6-2016

PACKET FILTER APPROACH TO DETECT DENIAL OF SERVICE ATTACKS

Essa Yahya M Muharish
essa.muharish@outlook.com

Recommended Citation

Muharish, Essa Yahya M, "PACKET FILTER APPROACH TO DETECT DENIAL OF SERVICE ATTACKS" (2016). *Electronic Theses, Projects, and Dissertations.* Paper 342.

PACKET FILTER APPROACH TO DETECT DENIAL OF SERVICE ATTACKS

A Project

Presented to the

Faculty of

California State University,

San Bernardino

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Computer Science

by

Essa Yahya M Muharish

June 2016

PACKET FILTER APPROACH TO DETECT DENIAL OF SERVICE ATTACKS

A Project
Presented to the
Faculty of
California State University,
San Bernardino

by

Essa Yahya M Muharish

June 2016

Approved by:

Dr. Zhengping Wu, Adviser, Computer Science

Dr. George M. Georgiou, Committee Member

Dr. Owen J. Murphy, Committee Member

ABSTRACT

Denial of service attacks (DoS) are a common threat to many online services. These attacks aim to overcome the availability of an online service with massive traffic from multiple sources. By spoofing legitimate users, an attacker floods a target system with a high quantity of packets or connections to crash its network resources, bandwidth, equipment, or servers. Packet filtering methods are the most known way to prevent these attacks via identifying and blocking the spoofed attack from reaching its target. In this project, the extent of the DoS attacks problem and attempts to prevent it are explored. The attacks categories and existing countermeasures based on preventing, detecting, and responding are reviewed. Henceforward, a neural network learning algorithms and statistical analysis are utilized into the designing of our proposed packet filtering system.

ACKNOWLEDGEMENTS

I take this opportunity to thank my family, classmates, friends, lecturers and professors for their consistent support as I pursued this research. Without their continuous support such as constant encouragement, advice, prayers and love I would not have successfully to conduct this work to completion.

I wish to give my special thanks and recognition to my advisor Dr. Wu for giving me the freedom to work in the area where I have deep interests and gave me tremendous support which I acknowledge with utmost sincerity. His moral support and advice gave me confidence and I was able to achieve more in the subject area and I felt inspired towards achieving my goals.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

In today's world, every service that we accessed traditionally, such as banking, payment of utility bills, buying a car or a house, buying groceries, and watching a movie, can be accessed as an online service. Many businesses such as finance, utility services, and entertainment have an online presence. For instance, it is now possible to buy and sell shares through a web portal or through a mobile application as opposed to the traditional way of calling a stockbroker to execute a trade for you. Another example is the entertainment industry, whereby companies such as Amazon and Netflix have video-on-demand online services such that people can stream movies as a service. Online banking is another critical industry, whereby customers use mobile applications or online portals to do financial transactions such as making payments as well as receiving payments online.

Online games are common, making it possible to find various players competing from different parts of the world simultaneously in the same game. Music streaming services such as Spotify, YouTube, and Apple's iTunes are another example of entertainment channels that are very dependent on the online presence and they are used by millions of users. Online payment platforms such as PayPal, Visa, American Express and Stripe are all dependent on the internet technologies. Other services such as online booking and ticketing of airline tickets, online booking of hotel rooms, online shopping, online booking of taxi services such as Uber are all examples of how our lives are dependent on online applications, which should have 100% availability all the time. However, this is can be made impossible by attacks such as denial of service (DoS), which can make an online service to be unavailable to its legitimate users.

What is a Denial of Service Attack?

A denial of service attack happens when an attacker make a website or other internet-based applications or services to be unreachable and unavailable to the legitimate users [1]. Denial of service can also be defined as a malicious attack, whereby an internet-based service becomes unavailable to the users by interrupting the normal functionality of the hosting server of the application [2]. Normally, the denial of service (DoS) attack involves compromising a single instance of a key internet device such as a hosting server or a network switch that uses a single internet connection and interrupting its normal functioning. A denial of service (DOS) attack involves compromises multiple devices such as servers, network switches, or routers that are located in different geographical locations across the globe. They are using different networks, and as such, the denial of service is distributed in multiple locations and there is no backup [2].

A denial of service attack involves an attacker who exploits a vulnerability in a device such as a hosting server and makes it a denial of service master. This DoS master is infected with malicious software and it infects other devices such as end-user devices with the malware [3]. A denial of service attack involves a number of computing devices that are compromised, also called master-bots or botnets. They consistently send a large amount of requests or data packets to slave-bots or to the internet connection of the slave-bots, which causes the entire system to collapse, making it unavailable or unreachable [4]. Attackers create many botnets that are also called zombies, which make multiple requests to the servers of an application such as website, jamming the servers, exhausting the computing resources such as RAM or network bandwidth such that legitimate requests form legitimate users cannot be fulfilled making the entire website unavailable [5]. Victims of a denial of service attack usually become overwhelmed by the huge volumes of data packets that are sent from multiple sources, causing congestion in the networks [6].

The Architecture of a Denial of Service Attack

To develop a solution to prevent or counter the denial of service attacks, we have to understand the nature and the architecture of denial of service attacks so that we can develop a solution to prevent them them. Denial of service attacks can be classified into three broad categories.

Scanning Attacks

These attacks involve attackers scanning computers on the internet and attacking those that are vulnerabilities. The attackers identify the weak computers, finds a backdoor through which to attack and install malicious software (malware). The malware is usually highly advanced such that it can replicate itself and spread quite fast without raising attention [7]. The most common scanning attacks are worm scanning which can affect the address resolution protocol (ARP) of network devices. Multicast scanning is another scanning attack whereby there are adverse effects on the memory and CPU utilization of networking equipment such as routers, servers and end user devices such as workstations.

- Random scanning

This is a scanning mechanism whereby attackers target random IP addresses globally, which affects the global traffic of the internet because these attacks lack synchronization. The random scanning often comes from computers that are compromised.

- Selective scanning

This is a denial of service attack whereby a predetermined list of IP addresses are scanned. Once the IP addresses in the predetermined list are scanned, the attacker can infect the computing infrastructure within the list with a malware within a very short duration such as 30 seconds. Since the predetermined list of targeted IP addresses is usually very long, the process of attacking the computers with a malware is supposed to be very short and fast (in 30 seconds). This process of attacking would usually be very resource intensive such that a huge amount of traffic would be generated and computing resources such as CPU time may be exhausted causing a serious DoS attack.

- Signpost scanning

This is a scanning technique that happens between a victim host and newly identified targets. This attack is heavily influenced by the fact that the victim host has a constant communication with the new targets such as emails. Here, computers that are under attack use any link that they have with target computers to propagate an attack. The attack could use email addresses stored in the compromised host to send malware to the targeted computers. Additionally, signpost scanning can be propagated by use of worms that can be hidden in web links such that when potential victims click on such links, they become infected. This kind of attack is does not generate high volumes of internet traffic but it depends on the rate at which victims click on such a link. Eventually, various computers in the internet would be infected by the malware such that a denial of service for victims may happen. It can be in the form of users not being able to use their online accounts for services that require them to login because the malware has corrupted the application, for instance, assume that a word document file which is corrupted and infected with a malware is shared as an email attachment or as a link which users click to view or download. Once they download and open the word document file, the hidden worm attacks the Microsoft office application in their computers such that it becomes corrupt and they cannot access their online Microsoft accounts, thereby causing a denial of service attack. This can happen in many other applications such as online games whereby attackers may attack a communication port that is a common route of communication between remote online gamers such that it is blocked and the gamers cannot access a particular service such as reward system that is online thereby causing a denial of service attack.

Spoofing Attacks

Attackers attack spoofed addresses in the following ways:

- Random spoofing

Attackers can spoof randomly generated source addresses and start sending huge amounts of packets to them. This can be made even more sophisticated when attackers use subnet spoofing so that they can get through firewalls and routers.

- Subnet spoofing

  Here, an attacker spoofs addresses in a random manner such that targets using router-based ingress filtering can be easily compromised. To prevent this kind of attack, IP addresses should be bound to a specific MAC address and a specific subnet in the network [7].

Target Attacks

Most denial of service attacks may target the following resources:

- Server application

  Attackers may lock out the legitimate users of an application by exhausting all the resources in a server such that the application becomes unavailable [7].

- Network access

  Attackers may cause a denial of service by making network access unavailable. Attackers may achieve this by cutting access to the network or by overloading the network with massive requests in attacks such as UDP flooding attack [7].

Denial of Service Attack Forms

A denial of service attack can be classified into the following three categories [8].

Increased Bandwidth Consumption Attack

Here, the denial of service attackers increases the network traffic to the victim servers and network devices, such that; there is overconsumption of bandwidth. The network is said to be flooded with requests, making it difficult for legitimate network requests from normal and genuine users to reach the website such that the infrastructure becomes jammed and it cannot route traffic to the resources that are being requested by the users. A request may be a PDF file or even just a login page to the website. This is the most basic DoS attack and it involves attackers using several slaves that are also known as zombie machines, which are infected with a malware that continuously makes requests to the servers of the victim network.

Exhaustion of Resources

This class of attack exhausts the computing resources that have been allocated to an application. For instance, a file-processing server or an email server may have limitations or resource constraints, such that; it can handle a certain number of user sessions at a given time. The servers may be designed in such a way that they can handle X number of HTTP user sessions and unless these sessions have been ended, no new sessions can be handled by the server. A possible attack could involve holding the system resources 'hostage' such that an attacker may decide to make a request that involves an X number of HTTP user sessions that are not legitimate, which has the consequences of locking out genuine sessions from being active. This can be achieved by the use of slave machines, which can establish X user sessions to other servers of the application.

