Network Intrusion Prevention System Based on Enhanced Snort Rules to Protect Network Resources from DoS & DDoS Attacks (An Empirical Approach)

G.R.C. Kumara

139193P

Dissertation submitted to the Faculty of Information Technology,
The University of Moratuwa, Sri Lanka for the partial fulfilment of the requirements of the Degree of Master of Science in Information Technology.

March 2021

Declaration

I declare that this dissertation is my work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Name of Student

Signature of Student (s)

Date:

Supervised by

Name of Supervisor (s)

Signature of Supervisor (s)

Date:

Dedicated to

my parents, my teachers

Acknowledgement

I would like to thank and gratitude the individuals for their contribution and motivation in various ways to make this research project a success.

First of all, I would like to express my gratitude to Dr C.R.J. Amalraj, my Supervisor, for his guidance, encouragement, tact, guidance, mature and intelligent advice, suggestions, positive thoughts and optimistic encouragement to continue to make this project a success. The supervisor also helped me to succeed in my future career as well as academics by encouraging me to overcome difficulties. I was lucky enough to do this research project under his supervision.

As well I am deeply grateful to senior lecturers Mr Saminda C. Premarathna and Mr Chaman P. Wijesiriwardhana for their support and continuous help to succeed in this endeavour.

Further, I would like to offer my special thanks to the senior lecturer of the faculty Mr B.H. Sudantha for his continuous support of the success of the project work.

I gratefully acknowledge Dr Lochandaka Ranatunga for initiating a research project and giving a more effective start on the path to success.

Again, I am also in debt too to the evaluators of the interim presentation of this project, Mr D.K. Withanage, Mr S.C. Premarathne, and Mr B.H. Sudantha for their valuable advice and comments to improve my work.

I am indebted to Mr D.K. Withanage, Mr S.C. Premaratne and Mr B.H. Sudantha, the Interim Project Evaluators, for their advice and suggestions on how to drive the research a success.

Also, I am once again indebted to the final evaluator's Dr C.R.J Amalraj, Mr B.H Sudantha and Dr T. G. I. Fernando of the project for their valuable advice in making the dissertation successful and correcting the shortcomings of the research project.

I would like to offer my special thanks to Professor Asoka Karunanada who taught us the "Research Methodology" subject for this master program on behalf of his guidance and motivations to research thinking and handling.

Furthermore, I would like to extend my sincere thanks to other academic staff, academic support and none academic staff members, those who support me in many ways in the IT faculty, UoM.

Again, I should grateful for Prof. Asoka Karunananda our teacher in "*Literature review & Thesis Writing"* who guided us for better researching and dissertation writing.

I would specially thank Mr Saminda Premaratne the course coordinator, MScIT, UoM for his encouragement and advice on this project.

Furthermore, I would like to extend my sincere thanks to my former director-general Prof. Kapila C.K. Perera who directed me to this M.Sc. program in the Faculty of IT, University of Moratuwa and my present director-general Dr W. Hilary E. Silva for always encouraging academic staff towards the research culture in Sri Lanka Institute of Advanced Technological Education (SLIATE). Further, I would be thankful to the director -Eng. Mr N.K.A. Rupasinghe, staff members and students in my institute, ATI Galle for being patient on my busy schedule. In addition, I would like to specially acknowledge the financial assistance offered by the HETC Project, Ministry of Higher Education.

I am grateful to my wife Manoja for her unwavering support in this endeavour and her patience and support and for the loving son Manula, who lost my attention and love during this period. Moreover, I would like to thank my friends and colleagues in the master program who encouraged me with their ideas and experiences to complete the project.

Finally, I would like to express my gratitude to the authors of the most important scholarly works that have been adopted for this research project. Here, Prof Buchanan B., Flandrin F., Richard Macfarlane, Dr Jamie Graves from Edinburgh Napier University are majors for utilizing their methodology to evaluate rate-based IPS in this research work and I am much graceful to them for their contribution to academia.

Finally, in the face of this scientifically significant research project and dissertation, I thank and thank all the gods who have given me unwavering courage, strength, and guidance.

Abstract

Virtual computers from anywhere in the world are designed to enable any user to access the computer resources contained in the cloud computing (CC) environment. The flour resources in the cloud environment, which pose a great threat to security, are shared so that they can be accessed by users anywhere in the world. Denial of service and denial of distributed services is one of the leading challenges posed by attackers who pose a serious threat to CC's security. Next Generation Invasion Prevention Systems (NGIPS), also known as non-traditional invasion prevention systems or next-generation invasion prevention systems, is being introduced as a tactic to overcome these offensive challenges. Thus, the author intends to find research results on the technological strategies used in NIPS and their evaluation and to find solutions to the challenges of cloud computing (CC).

The author has used a very systematic literature review to explore and identify the latest NIPS techniques using Inspec, IEEE, ACM Digital Library, Wiley, Scopus and Google scholar library databases. Articles are selected based on the acceptance and rejection formula when selecting for literature review. This experimental methodology has been selected as a research methodology for experimental comparison of the source and destination approaches of Snort NIGPS. The experimental bed is designed and deployed using Snort filtering techniques deployed in a virtual machine through a virtual switch in a virtual environment.

In this research, the author involves in finding the answers to the research problems and the first problem was the use of next-generation IPS technologies to protect the cloud computing surrounding from DoS and DDoS attacks. The second and third research questions are identifying different types of measurements to assess the performance of NIPS, and the third is the find the performance skill among both source and destination approaches of Snort Intrusion prevention systems.

Network engineers, network administrators and academia has been considered as users of the research scope. The hypothesis in this research is the filter will never work if the attack is launched on a large number of source IP addresses, and Snort will not be able to distinguish between legal and non-legal packets, even if the filter is functional. Finally, the null hypothesis used is the Snort does not show any difference between both source and destination approaches.

TCP, UDP, HTTP and mixture of the protocols used as attack input using LOIC attack tool and legitimate traffic inputs to the system generated using JMeter tool and Further, TCPreplay has been used to regenerate the same amount of both attack and normal traffic to maintain the justification and all those considered as independent variables. The dependent variables considered as the output of the research results are load on the CPU results, Utilization of the memory, availability of the Bandwidth, Delay (Latency), percentage (rate) of loss of packets.

In this research, the processes are the generating of both normal and attack traffics, detecting and preventing malicious traffic using Snort rate filtering rules. In the source approach, packet-based identification and filtering of packets are done by scanning the source IP address and enabling Snort to activate the filter if a specific packet rate is reached. Destination detection and filtering of packets are done by ordering Snort to swipe packets to the destination IP address and to enable filtering when a predetermined packet rate is reached.

NIPS strategic algorithms can be evaluated using classical metrics such as Load of the CPU, Utilization of Memory, Bandwidth availability, Delay (Latency), Rate of packet loss (both “false positives” and “false negatives”) and Accuracy. This experiment also found that when accessing parameters such as Load of the CPU, Utilization of Memory, Bandwidth availability, Delay (Latency) and Rate of packet loss, destination access was more efficient than source access. That is, the filtration rate system of the destination approach is more efficient.

Most of the NIPS technologies used in the Cloud Computing environment to protect from DoS and DDoS attacks are concluded as similar and interrelated. Furthermore, the author concludes that there is a difference in performance appraisal in the cloud computing environment between Snort's source and destination approaches.

Table of Contents

List of Figures

List of Tables

ACRONYMS

CC : Cloud Computing

PaaS : Platform as a service

SaaS : Software as a Service

IaaS : Infrastructure as a Service

DoS : Denial of Service

DDoS : Distributed Denial of Service

IPS : Intrusion Prevention Systems

IDPS : Intrusion Detection Prevention Systems

NGIPS : Non-Traditional Intrusion Prevention Systems

TIPS : Traditional Intrusion Prevention Systems

CSP : Cloud service providers

DC : Data Center

MITM : Man in The Middle Attack

NGIPS : Next-Generation Intrusion Prevention Systems

ICMP : Internet Control Message Protocol

TCP : Transmission Control Protocol

UDP : User Datagram Protocol

HTTP : Hypertext Transfer Protocol

LOIC : Low Orbit Ion Canon

GUI : Graphical User Interface

RTT : Round Trip Time

PRS : Packet Resonance Strategy

Chapter 01

Introduction

Introduction to Performance Evaluation of IPS based on Enhanced Snort Rules to Protect Network Resources from DoS & DDoS Attacks in Virtual Environment.

1.1. Prolegomena

A good network security system helps businesses reduce the risk of falling victim to data theft and sabotage. Network security helps protect your workstations from harmful spyware. It also ensures that shared data is kept secure.

Cyber security is a challenging topic and has different dimensions of the cyber security scope. Every network is generally, open to the victim of data theft and or sabotage. Therefore, the protection of network resources and critical information in an organization, from cyber-attacks is a crucial and essential matter.

Performing the below requirements over the Internet, through a remote computer network, instead of using local resources, is called CC and belongs to a third party. The ultimate goal of CC is to deal with data in a highly secure way and keep protecting privacy. The work with data can be divided into several main parts, namely, data storage, processing and management in a very secure manner. CC is a result of a combination of virtual computer technology and network computer technology. This allows users to allocate resources as they require, and configure the machines for data application with minimal effort from anywhere in the world.

CC provides customers with a more flexible effective platform through different architectures in a more cost-effective way [1]. In this architecture, there are two models of functioning named service and deployment. The forms of deployment models are public, private, hybrid and community clouds. PaaS, SaaS, and LaaS are different service architecture models [2]. Through the above deployment and service model architectures, the CC provides customers with friendly and cost-effective benefits for their business endeavours.

CC's primary objective is to ensure the security required to share the service provided with a third party. DoS and DDoS are the main attacking technology categories that threaten the security of CCs [3]. In DoS attacks, a single network is used, and in DDoS attacks, multiple networks are used to attack the end target. A common feature of these attacks is that they adversely affect the bandwidth or resources of the target system [4]. To do this, a zombie computer uses a large number of fake data packets to target the infected system. In these two types of attack threats, a single zombie computer or a zombie computer system get used to sending fake data packets, respectively, over networks around the world [4].

Cloud service providers, confirm that the security of cloud computing systems by deploying traditional firewall technology with intrusion detection and prevention systems (IDPS) to prevent attacks. The combination of both intrusion detection systems (IDS) and intrusion prevention systems (IPS) gives Intrusion Detection and Prevention Systems (IDPs). IDS can detect invasive and attacks but does not have sufficient capacity to prevent them by preventing them from accessing the target system. The researcher, (Kemmerer and Vigna, 2002), shows that the term intrusion detection system is somewhat misleading since they cannot detect attacks and can only show witnesses [5].

By minimizing the impact of attacks on the cloud computing system and preventing invasions from entering the DC, the IPS algorithm can further support the task of detecting attacks. According to researcher Piper (2011), the IPS has been divided into two categories called traditional and non-traditional based on their capabilities [6]. These non-traditional intrusion prevention systems (IPS) lead to the next generation of defence prevention systems (NGIPS). This will take IPS technology in a new direction and lead to an evolution of IPS technology.

This research explores existing NIPS technologies, improves them according to innovative strategies, and finds appropriate dimensions to evaluate and compare system viability. To prevent DoS and DDoS, an experimental comparison is made using the results provided by the two methods in Snort IPS. This uses a methodology based on performance parameters. The performance metrics are selected based on the availability of the resources and usage size of the systems.

1.2. Problem Definition

The main problem of DCs is to provide protection for the resources and to allow transactions to take place while maintaining privacy within the time limits set by the client's needs [1] [7]. Protecting the system and its processes is one of their most important requirement [8]. Major security challenges include data security, network security, and resource access. CC's server computers are designed to maintain a high level of the computational efficiency of the system in proportion to the increasing customer demand [9] [10]. However, there are limits to the number of customer requests that can be submitted in a single period. Once this limit is reached, the server system will not be able to respond to its client. Attackers use this phenomenon to launch DoS and DDoS attacks.

According to their research in Subashini and Kavitha (2011), eleven companies within four high-risk commercial companies of their own country have fallen victim to these DDoS attacks and are in dire straits, unable to provide services to their clients [11].

Attackers can launch DDoS attacks to the Amazon Elastic Cloud Computing (EC2) [12] as there are several technical issues on controlling the services. Therefore the attackers take the opportunity of operating eavesdrop messages. Therefore the operating of digitally signed messages are not take place. Utilizing this effect, attackers launch DDoS attacks on EC2 to increase the cost of their clients' bills.

Because DoS and DDoS generation tools are so common and easily manipulated, the attacker can be activated by using new advanced tactics, limiting the victim's resource use and limiting bandwidth [9] [13]. Restricting bandwidth is the main goal of DOS attackers to restrict or prevent users from accessing the system. Therefore, protecting CC resources and protecting the customer is a matter of urgency today.

There are numerous traditional and next-generation approaches to form an effective IPS on both the user side and server-side against DoS and DDoS attacks which are based on an empirical comparison based on selected Metrics for the performance evaluation of such IPSs [7] [11]. According to the research studies of Patel et. al [14] and Carlin et. al [11], today, modern IPSs are needed to protect DCs from advanced attacks, as traditional TIPS are unable to detect and prevent advanced attack tactics designed to destroy networks.

Searching through the various databases, the author searched the literature for protection against modern DoS and DDoS attackers, but for the next generation of IPS (NIPS), the amount of research done was very limited, as evidenced by the author's search using the following algorithm.

(("Data center" OR "Datacenter" OR "virtual Environment" OR "Cloud Environment") AND ("Intrusion Detection" OR "Stress Testing" OR "Cyber Attack" OR "Cyber Security") AND ("Intrusion Prevention" OR Protection) AND ("New generation" OR "Novel Attack" OR modernize*) AND ($experimental OR observation* OR practical OR empiric*) AND (analyz* OR measur*)).

Therefore, further research is a must for the empirical evaluation of the performance of new generation IPS and the said study guides to solve the research problem crafted by the author.

Since a single IPS technology system capable of preventing all protocol-generated DoS and DDoS attacks cannot be found among existing NIPS tactics [15], research on modern NIPS is essential to overcome these intrusions in the cloud environment.

1.3. Aim & Objective

The purpose of this section is to present the objectives underlying the objective problem of this research. The main purpose of this research is to empirically compare two DoS and DDoS attack technologies based on the availability of relevant computer resources and to discover current NIPS technologies. The target objectives proposed by the author are explained as below:

- **Objective 1:**

  Identifying the next generation IPS technologies currently used in a virtual computing environment in the Data Centre(DC)s.

  Motivation: This gives the author a better idea of modern NIPS strategies being developed across various attack battle fronts.

- **Objective 2:**

  Identify appropriate parameters to assess the feasibility of attack techniques and their prevention strategies.

  Motivation: As a result, by finding the parameters, some research problems can be easily answered toward assessing the new IPS through the selected metrics.

- **Objective 3:**

  To evaluate the strategies needed to experimentally evaluate the two approaches of source approach and destination approach that are compared for evaluation.

  Motivation: Therefore, the selected Snort IPS helps to select the appropriate approach from the two evaluated approaches.

1.4. Brief Introduction of Solution

To solve the security issues raised by the DoS and DDoS attacks in the CC environment, the author uses enhanced Snort rules to identify and preventing from the attacks. In this research, the author uses a virtual machine environment to launch the experiment and collect data for performance evaluation. This is empirical research.

1.5. Construction of the Dissertation

The remaining chapters and the parts of the dissertation are structured in this section. Chapter 2 is on critical review on the area in the study of IDS, IPS and SNORT based IDS/IPS. Chapter 3 presented technology adapted towards an efficient intrusion detection and prevention system based on SNORT. Chapter 4 provides the overall picture of our novel approach to design the most secure intrusion detection and prevention system. Chapter 5 discussed the design of the solution. Chapter 6 is about the implementation of the SNORT IDS/IPS. Chapter 7 elaborate the evaluation of the proposed solution. Chapter 8 concludes the results analysed in the evaluation with a note for further work.

1.6. Summary

This chapter gave an overview of the dissertation. The importance, background and motivation of this project have been discussed initially. Studies were done in this domain also presented briefly. Problem definition, Objectives, Hypothesis, and Resources required to perform this research project were also expressed. Finally, the structure of the dissertation is described chapter by chapter. The next chapter provides details regarding the developments in Performance Evaluation of IPS based on Enhanced Snort Rules to Protect Network Resources from DoS & DDoS Attacks in Virtual Environment.

Chapter 02

Review of others' work

The Performance Enhancement and Current Issues in Datacenter Security.

2.1. **Introduction**

This chapter elaborates what is done in for a review of the literature, and the format used for this process is described in detail with several sections as in below:

- Section 2.1: Fundamental Introduction to the Literature Review
- Section 2.2: The background of the research.
- Section 2.3: A comprehensive introduction to the launch of the Literary Review is presented under this category.
- Section 2.4: This section describes the formulated solutions to the research questions.
- Section 2.5: This section describes the study of how research can be planned and carried forward.
- Section 2.6: A synthesis of the essence extracted from the literature review discussed in this section.

2.2. Background

2.2.1. Introduction to the Background Information

In this stage, mainly, a brief study of the Data Centers, Cloud Computing, Virtualization, Snort and its rules, DoS and DDoS attacks, IDS and IPS systems, criteria for evaluating those systems, and methods that can be used to plan this type of research is conducted. The organization of this sub section describe as below:

- **2.2.1.1:** Discusses issues affecting the security of computers and related resources in a virtual computer environment in data centres.
- **2.2.1.2:** This section covers a study of DoS and DDoS attack generation tools, and underlying protocols.

- **2.2.1.3:** This section provides a brief idea about the IPS systems and their classification. This section gives brief elaboration on intrusion prevention systems.

2.2.1.1. Virtual computing environment in data centres (DC) and the security issues.

This section provides security issues and solutions, as well as detailed definitions and information on the design of CC and virtual computing environments.

2.2.1.1.1. Cloud Computing

According to Investopedia, Cloud computing can be defined as "*Cloud computing is the delivery of different services through the Internet. These resources include tools and applications like data storage, servers, databases, networking, and software."* [16]. The structure of the cloud environment described by the author, which is described by this definition, can be seen in Figure 1, and by this definition, the desired amount of valuable computer resources are available to the clients as their requirement.

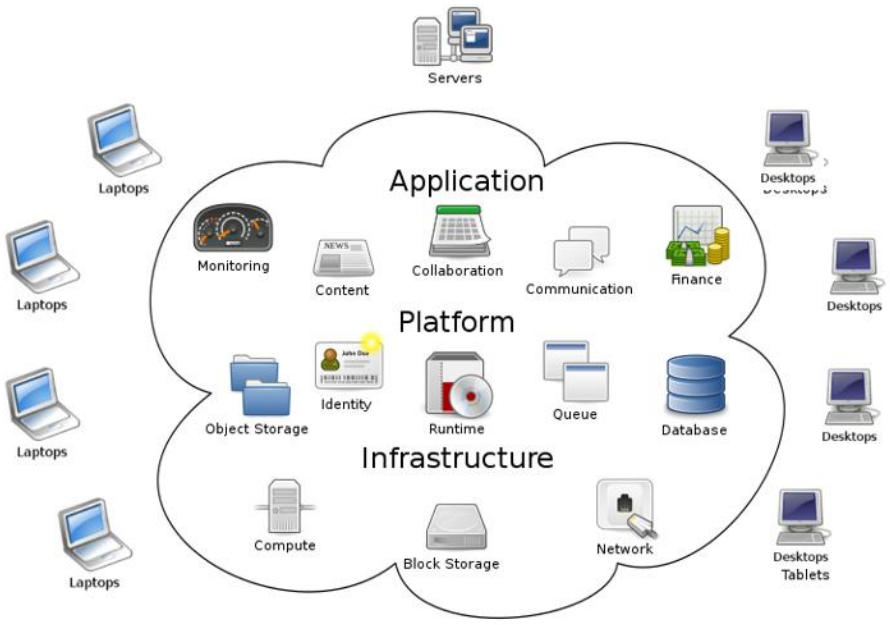

Figure 1: Top-level view of Virtual computing environment in DC.

The main objective of CC is to enable the customers to use computer resources with configured features according to their requirements over the network for the desired time at a less management cost.
In Li et al. [17], according to the author, Virtual Computing services are provided from time to time under dynamic conditions, depending on the needs of the user. Furthermore, Li et al. [18] focus on many CC-based computer models and further explain how these models can be described through the virtual computing service.

2.2.1.1.2. The Active models of cloud computing

According to researchers at [1] [18] [19], there are two active models of CC architecture: inputs and services. That is, there are three service formats, SaaS, PaaS and IaaS, respectively, offering to the customer's software as a service, a platform as a service and an infrastructure as a service. The first category provides software usage, the second category provides a software development environment, and the third category provides infrastructure such as networks, devices, and operating systems.

According to researchers [1] [18] studies, there are four types of cloud input environments: private, community, public, and hybrid, where the cloud is driven by a third party organization or group of organizations, and the second by a community organization. The fourth type consists of two, three, or four of the above input forms, either by or through a collection of organizations, or by public, private, or academic organization or a consortium that serves the general public in the third category.

There are five distinctive features of the CC model [19]. These include automatic cloud access, access to the cloud over a wide range of devices, transfer and reassignment of resources through a cloud resource aggregation, resource allocation or removal (resilience) as required, and billing as required by the CC service provider, Optimum Measurements, Access Control Measurements, Task Management Measurements.

2.2.1.1.3. Security in the cloud computing aspect

The main feature of cloud computing service providers is the provision of large amounts of virtual computing resources to the users through networking technology on a lease basis online. This raises a wide range of security issues, including confidentiality, privacy, and data leakage [19] [20]. Attacker invaders target and operate on these key security issues. According to researchers Alosami and Al-Begain [21], there are four CC attack classifications.

- **Hazard to the Institutional and Policy-based:** The threat that arises in this classification are conformity hazard, end of service threat, failure from control and failures in portability.
- **Issues relating to the legal aspect:** The issues underlying this classification are Contracts, locations of the Data, erasing of data, and agreements at the service level.
- **Uncertainty regarding the Physical Access:** The issues in this classification causes due to human hazards such as pursuit with penetration, staff activity like switch-off of the air conditioning machines, spreading liquids over the surfaces or natural hazards such as flooding, earth quakes, lightning, corrosion.

- **Failures in Technological:** The issues in this classification are laid under confidentiality, integrity and authenticity. Further, as a result of these issues, problems with encryption, web security, network attack like DoS and DDoS, port scanning, the man in the middle attack and vulnerabilities to the visualization scope can causes.

To provide a more effective service to their clients, security strategists must deploy IPS systems in a cloud environment, if not in a virtual environment, to protect against the above risks [22].

