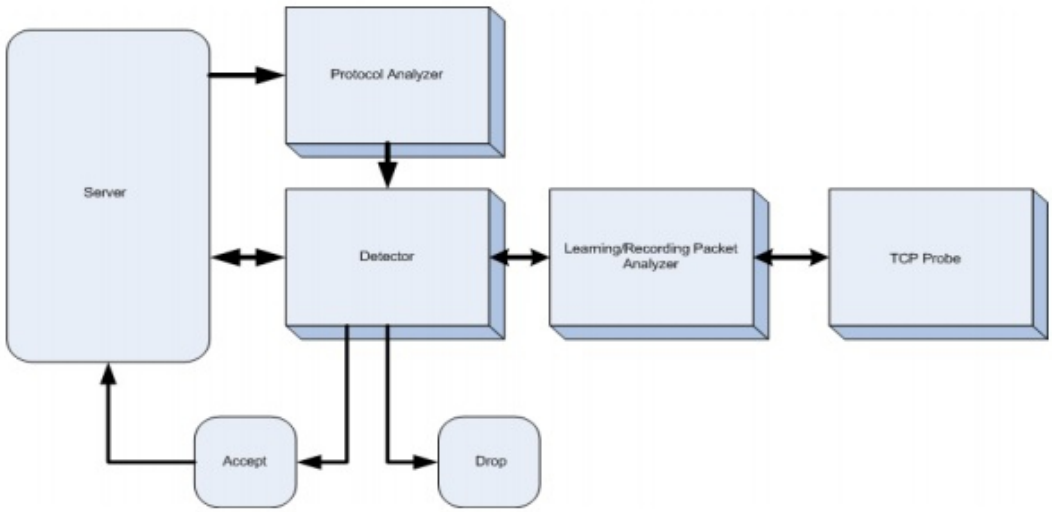

Figure 6: Flow chart (Kavisankar & Chellappan, 2011)

First, the server receives the TCP packet with SYN flag in the packet. Next the protocol analyser detects it utilizing TCP protocol, with the TCP probe sending

the client a request to re-ntransmit or change the TCP Window size. Depending on the reply by the client, the packet is sent to learning/recording packet analyser to be sent forward to detector, which based on the reply from the TCP Probing drops or accepts the packet.

TCP Probing for Reply Acknowledgement is a host-based architecture, which uses several components. TCP probe is used to send the specification to the client trying to connect to the server. The protocol analysers task is to analyse whether the packet is following the TCP protocol. In a later stage it is also used to verify whether the packet satisfies the specification given by the TCP probe. The Learning/Recording Packet Analyser is used to record the transfer of packets used in the handshake as well as verifying the specification given by the TCP Probe along with protocol analyser. Ultimately The Detector decides to drop or accept the packet based on the reply received by the server verifying whether the packet was modified by the client accordingly or not. In the case where IP spoofing is utilized by the attacker, packets will be dropped since the spoofed addresses will not be able to send the proper TCP probe reply. (Kavisankar & Chellappan, 2011).

3.1.2 Router-based defence

Samad et. al. (2014) described and analysed the performance of four different router-based defence methods, which are Reverse Path Forwarding (RPF), TCP Intercept, Access list (ACL) and Rate Limiting Defence.

RPF works in a similar way like part of an anti-spam solution. It takes the source IP address of a packet received from the Internet and looks up to see if the router has a route in its routing table to reply to that packet. Elementarily, if there is no route in the routing table for a response to return to the source IP, then is is likely to be a spoofed packet, resulting in the router dropping the packet. (Microsoft Forefront, 2013).

TCP Intercept, the second of the tested router based defences is a feature on the Cisco firewall. There are two modes for the functionality in question. First is the intercept mode, which as the name suggests, intercepts TCP connections which are incoming to the targets system. The router on receipt of the connection would respond impersonating the server to the client. As per the protocol, only on successful completion of the TCP three way handshake, the server is allowed the actual connection. (Cisco Systems, 2013).

The third of the tested defences is ACL known as IP addresses ingress filtering. It works on a premise that the most commonly spoofed IP addresses

are the private IP addresses and other types of shared/special IP addresses. ACL will block any private IP address from entering the local network because the private IP address should not be allowed to get inside the local network. (Microsoft Forefront, 2013).

The fourth of the tested router based defences, Rate Limiting places a cap or sets up a threshold limit of traffic that server would be able to withstand. The highlight feature of this technique is the functionality which allows the network administrator to decide how much traffic to let inside the network Cisco Router. (Microsoft Forefront, 2013).

Zhang et. al. (2010), have their own approach to the router-based defence They propose a per-IP behaviour analysis approach, which takes form in an online, real-time DDoS attack detection and prevention system. It is deployed at the entrance to the victim subnet, and can be divided into three layers: application layer, network layer and driver layer.

The application layer consists of user-controlled module to turn on and off the real-time detection, system management module to set detection parameters and data upload module to unload data in three buffers. Network layer includes attack feature training module to extract flow features and store them into the corresponding IP record, attack detection module to determine whether the traffic behaviour is abnormal and the data buffer update module for updating the data buffer. Finally, the driver layer consists of two modules of packet capture module and packet filtering module. (Zhang et. al., 2010).

After turning on the system, following the data packet classification algorithm packets are captured and stored in the data buffer. Based on test results, the system filters the attacker's traffic and forwards normal user traffic.

3.1.3 Firewall-based defence

Microsoft Forefront Threat Management Gateway (TMG) proxy server is one type of a firewall-based defence mechanism. TMG has parameters that determine traffic management coming from clients and specific port listening to web requests and handling authentication. TMG proxy also has the functionalities to stop the flood denial of service attack for TCP, UDP and ICMP packets. The above mentioned options control the TCP connection which includes TCP concurrent connections per IP address option, TCP half-open connections option, maximum TCP connect requests per minute per IP address option, HTTP requests per minute per IP address option among others. (Microsoft Forefront, 2013).

3.1.4 Agent-based defence

Anti DDoS Guardian is one example of Agent-based defence software It is a firewall software, which is mainly powered to prevent several DDoS Attacks including UDP, ICMP and TCP attacks. The functionalities of Anti DDoS Guardian include controlling TCP connections per second, maximum number of TCP connections for IP address, half-open connections allowed as well as the maximum number of concurrent client IP addresses. (Anti DDoS, 2015).

3.1.5 Analysis

The three defence mechanisms described above were compared and analysed in 2014 study by Samad et. al. They used three major metrics for comparison purposes. First one was Round Trip Time (RTT) delay standing for a delay between a request and a response time. Second one was CPU utilization of the victim, which can crash the system if it gets too high. Third and the last metric was bandwidth. It should be noticed that only three instances of three out of the four types of defence mechanisms against TCP SYN DDoS attacks were compared and analysed in this study. No instance of Server-based defence as classified by Peng et. al. (2004) was included.

In their comparison, Kolahi et. al. (2014) generated traffic for 5 minutes every attack and RTT, CPU utilization and bandwidth was measured. Over 20 attacks were simulated, after what results averaged and standard deviation measured. Attacks were repeated until standard deviation was 0.05% of the average results.

The study resulted in following outcomes. The average RTT for legitimate users was 1.92ms before the attack and 5252.52ms on average during the simulated TCP SYN attack (Table 1).

Scenario	Average RTT(ms)
Without an attack	1.92 ms
During the attack	5252.52 ms
With TCP Intercept	3.21 ms
With Rate Limiting	3749.68 ms
With Access List	3098.65 ms
With Reverse Path Forwarding	2728.26 ms
Anti DDOS Guardian	2553.71 ms
Forefront TMG Proxy 2010	2803.34 ms

Table 1: Average RTT before attack, during attack, and various defences (Kolahi et. al., 2014)

According to the measurements conducted by Kolahi et. al. (2014), TCP Intercept was the most effective defence, because it eliminated malicious SYN attack traffic spoofed from reaching the server. Rate limiting was analysed to result in the highest RTT because the malicious traffic was not blocked, but only limited up to the threshold. RTT of both anti DDoS and TMG Proxy server were analysed to give high RTT, because they only dropped the malicious traffic, but didn't stop the flooding, while RPFs poor performance was addresses to the fact that only spoofed address traffic was stopped, but not the traffic coming from the attacker with a valid IP address.

Measurements of the CPU utilisation of the victim system as a result of the attacks is presented below (Table 2).

Scenario	Utilization
During TCP SYN Attack	10 %
Without an attack	1 %
With TCP Intercept	1 %
With Rate Limiting	9 %
With Access List	9 %
With Reverse Path Forwarding	7 %
Anti DDOS Guardian	50 %
Forefront TMG Proxy 2010	1 %

Table 2: CPU utilisation before attack, during attack, and various defenses (Kolahi et. al., 2014)

It can be observed from table 5, that during the testing the CPU Utilisation percentage was 1% before the simulated attacks, but jumped up to 10% during TCP SYN flood. In terms of mitigation, TCP Intercept and TMG Proxy were noticed to have similar level of effectiveness, with TCP Intercept completely eliminating the malicious traffic from passing the router to the web server It is worthwhile noticing, that TMG Proxy just absorbed the impact, with proxy server's CPU going up 60% during the attacks.

Both Rate Limiting and Access List mitigation methods were noticed to have a CPU utilisation of 9%, because rate limiting still allows passing the packets and while ACL attempted to eliminate malicious packets some of them managed to pass through nevertheless. RPF was second in effectiveness with 7% while highest CPU utilisation percentage was caused by Anti DDoS guardian, as it was installed on the victim's machine taking 50% of the computing power.

