Master of Science in Computer Science
February 2017

Assessment of Snort Intrusion Prevention Systems in Virtual Environment Against DoS and DDoS attacks

An empirical evaluation between source mode and destination mode

Avinash Kiran Ivvala

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona Sweden

This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Computer Science. The thesis is equivalent to 20 weeks of full time studies.

Contact Information:
Author:
Avinash Kiran Ivvala
E-mail: aviv15@student.bth.se

University advisor:
Emiliano Casalicchio
Department of Computer Science and Engineering (DIDD)

Faculty of Computing	Internet	: www.bth.se
Blekinge Institute of Technology	Phone	: +46 455 38 50 00
SE-371 79 Karlskrona, Sweden	Fax	: +46 455 38 50 57

ABSTRACT

Context. Cloud computing (CC) is developed as a Human-centered computing model to facilitate its users to access resources anywhere on the globe. The resources can be shared among any cloud user which mainly questions the security in cloud computing. There are Denial of Service and Distributed Denial of Service attacks which are generated by the attackers to challenge the security of CC. The Next-Generation Intrusion Prevention Systems (sometimes referred as Non-Traditional Intrusion Prevention Systems (NGIPS) are being used as a measure to protect users against these attacks. This research is concerned with the NGIPS techniques that are implemented in the cloud computing environment and their evaluation.

Objectives. In this study, the main objective is to investigate the existing techniques of the NGIPS that can be deployed in the cloud environment and to provide an empirical comparison of source mode and destination mode in Snort IPS technique based on the metrics used for evaluation of the IPS systems.

Methods. In this study, a systematic literature review is used to identify the existing NGIPS techniques. The library databases used to search the literature are Inspec, IEEE Xplore, ACM Digital Library, Wiley, Scopus and Google scholar. The articles are selected based on an inclusion and exclusion criteria. The experiment is selected as a research method for the empirical comparison of Source mode and destination mode of Snort NGIPS found through literature review. The testbed is designed and implemented with the Snort filter techniques deployed in the virtual machine.

Results. Some common metrics used for evaluating the NGIPS techniques are CPU load, Memory usage, bandwidth availability, throughput, true positive rate, false positive rate, true negative rate, false negative rate and accuracy. From the experiment, it was found that Destination mode performs better than source mode in Snort. When compared with the CPU load, Bandwidth, Latency, Memory Utilization and rate of packet loss metrics.

Conclusions. It was concluded that many NGIPS of the cloud computing model are related to each other and use similar techniques to prevent the DoS and DDoS attacks. The author also concludes that using of source based and destination based intrusion detection modes in Snort has some difference the performance measures

Keywords: Intrusion Prevention Systems, cloud computing, snort, source mode, destination mode.

ACKNOWLEDGEMENTS

Studying masters in Sweden improved me a lot and taught me new skills. Studying at Blekinge Institute of Technology is really an awesome experience and learned new things. Memories I made while working on my thesis were unforgettable.

Firstly, I would like to thank my supervisor Emiliano Casalicchio, for his great support throughout the course and his suggestions were very useful to complete my course. His wicked humor in problem solving helped me to regain my strength when I got into trouble during my thesis.

I would like to thank my parents for their love, support, and hopes on me. I would also like to thanks my friends and their support during my course.

Finally, it was a great opportunity to work on my thesis, meet new people and learn new things.

Thanks for everyone who made what I am today. Best is yet to come.

Avinash Kiran Ivvala

CONTENTS

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
RMN	: Reflection Mirror Node
TMN	: Transparent Mirror Node
SDN	: Software Defined Networking
RP	: Reverse Proxy
CDAP	: Cloud DDoS Attacks Protection
DST	: Dempster Shafer Theory
BPA	: Basic Probabilities Assignments
SOA	: Service Oriented Architecture
CBF	: Confidence-Based Filtering

1 INTRODUCTION

Cloud computing (CC) is using a network of remote servers hosted on the internet (“The cloud”) to store, manage and process the data, into the third party data centers which are located at a distance ranging from across a town to across the world instead of using a personal computer or local servers. CC is integrated and developed with Grid computing, and virtualization technology. This facilitates the end users to enable ubiquitous configurable shared computer processing resources and data applications to computer and other devices on demand. This can be provisioned with the minimal management effort.

There are some services and models which make the CC more feasible and accessible to its customers [1]. The two working models of the cloud are Deployment Model and Service Model. Cloud mainly offers three types of service models to its customers. They are Platform as a service (PaaS), Software as a Service (SaaS) and Infrastructure as a Service (IaaS) [2]. There are four types of deployment models in the cloud. They are public cloud, private cloud, hybrid cloud, and community cloud [2]. These working models make CC more flexible to small and individual businesses to use great configurable computing services with low cost.

CC mainly depends on sharing of resources with the third party which is the major concern about the security threats. One of the major security threat to CC is Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks [3]. In DoS attack the attacker sends a huge amount of fake data packets over the single network from a single zombie computer to effect the bandwidth or resources of the victim systems [4]. Distributed Denial of Service (DDoS) attack is similar to DoS attack but is performed using more than one zombie systems known as botnets with multiple networks which are placed at the same location to anywhere in the globe, targeting single system [4].

For protecting the cloud against these attacks cloud service providers (CSPs) use Intrusion Detection Prevention Systems (IDPS) along with traditional firewall mechanism. IDPS is the combination of Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS). IDS generally identifies the security threats by detecting the probes and attacks but do not prevent them from reaching the Data Center (DC). According to Richard Kemmerer et.al., Intrusion Detection Systems are a bit of misnomer because they don't detect intrusions but show the evidence of the intrusion [5]

The IPS not only detect the intrusions using the detection algorithm but also prevent the intrusion from reaching the DC which will reduce the impact of the attacks in the cloud environment. According to Piper, These IPS can be segregated into two generations based on their intrusion prevention capabilities [6]. They are Traditional Intrusion Prevention Systems (TIPS) and Non-Traditional Intrusion Prevention Systems. These Non-Traditional IPS are also referred as Next-Generation Intrusion Prevention Systems (NGIPS). The IPS which have been around for years are known as Traditional IPS. The Non-Traditional or Next-Generation IPS are the prevention mechanisms that are evolved from traditional IPS with more solutions to the weakness of Traditional IPS.

In this research existing NGIPS techniques are found and also the metrics used for evaluating the NGIPS techniques empirically are also found. An empirical comparison is done based on the performance metrics for the two modes in Snort IPS (for preventing the DoS and DDOS attacks) are done to evaluate the best Non-Traditional IPS mode among them. They are selected based on the availability of the resources and usage popularity.

1.1 Problem Definition

The main challenges of cloud computing are security, data privacy, data availability on request, performance and latency [1][7]. “Out of these, security is considered as the major concern for the cloud” [7]. There are many security challenges like data security, Network security, resource availability etc.., with respective to the cloud service layers. To achieve them, many security attacks are generated by the attackers like DoS attacks, Cross-site Scripting attacks, DDoS attacks, Buffer overflow attacks etc..,[8] Out of them DoS and DDoS attacks are placed as the top nine threats to CC [9] [10]. The Cloud servers are designed to increase their computational power according to the number of queries. But, there is a limit to process the queries per unit time. When the certain amount of queries to the server reached, it stops entertaining further queries. This is the loophole which draws the attention of attackers towards generating DoS and DDoS attacks.

In the Norwegian history DDoS attack is the biggest attack which interrupted online payment systems of five banks, three airlines, two telecommunication companies, and one insurance company [11].

An DDoS attack on Amazon Elastic Cloud Computing (EC2) is also one of the major attacks [12]. Due to technical error in this control service, there is a possibility to manipulate eavesdropped message despite of digital signed operation. This made hackers to execute arbitrary code and perform DDoS attack leading EC2 costs on users bill.

These DoS and DDoS attacks can easily be implemented by the hacker even with a novice knowledge in hacking. This is possible due to the availability of DoS and DDoS attack generation tools [13]. The attackers are performing DoS and DDoS attacks to deplete the bandwidth or resources of the victim's system [9]. DDoS attacks mainly affect the availability of network resources by exhausting its bandwidth which results in legitimate users not to access the cloud resources. So, there is a need to prevent these attacks and provide secure services to the users.

There are many TIPS and NGIPS techniques which can be deployed in the virtual networks both at the server side and client side with respective to the cloud service layers against DoS and DDOS attacks. The empirical comparisons based on the metrics provide better understandings of performance standards of the applications [11]. There is an empirical comparison of the Traditional IPS which helps the CSPs to select the suitable IPS to their Cloud. The above statement can be motivated by analyzing the literatures [7] [11]. According to A. Patel et.al. and A. Carlin et.al [14] [11], TIPS are inefficient to deploy in the cloud computing environment as they are not designed to detect and prevent attacks which are generated by the attackers with latest attack generation tools which got better-enhanced features to destroy the network. So in these days using of the Non-Traditional IPS became indispensable to protect DC against these attacks.

When the author of this thesis performed literature search for empirical evaluation on NGIPS techniques using search string

(("Cloud computing" OR "Cloud environment" OR virtualization) AND (Intrusion OR Trespass OR Attack OR Penetration) AND (Prevention OR Protection OR Prediction OR "Early warning" OR Response OR Resilience) AND ("Next generation" OR "Non-traditional" OR update*) AND ($empirical OR experiment*) AND (compare*))

no literatures were retrieved from the data bases. By this we can say that there was limited research available in empirical comparison of the Non-Traditional IPS for preventing the DOS and DDOS attacks. So there is a need of empirical comparison for Non-Traditional IPS which is the research problem addressed in this study.

There are many NGIPS techniques to prevent DoS and DDoS attacks. But no single technique can prevent all protocols of DoS and DDoS attacks [15]. In order to gain knowledge about the NGIPS technique and the evaluation metrics, and to evaluate them there is a need of identifying different NGIPS techniques for preventing DoS and DDoS attacks in cloud computing environment.

1.2 Research Purpose

In the recent years, the strengths and weakness of the non-traditional IPS for preventing the DOS and DDOS attacks are not explained empirically. The purpose of this research is to identify the existing Non-Traditional Intrusion Prevention Mechanisms (IPS) for preventing the DoS attacks and DDoS attacks in cloud computing and to identify the best mode in Snort NGIPS with empirical comparison between them based on the evaluation metrics.

This research will help the CSP's to know the existing NGIPS techniques that can be deployed in the cloud environment to protect their DC against DoS and DDoS attacks. This research will also help CSP's to choose the appropriate mode in Snort according to their cloud environment.

1.3 Aims and Objectives

Based on the research problem the aims and objectives that are designed to fulfill the research motive are given in this section

The main aim of this research is to find existing NGIPS techniques and compare any two techniques empirically with respective to DoS and DDoS attacks based on availability of resource. To achieve this aim, objectives defined are as follows:

- **Objective 1:**
  Identifying the different Non-Traditional IPS which are used in Cloud Environment.
  Motivation: this helps in identifying the existing NGIPS techniques and will help the CSP to know the different NGIPS techniques.
- **Objective 2:**
  Identifying the performance and security metrics to measure the techniques.
  Motivation: this helps in answering the research question 2 by giving the list of mostly used metrics for evaluating the NGIPS techniques.
- **Objective 3:**
  To evaluate the source mode and Destination mode empirically and analyze the performance of the techniques.
  Motivation: on analyzing the performance of both modes empirically will help the CSP's to choose the appropriate mode in Snort NGIPS.

1.4 Research Questions

Based on the aims and objectives, the research questions framed to answer the motive of this research are as follows.

- **RQ 1:** What are the different Next-Generation Intrusion Prevention Systems (NGIPS) that can be used in cloud computing for preventing the DoS and DDOS attacks?
  Motivation: This research question will help in identifying the existing prevention techniques which can be used in securing the cloud environment.

- **RQ 2:** What are the different metrics used for evaluating Next-Generation Intrusion Prevention Systems (NGIPS) for Cloud Computing against DoS and DDOS attacks?
  Motivation: This research question will help in identifying the metrics to evaluate any NGIPS techniques for cloud computing.

- **RQ 3:** What is the performance of "track by source" mode and "track by destination" mode in Snort Non-Traditional Intrusion Prevention System (IPS) against DoS and DDoS attacks?

Motivation: This research question will help the third party user in selecting the suitable prevention mode while using Snort IPS for their cloud environment.

1.5 Audience

This thesis document is designed for CC security staff and program managers, Computer Security Incident Response Teams (CSIRT), CSP, system and network administrators who regulates and monitors IPS for CC. This document assumes that the reader has some knowledge in IDPS technologies and security challenges of CC.

1.6 Thesis Structure

This research report mainly consists of Introduction, Background, Literature Review, Snort (IPS), Experiment, Analysis, Discussion, Conclusion and Future work. The author provides introduction about the research area and motivation for the problem statement is provided. The author provides background knowledge about Cloud Computing, Types of DoS and DDoS attacks and IPS classification in Cloud Environment in Chapter 2. Detailed information about how the research method Literature Review is planned and conducted and corresponding results are provided in chapter 3. Chapter 4 discuss the basic concepts in understanding the Snort IPS and the how the rules are modified to suit experiment test bed. Chapter 5 presents how the experimental procedure is carried out throughout the research and the results of the experiment. Chapter 6 epitomizes the statistical analysis for the experimental data. Chapter 7 discuss the validity threats and answers to research questions briefly. Chapter 8 provides the conclusions of the research and future work for how can this research be further enhanced.

2 BACKGROUND

This chapter provides the basic concepts of cloud computing, types of DoS and DDoS attack classifications and attack generating tools, and brief discussion about IPS which help in better comprehend of the context of this research. the related work on this research is also presented. The structure of this chapter is as follows

- Section 2.1: Discusses about Cloud Computing architecture and its Security issues
- Section 2.2: This section epitomizes DoS and DDoS attack Protocols and available tools in generating these attacks.
- Section 2.3: This section provides the brief idea about the IPS systems and their classification.

2.1 Cloud Computing and its Security

This section is provided with CC definition, brief information about the architecture and service models of CC. It also includes the security problems of CC.

2.1.1 Cloud Computing

According to National Institute of Standards and Technology (NIST) Cloud computing can be defined as "*a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.*" [16]. This definition can be explained as the users can get the computing resources easily and quickly anywhere in the globe with less management cost. The basic cloud environment model can be seen in figure 1.

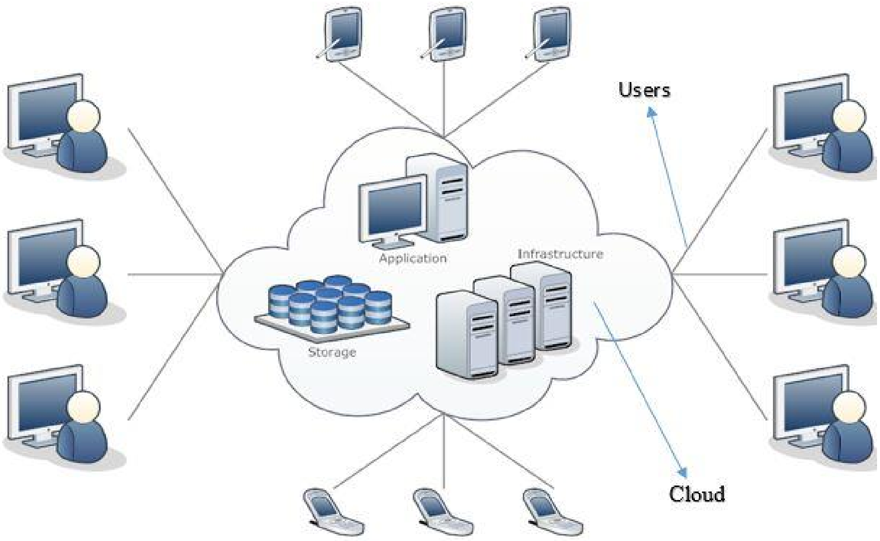


Figure 1 Basic Cloud Environment Model [link]

The main purpose of cloud computing is using configurable resources over the network without actually installing in local PC with minimal management cost.

According to the author W.Li et.al [17] CC is a computing service, providing dynamically virtualization resource which can be availed through internet and is extendible according to

user requirement. The statement accentuates that CC uses virtualization technology in providing its services to the users based on their requirement.
According to F.Li et.al [18] CC is an computing model based on parallel computing, distributed computing, and grid computing. CC can sometimes be designed with the realization of above mentioned computing models which percepts the concepts like virtualization, utility computing, and the three service layers of CC.

2.1.2 Working models of Cloud Computing

According to [1][19][20], there are two working models which define the cloud architecture. They are:

- Service Models
- Deployment Models

- **Service Models:** According to [1][19], there are three service models in cloud computing. They are:
  - **Software as a Service (SaaS):** Users can access the software services without installing in the local computers but can lease the software from the CSPs and use according to their requirement.
  - **Platform as a Service (PaaS):** Users are delivered with a collaborative platform for software development which is independent of the data source being used for the software applications. Some services that are provided in this category are application run-time environment, sharing service, automation management service etc.
  - **Infrastructure as a Service (IaaS):** Users are delivered with the cloud computing infrastructures such as servers, storage networks, storage devices, operating systems based on “demand on service” without purchasing the additional servers or software data center space or any storage equipment.