Attacks Involving Exploiting the Application

This is the most sophisticated of all the DoS attacks. The attackers exploit any weaknesses that they can identify in the application, including loopholes in the architecture or design of the hosting devices such as servers. This could exploit logical weaknesses or errors in the programs that have been hosted on the servers. For instance, an attacker may exploit a design flaw in a program that locks user accounts after an unsuccessful number of login attempts by attempting to login into user accounts and eventually locking out all users, making the system unusable by all genuine and legitimate users.

Types of Denial of Service Attacks

Most denial of service attacks involves overloading the networks requests with massive requests, which exceed the resources available to fulfill those requests. The victim infrastructure is usually filled with requests, which can reach the maximum number of resources set for that server [9]. Most denial of service attacks produce millions of packets per second, which can crash devices such as servers and end-user machines. There are several types of DoS attacks such as:

Volumetric Attacks

These are attacks that are on Layer 3 and Layer 4 of the OSI model, whereby they cause a DoS attack by generating huge amounts of network traffic. The high volume of network traffic affects the infrastructure of the network such as the servers, routers. These attacks may involve a TCP flood, UDP flood or ICMP flood. Attackers use botnets or slave computers to generate the huge volumes of requests, such that; the source address of the packets is hard to identify. This is the most common attack.

Protocol Attacks

These are a denial of service attacks that involve attackers who drain the resources on a server and other network devices such as routers and firewalls. For instance, in a TCP flood attack, an attacker targets all the TCP ports that are open so that he can exploit them by passing illegitimate traffic through the ports.

Reflection or Amplification Attacks

These are attacks, whereby attackers exploit weaknesses on the internet, making it possible to increase their internet traffic by a certain factor, such that; they can be able to reroute it through a third party server. This increases the traffic in the third party server slowing it down, which can deny the legitimate users the ability to access it.

The Open Systems Interconnection Layers Attacks (OSI)

There are several denial of service attack possibilities in the OSI layer as follows [10].

- At layer 7 – The application layer.

  This is the data layer where the creation of packets and messages starts. Databases can be accessed at this layer and protocols such as file transfer protocol (FTP), Telnet, email protocols such as POP3 and SMTP are found in this layer. Attackers may create denial of service attacks in the form of GET and POST requests such that there could be abnormal requests to certain resource files such as login.php or register.php. Attackers mayclose several botnets that may repeatedly make requests for login.php or register.php (which are GET requests) and eventually exhaust the system resources, thereby denying users who genuinely need to login or register with the application a chance to do so due to exhaustion of resources in the servers. Additionally, attackers may make multiple FTP (file transfer protocol) requests to the servers by requesting a particular file which can jam the servers due to resource starvation.

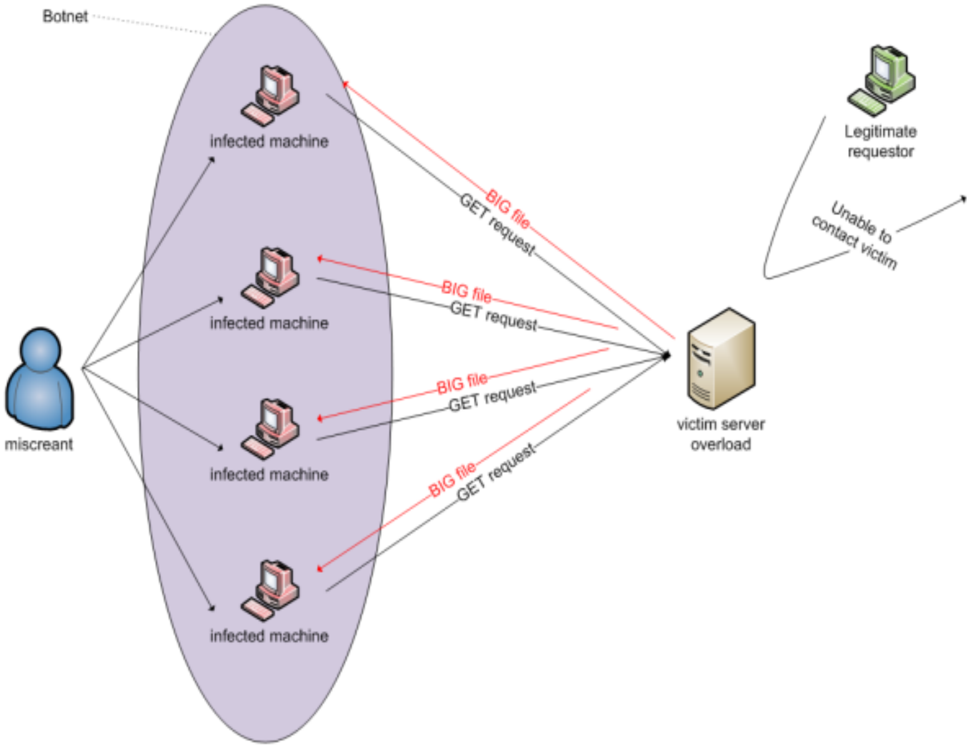

Figure 1: This figure shows a DoS Get request

This attack can be mitigated by use of third party application monitoring systems. These systems can report any increase in the number of requests and any increased load on the server. These systems can alert the system administrator through alerts such as an email, or an SMS. This could help the system administrator perform a collective action to prevent denial of service attack.

- Attack at layer 6 – the presentation layer

Layer 6 is also a data layer where there is the translation of the data that has been received between the sender and the receiver which is achieved by use of compression and encryption protocols such as SSL (secure session layer) certificates. An attacker may exploit weaknesses in the way the SSL certificates have been configured such that bad requests may start being sent to the server. This could make the server reject such requests. The web application may keep on restarting which renders it unusable for the users.

This attack can be mitigated by the use of applications delivery platform that guarantees that information on the server is always encrypted and that any information that resides on the server is secure.

- Attack at layer 5- the session layer

Layer 5 involves the synchronization and termination of network connections. An attacker may use a denial of service attack at this level such that flaws in the telnet are exploited at the switch level, which would make the switch services unavailable. This locks out the administrator from controlling the switch. This attack can be mitigated by updating the software of the router with the latest software from the equipment provider.

- Attack at layer 4 –the transport layer

  This layer has the role of ensuring that messages from layer 1, 2 and 3 are error free. This layer uses the UDP and the TCP protocols. There are several attacks at this layer.

  - UDP flood attack.

    According to [11], a UDP flood attack occurs when attackers send streams of UDP packets to several ports of the victim machine such that the victim would be busy sending ICMP messages to the attacker's machine. This blocks any legitimate requests from getting into the system [11].

  - Smurf attack.

    This is quite an old DoS attack, whereby the attacker sends an echo request to a routing device in a network such that the source of the data is hidden, but the request is sent to all the devices in the network through a broadcast address which causes each of the devices that received an echo request to send a reply the sending machine which is the victim [11].

  - SYN flood attack

    This is an attack that occurs in web based systems. Communication on the internet involves synchronization (SYN) between a sending and a receiving computer. A SYN flood attack occurs when a huge number of data packets are sent to a receiving server such that a huge number of responses come from the server followed by a huge number of pauses which exhausts all the resources of the server which blocks any legitimate requests [11].