The results of the last measured metric, Kbps representing bandwidth are presented in following table (Table 4).

Scenario	Traffic Rate (Kbps)
During TCP SYN Attack	776.51
Without an attack	1.342
With TCP Intercept	1.462
With Rate Limiting	424.07
With Access List	421.65
With Reverse Path Forwarding	388.26
Anti DDOS Guardian	630.30
Forefront TMG Proxy 2010	610.16

Table 3: Average traffic rate before attack, during attack, and various defences (Kolahi et. al., 2014)

As can be seen from the measurements, the traffic rate bandwidth without the attack was only 1.342 Kbps, significantly increasing to 776.51 Kbps during the TCP SYN Flood attack, which resulted in difficulties for legitimate users to reach the server. TCP Intercept was once again estimated to be the superior defence, with this mitigation method dropping the connection before it enters the network server. While Forefront TMG Proxy 2010 eliminated the malicious traffic, it did not prevent the traffic from getting into the network resulting in rather unimpressive mitigation. The same stands for all other mitigation methods except TCP intercept. They eliminated traffic within the victim's machine, but let some packets pass through.

It was concluded by Kolahi et. al. (2014), that TCP intercept, which was one of the tested router-based defences, mitigating all impact of the attacks almost completely. Rate-limiting defence mechanism also belonging to the router-based category was deemed to be the worst one from the RTT standpoint, mitigating the impact by less than 30% from 5252ms to 3749ms. The loser in CPU utilization was DDoS Guardian installed on the victim's machine increasing the utilisation to 50%. The same method was the worst in bandwidth measurements as well, decreasing the traffic rate by less than 20% from 776 Kbps during the attack to 630kbps.

While server-based defences were not a part of comparison by Kolahi et. al. (2014), the performance analysis for traditional SYN Cookie and a novel SYN Cookie defence proposed by them was done in the study by Bo & Ruimin (2009). In their study, they measured only the impact and mitigation effectiveness on CPU utilisation and average MRT. The CPU occupancy rate is presented below (Figure 7).

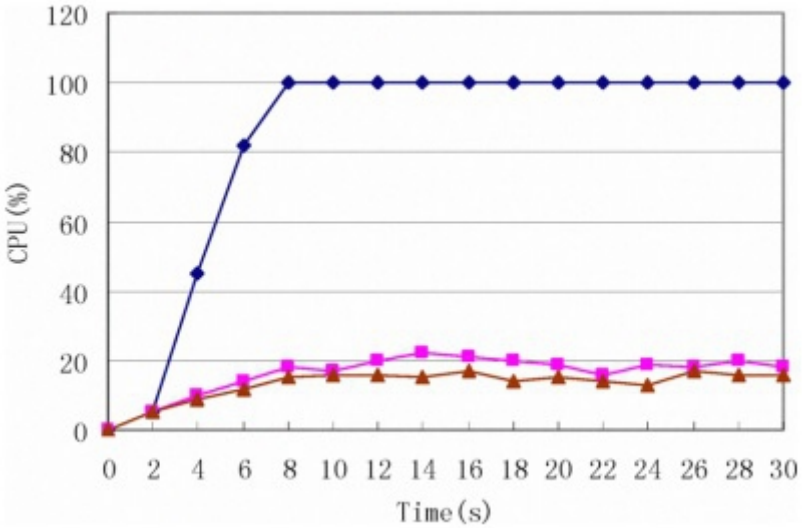

Figure 7: CPU utilisation rate (Bo & Ruimin, 2009)

Blue line indicated the CPU utilisation under attack with no mitigation, purple line presents the performance of traditional SYN Cookie method while red line the performance of the proposed novel SYN Cookie method. It can be observed, that both cookie methods reduce the CPU utilisation by around 80%, proving their effectiveness. A novel SYN Cookie method was also proved to have 30% lower average RTT compared to the traditional SYN Cookie.

In comparison of CPU utilisation rate only, while different setups were used by studies of Bo & Ruimin (2009) and Kolahi et. al. (2014), it can be assumed that SYN Cookie methods of server-based defence type reduce the CPU utilisation during attack by 80%, losing only to a router-based defence TCP intercept and firewall-based defence Forefront TMG Proxy.

3.2 Against DNS Amplification

Rozekrans & de Koning (2014) state, that as DNS amplification DDoS attacks are becoming more sophisticated, packet filters begin to struggle in catching the traffic. This has caused the filtering to be started being implemented on name servers instead.

Mitigation methods described below are for making it more difficult for the attacker to abuse the name servers. They are not designed to be used by the ultimate target of the attacks.

3.2.1 Firewall

Most firewalls can be easily configured to block specific packets or IP addresses. For example, a firewall can be configured to block all ANY requests, which would cause only a little harm, as most environments do not utilize these queries. The main drawback, however is that attacker could switch to other DNS queries, which would make adjusting the firewall accordingly more difficult. (Rozekrans & de Koning, 2014).

3.2.2 Network Ingress Filtering

Network Ingress Filtering is a mechanism, which makes it possible for routers to check the validity of an IP address. The filter can be adjusted to forward all traffic from a certain range of IP addresses and drop the rest. This method can potentially be effective in organization environments, where traffic from outside a certain network shouldn't be allowed in the first place. (Ferguson & Senie, 2000).

3.2.3 DNS Dampening

DNS Dampening is another mitigation method, proposed by Lutz (2012). The gist of this method is generation of so called penalty points per IP or network, the amount of which is determined by the query type and the request size. The penalty points should also have a characteristics of decaying over time.

The challenging point of developing the method seems to have been coming up with efficient penalty table. Ultimately, the efficient solution was found out to be giving 10 penalty points for every unknown client, 100 points for ANY query and 1 points for other queries. If a query would be repeated with the same ID, 100 penalty points per repeat would be applied. The amount of added points would also increase by the size of the response. (Lutz, 2012).

The idea of the method is that the amount of penalty point reaching the configured limit triggers dampening. When IP is in dampened state, server drops all requests from that address. As the penalty drops under a certain level as a result of exponential decaying, the dampening will stop and the server would start processing the requests again.

3.2.4 Response Rate Limiting

Response Rate Limiting (RRL), is a mitigation method proposed by Vixie (2012). Its point is to limit the rate of responses by a DNS server in order to mitigate the DNS reflection and amplification attacks. It is a method with several configurable parameters used in order to increase the effectiveness of the defence

Parameter RESPONSES-PER-SECOND limits the amount of the same responses an requester can get every second. This is deemed to be more effective than limiting the amount of same requests, as different types of requests can generate the same answer ERRORS-PER-SECOND limits the amount of error answers a requester can get every second. LOG-ONLY can make responses to be continued to get logged instead of being dropped. WINDOW determines the time frame for measuring the responses. IPV4-PREFIX-LENGTH determines the size for a container the addresses are being stored into, IPV6-PREFIX-LENGTH is the same parameter for IPV6 addresses. LEAK-RATE makes it possible to set the server to make a legitimate response once per LEAK-RATE queries to the queries which are about to get dropped, in order to give a forged IP addresses victim a chance to answer. TC-RATE works in a similar way, sending a forged IP addresses victim a request to retry using TCP. MAX-TABLE-SIZE sets a maximum number of states maintained within the server. MIN-TABLE-SIZE is the initial size to be allocated for an empty state blob table at startup. (Vixie, 2012).

The method works in the following way. Requester's IP address is taken and a state blob created for it. If the state blob indicates that the response have been sent too often, server sends a leaked response as per LEAK-RATE, truncated response as per TC-RATE or no response at all. (Vixie, 2012).

3.2.5 Analysis

Network ingress filtering was evaluated by Beitollahi et. Deconinck (2012). While they agree that it is easy for ISP to design the approproate ingress and egress filters in order to prevent IP spoofing based not only on the IP addresses, but by other factors such as protocol type, port number, or other criteria, they also list multiple challenges involved in this method.

The method is only effective if it is universally used by vast majority of ISPs in the world. Zombie machines of ISPs not using the filters can still be used to attack the target. Another challenges include admininstrative overhead, performance cost, and lack of motivation for ISPs to use these filters. In theory,

however, the method is good as long as all ISPs would be willing to use it. (Beitollahi et. Deconinck, 2012).

Response rate-limiting was thoroughly tested by Rozenkrans et. De Koning (2014). They claimed that as the repeating ANY attack is the most encountered attack on the internet, they would have to test the effectiveness of RRL against it. It is also stated in their research, that attacks are abusing all kinds of authoritative DNS servers including TLD's, web hosts and organizations' own servers. They speculate that as mitigation methods progress, attackers will find more sophisticated methods such as querying for various domain names and combining it with querying different record types.

For the first measurements a repeating ANY attack was sent to the DNS server hosting a single zone resulting in server returning all the records and signatures it has for the requested domain name. This resulted in high amplification ratio. (Rozenkrans et. De Koning, 2014).

Utilizing their testing setup, Rozenkrans & de Koning (2014) observed that 1000 incoming ANY queries per second for the single zone resulted in 80KB/S inbound and 4MB/S outbound traffic, meaning huge amplification. However, when RRL was enabled with the default settings, the outbound traffic quickly dropped to 39KB/S mitigating the amplification. When same attack and mitigation method was tested against a DNS server hosting a TLD like zone, the results were similar. Output was damped from 4MB/S to 73KB/S, the same value as the input. The method was proved to be effective against repeating query attacks.