- **Deployment Models:** According to [1], [19], there are four deployment models in the cloud environment.
  - **Private Cloud:** The cloud infrastructure which is indulged for a single organization which comprises multiple business units. The cloud maybe managed by third parties or by the same organization or by the combination of them.
  - **Community cloud:** The cloud infrastructure which is indulged for an explicit community of users from organizations that have shared concerns. The cloud may be managed by third parties or by the organizations in the community or sometimes by the combination of them.
  - **Public cloud:** The cloud infrastructure which is indulged for the general public around the globe. These type of cloud may be managed by the government organizations or business organizations, academic or combination of some of the organizations.
  - **Hybrid cloud:** The composition of two or more distinct cloud deployment infrastructure (Private, Community, or Public cloud) is the hybrid cloud. The distinct cloud deployments are bounded together with standardized or proprietary technology which supports data and application portability.

When compared to the other computing models cloud computing models have some unique characteristics [20].
On-demand self-service: Users can obtain the provision of cloud resources without any long delays whenever they are signed into the cloud.
Board network access: Accessing the cloud resources with a wide range of devices anywhere around the globe.

Resource pooling: Cloud service provider has an ability to pool the cloud resources to cloud consumers who are isolated with each other. The assigning and reassigning of the resources is done dynamically according to the demand of the user. Pooling of resources is generally done by multitenancy models which usually on virtualization technology.
Rapid elasticity: CC is designed to adapt the workload changes by facilitating users to allocate or de-allocate the additional space in the cloud, so as to meet the on-demand self-service meticulously.
Measured Service: Services of the cloud are monitored and controlled by the cloud service provider for access control, billing, optimizing resources and also to plan and manage tasks.

2.1.3 Cloud Computing Security

The main feature of CC is its online rental service of huge resources which is achieved through virtualization and networking technologies. So, there are security problems like secrecy, authenticity, confidentiality, data leakage [21] [22]. Many external attacks which target the cloud services are being implemented by the hackers to violate its features. According to Wael, Alosaimi, and Khalid Al-Begain [23], CC attacks can be classified into four types. They are:

- **Policy and Organizational Risks:** The risks that are dropped in this category are compliance risks, end of service risks, loss of control, and portability issues.
- **Legal Issues:** The risks that are dropped in this category are Contracts, Data Locations, Data deletions, and service level agreements.
- **Physical Security Issues:** Damaging the data center either by natural hazards such as floods, earthquakes. This also includes the intruders attempting to penetrate and also the staff switch off air conditions.
- **Technical Errors:** The risks that are dropped in this category are Data confidentiality, Data integrity, Data segregation, Encryption issues, Web application security issues, Network attacks such as DoS and DDoS, IP spoofing, virtualization vulnerabilities, Port scanning, Man in The Middle Attack (MITM).

To provide the best quality service of the cloud to users the above-mentioned security risks should be mitigated. To solve the Technical Errors, deploying IPS in a cloud environment is one of the solutions [24].

2.2 DoS and DDoS Attacks

In Denial of Service (DoS) attack the attacker sends a huge amount of fake data packets over the single network from a single zombie computer to effect the bandwidth or resources of the victim systems [4].

Distributed Denial of Service (DDoS) attack is similar to DoS attack but is performed using more than one zombie systems known as botnets with multiple networks which are placed at the same location to anywhere in the globe, targeting single system [4].

According to [9] [15], DDoS attacks are launched using botnets. The Botnets are selected by the attackers by snooping the network for the machines that are prone to vulnerabilities and use them as agents. These machines are known as zombie machines or botnets. Spoofed IP addresses are used by the host and zombie machines which make difficult to trace the attacker and its source. The main aim of generating this attack is to overload the resources. In the context of CC resources are Bandwidth, CPU cycles, Memory, File descriptors, Buffers etc., These attacks can sometimes crash the server.

2.2.1 Types of DoS and DDoS Attacks

There are different types of DoS and DDoS attacks. The characteristics of those attacks are explained below in the view of CC [1] [7] [13], [25]–[28].

- **UDP flood attack:** In this attack, the attacker sends User Datagram Protocol (UDP) messages with spoofed return addresses. This is flooded to the random port of victim host with numerous UDP packets which force the host to listen on that port continuously for an application. If no application is found, then it replies with an ICMP destination unreachable packet. The continuous listening to the port by the host saps the resources of the host which leads to inaccessibility of legitimate users.
- **ICMP flood attack:** This is similar to the UDP flood attacks. It sends a lot of Echo request packets (ping) rapidly to the target resource, without awaiting replies from the target. Also, the principle of ICMP floods is similar to UDP flood attack and can consume both incoming and outgoing bandwidth, as the victim's servers can slow down the overall system by responding with ICMP echo reply packets.
- **Protocol Attacks:** Protocol attacks consumes actual server resources or the intermediate communication components like firewalls, load balancers. This attack is measured in packets per second and it generally includes SYN floods, fragmented packet attacks, ping of death, Smurf DDoS etc.
- **Smurf Attacks:** In this type of attack the attacker machine sends a large number of (ICMP) Internet Control Message Protocol ICMP_ECHO_REQUESTS to the network devices that supports broadcasting technique with the spoofed IP address of the victim. Then the ICMP_ECHO_RESPONSE packets are sent to the victim by the machines in the particular broadcast network. This floods the victim with the fake ICMP_ECHO_REPLY messages.
- **IP Spoofing attack:** the attacker alters the headers of source IP with the legitimate IP address or by a false IP address. This makes cloud server held in loop state for a non-completed request. This makes server busy cannot process any further requests which affect the genuine user.
- **Teardrop attack:** this is a type of DoS attack where the IP packets are fragmented into smaller chunks. Each fragmented IP packets contains original IP packet's header. TCP/IP stack will overlap those IP fragments with each other when the destination server tries to reassemble them. This leads in crashing the victim system.
- **SYN flood attack:** In this type of attack the attacker floods the victim machine with TCP/SYN packets with spoofed IP address. In TCP/IP three-way handshaking the server sends the SYN_ACK back to the spoofed IP address as an acknowledgment of synchronize message sent by the attacker and waits for the response. But the response message will never come back as the IP address is fake. This half-open connection exhausts the connections to the server avoiding server response to legal requests.
- **PING of Death attack:** the attacker sends the mangled ping packet to a computer with a size greater than the limit of IP protocol (65,535). The victim's machine cannot able to handle the oversize packets causes the machine to crash or reboot. According to [7], the variants of this type of attack includes jolt, ICMP bug, IceNewk, SPING, ping o Death.
- **LAND attack:** this attack is akin to ping attack. The attacker uses "land.c" executable program to send the TCP/SYN packets with the victim's IP address. This results in sending of requests to the victim machine itself and finally, crashes.
- **HTTP flood:** The HTTP flood attack is a type of DDoS attack that does not use malfunctioned packets, spoofing or reflection techniques, but rather the attacker of this attack attacks the web server or the application by exploiting the legitimate HTTP GET or POST requests. So the attack requires less bandwidth than other attacks to attack the targeted site or server. The attack is more impactful and effective when it makes the server or application to reserve maximum possible resources in response to each single request.

- **TCP flood:** In this attack, the attacker sends Transmission Control Protocol (TCP) messages with spoofed return addresses. This is flooded to the random port of victim host with numerous UDP packets which force the host to listen on that port continuously for an application. The continuous listening to the port by the host saps the resources of the host which leads to inaccessibility of legitimate users..
- **HX-DoS attack** [29]**:** HTTP and XML attacks combinely known as HX-DoS attacks. This type of attacks mainly destroys the communication channel in cloud environment. This attack includes HTTP and XML type of message requests.

2.2.2 DoS and DDoS Attack Tools

DoS and DDoS attack generation tools make the attackers select the DoS and DDoS attacks to shut down the cloud server [15]. There are many tools to generate the attack over the internet [15]. Some of the common tools which are used by the hackers are listed below [15].

- **Trinoo** [30]: This tool can be used to launch the UDP flooding attacks. This tool provides the flexibility to control a number of Trinoo master machines as this tool is deployed with master/slave architecture. The communication between attacker to master is implemented through TCP protocol whereas master to slave is through UDP protocol. The attacks are generated for various systems running various services that are remotely exploitable buffer overflow threats like RPC services. Windows version of Trinoo is called Wintrinoo.
- **TFN** [31]**:** This tool can implement Smurf, ICMP flood, SYN Flood, UDP Flood attacks. This tool uses ICMP echo packet reply for the communication between master and slaves.
- **TFN2K:** This tool is designed to implement Smurf, ICMP flood, SYN Flood, UDP Flood attacks. It is also capable of performing vulnerable attacks by sending mangled packets. This is the most precocious of the primitive TFN network. The communication between master, slave, and the attacker is done using TCP, UDP, and ICMP protocols. The encryption for the communication between the attacker and master is done using a key-based CAST-256 algorithm.
- **Stacheldraht:** This tool combines the best features from both Trinoo and TFN tool. This tool is used to generate Smurf, ICMP flood, SYN Flood, UDP Flood attacks. The greatest advantage of this tool is that it can update slave machines automatically. The communication between master and attacker uses an encrypted TCP connection whereas the communication between master and slaves is through TCP and ICMP protocols.
- **Shaft** [32]**:** This tool is similar to the Trinoo tool. It can perform ICMP flood, TCP flood, and UDP attacks. The idiosyncratic feature of Shaft is to switch control master servers and ports in real time which makes detection of an intrusion by IDS tools difficult. The communication between master and slave machines is achieved through UDP but the communication between master and attacker is done through TCP telnet connection.
- **XOIC:** This tool is used for launching one or more types of attacks and ICMP flood attacks. This tool only works on Windows 7 and later systems. Various features provided by XOIC are test mode, normal DoS attack mode, and DoS attack with TCP/HTTP/UDP/ICMP Message. Both XOIC and LOIC can be used to start an attack by flooding the server with connection requests but in XOIC the performance of the system in sending of attack packets can be checked using the Test Mode feature.
- **High Orbit Ion Canon (HOIC):** This tool can be used to launch an attack only by sending valid HTTP packets. Once the packets are sent, HOIC extracts information from these fields such as target URLS and then uses this information to create an HTTP attack packet

- **Low Orbit Ion Canon (LOIC):** This tool is mainly used for targeting web servers by sending TCP packets, UDP packets and Http request to a target server for testing. LOIC can be used to create a design bug written in C# which when left in a tool makes it impossible from the user to stop an attack before completely exiting the current running process. This helps in increasing effect of the attack. The attack is carried by sending large amounts of data to a single IP address repeatedly.
- **Commview visual packet builder:** This tool is mainly used to generate Land Attack and Teardrop Attack. For Teardrop Attack, two fragmented packets with overlapping offset values must be built which belong to the same original packet and also have the same ID. This ID has a value assigned by the sender host to help in assembling the fragments. In Land Attack a spoofed TCP SYN packet is used which has IP address has been set to the destination IP address, the source port number to destination port number and finally the destination MAC address is set to the MAC address of the attacker host's gateway.
- **Http Unbearable Load King(HULK):** This tool is mainly used to avoid attack detection via known patterns. This tool generates a unique request for each and every generated request to hide traffic at the web server.

The comparison of the tables with the types of attacks implemented is provided in appendix A.

2.3 Intrusion Prevention Systems

Firewalls are capable to prevent the intrusions from the external sources by analyzing the packet headers. The data packets are considered as malicious and dropped by the predefined policies based on protocol type, source and destination address, source or destination ports [11]. IDS will analyze the whole packet, i.e., packet header and payload and if the attack policies are matched it generates the alerts of the attack [33] this is a software program which automatically checks the intrusion detection process. The software program which able to prevent the intrusions along with the detection capacity is known as IPS.

The main advantages [33] of using IDPS technologies is to identify the intrusions and to log the details of the intrusion for further analysis. These technologies can also prevent the attackers for limited period of time which helps the network administrators and CSP's to take counter measures to the attacks. The preventing of attacks in IDPS technologies can be done by any of the three ways

- The IDPS terminates the network connection or the secession which is established to perform the attacks.
- Blocking the access to the target server by offending user account or IP address.
- "Block all access to the targeted host, service, application, or other resource" [33].

IDPS uses some detection methodologies to detect the attacks. They are

Signature-Based Detection: In signature-based detection IDPS has some predefined signature patterns of the attacks which it verifies with the observed events to identify the malicious incidents.

Anomaly-Based Detection: In Anomaly-based detection has the predefined profiles of normal behavior of users, hosts, Network connections and applications. If there is any mismatch observed with respective to the profiles, then it alerts for the malicious incidents.

Stateful Protocol Analysis: This is similar to anomaly-based detection methods. This contains the vendor-developed universal profiles [33] of benign protocol activity. The attacks are detected based on the tracking the state of protocols.

2.3.1 IDS Classification

According to [11] IDS can be classified into four categories which are designed for CC. They are

Host-Bases IDS (HIDS): This monitors the log files, user login information, and security permission details of user to detect the intrusive behavior of the user.

Network-Based IDS (NIDS): The behavior of the data packets is tracked by IP and transport layer headers which in turn compares the behavior of packets with previously logged behavior inn real time.

Hypervisor-Based IDS HyIDS): These are designed to analyze and to scrutinize the communication between Virtual machines (VM's) hypervisor based virtual network and between the hypervisor and VM's.

Distributed IDS: the combination of more than two IDS mentioned above which communicate each other via a central analyser. These individual IDS are placed across a large network.

2.3.2 IPS Types

According to [6][11][33][34] there are two types of IPS which are designed for CC. They are

Traditional IPS (TIPS): the IPS technologies which have been around many years are known as TIPS. The detection process in this type of IPS will be implemented through in-band sensors or applications which were configured with a generally followed known threat signatures.

The main disadvantages of using TIPS are

- They cannot be updated with latest detection profiles according to the updated attack behavior pattern.
- They cannot satisfy the requirements of high-speed networks.
- There will be no uniform standard of metric for evaluation.
- They cannot be applied efficiently to mobile networks.
- There will be an frequent variation of traffic profiles which makes training of IPS difficult.

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

- Application awareness: These NGIPS provides the flexibility to restrict access to specific applications.
- Update profiles: These NGIPS systems are flexible to update the intrusion detection profiles timely based on the real time traffic learning manually or automatically.
- Automated Response: this automatically responds to the threats identified based on the policy.

2.4 Research area

Cloud Computing is concerned as one of the field where there is an enormous research that is taking place. Within this field, security is concerned as most important research area because of increase effect of DoS and DDoS attacks. To prevent those attacks, Next - Generation Intrusion Prevention Systems are the technologies that are being used in Cloud Computing. As motivated in the previous sections there is a need of empirical comparison of these techniques. So the research area in this thesis is **Next-Generation Intrusion Prevention Systems (NGIPS) for cloud computing against DoS and DDoS attacks.** The pictorial representation of research area can be seen in the figure 2.

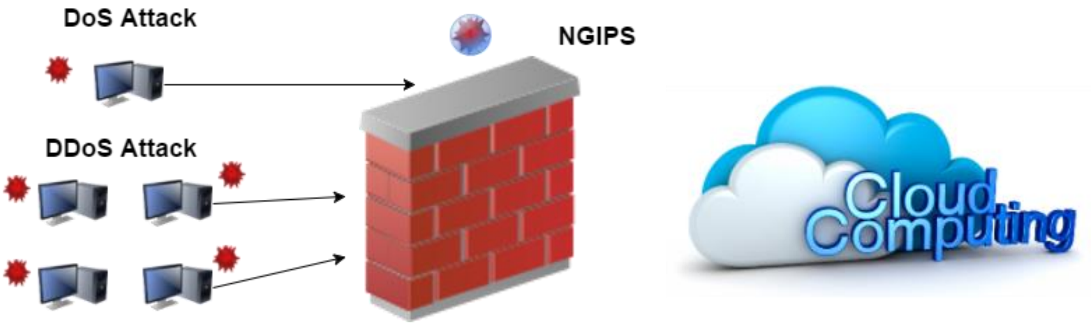


Figure 2: Research Area

3 LITERATURE REVIEW

This chapter focuses on the process carried out during the literature review. It is divided into three sub-sections and the structure is as follows:

- Section 3.1: Gives introduction about Literature review and outline for the process carried out for executing Literature Review.
- Section 3.2: Describes the planning process for answering the framed research question.
- Section 3.3: Concentrates on how the research process is carried out.
- Section 3.4: Epitomizes the results of the literature review.