2.2.1.2. DoS and DDoS Attacks

The bandwidth and other resources on the victim's side effect badly as DoS (Denial of Service) attackers send massive futile packets toward the target machine using a zombie machine [4] launcher connected to the internet. Using more than one zombie machine over the Internet, the DDoS attacks known as botnets are launched toward their target location over the globe as similar to DoS attacks [4].

According to the researchers in [9] [15], zombie machines use fake IP addresses to launch attacks, it is very difficult to find the primary computer or computers. Further, the attacker machines scan for weak networks and weak machines, and then generate attacks, using those detected machines as agents towards the target. Such machines consider as zombie machines or botnets, and such systems can put a massive amount of pressure on the target, causing the memory, processor, bandwidth, memory buffer, and file details to deactivate and then demolish the service of CC.

2.2.1.2.1. Classes of DoS and DDoS Attacks

The characteristics of the various DoS and DDoS attackers in the cloud environment are described below, according to the researchers [1] [7] [13] [22] [23] [24] [25].

- **UDP flood attack:** What is being done here is to divert the victim's attention to other unwanted false messages, preventing their legitimate users from accessing the victim machine. The attacker continuously sends user database protocol (UDP) messages with fake return addresses, drastically changing the focus of the fake victim machine. Here, without a destination, the victim machine sends the answer via ICMP packets with no destination. Hence, the legitimate customers cannot gain access to the victim server.
- **ICMP flood attack:** This is very similar to the UDP flood attacks mentioned above. The victim sends a large number of ping packets (Echo) continuously to the resources on the server and sends the response to their clients. In this case, both the bandwidth going out of the victim and the bandwidth reaching out to the victim is completely blocked and then the communication line is completely blocked.

- **Protocol Attacks:** Protocol attacks affect the victim's system as well as component systems such as load balancers and firewalls that act as intermediaries between accesses to the victim. Intensity is measured in terms of the number of packets fired per second, and in general, Smurf DDoS, Fractured Packet Attacks, and SYN Flood Attacks fall into this category.
- **SYN flood attack:** This is done using a fake IP address, sending TCP / SYN packets to the victim, and then sending the SYN _ACK message generated during the TCP / IP three-way handshake to the aforementioned fake IP address and waiting for a response. But because it was a fake IP address, there was no point in waiting for a response. That is, instead of considering client requests, it looks at the half-open TCP / SYN - SYN _ACK connection. Therefore, the client will not be able to reach the server further more.
- **HTTP flood:** The attacker blocks and attacks the HTTP GET or POST application associated with the web server or web application, blocking it. That is, it attacks the client by exploiting legitimate HTTP requests, using a very small bandwidth, without resorting to packet fraud, error packet deployment, or the use of reflection methods. Thus, in response to all requests, the attacker is subjected to a very high level of offensive status.
- **TCP flood:** In this type of intrusion, tricky return addresses are used by the Transmission control protocol (TCP) to send messages toward the target destination. As a consequence of this enormous tricky flooding of UDP packets, the server goes continuously listen to the relevant application through an opened port repeatedly which leads legitimate users to prevent access to the server.

2.2.1.2.2. Application Used to Generate DoS And DDoS

Using a variety of applications, hackers can launch DoS and DDoS attacks over the Internet and block the cloud server. Below is a summary of some of the popular applications used by attackers for this purpose [15].

- **Trinoo**: Wintrinoo is the version for the Windows operating system that can be used to launch remotely exploitative attacks through services such as RPC buffer override attacks, using a variety of systems across the Internet. The Trinoo application has the capability of operating a massive number of control machines to launch an attack at many locations over the world through the Internet. In the mechanism of Trinoo attacks, the two communications called master to slave and slave to master happens based on the protocols TCP and UDP. The intention of this attack is to launch the UPD flooding to the service provider [25].
- **TFN:** Through this application, diverse attacks can launch SYN flood, Smurf, ICMP flood and UDP flood by replying with ICMP echo packets among the master and slave machines [25].
- **TFN2K:** The TCP, UDP and ICMP protocols uses in this application to launch attacks between master, slave and the attacker. In addition, to make vulnerable the systems by sending managed packets, this application can generate flooding attacks of SYN, ICMO, UDO and Smurf toward the service provider side. The CAST-256 algorithm is used for the encryption of communication between the attacker and the master.
- **Stacheldraht:** The important advantage of this application is the ability to automatically upgrade the slave machines and this is a hybrid product of Trino and TFN applications. The ability to deploy the flooding attacks of ICMP, UDP, Smurf remarkable point. An encryption TCP connection mechanism uses based on TCP and UDP protocols for the master and slave communication in this attack and it provides additional security to the attacking the target.

- **Shaft:** This is very similar to the operation of Trinoo application, but there is a special ability on Shaft, that it can online switch and change the master, slave and ports while operating the attack. This ability makes it very difficult of identifying by IDS appliances and for the master-slave communications uses UDP protocol and attacker-master communication uses the TCP protocol [26].
- **XOIC:** Although both XOIC and LOIC application has the feature of flooding attack toward the server, the Test mode feature of the XOIC has the capability of measuring the performance of generating attack, which is not in the LOIC application. In addition to the Test mode, the other two approaches are available named normal DoS mode and DoS attack with messages based on TCP, UDP.ICMP and HTTP.
- **High Orbit Ion Canon (HOIC):** In this application, first send legitimate HTTP packets to target server, then collect information, next extract information from the field like target URLs and the extracted information uses for the attack generation.
- **Low Orbit Ion Canon (LOIC):** Web servers are targeted with this attack application and this application sent the TCP, UDP and HTTP requests towards the target server and the LOIC application capable of leaving C# bases design bug code that will make it prohibited to stop the attack from the user end lead to boost the effectiveness of the threat massively. A massive amount of attack packets are used toward the single IP address in this type of attack.
- **Commview visual packet builder:** Teardrops and land invasions are mainly generated using this attack tool. The term generates two main types of attacks: tear attack and ground attack. The Teardrop is launched by building two overlapping, identical, packets. In this case, using a value assigned by the sender's host helps ID to assemble the parts of the fragmented packet. When launching the Land Attack, a spoofed TCP-SYN packet is crafted which is equal to the IP address, port number, and MAC address of the destination host respectively.

See appendix A for the brief comparison of attacks investigated above.

2.2.1.3. Intrusion Prevention Systems

By inspecting the packet headers, firewalls have the capability of countering foreign attacks. By considering the protocol type, origin and target address, origin or foreign ports the predetermined policies constructs and the malicious packets discards based on that policy by the firewalls [11]. If found any matches between attack policies with packet header and payload, then the alert message generates by the IDS because this process analyses the whole packet [27] and is a software application that naturally analyses and audit the intrusion detection operation. This software application is known as IPS when able to prevent attacks in the IDS system.

The task of IDPs is to detect, log and prevent the attack and block the attack for a limited time until administrators on the CSP side take action against the attacks [26]. There are three ground os attacks and they are:

- close the attacker connection or session by aborting the network connection.
- By disappointing the user account or the IP address, stopping the access of the server.
- stop everything on the server-side such as services, applications and other resources and stop accessing the server.

Some of the detection methodologies elaborate as below:

Signature-Based Detection: In this detection method, a comparison between investigated incidents and with some pre-determined signatures take place to recognise the illegitimate packets.

Anomaly-Based Detection: If the behaviour of any user, application software, host or network connection deviates from normal behaviour as in predefined profile, is known as an anomaly and the detection based on such situation is known as anomaly-based detection.

Stateful Protocol Analysis: This method uses being protocol activity-based vendor-developed universal profile and is very similar to the anomaly-based detection process [27]. In the detection process of the attack, the state of the protocol is used.

2.2.1.3.1. The Classification of IDS

In the cloud service provider environment, the four classifications of IDS has been identified as elaborate below [11]:

HIDS: In this classification, to identify the invasive behaviour of the attacker, the system is concerned with the security permission, login profile, log files.

NIDS: In this classification, real-time early behaviour of logged packets compare with IT and Transport layer headers investigate the behaviour of network packets.

HIDS: In this classification, communication among the VM, hypervisor-based virtual network and the hypervisor are analyzed here.

Distributed IDS: A composition of the above IDS systems placed across the large network is known as distributed IDS.

2.2.1.4. IPS Types

According to the researchers in [6] [11] [27] [28], the IPS can be classified into two categories as TIPS and NGIPS for the CSPs.

Traditional IPS (TIPS): The two classifications namely TIPS and NIPS of IPS has been identified according to the researches. These have been used for a long period ago and normally used known attack signatures.

The drawbacks of the TIPS are elaborated below:

- Inability to update with late detection information.
- Faster networks are not compatible with this type.
- Lack of any mechanism for evaluation.
- Ubiquitous networks do not adapt to such systems.
- The training of the IPS is very harder because of frequent changes in the attack profile.

NGIPS (Next-Generation IPS): These IPS are designed by overcoming the disadvantages of TIPS. And in addition to that, these provide additional features like

This type addresses the issues in TIPS and has armed with the following features.

- Application awareness: NIGPS with this feature blocks access to a particular application.
- Update profiles: The NIPS of this category has the capacity of updating the IPS system automatically according to the real-time traffic learning updates.
- Automated Response: Ability to identify hazards based on the policy and respond accordingly.

2.3. The Literature Review

2.3.1. The Scope of the Research

CC is one of the most widely researched areas, and the impact of DoS and DDoS attacks have had a profound effect on this area. Thus, to protect against these hazards, the next generation of invasion prevention systems must be employed, and these prevention technologies can be understood through empirical evaluation. The context for this research involvement is an empirical evaluation of DoS and DDoS security strategies that challenge cloud computing, and *Figure 2* outlines the scope of the research.

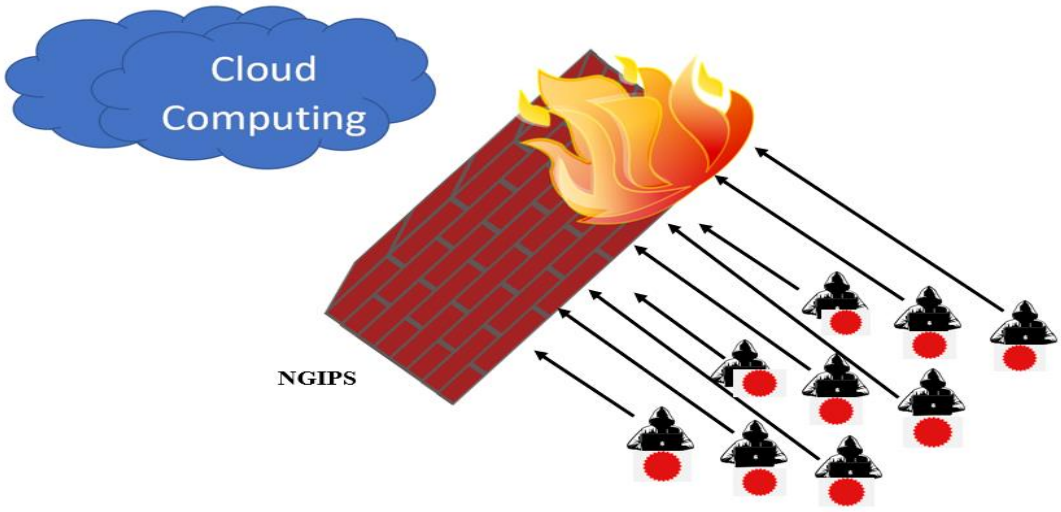

Figure 2: The scope of the research

2.3.2. **A Brief Introduction to Literature Review**

Experiments in the field of computers and communications are carried out using five research approaches: discussion and research, imitation, literature review, and experimentation [29] [30]. According to researcher Kitchenham, [29] all research strategies relevant to the field can be used to find solutions to research questions in the third method, so in this dissertation, the author has selected the literature review to find solutions to questions 1 and 2. The purpose of the first two research questions is to identify tasks relevant to the field of research, and Kitchenham [29]'s attempts to find solutions as planning, conducting, results and analysis. In the planning section, the need for literature, approach to developing literature, rules of analyzing literature is elaborated. In the conducting section, recognizing appropriate literature databases, selection protocols, and extracting data from the literature if elaborated. In the results and analysis section, the synthesis of the established literature is elaborated.

The further details of the above sections are described in the following sub sections.

2.3.3. Plan for better Literature Review

2.3.3.1. The Importance of the Literature Review

The main purpose of this literature is to review NIPS methods that can be used in cloud environments to prevent DoS and DDoS. Identifying the current NIPS methodology [28] is the first step, and the literature review research methodology has been selected for this. Therefore, based on the information obtained through the literature review, NIPS is evaluated experimentally by evaluating the data obtained from it.

2.3.3.2. Research Question

The following research problems have been discussed with the literature study:

- **Research Problem-01:** How to use the next generation Anti-Invasion System (NGIPS) to protect the cloud environment from DoS and DDoS attacks and how to use those systems?

  How does the literature survey study help to answer the RP1: Reviewing the study of literature helps to identify the latest strategies needed to protect the cloud environment.

- **Research Problem-02:** In a cloud environment, what are the next generation Invasion Prevention Systems (NGIPS) diagnostic measures used to protect against DoS and DDoS attacks?

  How does the literature survey study help to answer the RP2: This literature study review will identify the measurements and methods used to evaluate security measures in the cloud environment.

- **Research Problem-03:** How does it decide about the performance skill of "Destination approach" and "Source approach" tracking to prevent DoS and DDoS invasions from being used in Snort, in an NGIPS?

  How does the literature survey study help to answer the RP3: This guides the user to select the most suitable, anti-attack mode used of Snort IPS in the cloud environment. This research study will help the author to identify the most suitable statistical method to analyze the results extract from the implementation stage.

2.3.3.3. Selection of Keywords

According to the researcher [31], the author selects the keywords for this research using the advice that comes with it. Based on the fundamental points of Population, Intervention, Comparison, and outcomes the keywords have been selected.

- **Population:** "Cloud Computing" is the treated population in this research study.
- **Intervention:** Over this research, IPS are considered as an intervention because it gives solutions to the issues in cloud computing.

- **Comparison:** This is the contrast between Next-generation Intrusion Prevention Systems (NGIPS).
- **Outcomes**: The truthfulness of the performance is considered as the outcome because NGIPS is evaluated in this research based on the selected measures.

Background: In this research experiment, the "academic environment" has been selected as the background for the contrasting IPS.

2.3.3.4. Rules of Analyzing the literature

According to Kitchenham [31], 's follow the rules of the review, selecting library databases, selecting educational publications, selecting criteria, integrating quality, and analyzing found information. The rules used here are detailed in the following subsections.

To maintain the relevancy of the findings, the rules used to evaluate the literature has been discussed here.

2.3.3.5. Selection of Appropriate literature.

Base on the availability and the ease of access, the following academic databases have been selected for the academic publication extraction process.

- IEEE
- SCOPUS
- ACM
- WILEY
- Emerald
- ASCE

2.3.3.6. Method of Academic Study Selection

Literature is selected according to the author's opinion subject to inclusion and exclusion criteria. Thus, the study selection process is described below, according to the rules of entry and exclusion. The study selection procedure has been discussed as cording to the elaboration given in the following Table 1.

Table 1: The academic material selection process.

	Conditions consider for the selection.	**Abstract /introduction**	**Full text**	**Reference**
Inclusion Criteria	• Literature from the years 2010 to 2016 are considered. • The controlled vocabulary based on research question is used to select the relevant papers.	Theoretical study and experimental facts in describing the Non-Traditional IPS are considered.	• Papers describing Non-Traditional IPS in cloud computing environment. • Experiment procedure and results are clearly described	The reference list should be well documented.
Exclusion Criteria	• Literature which are in languages other than English are excluded. • Literature other than CSE domain are excluded	If the topic does not match with the literature of the abstracts and introduction is discarded.	If the material does not include the literature in the full text, the material is discarded.	

2.3.4. Other methodology to solve the similar problems

2.3.4.1. The outcome of the Literature Review

Through the literature review, this study looks for solutions to the research questions number one (RP 1) and two (RP 2), The below 2.3.4.2 and **Error! Reference source n ot found.** sections answer those research questions sequentially.

2.3.4.2. Techniques Used in NGIPS

The techniques used in NGIPS are described as shown below.

The researcher in [32] uses the strategy of Packet Resonance (PRS) and the construction of the packets consists of two methods of detection namely packet transit and packet bounce levels. The reflection Mirror Node (RMN) is based on the packet bouncer level and the incoming packet will be logged by the node and bounced back a very small packet to the cloud user. The replying should be done to the node along with the same packet. Using the IP and MAC addresses, the identity is verified. The second level, which is named Packet Transit Level acts as Transparent Mirror Node (TMN) and in this level, the origin passcode is created at account creation time, requested by the node. If the user is unable to pass through any of these levels, the packets are acknowledged as illegitimate and discarded.

In this research [22], through the SYN cookies, the author uses the TCP protocol strategy and provides two security sectors to data packet and control packet. To protect from DoS and DDoS attacks in CC, sequence number encoding and hop count is used. The malicious and legitimate packets can be identified by the MAC generator.

By consolidating CC and Software-Defined Networking (SDN), successfully the survey results have achieved that the complete protection of the CC environment from the DoS and DDoS attacks according to the author [33]. Further, the author Yan Q et al., has concluded that the ability to fight with DoS and DDoS attacks is the advantage of enhanced SDNs.

The application layer based solution has been proposed in the paper [34] to protect from DDoS attacks. According to the author's opinion, several attacks including HTTP flooding and Web-Service address spoofing can prevent and the way of filtering in this type is different from the research [35] and does not affect the user services because heavier workloads are not included. The security service is requested only through the protected server. Therefore, the technique used in this research has proven higher security through authentication and is effective.

In his research [36] the author suggests DoS and DDoS avoidance techniques of application-based IPS. Further, in this research, the author has crafted the source approach rules and the rule can decide and trigger the DoS and DDoS attacks if the request from the one user reaches the 30 request count within one second.

In their research [37], attacks trace back through the adding tag to SOA packets and it leads to protect the system through finding the attack path and then filtering the traffic to prevent DoS and DDoS in the CC environment. Furthermore, the above two methods are impractical because of difficulties in identifying the source of an attack. In this paper, the source address is not considered and there is a research gap.

Alosami et al. [38] proposed a framework called the "Advanced System to Suppress DDoS" with five subsets: Firewall, Client Puzzle Server, Verifier Nodes, and Reverse Proxy (RP). Here the firewall separates the legally casted packets from the non-legal packets based on the results of the Legal Packet Verification System (CAPTCHA). IPS then detects malicious software code hidden in data packets. Controlling the amount of workload on a secure server by IPS through balancing and keeping that server hidden from outsiders. Additionally, if the RP limits are exceeded, the puzzling server is used to isolate illegal users.

In his research [39], the author has used Service Oriented Architecture (SOA) for introducing security service with a filtering tree. In this technique, the DoS and DDoS threats to the cloud servers are secure by a virtual cloud defender while user requests are converted into the form of an XML tree. Further, adding an element to the SOAP request clarifies the legitimacy of the data packets. Using the hash features contained in the relevant SOAP envelope, it generates dual signatures such as "number of children, number of head elements, number of body elements" and provides extra protection against XML overwriting.

A cloud protection system mechanism to get rid of DDoS attackers, researcher Jayanthi et al. [40] has proposed a mechanism by deploying active military nodes (ADAP) to protect cloud infrastructure from damage with maintaining a higher level of DC capability, even in the event of a DDoS attack [40]. In this mechanism, users' validity is checked using the secession key that is generated and shared at the time of account creation. If the number of user requests exceeds N within a certain period, those requests are verified by REGISTER _STATUS and recorded in the BLACK _LIST database, and forwarded to the DUMB terminal located outside the cloud environment.

An IDPS methodology based on a third-party audit mechanism (TPA) has been developed by researcher Saksena et al. [41]. DoS and DDoS attackers can be confronted in a cloud environment, and this methodology consists of three levels: detection, conversion, and, attack assessment. The first level uses the Snort security application to identify flood packets and log them. In the second level, the front server converts the alerts generated by Snort into basic probability assignments (BPA) [41]. Here the dumpsters use the combination rule, using the average factor of the fused BPA value to estimate the attack.

Researcher Rahman [4] has proposed an additional layer of hardware-based security mechanism to protect the CC environment from DoS and DDoS hazards. Here, TTL and hop count, a cross-check that traces back and validates data packets. If the legitimacy of a packet is not verified, it will be abandoned and stopped reaching the service provider, and if the packet is legal, it will be labelled following the lawful and further verified by a "knowledge-based database" and submitted to the relevant host server. All other non-applicable packets will be marked as suspicious and prevented from reaching the destination host server.

Researcher Bakshi et al [42] suggests a strategy to protect CC from the dangers of DoS and DDoS attacks by changing the virtual environment. In this research study, Snort or similar functional IDS systems are installed for auditing purposes. Snort logs all the traffic streams in and out of the network. If there is a sharp rise in the chart drawn for this then behaviour for traffic stream it is confirmed that the sender's end has been acknowledged. However, if the sender's tip is not acknowledged, the IDS commands the honeypot to send a ping message to the sender's IP address, and if it does not respond, it will be considered a DoS attack and will stop the botnet from performing

requests. Thus, the existing server is transferred to another virtual environment server, further enhancing security and updating the routing table.

In his research Buchanan et al (2011)[43] uses IPS based on Snort application and the author has crafted rate filter rules for destination approach and the rule will trigger if it reaches 30 counts within a second and then be considered as DoS or DDoS attack.

Sushi Yu et al. [44] developed a dynamic resource allocation strategy dismantling model based on queue theory to protect CC customers from DoS and DDoS hazards.

Jayanthi et al [45] has proposed a three-stage system of protection against the cloud user environment, the most acceptable, acceptable, and non-acceptable stages for protecting from DoS and DDoS attacks. The first stage is a puzzle system to distinguish legitimate users from automated programs, and the second stage is a network-level attack filtering system and the third stage uses a service-level tweak method to monitor service demand intervals between subsequent services.

Author Anandhi et al [46] developed a port lock method using the Service File Access (SFA) algorithm, which prevents the effects of DoS and DDoS on CC servers. The portal is locked as soon as the user logs into the system, and the Group Policy Object (GPO) locks the relevant portal for the client files and users. A user can only access a file for a short period if the tripping password issued for it is entered before the expiration time, otherwise, the user will be treated as an invalid user.

Dou et al [47] has developed a methodology to protect the CC environment from DoS and DDoS attackers using a confidence-based filtering system(CBF). Confidence values are generated based on the characteristic pairs of TCP and IP headers and stored in the nominal profile. For data packets from users, the CBF calculates points based on nominal profiles. The frequency of data packets depends on reliable values, and if the packet score is low, it is considered illegal and discarded. If the packet score is high, the packet is considered a legitimate packet and allows access to the cloud resource.