Next, Rozenkrans & de Koning (2014) tested how the method works against a second type, varying query attack. Generating a varying query attack involves attacker generating random queries or utilizing information gathered about the domain names hosted on the authoritative name server (using dictionary attack or webcrawler, for example).

Rozenkrans & de Koning (2014) simulated the attack by sending ANY queries for a variety of domain names of the zones. The attacks were executed with an intensity of 1000 ANY queries per second. First attack was simulated to a zone with 0% existing domain names, which, naturally would not result in any amplification for the attacker. RRL, however, got triggered as intended, and dropped the output traffic from already low values.

Second attack was simulated against a zone with 25% existing domain names. The attacks caused a server to sent responses at a speed of 1350 KB/S resulting in an amplification ratio of 17. The amount of outbound traffic decreases as TC-RATE value is increased. RRL was measured to decrease the amplification ratio from 17 to 3.

In the third measurement, with RRL disabled, an attack against 50% existing domain name zone resulted in an amplification value of 14.8. Rozenkrans & de Koning (2014), speculated that the reason was that the NOERROR answer from the DNS server is smaller than an NXDOMAIN answer resulting from a request to an non-existing target. Enabling RRL caused the amplification value to drop to around 6, which is not as impressive as in the 25% case scenario, because 50% of responses were unique, and thus not rate-limited.

Fourth attack against a zone with 75% of the domain names real resulted in output of 1060 KB/S and an amplification rate of 13.4. The reason for the ratio being lower than previous attacks is the same as in scenario with 50% existing domain names: the NOERROR answer from a DNS server is smaller than the NXDOMAIN answer. With RRL enabled, the amplification ratio decreased from 13.4 to 8.7, an unimpressive mitigation resulting from the fact that RRL only limits the NXDOMAIN responses, which are getting fewer as the percentage of real domain names grows.

In the last scenario, where attacker knows all 100% of the domain names for which the server is responsible, all requests are answered with an unique response, making RRL obsolete. The amplification rate of 11.1 is not mitigated, as RRL is not triggered in this attack type. (Rozenkrans & de Koning, 2014).

The effectiveness of RRL against varying query attack is illustrated below (Figure 8).

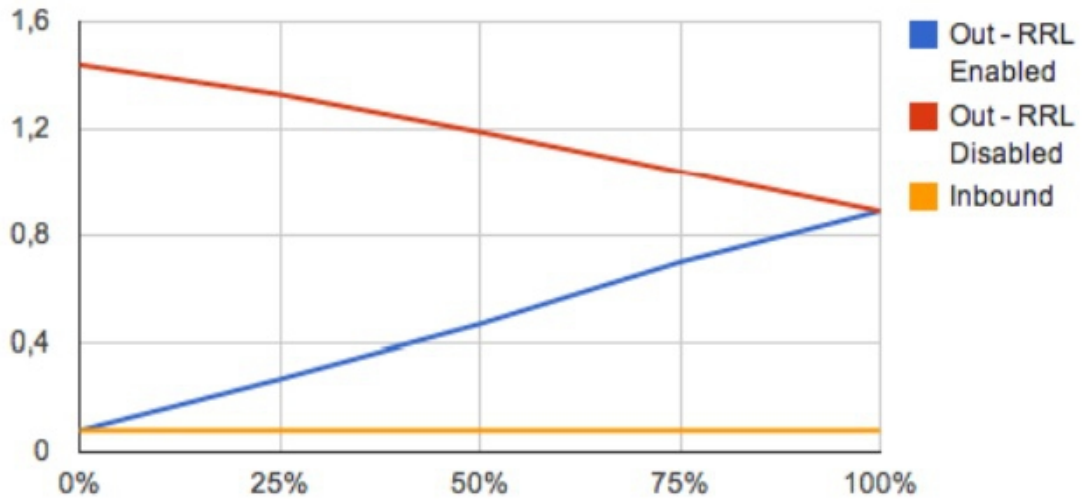

Figure 8: Effectiveness of RRL (Rozenkrans & de Koning, 2014)

It can be observed, that the RRL works best in the situation, where attacker knows from 25 to 50% of the domain-names. As the percentage gets higher, the effectiveness of RRL method disappears, as attacker can hit a higher ratio of

existing domains generating more unique responses, which causes RRL method to distribute the responses to different buckets.

Rozenkrans & de Koning (2014) finish the RRL analysis by speculating the RRL method's effectiveness in the scenario where attacker uses distributed attack using multiple name servers instead of just one. According to them, distributed attack will increase the attack traffic as well as potentially prevent RRL from triggering. That seems probable, as it would mean less repeated requests to the protected server, reducing the effectiveness of RRL while increasing the power of the attack due to the multiple servers being abused.

In their paper, Rozenkrans & de Koning (2014), also present results from experimenting the effectiveness of dampening against distributed DNS Amplification attack as they deem RRL to be ineffective against it. DNS dampening was not covered in such a depth as RRL, but still presented some interesting results (Figure 9).

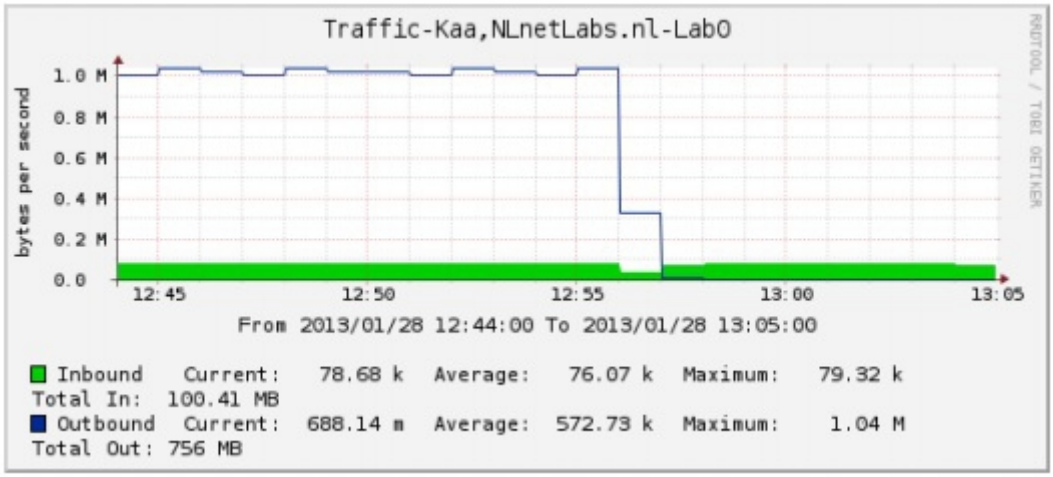

Figure 9: Average inbound and outbound traffic per minute (Rozekrans & de Koning, 2014)

The DNS dampening was activated at 12.56, resulting in instant mitigation. However, Rozenkrans & de Koning (2014), bring up some demerits to the method. They claim, that the biggest drawback is that DNS dampening does not implement a mechanism to counter false positives. As client gets limited, all the traffic, even legitimate is blocked from that client. Another demerit is considered to be the lack of possibility to tailor the method by adjusting DNS dampening parameters.

3.3 Against NTP Amplification

As NTP amplification first occurred in 2014, no papers on the topic could be found at IEEE explore. Everything Google Scholar could provide was a collection of news and reports in addition to one seminar publication related to the topic. While non-academic writings without peer review can not be considered completely trustworthy, it was deemed that these publications could provide relatively reliable information.

Kuhrer et. al. (2014), claim that in the end of 2013 together with some security organizations, they launched a campaign to send advisories to the systems hosting NTP servers with MONLIST enabled. By actively releasing lists of IP addresses, potential amplifiers and advisories to the service providers, during a period of two months, number of amplifiers with MONLIST function decreased by a stunning 92.4% while the number of amplifiers with VERSION functionality dropped by 33.9%. And internet-wide scan was performed on June 20th, with number of hosts vulnerable to MONLIST amplification found out to be 87463 down from 1.6 million half a year ago.

It seems that the most efficient way to mitigate NTP Amplification attacks is simply disabling the MONLIST function, which has little practical use but provides an attacker with huge amplification opportunity. General approach towards NTP Amplification attack mitigation is that the NTP servers should concentrate on preventing of becoming the target of amplification, in contrast to the end victims trying to protect themselves against the amplified attack traffic.

This seems as a right approach, as MONLIST and VERSION functions are more or less obsolete while provide the great amplification potential for the attacker. Disabling these functions would hardly cause any significant problems for the server administrators This is not the case for DNS server administrators, as queries used for DNS Amplification attacks can not be seen obsolete as they have concrete functional uses.

3.4 Against DNS Attacks

Gillman et. al. (2015) propose two main ways to mitigate the DNS attacks. The main one is to deploy many name servers in many locations. It is vital for the DNS system to be able to respond quickly for requests from all corners of the globe, even in situations where it is under attack.

3.4.1 IP Anycast Routing

IP Anycast Routing is the name for the method which makes a service address available to a routing system at Anycast Nodes in multiple discrete locations. The service provided by each respective node is consistent regardless of the node chosen by the routing system to handle any particular request. This method is virtually superior in DDoS mitigation, as a single Anycast Node can act as a sink for attack traffic, preventing nodes in other locations from having the need to deal with that traffic (Abley & Lindqvist, 2006). Abley & Lindqvist (2006), continue by stating that since with Anycasting, the burden of sorting between legitimate and attack traffic is distributed, this method may have potentially better scaling properties compared to a non-distributed service.