3.1 Introduction

In computer science engineering discipline, there are mainly five research methods [35], [36]. They are literature review, experiment, simulation, interviews and surveys. In this research, author selected literature review as one of the research strategy to answer the first and second research questions, because according to kitchenham [35] *"literature review is the best research method for identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest."* The first and second research questions is to identifying the existing work on the research area, which emphasizes the statement of Kitchenham. So literature review is selected to answer those research questions. Literature review is conducted based on the guidelines provided by kitchenham [35]. The steps for executing SLR are as follows:

- **Planning:** The need for a literature review in this research, its context, development, and need of review protocols is explained in this section.
- **Conducting:** The process identification of literature in databases and the selection procedure and how the data is extracted is discussed.
- **Results and Analysis:** The results and analysis of the literature review are presented.

The description for each fragment is explained in the subsections below.

3.2 Planning Literature Review

3.2.1 The Need of Literature Review

The main rationale behind performing the literature review for this study is to summarize the NGIPS techniques which can be deployed in the cloud against DoS and DDoS attacks. Also the study aims to evaluate the NGIPS techniques. So identifying the existing NGIPS techniques forms the first step of this study. In order to perform this first step, the author has chosen literature review as an appropriate research method. As the literature review can help the study by identifying the existing NGIPS techniques [35], as a next step based on the data obtained from literature review the identified NGIPS techniques will be evaluated in the experiment.

3.2.2 Research Question

The research question which is answered through this literature review is:

- **RQ 1:** What are the different Next-Generation Intrusion Prevention Systems (NGIPS) that can be used in cloud computing for preventing the DoS and DDOS attacks?
  How: The literature review will help in identifying the existing prevention techniques which are used in securing the cloud environment.

- **RQ 2:** What are the different metrics used for evaluating Next-Generation Intrusion Prevention Systems (NGIPS) for Cloud Computing against DoS and DDOS attacks?
  How: The literature review will help in identifying the general metrics used for evaluating the prevention techniques which are used in securing the cloud environment.

3.2.3 Selection of Keywords

The keywords for this research is selected by using the guidelines provided by the [37]. This uses the Population Intervention Comparison Outcomes Context (PICOC) criteria in selecting the keywords for the research.

- **Population:** The area of the research is considered as population. In this research, the population resembles as "Cloud Computing".
- **Intervention:** In this research Intrusion Prevention Systems (IPS) address the security issues for the population i.e cloud computing. So Intrusion prevention systems are considered as intervention.
- **Comparison:** Comparison of Next-Generation Intrusion Prevention Systems (NGIPS) is done which is the new upcoming technology in Intrusion Prevention Systems (IPS). So the comparison in this research is Next-Generation Intrusion Prevention Systems.
- **Outcomes:** In this research the NGIPS techniques are evaluated and their performance is evaluated based on the metrics. So "performance accuracy" is selected as outcomes.
- **Context:** The comparison of IPS is based on the academic environment. So the context in this research is "academic".

3.2.4 Review Protocol

The review protocol discusses the methods that help initiate the literature review to reduce the subjective bias and also ensures that the literatures collected should be relevant to the research. The components of the review protocol include Library Database Selection, Study Selection Criteria, Study Selection Procedure, Quality Assessment Checklists, Data Extraction Strategy, Synthesis of extracted data. The review protocol is designed using the guidelines of Kitchenham [35]. Each of the above mentioned protocols is discussed in the upcoming sections.

3.2.5 Library Database Selection

Based on the availability of databases in Blekinge Institute of Technology (BTH) library catalog and relevance, the author had selected the following databases for searching the articles. They are:

- INSPEC
- IEEE
- SCOPUS
- ACM
- WILEY

3.2.6 Study Selection Criteria and Procedures

Literature are selected based on the inclusion and exclusion criteria devised by the author. The inclusion criteria influence in selecting the literature. The exclusion criteria influence in rejection of literature.

Table 1: Study Selection Criteria

	Search Constraints	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	Abstracts and introductions of the literature that does not relate to the topic.	Literature which is not available by full text is excluded.	

3.2.7 Study Quality Assessment

To maximize the external validity, internal validity and to minimize biases quality assessment of the literature is essential [35]. The scale of measure for the literature is given from one to five where, 1= very poor, 2= poor, 3= medium, 4= good, 5= excellent. The above scaling is given only if criteria is discussed in the paper. If the criteria is discussed then the value is given as YES, if not discussed its value reported as NO. If the criteria are answered partially then the value is recorded "PARTIAL". By following the Kitchenham guidelines following quality assessment table is prepared.

Table 2: Study Quality Assessment

S No	Quality Assessment Criteria	Value	Scale
1	Does the literature answer the research question?	Yes/No/partial	1 to 5
2	Did the author clearly describe the research method?	Yes/No/partial	1 to 5
3	Does the selected literature discuss the validity threats?	Yes/No/partial	1 to 5
4	Does the proposed IPS method limitations are discussed?	Yes/No/partial	1 to 5
5	Did the author evaluate proposed IPS with any of the metrics?	Yes/No/partial	1 to 5
6	Number of times the literature is cited	Number	

3.3 Conducting Literature Review

In this section the author explains how the literature review is conducted.

3.3.1 Identification of Existing Research

The aim of literature review in this research is to identify the previous studies and answer the research question framed. To achieve this, a proper search string is required in order to extract the most relevant literature from the databases.

Table 3: Keywords used for search string formulation

	Group 1	**Group 2**	**Group 3**	**Group 4**
Term 1	“Cloud computing	Intrusion	Prevention	Next generation
Term 2	“Cloud environment”	Trespass	Protection	Non-tradition
Term 3	virtualization	Attack	Response	
Term 4		Penetration	Resilience	

For the search string formulation, we need keywords. Based on the keywords selected in section 3.2.3, the author has divided them into groups and their synonyms are placed into their respective groups as shown in table 3. Blank cells indicate no available synonyms. The pictorial diagram of resultant literatures that can be obtained on using this keyword can be found in figure 3.

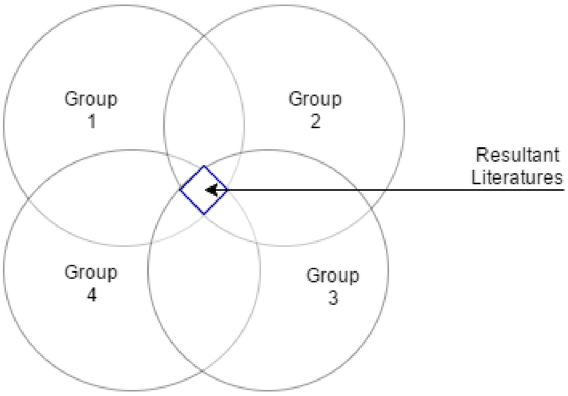


Figure 3: Venn diagram of search string formulation

3.3.2 Database search

The search string is formulated by using the two operators. They are “AND” and “OR”. The groups are connected with AND operators and the terms are connected with OR operators. The search string used to extract the literature from the scientific database is **((“Cloud computing” OR “Cloud environment” OR virtualization) AND (Intrusion OR Trespass OR Attack OR Penetration) AND (Prevention OR Protection OR Response OR Resilience OR detection) AND (“Next generation” OR “Non-traditional”))**
The search string used to extract literature from different databases are shown in table 4.

Table 4: Search strings used for extracting literature from library databases

Database	**Search String**
INSPEC	(("Cloud computing" OR "Cloud environment" OR virtualization) AND (Intrusion OR Trespass OR Attack OR Penetration) AND (Prevention OR Protection OR Response OR Resilience) AND ("Next generation" OR "Non-traditional"))
IEEE	(("Cloud computing" OR "Cloud environment" OR virtualization) AND

	(Intrusion OR Trespass OR Attack OR Penetration) AND (Prevention OR Protection OR Response OR Resilience) AND ("Next generation" OR "Non-traditional"))
SCOPUS	TITLE-ABS-KEY (("Cloud computing" OR "Cloud environment" OR virtualization) AND (Intrusion OR Trespass OR Attack OR Penetration) AND (Prevention OR Protection OR Response OR Resilience) AND ("Next generation" OR "Non-traditional"))
ACM	(+("Cloud computing" "Cloud environment" virtualization)+(intrusion trespass attack penetration)+(prevention protection response resilience)+("Next generation" "Non-traditional"))
WILEY	"Cloud computing" OR "Cloud environment" OR virtualization in Abstract AND Intrusion OR Trespass OR Attack OR Penetration in Abstract AND Prevention OR Protection OR Response OR Resilience in Abstract AND "Next generation" OR "Non-traditional" in Abstract

3.3.3 Study Selection Criteria

The selection criteria are explained in this section. These criteria are applied while conducting the research to primarily select the literature. In addition, the following inclusion and exclusion criteria are also applied to reduce the research bias.

INCLUSION CRITERIA:

- The article should describe or summarize the NGIPS techniques for CC against DoS and DDoS attacks.
- The article should cover the security overcome issues in CC environment.
- The article should discuss the effects of DoS and DDoS attacks.
- The article should evaluate or present the outcomes for the proposed NGIPS technique.

EXCLUSION CRITERIA:

- The article that does not provide theoretical evidence or experimental evidence for the proposed Non-Traditional IPS is excluded from the studies.
- The articles which are not published in the English language are excluded.
- The articles which are a part of a book or magazine.

3.3.4 Study Selection Procedure

The literature selection followed the procedure as shown in the step-by-step representation in figure 2.
Description of each step followed is given below:

STEP 1: The search string formed with the selected keywords is applied to the databases to extract the literature. The literatures are retrieved from the database by the formed search string as explained in the section 3.3.2. The number of articles selected from the initial search of each database is shown in table 5.

Table 5: articles resulted from each database after initial search

Database	**Initial Search Results**

INSPEC	16
IEEE	15
SCOPUS	14
ACM	2
WILEY	2
Total	49

STEP 2: From the initial set of articles, further filtration of articles is done to obtain the relevant literatures for the research. The study selection process is explained in the section 3.2.6. The literature is selected by going through the abstract. If the abstract is pertinent to the research question, then the article is selected to perform the literature review. If the article is inappropriate, then it is rejected. The search results after applying the selection criteria for each database is shown in table 6.

Table 6: Resultant literatures after applying study selection criteria

Database	**Resultant literatures after study selection criteria**
INSPEC	13
IEEE	11
SCOPUS	10
ACM	1
WILEY	1
Total	36

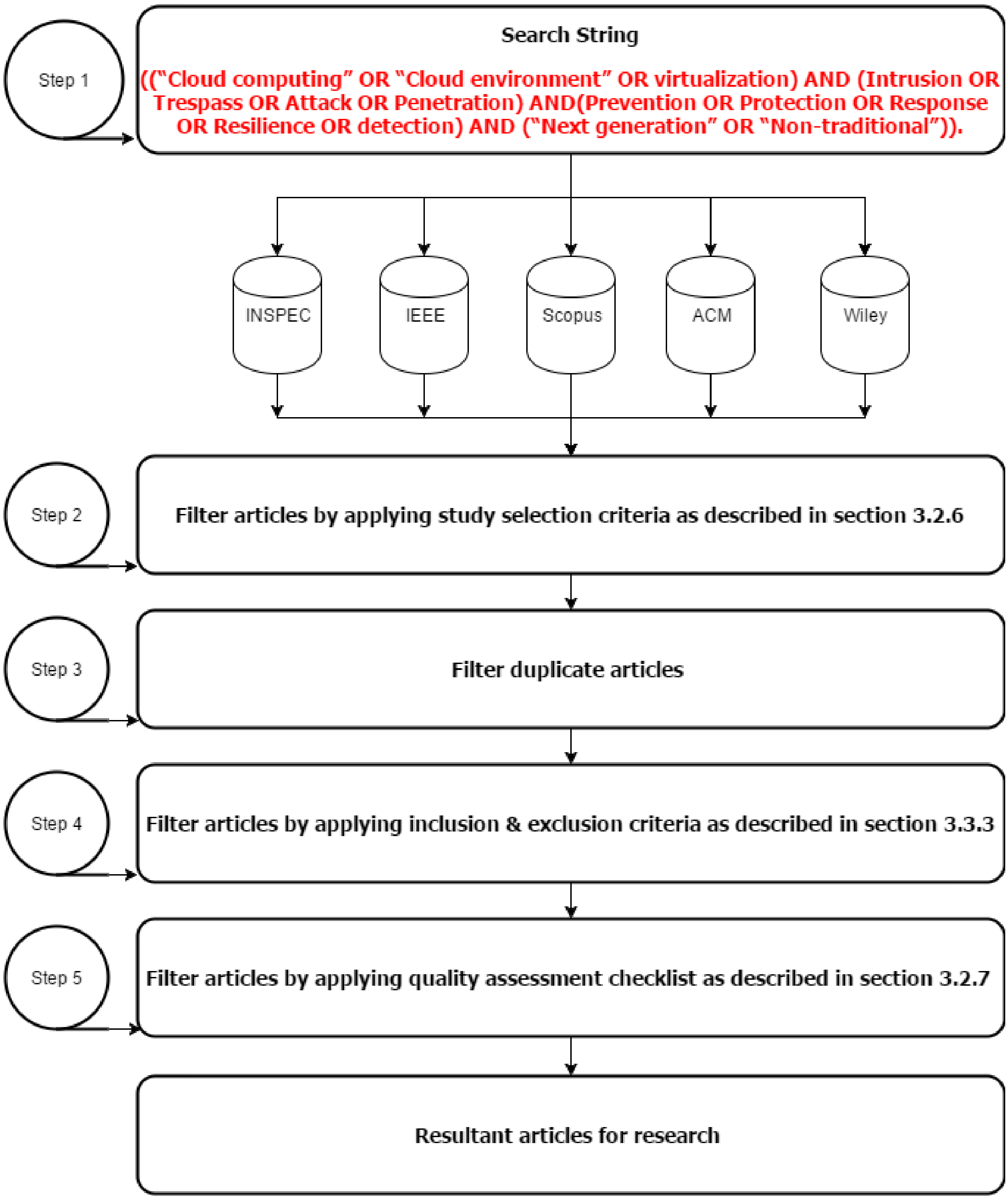

Step 1
Search String
(("Cloud computing" OR "Cloud environment" OR virtualization) AND (Intrusion OR Trespass OR Attack OR Penetration) AND(Prevention OR Protection OR Response OR Resilience OR detection) AND ("Next generation" OR "Non-traditional")).
INSPEC
IEEE
Scopus
ACM
Wiley
Step 2
Filter articles by applying study selection criteria as described in section 3.2.6
Step 3
Filter duplicate articles
Step 4
Filter articles by applying inclusion & exclusion criteria as described in section 3.3.3
Step 5
Filter articles by applying quality assessment checklist as described in section 3.2.7
Resultant articles for research

Figure 4: Study Selection Procedure

STEP 3: After Step 2, some duplicate articles were found, which were selected from different databases. Those duplicate articles are removed by using Microsoft Excel where column B is filled with the literature titles and column C is filled with the authors.
The function used to remove the duplicate articles in Excel is:

=IF(SUMPRODUCT(($B$2:$B$68=B8)*1,($C$2:$C$68=C8)*1)>1,"Duplicates","No duplicates")

The resultant number of articles after the removal of duplicate articles is tabulated in table 7.

Table 7: Selected literature after filtering duplicates

Database	**Resultant articles from step 2**	**Repeated Articles**	**Remaining articles**

INSPEC	13	6	7
IEEE	11	3	8
SCOPUS	10	5	6
ACM	1	0	1
WILEY	1	0	1
Total	36	14	18

From the selected library databases, a total of 18 articles were filtered. These 18 articles also include duplicate with respective to articles in each database i.e. articles resulted from INSPEC database may contain the same articles which were resulted in IEEE database. On merging of articles from all five databases, a total of 6 duplicate articles were found and 31 articles remained.

STEP 4: For the remaining articles the study selection criteria is applied further to reduce research bias. The criteria is explained in the section 4.3.3. Table 8 shows the resultant articles on filtering.

Table 8: Selected literature after applying selection criteria

Database	**Number of articles after filtering duplicates**	**Resultant articles after applying selection criteria**
INSPEC	13	11
IEEE	9	9
SCOPUS	7	7
ACM	1	1
WILEY	1	1
Total	31	29

STEP 5: Based on the quality assessment checklist which is designed in the section 4.2.7, the articles are reviewed from each database and selected for the literature review. From a total of 68 articles, 28 articles are excluded from the study as they did not discuss the topic related to the research. The articles resulted after quality assessment for each database is shown in table 9.

Table 9: Selected literature after applying quality assessment criteria

Database	**Resultant articles after applying selection criteria**	**Resultant articles after quality assessment criteria**
INSPEC	11	10
IEEE	9	9
SCOPUS	7	7
ACM	1	1
WILEY	1	1
Total	29	28

The selected articles are reviewed to identify the Non-Traditional IPS which can be deployed in cloud environment against DoS and DDoS attacks and the metrics which are used by authors in accessing the performance of the Non-Traditional IPS. The data which is vital in performing this research is highlighted while reading.

3.3.5 Data Extraction and Primary Data Synthesis

For data extraction, the author designed the data extraction form which consists of data key and value pair. Data extraction form is shown in table 10.