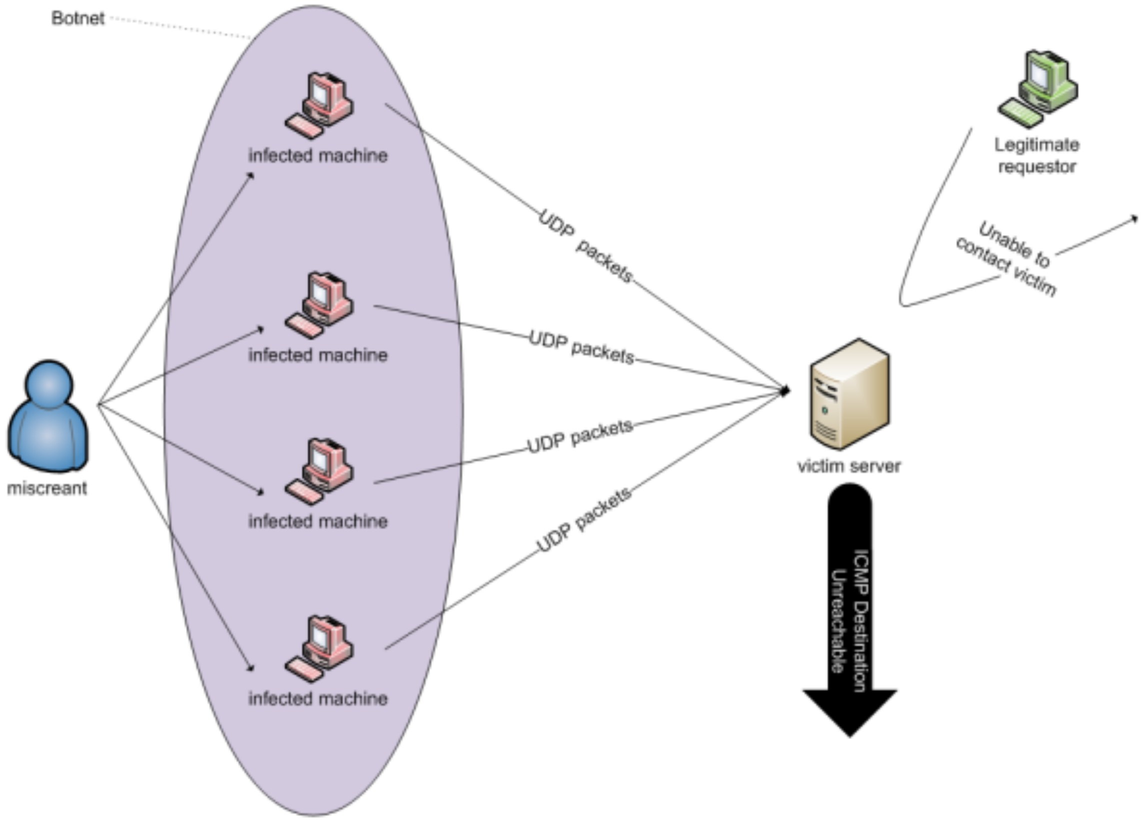

Figure 2: A UDP flood attack

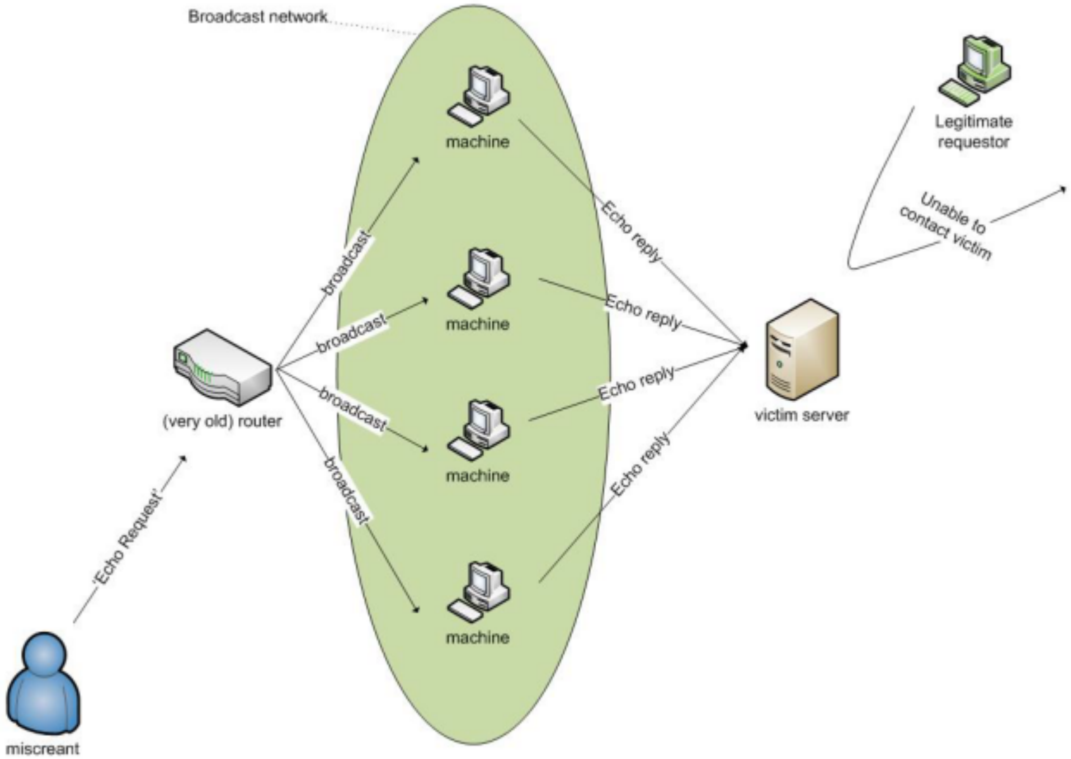

Figure 3: A smurf attack

- Layer 3 attack – Network layer

This is the OSI layer level that plays the role of routing and switching of data packets to different networks and LANs. It relies on the ARP, ICMP, IP and RIP protocols, and relies on routers. A denial of service attack occurs in this layer such as ICMP flooding [10]. ICMP flooding involves a large number of ICMP messages that jam the bandwidth of the victim network. This overloads the firewall.

Attacks on this layer can be mitigated by the use of various DoS attacks-blocking techniques such as block-holing. Black-holing can be implemented by the internet service providers (ISP) to stop possible attacks on their customers. Possible denial of service attacks may include a slow internet connection or disruption of internet connection services to the customers.

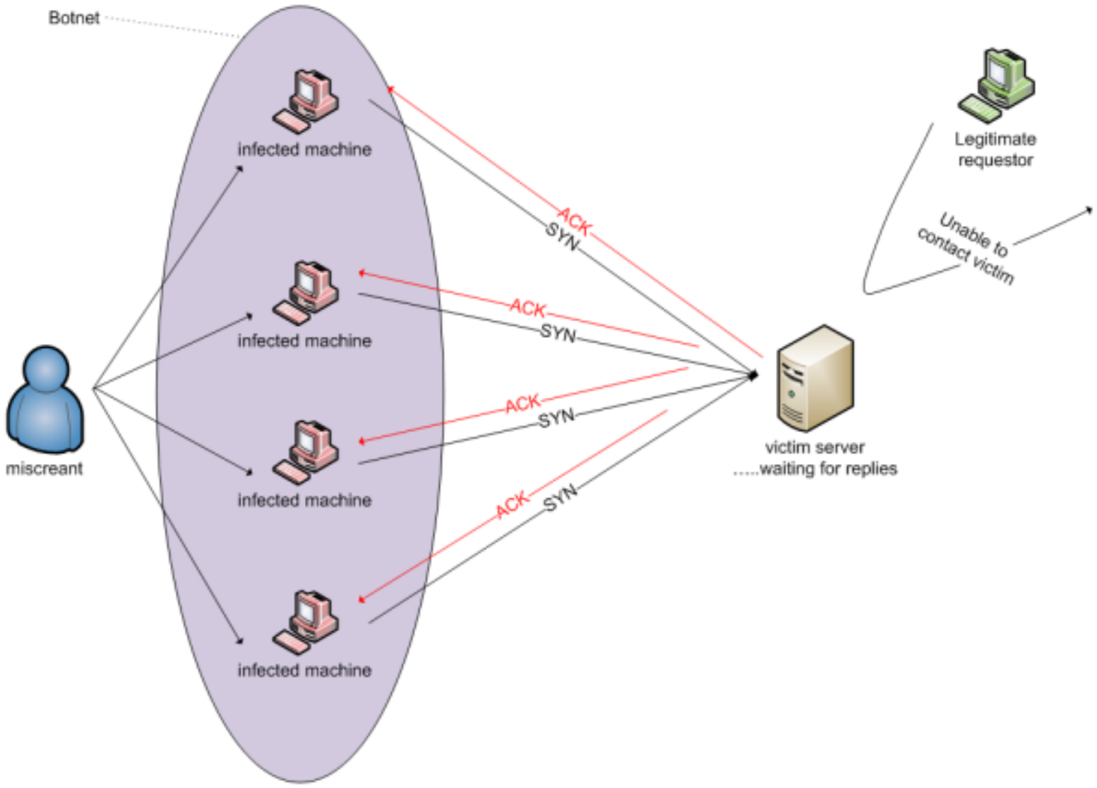

Figure 4: A SYN flood attack

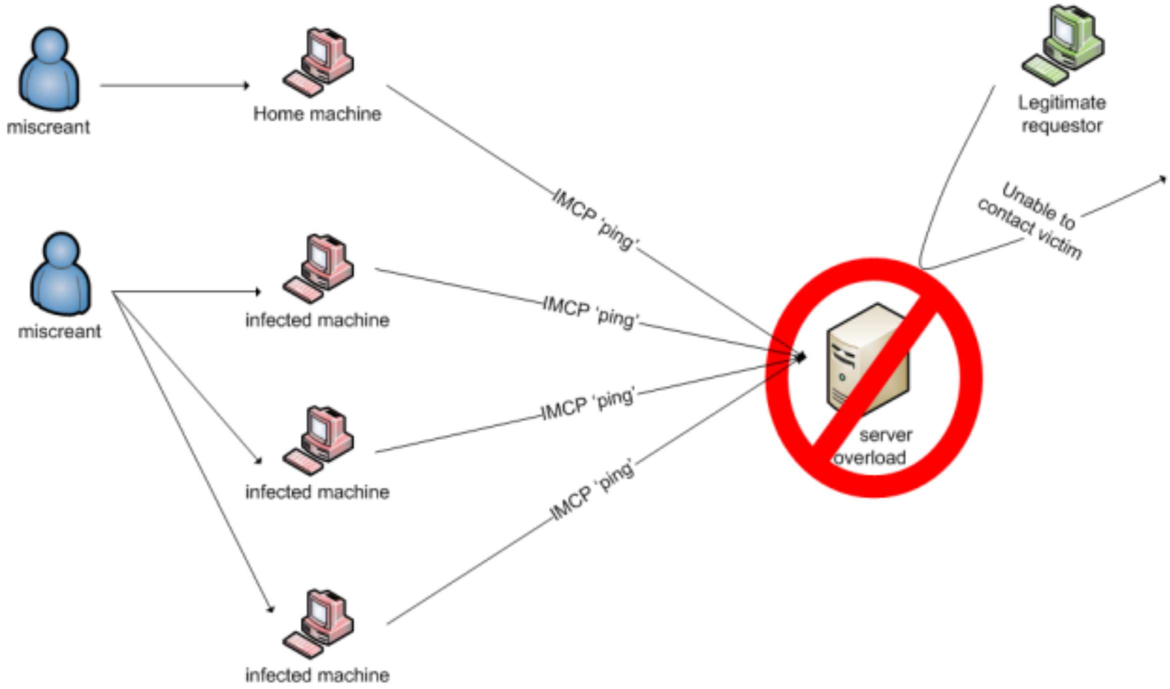

Figure 5: ICMP ping flooding attack

- Attack on the data link layer

  According to [10] this layer has the role of ensuring that data is successfully transferred over the physical layer. A DoS attack involves MAC flooding which affects how the data is sent from sender to recipient

What is the Motivation of DoS Attacks?