Researcher Joshi et al., 2012 [48] recommended a DCDIP, a three-level distributed collaborative data-based hazard identification and prevention framework: infrastructure, platform, and software. The clusters in the DCDIDP architecture interact with three local databases, called the Invasion Assessment Information Foundation (IAIB), the Audit Logs, the Policy, and the Rules Foundation. This DCDIDP architecture has the potential to deliver DOS and DDoS hazards with great potential.

The IAIB maintains an attack pattern, the general behaviour of the packets, and information about access to the data. Audit logs perform logs of other IDPs events and their associated relationships. The Policy and Rules Foundation allows dynamic and policy-based legislation to allow travellers to control their systems from different architectural levels.

The authors of [49] suggest a method of detecting traffic rates, introducing a search technology based on the distance estimation method, and a flood application-assisted DDoS attack method. Using this distance measurement unit, the TTL value is calculated. The deviation is calculated to identify whether the behaviour is normal or abnormal. Using the time-delay approach overcomes the property dependencies of the approaches previously adopted. Using this approach, ISPs rely on data transfer filtering algorithms commonly.

2.3.4.3. Parametric for system evaluation

This sub section illustrates the parametric extracted from the literature review and suggest solutions to the research problem 2. The elaboration of extracted parametric has shown in the following Table 2.

Table 2: The parametric adopted for assessment of NGIPS methods

The congestion at node level (traffic at nodes)	Utilization time of the CPU
The rate of packet losses	The available legal connections
The capacity (Throughput)	**Accuracy**
Time of Response	The transmission capacity (bandwidth)
Number of the aborted connections	The waiting time (latency)
False positives	**False negatives**
The time takes to process (processing time)	Utilization of the memory
The rate of detection	The Round-Trip Time (RTT)
The number of hops	-

The elaborative description of the metrics in *Table 2* is given in below.

Time of Response **(Response Time):** This refers to the time it takes for the NIPS system to respond to the inclusion of attack packets, i.e., the number of packets missing. For a NIPS system to be optimal, it must have a minimum amount of time to avoid pocketing.

False positives: This describes the number of packets that will be dropped, even if they are legal castes. What happens here is that even if the packet is confirmed to be legal, the NIPS considers the packet to be unnecessary. Thus, a more optimal NIPS system can be obtained by minimizing the number of false-positive packets.

Utilization time of the CPU: The utilization time of the CPU describes what percentage of the CPU is used by NIPS.

The number of hops: In an environment protected by NGIPS, the number of hops determines the optimal route that a legitimate packet can take to reach its destination.

Throughput: The number of data packets that reach the NIPS system per second is considered as throughput.

Utilization of the memory: Memory Usage: The memory usage percentage is described by the "Utilization of memory".

Active legitimate connections: In the event of a threat, it is important to calculate how many active legitimate connections are connected to the cloud server. The number of legitimate connections can be maintained at a maximum then the NIPS system can be evaluated as a very high level of the security provider.

Transmission capacity (Bandwidth): The number of packets of data that can be transmitted over a while over a network is called the transmission capacity. Legal users will be able to provide better transmission capacity by dropping illegal data packets into the NIPS system.

The number of connections aborted: Calculating the number of legal contacts a cloud server has abandoned in the event of an attack is called the number of aborted contacts. The best performance of an NGIPS can be confirmed by evaluating a cloud server by calculating the number of active legal connections associated with a CC server.

Waiting time (Latency): The time it takes for a data packet to reach into its service provider (server) in response and return to the user is called waiting time (latency).

False Negatives: If the number of false negatives can be kept to a minimum, a good NIPS system can be obtained by minimizing the number of instances where non-legal packets are assumed to be legitimate by NIPS.

The time takes to process (processing time): The time it takes NIPS to read a data packet to make decisions is called processing time.

Accuracy: This is called the fraction of accuracy required for predictive results, true positive, positive and false positive. In a complexity table, there are four types of predictive results and the fourth type of predictive outcome, false negative, is not considered for **Accuracy**.

The congestion at node level (traffic at nodes): For analysis, the amount of traffic presented to the NIPS refers to the congestion at the node level.

The Round Trip Time (RTT): The time duration between sending data from the user to the cloud server and receiving responses from the server to the user is known as Round Trip Time (RTT).

The Rate of Paket Loss: The rate at which a packet is discarded untested, whether it is pure or malicious due to neglect by NIPS, is called the packet loss rate.

The Rate of Paket Detection: The packet rate that a NIPS system can detect, either legitimacy or maliciously, without being discarded is called the "packet detection rate".

2.2. Summary

This chapter discussed the studies done in the areas of cyber security in a cloud environment, denial-of-service attacks, both attack and normal traffic generation, customizing Snort rules to detect and prevent a virtual environment for the protection of virtual network environment. Further, it has discussed possible metrics to evaluate the system. The next chapter will discuss the approach for the performance evaluation of the IPS system.

Chapter 03

Technology Adopted

Enhanced Rule-Based IPS - Beyond the Traditional

.

3.1. Introduction

Chapter 2 discussed the studies that have been done in the attack traffic protocols, traffic generation tools and particular traffic detection and prevention software technologies. Some suitable technologies for the design and implementation of the proposed IPS for the protection from DoS and DDoS attack have been identified in the literature reviews. Those technologies adopted are Tcpreplay, LOIC, and JMeter traffic generation tools. Windows & CentOS Linux operating systems. Further, SNORT a rule-based IPS application software tool, as used in many literature reviews, is chosen to write enhanced rules to detect and prevent DoS & DDoS attacks.

3.2. Technology Adopt to Solve the Problem

3.3. TCPreplay

Since any set of data packets can be repeated with different parameters such as network speed, TCPreplay [43] [50] tool is used instead of tools like Harpon because the Harpoon tool cannot generate real-time traffic.

Recreate the captured traffic using TCPreplay as shown in Figure 3.

```
sudo tcpreplay -i eth0 -p 30000 -L 200000 –loop 5 final.pcapng
```

Figure 3: Regenerate the captured traffic 1

Traffic is looped and the test is made to run for a very long time, the delay is then can be estimated. Using the tool, repeating the traffic as shown in the following command in Figure 4:

```
sudo tcpreplay -i eth0 -p 30000 -L 2000000 –loop 100 final.pcapng
```

Figure 4: The command to repeat the traffic captured 2

3.4. LOIC (Low Orbit Ion Canon):

This tool fire with TCP or UDP or HTTP packet forwarding to streamline targeted servers for disabling the services of web servers. Also, this tool is a very good automated tool that can execute attacks in real-time [50] [51] [52]. The interface of the LOIC tool is shown in Figure 5.

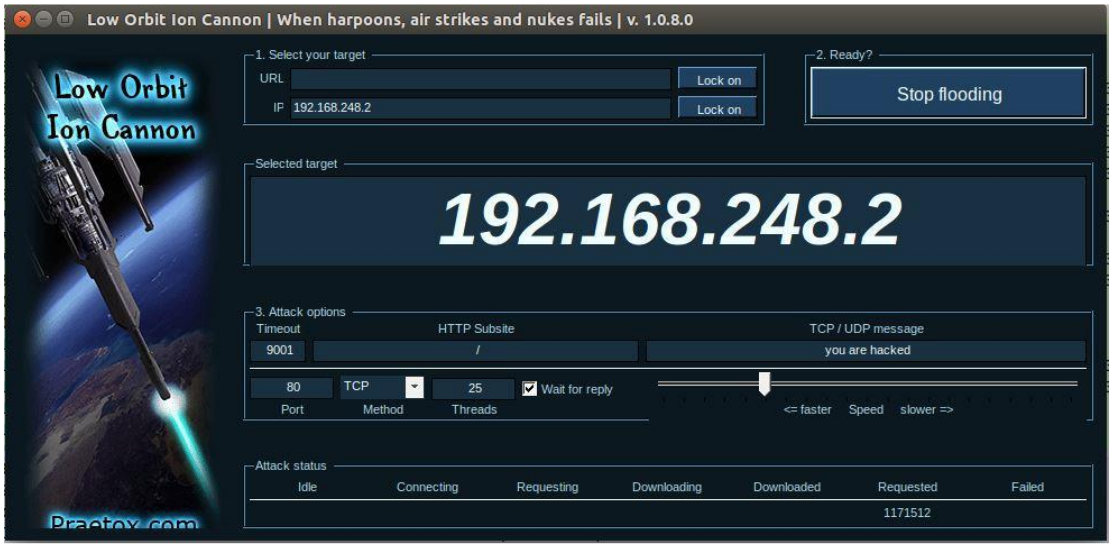

Figure 5: Preparing TCP attack traffic toward the targeted Host.

3.5. JMeter

Using this JMeter [53] tool generates the genuine traffic needed to test the client's endurance. This JMeter tool is commonly used to test web applications and is an open-source testing tool developed by Apache. This tool generates HTTP traffic using a GUI interface, keeping regular time intervals, and relying on a number of selected user threads.

3.6. Snort

Snort is one of the most commonly used tools to detect or prevent attacks. The authors of [48] [36] [46] have used Snort with their proposed architecture in their research to protect against DoS and DDoS attacks.

In his research, the author [36] prepared an alert message as shown in the following Figure 6.

```
" drop TCP any any -> any 80 ( \
Msg:"Reset outside window", \
Detection_filter:track by_src, count 30, seconds 1; \
New_action drop; timeout 50; sid:100001;)"
```

Figure 6: An alert message produced with Snort IPS

3.7. Other tools and Technologies

In addition to the above technologies following software tools techniques has been used for the implementation of the independent and dependent variables (parametric) of this research.

Wireshark: Using Wireshark's ability to analyze packets, packets sent to the server via IPS are monitored.

Iperf: Using the Iperf tool, the available transmission capacity can be calculated.

Hping3 & Wireshark: To further verify the results provided by the Wireshark tool, the packet damage rate is calculated using the Hping3 tool.

LOIC & Hping3: What is done here using this technology is that when the IPS system is attacked, the LOIC tool sends packets to the IPS at a second interval, using the Hping3 tool to calculate the percentage of packet damage.

Ping: Using this ping tool, researchers find the delay.

Uptime: The load on the CPU is determined using a Unix tool called Uptime.

Free: Using this free Unix tool, the system memory consumption is calculated.

3.8. Why These Techniques are Appropriate to Solve the Problem

Tcpreplay gives you the ability to repeat the dataset with various options such as network speed. The TCPreplay tool is capable of regenerating the same database with different choices, such as network speed. **LOIC** technique is widely used to stress testing as well as DoS and DDoS attack traffic generation as it has the capability of flooding traffic composed with different protocols TCP, UDP and HTTP toward the target. Using the JMeter tool you can be tested the capabilities of dynamic applications in the web environment, as well as static and dynamic resources. This JMeter tool is capable to reproduce the full capabilities of a system across different loads (under weight classes) or testing the viability of a system by reproducing an overloaded flow of traffic on an object, network, server group, or server. Thus, different types of applications, servers, or Web - HTTP, HTTPS (Java, NodeJS, PHP, ASP.NET, ...), SOAP / REST web services, FTP, JDBC database, LDAP, JMS Message - The JMeter tool capable to load and evaluating traffic on Message-oriented-middleware (MOM), MAIL - SMTP (S), POP3 (S) and IMAP (S), Native commands or shell scripts, TCP, Java Objects protocols.

The base reasons for choosing Snort are the ability to install in any network environment, logs, open-source, operating system fingerprinting, packet logging, real-time traffic monitoring, and ease of Sort's law enforcement. The second reason to select this Snort as IPS is that it is open-source that any user can use in any way and the software is free to use.

3.9. Summary

This chapter has discussed selected technologies for the experimental environment and tools used for the selected metrics evaluation and data collection. The reason to select both malicious and normal traffic generation, IPS system, metrics evaluation techniques has been clearly explained. The next chapter explains the approach of the research.

Chapter 04

Approach

Using Enhanced Rule-based IPS for Expanding Performance For Evaluating Performance

4.1. Introduction

Chapter 2 presented the critical review of many pieces of literature which were required to design and develop an efficient and secure system for the protection from the DoS and DDoS attacks in the virtual cloud environment. Chapter 3 discussed the technology for implementing performance-enhanced IPS based on customized Snort rules to protect network resources in a cloud environment from DoS and DDoS attacks. This chapter presents our approach to assess the Snort based IPS in the virtual cloud surrounding verses DoS and DDoS attacks under several headings, namely, hypothesis, input, output, process, users, and features. Our approach is evaluating the performance of the IPS by customizing Snort rules for selected metrics on both source approach and destination approach. This chapter highlights the key features that distinguish our approach from other existing approaches for necessary protocols, evaluation metrics and rate based enhanced Snort rules. Further, this chapter highlights how the normal approach offers a cost-effective and efficient solution for strong protection of the virtual environment from the DoS and DDoS attacks.

4.2. How does the author adopt the technology to solve the said problem?

The adaptation of the technology describes under sections 4.3, 4.4, 4.5, 4.6, 4.7 and 4.8 in this chapter through elaborating hypotheses, inputs to the system, outputs from the system, different processes (generation of input traffic both legitimate and attack), detect and prevent malicious traffic), features and users of the system in orderly.

4.3. Hypothesis

Packet identification (source filtering) by finding the source of packets and filtering accordingly [36]**:** The hypothesis [43] here is that this filter will never work if a large number of IP addresses are used for the attack.

Packet identification (destination filtering) by finding the source of packets and filtering accordingly [38]**:** The hypothesis [38] here is that this filter work if the difference between legitimate and illegitimate packets cannot be determined by the Snort IPS.

The main implication of this hypothesis is that packet loss speeds up when there is an increase in the traffic of attack packets through the IPS system.

4.4. Input (Independent Variable)

TCP, UDP, HTTP and Mixed approach DoS & DDoS attack traffic input to the user or target machine environment. This traffic generates by the LOIC traffic generation tool. Further, the JMeter tool is used to input the normal traffic composed of different protocols. Furthermore, the TCPreplay tool is used to maintain the same continuous attack traffic as input throughout this testing process and the elaboration summary of attack traffic inputs to the IPS system has given in Table 3.

Table 3: Malicious traffic inputs to the system

#	Input	Measured by	The Tool Used
1	Traffic rate of attack traffic of: ▪ TCP Packet floods for both source and destination approach.	PPS (packets per second)	LOIC
2	Traffic rate of attack traffic of: ▪ UDP packet floods for both source and destination approach.	PPS (packets per second)	LOIC
3	Traffic rate of attack traffic of: ▪ HTTP packet floods for both source and destination approach.	PPS (packets per second)	LOIC
4	Traffic rate of attack traffic of: ▪ Mix Packet floods for both source and destination approach.	PPS (packets per second)	LOIC

4.5. Output (Dependent variables)

Dependent Variable: The output values provide information (speech) about a dependent variable based on the selected parameter.

The output of the system is shown in Table 4.

Table 4: Output from the system

#	Output	Measured by	The Tool Used
1	CPU Load results of TCP, UDP, HTTP & Mix packet floods as a percentage.	Percentage of CPU capacity	The Unix tool: "Uptime"
2	Memory utilization results of TCP, UDP, HTTP & Mix packet floods as a percentage.	Mbps (megabits per second)	The Unix tool: "Free"
3	Bandwidth availability of TCP, UDP, HTTP & Mix packet floods as a percentage.	Mbps (megabits per second)	The tool: "Iperf"
4	Latency availability results of TCP, UDP, HTTP & Mix packet floods as a percentage.	milliseconds	The tool: "Ping"
5	Packet loss rate results of TCP, UDP, HTTP & Mix packet floods as a percentage.	PPS (packets per second) Or $\frac{\text{number of packets not received}}{\text{total number of packets sent}}$	The tools: "Wireshark" & "Hping3"

4.1. Process

4.1.1. Generation of input traffic

During the input traffic generation process, both attack and normal traffic will be generated.

4.1.1.1. Generation of attack traffic

Using the LOIC tool and the server's IP address, it generates illegal traffic, activates attack mode by selecting the target IP address, and the protocol used via the GUI. Using the TCPreplay tool allows the attacker to maintain the same amount of traffic throughout the test.

4.1.1.2. Generation of normal traffic

The JMeter tool is capable to generate legitimate traffic, including loops, ramp-up duration, and user threads. In this research, the author has used two user threads with a speed of 4 seconds and a loop count of 5 seconds. Uses the TCPreplay tool to capture traffic generated by other tools. Using the Wireshark tool, traffic packets are captured and stored as a ".pacap" type file. Passing a console command can restart the captured traffic to replay.

4.1.2. Detect & Prevent malicious traffic

Two different rule techniques were applied with the Snort tool used in this experiment. To analyze data packets stored in an iptable queue, the Snort IPS tool runs in inline mode. The author analyzes the data packets by verifying the results of the “Snort.conf” file, printing the outputs on the console, and placing a copy of the outputs in the log file, through the instructing Snort to run inline mode.

4.2. Features / Technology

In the source approach, packet-based identification and filtering of packets are done by scanning the source IP address and enabling Snort to activate the filter if a specific packet rate is reached. In this approach, the Snort rule commands the IPS system to start the filter by defaulting the source packet at reaching a certain ratio. (The problem that arose under this hypothesis is that if an attack is launched using a large number of IP addresses, the filter will never work.)

Destination detection and filtering of packets are done by ordering Snort to swipe packets to the destination IP address and to enable filtering when a predetermined packet rate is reached. In this approach, the Snort rule commands that fraudulent packets arrive at the destination and activate the filter when it reaches a certain predetermined ratio. (It is a matter of Snort's inability to distinguish between legal traffic and illegal traffic when the filter is activated. As a result, the packets arriving at the destination are dropped, and as a result, cloud resource users are down.)

4.3. Users

This Snort rule enhancement to protect from DoS and DDoS will be used by network engineers, administrators, academia in the field. Further, this will apply to the network and data centre protection staff to protect the virtual environment.

4.4. Summary

This chapter presented our approach to enhancing the capability of Snort rule-based IPS to protect the network resources in the virtual environment from the DoS and DDoS attack. The next chapter shows the design of the novel approach presented here.

Chapter 05

Analysis and Design

Analysis and Design of Performance Evaluation System for Enhanced Rule-Based IPS.

5.1. Introduction

Chapter 3 discussed the technology for evaluating the Snort rule-based DoS and DDoS protection IPS system. Chapter 4 presented the approach to developing an efficient and secure IPS for the protection of network resources in the cloud environment from the above attacks. This chapter amplifies the approach and defines the construction of the proposed solution. The high-level architecture of the solution is composed of six units particularly Generating Both Attack and Legitimate traffic, Capture Network Traffic, Passing Captured Network Traffic Through Snort IPS, Drop or Pass the Network Traffic, Measure the Evaluation Metrics on the Target Machine, Tabulate-Graphing & Evaluating of the Traffic. Further, this chapter concisely depicts the theoretical construction of the units which have been used to estimate the source and destination approach of the Snort IPS including statistical test and hypothesis design.

5.2. Details of Design (analysis & Design)

Details of the design diagram are present in section 6.3 as high-level architecture. More details about high-level architecture elaborate under the sections 5.2.1.1, 5.2.1.2, 5.2.1.3, 5.2.1.4, 5.2.1.5 and 5.2.1.6 in orderly as traffic generation, insert traffic, passing traffic through IPS, drop or pass the network traffic, measure the evaluation metrics, designing the evaluation.

5.2.1. High-level Design Diagram of the Proposed System

The high-level architecture of the evaluating snort rule-based IPS in the virtual environment to protect from DoS & DDoS attacks is shown in Figure *7*. Within the architecture, modules A, D, and E work as the core of the solution. Module B and C are connected and facilitate the functions of the modules, A, D and E. Module F does the performance evaluation of the system. Next, we briefly describe the function of each module.

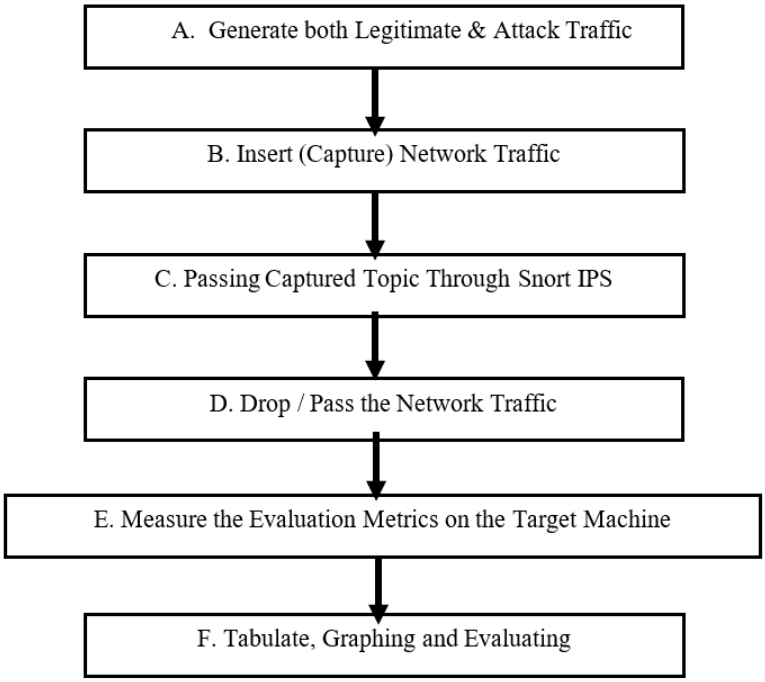

Figure 7: High-level design diagram of the proposed system

The high-level architecture of the virtual environment inside the Host machine shows in the following Figure 8. In this architecture, separate virtual machines for the attacker (VM2), the legitimate user (VM1), Snort intrusion prevention (VM0) and target server (VM3) are included. This architecture with four virtual machines can extend into virtual machines with separate Hosts. The most accurate output of the metrics values can extract with such extension.