Table 10: Data extraction form

Data key	Value	Research Question	Additional Notes
General			
Name of the Extractor	Name of the researcher who is performing the data extraction process		
Name of the database	Name of the database in which the article is selected		
Total no: of articles	Total number of articles obtained from the database		
Article Title	Name of the article		
Year of publication	Article published date		
Author name	Name of the authors		
Publication venue	Domain in which article published		
Research Method	Which Research Method used by the author in the literature		
Data Extraction Info			
Non-Traditional IPS	What Non-Traditional IPS technique was used to prevent DoS and DDoS attacks	RQ1	
	If several techniques were used which was most accurate	RQ1	
Attacks	Which type of DoS and DDoS attack is prevention technique mostly focused on	RQ1	
Metrics	Which metrics are used for assessing the performance and security level of IPS in the article?	RQ2	
	Which metrics are described in the article	RQ2	

3.4 Results of Literature Review

In this study, research question one (RQ1) and Research question two (RQ2) is answered through the literature review. By following the procedure described in the sections 3.2 and 3.3, the results concluded are tabulated and are shown in table 11.

This table gives the technique name used to protect Cloud servers from DoS and DDoS attacks. If the proposed method is evaluated, then the data of metrics used for evaluating the proposed NGIPS technique is recorded. If the author in the article dose not evaluate the technique, but just proposes the method then the metric used for evaluation column is recorded as "did not propose any metric". If the author in the article clearly states to which DoS and DDoS attack protocol the proposed technique prevents then the type of attack column is recorded with the attack protocol.

Table 11: Overview of literature review

S.no	Article	Technique description	Type of attack the technique is focused
1	[38]	Packet Resonance strategy (PRS)	Spoofing attacks like Impersonation, Hiding attack, Reflection attack, impersonation
2	[22]	TCP mitigation strategy using SYN cookies	Dos and DDoS
3	[39]	Software Defined Networking Architecture Implementation	Dos and DDoS
4	[40]	Defense server for application layer	Dos and DDoS
5	[41]	Snort	HTTP attacks
6	[42]	Trace back filtering system	cyber attacks
7	[43]	Enhanced DDoS-Mitigation system	DDoS attacks
8	[44]	Filtering tree in SOA model	DDoS attacks
9	[45]	CDAP	DDoS attacks
10	[46]	TPA based IDPS technique	DDoS attacks
11	[4]	hardware based watermarking framework technology	Dos and DDoS
12	[34]	maneuver IT virtualization strategy	Dos and DDoS
13	[47]	Service-oriented architecture	Dos and DDoS
14	[48]	clone multiple parallel IPSs	Dos and DDoS
15	[49]	Hybrid Cloud-Based Firewalling Architecture	DDoS
16	[50]	Dynamic Binary User Splits (DBUS)	DDoS
17	[51]	Reputation Based Service For Cloud User Environment (RESCUE)	DDoS

18	[52]	Port Lock Mechanism using SFA algorithim	DDoS
19	[53]	PCF-M2 and PCF-O2 models	DoS and DDoS
20	[29]	Reconstruct and Drop (RAD) method	HX-DoS attack.spoofing attack,
21	[54]	CLOUD BASED FIREWALLING SERVICE APPROACH	Flooding attacks
22	[55]	Flooding tool	DDoS flooding attack
23	[56]	CBF method	DDoS
24	[57]	Cloud enabled DDOS defense mechanism	DDoS
25	[58]	DCDIDP: A Distributed, Collaborative, and Data-driven Intrusion Detection and Prevention framework.	All type of attacks including DoS and DDoS
26	[59]	Track back mechanisim	DDOS FLOODING ATTACK
27	[60]	multilevel thrust filtration defending mechanism	DDoS flooding attack
28	[61]	Snort	TCP, UDP, ICMP flood attacks

3.4.1 NGIPS Techniques

The brief description of each technique is explained as follows

In [38] author uses “Packet Resonance strategy (PRS)”. The architecture of PRS mainly consists of two levels of detection methods. They are Packet Bouncer and Packet Transit Detection levels. In packet Bouncer level there will be a Reflection Mirror Node (RMN). In this level the node will log the incoming packet and a small packet is bounced back to the user of cloud. The user should reply to the node along with the same packet for authentication. The authentication is verified by investigating the MAC and IP address combination. In Packet Transit level there will be a Transparent Mirror Node (TMN). “In this level, the node will inquire for the origin Pass code which is created at the time of account creation.” If the users failed in any of the detection levels the packets are considered as malicious and are dropped.

In [22] the author uses “Transmission Control Protocol strategy using SYN cookies”. The author provides two layers of security (control packet security and Data packet security) in this strategy. Hop count filtering method and sequence number encoding strategy is used to protect cloud servers against DoS and DDoS attacks. The MAC generator differentiates the malicious packets from the legitimate packets.

In [39] the author provides evidences form survey results and concludes that combination of Software Defined Networking (SDN) architecture and cloud computing will successfully help to protect cloud servers from DoS and DDoS attacks. Author Yan.Q et.al states that enhancing the features of SDN networking and full utilization of SDN advantages will help in defeating DoS and DDoS attacks.

To prevent DDoS attacks, the application layer was targeted in paper [40]. As per the author, this system provides security against Oversized XML, Oversized Encryption, Coercive parsing, HTTP flooding, and Web Service-Addressing spoofing. This is done by filtering all service requests received using a reverse proxy protocol. This is a type of filter which is different than paper [17], which does not affect user service since no overheads are added. The security feature is enforced by accepting service requests only from a defense server. The downside is that the server is prone to flooding attacks from known users. Therefore, by adding better authentication requirements for access to the web services, this approach can be proven very useful.

In [41] author proposes a tool based IPS technique to prevent DoS and DDoS attacks. The tool used was Snort IPS. Author has written the rules in source mode. The rule states that if the requests from the single user reach the rate of 30 within one second, then consider there is a DoS and DDoS attack and trigger the filter.

The authors in [42] proposes trace back filtering system to prevent DoS and DDoS attacks in cloud which a tag is added to SOA packets to find out which route is taken by the attacker, and filters the traffic to provide security However, both methods prove futile in identifying the attack source. Spoofed IP addresses were not considered in this paper, upon which considering could provide better results.

The authors in [47] proposes a technique which helps identify the attackers, overcoming the shortcomings of Paper [42]. A greedy algorithm which involves repeated interchange of known users between newly generated nodes allows the identification of the attackers, provided they are insiders. This research was extended in paper [57], in which the selection of new algorithms in order to optimize the runtime is introduced, since the current method is considered ‘near-optimal’ by the author. Both approaches consider persistent bots whose purpose is to migrate between servers to calculate the optimized shuffle pattern. This value can be however, only estimated in real-world scenarios.

In Paper [57]focuses on security yet again in the application layer, and also removes the requirement for a stronger authentication for its users as mentioned in paper [40]. This is possible because the DDoS protections is carried out by non-ISP organizations. Both [47] and [57] do not describe how the proxy node detects the attack. Overheads are mostly dependent on the number of shuffles required and sometimes the size of the geographical area considered.

In [43] author Alosaimi.W et.al proposed the framework named “Enhanced DDoS mitigation System.” This framework mainly consists of five components. They are Firewall, Verifier Node(s), Client Puzzle server, an IPS, and a Reverse Proxy (RP). Depending upon the outcome of legitimate packet verification method (CAPTCHA) conducted by the verifier node the firewall distinguish malicious packets form legitimate packets. The IPS will check over the hidden malicious software in the data packets. The RP server will conceal and controls the location of protected servers and maintain load balance between them. The puzzle server will come into this field to control suspected users only when they cross the threshold value in RP.

In [44] author T.Karnwal et.al introduces a filtering tree security service in Service Oriented Architecture (SOA) model. The main proposal of this technique is, the user requests are converted into XML tree form and virtual cloud defender is used to protect cloud servers from DoS and DDoS attacks. To ensure the legitimacy of the data packets a signature reference element is added to SOAP request. Double signature are generated such as, “number of children, Number of Header element, Number of body element” for extra protection against XML rewriting. They are generated using hashed characteristics of each SOAP envelope.

In [45] author N.Jeyanthi et.al proposed a novel mechanism which contains “Cloud DDoS Attacks Protection (CDAP) nodes that are to be installed at cloud server. “These army nodes act as the virtual firewall without destroying the Cloud infrastructure and improve the availability of DC, even at the time of DDoS attacks” [45]. These CDAP nodes will check the legitimacy of users by the secession key which is generated and shared at the time of account creation. If the user requests increase more than N times in a particular time period then by verifying the REGISTER _STATUS the user will be placed in BLACKLIST_CLIENT table and the future requests made by the particular user will be redirected to DUMP terminal which is placed outside the cloud environment.

In [46] author R. Saxena et.al proposed Third Party Auditor (TPA) based IDPS technique for CC against DoS and DDoS attacks. This approach uses Dempster Shafer Theory (DST). In the proposed method there will be three phases. They are detection phase, conversion phase and attack assessment phase. In detection phase, Snort is used to detect and log the values of flooded packets. In conversion phase, “ front server convert alerts into basic probabilities assignments (BPA) based on the attack alerts generated by snort” [46]. In assessment phase the converted bpa’s is fused and based on normalized factor the attack is assessed. To achieve this, it uses Dempseters combination rule.

In [4] author M. Rahman proposed hardware based watermarking and filtering mechanism to provide an additional layer in defending of CC against DoS and DDoS attacks. In this technique the legitimacy of the data is cross checked by using trace back mechanism by hop count and TTL. If the authenticity is not verified, then the packet is distinguished as malicious packet and dropped without reaching the server. If the packets are approved and labeled as legitimate packets from trace back mechanism, then they will undergo for further verification which will be checked against “knowledge based database”. If any suspicious packets are found, then they will have marked as untrusted packets and will be dropped. Only trusted packets will reach the server.

In [34] author A. Bakshi et.al proposed maneuver IT virtualization strategy to protect CC against DoS and DDoS attacks. In this strategy an IDS like Snort is installed for auditing. Snort logs the in-bound and Out-bound traffic which passes through the network. If there is a spike in graph is observed then, acknowledgement for the senders end is observed. In the acknowledgements are not received then IDS requests honeypot to ping the IP address of the sender. If there is no reply received, then it is considered as DoS attack and the further requests sent from the botnets will be blocked. For further enhancing the security features the server is moved to another virtual server and routing tables are updated.

In [61] author proposes a tool based IPS technique to prevent DoS and DDoS attacks. The tool used was Snort IPS. Author has written the rules in destination mode of rate filter. The rule states that if the requests to the server reaches the rate of 30 within one second, then consider there is a DoS and DDoS attack and trigger the filter.

In [59] contains another trace back mechanism, which involves the use of a Data Protection Manager (DPM), and training data to the neural network’s filters. The system can detect the attack traffic up to a 75% accuracy within the time limit of 20ms and 1s, as per the author. This time difference can pose as a problem to even the regular users as well, due to the delay overhead. Also, the invention of IPv6 renders the usability of DPM to be less significant, than as specified.

In [48] author sushi yu et.al proposed dynamic resource allocation strategy for cloud customers against DoS and DDoS attacks. To estimate the resource allocation queuing theory based model was proposed against DoS and DDoS attacks.

In [49] author F.Guenane et.al proposed an hybrid cloud based firewalling architecture. The main components in this proposed architecture are physical and virtual. In virtual part there are specialized virtual machines in which each machine executes firewall program operations like analysis, monitoring and reporting the data packets that enter the network with dynamic resource provisioning. In physical part gives physical security of the servers. When the traffic to the physical servers gets overloaded then it is redirected to the virtual servers. The forwarding is done between the architectures by Secure Forwarding Architecture (SFA).

The system proposed in [50] helps to address some limitations of network overlays i.e. regarding hiding the address of the target server by using gateway routers. Protection is provided from attacks from the inside and compromised user host machines as per the author. This technique avoids the requirement of monitoring all the data in the network traffic, thereby reducing overhead. A bloom filter can be found in each proxy node which tests certain values and efficiently identify an attack. A warning is issued on an attack detection which reduces the number of users assigned by half until the attackers are recognized. The results from the authors simulation were claimed to be promising, but no real-world validation was provided

In [51] author N.Jeyanthi et.al proposed a three phase authentication scheme Reputstion based service for cloud user environment (RESCUE) for CC against DoS and DDoS attacks. This approach will classify users into three different categories. They are well-reputed, reputed and ill-reputed. In the first phase of filtration, a puzzle is given to distinguish between humans and automated programs. Failing in the first phase will drop the packets immediately. Based on the predefined attack pattern signatures, the network level attacks are filtered by dropping the packets in the second phase. By observing the request intervals between consequent service requests the service level attackers are dropped in third phase.

In [52] author R.Anandhi et.al proposed a solution against DoS and DDoS attacks on distributed cloud servers by port lock mechanism using Service File Access (SFA) algorithm. The main approach in this solution is to lock the port after there is an access by the user. To handle the ports at the server end Group Policy Object (GPO) is built for users and files. If the user request for accessing the file, then the tripping password is generated with a limited period validation. As the port is locked there is no chance of other users to enter the port and if the tripping password is not entered by the user within the time frame then the user is considered as malicious user and dropped.

In [53] author M.Malekzadeh et.al proposed two distinct security models. “The first proposed model is called PCF-O2, which is based on the original HMAC-SHA2-256 algorithm (O-hmac2)”.[53] This mainly consists of three main parts. They are Proposed Key Derivation Algorithm (KDA), New Security Elements, and Replay attack preventing scheme. “For the second proposed model, we modify the HMAC-SHA2-256, which is referred to as M-hmac2. Then, we apply M-hmac2 as the underlying authentication algorithm of the second proposed model, which is called PCF-M2. The M-hmac2 is developed to reduce the security cost and communication overheads of the O-hmac2 that will consequently enhance and optimize efficiency of the PCF-M2 model compared with the PCF-O2 model.”[53]

In [29] author E.Anitha et.al used a rule set based detection (CLASSIE) and modulo marking method to avoid spoofing attacks. To take the decision about dropping of the packets, Reconstruct and Drop method is used. CLASSIE which is trained to identify the pattern attributes of DoS attacks is placed at one-hop distance from the user. If the attack pattern is found, it drops the packets. After this stage the packets are sent to marking stage. In this stage a MACtoID table is maintained by the routers which contains the physical address of hosts. For marking the packets the routers uses modulo technique.

In [54] author F.Guenane et.al proposes a new architecture named cloud based firewalling architecture which mainly consists of three main components. They are Front-Gateway, virtual firewall instances and Back-Gateway. The Front-Gateway is a virtual router which is responsible for authenticating and distribution of incoming traffic to different instances of virtual firewall. The decision for distributing the traffic is taken by decision module. The firewalling module present in the virtual firewall instance is responsible for authenticating the data packets based on the inspection rule module which contains predefined rules of attack traffic pattern. Back-Gateway is responsible for re-assembling the data packets and send the legitimate packets to the server based on the packet reassembly and control modules.

In [56] author W.Dou et.al proposed an Confidence-Based Filtering (CBF) method to protect cloud server against DoS and DDoS attacks. The confidence values are generated and stored in the nominal profile based on the attribute value pairs in TCP and IP headers. Based on the nominal profile CBF calculates scores for the incomming data packets from the users. The cofidence values represent the occurrence frequency of the data packets the score given to the packets are low. If the packet score is high then the packets from that source is considered as legitimate packets and are forwarded to the server to access the cloud resources.

Huang in paper [60] presented a low reflection ratio migration system which helps in identifying the source, detecting attacks, turning testing, and generating question modules. This system was set to function before the IaaS, since it calculates computational efficiency and the implementation of overheads on actual users. Lists are generated based on whether they are a threat or not into whitelist, black list, unknown list, and block list by recognizing IP addresses. Administrators govern these APIs; however, this opens the system to potential manipulation by insiders. Operational degradation of an 8.5% is observed when monitoring 100,000 addresses

In [58] a Distributed, Collaborative, and Data-driven Intrusion Detection and Prevention framework (DCDIDP) was proposed by the authors S.T Zargar et.al. this framework is built with three (infrastructure, platform and Software) levels. For effective detection and prevention of DoS and DDoS attacks, the clusters in DCDIDP architecture interact with three (Intrusion Assessment Information Base (IAIB), Policy and Rule Base, Audit Logs) local databases. IAIB holds the information about the attack patterns, packet regular behavior, data access permission details. Policy and rule base holds both dynamic and policy based rules to provide users to control their systems at different architecture levels. Audit logs are responsible for logging sources of other IDPS events and correlate them.

In paper [55], a flooding tool is proposed to detect DDoS attacks. The authors introduce a technique which is based on distance estimation for identifying the traffic rates. The Time-To-Live (TTL) is calculated using this distance measure. Real-time measurement is also provided using exponential smoothing for the IP traffic. Conclusively, deviation is calculated to identify if the behavior is normal or not. The attribute dependencies mentioned in previous approaches are overcome in this approach using time delays. Generally, ISPs are trusted with the implementation of filters on data traffic, hence making this approach more unlikely to be adopted.