There are various reasons attackers would target a website with a denial of service attack.

Extortion and Profit

Attackers may be motivated to make profits through extortion. They create DoS attacks and blackmail their victims with the hope of making money when the reverse the DoS attack so as to allow the users have access to the system [9]. The victims which tend to rely on the system to conduct business end up giving in to the demands of the attackers so that they can regain control of the mission critical system and continue conducting their business. There are several ways to profit such as disrupting competitor businesses so as to make margins when customer traffic is routed to other online businesses such as online stores. Attackers may attack on weekends where the businesses are busy fulfilling customer requests, but there are no IT staff members to offer IT support against a DoS attack [9].

Distraction

Attackers may create an attack so that they can distract users from the attention of where the real attack is. Attackers may cause a denial of service just to keep the resources of a victim exhausted and busy as the attackers are engaged in stealing data in another site [9].

Collateral Damage

A website may experience a DoS mainly because it was not performed with malicious goals. Some sites may face a DoS attack mainly because they do not have the capacity to handle huge volumes of traffic requests. Also, attackers may attack a whole hosting server that hosts other websites that were targeted by the attackers and the website becomes a victim in the process [9].

Hack-Activism

Here, attackers may cause a denial of service attack so that they can campaign against something. A famous group of online hackers called Anonymous is famous for the internet activism, whereby they give warnings beforehand that a certain online service will be down for various reasons such as political reasons [9].

CHAPTER TWO

RELATED WORK

A DoS attack can be a disaster for a business and it can wipe out a significant amount of revenues from the business earnings. For instance, Yahoo experienced a DoS attack in 2000, which affected its online services for about two hours. This led to a significant loss in the advertising revenues of the company [12]. It is, therefore, important for companies to be preventive instead of being reactive in their defense mechanisms against DoS attacks. There are various defensive mechanisms against a DoS attack. Some are discussed below:

Popular Used Techniques

These techniques are those defensive mechanisms that are not applied to a specific method of DoS attack. Here is a list of some of these techniques:

- Application monitoring

  Continuous applications using various technologies and algorithms can be a defensive mechanism so as to identify indicators of compromise such as increased network traffic, increased database requests, and increased bandwidth consumption. If any of these indicators of compromise have been detected, corrective and mitigation measures can be taken to defend the application from a DoS attack [10].

  There should be systems and applications in place to remotely monitor servers for instance. In-house monitoring tools could be of little help if they are under attack as well and companies are advised to have remote monitoring infrastructure such as third party services that can monitor the websites and servers remotely and can send alerts such as email or SMS remotely [13].

- Black-holing or sink-holing

This is a defensive mechanism, whereby all traffic is blocked and transferred somewhere else that is called a black hole such that it is discarded. This is ineffective because it affects both legitimate and illegitimate traffic. This can be useful in protecting computer infrastructure from further damage if the denial of service attack has already occurred [13].

- Applying security patches

Host computers should always be updated with the latest security updates so as to be ready for the latest denial of service attacks.

- Changing IP addresses

This is a technique whereby a victim IP address is changed so as to invalidate the address to which the attacker is sending massive requests. This is a temporary defensive technique because an attacker may start sending massive packets to the new IP address.

- Over provisioning of resources

Companies can buy extra bandwidth or extra network infrastructure to handle fluctuations in resource usage that can be caused by a denial of service attack. This can be achieved by outsourcing the provision of services such a bandwidth and network infrastructure which can be requested on a demand basis, which is cost effective because there is no capital expenditure for the business to invest in idle and redundant infrastructure. Businesses should choose internet providers that are quick enough to adjust bandwidth provision on a need basis and very quickly [14].

- Load balancing

This is a defense mechanism that enables infrastructure providers to increase bandwidth connections so as to prevent mission critical systems from collapsing in case there is an attack. Server instances can be replicated such that there is load balance and once a server is attacked by a denial of service attack, there is a replica of the same program running on another server and the whole system would not collapse.

- Cloud mitigation provider

The cloud computing platform is the way to go in the current technological world. Companies may rely on cloud mitigation providers for defense mechanisms against any denial of service attack [13]. This is because cloud solution providers have the technical capacity and infrastructure capabilities in multiple locations that can be able to carry any network traffic.

Depending on a cloud provider for hosting applications as opposed to locally hosting applications in-house has numerous advantages such as.

  - Technical expertise

    Cloud providers have the technical expertise that is needed to secure infrastructure since that is their core business. They can afford to have the best network and security engineers that are expensive to hire in-house. A cloud provider may have engineers who are more knowledgeable in mitigating denial of service attacks.

    - Huge bandwidth capacities

      Cloud providers have a huge bandwidth capacity, which is advantageous to a business because it may be useful in case there is a denial of service that might consume a lot of bandwidth. The cloud provider may increase the bandwidth requirements on a need basis, which can save an application from being inaccessible.

    - Lots of DoS Mitigation Infrastructure

      Cloud providers have huge investments in the best and powerful hardware technologies that can be useful in mitigating against the complex denial of service attacks, which can crash hardware of a client if their application is hosted in-house.

- Use of an internet service provider (ISP)

  This is a technique that companies can use as a defensive mechanism against denial of service attacks. Internet service providers usually have large computing resources than enterprises would. Therefore, companies can take advantage of this and mitigate against denial of service attacks, which may consume large volumes of bandwidth and other computing resources.

Filtering Techniques

There is no complete solution for detecting DoS attacks. Each technique has its benefits and limitations. Here is a list of some of these techniques:

- Intrusion prevention

  The best way to prevent denial of service attack from happening is complete prevention, such as using filters that are synchronized and coordinated globally such that data packets that are detected to be originating from attackers can be stopped before they cause real damage [15]. This can be achieved using two filtering techniques [4].

  - Ingress filtering

    It is a defensive approach whereby a router is designed such that inbound traffic is blocked if its source address is suspicious flagged as not genuine. Incoming packets originating from IP addresses that don't match the prefix of the domain are blocked. This reduces a denial of attack that can be caused by IP spoofing. This protects the system from resource exhaustion.

  - Egress filtering

    This is a defense mechanism where outbound traffic is only assigned to IP addresses that are known to be legitimate such that it protects other domains from being hit by huge traffic from the IP addresses of the company if the company devices become botnets.

- Distributed packet filtering

  This is a defense mechanism, whereby spoofed IP addresses can be filtered to prevent an attack on targets and it helps in getting a traceback of IP addresses. It is route based [13].

- History based IP filtering

  Using this mechanism, a database of IP addresses is prebuilt based on the connection history of a router. This mechanism is quite effective and robust and it can be applied on a variety of types packets. This helps to block incoming traffic requests from source addresses that are unknown to the database [1].

- Secure overly services

  This is a defense mechanism whereby only traffic that is coming from a selected few network nodes is accepted to be genuine and is allowed to reach the servers while any other traffic is rejected. This mechanism is ideal for protecting a particular server and it may not be ideal for protecting public servers.

Intrusion Detection Systems

According to research, intrusion detection systems can be used to detect any changes in the traffic network of the system [14]. This protects the system from being the target of an attack or being the source of an attack [16]. Intrusion detection systems detect whether there are any anomalies in the system such as increased traffic. This can be achieved by the use of firewalls and routers which can be designed in such a way that they can prevent the system from spoofing and ping attacks by intruders. They can be configured in such a way that the can block IP addresses that constantly ping a network for a number of times. A router may not be able to effectively mitigate against complex spoofing attacks as well as attacks at the application level, especially if they are using legitimate IP address. Firewalls, on the other hand, can be able to detect and block unusual traffic, but complex DoS attacks can bypass the wall.

Using Data Mining and Machine Learning Methods

Data mining is a technique that is used to establish patterns and relationships within large datasets which are achieved by use of tools such as visualization, clustering of data as well as classifications and associations of data.

Data mining has been widely used in the fields of marketing so as to establish customer buying and spending patterns which are useful in businesses deep insights into customer behavior. Data mining is now being chosen as a reliable technique for analyzing network data. The application of data mining techniques in the analysis of networking data is very useful in identifying network data patterns that can be used in monitoring the security of a network

Data mining uses two strategies known as supervised learning and unsupervised learning techniques. In supervised learning, data instances are usually mapped to a data label and find patterns associated with that data label. On the other hand, unsupervised learning is used to discover data patterns, subsets within data without having prior knowledge of the data. Blind signal separation, self-organizing-maps and clustering are examples of unsupervised learning [17, 18].