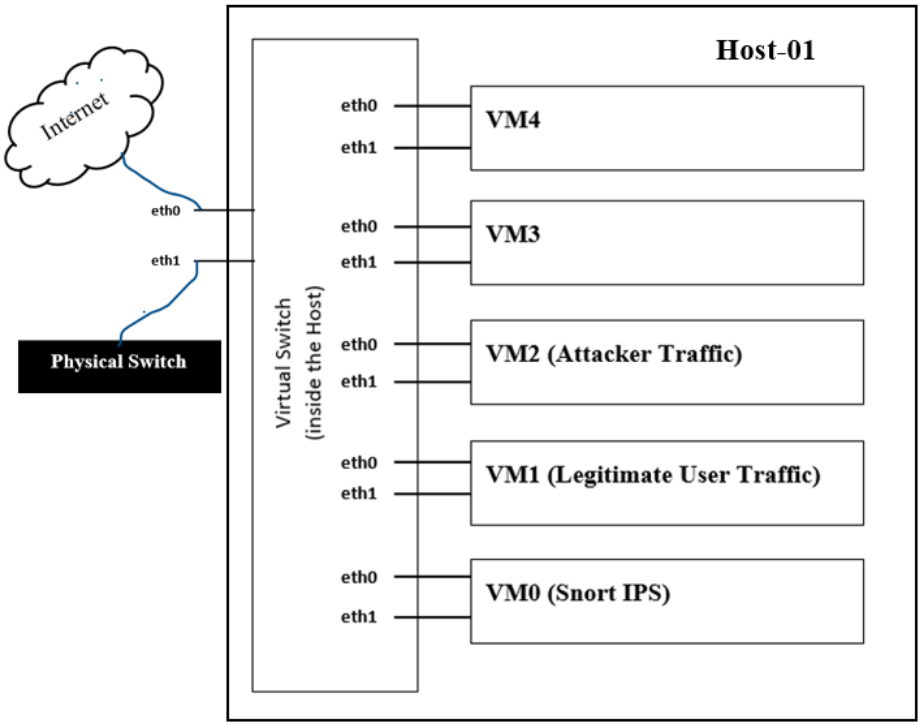

Figure 8: High-level diagram of the Virtual Environment

The high-level architecture of the Snort IPS machine shows in the following Figure 9.

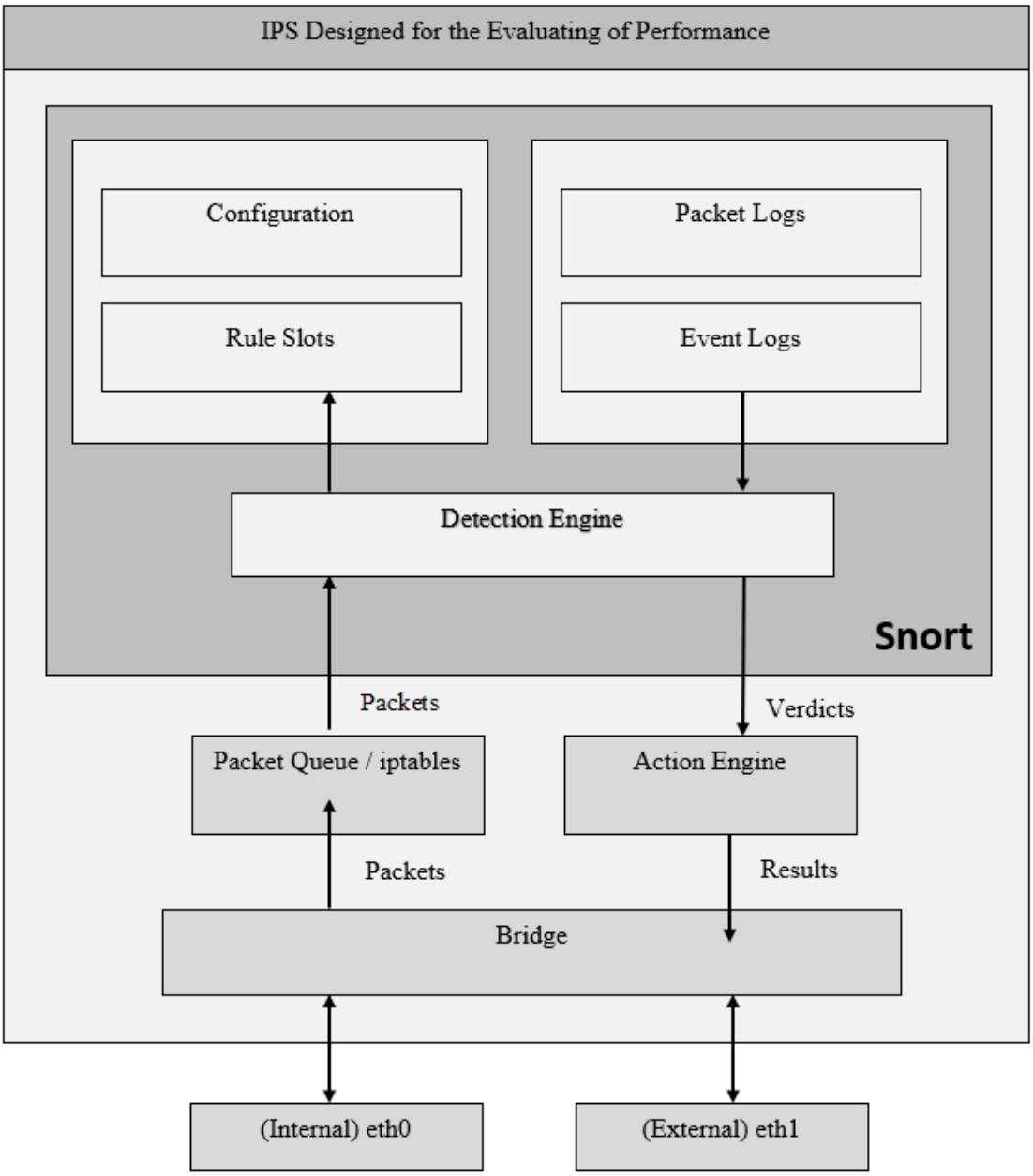

Figure 9: High-level architecture diagram of the Snort IPS.

As depicted in *Figure 9*, Snort IPS machine consists of two ethernet connections as internal and external. Through the internal ethernet connection, packets are read into inside the system and forwarded to iptables in the packet queue. The detection engine reives packets from the packet queue and the malicious packets are detected by the configured Snort rules. Every packet is logged by the packet logger. Every malicious packet is logged by the event logger. Then the packets are forwarded to the action engine and then the malicious packets are dropped by modifying the iptables and normal packets are forwarded to the network through the external ethernet connection.

5.2.1.1. The Generation of Traffic

The real-time traffic generated to evaluate the IPS and is very challenging to produce without the support of automation. The LOIC tool is the result of literary discoveries, an ingenious tool that can generate real-time traffic and launch an attack on a server with a heavy load. Chapter 2 deals with three types of DoS and DDoS attackers that are generated based on TCP, UDP and HTTP with the literature survey. The TCPreplay tool is used to launch and control the same amount of traffic throughout the testing process.

To generate illegal traffic within this framework, the LOIC tool provides easy functionality and GUI is deployed in the function. LOIC is also able to generate real-time TCP, UDP, and HTTP flood attacks based on the DDoS format, and use all three attacks against the Apache webserver. It provides an insight into how different protocols affect the performance of IPS. JMeter is used as a testing tool to deliver legitimate traffic and load test traffic against the server.

JMeter, an open-source computer tool developed and developed by Apache, is widely used to generate legitimate traffic and test web applications. Attacks are made by generating HTTP traffic over a GUI medium at specified time intervals based on a predetermined number of user threads. The system must ensure that the same amount of data flow is maintained throughout the testing process to evaluate both IPS technical strategies. After the data is generated for the first time in the test environment, it is captured by "Capture and Save Datasets" and does not use existing datasets such as DARPA and Bro. Although tools such as Harpon cannot generate real-time traffic, the TCPreplay tool is used because it has the ability to repeat datasets using different dimensions, such as network speed.

5.2.1.2. Insert capture traffic

The traffics generated by the attacker and legitimate source virtual environment, in naturally inserted into the virtual switch and then to the local physical switch as shown in Figure 8. The inserted traffic to the system passes through the Snort IPS as it automatically listens to the network traffic in the virtual switch and the in the physical switch.

5.2.1.3. Passing traffic through IPS

Traffic in the local network automatically passes to the Snort virtual machine environment as there is a virtual switch inside the host with the virtual environment as according to Figure 8.

5.2.1.4. drop or pass the network traffic

The author uses the findings in the literature to distinguish and filter illegal packets and legal packets using "alert" and "rate_filter" that travel within the network. Alert messages are generated using the "Alert" and the attribute values of the Snort rules file must be satisfied to generate alert messages. The "rate_filter" parameter, a configuration parameter used to filter packets, contains many properties. Thus, there are many properties in the "rate_filter" parameter, and there are two different ways of dropping packets depending on the dropping packets based on destination and the source.

5.2.1.5. measure the evaluation metrics

In this research to evaluate IPS systems, the parameters are used as shown in Table *5* and are divided into three categories according to the literature study.

- **Metrics of Input:** Under this classification, response time, availability of bandwidth, reliability, and delay are presented.
- **Metrics of Resource:** This classification presents the amount of memory usage and load on the CPU.
- **Metrics of Response:** In this classification, the packet loss rates such as false negatives and false positives are subject to be considered.

The parameters selected here are the most commonly used and are tested according to the literature review in a service environment similar to those in which the parameters are used. Thus, as a result of the literature review, it was selected to use the above-mentioned parameters that are commonly used to evaluate IPS systems and have summarised them in *Table 5*.

Table 5: Parametric and the category of the selected parametric for evaluation

#	The Parametric	Category of parametric
1	Load of the CPU	Resource metric
2	Utilization of Memory	Resource metric
3	Bandwidth availability	Input Metric
4	Delay (Latency)	Input Metric
5	Rate of packet loss (both “false positives” and “false negatives”)	Response Metric

5.2.1.6. designing the evaluation

In this research, performance should be assessed on two groups as the source and destination approaches. Here the performance of both categories is measured by a single packet rate called "packets per second". Also, here, there is one level of independent parameter matching with the two dependency categories. By analyzing the data types in this research with statistical data types, it is concluded from the literature that the data of the dependent variable are general. According to the study of literature, the statistical method to be used to validate this type of data is the "Wilcoxon" test. Here the author performed the "Wilcoxon signed-rank test".

5.3. Theoretical Analysis of the Architecture.

This section elaborates that the descriptions and roles of each diagram component and how one component interacts with the other component.

5.3.1. Network Intrusion Detection approach

Alert messages can be generated in a variety of ways in the network intrusion detection approach. Of these, "alert" and "rate_filter" were used to filter to identify non-legal packets. "Alert" is used to generate attachment messages that satisfy the attribute value of the rule file. The parameter configuration used to filter the packets is called "rate_filter" and this parameter contains many numbers of properties. Among these properties dropping packets based on the two different ways of identification by source and identifying by destination are considered here.

The author [36] has formed the following indication message.

*"drop TCP any any -> any 80 (*
*Msg:"Reset outside window", *
*Detection_filter:track by_src, count 30, seconds 1; *
New_action drop; timeout 50; sid:100001;)"

If the rule specified by sid: 1000001 above is broken thirty times in a second, that is, if the TCP packet from any IP address and port reaches any number of IP addresses and ports up to 80 times in a second, the resulting packets commands to stop hereafter. Packet detection here is the "source identification" approach and the "by_src" attribute is used for this approach. This approach notifies the Snort IPS that the packet coming from the source IP address is fraudulent. Whatever the destination, Snort will discard the packets by triggering the rule.
The author [43] uses the following filter rule.

"rate_filter \
gen_id 1, sig_id 100001, track by_dst, \ count 30, seconds 5, new_action drop, timeout 30"

By the above rule, when the filter is triggered, the packets coming to the destination IP address are discarded, regardless of the source.

The following is the rule of detection used by the author in action.

TCP rule that detects TCP packet with the SYN flag on in destination of an FTP server.

"alert tcp $EXTERNAL_NET any -> $HOME_NET 21 \

(flags: s; msg:"FTP – TCP FLAG"; classtype:attempted-dos;\ sid:100001; rev:1;)"

The function of the Snort rule mentioned above is to create a rule with sid. 100001. It issues alert messages about TCP packets coming from any port and EXTERNAL_NET (user IP addresses), warning of dangerous packets coming to HOME_NET (i.e. the IP address of the server machine to be protected by the IPS system) and discarding those packets.

5.3.2. Snort Rule Modification

Researchers [43] and [54] have pointed out that there should be a generalized environment for the evaluation of IPS systems, even if the authors enforce the rules in the environment simulated under subsection 5.3.1 According to the rules, the environment created in Section 5.3 is different from each other.

Here the rules are formulated as follows according to the context surrounding the research.

#TCP rule to detect TCP packet with SYN flag in the network.

alert tcp !$HOME_NET any -> $HOME_NET 80 (flags: s; msg:"Attempt to access server is made with TCP packets"; classtype:attempted-dos; sid:1000990; rev:1;)

#UDP rule to detect UDP packets in the network

alert udp !$HOME_NET any -> $HOME_NET 80 (msg:" Attempt to access server is made with UDP packets"; classtype:attempted-dos; sid:1000991; rev:1;)

From the rules set out above, alert messages are generated for packets passing through the IPS system. Because HTTP packet streaming takes place through the TCP protocol, no separate rules are set for HTTP and the rules written for TCP are used. Following are the rules written by two technical strategies for filtering illegal packets.

- Source Identify Packet Filtering (Source Approach) [36]:

In this approach, when a specific packet rate is reached, Snort rules command to activate the filter and to tamper with the packets coming from the source IP address. The hypothesis of this technology is that the filter will never work if the attack is launched on a large number of source IP addresses [43].

The filtering rules for this technology are as follows.

```
# IPS rule to filter TCP packets
        rate_filter \
            gen_id 1, sig_id 1000990, \
  track by_src, \        count 30,
seconds 1, \        new_action drop,
timeout 30
# IPS rule to filter UDP
packets rate_filter \
gen_id 1, sig_id 1000991, \
      track by_src, \
count 10, seconds 2, \
      new_action drop, timeout 30
```

If the number of requests to the server reaches the limit of 30 within a second, Snort is instructed to drop the packet and terminate the connection for 30 seconds and to deceive the data packet coming from the source.

- Packet Detection and Filtering (Destination Approach) [43]: Here Snort cheats the packets going to the destination IP address and activates the filter that reaches the packet rate. The hypothesis [43] of this technology is that Snort will not be able to distinguish between legal and non-legal packets, even if the filter is functional. As a result, all packets arriving at the destination will be dropped, and legitimate user packets will not reach the destination.

  The filtering rules used within IPS in this technology are as follows.

```
# IPS rule to filter TCP
packets rate_filter \
gen_id 1, sig_id 1000990, \
    track by_dst, \     count
30, seconds 1, \
new_action drop, timeout 30 #
IPS rule to filter UDP packets
rate_filter \     gen_id 1,
sig_id 1000991, \
    track by_dst, \
count 10, seconds 2, \
    new_action drop, timeout 30
```

When the request rate to the server reaches 30 per second, Snort tells to drop the packet and disconnect the connection for 30 seconds, and to deceive the packet arriving at the destination from any source in accordance with the above filtering technology.

5.4. Statistical Analysis

5.4.1. Introduction to Design of Data Analysis

Statistical analysis is performed under the guidance of the authors of [55] and [56], and statistical analysis is performed according to the authors of [56] and [57] to confirm the results.

The Wilcoxon test was performed by the author to confirm the results of this experiment. With this parameter test, a difference can be compared and this can be done for two groups in which samples are correlated.

i.e. …:

- two groups = source approach and destination approach;
- correlated samples = each source and destination approach use same N-PPS attack traffic rate (e.g. 2000pps).

The author has chosen this Wilcoxon test because there are two matching groups and the data type is considered the normal data type. According to the study in literature, the authors of [55] [58] and [59] have chosen the Wilcoxon test to evaluate an algorithm with a data type similar to the data used in this test, which may justify the selection of the Wilcoxon test for this experiment.

5.4.2. Parametric Test

Once the data variable type and the data distribution type are identified, statistical tests can be performed and it is clear that the data type in this research is general data [55] [60]. There is evidence that the resulting data do not behave in a normal distribution. For this reason, the author decided to analyze the data with the Wilcoxon test, a non-parametric test, because sample groups can be taken in pairs.

5.4.3. Justifying the Results from Iterated Experiment Sessions (Mean Square Error (MSE)

In this research, ten trials were conducted for each traffic rate from 2000pps to 30000pps and the mean value was used for the Wilcoxon test [57] because it gives very good results. However, since it is not possible to justify the mean value of the test for repetitive test result values, the author justifies using the median value of the repetitive results for the Wilcoxon test by calculating the "mean square error (MSE)". According to the author – Emiliano

Furthermore, the researcher in [61] used the minimum MSE for the optimum results. Therefore, it is fair that conduct a Wilcoxon test if the dataset provides the small MSE values.

The author plans to calculate the MSE value by following the phases below.

Phase 1: Using the following formula, the error of the values obtained by repeating the experiment each time is calculated.

$$\text{Error} = (\mu - X)$$

where;

μ= The mean value calculated for the data set used in the iteration test

X = The resulting value extracted from the experimental dataset which is corresponded.

Phase 2: At this stage, the negative values in the dataset are removed by calculating the square of error value as follows.

$$(\text{Error})^2$$

Phase 3: Thus, dividing the sum of squared Error by the number of samples, find the MSE as follows:

$$Mean\ Square\ Error\ (MSE) = (\sum_{i=1}^{n} Error_i^2)/n \quad (1)$$

Where;

n = number of samples and i = sample number

Preliminary testing confirmed that the MSE value for all samples ranged from 0.06 to 0.71. Such a low mean value gives a very good tendency to perform the Wilcoxon test.

As described in equation (1), the calculation of the MSE value corresponding to the traffic rate of 2000pps at source access for the CPU load parameter is systematically shown in Table 6.

Table 6: Calculate the MSE corresponding to a ratio of 2000pps at source access for the CPU load parameter.

Traffic rate in PPS	CPU load values for Source approach		
2000	X value (%)	Error = (μ-X)	Error square $(Error)^2$
	10.5	-1.08	1.16
	12.1	0.52	0.27
	12.4	0.82	0.67
	12.6	1.02	1.04
	11.6	0.02	0.00
	10.8	-0.78	0.60
	11.6	0.02	0.00
	11.6	0.02	0.00
	12.4	0.82	0.67
	10.2	-1.38	1.90
Number of the population (N) for both source and destination approaches for each protocol composition			16
Mean (μ)			11.58
SSE			6.33
MSE = [(1/n) *SSE] = [(1/10) * 6.33]; Here, the n = number of times it repeated the test=10.			0.63

5.4.4. What is Wilcoxon Assessment?

Two groups should be evaluated in this research, as described in Section 5.2.1.6 as source and destination. The performance of both groups is equally evaluated by measuring the number of packets per second as a unit and the packet ratio. Thus, there are two corresponding groups that match one level of an independent variable.

A conclusion can be drawn by analyzing the dependent variable data into normal data with the general data in this dissertation [60]. Furthermore, research [55] [56] and [58] has shown that this type of data can be validated using the Wilcoxon test. Thus, the author of this research conducted the Wilcoxon signed-rank test as follows.

5.4.5. Design of hypothesis

According to section 4.3, the hypothesis designed is as follows.

The formed null hypothesis and alternative hypothesis for the statistical assessment is as follows.

Null hypothesis:

> **H_0**: Snort does not show any difference between both source and destination approaches.

Alternative hypothesis:

> **H_1**: Snort show at least any difference between both source and destination approaches.

5.5. Summary

This chapter discussed the design of a performance evaluation system for Enhanced Snort rule-based IPS to protect network resources from DoS & DDoS attacks with a high-level architecture diagram. High-level architecture has six modules such as "Traffic generation", "Input network traffic to target server", "Capture the network traffic by Snort IPS", "Drop or pass the network traffic", "Measure evaluation metrics in the target machine" and "Table, graph & evaluation" modules. The architecture diagram of these modules has been presented. Then, theoretical aspects of traffic generation, source and destination approaches Snort rule customization, measure the evaluation metrics and evaluation proves with statistical tests have been elaborated. The next chapter describes the implementation aspects of the design and implementation for the performance evaluation system for Snort IPS in the virtual environment.

Chapter 06

Implementation

Implementation of Performance Evaluation System for Enhanced Snort Rule-Based IPS.

6.1. Introduction

In chapter 5, the high-level design of the performance evaluation system for Enhanced Snort rule-based IPS to assure the network resources from DoS & DDoS attacks has been described in terms of what each component does. Moreover, the assessment of the selected metric has been described. This chapter describes the implementation of each component regarding performance measurement of selected metrics against different DoS & DDoS attacks with TCP, UDP, HTTP protocols for both source and destination approaches. In this sense, this chapter is about how the system is implemented.

The whole test process is considered in detail here and the structure of the experimental process is as follows in the sub-sections:

- **Section 6.2:** This section provides an overview of the experiment and the process of the test, along with the objectives and intentions of the test.
- **Section 6.2:** The experiment discusses the hypothesis tested and focuses on how the experiment is performed.
- **Section 6.3:** Demonstrates how the experiment is performed and its process.
- **Section 6.4:** Provides an explanation of the data audit and investigation procedure.
- **Section 6.5:** Summarize and present the rewards (results) of the experiment.

6.2. Overview of the implementation

Initially, the design described in the previous chapter has been implemented with each of the attack types based on protocols TCP, UDP and HTTP (metrics) with the mixed protocol based attack floods chosen as a controlling factor. Then the performance testing of the Snort IPS for the protestation from DoS and DDoS attack has been implemented with each of the performance metrics selected. Five different performance metric values and three protocol-based attacks on both source and destination approaches have been implemented.

6.3. Experiment Design

In this section, the composer illustrates the way of the testing process tasks is designed to assess IPS approaches. This section is explained in three sub section as described below.

- **Seb section 6.3.1:** This section provides a detailed overview of the software and hardware requirements for use and the experimental bed design configurations.
- **Sub section 6.3.2:** The legitimate and non-legitimate packet streams generated during the entire research process are discussed here.
- **Sub section 6.3.3:** This section explains the replica of how to evaluate IPS technology strategies using selected parameters.

6.3.1. Construction of the testbed

Both IPS techniques are evaluated by comparing the results of a separate test for three different dimensions of DDoS attacks. Thus, this task is accomplished by designing a testbed so that any IPS techniques can evaluate the system versus three different attacks. In an Apache web server, a web page with basic properties is created and stored on an FTP server computer. The IPS system installed with Snort is configured in inline mode and installed on the network, adjacent to an FTP server. The network is configured so that the traffic flow passes through the IPS router and reaches the server. The TCPreplay tool has been configured on the TCPreplay machine that is used to assess both technologies used in the IPS system. The LOIC Tool software is installed on two attacker machines to flood the traffic toward the webserver. A typical user machine with the JMeter tool installed uses it to generate legitimate traffic within a specified time frame. The description of the machines in the testbed is shown in *Table 7* and the graphical description is given in Figure *10*. The IPS technology is further tested using the line-by-line black box testing methodology in an offline environment.