3.4.2 Metrics

In this section the identified metrics form the literatures are presented which answers the research question 2. The overview of the identified metrics can be found in the table 12.

Table 12: List of identified metrics used for evaluation of NGIPS techniques

Response Time	False positives
CPU Utilization time	Hop count
Throughput	Memory Utilization
Active legitimate connections	Bandwidth
Number of connections aborted	Latency
False negatives	Processing time
Accuracy	Traffic at nodes
Detection rate	Round Trip Time
Packet Loss rate	-

The explanation of each metric is given below

Response Time: The response time (dropping the packet) by the NGIPS with the attack packet input. This can be explained as the time difference between the attack packet generation and the time taken by the IPS to drop the packet. For the better NGIPS the packet response time should be minimum.

False positives: This describes the count for the NGIPS dropping the packets even if they are legitimate packets. NGIPS assumes that the detected packet is malicious even it is a legitimate packet. For the better NGIPS the count of false positives should be minimum.

CPU utilization time: This describes the percentage of CPU utilized by the NGIPS.

Hop count: This describes the best possible route for the legitimate packet to reach the destination from the source within the architecture of the NGIPS.

Throughput: The number of data packets received by the NGIPS per second.

Memory Utilization: This describes the percentage of memory (RAM) utilized by the NGIPS.

Active legitimate connections: This describes the count for the number of legitimate connection that are active when the cloud server is under attack. To evaluate the best NGIPS the count of active legitimate connections should be maximum with respective to the legitimate users connected to the server.

Bandwidth: Within the fixed amount of time the number of packets that can be transmitted over the network is considered as bandwidth. NGIPS should make bandwidth available to the legitimate users by dropping the malicious packets.

Number of connections aborted: This describes the count for the number of legitimate connection that are dropped when the cloud server is under attack. To evaluate the best NGIPS the count of active legitimate connections should be maximum with respective to the legitimate users connected to the server.

Latency: The time taken for the data packet to reach the server and return back to the user.

False Negatives: NGIPS assumes that the detected packet is legitimate even it is a malicious packet. For the better NGIPS the count of false negatives should be minimum.

Processing Time: The time taken by the NGIPS to read the data packets to take the decision.

Accuracy: This provides the precision fraction with respective to true positives, false positives, true negatives and false negatives.

Traffic at Node: The rate of traffic which is waiting in queue for analyzing at NGIPS node.

Round Trip Time: This can be explained as total time taken for the data sent to the server and the data received by the legitimate user of cloud.

Packet Loss Rate: The rate of packets which gets lost irrespective to the malicious and legitimate packets without analysed.

Detection rate: The rate of data packets which gets detected either as malicious or legitimate without getting lost.

4 SNORT (INTRUSION PREVENTION SYSTEM)

To generalize the experiment evaluation, the IPS techniques proposed by the authors in literature [61], [41] are modified. The modification without any change in the parameters of filters is explained in this section.

4.1 Introduction

Snort is a Network Intrusion Detection System (NIDS). It uses Libpcap to sniff and logs the packets in the network. Its rule-based content pattern matching feature detects the real time attacks like buffer overflows, CGI attacks, DoS and DDoS attacks. The filter techniques which snort provide will turn the IDS system to IPS system. Snort mainly offers to run in three different modes. They are the sniffer mode, packet logger mode, and Network Intrusion detection system mode. In sniffer mode, the packets are sniffed in the network and displayed in the console. It generates an alert messages of the packets that are being transmitted in the network. In Packet logger mode, snort logs the packets to the storage device. In NIDS mode, snort generates alert messages for the detection and analysis made on the network traffic. The rules written by the authors in [41] [61] are in NIDS mode.

The configurations to filter the malicious packets from legitimate packets provided by the snort are Rate filter, Event filter. In this research author had compared two modes of rate filter technique.

4.2 Network Intrusion Detection Mode

There are numerous ways to generate alert messages in network intrusion detection mode. Among them the authors in the literature used "alert" and "Rate_Filter" to detect and filter malicious packets from legitimate packets in the network. "alert" is used to generate the alert messages which satisfies the attribute values used in the rule file. "Rate_Filter" is a configuration parameter used to filter the packets. There are many attributes in this parameter. Among them, detection by source and detection by destination are two different modes to drop the packets.

The alert message crafted by the author [41] is given below

```
" 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;)"
```

The above rule specifies snort to drop the packets of type TCP coming from any IP address any port destined to any IP address to port 80 if it violates the sid: 1000001 thirty times in one second. The detection of the packets is done using "detection by source" mode. The attribute used for this mode is "*by_src*". This mode tells snort to spoof the packets from the source IP address. On triggering the rule with this mode, snort will drop the packets regardless of destination.

The filter rule implemented by the author [61] is given below

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

The above rule tells the snort to spoof the destination IP address when the filter is triggered and drop the packets irrespective of the source.
The detection rule implemented by the author is described below

# 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 above rule specifies snort to create a rule with sid 100001. It generates the alert messages for the TCP packets coming from EXTERNAL_NET (IP address of the user systems) with any port destined to HOME_NET (IP address of the server which IPS should protect).

4.3 Snort rule modification

The rules mentioned in section 4.2 were implemented by different authors according to their simulated environment. But, to evaluate the IPS systems a generalized environment should be created [61], [62]. The environment designed in section 5.3 are different from each other so as the rules.

The rules designed for this research environment is as follows.

#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;)

The above rules are crafted to generate the alert message for the packets crossing the IPS system. There is no rule written explicitly for HTTP packet flooding because HTTP packets communicate using TCP protocol. And to filter the malicious packets the rules written in two techniques are as follows:

- Detecting the packets by source and filtering (source mode) [41]: This tells snort to spoof the packets of source IP address and trigger the filter if the rate is reached. In this mode, the rule tells snort to spoof the packets coming from the source and trigger the filter if the rate is reached. The hypothesis [61] in this technique is, if a large number of source IP addresses are used to generate the attack then the filter will never be triggered. The IPS filter rule in this technique is written 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
```

The above filter technique tells snort to spoof the data packets coming from the source and if the rate of requests to the server reaches 30 within one second then drop the packets and abort the connections for 30 seconds.

- Detecting the packets by destination and filtering (destination mode) [61]: This tells snort to spoof the packets to destination IP address and trigger the filter if the rate is reached. In this mode, the rule tells snort to spoof the packets reaching the destination and trigger the filter if the rate is reached. The hypothesis [61]in this technique is when the filter is triggered Snort cannot distinguish between malicious traffic and genuine traffic. It drops all the packets directed towards the destination resulting in the drop of legitimate users of cloud. The IPS filter rule in this technique is written 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
  ```

  The above filter technique tells snort to spoof the data packets reaching the destination from any source and if the rate of requests to the server reaches 30 within one second then drop the packets and abort the connections for 30 seconds.

5 EXPERIMENT

This chapter focuses on process carried out during the experimentation. It is divided into five sub-sections and the structure will be as follows:

- Section 5.1 Give introduction about experimentation and outline for the process carried out for executing it and also The objective and purpose of the experiment in this research is specified.
- Section 5.2 concentrates on how the experimentation is designed and discuss the hypothesis that is tested in the experiment.
- Section 5.3 explains how experiment process is carried out.
- Section 5.4 data analysis procedure is explained.
- Section 5.5 epitomizes the results of the experiment.

5.1 Introduction

In this research, the author selected Experiment as a research method to answer the third research question because according to C Wohlin et.al.[36] among the five research methods [35], [36] experimentation is the best research method to evaluate the accuracy of the models. In this chapter, the author evaluates source mode and destination mode in Snort IPS and compares the accuracy of the modes based on some selected metrics from section 3.4.2 So experimentation bests suit this research and also to validate the results. The execution steps of experiment is as follows:

- **Planning:** The context of the experiment along with the dependent variables and independent variables in the experiment are explained.
- **Experiment Design:** Testbed design, malicious and legitimate packets generation design and metrics evaluation design are explained.
- **Operation:** The implementation of the experiment design is explained in this section.
- **Results and Analysis:** The experiment results and analysis are provided in this section.

5.1.1 Object and Purpose of the Study

The authors J. Buchanan et.al. [61] and B. Khadka et.al [41] proposes a tool (snort) based solution against DoS and DDoS attacks. According to Author J. Buchanan et.al [61], detecting the intrusions based on their destinations will give the best results in dropping the malicious packets and helps in securing the cloud server. He evaluates his proposed system using the metrics Bandwidth, CPU loading, latency, reliability and memory usage. Author B. Khadka et.al [41] states that detecting the intrusions based on their source will give the best results in dropping the malicious packets and helps in securing the cloud. He evaluates his proposed system based on the metrics CPU performance and the rate of malicious packets dropped. Both authors had used Rate_filter but not the event-filter technique in filtering the malicious packets.

Considering the context of this research i.e to evaluate the techniques of NGIPS which were resulted from findings of literature review, author had found that Snort is mostly used tool in Cloud environment to detect or to prevent the DoS and DDoS attacks. It is found that authors of [41] and [61] proposed two different techniques in snort tool. In those literatures authors had not even mentioned about other modes providing evidence that these two techniques perform in a similar way or they had some difference. This had increased the curiosity in the author of this thesis to find if there is any performance difference between those two modes. So to evaluate those techniques authors used mostly used metrics which are given in the section 3.4.

among the list of the metrics mostly used metrics are Bandwidth, CPU lading, Latency, Reliability, Memory Usage and packet loss rate. Based on the resource availability and considering the mostly used NGIPS technique list from the literature review Snort is selected for the experiment.

The other techniques are not considered for the evaluation in this research is due to the resources availability and also the expected duration to setup the other techniques which were resulted from literature review and conduct the experiment is large

The main purpose of this experiment is to compare the two proposed systems based on the metrics like Bandwidth, CPU loading, latency, reliability, memory usage and rate of malicious packets dropped in a similar environment and evaluate the best IPS technique to prevent the malicious DoS and DDoS packets entering the cloud environment. The entity that is studied in the experiment is Snort IPS.

5.1.2 Quality focus

To evaluate the IPS, a real-time traffic should be generated which is very difficult to create manually. So an automated tool, LOIC has been in this framework because this is one of the best tool in generating the real-time attack traffic to the server [63][64][65]. The three types of DDoS attacks are used in attacking the server machine. They are TCP, UDP, and HTTP-based attacks. The brief description of the attacks and the tool is given in chapter 2. To maintain the same amount of attack traffic throughout the experiment process Tcpreplay tool is used.

The process and results of the Experiment are explained in the subsecctions below. The entire process is designed and conducted by following the guidelines of Wohlin [36].

5.2 Planning

Planning of the experiment is vital for proper validation of the results [36]. The context of the experiment is designed and conducted by the non-professionals. This may not be as accurate as the experiment conducted by the professionals. In this section, the selection of independent variable and the dependent variable is made which helps in designing the experiment.

- **Independent Variable:** the traffic rate of attack packets generated per second and the IPS rule signatures which differ from source and destination mode to detect and prevent the attack is set as independent variables in the experiment.
- **Dependent Variable:** the output value speaks for a dependent variable which depends on selected metrics.

5.3 Experiment Design

Initially author of this thesis tried to setup experiment environment in the amazon EC2 cloud. But setting the experiment in EC2 cloud and generating the attack for a long instance will be expensive. And also to generate the attack in the EC2 instances requires permission form amazon which is expected to be 40 days/ which is wasted. And also to evaluate their NGIPS techniques authors [41] and [61] used virtual machines. According to [34] virtual environment is similar to the cloud environment as cloud uses virtualization techniques in providing services to it users.

In this section, the author explains how the experiment process is designed to evaluate the IPS techniques. This chapter is subdivided into three sections and the structure is as follows.

- **Section 5.3.1:** This section concentrates on how the testbed is designed and the hardware and software requirements that are used in the machines are explained.
- **Section 5.3.2:** This section discusses how the attack and genuine traffic is generated in the experiment environment.
- **Section 5.3.3:** This section epitomizes the metrics used in evaluating the IPS techniques.

5.3.1 Testbed Design

Initially In this experiment, the evaluation of both IPS techniques are done against the three different profiles of DDoS attacks individually and the results are compared. To evaluate the IPS techniques, a testbed should be created such that it should support evaluation of any IPS against these attacks. So the testbed is designed with an Apache web server with a design of basic web page and placed in FTP server. The IPS machine is placed next to the server in the network with the snort IPS configured in inline mode. The network is configured such that the traffic flows through the IPS router and reaches the server. A Tcpreplay machine is configured to with Tcpreplay tool to repeat the same amount of traffic for evaluating both IPS techniques. The two attacker machines are configured with LOIC tool to generate flooding attack to the web server. A normal user machine is set up with Jmeter tool to generate a legitimate traffic at regular intervals of time. The pictorial representation of testbed and description of machines used in the testbed are shown in figure 3 and table 12. The IPS technique further tested using the black-box testing method in an offline environment enabled to run in in-line mode.

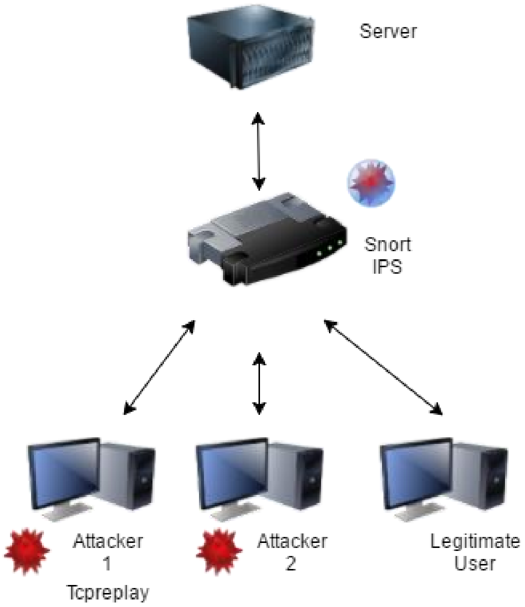


Figure 5: Test bed design

The software and hardware requirements used in the machines are described in table 13.

Table 13: Description of Machines

Item	Description
Attacker machine	The malicious user of cloud resources.
Legitimate User	The genuine user of cloud resources,

Tcpreplay	The machine to repeat the same data set of traffic
Snort IPS	The IPS router which is responsible for dropping the malicious packets.
Server	A server to provide the web resource to the cloud users.

Table 14: Specifications of Machines

Machine	**Operating System**	**CPU**	**Memory**
Server	Ubuntu 14.04	A8,2 GHz	512 Mbytes
Snort IPS	Ubuntu 14.04	A8,2 GHz	512 Mbytes
Tcpreplay	Ubuntu 14.04	A8,2 GHz	512 Mbytes
Attacker 1	Ubuntu 14.04	A8,2 GHz	512 Mbytes
Attacker 2	Ubuntu 14.04	A8,2 GHz	512 Mbytes
Legitimate User	Ubuntu 14.04	A8,2 GHz	512 Mbytes

5.3.2 Malicious and legitimate traffic component design

The LOIC [65] tool is placed in this framework for generating the malicious traffic as this provides the GUI and easy operating. It also has the capacity of generating the real-time traffic of TCP, UDP, and HTTP flood attacks which match the behavior of DDoS profile. The three flooding attacks are used against the apache web server. We have used three different DDoS attacks because, it will provide a possible view, whether the protocols have any effect in the handling of attacks efficiently for IPS technique. To generate the genuine traffic to the server a load testing tool JMeter is used. JMeter [66] is used because it is an open source testing tool developed by Apache and is most widely used for testing the web applications. This tool provides the GUI for generating the HTTP traffic with regular intervals of time as depends on a number of user threads selected.

We should ensure that same amount of real time traffic should be maintained for evaluation of both IPS techniques. Generally, there are two methods to replicate the same amount of traffic each time.

- **To use existing data sets:** Using of data sets like DARPA [67] and Bro can help in generating the traffic. Among them, the most used data set is DARPA data set.
- **To capture and save datasets:** To generate the traffic in an environment and then capture it.

Since there are many data sets better than DARPA and Bro [61], the author had selected to generate and capture the traffic. So, Tcpreplay [61] [68] tool is used as it has the capacity of repeating same data set with various options such as network speed whereas other tools like Harpoon does not generate real-time traffic.

5.3.3 Evaluation Metrics Design

The metrics that are used to evaluate the IPS in this experiment are segregated into three categories [61]. The description of the metrics is given in section 3.4.2.

- **Input metrics:** Bandwidth availability, Latency, time to respond metrics and reliability are placed in this category.
- **Resource Metrics:** CPU load and memory usage are placed in this category.
- **Response Metrics:** The rate of packets lost i.e. false positives and false negatives are included in this category.

The above metrics are selected because according to Sommers et.al [69] and J. Buchanan et.al [61] the metrics used for evaluating the IDS should be similar in evaluating the IPS and also they are the mostly used metrics as they have the similar working environment. So, the above mentioned metrics selected and also which are the most commonly used in evaluating the IDS which helps in evaluating the IPS. These list of metrics are mostly used metrics which are also resulted from the literature review.