Several data analysis techniques can be used to analyze and monitor the state of a network. The top-down learning method is a statistical method that is used to detect anomalies in a network especially when there is knowledge about the relations in a dataset and mathematical computations are used to arrive at conclusions. Machine learning is used when there is prior knowledge about data patterns and is know as a bottom-up learning method. Machine learning and data mining are two data analysis patterns whose roles overlap in many ways. However, machine learning is unique because it uses predictive methods based on known properties of the data while data mining, on the other hand, involves discovering of unknown properties of the data. Machine learning has advanced statistical techniques to do regression and classification of data that has numerous dependent and independent variables.

Decision trees are used to detect possible attacks because they are useful in detecting anomalies in network data. Additionally, data mining techniques are useful for detecting denial of service attacks. Machine learning techniques like Random Forest, Naïve Bayes and support vector machines can also be used to detect the anomaly in network traffic such that a DoS attack can be detected and prevented early enough. Classifier machine algorithms such as KNN(Kth Nearest Neighbor) and fuzzy logic are useful in detecting anomalies in the network traffic which can help identify an imminent denial of service attack [17].

CHAPTER THREE

SYSTEM ARCHITECTURE

This chapter discusses the primary work of this project. It will address the architecture design of the project as well as the formulas and the algorithms that are used to do the work.

Here is the projected system that we which is designed as defensive approach against denial of service attacks. The system architecture is shown below.

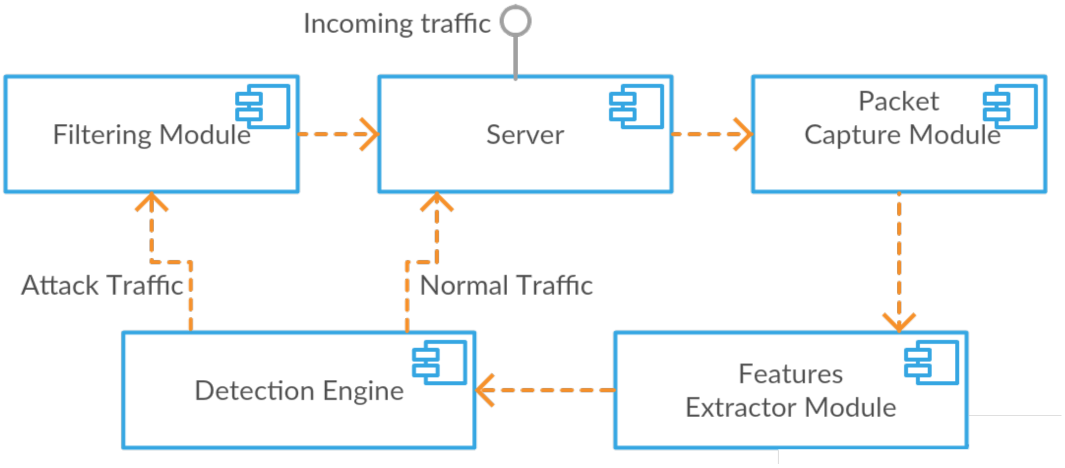

Figure 6: The proposed DoS detection system

The Proposed System's Modules

As in figure 6, the proposed system has several modules which are:

Packet Sniffing/ Capturing Module

This is a very key module in the system because it captures all the inbound traffic and captures the traffic on a fixed time duration. once the packets have been captured they are transferred to the detection module.

Feature Extraction Module

In this module, some features of packets (P_i) such as packet lengths (PL) and packet time lengths (PT) are extracted from every unique IP address such that they are compared with the features of attack packets. Correspondingly, a neural network self-organize-map clustering is applied to show the variation of how attack and normal traffic are clustered. Some of the features that are introspected include the following:

- Total Packets Count (TPC)

$$TPC = \sum_{i=1}^{N} P_i$$

- Total Packets Length(TPL)

$$TPL = \sum_{i=1}^{N} PL_i$$

- Average Packets Length (APL)

$$APL = \frac{1}{N} \sum_{i=1}^{N} PL_i$$

- Packets Length Variance (PLV)

$$PLV = \frac{1}{N} \sum_{i=1}^{N} |PL_i - APL|^2$$

- Average Length Differences (ALD)

$$ALD = \frac{1}{N} \sum_{i=1}^{N} (PL_{i+1} - PL_i)$$

- Total Packets Time (TPT)

$$TPT = \sum_{i=1}^{N} PT_i$$

- Average Packet Time (APT)

$$APT = \frac{1}{N}\sum_{i=1}^{N} PT_i$$

- Packet Time Variance (PTV)

$$PTV = \frac{1}{N}\sum_{i=1}^{N} |PT_i - APT|^2$$

- Average Time Differences(ATD)

$$ATD = \frac{1}{N}\sum_{i=1}^{N} (PT_{i+1} - PT_i)$$

- Packet Transfer Rate (PTR)

$$ALD = \frac{TPL}{TPT}$$

The extracted features are grouped from different sources to one specific destination are regulated in a table to sends to the Detection Engine to analyzed and compared.

Detection Engine

In this module, a neural network algorithm are used to classify whether the features of a packet that have been extracted by the feature extraction module above belong to normal packets or to packets that can be associated with an attack. The construction of this detection model is mainly using a Feed-forward supervised neural network with MLP backpropagation learning algorithm, mean square error, and sigmoid function to identify the packets that are related to an attack.

In this neural network type, connective links between the input layer and the output layer are always going forward and does not go back to the previous layer. Each layer has several number of neuron that is connected to all the neuron in the prior layer. The connection between neurons has a variety of different weights. These weights are the determinant of the network knowledge base. The following diagram shows the flow of the Feed-Forward Neural. The data from feature extraction module will access the input layer where its weight calculated and move across the network to the output layers. In the output layer, weight calculation will be used to compare the output to the pre-calculated weights of the attacks.

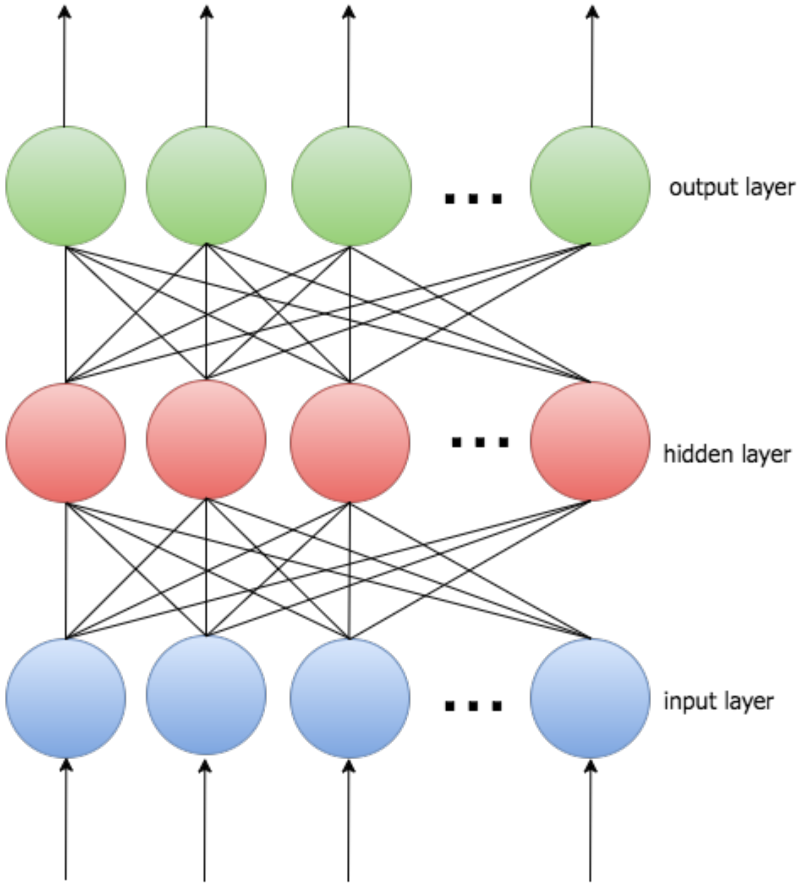

Figure 7: The feed-forward neural network

The Mean Square Error. In the neural network, the network performance can be measured using the mean square error function. If we have n projections of feature f' and perceived vectors of f values that corresponded to the input function that produced the projections, the mean square error of the projector is calculated as following [19]:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\bar{f_i} - f_i)^2$$

Sigmoid activation function. The sigmoid derivative function is used to calculate the layer's output from its net input [18].

$$f(n) = \frac{1}{1 + e^{-n}}$$

Backpropagation Learning Algorithm. This learning algorithm performed in two main stages:

- First stage

  The first stage is the deployment where each sample input has to be propagated forwardly from input layer over the neural network to calculate its activations output. Then, the obtained results are compared with desired output using the calculate of the mean square error from each neuron in hidden layer.

- Second stage

  The second stage is updating the weight values. Each neuron weights' value, that's belongs to the hidden layers, is going to be updated by multiplying its output and input activation to acquire the weighted gradient.