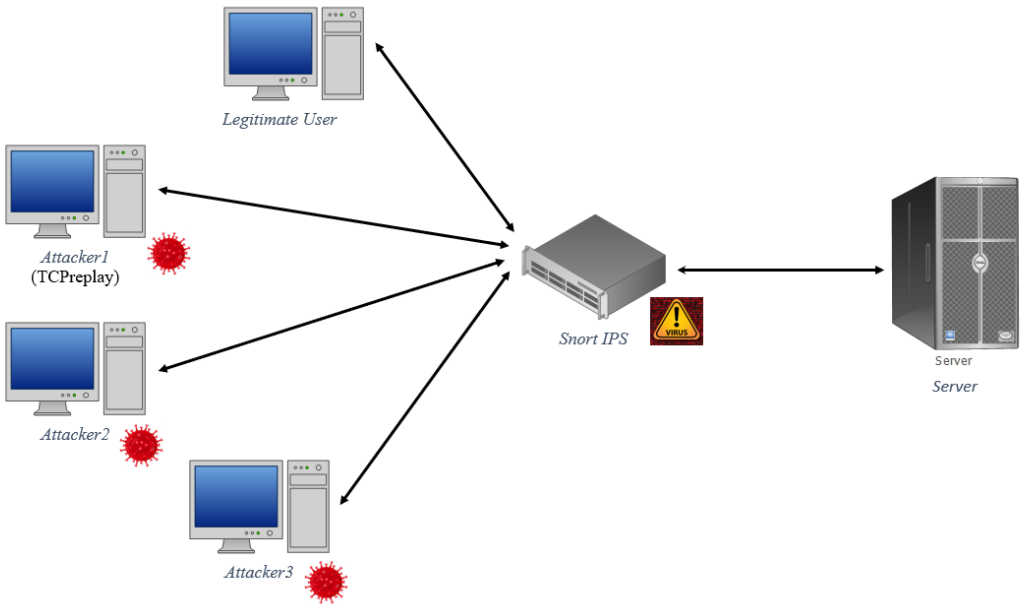

Figure 10: The arrangement of the testbed

The hardware and software requirements configuration of the machines described in Table 7 for the test environment is described in Table *8*

Table 7: The blueprint of machine design

Item	**Description**
Attacker machine	The user exploits the cloud environment improperly.
Legitimate User	The user who makes proper use of the cloud environment.
The machine configured with TCPreplay	The computer to recreate the same traffic flow over and over again.
The IPS Machine configured with Snort	An IPS router machine is used to dispose of illegal packets.
Service donator (server)	A server is set up in the cloud environment to provide web resources.

Table 8: The hardware and software configurations of machines.

Machine	**Operating System**	**CPU**	**Memory**
Server	CentOS 6.5	i7, 1.8 GHz	1024 Mbytes
Snort IPS	CentOS 6.5	i7, 1.8 GHz	1024 Mbytes
Tcpreplay	CentOS 6.5	i7, 1.8 GHz	1024 Mbytes
Attacker 1	CentOS 6.5	i7, 1.8 GHz	1024 Mbytes
Attacker 2	CentOS 6.5	i7, 1.8 GHz	1024 Mbytes
Legitimate User	CentOS 6.5	i7, 1.8 GHz	1024 Mbytes

6.3.2. Construction of two variants of traffic production, legal and non-legal components.

We must maintain a responsibility to ensure that both IPS techniques are valued at all times while maintaining equally real-time traffic. Thus, there are two ways to create congestion equally. Namely, "**Capture a traffic data set and keep it within a database**" and "**Use available data set for general use**". The author has decided to generate and regenerate the congestion because there are so many suitable dental sets and they are at a better level than DARPA and BRO [43]. Thus, since tools such as Harpon cannot generate real-time traffic, TCPreplay [43] [62] is used because it can work with a variety of options with different parameters, such as network speed.

6.3.3. Determining the evaluation parameters.

This IPS system evaluation is empirically tested under three-parameter dimensions [43] and the description of the parameters is given in Section 5.2.1.5.

- **Metrics of Input:** Under this classification, response time, availability of bandwidth, reliability, and delay are presented.
- **Metrics of Resource:** This classification presents the amount of memory usage and load on the CPU.
- **Metrics of Response:** In this classification, the packet loss rates such as false negatives and false positives are subject to be considered.

The above criteria were chosen based on the Sommers et.al [63] and Buchanan et.al [43] researchers. That is, the paradigms used to assess IDS should thus be the same as those used in IPS evaluation, as they are similar because they have the same working environment. They are also the most commonly used metrics. Therefore, the most commonly used parameter mentioned above, the standard used for IDS evaluation, is also helpful for IPS evaluation. Furthermore, these parameters are widely used and are synthesized by literary exploration.

6.4. This section Provide implementation details of each module that is stated in the design diagram.

According to figure 10 the server, Snort IPS router, attacker 1, attacker 2, and legitimate user (all five) machines are implemented either in a single virtual environment as in figure-08 (inside single host) or combined virtual environment. All virtual machines in the same host are connected with a virtual switch in the host or combined with virtual switches in the different host interconnected and configured to the single virtual switch network environment. Therefore, the traffic coming to the virtual environment are on the same network and are captured by Snort IPS then directed to detection & prevention process. Therefore, both general and attack traffic is in the same virtual network environment and then can be detected by IPS. Further, any particular network traffic can be captured by configured machine and store in a database for replaying. Furthermore, the stored traffic can regenerate (replay) using the TCPreplay tool configured machine in the same virtual network environment. The server machine is the web server in which a simple web page is stored in an FTP server configured in the same machine and is used as the server machine which is targeted by the two attacker machines and requests service by the legitimate user. All the legitimate and attack traffics are generated toward the server machine and is considered to pass through the Snort IPS router as it is in the same virtual switch environment as in Figure *10*. The legitimate user tries with legitimate traffic generated by JMeter tool to connect to the web page in the server and attackers targeting to attack with DDoS traffic using LOIC tool to down the server to prohibiting access the server by the legitimate user.

If the rule is triggered when meeting the rate and then all the packets are dropped by the IPS and otherwise pass. Then at the server machine, each selected evaluation mastics have measured separately against each three-testing attack protocol type. Finally tabulates the data gathered and form graphs using data and evaluate according to the selected statistical test namely "Wilcoxon signed-rank test which found from the literature study to evaluate the performance of rules and both approaches of Snort techniques destination and source.

6.5. The Implementation of Snort IPS

In this experimental research, the packets are analyzed by the Snort tool through two rule strategies. Packets stored in an iptables queue are analyzed by running Snort inline mode. Here the author performs the packet analysis by running the following two commands in orderly:

Iptables -A FORWARD -j NFQUEUE –queue-num 0

Snort -Q –daq nfq –daq-mode inline –daq-var queue=0 -c /etc/snort/snort.conf -A console -l /var/log/snort

The above commands confirm the validity of the rules contained in the Snort.conf file, print the output on the console, place a copy in the log file through the commanding Snort to execute inline mode. Also, the -Q here commands Snort to pick up the packets in the queue and read them.

The rules used in this section are stored in a file named DDoS.rules, which is stored in the / var/log/snort/location. The rules for filtering unwanted traffic packets are stored inside the Snort.conf file.

6.6. Generating the legal and illegal data traffic flows

Section 5.2.1.1 of the Discussion on Design mentions the use of the LOIC tool on the creation of malicious traffic flow. In this case, the tool handles the attack traffic using the server's IP address. Here the attack mode is obviously selected through the GUI interface. Figure 11 illustrates how to create TCP packet flooding using the LOIC tool.

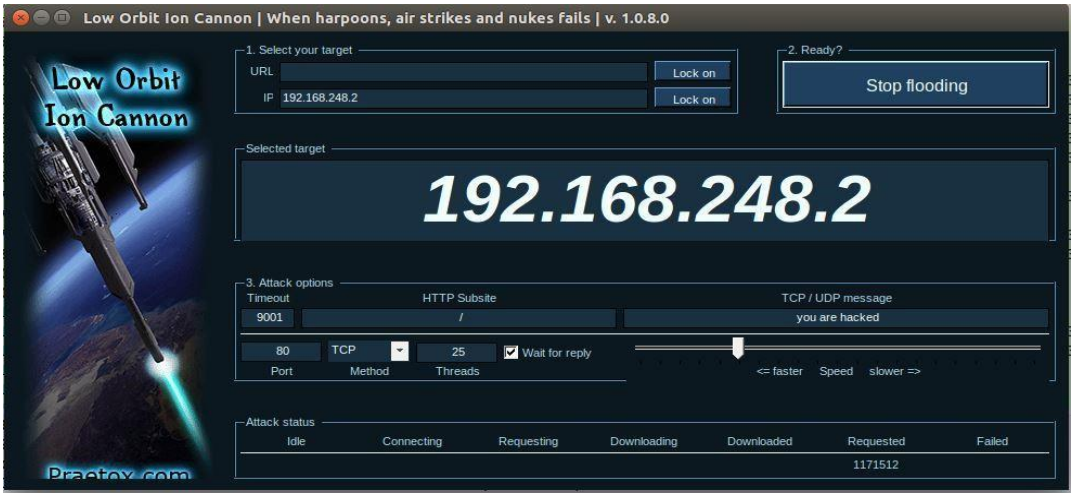

Figure 11: The use of LOIC tool to generate a TCP protocol attack to the server

Figure 12 shows how the JMeter tool is used to create a legitimate traffic stream. The mechanism of action here is based on the number of loops, user threads and ramp-up duration. The author has chosen an attack mechanism consisting of two user threads with a ramp-up duration of 4 seconds and with a loop count of 5 seconds for this experimental research.

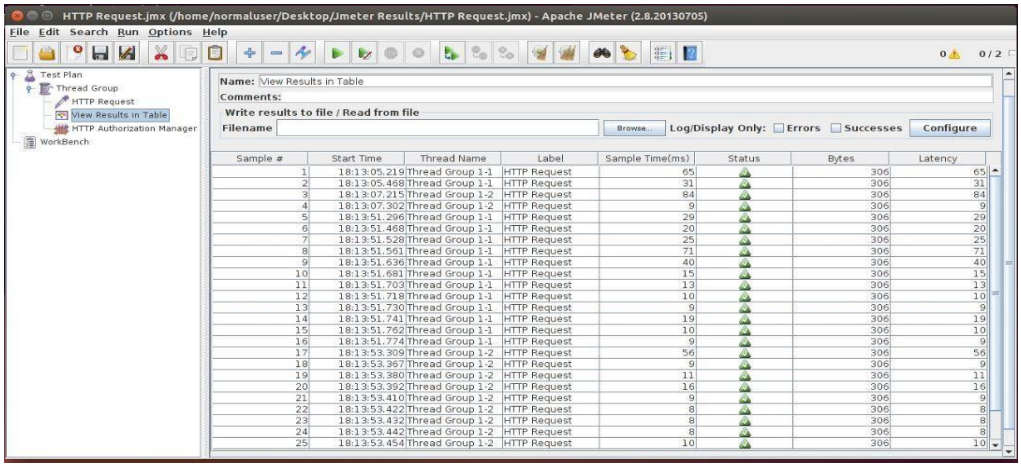

Figure 12: The use of JMeter tool to generate legitimate traffic flow toward the server

The legitimate and illegal traffic streams generated by the tool are captured by a tool called TCPreplay and then re-enabled. The Wireshark tool also captures data packets and saves them as a ".pcap" type file.

The command is given to replay the captured traffic as shown below:

sudo tcpreplay -i eth0 -p 30000 -L 200000 –loop 5 final.pcapng

To estimate the delay, the trial is set to loop traffic as it should run longer. The traffic restart command is set as follows:

sudo tcpreplay -i eth0 -p 30000 -L 2000000 –loop 100 final.pcapng

The Wireshark tool on the unit that installs the IPS system is used to capture and secure traffic, and then use it to repeat the attack traffic again and again toward the target.

6.7. The Implementation of Metrics

- **The Packet Loss Rate:** Using the Wireshark tool, packets coming to the server via the IPS system are analyzed. Lost_segment packets are analyzed to indicate the value of packet loss, ie packet loss, within the network. Using the Hping3 tool, the packet damage rate is calculated and further confirms the resulting extract from the Wireshark. The Hping3 tool transmits hundreds of packets of data per second, one second apart, when the IPS system is under attack. The Hping3 tool also displays the percentage of packet loss. As a hypothesis, the rate of packet damage is considered to increase as the throughput of the attack traffic increases.

- **The Availability of Bandwidth:** Bandwidth is calculated using the Iperf tool, which is used to calculate the bandwidth of an IPS system in the event of an attack. The TCP_STREAM test is used to calculate the bandwidth of an IPS system. The time it takes to transfer TCP packets from the Iperf client to the Iperf server is used to calculate the throughput available here, called the TCP_STREAM test.
  The Iperf toll used the following command:

  Iperf -c <ip address of Iperf server>

  The following command used in the Iperf server:

  Iperf -s

- **The Latency:** The Latency parameter is used to measure Round-Trip-Time (RTT), and the delay measurement is measured by the author using the ping [64] tool. First of all, 100 packets are sent over the network to the server through the IPS system. The average time elapsed for the packets accepted by the client is then calculated. The ping command is used for this purpose as follows.

  Ping 192.168.248.2 -c 60

- **The Throughput of data:** At a time of under the attack of the system (server), to calculate the throughput of the IPS is used the Wireshark tool.
- **Reliability:** The IPS system is assessed for its ability to withstand a continuous attack for up to 5 hours and then restart the illegal traffic with the TCPreplay tool.
- **The Load of CPU:** Due to malicious traffic, the uptime tool in Unix is used to calculate the impact on system resources, and in this experiment, the tool is used to calculate the CPU load.
- **The Utilization of Memory:** The "free" tool in Unix is used to calculate the impact of the memory consumption parameter on attack traffic.

The evaluation metrics of this experiment are measured using specific tools relevant to particular machines and the metrics, tools used and machine names are described in Table *9* and then calculated the metric value accordingly.

Table 9: Application locations for each tool to assess the metrics in the testbed

Metrics	**Tool**	**Machine**
Bandwidth	Iperf	Tcpreplay tool
Packet Loss percentage	Wireshark, Hping3	IPS system based on Snort, Tcpreplay tool
Throughput of Data Packets	Wireshark	IPS system based on Snort
Load of the CPU	Uptime	IPS system based on Snort
Delay (Latency)	Ping	Tcpreplay tool
Utilization of the Memory	Free	IPS system based on Snort

6.8. Summary

This chapter presented the implementation aspects of the performance evaluation of the IPS based on Snort rules to protect the virtual environment from DoS and DDoS attacks with TCP, UDP and HTTP protocols. The next chapter describes the results of the performance evaluation of each implementation in the above protocol attacks with the selected evaluation metrics.

Chapter 07

Evaluation and Discussion

Evaluation of the Snort Rule-Based IPS with the Implemented Evaluation Metrics.

7.1. Introduction

The previous chapter described the implementation of each component regarding performance measurement of the IPS with selected metrics against different DoS & DDoS attacks regarding TCP, UDP, HTTP protocols for both source and destination approaches. This chapter describes how the implementation has been tested and evaluated to identify the performance enhancement of the Snort rule-based IPS. Initially, the implemented four modules have been tested for their functionality performance evaluation. Later, performance metrics have been measured and presented in the table and graphical formats.

This chapter discusses how to evaluate the solution and how the author achieves his or her goals accordingly.

7.2. Examination of the deployment of the Experiment.

The deployment of the core of the design architecture which includes three significant models have been tested to verify its functionality in the virtual environment. The test results are given in the appendix – D. Test results ensured the implementation of performance evaluation of the IPS is correct.

7.3. Evaluating Strategy

To increase the accuracy in the measurement of execution times, the following matters have been considered in measurements.

- Evaluations have been performed on two identical Virtual Machines having no unnecessary applications or devices. Then results obtained in the two machines have been compared for any considerable differences.
- The minimum size of input data has been taken as large enough for its execution time has be more than 40 seconds.
- Measuring twenty execution times, average times have been considered for evaluation.
- During the measurement, the first measurements have not been considered to find the average to avoid any start-up times.

7.4. For both source and destination approaches, finding the Average time of the execution for deployment with distinct protocols.

The implementation results of the following selected metrics for the evaluation has been elaborated as follows according to the explanations in sections 6.5, 6.6 and 6.7.

To safeguard the testbed surrounding is similar to assessing and performing the following:

- The data collected for each parameter for the two IPS technical strategies are then stored in the machine for later analysis and the machine is restarted.
- Each iteration of the experiment is repeated ten times and the mean value is calculated and recorded in the results section.
- Each repetition of the experimental process is repeated ten times and the mean value is recorded.

7.5. Results representation

7.5.1. The load of the CPU

Depending on the weight of the packet processing in the IPS system, the load on the CPU varies. Attacks using the TCP protocol use the least amount of resources compared to other attacks. Attacks using UDP packets or composite packets place a heavy load on the CPU of about 75%, while attacks using TCP packets reduce resource usage by 55% to 65%. This may be due to the fact that TCP packets are processed in a relatively short period and the time taken to process other packets is slightly longer. Furthermore, it is clear that the packet rate is maintained between 8000pps and 1000pps when the load on the CPU is kept constant (55% to 58%). That is, here, when a ratio is reached, it can be assumed that the load on the CPU is stable, as Snort performs some optimization process. This is because the UDP protocol takes a much longer processing time, and the mix flood follows the UDP protocol.

As shown in Figure 13 and Figure 14 of source and destination approaches, respectively through the data tables Table *10* and Table *11*, the graphical representation of the CPU load results for the attacks on the various protocols are given with “Scatter with smooth/straight lines and Markers” graphs as in *Figure 13* and *Figure 14*.

Table 10: The load on the CPU for the source approach

The load on the CPU for TCP, UDP, HTTP & MIX floods in the Source approach					
	PPS	TCP Source (%)	UDP Source (%)	HTTP Source (%)	MIX Source (%)
1	0	10	10	10	10
2	2000	19	12.1	29.8	21
3	4000	30	16.5	38.1	38.1
4	6000	57.4	55.2	66.9	63.3
5	8000	54	58.1	65.5	62.2
6	10000	53.2	57.6	64.9	62
7	12000	53	55.1	64.3	61.8
8	14000	53.5	52.3	63.9	60.7
9	16000	53	52.9	63.1	60.1
10	18000	52.8	53.5	62.8	59.8
11	20000	52.1	58.9	62.1	59
12	22000	51.6	59.2	61.2	58.2
13	24000	49.2	62.9	59.7	60.3
14	26000	49.6	63.2	59.1	63.2
15	28000	49.1	66.1	58.2	66.9
16	30000	48.4	68.1	57.9	69.9

Table 11: The load on the CPU for the destination approach

The load on the CPU for TCP, UDP, HTTP & MIX floods in the Destination approach					
#	PPS	TCP Destination (%)	UDP Destination (%)	HTTP Destination (%)	Mix Destination (%)
1	0	10	10	10	10
2	2000	20.1	14.2	30.1	25.1
3	4000	32	16.8	39.1	50.8
4	6000	58	55.4	68.5	66.5
5	8000	55	58.9	66.2	64.9
6	10000	54.9	58.2	65.8	63.2
7	12000	54.6	56.7	65.3	63.9
8	14000	53.8	52	64.2	62.9
9	16000	52.8	52.5	63.9	62.1
10	18000	52.5	53.2	63.5	61.7
11	20000	52	58.5	63.1	61.1
12	22000	51.8	59.8	62.8	59.9
13	24000	50	63.1	62.6	60.5
14	26000	49.7	64.2	61.1	64.3
15	28000	49.2	66.5	60.8	67.2
16	30000	48.6	68.9	60.2	70.2

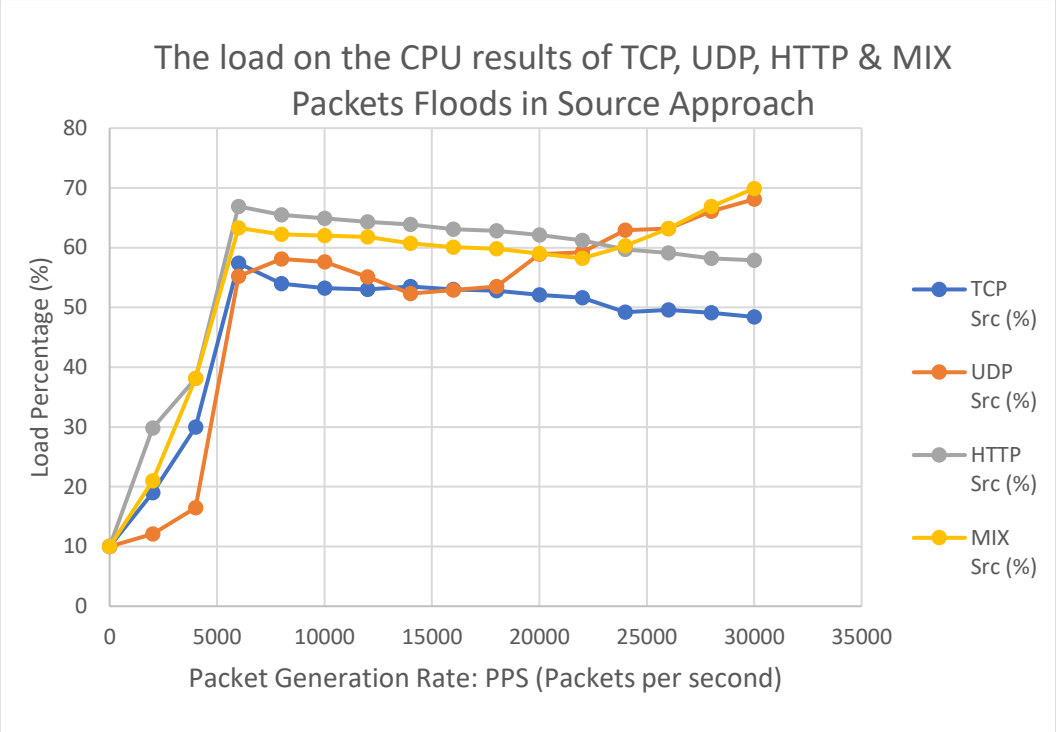

Figure 13: The load on the CPU Results in Source Approach

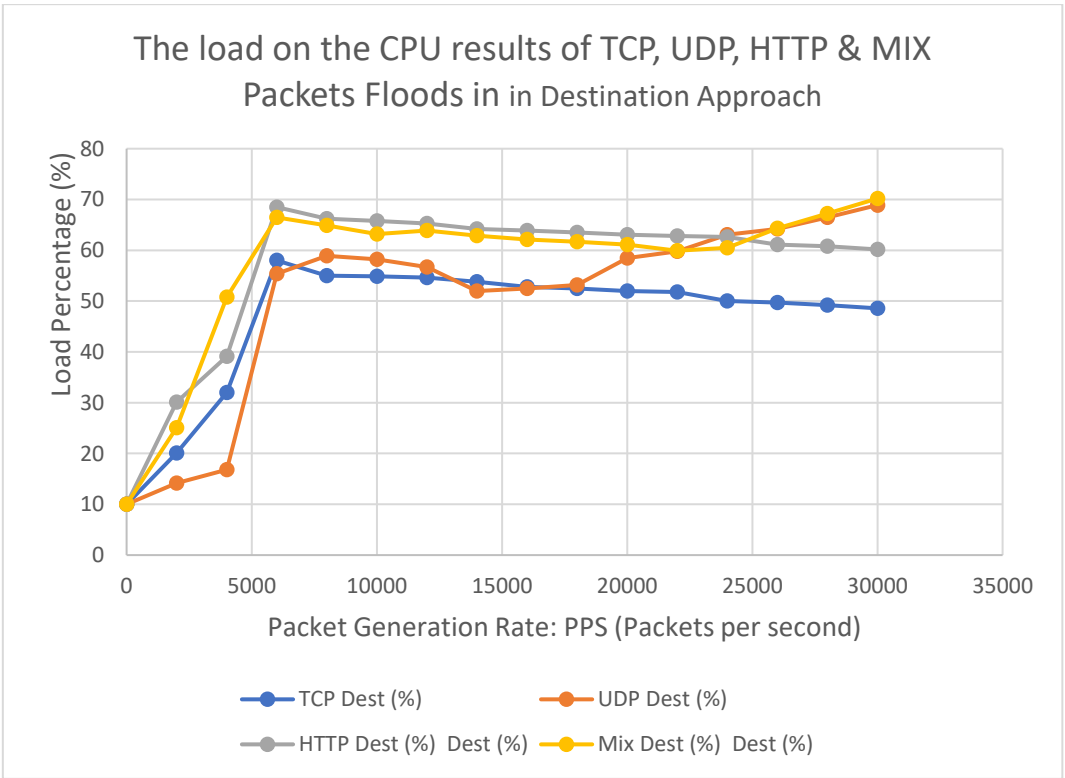

Figure 14: The load on the CPU Results in Destination Approach

7.5.2. Utilization of the Memory

The memory utilization of each protocol is as expected in the destination approach. But there is a huge difference in memory utilization of the protocols in this approach. The difference rate is nearly 30Mbps. Moreover, the UDP attack protocol has unpredictable memory usage as it has huge variations in memory usage. When the packet rate is reached to 6000pps from 4000pps there is a huge drop out in the memory usage and then it continues in the same range of 290Mbps.
The memory utilization in the source approach is almost in the same range throughout the experiment for all protocols.