5.4 Operation

5.4.1 Snort Implementation

Snort is the tool which is analyzed in this experiment with two different rule techniques. Snort runs in an inline mode which helps in analyzing the packets which are stored in a queue of iptables. To analyze those data packets author used

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

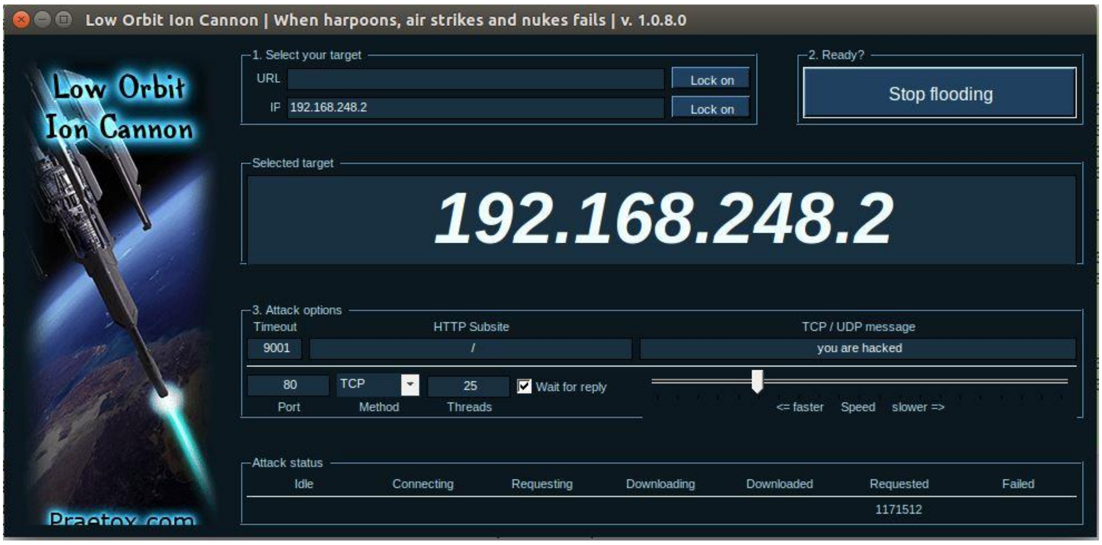
Snort -Q –daq nfq –daq-mode inline –daq-var queue=0 -c /etc/snort/snort.conf -A console -l /var/log/snort

The above commands orders snort to run in inline mode by verifying the rules in Snort.conf file and print the output to the console and make a copy in the log file. -Q orders to Snort to read packets from the queue.

The rules explained in the section are used and created a file DDoS.rules in the location /var/log/snort/. The filtering rules are implemented in “snort.conf” file.

5.4.2 Legitimate and Malicious Traffic Generation

As mentioned in the section 5.3.2, LOIC tool is used in generating the malicious traffic. This tool requires the server IP address to generate the traffic. The mode of the attack can be selected from the GUI. The pictorial representation of using LOIC tool in generating the TCP flooding is given in figure 4.



Low Orbit Ion Cannon | When harpoons, air strikes and nukes fails | v. 1.0.8.0

1. Select your target
URL
IP 192.168.248.2

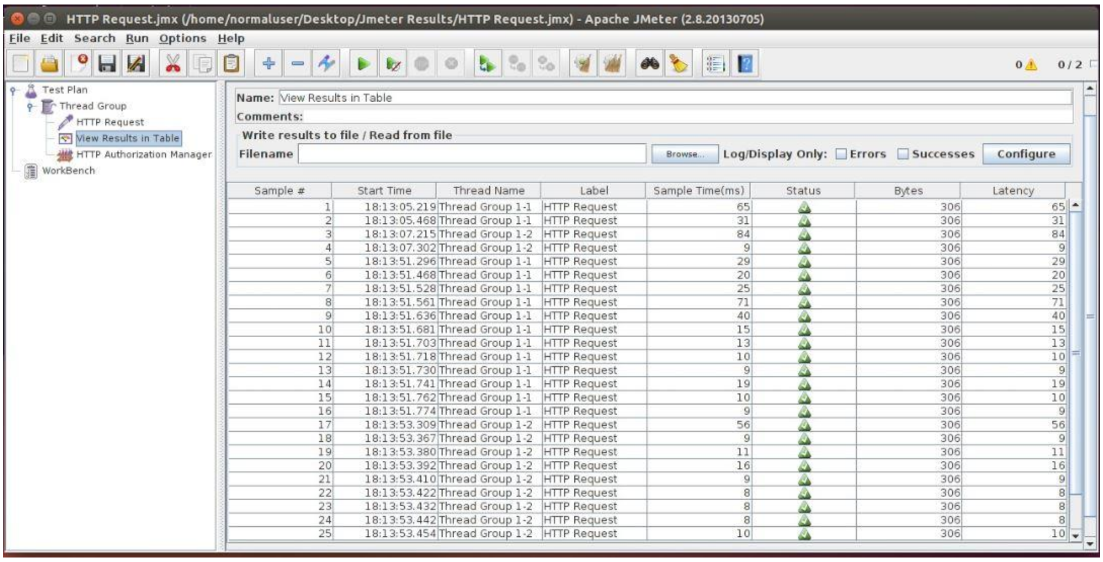
Selected target
192.168.248.2

3. Attack options
Timeout 9001
HTTP Subsite /
TCP / UDP message you are hacked
Port 80
Method TCP
Threads 25
☑ Wait for reply

Attack status
Idle
Connecting
Requesting
Downloading
Downloaded
Requested 1171512
Failed

Figure 6: LOIC tool generating TCP attack to the server

In generating the legitimate traffic Jmeter tool is used. This tool helps in generating with user threads, ramp-up period and loop count. The author had selected two user threads with a ramp-up period of 4 seconds and a loop count of 5 seconds. The pictorial representation of using Jmeter tool in generating the legitimate traffic is given in figure 5.



HTTP Request.jmx (/home/normaluser/Desktop/Jmeter Results/HTTP Request.jmx) - Apache JMeter (2.8.20130705)

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename

Log/Display Only: □ Errors □ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Latency
1	18:13:05.219	Thread Group 1-1	HTTP Request	65	✓	306	65
2	18:13:05.468	Thread Group 1-1	HTTP Request	31	✓	306	31
3	18:13:07.215	Thread Group 1-2	HTTP Request	84	✓	306	84
4	18:13:07.302	Thread Group 1-2	HTTP Request	9	✓	306	9
5	18:13:51.296	Thread Group 1-1	HTTP Request	29	✓	306	29
6	18:13:51.468	Thread Group 1-1	HTTP Request	20	✓	306	20
7	18:13:51.528	Thread Group 1-1	HTTP Request	25	✓	306	25
8	18:13:51.561	Thread Group 1-1	HTTP Request	71	✓	306	71
9	18:13:51.636	Thread Group 1-1	HTTP Request	40	✓	306	40
10	18:13:51.681	Thread Group 1-1	HTTP Request	15	✓	306	15
11	18:13:51.703	Thread Group 1-1	HTTP Request	13	✓	306	13
12	18:13:51.718	Thread Group 1-1	HTTP Request	10	✓	306	10
13	18:13:51.730	Thread Group 1-1	HTTP Request	9	✓	306	9
14	18:13:51.741	Thread Group 1-1	HTTP Request	19	✓	306	19
15	18:13:51.762	Thread Group 1-1	HTTP Request	10	✓	306	10
16	18:13:51.774	Thread Group 1-1	HTTP Request	9	✓	306	9
17	18:13:53.309	Thread Group 1-2	HTTP Request	56	✓	306	56
18	18:13:53.367	Thread Group 1-2	HTTP Request	9	✓	306	9
19	18:13:53.380	Thread Group 1-2	HTTP Request	11	✓	306	11
20	18:13:53.392	Thread Group 1-2	HTTP Request	16	✓	306	16
21	18:13:53.410	Thread Group 1-2	HTTP Request	9	✓	306	9
22	18:13:53.422	Thread Group 1-2	HTTP Request	8	✓	306	8
23	18:13:53.432	Thread Group 1-2	HTTP Request	8	✓	306	8
24	18:13:53.442	Thread Group 1-2	HTTP Request	8	✓	306	8
25	18:13:53.454	Thread Group 1-2	HTTP Request	10	✓	306	10

Figure 7: JMeter tool generating legitimate traffic to the server

The traffic generated by the tools are captured and replayed by the Tcpreplay tool. The packets are captured using Wireshark and are saved as “.pcap” file.

To replay the captured traffic the command was written is as follows

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

To evaluate the latency, the traffic is set to loop as the experiment should run for a long period of times. The command to replay the traffic is crafted as

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

In this experiment the capturing of traffic is done on the Wireshark where the IPS system is installed. Those captured packets are saved and used to replay the traffic in attacker machine.

5.4.3 Metrics Implementation

- **Rate of Packet loss:** Wireshark is used to analyze the packets sent to the server through IPS. Lost_segment packet is analyzed to show the value of packet loss in the network. To further confirmation with the results with Wireshark, the packets loss rate is calculated with the Hping3 tool. 100 Data packets are sent by the tool with an interval of one second between each packet while the IPS is under attack. And Hping3 will display the percentage of packet loss. A hypothesis is the rate of packet loss increases as there is an increase in throughput of the attack traffic.
- **Bandwidth Availability:** To calculate the available bandwidth a tool Iperf was used. This metric will help in calculating the bandwidth of IPS under attack. TCP_STREAM test was implemented to calculate the bandwidth of IPS. This test includes sending of TCP packets from Iperf client to Iperf server and the time of transmission of packets between them is used in calculating the current throughput. The command used to commence the tool is as follows

  Iperf -c <ip address of Iperf server>

  The iperf server is implemented by the command

Iperf -s

- **Latency:** The latency metrics is used to measure as the Round Trip Time (RTT). To measure the latency author used ping [70] tool. 100 packets are sent over the network to the server over IPS and the average time is calculated for the packets that acknowledged from the server. The command for ping is as follows:

  Ping 192.168.248.2 -c 60

- **Data throughput:** Wireshark is used to calculate throughput of IPS under the attack.
- **Reliability:** Malicious traffic is replayed with Tcpreplay for the five hours and the evaluation is done whether IPS can able to handle the attack for such long period of time.
- **CPU load:** *Uptime* is the Unix tool to calculate the CPU load. This metric will help in calculating the effect of malicious traffic on the resources of the system.
- **Memory Utilization:** The memory consumption of the system is calculated by using the *free* Unix tool. This metric will help to evaluate the impact of memory consumption with respective to attacking traffic.

Table 14 shows the list of tools used for calculating the metrics and the deployment location of the tools.

Table 15: Deployment location of metric tools in test bed

Metrics	**Tool**	**Machine**
Rate of Packet Loss	Wireshark, Hping3	Snort IPS, Tcpreplay
Bandwidth	Iperf	Tcpreplay
Latency	Ping	Tcpreplay
Data Throughput	Wireshark	Snort IPS
CPU Load	Uptime	Snort IPS
Memory Utilization	Free	Snort IPS

5.5 Results

By following the experimental procedure as explained in section 5.3 and 5.4, the obtained results are explained in this section. To ensure the experiment environment is same in evaluating the IPS techniques with metrics Snort IPS machine is rebooted after saving the collected data and analyzed later. Each iteration of the experiment is repeated ten times and the mean value is calculated and recorded the results section.

5.5.1 CPU Load

The CPU load varies with the packet processing load of the IPS. TCP protocol uses the fewer resources when compared to the other type of attacks. The TCP packets usage is about 50% to 60% usage of the resources whereas the UDP and Mixed packets is about 70% CPU usage. This might be because of TCP packets take less processing time when compared to other packets. For this packets when the rate of 8000pps reached the load is stable (55% to 58%) till it reaches the rate of 1000pps. The CPU load is stable because there might be some optimization process taking place by Snort when the rate is reached. The mix protocol follows the UDP packets flow as it says that the UDP protocol needs much packet processing time. The graphical representation of CPU load with different attack protocols is shown in figure 6 and figure 7 of source and destination modes.

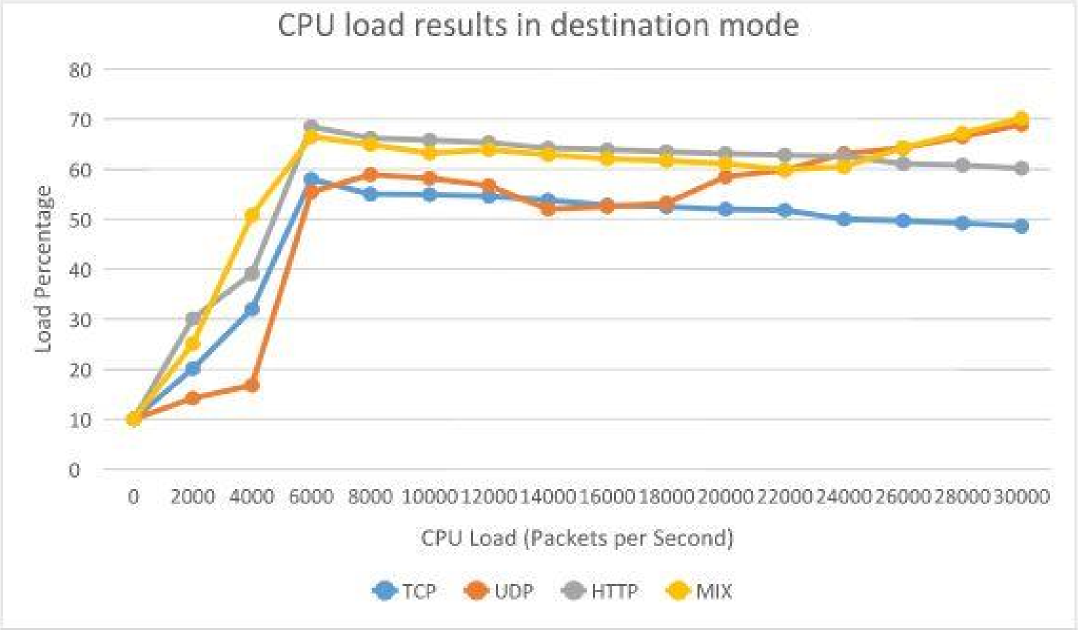


Figure 8: CPU Load results in destination mode

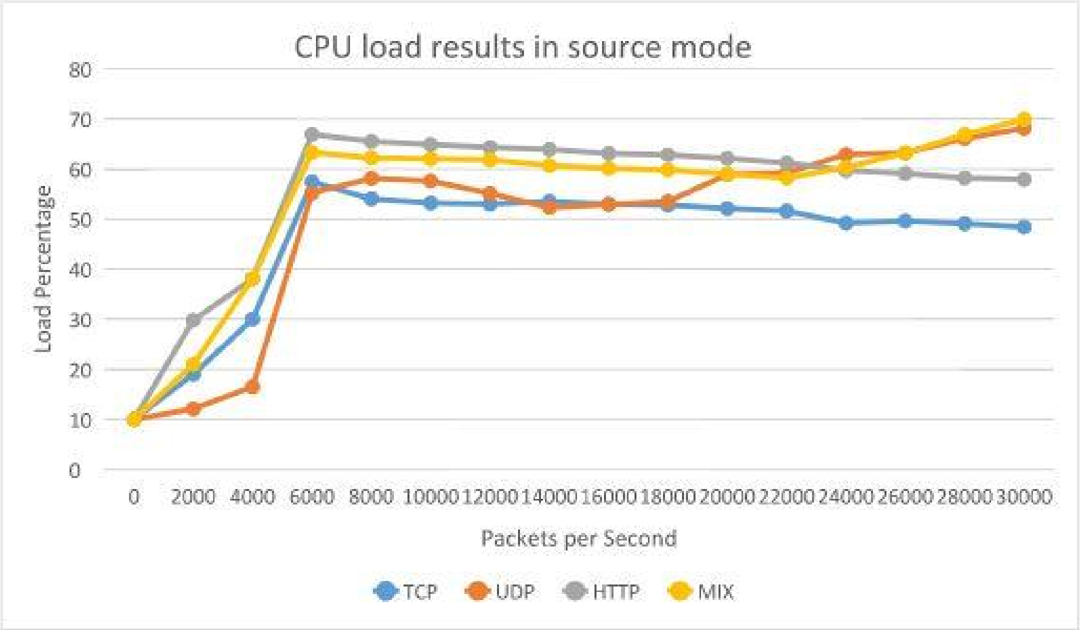


Figure 9: CPU load results in source mode

5.5.2 Memory Utilization

The memory utilization of each protocol is as expected in destination mode. But there is a huge difference in memory utilization of the protocols in this mode. The difference rate is nearly 30Mbps. Moreover, the UDP attack protocol has unpredictable memory usage as it has huge variations in the 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 source mode is almost in the same range throughout the experiment for all protocols.