Repeat this two stage until the gradient weight and mean square error stop decreasing where the performance of the network could be described as adequate. Here is a pseudo code for how the training is done:

Table 1: Backpropagation learning algorithm

```
Input= input data
Hidden= # of node in the hidden layer
Output= the desired output
Squash_Function = sigmoid activation
Input_weights[Input, Hidden]
Output_weights[Output, Hidden]
Propagate [exemplar,Input]
BackPropagate [exemplar,Output]
Random_Weight_Vector(Vector,V)
//Initializing random weights:
    for i = 1: Input
        Random_Weight_Vector(Input_weights[i,:], Hidden)
    end
    for i = 1: Hidden
        Random_Weight_Vector (Output_weights[i,:], Output)
    End
//select a training feature and determining output
Input = Propagate[Rand,:]
//Activation of hidden layer ,   $H_i = \sum_{j=1}^{j} j_j \ w_{ij}$
    for i = 1: Hidden
        for j = 1, Input
            Hidden[i] = Hidden[i] + Input[j] * Input_weights[j,i]
        end
    end
    Apply the Squash_Function
//Activation of output layer
    for h = 1: Output
        for i = 1: Hidden
            Ouput[h] = Output[h] + Hidden[i] * Output_weights[i,h]
        end
    end
    Apply the Squash_Function
//Defined output layer error
    for h = 1, Output
        Diffrence[h] = Backprpogate[Rand,:] – Output[h]
        End
Weights_Update(Output_weights, Diffrence, Output, Hidden)
//Defined Hidden Layer error $\delta_h = H_i(1 - H_i)\sum_{h=1}^{h} w_{hi}\ \delta_h$
                for i = 1: Hidden
                        for h = 1: Output
Output_Error = Output_Error + Output_weights[i,h]*Diffrence[h]
                        end
                        Diffrence[i] = Hidden[i]*(1-Hidden[i])*Output_Error
                end
Weights_Update (Input_weights, Diffrence, Hidden, Input)
... repeat until it reaches the minimum gradient

Return the network
```

Hence, the attack can be determent based on the final mean square error and the regression values where the relationship between the outputs and the targets are measured between 0 and 1. A value of close to 1 is considered as spoofed packets, and a value close to 0 is considered as normal packets.

$$Packets = \begin{cases} Attack, & if \text{ regression} \sim 1 \\ & . \\ Normal, & if \text{ regression} \sim 0 \end{cases}$$

Filtering Module and The Server

The result of every calculated output from the detection engine is gathered. These which have a value of approximated to 1 are sent to the server. Where the server firewall can list these IPs in order to reject any incoming or outgoing connections to these IPs.

CHAPTER FOUR

EXPERIMENTS

This is the part where the proposed solution is tested to check whether it is effective. The system was evaluated by use of network traffic traces of denial of service attacks which were collected from CAIDA and UCLA trace dataset. The evaluation is a three step process that involves: firstly, capturing of packets by use of a Linux-based packet sniffer, secondly calculation of an IP feature table from different sources and thirdly, it involved the use of the training of the neural network algorithm for testing the system.

Attack Dataset

The first Dataset is obtained from University of California, San Diego data center Center for Applied Internet Data Analysis (CAIDA) [20]. This acquired data is a length of one-hour packet capture of a distrusted denial-of-service attack which targeted a server to block its legitimate user from access. It overwhelms the entire network bandwidth. It contains about 360 million sniffed packet which is too large to deal with. Therefore, two million spoofed packets have been extracted randomly to conduct this experiment.

Normal Dataset

The second dataset is obtained from University of California, Los Angeles, Laboratory for Advanced Systems Research [21]. This obtained data is a length of 5 minutes' packet capture of normal use from the border router of UCLA Computer Science Department. It contains around 265 thousand capture packets. To mirror the attack dataset, selected packets within the time window of the attack dataset have been chosen. The following table illustrates the information of the used datasets.

Table 2: Summary of the selected samples

Dataset	Total Time	Total Packets Count	Total unique IP addresses
Attack	15	2000000	5181
Normal	15	10213	312

Also, 20 random samples from the (Feature Extractor Module) are showing in tables 4 and 5 in the appendix.

Dataset Clustering

Here is a demonstration for clustering the two data set. This clustering analysis is performed using the neural network self-organized-map algorithm. The input data of this analysis is given by the features of the feature extractor module. The goal is to split the datasets to an indicated number of clusters so that IPs features that are different from each other are fallen into different clusters. At the same time, we want to see how these data would be distributed over these clusters. To perform this clustering, a number of (3) clusters and (50) epochs are selected with a Kohonen learning weight of (0.1). Also, the total packet length, packets length variance, and packet time variance are selected as the most significant feature to project the dynamic graphs of the clusters. Here are the results of clustering.

For Attack data set, figure 7 demonstrates the elements of the attack data set into the corresponding centroid to the total packet length, packets length variance, and packet time variance. Comparably, figure 8 expresses the cluster analysis for the normal dataset.

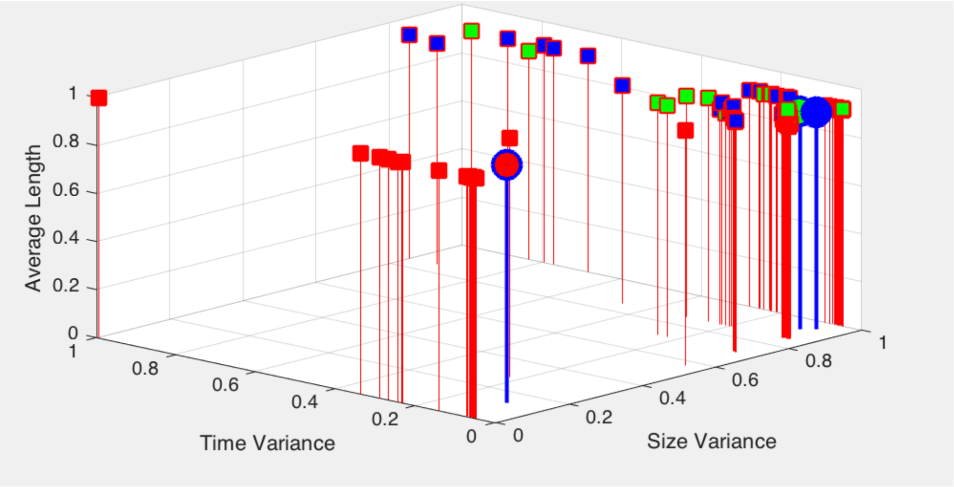

Figure 8: Attack data cluster analysis

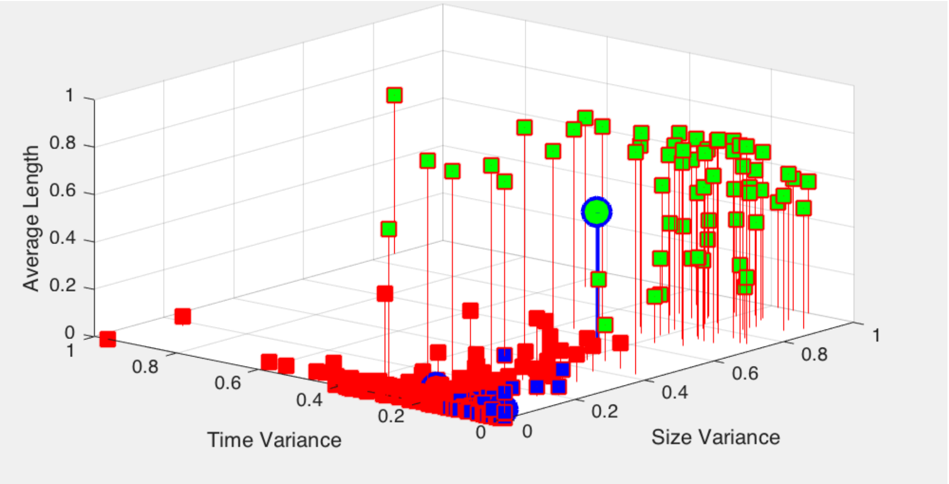

Figure 9: Normal data cluster analysis

The features of the attack data set tend to be distributed consistently over the three clusters, where the features of the normal dataset are irregularly distributed through the cluster. The reason for this difference is that the attack data set leans to the point where it is clear to see that there are repeated patterns for the transmitted packets' lengths and times; while, the normal behavior of the normal packets has a noticeable variation in terms of length and time. The following two tables confirm how the weights of the packets features are distributed over these clusters.

Table 3: Attacks weights distribution over the clusters

Cluste	TPC	Length				Time				Rate
		TPL	APL	PLV	ALD	TPT	APT	ATD	PTV	
1	0.4052	0.4016	0.9903	0.1320	0.0000	0.4052	0.8508	-0.0273	0.0933	0.4016
2	0.4544	0.4105	0.9048	0.9304	-0.7172	0.4558	0.8550	0.0161	0.0896	0.4105
3	0.4404	0.3970	0.9014	0.9491	0.3158	0.4401	0.8503	0.0042	0.0648	0.3970

Table 4: Normal weights distribution over the clusters

Cluste	TPC	Length				Time				Rate
		TPL	APL	PLV	ALD	TPT	APT	ATD	PTV	
1	0.0057	0.0002	0.0279	0.0673	-0.0122	0.0375	0.0800	0.0209	0.2221	0.0002
2	0.0238	0.0157	0.5507	0.5681	0.0425	0.1542	0.0823	0.0439	0.2576	0.0157
3	0.0064	0.0003	0.0106	0.0361	0.0022	0.0137	0.0481	-0.0012	0.0341	0.0003

Detection Engine

After the extracted data arrives at our detection engine it will be selected as inputs. Where, on the other hand, the desired output is fed by the attack data set. A total of random 80% of each the input data will be selected to train the algorithm. The other 20% will be used for the validating of the learning process.