When a service is anycast between multiple nodes, the routing system makes the node selection. Because usually every single client-server interaction is carried out between a client and the same server node for whole transaction, the choice of the node has to be stable for the whole transaction. (Abley & Lindqvist, 2006).

Anycast Routing can be implemented within an Inferior Gateway Protocol (IGP) or within the Global Internet (Abley & Lindqvist, 2006). In the case of mitigating DNS attack, an approach to anycast the nodes in the global internet is preferable, because the availability of redundant DNS capacity around the world make it harder for an attacker to overwhelm the name servers even in more isolated parts of the world (Gilman et. al., 2015).

According to recommendations of Abley & Lindqvist (2006), the placement of Anycast Nodes should be decided depending on the goals of the service distribution.

3.4.2 Enhancing DNS resilience with focus on zone popularity and caching

Vasileios et. al. (2007), propose a method of enhancing the DNS resilience against DNS DDoS attacks with a focus on zone popularity and caching. According to them, all the previous attempts to enhance DNS resilience, whether they were successful or not, required noticeable changes done in the DNS infrastructure.

The approach of Vasileios et. al. (2007), introduces changes only to the caching servers without any need to modify the underlying DNS infrastructure. The enhancements' main goal is to force the caching servers (CS) to maintain time copies of the infrastructure resource records (IRRs) for the zones they most

use for a longer time. This will result in a decrease in a number of queries send by a CS to a parent zone for resolving the names belonging to a child zone. This will make a popularity of the zone dependent on number of queries generated for the names of the zone instead of the number of queries generated for the names of the child zone.

Caching servers are servers which store DNS query results for a period of time in order to make it easier to access the addresses. These servers usually implement the recursive algorithm needed to resolve a requested address starting from the DNS root through authoritative name servers of the queried domain. Infrastructure Resource Records are basic data elements in the domain name system, carrying the information about the DNS tree structure.

The following scenario represents this mitigation method in action. In the example, attackers make a successful attack against *com* zone. This results in CS being unable to resolve a zone residing just below the *com* zone, if it does not have IRRs for that zone. The more popular the zone, or the longer TTL the IRR has, the more probable it is that the IRRs for the zone are cached (Vasileios et. al., 2007). If the zone is locally cached, it will be available even if the parent zone will become unavailable.

To increase the probability of having the IRRs, a CS can artificially make the zone more popular by querying it whenever the cached IRRs are about to expire. Another approach is for zone's administrator to increase the TTL value of the zone's IRRs. While it is possible for CS to indefinitely query the IRRs and for administrator to unlimitedly increase the TTL value, it is not recommended as both extreme measures also have their own demerits. Querying the IRRs too much can cause a considerable message overhead while overzealous increasing of TTL value can potentially cause IRRs inconsistencies. (Vasileios et. al., 2007).

The way to establish initial IRR records for zone *A* is following. A CS learns the IRR for a certain zone *A* from its parent zone *P*. *P* sends a referral and authority including the IRRs for *A*. The CS caches the records for the zone *A* and then contacts one of the *A*'s name-servers to obtain the requested data. *A*'s name server's reply include a IRR for *A* in the authority and additional sections of the reply. Then the CS proceeds to replace the cached IRR coming from the parent with the IRR coming from the child zone if they are not identical (Hardie, 1997).

The following queries for names in *A* can utilize IRR data to go directly to *A*'s name-servers. Each such query includes a copy of *A*'s IRR data, which is used by TTL refresh to refresh the TTL on *A*'s IRR. However, many popular DNS caching server implementations do not refresh the value. (Vasileios et. al., 2007).

TTL Refresh is one of the modifications which can enhance the DNS resilience. If TTL Refresh is enabled, every query targeted to *A*'s name servers will reset the TTL and *A*'s TTL will be always locally cached. In implementations not using TTL Refresh, the CS will have to visit the parent zone when the IRR expires.

Second modification proposed in the paper by Vasileios et. al. (2007) is called TTL Renewal. This modification makes it possible for IRRs for the most popular zones to stay in the CS for longer time. The gist of the method is re-fetching and renewing the TTL of the IRRs just as they are about to expire. There are four different policies to apply presented in the study.

LRU[c] is a policy which sets a zone a credit equal to *C*. Every time the IRRs are about to expire the credit is decreased by one and IRRs are re-fetched. This makes the IRRs to stay in the cache for *C**TTL. *LFU[c]* is a policy, which increases a zone's credit equal to C every time the zone is queried. A credit cap of M can also be implemented in this policy. Third policy *A-LRU[c]* is a modification of *LRU[c]* which is introduced with a goal to make credits depend on the TTL value. *A-LFU[c]* is similarly an adaptive version of *LFU[c].* (Vasileios et. al., 2007).

Third modification is Long TTL. Vasileios et. al. (2007), state that simply increasing the TTL value of the IRRs will lower the frequency at which renewing the IRRs is ideal. The main benefit of this approach is that increasing TTL values does not require any modifications to the caching servers and it can be simply implemented by the zone administrators Other merits are reduction of overall DNS traffic and improvement of DNS query response time, since the need for hierarchical DNS tree walking is removed.

3.4.3 Analysis

Vasileios et. al. (2007), tested the effectiveness of their DNS Enhancing approach using the following experiment. They set up multiple different DNS servers implementing different combinations of the modifications described in the previous chapter. For the first six days of the experiment they assumed that all the zones work normally and launched a DDoS attacks with duration from 3 to 24 hours. Finally they measured the percentage of queries failing to resolve during the attack. Both failed queries sent by the Stub Resolvers (SRs) to the CS and from CSs to the ANs are measured. Stub Resolvers, are simple resolvers which rely on a recursive name server to perform the work of finding the needed information.

The following setups were tested. A vanilla system representing default DNS, a system implementing TTL-Refresh, a system implementing TTL-Refresh and RRL-renew, a system that implements TTL-refresh and long-TTL and a system that implements TTL-refresh, RRL-renew and long-TTL. (Figure 10).

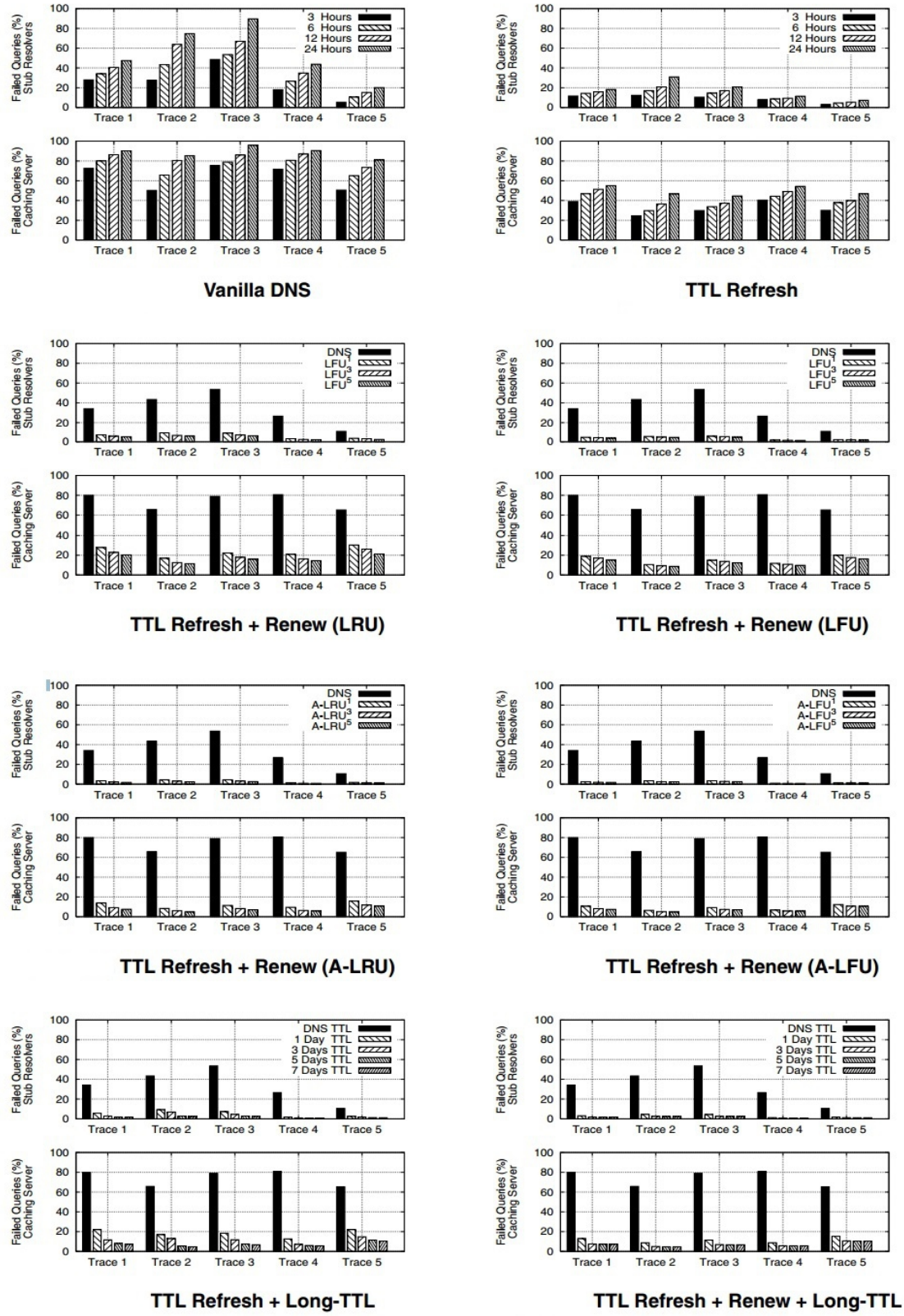

Figure 10: Performance comparison (Vasileios et. al., 2007)

It can be observed that all the tested implementations of DNS resilience enhancing performed better than the vanilla DNS server in DDoS attack mitigation. The figures show the percentage of queries that fail to resolve during the time of DDoS attack. The upper graph of every respective setup shows the percentage of failed queries sent by the SRs, while the lower graph shows the relative percentage of failed queries that are sent by the CSs. Naturally, as the attack duration increases, so does the percentage of failed queries as more and more records start to expire.