The memory usage for both modes is unpredictable. This might be because of using Linux virtual machine in the same host. Using of such machine, memory cannot be predicted as there will be an influence of the host machine on the memory.

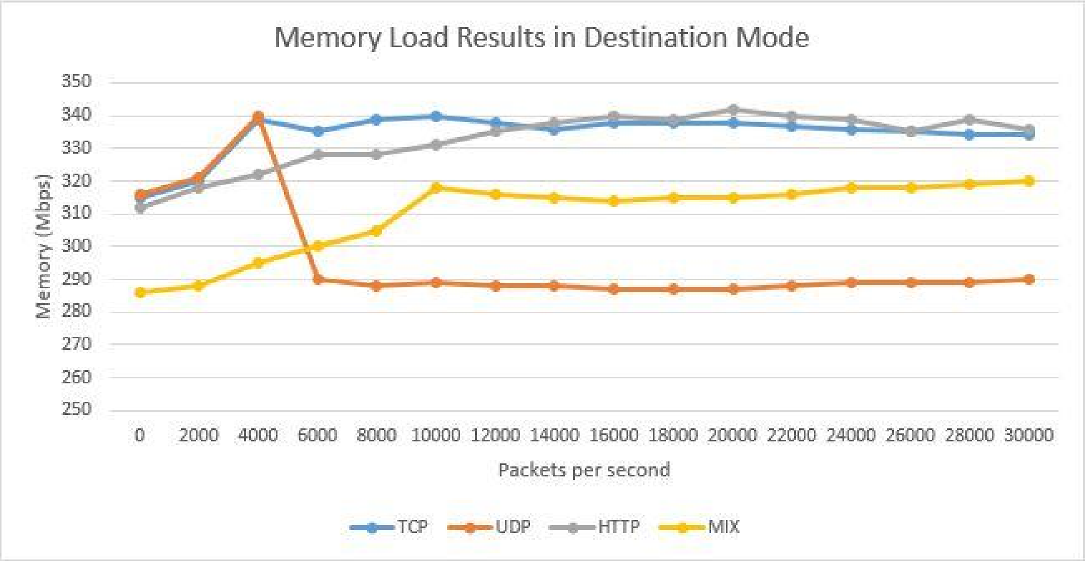


Figure 10: Memory load results in destination mode

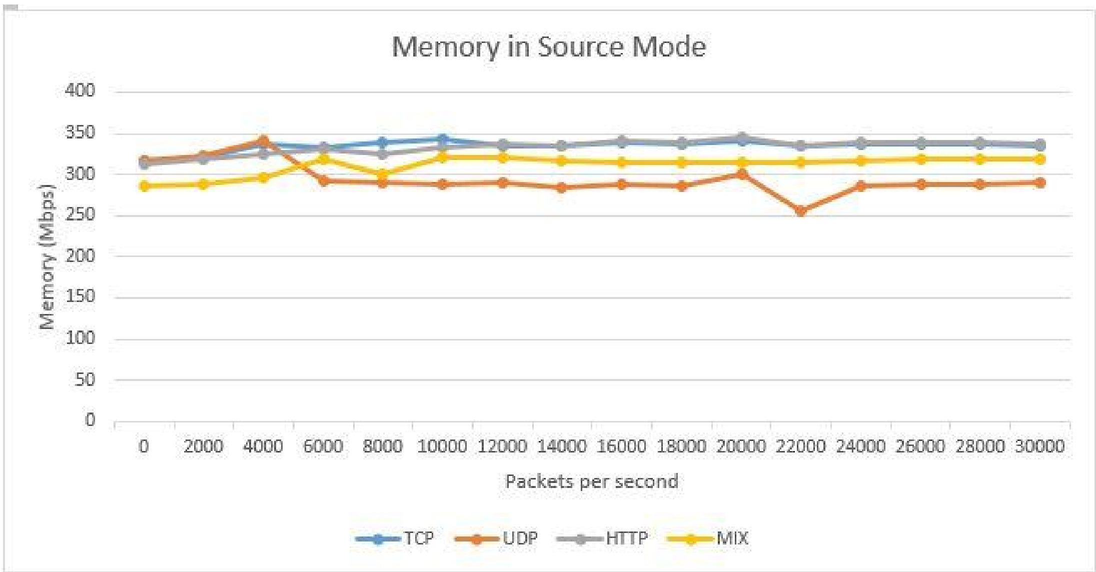


Figure 11: Memory load results in source mode

5.5.3 Bandwidth

When no traffic is on the network the bandwidth resulted was about 60Mbps. When the traffic was flooded in the network the bandwidth was reduced and when the packets reach the rate of 14000pps the bandwidth is almost zero. Snort with destination mode cannot make bandwidth available when the traffic rate reaches about 14000pps.

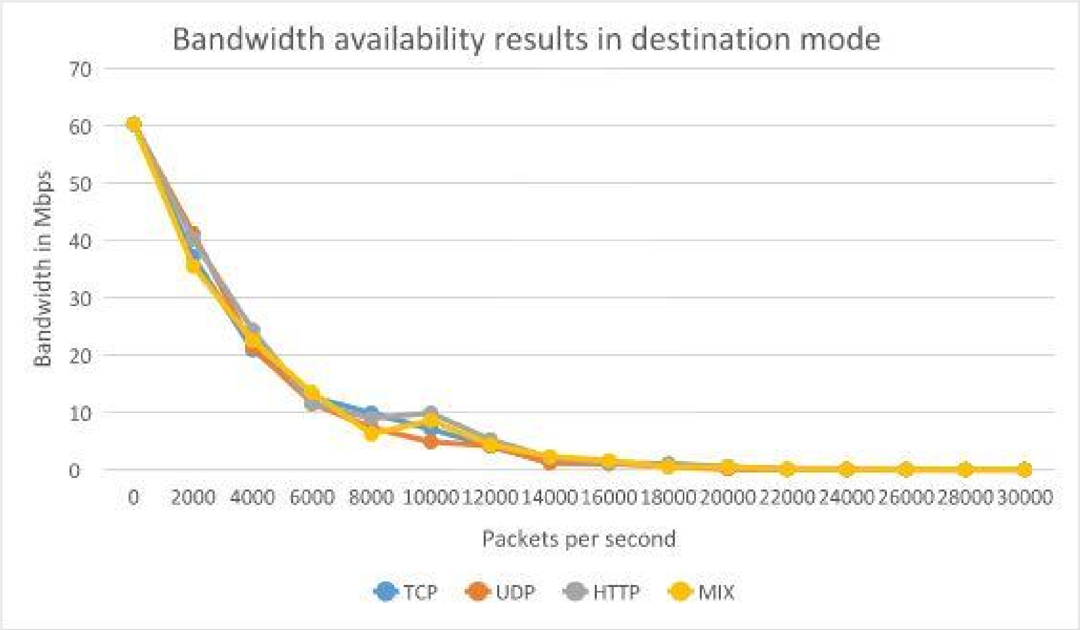


Figure 12: Bandwidth availability results in destination mode

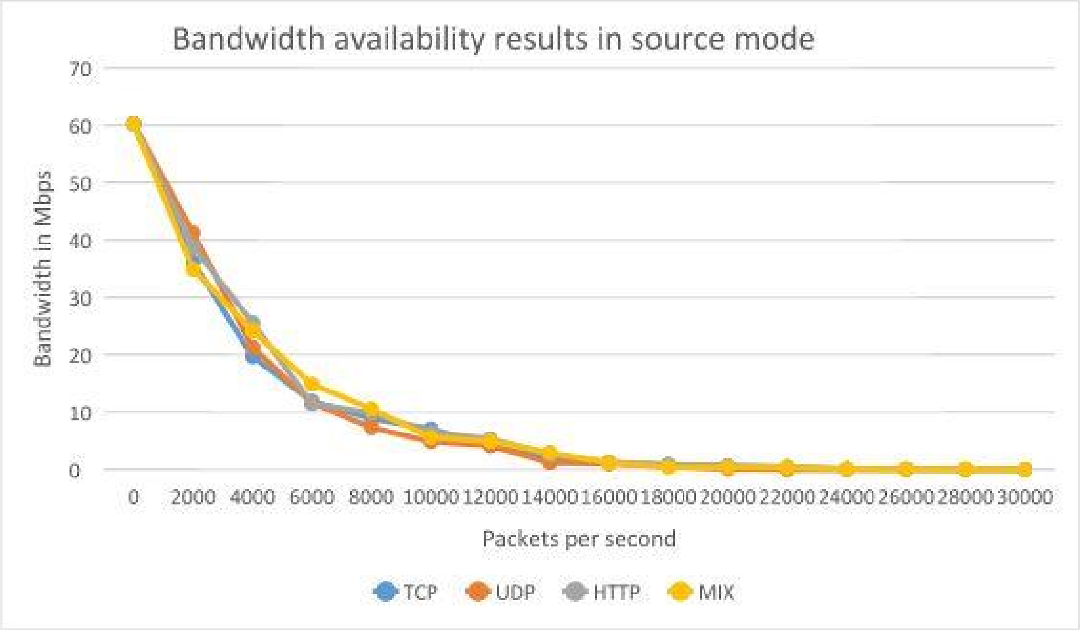


Figure 13: Bandwidth availability results in source mode

5.5.4 Latency

As the traffic rate increasing in the network the latency increases. When the rate is about 2000pps the latency of all the attack protocols lie about the same range about 5ms. The latency between the traffic rate 10000pps to 24000pps the latency of the protocols decreased which explains that snort in destination mode tries to optimize when latency is more important till its threshold limit.

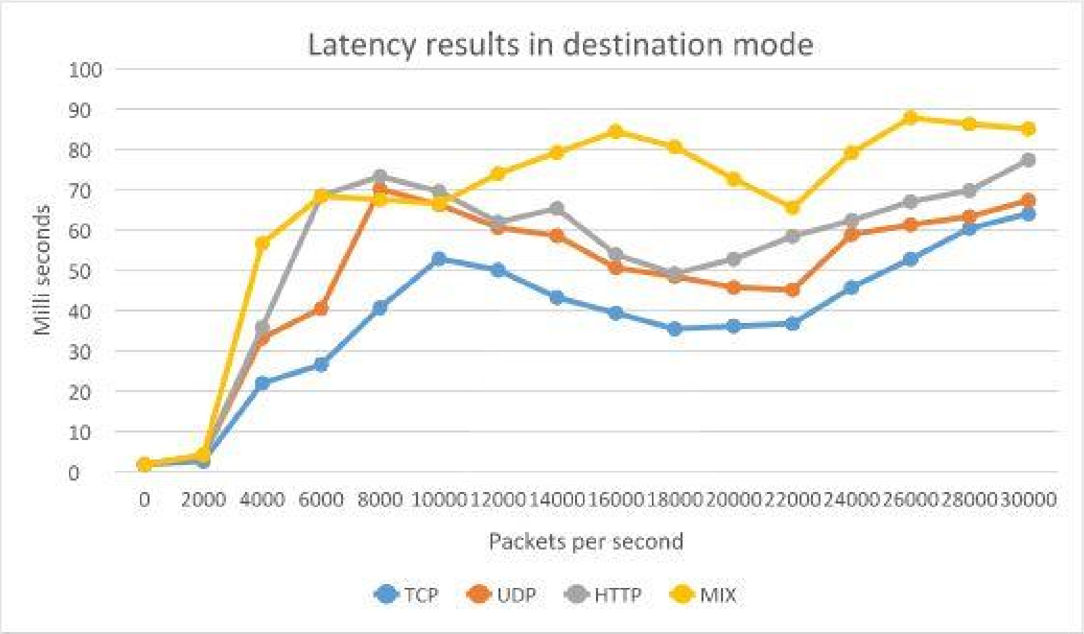


Figure 14: Latency results in destination mode

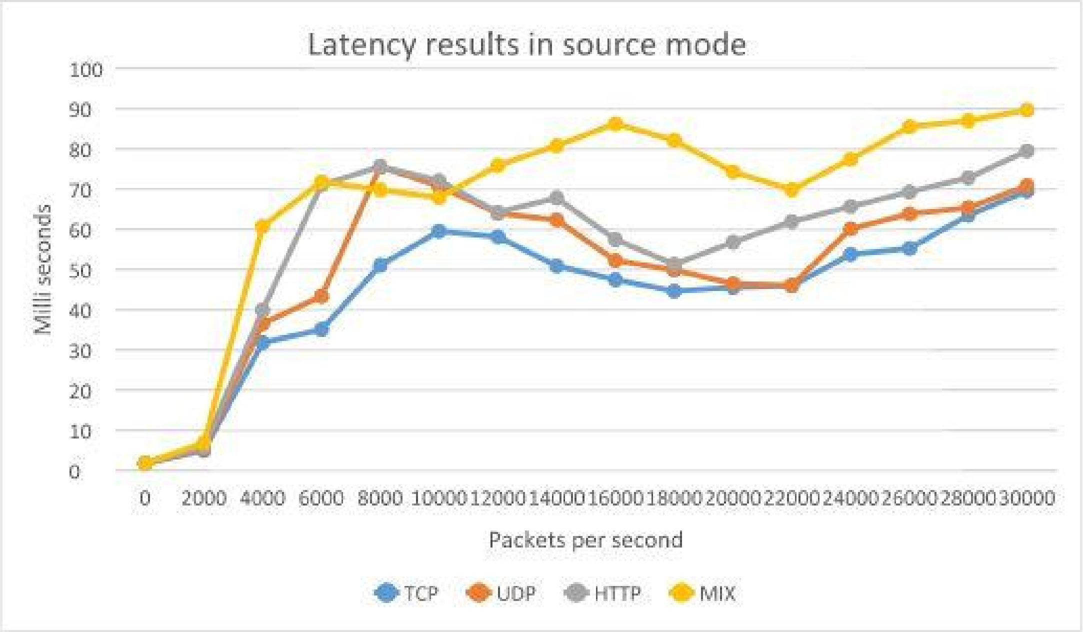


Figure 15: Latency results in source mode

5.5.5 Packet Loss Rate

The packet loss rate is directly proportional to the rate of traffic in both the modes. As the traffic rate in the network increases the rate of packet drop increases. When the traffic rate reaches 18000pps the packet drop for both the modes increases at a sharp curve. And the TCP protocol has the poorest results following with HTTP and UDP. This is analyzed as the processing time for TCP packets is more as this is to be done along with other TCP secessions.

But in source mode, the results are slightly varying as mix protocol shows less packet loss rate and can be analyzed as the rate of processing the UDP protocol because the time taken to process the ICMP packets take less time.

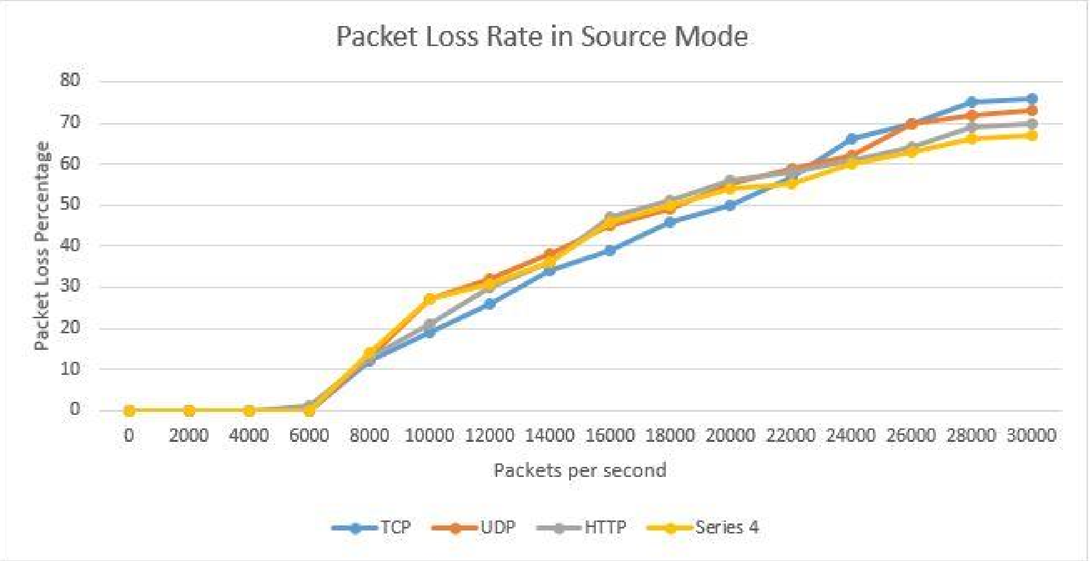


Figure 16: Packet Loss rate in Source mode

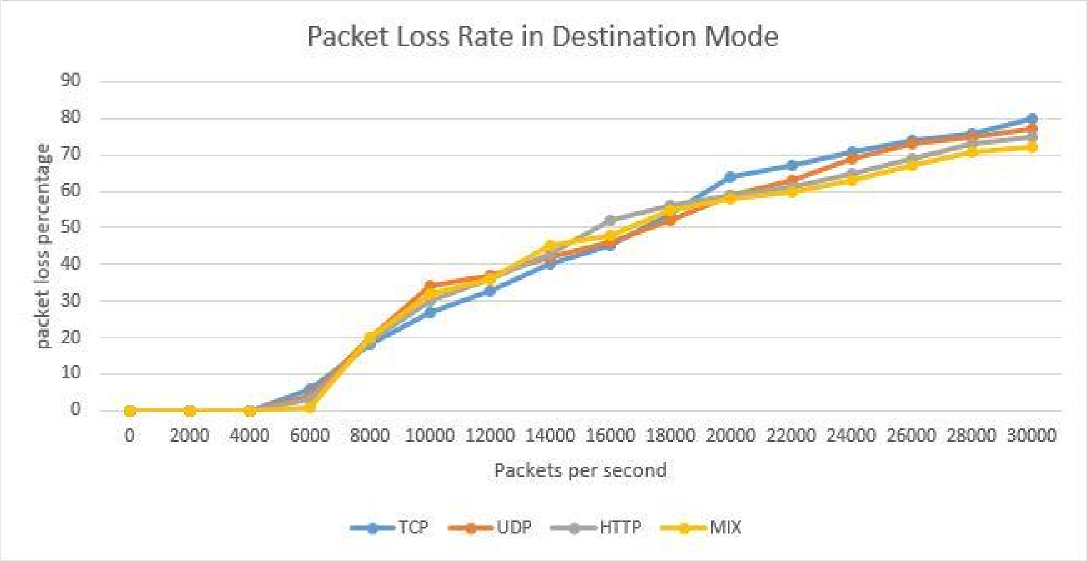


Figure 17: Packet Loss rate in Destination mode

6 ANALYSIS

To validate the results obtained from the experiment, statistical analysis should be performed [71] [72]. The statistical analysis process carried out in this research is followed by using the guidelines [36], [71].
To validate the results, author performed Wilcoxon test. This is parametric test to compare the difference between two groups which samples are correlated with each other. The author had chosen this test because the data types are considered as normal data type and there are two matched groups. Support for the selection of above statistical test can be found in the literatures [36] [73] [74] as the authors performed Wilcoxon test for the evaluation of two algorithms with the data type which is similar to data in this thesis.