Here is the result of testing the detection engine from randomly selected samples from both the attack and normal dataset.

Table 5: The detection result from attacks and normal samples

Samples	Mean Squared Error	Regression
Attack	0.000653	0.99999
Normal	524	0.254378

The following figure shows the histogram of the mean squared error.

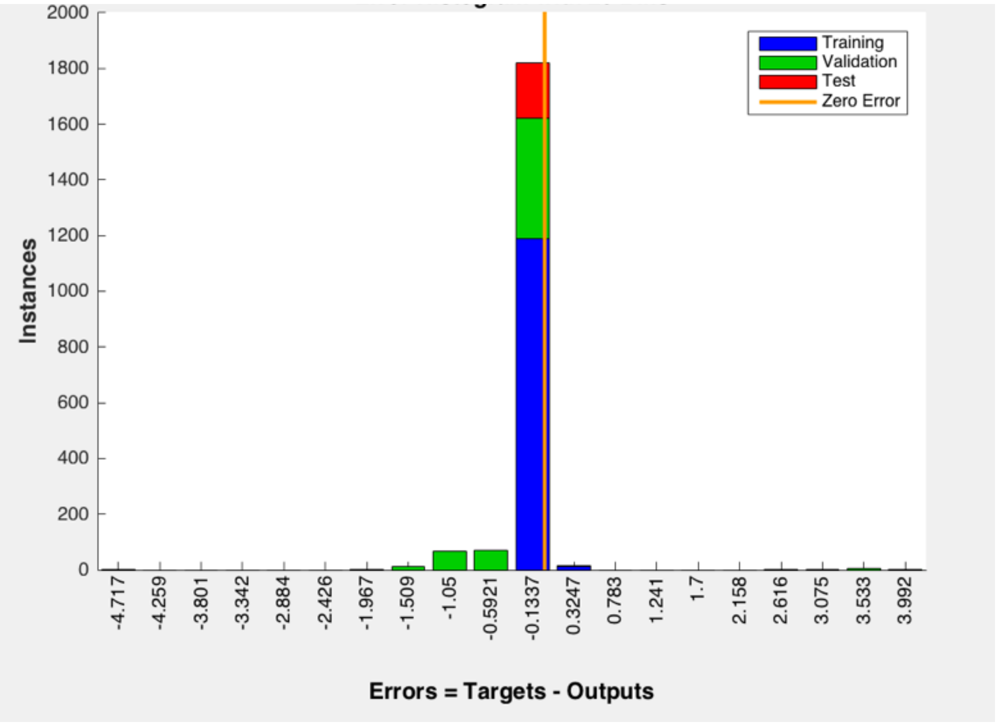

Figure 10: The error histogram of the attack sample training

In training of attack sample, the highest error is 0.324 which indicates 68% that the trained sample is spoofed, then validation sample highest error was 0.01 which indicates 99% of trained sample is spoofed testing highest was below 0 which indicate a 100% of detection.

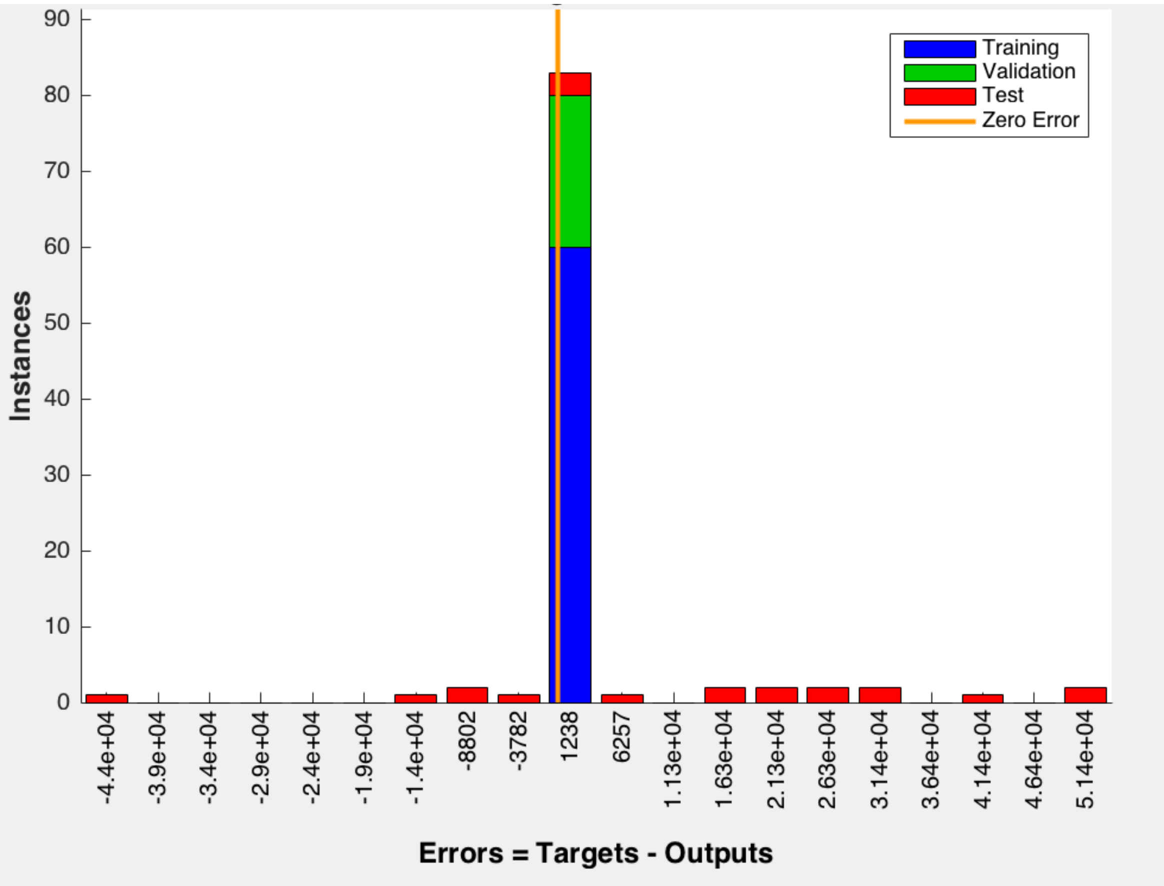

Figure 11: The error histogram of the normal sample training

Here, in the training of normal sample we can see the error is very high which indicate that the packet is not normal.

CHAPTER FIVE
TOOLS USED

The following tools were used to achieve the results of the above experiment:

Matlab

Matlab is a software that can be an integrated development environment for developing solutions that are highly mathematical. Matlab was used to develop computational formulas for the above experiment. Also, the Matlab's neural network toolbox is used to configure the detection engine.

Wireshark

Wireshark is an open source tool for analyzing packets sniffing in a network. It was used to read and to analyze the network traffic of the obtained datasets.

Pentagon Crew DoS Tool

It is a tool designed by members of Pentagon Crew to drive an attack to a targeted machine or a website. I used here to simulations on Mikrotik Router OS

Mikrotik Router OS

It is a router operating system that can be installed on a computer and convert it into a network router. I installed it on one machine to do several attacks experiment

Wincap

Wincap is a C++ programming Interface application that was used extracts the capture packets features. The information about network packets features is stored in a file which was imported by Matlab for further analysis.

CHAPTER SIX

CONCLUSION

Denial of service attacks are a common threat to online technology services such as online banking, music, and video streaming. In today's world where every aspect of our lives are online, especially mission-critical systems such as healthcare systems and hospital management systems, system designers and architects should design such systems with security in mind, particularly on how to create systems with a defensive and protective approach to denial of service attacks.

In this project, an effective detection neural networks method is presented. To this purpose, packets of denial of service attack are investigated and ten features, which show the abnormal variations in the inbound traffic, are extracted. After that, the self-organized-map neural network clustering is applied to classify traffic into normal and attack traffic classes.

The results of the experiment expressed that the designed system is able to distinguish denial of service attacks successfully. Furthermore, the proposed system is able to filter the attack incoming packets quickly. Concurrently, it forwards the normal packets back to the server. It is shown that the proposed system is able to successfully identify DoS attacks with very high detection rates.