TTL Refresh displays at least 50% mitigation, with modification leading to respective percentage drop compared to a vanilla setup, TTL Refresh and Renewal with four different settings. All of them, still, perform almost with the same effectiveness. The adaptive policies are better because of their neutrality towards the different values of the IRRs and TTLs. LFU policies perform better than the LRU because they favour the most frequently used zones. Testing of the TTL Refresh and Long-TTL resulted in long-TTL scheme achieving the same resilience as the most effective IRRs renewal policy. The most effective mitigation method, however, was observed to be a combination of all three techniques: TTL Refresh, TTL Renewal and Long-TTL. TTL value of three days was discovered to be enough to achieve maximum possible resilience, with longer TTL values not affecting the performance of mitigation any more positively. (Vasileios et. al., 2007).

Vasileios et. al., (2007), also examined an overhead of these mitigation methods caused on message (the number of additional queries generated by a CS) and on memory resulting from additional zones cached at the CSs. They discovered that in the worst case the adaptive schemes cause a significant overhead of 500%. Once again, the combination of three techniques: TTL Refresh, TTL Renewal and Long-TTL was deemed to be superior with the smallest message overhead. All of the modifications increased the memory overhead as well, as they all require the caching of IRRs for the longer periods of time. It was concluded that additional memory overhead would not be an issue to the systems of that time, as the caching schemes would only increase the number of cached objects by two to three times. Given the progresses in technology during the years since research, the increase in memory overhead is negligible.

4 PROTECTING WEB SERVER AGAINST DDOS ATTACKS USING FIREWALL

Out of the four attack types studied in second and third chapters of this thesis, DNS DDoS is not going to be tackled with in this chapter as it targets DNS servers and not web servers without own DNS capabilities. As both DNS Amplification and NTP Amplification DDoS attacks are reflection and amplification-based bandwidth exhaustion attacks, and their traffic is similarily transferred via UDP and is similar in nature, results for testing mitigation of the DNS Amplification attack at the web server firewall are going to stand both for NTP Amplification and DNS Amplification. TCP Flood DDoS uses a TCP instead, which is why a different approach to mitigation accompanied by a different set of experiments is needed. Only firewall-based mitigation, as per Peng et. al. (2004) classification is going to be tested. And while Kolahi et. al. Mentioned Microsoft Forefront Threat Management Gateway (TMG) as one solution, a similar approach suitable for Linux based web servers will be under examination in this paper as TMG is not suitable for Linux based systems.

In the experiment, we assume the ultimate target of the DDoS attack to be a simple web server setup hosting a web page or an information system. A common variation for this kind of setup called LAMP stack which stands for Linux, Apache, MySQL and PHP hosted on a single server is used. This kind of setup is likely to be used by small or medium sized businesses hosting a web store or a web page on a web server as a part of their information system. Another likely user is an individual or organization, who publishes information on his own web page hosted on his own server which for one reason or another becomes a target for take down using DDoS attack. As one definition for Information System is ”system with ultimate purpose of storing and managing information” (Land, 2004), businesses hosting a web server and organizations

publishing information on their web page can be considered an information system as well.

As noticed in the first analysis of DDoS attack type occurrences in the year 2014, amplification attacks are the biggest growing trend in the field. Previous researches related to amplification DDoS attack mitigation by Kuhrer et. al. (2014), Rozekrans et. De Koning (2014), Lutz (2012), Vixie (2012) and Ferguson & Senie (2000) provided input related to protecting DNS servers themselves in order to prevent their abuse for DDoS amplification attacks. However, none of the researches published any papers related to protecting web servers against the attacks.

Rozenkrans et. de Koning (2014) stated, DNS Amplification attacks are abusing all kinds of authoritative DNS servers including TLD's, web hosts and organizations' own servers. As there are numerous businesses employing their own DNS servers as a part of a bigger information system, it is important to further study the ways to mitigate DNS Amplification DDoS attacks capable of overloading the final end target organization's server.

While Rozenkrans et. de Koning (2014) analysed numerous different mitigation methods against this type of attack to be used to protect the DNS servers used for amplification and even tested the performance of Response Rate Limiting (RRL) and claimed it's superiority compared to other mitigation methods such as DNS Dampening proposed by Lutz (2012), they provided only a little input in a form of pure speculation in regard of Firewall effectiveness as a mitigation method for the ultimate target of the attack.

As there were no other scientific publications related to mitigating DNS Amplification DDoS attacks on a web server using a firewall, it was chosen as one subject for closer examination and empirical testing for this thesis. The issue is important, as especially starting smaller scale e-commerce web businesses utilizing information system which include a separate simple web server can suffer significant amounts of economic and reputational damage in the scenario where their web server would stop working adequately and even become unavailable as a result of DDoS attack.

The goal of DNS Amplification DDoS attacks against web servers is to completely fill up the bandwidth, resulting in legitimate users becoming unable to access the service.

It should be noticed, that NTP Amplification attacks work on the same principle. While the amplification method is different, both DNS Amplification and NTP Amplification use connectionless UDP protocol to overwhelm a target with amplified traffic, which is redirected to target's servers using a spoofed IP address. This being the case, the solutions and reasoning of this chapter apply

to mitigating both DNS Amplification as well as NTP Amplification, while DNS Amplification is examined closer.

Entities such as big e-commerce and media companies relying on Content Delivery Networks (CDN) to deliver the content to their end users can find GEO IP based load balancing a saving grace against DNS Amplification attacks. CDNs usually place multiple copies of the same server at data centres in different geographical locations with the goal of quick and accurate content delivery to any given client at any location around the world (Lin et. Al, 2012).

Global Load Balance (GLB) systems utilized in those CDNs are used for selecting the best server for the client taking response time and availability in consideration (Lin et. Al, 2012). In general, using web pages hosted by CDNs producing GEO IP based load balancing results in DNS resolving the server closest to the user. In the case of DDoS attack, this technology helps to distribute the traffic load potentially mitigating the attack as well as offers a possibility to isolate service outages to a specific region.

Another good way is filtering amplified attack traffic in the form of UDP packets by filtering them at the edge router. Unfortunately, only Internet Service Providers (ISP) have this possibility, and the ultimate target, in our case web server have to operate under the assumption that traffic is not being filtered unless a relevant service is being purchased and used.

While according to US-CERT (2014) the massive traffic volume potentially generated by DNS Amplification attacks is difficult to mitigate from a target's side, the firewall solution on the web server is going to be tested in this experiment. The hypothesis is that there are some ways to adjust firewall in order for it to be able to mitigate the impact of DNS Amplification attack on the server. However, those methods are speculated to affect normal traffic as well. Finding out the best settings maximizing the mitigation and minimizing the impact on legitimate traffic for the firewall are the ultimate goal of this experiment.

4.1 Research method

In the experiment, two virtual machines are being used. One is running Kali Linux 32-bit for the attacker running on 1024mb of random-access memory (RAM), another is running Ubuntu 32-bit Linux system configured as a web server running on 2048mb of RAM. Wireshark is used on the web servers computer for measuring web traffic and impact of the attack.

Bandwidth traffic between the virtual machines was limited to 100mb/s by using VBoxManage. The following commands were used to create a bandwidth group called ”Limit”, set the limit to 100mb/s and adds the adapter used by both virtual machines *intnet* to the group.

VboxManage bandwidthctl ”Kali-linux” add Limit -100m

VboxManage modifyvm ”webserv” --intnet1 Limit

VboxManage modifyvm ”Kali-linux” --intnet1 Limit

In the experiment, we use a web server which does not have DNS resolving capabilities. It means that our setup does not accept and resolve DNS requests, it simply uses DNS functionalities through port 53. That being said, all the mitigation methods and technologies being proposed in the chapter 3.2 do not apply to our scenario as webserver is the end target for amplified traffic. While it is clear that the lack of abusable DNS resolvers would put an end to DNS Amplification attacks, from the end target viewpoint a different approach has to be taken to mitigation.