Memory usage is provided for each protocol in the destination approach as expected. However, it is clear that there is a big difference in memory usage when using different protocols for attackers. Also, the rate of change is about 30 megabits per second. Furthermore, it is clear that UDP can be referred to as a protocol for unexpected memory usage, where there is a very high variance in memory usage. Here, the UDP protocol shows a large drop in memory usage when the packet rate reaches 4000pps to 6000pps and then the memory usage ranges from 340Mbps to 240Mbps. It appears that all protocols in the source approach showed to maintain the same level of memory usage throughout the entire experimental process.

It turns out that memory usage is unstable in both source and destination approaches. This may be as a result of all the Linux Virtual Machines being on the same host computer. This environment is unsuitable for predicting memory, as the host computer's effect on the virtual machine's memory.

As shown in Figure *15* and Figure 16 of source and destination approaches, respectively through the data tables Table *12* and Table 13, the graphical representation of the “Utilization of Memory” results for the attacks on the various protocols are demonstrated.

Table 12: The utilization of memory in source approach

The utilization of memory for TCP, UDP, HTTP and Mixed floods in the source approach					
#	PPS	TCP destination Mbps	UDP destination Mbps	HTTP destination Mbps	Mix destination Mbps
1	0	316	317	313	287
2	2000	321	322	319	289
3	4000	336	341	325	296
4	6000	332	292	330	319
5	8000	339	290	325	300
6	10000	342	289	333	320
7	12000	335	290	336	320
8	14000	335	285	335	316
9	16000	339	288	341	315
10	18000	337	286	338	314
11	20000	340	300	345	315
12	22000	335	255	335	315
13	24000	336	287	338	317
14	26000	337	289	338	318
15	28000	337	289	339	319
16	30000	335	291	337	318

Table 13: The utilization of memory in the Destination approach

The utilization of memory for TCP, UDP, HTTP and Mixed flood in the Destination approach					
#	PPS	TCP destination Mbps	UDP destination Mbps	HTTP destination Mbps	Mix destination Mbps
1	0	315	316	312	286
2	2000	320	321	318	288
3	4000	339	340	322	295
4	6000	335	290	328	300
5	8000	339	288	328	305
6	10000	340	289	331	318
7	12000	338	288	335	316
8	14000	336	288	338	315
9	16000	338	287	340	314
10	18000	338	287	339	315
11	20000	338	287	342	315
12	22000	337	288	340	316
13	24000	336	289	339	318
14	26000	335	289	335	318
15	28000	334	289	339	319
16	30000	334	290	336	320

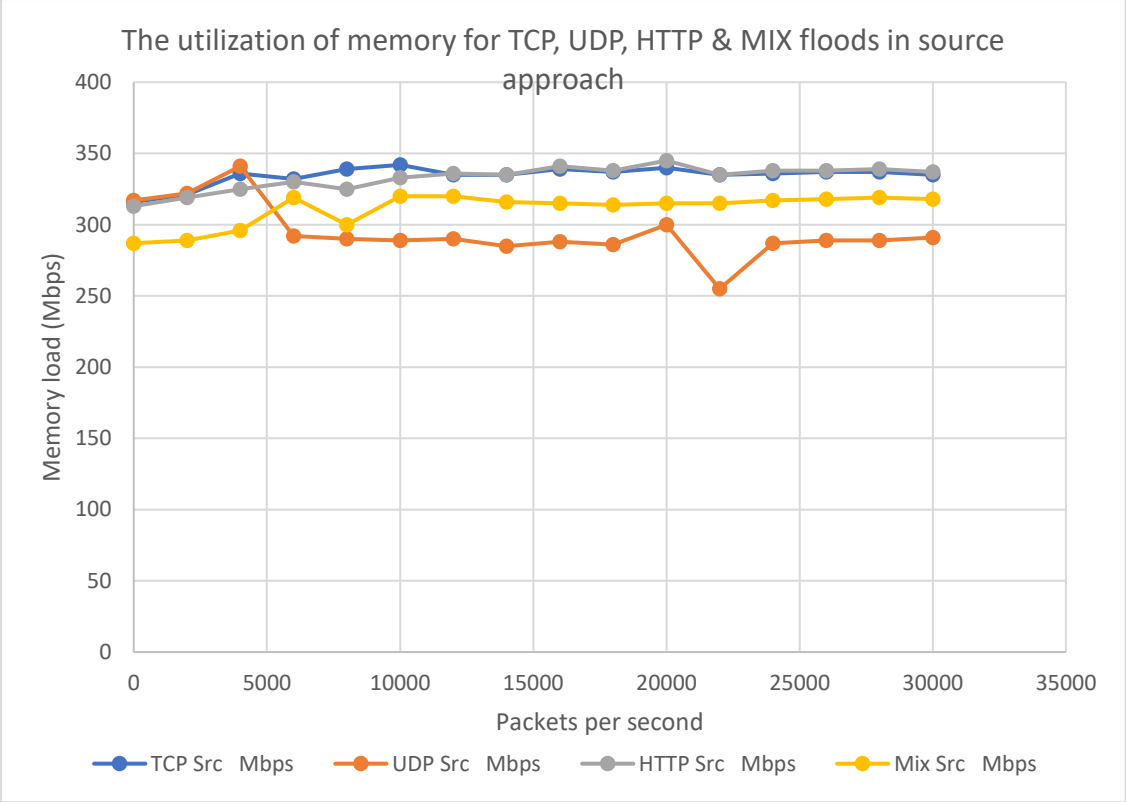

Figure 15: The utilization of memory in source approach

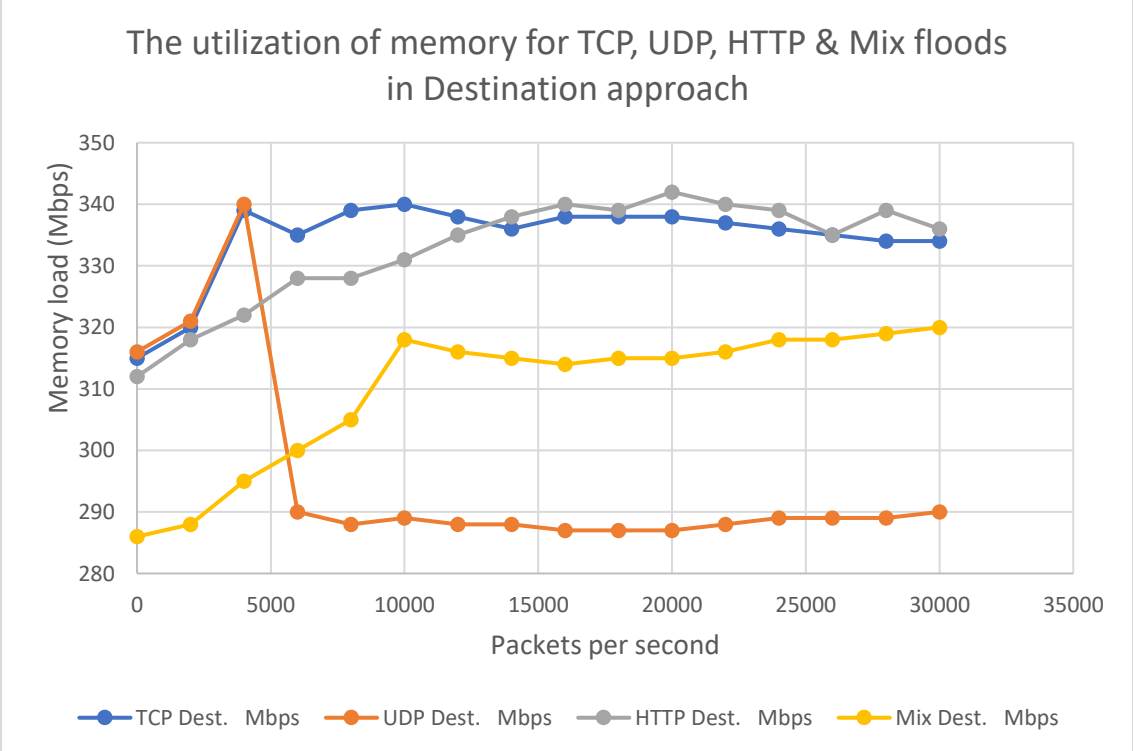

Figure 16: The utilization of memory in destination approach

7.5.3. Availability of the Bandwidth

The bandwidth in the network was about 60Mbps when there was no traffic. As the network loads, the bandwidth gradually decreases and the bandwidth drops to zero as the packet rate approaches 14000pps. At around 14000pps of traffic, bandwidth seems to be unavailable in the destination approach with Snort.

As shown in Figure *17* and Figure *18* of source and destination approaches, respectively through the data tables Table *14* and Table 15, the graphical representation of the "Availability of Bandwidth" results for the attacks on the various protocols are demonstrated.

Table 14: Availability of Bandwidth in Source approach

	Availability of Bandwidth for TCP, UDP, HTTP & Mix floods in Source approach				
#	Pps	TCP Src Mbps	UDP Src Mbps	HTTP Src Mbps	Mix Src Mbps
1	0	60.3	60.3	60.3	60.3
2	2000	36.2	41.25	38.861	34.9
3	4000	19.7	21.35	25.63	24.21
4	6000	11.9	11.63	11.5	14.9
5	8000	8.89	7.32	9.89	10.6
6	10000	7.08	4.861	6.46	5.62
7	12000	4.168	4.19	5.39	4.92
8	14000	2.15	1.23	2.62	2.98
9	16000	1.16	1.06	1.35	1.22
10	18000	0.896	0.589	0.795	0.44
11	20000	0.705	0.109	0.599	0.42
12	22000	0.507	0	0.365	0.39
13	24000	0.129	0	0.1	0.106
14	26000	0.105	0	0.076	0.059
15	28000	0.08	0	0.052	0.079
16	30000	0.054	0	0.013	0.01

Table 15: Availability of Bandwidth in Destination approach

Availability of Bandwidth for TCP, UDP, HTTP & Mix floods in Destination approach					
#	PPS	TCP Dest Mbps	UDP dest Mbps	HTTP dest Mbps	MIX dest Mbps
1	0	60.3	60.3	60.3	60.3
2	2000	37.2	41.25	40.2	35.5
3	4000	20.95	21.35	24.4	22.53
4	6000	12.7	11.63	11.5	13.5
5	8000	9.89	7.32	8.98	6.23
6	10000	7.08	4.861	9.89	8.61
7	12000	4.168	4.19	5.28	4.35
8	14000	1.256	1.23	2.125	2.31
9	16000	1.076	1.06	1.18	1.62
10	18000	0.896	0.589	1.076	0.49
11	20000	0.507	0.109	0.557	0.47
12	22000	0.119	0.04	0.1	0.106
13	24000	0.102	0	0.076	0.079
14	26000	0.086	0	0.066	0.053
15	28000	0.07	0	0.0369	0.01
16	30000	0.054	0	0.0296	0

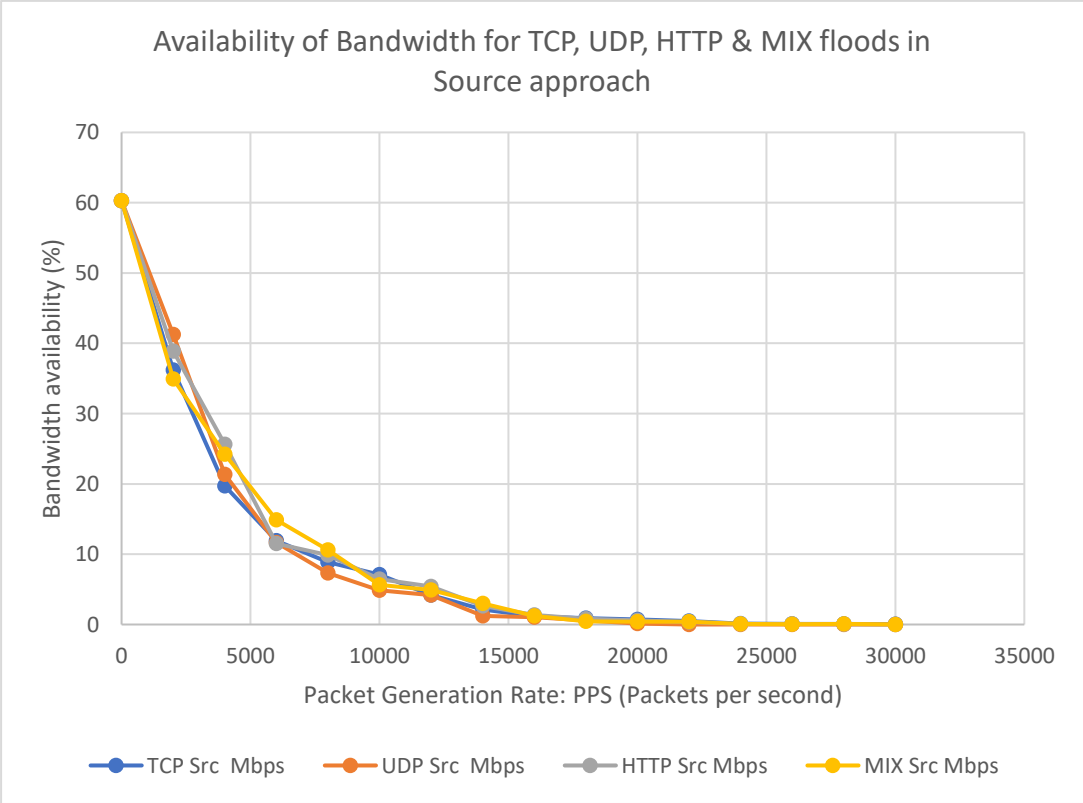

Figure 17: Availability of Bandwidth results in source approach

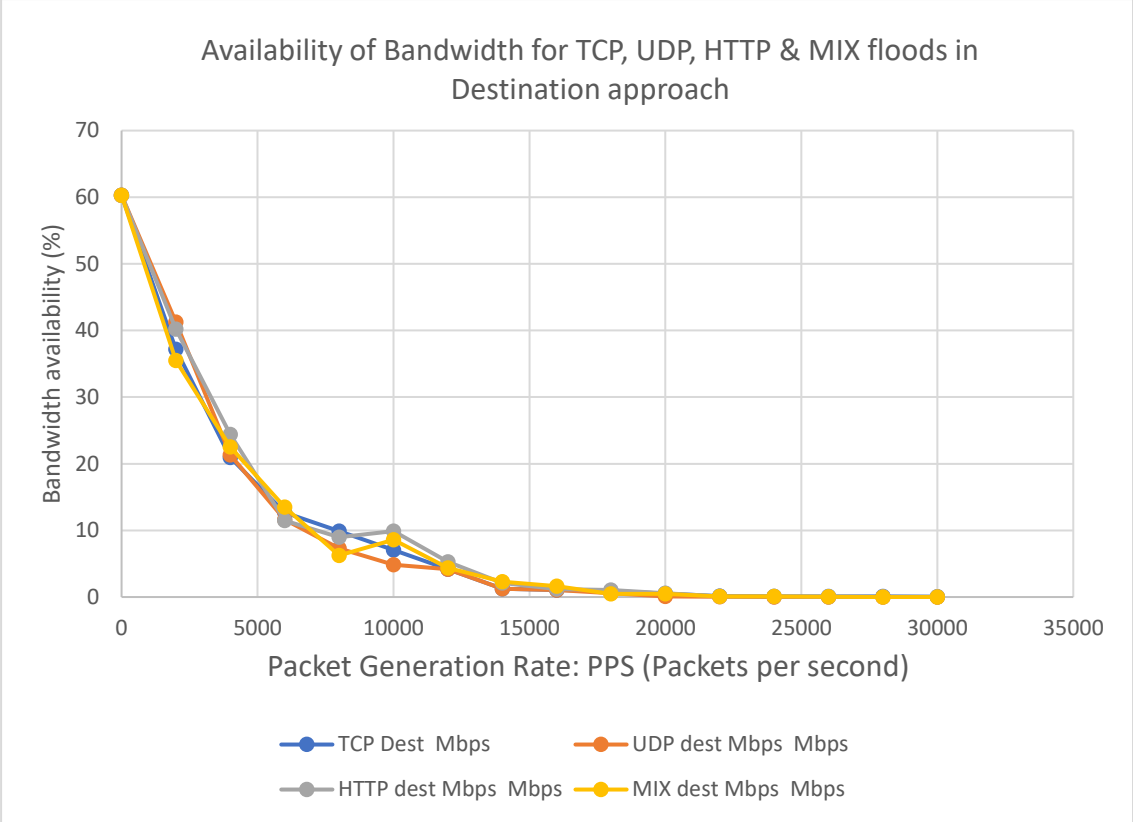

Figure 18: Availability of Bandwidth results in destination approach

7.5.4. Delay (Latency)

The reason for the increase in delays can be attributed to the increase in traffic speed within the network. The delay of all protocols is about 5ms and all attack protocols are subject to the same continuous range when the rate is around 2000pps. It is clear that the delay in destination access is very important and optimally reduces the traffic rate to between 10000pps and 24000pps for all protocols.

As shown in Figure *19* and Figure *20* of source and destination approaches, respectively through the data tables Table *16* and Table 17, the graphical representation of the "Delay (Latency)" results for the attacks on the various protocols is demonstrated.