APPENDIX

FEATURES EXTRACTED SAMPLES

This table shows the sample extracted attack traffic from CAIDA

No	IP	Packets number	Length				Time				Rate Kb/s
			Total	Average	Variance	Differences	Total	Average	Variance	Differences	
1	'130.119.190.246'	1105	60180	54.46153846	6	-0.010869565	0.0063	0.00567	1.57E-06	-2.30E-24	5.471
2	'133.85.232.42'	1116	60432	54.15053763	7	-0.010762332	0.0062	0.00557	1.36E-06	-2.69E-09	5.494
3	'167.13.39.9'	1181	63792	54.01524132	7	0	0.0067	0.00566	1.39E-06	0	5.799
4	'167.15.184.229'	1172	63192	53.91808874	7	0	0.0067	0.00573	1.50E-06	-2.56E-09	5.745
5	'167.226.237.157'	1189	64296	54.07569386	7	-0.01010101	0.0067	0.00560	1.47E-06	-8.42E-10	5.845
6	'192.1.206.217'	1131	61020	53.95225464	7	-0.010619469	0.0064	0.00565	1.35E-06	8.85E-10	5.547
7	'192.153.200.55'	1223	66288	54.20114473	6	-0.009819967	0.0069	0.00567	1.38E-06	8.18E-10	6.026
8	'192.229.242.166'	1216	65796	54.10855263	7	0	0.0069	0.00570	2.09E-06	-8.23E-10	5.981
9	'192.5.232.167'	1191	64368	54.04534005	7	-0.010084034	0.0067	0.00560	1.27E-06	8.40E-10	5.852
10	'192.87.52.192'	1173	63348	54.00511509	7	0	0.0068	0.00581	5.00E-06	8.53E-10	5.759
11	'193.86.91.118'	1117	60228	53.91942704	7	0	0.0064	0.00574	1.42E-06	8.96E-10	5.475
12	'195.109.215.57'	1119	67140	60	0	0	0.0063	0.00560	1.28E-06	-1.79E-09	6.104
13	'195.142.7.233'	1119	60672	54.21983914	6	-0.010733453	0.0063	0.00563	1.65E-06	1.79E-09	5.516
14	'195.58.5.16'	1177	63480	53.93372982	7	0.010204082	0.0068	0.00582	5.74E-06	2.88E-24	5.771
15	'195.88.16.202'	1125	60540	53.81333333	6	0	0.0064	0.00567	1.42E-06	8.90E-10	5.504
16	'196.208.28.12'	1227	66516	54.21026895	6	-0.009787928	0.0070	0.00570	1.57E-06	-1.63E-09	6.047
17	'196.216.146.135'	1120	60636	54.13928571	7	-0.010723861	0.0063	0.00563	1.48E-06	3.57E-09	5.512
18	'197.111.128.138'	1276	76560	60	0	0	0.0073	0.00572	1.97E-06	7.84E-10	6.960
19	'197.208.124.44'	1145	61956	54.11004367	7	0.01048951	0.0065	0.00570	1.56E-06	1.75E-09	5.632
20	'199.132.187.180'	1173	60896	51.91474851	7	-0.011945392	0.0074	0.00567	2.26E-05	-8.53E-10	5.536

This table shows a sample capture of normal traffic from UCLA

No	IP	Packets number	Length				Time				Rate Kb/s
			Total	Average	Variance	Differences	Total	Average	Variance	Differences	
1	'1.1.4.4'	411	102986	250.5742092	529	0.00000	0.315433	0.00077	0.00190	-0.00512	9.362
2	'1.1.54.59'	91	45780	503.0769231	550	0.88889	0.084899	0.00093	0.00124	-0.00971	4.162
3	'1.1.8.18'	445	139100	312.5842697	597	-3.26126	0.082545	0.00019	0.00067	-0.00037	12.645
4	'1.1.8.8'	884	50110	56.68552036	200	0.00000	0.286517	0.00032	0.00095	-0.00029	4.555
5	'10.81.136.191'	183	224769	1228.245902	481	-0.48352	0.492978	0.00269	0.00334	0.01067	20.434
6	'11.13.10.17'	115	142071	1235.4	515	12.52632	0.339105	0.00295	0.00305	-0.06775	12.916
7	'12.126.91.40'	31	38174	1231.419355	522	0.00000	0.033965	0.00110	0.00223	-0.01567	3.470
8	'12.134.96.131'	101	138791	1374.168317	335	14.60000	0.171991	0.00170	0.00295	-0.02750	12.617
9	'12.170.107.86'	39	40141	1029.25641	545	31.42105	0.027551	0.00071	0.00086	-0.00279	3.649
10	'17.157.244.84'	210	286874	1366.066667	255	6.98086	0.24527	0.00117	0.00151	0.00547	26.079
11	'22.39.46.51'	42	61156	1456.095238	26	-4.00000	0.085123	0.00203	0.00199	0.06941	5.560
12	'24.46.113.55'	42	48241	1148.595238	587	1.00000	0.023976	0.00057	0.00083	0.00954	4.386
13	'25.128.5.28'	40	45391	1134.775	572	0.00000	0.074859	0.00187	0.00200	-0.06118	4.126
14	'33.166.210.99'	50	50529	1010.58	632	3.59184	0.036784	0.00074	0.00107	-0.01331	4.594
15	'39.199.240.67'	70	41372	591.0285714	662	0.00000	0.111777	0.00160	0.00215	-0.02378	3.761
16	'51.78.58.59'	66	37315	565.3787879	632	0.00000	0.135015	0.00205	0.00274	-0.14982	3.392
17	'58.129.52.170'	100	46298	462.98	384	3.87879	0.099544	0.00100	0.00179	-0.01146	4.209
18	'60.25.89.5'	336	458752	1365.333333	135	-0.84776	0.300233	0.00089	0.00166	0.00037	41.705
19	'24.46.253.113'	33	35835	1085.909091	535	-31.09375	0.032681	0.00099	0.00146	-0.17159	3.258
20	'7.11.107.16'	29	34895	1203.275862	505	52.14286	0.019088	0.00066	0.00114	0.15496	3.172

REFERENCES

[1] S. T. Zargar, J. Joshi and D. Tipper, "A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks," *IEEE Communications Surveys & Tutorials,* vol. 15, no. 4, pp. 2046-2069, 2013.

[2] B. Santhi and G. J. Bharathi, "Study on Distributed Denial-of-Service Attack," *Research Journal of Applied Sciences,* vol. 4, no. 10, pp. 1366-1370, 2012.

[3] M. H. Bhuyan, H. J. Kashyap, D. K. Bhattacharyya and J. K. Kalita, "Detecting Distributed Denial of Service Attacks: Methods, Tools and Future Directions," *The Computer Journal,* vol. 57, no. 4, p. 537, 2014.

[4] R. Chang, "Defending against flooding-based distributed denial-of-service attacks: A tutorial," *IEEE Communications Magazine,* vol. 40, no. 10, pp. 42-51, 18 April 2012.

[5] Radware Ltd, DDOs attack Survival handbook, Chicago: Radware, 2013.

[6] J. MirkovicJ and P. Reiher, "D-WARD: a source-end defense against flooding denial-of-service attacks," *IEEE Transactions on Dependable and Secure Computing,* vol. 2, no. 3, pp. 216-232, 23 July-Sept 2005.

[7] Y. Xie and S. Z. Yu, "Monitoring the Application-layer DDoS Attacks for Popular Websites," vol. 17, no. 1, pp. 15-25, 2009.

[8] Z. Wu, G. Li, M. Yue and H. Zeng, "DDoS: Flood Vs. Shrew," *Journal of Computers,* vol. 9, no. 6, pp. 1426-1435, 3 January 2014.

[9] G. Somani, M. Gaur and D. Sanghi, "DDoS/EDoS Attack in Cloud: Affecting Everyone out There," in *Proceedings of the 8th International Conference on Security of Information and Networks*, Chicago, 2015.

[10] S. B. Ankali and D. D. V. Ashoka, "Detection Architecture of Application Layer DDoS Attack for Internet," *International Journal of Advanced Networking and Applications,* vol. 3, no. 1, pp. 984-990, 2011.

[11] S. Asri and B. Pranggono, "Impact of Distributed Denial-of-Service Attack on Advanced Metering Infrastructure," vol. 83, no. 3, pp. 2211-2223, 24 February 2015.

[12] T. Peng, C. Leckie and K. Ramamohanarao, "Survey of Network-based Defense Mechanisms Countering the DoS and DDoS Problems," *ACM Computing Surveys (CSUR),* vol. 39, no. 1, 2007.

[13] F. Wong and C. X. Tan, "A Survey of Trends in Massive DDOS Attacks and Cloud-Based Mitigations," *International Journal of Network Security & Its Applications,* vol. 6, no. 3, pp. 57-71, 2014.

[14] R. Sadre, A. Sperotto and A. Pras, "The Effects of DDoS Attacks on Flow Monitoring Applications," in *IEEE Network Operations and Management Symposium*, 269-277.

[15] C. Douligeris and A. Mitrokotsa, "DDoS attacks and defense mechanisms: classification and state-of-the-art," *Computer networks,* vol. 44, no. 5, pp. 643-666, 2004.

[16] R. Anurekha, K. Duraiswamy, A.Viswanathan, V. P. Arunachalam, A. R. Kannan and K. G. Kumar, "A Dynamic Approach to Defend Against Anonymous DDoS Flooding Attacks," *International Journal of Computer Science and Information Security,* vol. 8, no. 7, pp. 279-284, 2010.

[17] Reyhaneh Karimazad , "An Anomaly-Based Method for DDoS Attacks Detection using RBF," in *2011 International Conference on Network and Electronics Engineering* , Singapore, 2011.

[18] W. Bhaya and M. E. Manaa, "Review Clustering Mechanisms of Distributed Denial of Service Attacks," *Journal of Computer Science,* vol. 10, no. 10, pp. 2037-2046 , 2014.

[19] D. Wackerly, W. Mendenhall and R. L. Scheaffer, Mathematical Statistics with Applications, 7th Edition ed., Belmont, CA: Thomson Higher Education, 2008.

[20] "The CAIDA "DDoS Attack 2007" Dataset," University of California, San Diego, 4 August 2007. [Online]. Available: http://www.caida.org/data/passive/ddos-20070804_dataset.xml. [Accessed 12 April 2016].

[21] "UCLA CSD Packet Traces," University of California, Los Angeles, August 2001. [Online]. Available: http://www.lasr.cs.ucla.edu/ddos/traces/. [Accessed 28 February 2016].