A new application called *tsunami* (Infosec Ninjas, 2015) is used by an attacker in order to simulate the DNS Amplification DDoS attack. Tsunami is an open source application by Samiux (GPLv3) forked from Namescan, which is a massive port scanner which can be used for finding open relays. The application utilizes a list of open recursive DNS resolvers, which are abused for amplification purposes and the traffic redirected to the target. While it is mentioned by the developers that the performance of the Tsunami is poor and should not be used for real attacking, it is deemed to be sufficient for testing the web server defences

As for the DNS Amplification DDoS attack, first, the performance of web server without a firewall is going to be measured before and during attack. In the following experiments, different parameters are going to be changed in order to modify the firewall to better mitigate the attack. The effectiveness of this solution is going to be tested and analysed

The similar approach is going to be used when mitigating TCP SYN DDoS attack. Originally vanilla firewall is going to be tested, after what settings are going to be changed and adjusted in attempt to mitigate the attack. For TCP SYN DDoS simulation, application called Hping is going to be used. It is a free packet generator and analyser for TCP/IP protocol used for security auditing and testing of firewalls and networks.

The firewall used in both experiments is Uncomplicated Firewall (UFW). UFW is a front-end for iptables, which is an program on Linux systems which allows a system administrator to configure the tables provided by the Linux

Kernel firewall. Some default functionalities of UFW include allowing and denying incoming and outgoing traffic completely, managing and configuring ports to be used for different connections as well as allow or block connections from specific IP addresses. Technical intricacies of both DNS Amplification and TCP SYN DDoS attacks are going to be analysed in order to come up with solutions to mitigating both of the attacks.

Custom rules can be made for the UFW to follow. Such values as connections per IP, connections per Class C as well as packets per IP can be adjusted. In the case of mitigating DNS Amplification attacks, packets per IP was deemed to be the rule to introduce, most of the traffic is resulted from a single or multiple recursive open DNS servers, which generate a multitude of packets to flood the target as requested by the attacker.

4.2 Measurements

First the DNS Amplification attack mitigation was tested. According to the measurements of the common network traffic under normal circumstances on web server done using Wireshark, the normal flow of inbound traffic was about 10 packages a second. Using Tsunami for 52 seconds during the first experiment on the vanilla web server without any firewall resulted in a inbound traffic equalling to over 9000 packets per second or around 730000 b/s (0.73mb/s) in traffic volume. With outbound traffic being 130000b/s, an amplification factor of 730000/130000 = 5.6 was attained. The command used was:

./tsunami -s 192.168.1.3 -f recursive_dns.txt

In this test run, a text file including addresses of thousands of open recursive DNS servers was used as an parameter in order to provide the Tsunami with a staggering amount of servers to use for amplification. This makes the attack type distributed, as queries are distributed over zones and different servers. While the option of adding a custom rule to UFW for limiting the number of packets per IP address seemed appealing at first, the realization of the sheer number of different servers used in the attack made this option obsolete.

As the attack was executed using only one node, it can not be considered a DDoS attack, but a DoS attack instead. As the defined bandwidth for the web server was 100mb/s, this type of attack was unsufficient to have any kind of negative impact on web server's performance. Kotenko et. Ulanov (2006), developed a tool for simulating various kinds of DDoS attacks called DdoSSim,

which seemed to have potential for modeling DNS Amplification DDoS attack. However, as the tool is not publicly available, the DDoS simulation could not be done within the frame of this thesis.

Nevertheless, by conducting some simple calculations it can be speculated that a botnet employing even as little as 150 infected machines with the same bandwidth as attacker in the experiment can be enough to fill the bandwidth leading to the webserver by sending an amplified amount of traffic equaling to 0.73mb/s x 150 = 109.5mb/s. It should be noticed, that this number would be a result of an amplification using *tsunami* testing tool having a relatively poor amplification factor of 5.6. With a real DNS amplification reaching amplification factor rated up to 50 (Prince, 2012), a combined output traffic of a little over 2mb by a botnet would theoretically be sufficient to cripple the webserver. And with some DDoS attacks reaching volume of over 20GB/s already in 2012 (Prince, 2012), any kind of a simple web server solution could be swiped by a large scale DNS amplification attack.

From capturing the amplified attack traffic, it was noticed that DNS servers used port 53 for the traffic. As web server is not supposed to be using DNS servers for resolving any kind of addresses in our scenario, a special rule was decided to be added to UFW, which would simply deny incoming UDP packets on port 53, which is the port for incoming packets from DNS servers. This was done by a simple terminal command:

sudo ufw deny 53/udp

The command resulted in all the traffic from the DNS servers being denied. However, measurements from wireshark on the web server detected the same amount of traffic as without any mitigation.

While introducing the rule serves its role as traffic filter, it also prevents web server from getting any responses for its legitimate requests to the DNS server. This problem can be overlooked, as web servers main role is serving the files that form the web pages to the users using HTTP client. The only time DNS resolving might be needed is server maintenance and updating, in which case the rule introduced above can be temporarily switched off.

Second, the TCP SYN Flood DoS mitigation was tested. Hping3 was used to continuously send TCP SYN packets to the the web server using the following command.

sudo hping3 -i u1 -S -p 80 192.168.1.3

The parameters have the following meaning. *-i* stands for interval, which means the amount of time to wait before sending next packet. The value *u1* stands for one microsecond. *-S* command sets the SYN TCP flag while *-p*

command defines the destination port, which in our case is the standard 80. The IP address in the end is the address of the target.

As expected, the attack caused the web server to become unresponsive to normal traffic. The web site could not be accessed by another computer during the attack. This proved the efficiency of TCP SYN Flood as an resource exhaustion type DoS attack. As a mitigation technique, UFW was adjusted with a set of rules in order to mitigate the TCP SYN Flood. There were several rules which seemed relevant to mitigating TCP SYN DoS attacks.

One rule which could be introduced was *Connections per IP.* As a typical browser normally uses only several connections per page load lasting several seconds, any more than that can be considered suspicious. That being the case, a new rule was introduced, which would block connections if the number would be over 20 connections / 10 seconds / IP. A rule was introduced with the following script.

-A ufw-http -m state --state NEW -m recent --name conn_per_ip --set

-A ufw-http -m state --state NEW -m recent --name conn_per_ip --update --seconds 10 --hitcount 20 -j ufw-http-logdrop

The TCP SYN Flood simulation was ran once again with the new rule active. Measurements showed the connections from the attacker refused after 20 as intended. The web site was available during the attack, and the attack effectively mitigated as the number of half-open connections initiated by the attacker was not enough to overwhelm the server.

While this rule might prove effective against TCP SYN DoS, in the case of distributed denial of service attack utilizing multiple machines in a botnet, the rule loses it effectiveness. The best option would be a rule which would set a limit for half-open TCP connections, effectively preventing further connections from multiple attacking addresses.

The similar approach was already proposed by Lemon (2002) in his SYN Cookies technique, which can readily be activated in Linux-based web server systems. This method makes it unnecessary for the server to drop legitimate connections after SYN queue fills up with requests. In the handshake, the appropriate SYN+ACK is sent back to the sender, while the SYN queue entry is discarded. If the request is legit, the server is able to reconstruct the handshake queue with the encoded TCP sequence number, which web servers sends as a value in the SYN+ACK response.

There are, however, a few drawbacks to the approach. First one is that only 8 unique MSS values can be stored on the server, which limits the number of sequence numbers that can be stored and utilized in the reconstruction process. This drawback, however, was deemed not to affect our scenario

considerably, as the total number of legitimate requests for our small-scale web server is assumed not to bee too high.

Another drawback of SYN cookies is that using TCP options becomes impossible due to discarding the SYN queue entries by the servers applying SYN Cookies method along with the TCP option information. Main TCP options available are maximum segment size, window scale, selective acknowledgement and time stamp. Under attack the loss of these options, however, was deemed to be a reasonable trade for keeping a working connection. Thus, the SYN cookies was the second experimented method. It was introduced by editing the *sysctl.conf* file with the following lines:

sudo nano /etc/sysctl.conf

net.ipv4.tcp_syncookies = 1

sysctl -p

In the following measurements, legitimate users could access the web server with the SYN cookies storing their TCP sequence numbers and reconstructing handshakes as intended. The load on the web server lessened considerably with the SYN Cookies applied.

4.3 Conclusions

A rule added to the UFW denying UDP traffic on port 53 was found out, and measured to be an effective tool for dropping all amplified attack traffic which uses UDP protocol on port 53. While Tsunami could naturally amplify the traffic abusing open recursive DNS servers as normal, the amplified traffic did not affect web server as it had a rule for the port 53 in place.

There still might be some cases in which UDP traffic is on port 53 is relevant for the functionality of web server. The obvious example would be the web server running its own DNS server. Another one could be a script run on the web server, which requires it to connect to another machine in order to get some data. A command called */sbin/services list -all* can be used for finding out if any application on the web servers uses the port 53 for its own purposes. In our case of a simple web server, however, no significant downsides were detected in closing the port 53 from incoming UDP traffic, making denying incoming UDP traffic on the port a viable solution for denying the traffic at the web server firewall level.

Dropping the attack traffic at a web server firewall is still too little too late. DNS Amplification DDoS attack is a volumetric bandwidth depletion attack, with a sole purpose of causing congestion between the target and the rest of the

internet. The nature of high volume DNS amplification attack, the amplified attack traffic exhausts the bandwidth leading up to the server, making the content hosted on the web server inaccessible during the attack. From a standpoint of a web server administrator, the solution to the DNS Amplification attack mitigation is further up the stream.