Table 16: Delay (Latency) results in source approach

Presence of Delay (Latency) for TCP, UDP, HTTP & MIX floods in Source approach					
#	PPS	TCP - Milli seconds (ms)	UDP - Milliseconds (ms)	HTTP - Milli seconds (ms)	MIX - Milli seconds (ms)
1	0	1.876	1.876	1.876	1.876
2	2000	1.994	2.113	1.958	2.548
3	4000	2.215	2.659	2.324	2.245
4	6000	2.425	3.123	3.245	1.127
5	8000	37.765	38.657	36.324	31.548
6	10000	32.453	38.898	37.965	33.731
7	12000	32.899	38.834	38.335	34.234
8	14000	30.993	35.357	38.992	34.992
9	16000	29.899	34.924	37.125	34.212
10	18000	30.523	35.254	36.895	34.324
11	20000	31.626	36.215	36.325	33.752
12	22000	31.898	36.751	36.795	34.987
13	24000	32.628	36.148	36.982	35.578
14	26000	32.958	35.851	36.743	36.245
15	28000	33.485	35.519	36.143	37.147
16	30000	33.412	36.112	35.124	39.125

Table 17: Delay (Latency) results in destination approach

Presence of Latency availability for TCP, UDP, HTTP & MIX floods in Destination approach					
#	Pps	TCP - Milli seconds (ms)	UDP - Milli seconds (ms)	HTTP - Milli seconds (ms)	MIX - Milli seconds (ms)
1	0	1.876	1.876	1.876	1.876
2	2000	2.125	2.659	1.958	3.548
3	4000	2.993	3.123	2.324	4.245
4	6000	3.632	3.985	2.955	2.127
5	8000	40.765	39.657	37.324	32.548
6	10000	35.765	40.895	38.965	34.731
7	12000	33.899	39.834	39.335	35.234
8	14000	31.993	36.357	39.992	36.129
9	16000	30.889	35.924	38.125	35.421
10	18000	31.523	36.245	37.895	35.324
11	20000	32.626	36.954	37.325	34.752
12	22000	32.898	37.751	37.759	35.987
13	24000	33.628	37.148	37.982	36.578
14	26000	33.958	36.851	37.743	37.245
15	28000	34.485	36.519	37.143	38.147
16	30000	34.995	36.425	36.941	39.357

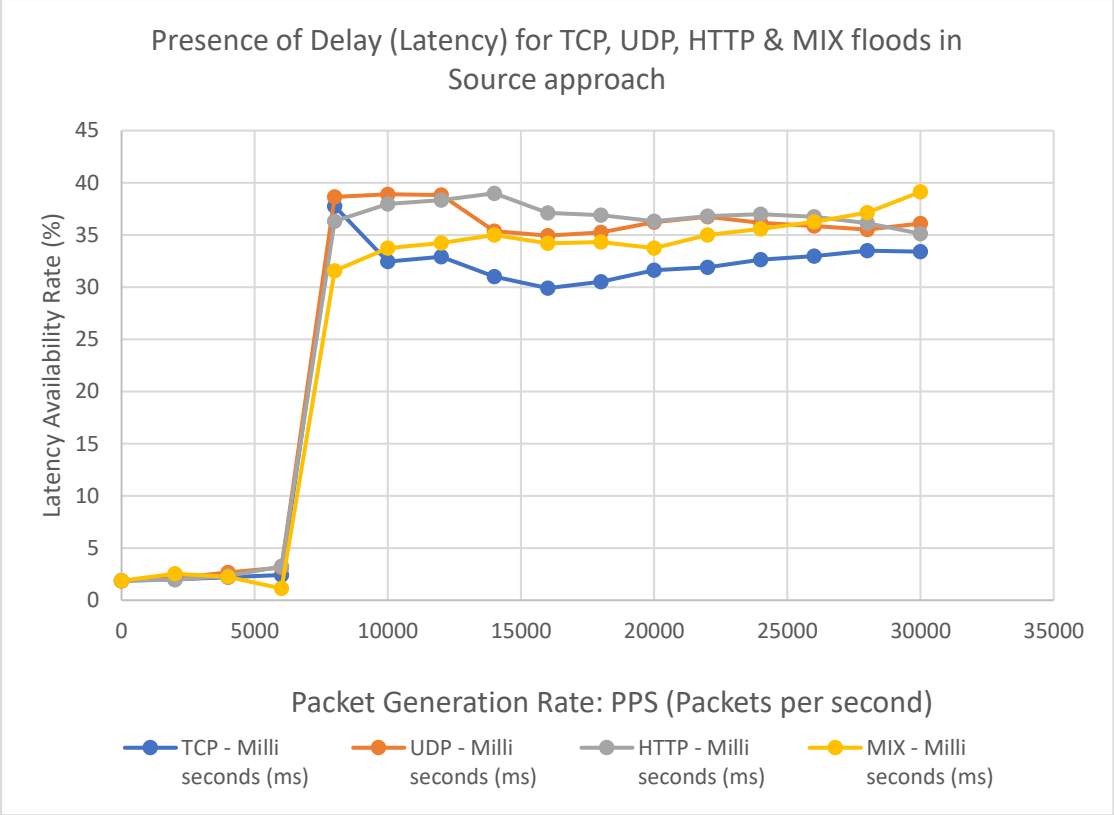

Figure 19: Presence of Delay (Latency) results in source approach

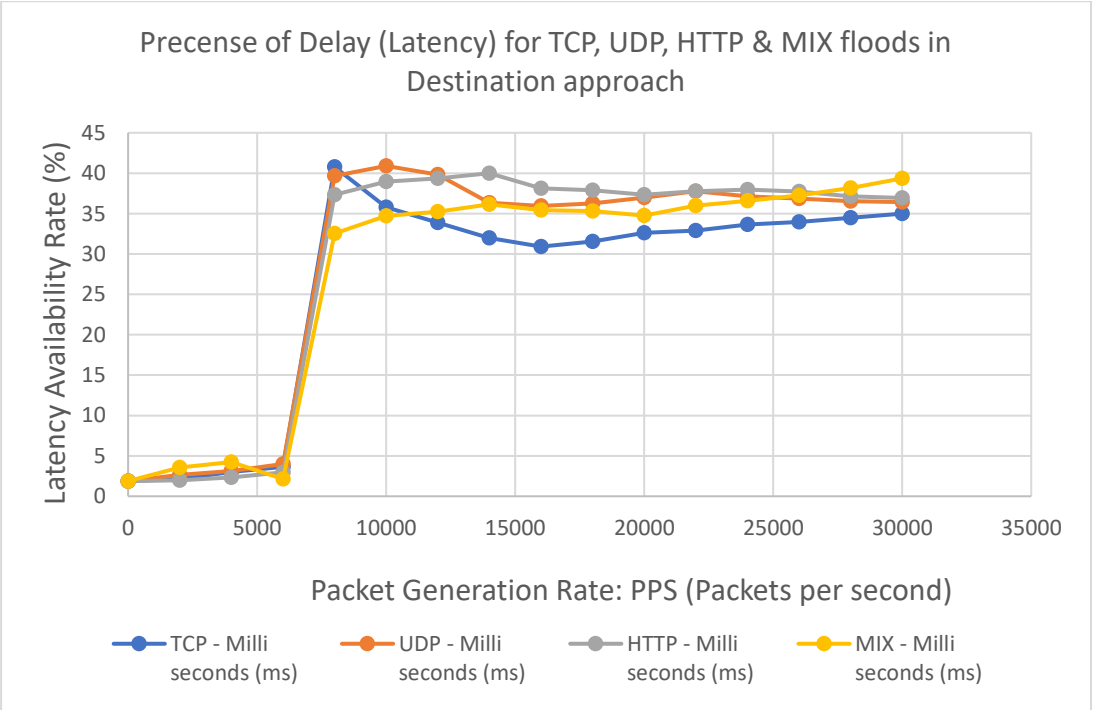

Figure 20: Presence of Delay (Latency) results in destination approach

7.5.5. Percentage of Loss of Packets

In both source and destination approaches the rate of traffic is directly proportional to the percentage of loss of packets. The speed of packet damage percentage increases as the traffic speed in the network increases. The results show that the amount of packet damage (loss) is very sharp, increasing for both source and destination approaches when the traffic rate reaches 18000pps. And, following the HTTP and UDP, the TCP protocol also showed the weakest result. It can be interpreted that it takes more time to prepare TCP packets through other TCP sessions.

Because ICMP packets take less time to process, less time to process the UDP protocol can be cited as the reason why the mixed protocol shows a lower percentage of packet damage in the source approach.

As shown in Figure *21* and Figure *22* of source and destination approaches, respectively through the data tables Table *18* and Table *19*, the graphical representation of the "Percentage of loss of Packets" results for the attacks on the various protocols are demonstrated.

Table 18: Percentage of loss of packets in Source approach

Percentage of loss of packets in TCP, UDP, HTTP & Mix floods in Source approach					
#	PPS	TCP loss % in Source Approach	UDP loss % in Source Approach	HTTP loss % in Source Approach	MIX loss % in Source Approach
1	0	0	0	0	0
2	2000	0	0	0	0
3	4000	0	0	0	0
4	6000	0	0	1	0
5	8000	12	13	13	14
6	10000	19	27	21	27
7	12000	26	32	30	31
8	14000	34	38	36	36
9	16000	39	45	47	46
10	18000	46	49	51	50
11	20000	50	55	56	54
12	22000	57	59	58	55
13	24000	66	62	61	60
14	26000	70	70	64	63
15	28000	75	72	69	66
16	30000	76	73	70	67

Table 19: Percentage of loss of packets in Destination approach

The packet loss rate in TCP, UDP, HTTP & MIX floods in Destination Approach					
#	Pps	TCP loss % in Destination Approach	UDP loss % in Destination Approach	HTTP loss % in Destination Approach	MIX loss % in Destination Approach
1	0	0	0	0	0
2	2000	0	0	0	0
3	4000	0	0	0	0
4	6000	6	4	3	1
5	8000	18	20	19	20
6	10000	27	34	30	32
7	12000	33	37	36	36
8	14000	40	42	43	45
9	16000	45	46	52	48
10	18000	54	52	56	55
11	20000	64	59	59	58
12	22000	67	63	61	60
13	24000	71	69	65	63
14	26000	74	73	69	67
15	28000	76	75	73	71
16	30000	80	77	75	72

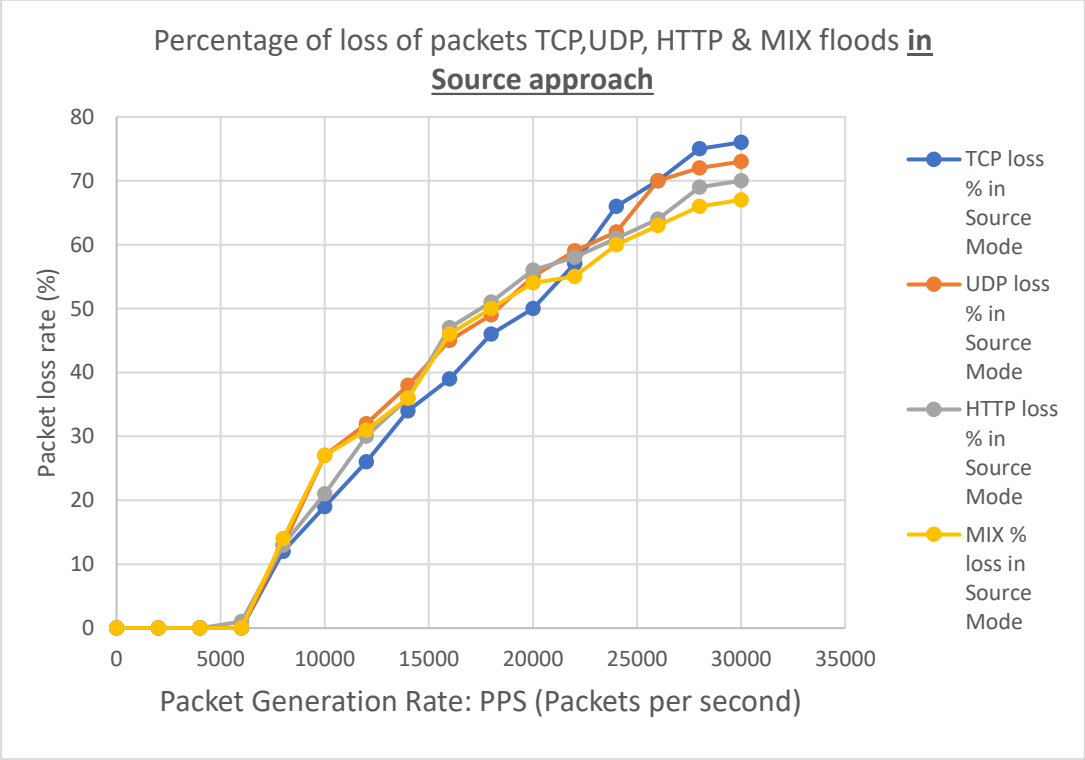

Figure 21: Percentage of loss of packets in Source approach

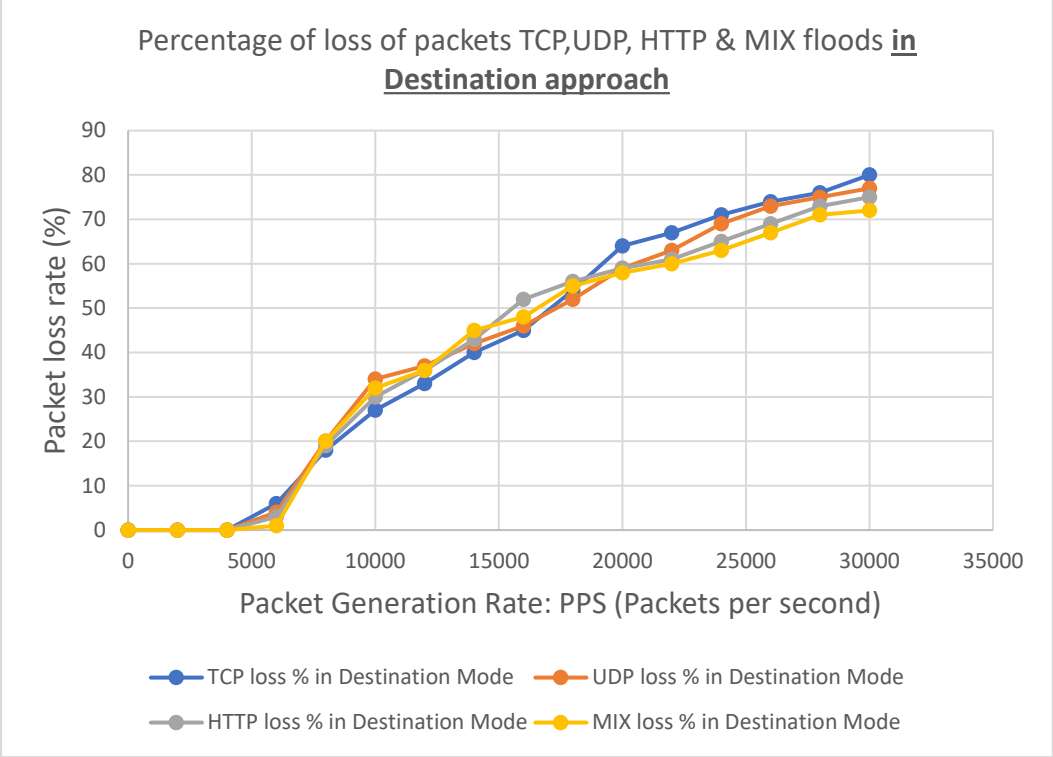

Figure 22: Percentage of loss of packets in Destination approach

7.6. Conducting the Wilcoxon Test

7.6.1. Hypothesis assumption

According to Section 5.4.2 in Chapter 05, the hypothesis for this experiment is as follows:

H_0: Snort does not show any difference between both source and destination approaches.

H_1: Snort show at least any difference between both source and destination approaches.

When X and Y are the data pairs of the source and destination approaches, the performance difference is calculated by $d = (X_i - Y_i)$. Further, the difference has given the "sign" S while +1 represent the positive values and -1 represents the negative values.

The absolute difference can be calculated and the performance reached. The 16 corresponding ranks from 1 to 16 are given because 16 populations come from a combination of the four protocol formats and the two source and destination modes. The result obtained by multiplying the data pair S and rank is the signed-rank test. The

author engaged that for all the considered parameters, R+ and R- are less than the critical value of 0.05, which is the level value of the phase value, by calculating the sum of the positive values of signed ranks (R+) and the sum of the negative values signed-ranks (R-).

Accordingly, the rejection of the null hypothesis confirms that it is reasonable for all the metrics considered in this research. That is, it implies that there is a difference between the two rate filtering approaches in Snort NIPS.

The following Table 20 is a table of statistical analysis data for both source and destination approach for CPU loads.

Table 20: Statistical analysis of both source and destination approaches for CPU loads (Wilcoxon test)

The sum of the positive values: R+	114
The sum of the negative values: R-	-17
Test statistic	17
Critical value	29

Thus, the results show that the hypothesis is rejected for all parameters. Further, the statistical analysis reveals that the two source and destination modes of Snort NGIPS differ from each other. Furthermore, for both models, there is evidence of performance differences in all the parameters used.

Considering that the sum of the negative values, i.e. the R-value, is less than the critical value, the observation of the R+ and R- values reveals that the destination approach operates better than the source approach.

7.7. Summary

This chapter describes the results obtained in the evaluation process. Also, it has expressed the evaluation strategy applied during the evaluation process. The results obtained in the evaluation are shown in the tabular form and graphical representation as “Scatters with smooth lines and markers” graphs. The next chapter will interpret the results described in this chapter in a meaningful way.

Chapter 08

Conclusion & Further Work

8.1. Introduction

The previous chapter has presented the results of the various evaluations regarding the functionality of the implementation and execution times of various designs. Results have been expressed in tabular and graphical forms. This chapter interprets the results obtained in the evaluation chapter.

8.2. Conclusion

The author attempts to compare two types of rate filters in Snort NIPS technology, a tool commonly used in networks to protect the cloud environment from DoS and DDoS attacks. This led to a better model evaluation based on the parameters. Statistical analysis is also performed to validate the data.

The results of the literature review show that the focus is on controlling many DoS and DDoS attacks through the detection of network-based attacks.
NIPS infiltrates the network for malicious packet traffics and uses signature-based methods for the NGIPS.

Implementing real-time DoS and DDoS strategies may be somewhat complicated in this research environment, but Snort's two approaches to rate_filter technologies support the validation of the results of these experimental attack protocols.

8.3. Problems encountered, limitations of the solution, and some further works.

Experimental results show that Snort is capable of successfully manipulating both DoS and DDoS attack types, up to 30000pps, for both source and destination access. The main disadvantage of Snort is that the so-called true-positive legal packets are dropped along with the non-legal packets. As a result, legitimate clients are blocked from accessing the server to access the service.

Both models have been shown to handle attacks for up to four hours with a high degree of reliability. But other parameters are not taken into consideration here. Therefore, it is important to note that over time, legal packets may drop along with non-legal packets.

In the future, this work can be extended by considering other attack tools in the same testbed and evaluating the results. The testbed designed during this experiment had enough capability to be tested even with other data sets like DARPA and bro. It can also be implemented with other types of DoS and DDoS attacks and other metrics implemented in the same testbed.

Other attack tools can be used to extend this research to a result evaluation using the same experimental environment. This research environment also has the potential to be tested using datasets such as DARPA and Bro. Furthermore, this experiment can be tested with other parameters as well as with different kinds of other DoS and DDoS attacks.

8.4. Summary

This chapter interprets the results obtained in the previous chapter, propose an architecture for efficiency and have done the performance evaluation of the NGIPS in a virtual environment to protect from DoS and DDoS attack and suggests future research to using other available data sets, other types of DDoS attacks and for other parametric.

References

[1] P. Yadav and S. Sujata, “Security Issues in Cloud Computing Solution of DDOS and Introducing Two-Tier CAPTCHA,” *IJCCSA*, vol. 3, no. 3, pp. 25–40, Jun. 2013, DOI: 10.5121/ijccsa.2013.3303.

[2] K. Hwang, S. Kulkarni, and Y. Hu, “Cloud security with virtualized defence and reputation-based trust management,” 2009, pp. 717–722.

[3] A. Chonka, Y. Xiang, W. Zhou, and A. Bonti, “Cloud security defence to protect cloud computing against HTTP-DoS and XML-DoS attacks,” *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1097–1107, 2011, DOI: 10.1016/j.jnca.2010.06.004.

[4] M. Rahman and W. Man, “A Novel Cloud Computing Security Model to Detect and Prevent DoS and DDoS Attack,” *ijacsa*, vol. 5, no. 6, 2014, DOI: 10.14569/IJACSA.2014.050618.

[5] D. Sequeira, “Intrusion Prevention Systems- Security’s Silver Bullet?” p. 14, 2002.

[6] S. Piper, CISSP, and SFCP, *Intrusion Prevention Systems For Dummies®*. Willey Publishing Inc., 2011.

[7] N. Kilari and Dr R. Sridaran, “An Overview of DDoS Attacks in Cloud Environment,” in *International Journal of Advanced Networking Applications (IJANA)*, 2014, pp. 124–127.

[8] A. Harper, *Gray hat hacking: the ethical hacker’s handbook*. New York: McGraw-Hill, 2011.

[9] R. V. Deshmukh and K. K. Devadkar, “Understanding DDoS Attack & its Effect in Cloud Environment,” *Procedia Computer Science*, vol. 49, pp. 202–210, 2015, DOI: 10.1016/j.procs.2015.04.245.

[10] S. Subashini and V. Kavitha, “A survey on security issues in service delivery models of cloud computing,” *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 1–11, 2011, DOI: 10.1016/j.jnca.2010.07.006.

[11] A. Carlin, M. Hammoudeh, and O. Aldabbas, “Defence for Distributed Denial of Service Attacks in Cloud Computing,” *Procedia Computer Science*, vol. 73, pp. 490–497, 2015, DOI: 10.1016/j.procs.2015.12.037.

[12] N. Gruschka and M. Jensen, “Attack Surfaces: A Taxonomy for Attacks on Cloud Services,” in *2010 IEEE 3rd International Conference on Cloud Computing*, Miami, FL, USA, 2010, pp. 276–279. DOI: 10.1109/CLOUD.2010.23.

[13] S. M. Specht and R. B. Lee, “Distributed Denial of Service: Taxonomies of Attacks, Tools, and Countermeasures.,” The Canterbury Hotel, San Francisco, California, USA, 2004.

[14] A. Patel, M. Taghavi, K. Bakhtiyari, and J. Celestino Júnior, “An intrusion detection and prevention system in cloud computing: A systematic review,” *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 25–41, 2013, DOI: 10.1016/j.jnca.2012.08.007.

[15] B. B. Gupta, R. C. Joshi, and M. Misra, “Distributed Denial of Service Prevention Techniques,” *IJCEE*, pp. 268–276, 2010, DOI: 10.7763/IJCEE.2010.V2.148.

[16] Frankenfield, “How Cloud Computing Works,” *Investopedia*, Jul. 28, 2010. https://www.investopedia.com/terms/c/cloud-computing.asp (accessed Nov. 2018).

[17] Xi. Sun, G. Chang, and F. Li, “A Trust Management Model to Enhance Security of Cloud Computing Environments,” in *2011 Second International Conference on Networking and Distributed Computing*, Beijing, China, 2011, pp. 244–248. DOI: 10.1109/ICNDC.2011.56.

[18] P. Mell and T. Grance, “The NIST Definition of Cloud Computing,” p. 7.

[19] N. Jeyanthi, “An Enhanced Entropy Approach to Detect and Prevent DDoS in Cloud Environment,” vol. 5, no. 2, p. 11, 2013.

[20] I. Sattar, M. Shahid, and Y. Abbas, “A Review of Techniques to Detect and Prevent Distributed Denial of Service (DDoS) Attack in Cloud Computing Environment,” *IJCA*, vol. 115, no. 8, pp. 23–27, 2015, DOI: 10.5120/20173-2370.

[21] R. Aishwarya and S. Malliga, “Intrusion detection system- An efficient way to thwart against Dos/DDos attack in the cloud environment,” in *2014 International Conference on Recent Trends in Information Technology*, Chennai, India, 2014, pp. 1–6. DOI: 10.1109/ICRTIT.2014.6996163.

[22] W. Alosaimi and K. Al-Begain, “An Enhanced Economic Denial of Sustainability Mitigation System for the Cloud,” in *2013 Seventh International Conference on Next Generation Mobile Apps, Services and Technologies*, Prague, Czech Republic, 2013, pp. 19–25. DOI: 10.1109/NGMAST.2013.13.

[23] D. Kouril *et al.*, “Cloud-based testbed for simulation of cyberattacks,” in *2014 IEEE Network Operations and Management Symposium (NOMS)*, Krakow, Poland, 2014, pp. 1–6. DOI: 10.1109/NOMS.2014.6838298.

[24] R. Latif, H. Abbas, and S. Assar, “Distributed Denial of Service (DDoS) Attack in Cloud- Assisted Wireless Body Area Networks: A Systematic Literature Review,” *J Med Syst*, vol. 38, no. 11, p. 128, 2014, DOI: 10.1007/s10916-014-0128-8.

[25] L. Garber, “Denial-of-service attacks rip the internet,” *Computer*, vol. 33, no. 4, pp. 12–17, 2000, DOI: 10.1109/MC.2000.839316.

[26] J.-M. Kim, H.-Y. Jeong, I. Cho, S. M. Kang, and J. H. Park, “A secure smart-work service model based OpenStack for Cloud computing,” *Springer*, vol. 17, pp. 691–702, 2013, DOI: 10.1007/s10586-013-0251-1.

[27] A. A. Braiki, K. Hayawi, S. S. Mathew, and Z. Trabelsi, *Network Attacks and Defenses: A Hands-on Approach*. CRC Press, 2012.