This goes along with the findings by Peng et. al. (2007). They defined three different types of attack mitigation: bottleneck resource management, intermediate network reaction and source end reaction. Closing port 53, which is a kind of a bottleneck resource management method is not an efficient way to mitigate DNS amplification attack. While the host resources may be effectively managed, network resources are likely to become a bottleneck during DDoS attack (Peng et. Al, 2007). That being the case, intermediate network reaction and source end reaction, filtering the attack traffic close to attack source are the only reasonable mitigation methods.

Plausible solutions being described in the beginning of this chapter are using a Content Delivery Network's (CDN) services, which, while unable to nullify the impact of DNS Amplification attack, mitigate it effectively just by having more bandwidth than the attack is capable of filling as well as utilizing Global Load Balance (GLP) systems in order to soften the impact of the attack by spreading it to multiple locations. Another solution is rejecting any DNS traffic with spoofed addresses long before it reaches web server firewall, which is unfortunately not within web server administrators power.

SYN Cookies was deemed to be a good solution to mitigating TCP SYN DoS attacks on web server. Implementing a *Connections per IP* rule on the UFW, while effective against a single DoS, was estimated to be ineffective against distributed attacks. Even with the drawbacks SYN Cookies method has, it still seems to be the the most efficient server-firewall based mitigation approach requiring only a little increase in computational power during implementation.

4.4 Future work

In the case of bigger and more complex solutions utilizing multiple web servers such as ones by Microsoft or Facebook, simply diluting the impact of DDoS amplification attacks on multiple data centres might prove to be effective. Such DNS services as Cloudfare (Prince, 2012), are also one option for individuals, by offering a possibility of dilluting the attack impact by spreading the traffic globally.

Amplification attack mitigation research have been concentrated around DNS servers and resolvers themselves, meaning that there is clearly not enough

studies published from the perspective of the web servers being the ultimate end target of those attacks. And while a perfect world, where all DNS server administrators follow the best practices and make their servers less susceptible to amplification abuse might sound ideal, it is not the reality. As long as there are DNS and NTP servers vulnerable to amplification attacks, a sufferer of those attacks have to be ready to deal with whatever amplified attack traffic might be coming their way.

In the case of TCP, an improved version of SYN Cookies is preferable to be developed, with less drawbacks than the current approach and compatibility with other TCP extensions. While there is a newer version called TCP SYN Cookie Transactions (TCPCT), which is an extension of TCP on its own, it requires TCPCT support from both sides of the handshake and has a performance cost resulting it to never gaining friction.

5 SUMMARY

The goal of this research was finding out what are the types of DDoS attack types being popular in the past year and how do they work. The research question was ***"How do contemporary widely used DDoS attacks work and how to efficiently mitigate them?".***

First it had to be defined what are contemporary widely used DDoS attacks. Considering the rapid changes in the field, attacks occurred in 2014 and 2015 were taken in consideration.

Some noticeable differences were observed when comparing the data provided by a different reports of different organizations. A consensus on the attack type popularity was attempted to be done in order to define what are *contemporary widely used DDoS attacks*. The most widespread types of DDoS attacks seemed to be TCP SYN, DNS Amplification, NTP Amplification, UDP Flood and DNS attacks consisting majority of all DDoS attack instances in the past year. DNS attacks, NTP Amplification attacks, DNS amplification attacks, large and normal SYN attacks as well as UDP Flood attacks occur in the transmission (fourth) layer of OSI model. Out of these attacks, the ones referred in most of the reports from the last year were taken under closer inspection. These attacks were SYN Flood, DNS amplification, NTP amplification and DNS attacks.

Abusing certain commands and functionalities of different protocols in order to amplify the power of the DDoS attack was discovered to be a common theme in some of the attack types which have been popular in the past year. Especially amplifying can be seen as a recent trend, with little to no occurrences in 2013. That is the reason for including DNS amplification and NTP amplification in the study. However, SYN attacks being conducted in the transmission layer still seemed to consist the major share of DDoS occurrences

After choosing the attack types for closer inspection it had to be found out how do the attacks work. Literature overview resulted in the following findings.

TCP SYN Flood attack type is a type of DDoS in which attacker uses TCP protocol to send multiple SYN requests to a target's system with a purpose of consuming enough server resources to make the system unresponsive to legitimate requests.

DNS Amplification attack type is a popular DDoS attack type, where attacker uses public open DNS servers to flood a target system with DNS response traffic. The gist of the attack is an attacker sending a DNS name lookup request to an open DNS server using the spoofed address of a victim as a source address. As the DNS server sends DNS record response, which is many times bigger than the size of the request, the target system gets overwhelmed with traffic. Most commonly used request by an attacker is ”ANY”, which returns all known information about a DNS zone in a single request. By using multiple computers combined into a botnet, an attacker is able to generate a large amount of traffic.

NTP Amplification attack works in a similar way as the DNS Amplification attack, except that the attacker abuses NTP servers with a feature called MONLIST, which returns of the last 600 hosts connected to the server. Similarly to the DNS Amplification, a small query generates a large answer, which when redirected to the target's spoofed address can cause target's servers to become unresponsive to legitimate traffic.

DNS Attack does not directly target a certain system, but the DNS server instead. It is conducted by attacker sending many DNS requests to the name server with the purpose of overloading it. As when faced with too large amount of traffic, DNS will start dropping requests indiscriminately, resulting in potential inaccessibility for legitimate users requesting the server to resolve an address they are trying to reach.

The most efficient mitigation method against TCP SYN attacks was found out to be TCP intercept in a research by Kolahi et. al (2014), which managed to mitigate the attacks almost completely by eliminating malicious SYN attack traffic spoofed from reaching the server, while using up only a little computing power. Also even under TCP SYN attack, CPU utilization and bandwidth were at the normal levels proving the superiority of TCP intercept. All the other methods tested in the research had major flaws such as inability to adequately reduce the impact of the attacks on traffic rate or RTT. The only other method which can be seen as promising was SYN Cookies, tested in a paper by Bo & Ruimin (2009). And while the mitigation prowess of the SYN Cookies on CPU utilization and RTT meters seemed promising, the research did not measure the impact on bandwidth, making the adequacy of this technology not unambiguous.

In a research by Rozenkrands et. De Koning (2014), Response Rate Limiting was proven to be a great tool for protection the DNS servers against the most common DNS Amplification attacks. However, the technology was

struggling against varying query attacks, with performance depending on the attacker's knowledge about the domain names. And while DNS Dampening also resulted in adequate mitigation, the inability to handle false positives i.e. respond to the legitimate traffic while under attack was speculated to be its significant weakness.

The most significant input as to dealing with NTP Amplification attack, one of the recent trends from 2014 was provided by Kuhrer et. al. (2014). A simple campaign targeted at the administrators of NTP servers spreading the awareness about the NTP Amplification issue and urging to disable the more or less obsolete MONLIST and VERSION function in order to reduce the chance of a server itself becoming a target of amplification was proposed as the way of dealing with the problem.

Out of the four DDoS attack types described in the first main chapter of this thesis, mitigation of attacks using methods which can target simple web server without DNS resolving capabilities was studied. Those attack types were Amplification and TCP SYN DDoS.

In the first empirical part of the thesis, amplification attack mitigation on the web server was examined closer. A solution for a simple web server not providing any DNS services was found out to be blocking UDP traffic on port 53, which was found out to be the port for all incoming traffic resulting from DNS Amplification attack. As an ordinary web server is not supposed to be using DNS resolving except for updating and maintenance purposes, the special rule for port 53 was deemed to be a reasonable option to introduce. However, the DNS Amplification attack traffic causes the pipelines leading to the web server to get congested, resulting in loss of server availability for legitimate users. Thus, blocking UDP traffic from port 53 can not be seen as any kind of mitigation solution, as ultimately it just drops the traffic at the web server gate, which is irrelevant as the bandwidth gets exhausted with attack traffic in any case. Possible solutions such as having ISP drop attack traffic at the gateway and the usage of CDN services were considered, albeit deemed unreasonable for small-scale web server hosts.

In the second empirical part of the thesis, TCP SYN flood mitigation on the web server was researched. A solution of introducing a custom rule to the UFW limiting the number of connections from a single IP per time frame was hypothesized, and measured to be an effective mitigation approach against TCP SYN Flood DoS coming from a single initiator and a single IP. However, the solution was seen to be ineffective in the case of distributed denial of service attack utilizing botnet. Another reasonable solution, proposed by Lemon (2002), called SYN Cookies was implemented and tested. The technology in question discards the SYN queue entries after sending a ACK+SYN response to the originator. This eliminated the ultimate problem caused by TCP SYN DDoS, while costing a loss of TCP Options and an increase in computational power

required in order to solve the secret function in order to reconstruct the TCP handshake after receiving an ACK from the initiator.

While having significant differences, in essence all transport layer DDoS attacks are techniques used to damage the availability of a web service by sending a huge amounts of web traffic. And while the intuitive solution would be having more bandwidth or resources than the attacker, it is not always realistic with most of the botnets overwhelming the target with a power of hundreds or thousands of compromised machines. This is why a deep understanding of technologies being targeted by attacks is vital in order to come up with counter measures and mitigation techniques against the DDoS attacks. Ultimately, it should be remembered that there are numerous ways to deal with DDoS attacks. Whether it is demotivating the attackers, destroying botnets, having a sufficient amount of network resources, relying on specific technology or preventing the systems and servers from being a part of the attack, optimal methods have to be defined depending on the nature of DDoS attack type to be dealt with.

SOURCES

Abley, J. & Lindqvist, K. (2006). Operation of Anycast Services. *IETF RFC 4786*

Akamai (2015). State of the Internet Report Q4 2014

Arbor Networks. (2015). Worldwide Infrastructure Security Report, Volume X.

Banday, M. T., Qadry, J. A., Shah, N. A. (2009). Study of Botnets and Their Threats to Internet Security. *All Sprouts Content. Paper 279.*

BeeThink Software (2015). Anti DDoS Guardian

Beitollahi, H., Deconinck, G. (2012). Analyzing wellknown countermeasures against distributed denial of service attacks *Computer Communications, vol. 35, no. 11, pp. 1312–1332, Jun. 2012.*

Bo, H. & Ruimin, H. (2009). A Novel SYN Cookie Method for TCP Layer DDoS Attack. *2009 International Conference on Future BioMedical Information Engineering*

CDNetworks Security Service Team. (2015). 2014 DDoS Attack Trends and Outlook for 2015

Cisco Systems (2013). TCP Intercept Commands

Criscuolo, P. J. (2000). Distributed Denial of Service, Tribe Flood Network 2000, and Stacheldraht CIAC-2319, Department of Energy Computer Incident Advisory Capability (CIAC), UCRL-ID-136939, Rev. 1., Lawrence Livermore National Laboratory, February 14, 2000.

Dubojs, É., Heymans, P., Mayer, N., Matulevičius, R. (2010). A Systematic Approach to Define the Domain of Information System Security Risk Management. *Intentional Perspectives on Information Systems Engineering, 2010, 289-306.*

Eddy, W. M. (2006). Defenses Against TCP SYN Flooding Attacks. *The Internet Protocol Journal, 9.*

Fanglu, G., Jiawu, C., Tzi-cker, C. (2006) Spoof Detection for Preventing DoS Attacks against DNS Servers. *Computer Science Department. Stony Brook University, NY 11794.*

Ferguson, P. & Senie, D. (2000) Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. *Network Working Group*

Gilman, D., Lin, Y., Maggs, B., Sitaraman, R. K. (2015). Protecting Websites from Attack with Secure Delivery Networks *IEEE Computer, volume 48.*

Guardian (2010). Operation Payback cripples MasterCard site in revenge for WikiLeaks ban, *Dec. 8, 2010,* http://www.guardian.co.uk/media/2010/dec/08/operation–payback –mastercard–website–wikileaks

Greenemeier, L. (2007). Estonian Attacks Raise Concern Over Cyber "Nuclear Winter", Information Week, May 24, 2007. http://www.informationweek.com/estonian-attacks-raise-concern-over-cyber-nuclear-winter/d/d-id/1055474?

Hardie, T. (1997). Clarifications to the DNS Specification. RFC 2181.

Imperva. (2015). The Top DDoS Attack Trends

ISO/EIC (1994). Standard 7498-1:1994. Information Technology – Open Systems Interconnection – Basic Reference Model: The Basic Model

Kavisankar, L., Chellappan, C. (2011) A Mitigation model for TCP SYN flooding with IP Spoofing. 2011 *IEEE-International Conference on Recent Trends in Information Technology.*

Kolahi, S. S., Alghalbi, A. A., Alotaibi, A. F., Ahmed, S. S., Lad, D. (2014). Performance Comparison of Defence Mechanisms Against TCP SYN Flood DDoS Attack. *6th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*

Kolahi, S. S., Treseangrat, K., Sarrafpour, B. (2015). Analysis of UDP DDoS Flood Cyber Attack and Defense Mechanisms on Web Server with Linux Ubuntu 13. *International Conference on Communications, Signal Processing, and their Applications (ICCSPA), 2015*

Kotenko, I., Ulanov, A. (2006). Simulation of Internet DDoS Attacks and Defense. *Volume 4176 of the series Lecture Notes in Computer Science pp 327-342*

Koutepas, G., Stamatelopoulos, F., Maglaris, B. (2004). Distributed Management Architecture for cooperative detection and reaction to DDoS attacks. *Journal of Network and Systems Management, 12. (1),* 73-94.

Kuhrer, M., Hupperich, T., Rossow, C., Holz, T. (2014). Exit from Hell? Reducing the Impact of Amplification DDoS Attacks *Proceedings of the 23rd USENIX Security Symposium*

Land, R. (2004). Understanding evolution of information systems by applying the general definition of information. *26th International Conference on Information Technology Interfaces, 2004.*

Lanneli, N. & Hackworth, A. (2005). Botnets as a vehicle for online crime. *Cert Coordination Center, Pittsburgh PA, http://www.cert.org/archive/pdf/Botnets.pdf*

Lemon, J. (2002). Resisting SYN flood DoS attacks with a SYN cache. *BSDC'02 Proceedings of the BSD Conference 2002. 10.*

Li, K., Zhou, W., Li, P., Hai, J. (2009). Distinguishing DDoS Attacks from Flash Crowds Using Probability Metrics. *Thirds Internation Conference on Network and System Security, 2009. NSS '09.*

Li, W., Chen, L., Lei, Z. (2010). Alleviating the impact of DNS DDoS Attacks. *2010 Second International Conference on Networks Security, Wireless Communications and Trusted Computing*

Lin, Q., Xie, J., Shen, Z., Xu, X. (2012). DR^3: Optimizing Site Selection for Global Load Balance in Application Delivery Controller. *2012. Open Cirrus Summit (OCS), 2012 7th*

Lutz, D. (2012). DNS Dampening. *http://lutz.donnerhacke.de/eng/Blog/DNS-Dampening*

Microsoft (2013). Overview of Flood Mitigation

Mirkovic, J., Reiher, P. (2004). A taxonomy of DDoS attack and DDoS defense mechanisms. *ACM SIGCOMM Computer Communications Review, vol. 34, no. 2, pp. 39-53, April 2004.*

Ohsita, Y., Ata, S., Murata, M. (2012.). Detecting distributed denial of-service attacks by analyzing TCP SYN packets statistically. *Global Telecommunications Conference, 2004. GLOBECOM. IEEE, 4, 2043 - 2049.*

Peng, T., Leckie, C., Ramamohanarao, K. (2007) Survey of Networkbased Defense Mechanisms Countering the DoS and DDoS Problems *ACM Comput. Surv., vol. 39, no. 1, Apr. 2007.*

Prince, M. (2012). Deep Inside a DNS Amplification DDoS Attack. *https://blog.cloudflare.com/deep-inside-a-dns-amplification-ddos-attack/*

Prince, M. (2014). Technical Details Behind a 40GBPS NTP Amplification DDoS Attack *http:// blog.cloudflare.com/technical-details-behinda-400gbps-ntp-amplification-DDoS-attack*

Provos, N., Rajab, M. A., Mavrommatis, P. (2009). Cybercrime 2.0: When the cloud turns dark. *Comm. ACM 52, 42–47.*

Purvanto, Y., Kuspriyanto, Hendrawan, Rahardjo, B. (2014). Traffic Anomaly Detection in DDoS Flooding Attack. *8th International Conference on Telecommunication Systems Services and Applications (TSSA), 2014.*

Radware Emergency Response Team. (2014). *Radware Threat Alert – Tsunami SYN Flood Attack – 10/7/14*

Rawal, B., Ramcharan, H., Tsetse, A. (2013). Emergence of DDoS resistant augmented Split architecture. *10th International Conference on High Capacity Optical Networks and Enabling Technologies.* 37-43

Reading, D. (2014). High Bandwidth NTP Amplification DDoS Attacks Escalate 371 Percent In The Last 30 Days. *http://www.darkreading.com/attacks-breaches/high-bandwidth-ntp-amplification-DDoS-attacks-escalate-371-percent-in-the-last-30-days/d/d-id/1141460?*

Rodriguez-Gomez, R. A., Macia-Fernandez, G., Garcia-Teodoro, P. (2013). Survey and Taxonomy of Botnet Research through Life-Cycle. *ACM Computing Surveys, Volume 45, Issue , 45:1-45:33, Aug. 2013.*

Rossow, C. (2014). Amplification Hell: Revisiting Network Protocols for DDoS Abuse. *In Symposium on Network and Distributed System Security (NDSS).*

Rozekrands, T., de Koning, J. (2014). Defending against DNS reflection amplification attacks. *University of Amsterdam, System & Network Engineering RP1*

Sejdini, V., Xiaoming, L., Chowdhury, H. (2006) Denial of Service (DoS) attack with udp flood. *School of Computer Science, University of Windsor, Windsor, Ontario, Canada.*

Singh, A. & Junefa, D. (2010), Agent Based Preventive Measure for UDP Flood Attack in DDoS Attacks. *IJEST, vol.2, no. 8, 2010, pp. 3405- 3411.*

US-CERT. (2013). DNS Amplification Attacks. *Alert (TA13- 088A)*

Vasileios, P., Dan, M., Lixia, Z. (2007). Enhancing DNS Resilience against Denial of Service Attacks. *International Conference on Dependable Systems and Networks, 2007. DSN '07. 37th Annual IEEE/IFIP*

Wesley, R. (1993). TCP/IP Illustrated Volume 1: The Protocols

Wired.com (2000). Yahoo on Trail of Site Hackers, Feb. 8, 2000. http://archive.wired.com/techbiz/media/news/2000/02/34221

Zhang, Y., Liu, Q., Zhao, G. (2010). A real-time DDoS attack detection and prevention system based on per-IP traffic behaviorial analysis. *2010 3rd IEEE International Conference on Computer Science and Information Technology*