[28] E. Anitha and S. Malliga, “A packet marking the approach to protect cloud environment against DDoS attacks,” in *2013 International Conference on Information Communication and Embedded Systems (ICICES)*, Chennai, 2013, pp. 367–370. DOI: 10.1109/ICICES.2013.6508330.

[29] R. K. Arun P. and S. Selvakumar, “Distributed Denial-of-Service (DDoS) Threat in Collaborative Environment - A Survey on DDoS Attack Tools and Traceback Mechanisms,” in *2009 IEEE International Advance Computing Conference*, Patiala, 2009, pp. 1275–1280. DOI: 10.1109/IADCC.2009.4809199.

[30] S. Dietrich, N. Long, and D. Dittrich, “Analyzing Distributed Denial of Service Tools: The Shaft Case,” *Usenix Assoc. Proc. Fourteenth Syst. Adm. Conf. (Lisa Xiv)*, pp. 329–339, 2000.

[31] K. A. Scarfone and P. M. Mell, “Guide to Intrusion Detection and Prevention Systems (IDPs),” National Institute of Standards and Technology, Gaithersburg, MD, NIST SP 800-94, 2007. DOI: 10.6028/NIST.SP.800-94.

[32] J. Nagamalai and N. C. S. N. Iyenger, “Packet resonance strategy: A spoof attack detection and prevention mechanism in cloud computing environment,” *IJCNIS*, vol. 4, no. 3, pp. 163–173, 2012.

[33] Q. Yan, F. R. Yu, Q. Gong, and J. Li, “Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges,” *IEEE*, vol. 18, no. 1, pp. 602–622, FIRST QUARTER 2016.

[34] T. Vissers, T. S. Somasundaram, L. Pieters, K. Govindarajan, and P. Hellinckx, “DDoS defence system for web services in a cloud environment,” *ELSEVIER*, vol. 37, pp. 37–45, 2014.

[35] W. Li, L. Ping, and X. Pan, “Use trust management module to achieve effective security mechanisms in cloud environment,” in *2010 International Conference on Electronics and Information Engineering*, Kyoto, Japan, 2010, pp. V1-14-V1-19. DOI: 10.1109/ICEIE.2010.5559829.

[36] B. Khadka, C. Withana, A. Alsadoon, and A. Elchouemi, “Distributed Denial of Service attack on Cloud: Detection and Prevention,” *IEEE*, p. 6, 2015.

[37] L. Yang, T. Zhang, J. Song, J. S. Wang, and P. Chen, “Defense of DDoS attack for cloud computing,” in *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, Zhangjiajie, China, 2012, pp. 626–629. DOI: 10.1109/CSAE.2012.6272848.

[38] W. Alosaimi, M. Zak, and K. Al-Begain, “Denial of Service Attacks Mitigation in the Cloud,” in *2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies*, Cambridge, United Kingdom, 2015, pp. 47–53. DOI: 10.1109/NGMAST.2015.48.

[39] T. Karnwal, T. Sivakumar, and G. Aghila, “A comber approach to protect cloud computing against XML DDoS and HTTP DDoS attack,” in *2012 IEEE Students’ Conference on Electrical, Electronics and Computer Science*, Bhopal, India, 2012, pp. 1–5. DOI: 10.1109/SCEECS.2012.6184829.

[40] N. Jeyanthi and P. C. Mogankumar, “A Virtual Firewall Mechanism Using Army Nodes to Protect Cloud Infrastructure from DDoS Attacks.,” *BULGARIAN ACADEMY OF SCIENCES*, vol. 14, no. 3, pp. 71–85, 2014.

[41] C. Borean, R. Giannantonio, M. Mamei, D. Mana, A. Sassi, and F. Zambonelli, “Internet and Distributed Computing Systems,” in *Urban Crowd Steering: An Overview*, vol. 9258, Italy: Springer, 2015, pp. 143–154.

[42] A. Bakshi and Y. B. Dujodwala, “Securing Cloud from DDOS Attacks Using Intrusion Detection System in Virtual Machine,” in *2010 Second International Conference on Communication Software and Networks*, Singapore, 2010, pp. 260–264. DOI: 10.1109/ICCSN.2010.56.

[43] B. Buchanan, F. Flandrin, R. Macfarlane, and D. J. Graves, “A Methodology to Evaluate Rate-Based Intrusion Prevention System against Distributed Denial-of-Service (DDoS),” *Edinburgh Napier University, UK*, p. 17, 2011.

[44] S. Yu, Y. Tian, S. Guo, and D. O. Wu, “Can We Beat DDoS Attacks in Clouds?,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 9, pp. 2245–2254, 2014, DOI: 10.1109/TPDS.2013.181.

[45] N. Jeyanthi, H. Shabeeb, M. A. S. Durai, and R. Thandeeswaran, “Reputation Based Service for Cloud User Environment,” *IJE*, vol. 27, no. 8 (B), 2014, doi: 10.5829/idosi.ije.2014.27.08b.03.

[46] R. Anandhi and V. N. Raj, “PREVENTION OF DDOS ATTACKS ON DISTRIBUTED CLOUD SERVERS BY PORT LOCK MECHANISM,” vol. 11, no. 5, p. 7, 2016.

[47] W. Dou, Q. Chen, and J. Chen, “A confidence-based filtering method for DDoS attack defence in a cloud environment,” *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1838–1850, 2013, DOI: 10.1016/j.future.2012.12.011.

[48] B. Joshi, A. S. Vijayan, and B. K. Joshi, “Securing cloud computing environment against DDoS attacks,” in *2012 International Conference on Computer Communication and Informatics*, Coimbatore, India, 2012, pp. 1–5. DOI: 10.1109/ICCCI.2012.6158817.

[49] S. S. Chapade, K. U. Pandey, and D. S. Bhade, “Securing Cloud Servers Against Flooding Based DDOS Attacks,” in *2013 International Conference on Communication Systems and Network Technologies*, Gwalior, 2013, pp. 524–528. DOI: 10.1109/CSNT.2013.114.

[50] M. Sauter, “‘LOIC Will Tear Us Apart’: The Impact of Tool Design and Media Portrayals in the Success of Activist DDOS Attacks,” *American Behavioral Scientist*, vol. 57, no. 7, pp. 983–1007, 2013, DOI: 10.1177/0002764213479370.

[51] N. Hoque, M. H. Bhuyan, R. C. Baishya, D. K. Bhattacharyya, and J. K. Kalita, “Network attacks: Taxonomy, tools and systems,” *Journal of Network and Computer Applications*, vol. 40, pp. 307–324, 2014, DOI: 10.1016/j.jnca.2013.08.001.

[52] 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*, p. 20, 2013.

[53] Xiaokai Xia, Qiuhong Pei, Yongpo Liu, Ji Wu, and Chao Liu, “Multi-level logs based web performance evaluation and analysis,” in *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, Taiyuan, China, 2010, pp. V4-37-V4-41. DOI: 10.1109/ICCASM.2010.5620512.

[54] M. N. Banu, “Cloud Computing Simulation Tools - A Study,” India, 2015, vol. 7, pp. 13–25.

[55] C. Wohlin, P. Runeson, M. Hst, M. C. Ohlsson, B. Regnell, and A. Wessln, *Experimentation in Software Engineering*. Springer Publishing Company, Incorporated, 2012.

[56] E. McCrum-Gardner, “Which is the correct statistical test to use?,” *British Journal of Oral and Maxillofacial Surgery*, vol. 46, no. 1, pp. 38–41, 2008, DOI: 10.1016/j.bjoms.2007.09.002.

[57] M. Berndtsson, Ed., *Thesis projects: a guide for students in computer science and information systems*, 2nd ed. London: Springer, 2008.

[58] L. Wenguang, “Study on Characteristic of Emerging Technology,” *IEEE*, 2011, DOI: 978-1-4244-6581-1/11.

[59] O. Hamrén, “Mobile phones and cloud computing,” Umeå University, Sweden, Sweden, 2012.

[60] S. Greene, “Quantitative Data Analysis: Choosing a statistical test,” University of Wisconsin-Stout, 2015.

[61] D. Eckhard, H. Hjalmarsson, C. R. Rojas, and M. Gevers, “Mean-squared error experiment design for linear regression models*,” *IFAC Proceedings Volumes*, vol. 45, no. 16, pp. 1629–1634, 2012, DOI: 10.3182/20120711-3-BE-2027.00339.

[62] J. Corsini, “Analysis and Evaluation of Network Intrusion Detection Methods To Uncover Data Theft,” MSc. in Advanced Networking, Edinburgh Napier University, UK, 2009.

[63] J. Sammers, V. Yegneswaran, and Barford, “Toward Comprehensive Traffic Generation for Online IDS Evaluation,” 2005, p. 12.

[64] H. Jiang and C. Dovrolis, “Passive estimation of TCP round-trip times,” *SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 3, pp. 75–88, 2002, DOI: 10.1145/571697.571725.

Appendixes

Appendix A - *Comparison of DoS and DDoS attack generation tools*

Table 21: Comparison between attack generation applications of DoS and DDoS

Attacking Application	**Features**						
	Attack Form	**Availability of GUI**	**Language based on**	**Compatible Operating System**	**Basic Requirements Need to satisfy**	**Name of attack(s) generated**	**Constraints**
Commview visual packet builder	DoS	yes	No	Works based on the juniper	Victim computer, attacker computer	Land attack, Teardrop attack	most expensive because testing requires an actual experimental setup with ethernet cables.
LOIC	DOS	Yes	C#	GNU/Linux, Windows, Mac OS, and Android	Download the software	HTTP attack, Application / Transport Layer Attack, Buffer overflow attack,	Not much effective when compared to XOIC
XOIC	DOS	Yes	C++	Only for windows 7 andWindows 8	Download the software	HTTP attack, Application / Transport Layer Attack, Buffer overflow attack, ping to death, Smurf attack	There are some bugs in the tool

HOIC	DOS, DDOS (when coordinated with multiple individuals)	yes	Real Basic	Windows, Linux, or Mac OS	Download the software	HTTP attack, Application / Transport Layer Attack, Buffer overflow attack, ping to death, Smurf attack	Not GUI friendly when compared to LOIC and XOIC
HULK	DoS	no	Don't know	Windows, Linux (any of them with python installed)	Download the software and learn the commands to run the tool	Application Layer attack, HTTP attack.	Cannot make quantitative analysis of data
DDOSIM	DDOS	no	C++	Linux	Download the software and learn the commands to run the tool	Application, Transport Layer attack	No GUI
Trinoo	DDoS	no	C++	Linux	Download the software and learn the commands to run the tool	All except land and Teardrop	No GUI

Appendix B – Results of the functionality testing

Table 22: The load on the CPU for TCP flood

The load on the CPU for TCP flood			
#	PPS	In Source Approach (%)	In Destination Approach (%)
1	0	10	10
2	2000	19	20.1
3	4000	30	32
4	6000	57.4	58
5	8000	54	55
6	10000	53.2	54.9
7	12000	53	54.6
8	14000	53.5	53.8
9	16000	53	52.8
10	18000	52.8	52.5
11	20000	52.1	52
12	22000	51.6	51.8
13	24000	49.2	50
14	26000	49.6	49.7
15	28000	49.1	49.2
16	30000	48.4	48.6

Table 23: The load on the CPU for UDP flood

Table 20: The load on the CPU for UDP flood			
#	PPS	In Source Approach (%)	In Destination Approach (%)
1	0	10	10
2	2000	12.1	14.2
3	4000	16.5	16.8
4	6000	55.2	55.4
5	8000	58.1	58.9
6	10000	57.6	58.2
7	12000	55.1	56.7
8	14000	52.3	52
9	16000	52.9	52.5
10	18000	53.5	53.2
11	20000	58.9	58.5
12	22000	59.2	59.8
13	24000	62.9	63.1
14	26000	63.2	64.2
15	28000	66.1	66.5
16	30000	68.1	68.9

Table 24: The load on the CPU for HTTP flood

The load on the CPU for HTTP flood			
#	PPS	In Source Approach (%)	In Destination Approach (%)
1	0	10	10
2	2000	29.8	30.1
3	4000	38.1	39.1
4	6000	66.9	68.5
5	8000	65.5	66.2
6	10000	64.9	65.8
7	12000	64.3	65.3
8	14000	63.9	64.2
9	16000	63.1	63.9
10	18000	62.8	63.5
11	20000	62.1	63.1
12	22000	61.2	62.8
13	24000	59.7	62.6
14	26000	59.1	61.1
15	28000	58.2	60.8
16	30000	57.9	60.2

Table 25: The load on the CPU for Mix Protocol flood

The load on the CPU for Mix Protocol flood			
#	PPS	In Source Approach (%)	In Destination Approach (%)
1	0	10	10
2	2000	21	25.1
3	4000	38.1	50.8
4	6000	63.3	66.5
5	8000	62.2	64.9
6	10000	62	63.2
7	12000	61.8	63.9
8	14000	60.7	62.9
9	16000	60.1	62.1
10	18000	59.8	61.7
11	20000	59	61.1
12	22000	58.2	59.9
13	24000	60.3	60.5
14	26000	63.2	64.3
15	28000	66.9	67.2
16	30000	69.9	70.2

Table 26: The utilization of memory for TCP floods

The utilization of memory for TCP floods			
#	PPS	In Source Approach (Mbps)	In Destination Approach (Mbps)
1	0	316	315
2	2000	321	320
3	4000	336	339
4	6000	332	335
5	8000	339	339
6	10000	342	340
7	12000	335	338
8	14000	335	336
9	16000	339	338
10	18000	337	338
11	20000	340	338
12	22000	335	337
13	24000	336	336
14	26000	337	335
15	28000	337	334
16	30000	335	334

Table 27: The utilization of memory for UDP floods

The utilization of memory for UDP floods			
#	PPS	In Source Approach (Mbps)	In Destination Approach (Mbps)
1	0	317	316
2	2000	322	321
3	4000	341	340
4	6000	292	290
5	8000	290	288
6	10000	289	289
7	12000	290	288
8	14000	285	288
9	16000	288	287
10	18000	286	287
11	20000	300	287
12	22000	255	288
13	24000	287	289
14	26000	289	289
15	28000	289	289
16	30000	291	290

Table 28: The utilization of memory for HTTP floods

The utilization of memory for HTTP floods			
#	PPS	In Source Approach (Mbps)	In Destination Approach (Mbps)
1	0	313	312
2	2000	319	318
3	4000	325	322
4	6000	330	328
5	8000	325	328
6	10000	333	331
7	12000	336	335
8	14000	335	338
9	16000	341	340
10	18000	338	339
11	20000	345	342
12	22000	335	340
13	24000	338	339
14	26000	338	335
15	28000	339	339
16	30000	337	336

Table 29: The utilization of memory for Mix Protocol floods

The utilization of memory for Mix Protocol floods			
#	PPS	In Source Approach (Mbps)	In Destination Approach (Mbps)
1	0	287	286
2	2000	289	288
3	4000	296	295
4	6000	319	300
5	8000	300	305
6	10000	320	318
7	12000	320	316
8	14000	316	315
9	16000	315	314
10	18000	314	315
11	20000	315	315
12	22000	315	316
13	24000	317	318
14	26000	318	318
15	28000	319	319
16	30000	318	320

Table 30: Availability of Bandwidth for TCP flood

Availability of Bandwidth for TCP flood			
#	PPS	In Source Approach (Mbps)	In Destination Approach (Mbps)
1	0	60.3	60.3
2	2000	36.2	37.2
3	4000	19.7	20.95
4	6000	11.9	12.7
5	8000	8.89	9.89
6	10000	7.08	7.08
7	12000	4.168	4.168
8	14000	2.15	1.256
9	16000	1.16	1.076
10	18000	0.896	0.896
11	20000	0.705	0.507
12	22000	0.507	0.119
13	24000	0.129	0.102
14	26000	0.105	0.086
15	28000	0.08	0.07
16	30000	0.054	0.054

Table 31: Availability of Bandwidth for UDP flood

Availability of Bandwidth for UDP flood			
#	PPS	In Source Approach (Mbps)	In Destination Approach (Mbps)
1	0	60.3	60.3
2	2000	41.25	41.25
3	4000	21.35	21.35
4	6000	11.63	11.63
5	8000	7.32	7.32
6	10000	4.861	4.861
7	12000	4.19	4.19
8	14000	1.23	1.23
9	16000	1.06	1.06
10	18000	0.589	0.589
11	20000	0.109	0.109
12	22000	0	0.04
13	24000	0	0
14	26000	0	0
15	28000	0	0
16	30000	0	0

Table 32: Availability of Bandwidth for HTTP flood

Availability of Bandwidth for HTTP flood			
#	PPS	In Source Approach (Mbps)	In Destination Approach (Mbps)
1	0	60.3	60.3
2	2000	38.861	40.2
3	4000	25.63	24.4
4	6000	11.5	11.5
5	8000	9.89	8.98
6	10000	6.46	9.89
7	12000	5.39	5.28
8	14000	2.62	2.125
9	16000	1.35	1.18
10	18000	0.795	1.076
11	20000	0.599	0.557
12	22000	0.365	0.1
13	24000	0.1	0.076
14	26000	0.076	0.066
15	28000	0.052	0.0369
16	30000	0.013	0.0296

Table 33: Availability of Bandwidth for Mix Protocol flood

Availability of Bandwidth for Mix protocol flood			
#	PPS	In Source Approach (Mbps)	In Destination Approach (Mbps)
1	0	60.3	60.3
2	2000	34.9	35.5
3	4000	24.21	22.53
4	6000	14.9	13.5
5	8000	10.6	6.23
6	10000	5.62	8.61
7	12000	4.92	4.35
8	14000	2.98	2.31
9	16000	1.22	1.62
10	18000	0.44	0.49
11	20000	0.42	0.47
12	22000	0.39	0.106
13	24000	0.106	0.079
14	26000	0.059	0.053
15	28000	0.079	0.01
16	30000	0.01	0

Table 34: Presence of Delay (Latency) for TCP flood

Presence of Delay (Latency) for TCP flood			
#	PPS	In Source Approach (in milliseconds – ms)	In Destination Approach (in milliseconds – ms)
1	0	1.876	1.876
2	2000	1.994	2.125
3	4000	2.215	2.993
4	6000	2.425	3.632
5	8000	37.765	40.765
6	10000	32.453	35.765
7	12000	32.899	33.899
8	14000	30.993	31.993
9	16000	29.899	30.889
10	18000	30.523	31.523
11	20000	31.626	32.626
12	22000	31.898	32.898
13	24000	32.628	33.628
14	26000	32.958	33.958
15	28000	33.485	34.485
16	30000	33.412	34.995

Table 35: Presence of Delay (Latency) for UDP flood

Presence of Delay (Latency) for UDP flood			
#	PPS	In Source Approach (in milliseconds – ms)	In Destination Approach (in milliseconds – ms)
1	0	1.876	1.876
2	2000	2.113	2.659
3	4000	2.659	3.123
4	6000	3.123	3.985
5	8000	38.657	39.657
6	10000	38.898	40.895
7	12000	38.834	39.834
8	14000	35.357	36.357
9	16000	34.924	35.924
10	18000	35.254	36.245
11	20000	36.215	36.954
12	22000	36.751	37.751
13	24000	36.148	37.148
14	26000	35.851	36.851
15	28000	35.519	36.519
16	30000	36.112	36.425

Table 36: Presence of Delay (Latency) for HTTP flood

Presence of Delay (Latency) for HTTP flood			
#	PPS	In Source Approach (in milliseconds – ms)	In Destination Approach (in milliseconds – ms)
1	0	1.876	1.876
2	2000	1.958	1.958
3	4000	2.324	2.324
4	6000	3.245	2.955
5	8000	36.324	37.324
6	10000	37.965	38.965
7	12000	38.335	39.335
8	14000	38.992	39.992
9	16000	37.125	38.125
10	18000	36.895	37.895
11	20000	36.325	37.325
12	22000	36.795	37.759
13	24000	36.982	37.982
14	26000	36.743	37.743
15	28000	36.143	37.143
16	30000	35.124	36.941

Table 37: Presence of Delay (Latency) for Mix Protocol flood

Presence of Delay (Latency) for Mix Protocol flood			
#	PPS	In Source Approach (in milliseconds – ms)	In Destination Approach (in milliseconds – ms)
1	0	1.876	1.876
2	2000	2.548	3.548
3	4000	2.245	4.245
4	6000	1.127	2.127
5	8000	31.548	32.548
6	10000	33.731	34.731
7	12000	34.234	35.234
8	14000	34.992	36.129
9	16000	34.212	35.421
10	18000	34.324	35.324
11	20000	33.752	34.752
12	22000	34.987	35.987
13	24000	35.578	36.578
14	26000	36.245	37.245
15	28000	37.147	38.147
16	30000	39.125	39.357

Table 38: Percentage of loss of packets in TCP flood

Percentage of loss of packets in TCP flood			
#	PPS	In Source Approach (%)	In Destination Approach (%)
1	0	0	0
2	2000	0	0
3	4000	0	0
4	6000	0	6
5	8000	12	18
6	10000	19	27
7	12000	26	33
8	14000	34	40
9	16000	39	45
10	18000	46	54
11	20000	50	64
12	22000	57	67
13	24000	66	71
14	26000	70	74
15	28000	75	76
16	30000	76	80

Table 39: Percentage of loss of packets in UDP flood

Percentage of loss of packets in UDP flood			
#	PPS	In Source Approach (%)	In Destination Approach (%)
#	Pps	Src %	dest %
1	0	0	0
2	2000	0	0
3	4000	0	0
4	6000	0	4
5	8000	13	20
6	10000	27	34
7	12000	32	37
8	14000	38	42
9	16000	45	46
10	18000	49	52
11	20000	55	59
12	22000	59	63
13	24000	62	69
14	26000	70	73
15	28000	72	75
16	30000	73	77

Table 40: Percentage of loss of packets in HTTP flood

Percentage of loss of packets in HTTP flood			
#	PPS	In Source Approach (%)	In Destination Approach (%)
1	0	0	0
2	2000	0	0
3	4000	0	0
4	6000	1	3
5	8000	13	19
6	10000	21	30
7	12000	30	36
8	14000	36	43
9	16000	47	52
10	18000	51	56
11	20000	56	59
12	22000	58	61
13	24000	61	65
14	26000	64	69
15	28000	69	73
16	30000	70	75

Table 41: Percentage of loss of packets in Mix Protocol flood

Percentage of loss of packets in Mix Protocol flood			
#	PPS	In Source Approach (%)	In Destination Approach (%)
1	0	0	0
2	2000	0	0
3	4000	0	0
4	6000	0	1
5	8000	14	20
6	10000	27	32
7	12000	31	36
8	14000	36	45
9	16000	46	48
10	18000	50	55
11	20000	54	58
12	22000	55	60
13	24000	60	63
14	26000	63	67
15	28000	66	71
16	30000	67	72