6.1 Parametric Test

To perform the statistical analysis there is a need to identify the type of data variable and data distribution type [36]. On observing the data in this thesis we can say that the data is ordinal data [75]. Now to find the sample is normally distributed or not author had performed histograms which is one of the method among Kolmogorov-smirnov and shapiro-wilks tests [71]. The result has not provided any evidence to support that the data is normally distributed. So the author had done Wilcoxon test which is a non-parametric test for paired sample groups.

6.2 Mean Square Error (MSE)

The author repeated the experiment ten times with respective to each traffic rate (i.e. 2000pps, 4000pps, and so on till 30000pps) in both source and destination modes. The mean value is calculated and recorded for the conducting of Wilcoxon test. Experiment is repeated and the mean is calculated because according to [71] repetition of the experiment and calculating mean provides better results. But the mean calculation may not always provide validation for the repeated values. So to provide the validation for the repeated values author calculated "Mean Square Error (MSE)".

"you can evaluate the MSE to know about how your mean estimator is good with respect the standard deviation of the sample data. If your MSE is small, it means that the average value of the parameters is a good representation of the measured data" -Emiliano Casalicchio

The procedure followed by the author in calculating MSE is shown in step-by-step manner.
Step 1: The error value is calculated for the repetition values by using the formula

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

where μ= mean value for the data set.
X = corresponding data set value

Step 2: Calculate the square for the error values. Executing of this step eliminates negative values present in the data set of error value.

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

Step 3: Now MSE is calculated using the formula

$$\text{Mean Square Error (MSE)} = \text{Sum of Squared Error (SSE)} \div N$$

Where SSE = sum of squared errors and N is the number of sample and
N= number of population.

It is found that MSE resulted for all data samples ranges between 0.06 to 0.71. By this we can conclude that the mean value which is recorded to calculate Wilcoxon test gives best results.

The example process for calculation of MSE in source mode with CPU load metric when traffic rate is 2000pps is shown in table 16.

Table 16:Calculation of MSE for 2000pps in source mode (CPU Load metric).

PPS	In Source Mode		
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 population (N)			16
Mean (μ)			11.58
SSE			6.33
MSE			0.63

6.3 Wilcoxon Test

As described in the chapter 5, in this thesis there are two groups (source and destination) where the performance is to be evaluated. The performance of both groups is measured with the same rate of packets (packets per second). So there is one level of matched independent variable with two dependent groups.

Now by analyzing the type of data of this thesis with the statistical data types, it is concluded that dependent variable data is normal data [75]. According to [36] [71] [73] the statistical method that is used to validate this type of data is "Wilcoxon test". The author performed signed ranked test as follows

Hypothesis assumption

To perform the statistical analysis Null hypothesis and Alternative hypothesis is designed. Null hypothesis designed was

Ho: both modes in snort has no difference between them

H1: there is a difference between the two modes.

The performance difference $d=(X_i - Y_i)$ where X is the destination mode data pair and Y is Source mode data pair is calculated between the two modes and "sign" S was given to the difference values stating +1 for positive value difference and -1 for negative value difference.

Now absolute difference for the performance difference is calculated and corresponding rank is given starting from 1 to 16 as we have 16 number of population. Signed ranked test is calculated by the multiplication of S and rank of the data pairs. Then the positive sum T^+ and negative sum T^- of the signed rank is calculated. Author absorbed that the least value among T^+ and T^- in every metric is less than the critical value of significance level of 0.05.

So based on that for every metric which is considered in this thesis claims the rejection of Null hypothesis. Stating that there is some difference in performance of both the rate filter modes of Snort NGIPS technique.

The statistical table for the CPU Load metric for source and destination mode is shown in the table

Table 17: Wilcoxon Test for CPU Load

Positive Sum (T^+)	114
Negative Sum (T^-)	-17
Test statistic	17
Critical value	29

For all the metrics the data resulted was similar saying that Null hypothesis can be rejected. So by statistical analysis we can say that there is some difference between two modes in Snort NGIPS. As this is the case in all the metrics, we have enough evidence to prove that both modes has some performance difference between them.

Now to know which mode performs better by observing the T^+ and T^- values the negative sum T^- value is less than the critical value. So we can say that destination mode performs better when compared to source mode.

7 DISCUSSION

This chapter discusses about validity threats in this research and solutions to the research questions.

7.1 Threats to validity

7.1.1 Internal validity

The internal validity refers to the variation in the dependent variables of the study which are caused by the independent variables.

- In experimentation number of packets transmitted in the network are limited as this may not resemble the real time traffic generation.
- In capturing the traffic Wireshark is used. This tool initially logs the incoming data. Sometimes the wireshark tool drops some packets which can affect the data transmission from source to destination.

7.1.2 External validity

According to [36] "*external validity are conditions that limit our ability to generalize the result of the experiment to industrial practice".* In this research, the external validity can be as follows

- The test bed is designed using the virtual machines on the same host. This may bias the results achieved when designed in the actual cloud environment. So this thesis can be generalized to virtual environments.
- According to [6], it is shown that IPS should have the capacity to send 80% of maximum throughput. But in this experiment the throughput of attack traffic is not calculated. This can be a threat.
- In the current study LOIC tool is used to generate the attack packets. There are similar other tools as mentioned in the section 2.2.2. As the characteristics of the packets generated from these tools varies based on the tool, the results from this study are only confined to traffic that is similar to LOIC generated traffic.

7.2 Answers to research questions

RQ 1: What are the different Non-Traditional Intrusion Prevention Systems (IPS) that can be used in cloud computing for preventing the DoS and DDOS attacks?

ANS: Literature review is conducted to answer this research questions. The NGIPS techniques that can be deployed in the cloud environment are represented in the table in the section 3.4.1. there is no certain standard technique that can be used to deploy it in cloud computing. We can conclude that there is no standard semantics that are considered for developing a NGIPS. From the results of Literature review it found that snort is one of the mostly used tool either for detecting or preventing the attacks. Authors in [59] [41] [52] used snort along with their propose darchitectture to defend against DoS and DDoS attacks.

RQ 2: What are the different metrics used for evaluating Next-Generation Intrusion Prevention Systems (NGIPS) for Cloud Computing against DoS and DDOS attacks?

ANS: Different metrics were collected form the literatures obtained on performing literature review. The list of identified metrics are given in the section 3.4.2. There are some common metrics that are used in evaluating the developed techniques. Some common metrics used for evaluating the NGIPS techniques are CPU load, Memory usage, bandwidth availability, throughput, true positive rate, false positive rate, true negative rate, false negative rate and accuracy. For detecting the attack packets most of the NGIPS techniques will sniff the networks and are based on the signature based detection algorithms. This can be explained as NGIPS techniques are developed mostly are network-based IPS and follow signature based algorithims. But there are certain mechanisms like packet resonance strategy implementation [38] which uses schematic detection method in preventing the attacks reaching to the data center

RQ 3: What is the performance of "track by source" mode and "track by destination" mode in Snort Non-Traditional Intrusion Prevention System (IPS) against DoS and DDoS attacks?

ANS: When two RATE_FILTER modes in the Snort are compared with the metrics CPU utilization, memory usage, latency, bandwidth and rate of packet loss it proves that there is some difference exists between those two modes. From the statistical analysis results there is enough evidence to prove that destination mode provides better results when compared to source mode.

The CPU load depends on the traffic rate in the network. When the rate of traffic increases Snort drops the packets with respective to the traffic rate.

Regarding the packet drop rate both modes show similar results based on the trigger rate of the rule.

Using the source mode has a stable memory usage. But in destination mode the rate of memory is increasing as there is an increase in the traffic rate when TCP traffic is not considered. As the testbed is implemented in a virtual machine in the same host the memory results cannot be validated. So using the rate filter technique with any of the modes does not affect the cloud service provider

if the attacker uses different zombie machines with different source IP address less than the threshold limit of the rule filter. And of the rate filter in destination mode will never drop the packets of legitimate users also if the filter is triggered due to the malicious activity of attackers.

When the characteristics of the two modes are studied the time of rule trigger can vary based on the attacker. If the attacker uses many botnets which is less than the rate limit declared in snort rules, then the rate filter in source mode will never trigger. And the destination mode there is a chance of dropping the legitimate user packets if the server requests reach the threshold limit of the user.

8 CONCLUSION AND FUTURE WORK

This research was an attempt to compare two rate-filter modes in snort NGIPS technique because Snort is the mostly used technique in cloud environment to defend against DoS and DDoS attacks. The evaluation is done based on the metrics to determine which technique is better. The statistical analysis is performed to validate the data.
Literature review results show that NGIPS techniques mostly concentrate on network-based intrusion detection for handling the DoS and DDoS attacks. While sniffing the network for the malicious packets the NGIPS use signature based methods.

The experiment results show that Snort (both in source mode and Destination) is capable of handling DoS and DDoS attacks effectively till the packet traffic rate in the network reaches till 30000pps. But the main drawbacks are the rate of packet drop. The Snort drops most of the legitimate packets along with the malicious packets (true positives). This results legitimate users cannot have access to the server.

The both systems have shown high reliability in handling the attacks up to five hours. But the other metrics are not validated during this experiment. So there might be some packet drops of legitimate users along with the malicious packets as the time increases.

The DoS and DDoS methodology implementation in the real time may be much complex when compared to the experiment environment designed in this research. but the attack protocols used in this experiment gives strength to validate the results of two modes of Snort rate_filter techniques.

The future, this work can be extended by considering other attack tools it in the same test bed and evaluate 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 test bed.

REFERENCES

[1] P. Yadav and S. Sujata, “Security Issues in Cloud Computing Solution of DDOS and Introducing Two-Tier CAPTCHA,” *Int. J. Cloud Comput. Serv. Archit.*, vol. 3, no. 3, pp. 25–40, 2013.

[2] K. Hwang, S. Kulkarni, and Y. Hu, “Cloud security with virtualized defense and reputation-based trust management,” in *8th IEEE International Symposium on Dependable, Autonomic and Secure Computing, DASC 2009*, 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,” *J. Netw. Comput. Appl.*, vol. 34, no. 4, pp. 1097–1107, 2011.

[4] M. Rahman and W. M. Cheung, “A Novel Cloud Computing Security Model to Detect and Prevent DoS and DDoS Attack,” *J. Adv. Comput. Sci. Appl.*, vol. 5, no. 6, pp. 119–122, 2014.

[5] D. Sequeira, “Types of,” in *Intrusion Prevention Systems: Security’s Silver Bullet?*, 2003, pp. 36–41.

[6] S. Piper, CISSP, and SFCP, *Intrusion Prevention Systems for Dummies*. 2011.

[7] K. Nagaraju and R. Dr.Sridaran, “An Overview of DDoS Attacks in Cloud Environment,” *Int. J. Adv. Netw. Appl.*, pp. 124–127.

[8] A. Harper, S. Harris, J. Ness, C. Eagle, G. Lenkey, and T. Williams, “Gray Hat Hacking The Ethical Hackers Handbook,” Jan. 2011.

[9] R. V. Deshmukh and K. K. Devadkar, “Understanding DDoS attack & its effect in cloud environment,” *Procedia Comput. Sci.*, vol. 49, no. 1, pp. 202–210, 2015.

[10] S. Subashini and V. Kavitha, “A survey on security issues in service delivery models of cloud computing,” *J. Netw. Comput. Appl.*, vol. 34, no. 1, pp. 1–11, Jan. 2011.

[11] A. Carlin, M. Hammoudeh, and O. Aldabbas, “Defence for Distributed Denial of Service Attacks in Cloud Computing,” in *The International Conference on Advanced Wireless, Information, and Communication Technologies (AWICT 2015)*, 2015, vol. 73, no. (AWICT 2015), pp. 490–497.

[12] N. Gruschka and M. Jensen, “Attack surfaces: A taxonomy for attacks on cloud services,” *Proc. - 2010 IEEE 3rd Int. Conf. Cloud Comput. CLOUD 2010*, pp. 276–279, 2010.

[13] S. M. Specht and R. B. Lee, “Distributed Denial of Service: Taxonomies of Attacks, Tools and Countermeasures,” in *17th International Conference on Parallel and Distributed Computing Systems*, 2004, no. September, pp. 543–550.

[14] A. Patel, M. Taghavi, K. Bakhtiyari, and J. Celestino Júnior, “An intrusion detection and prevention system in cloud computing: A systematic review,” *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 25–41, 2013.

[15] B. B. Gupta, R. C. Joshi, and M. Misra, “Distributed Denial of Service Prevention Techniques,” *Int. J. Comput. Electr. Eng.*, vol. 2, no. 2, pp. 268–276, 2012.

[16] B. Di Martino, G. Cretella, and A. Esposito, “Classification and positioning of cloud definitions and use case scenarios for portability and interoperability,” *Proc. - 2015 Int. Conf. Futur. Internet Things Cloud, FiCloud 2015 2015 Int. Conf. Open Big Data, OBD 2015*, pp. 538–544, 2015.

[17] L. Wenjuan, P. Lingdi, and X. Pan, “Use trust management module to achieve effective security mechanisms in cloud environment,” in *Electronics and Information Engineering (ICEIE), 2010 International Conference*, 2010, pp. V1– 14–V1–19.

[18] S. XIaodong, C. Guiran, and L. Fengyun, “A Trust Management Model to Enhance Security of Cloud Computing Environments,” in *Second International Conference on Networking and Distributed Computing (ICNDC)*, 2011, pp. 244–248.

[19] P. Mell and T. Grance, “The NIST definition of cloud computing,” *NIST Spec. Publ.*, vol. 145, p. 7, 2011.

[20] N. Jeyanthi, N. C. S. N. Iyengar, P. C. M. Kumar, and A. Kannammal, “An enhanced

entropy approach to detect and prevent DDOS in cloud environment," *Int. J. Commun. Networks Inf. Secur.*, vol. 5, no. 2, pp. 110–119, 2013.

[21] I. Sattar and M. Shahid, "A Review of Techniques to Detect and Prevent Distributed Denial of Service (DDoS) Attack in Cloud Computing Environment," vol. 115, no. 8, pp. 23–27, 2015.

[22] R. Aishwarya and S. Malliga, "Intrusion Detection System- An Efficient way to Thwart against Dos / DDos Attack in the Cloud Environment," *Ieee*, 2014.

[23] W. Alosaimi and K. Al-Begain, "An enhanced economical denial of sustainability mitigation system for the cloud," in *International Conference on Next Generation Mobile Applications, Services, and Technologies*, 2013, pp. 19–25.

[24] M. V. and J. V. Kouril, Daniel, Tomas Rebok, Tomáš Jirsik, Jakub Čegan, Martin Drasar, "Cloud-based Testbed for Simulation of Cyber Attacks," *Proc. 2014 IEEE Netw. Oper. Manag. Symp.*, 2014.

[25] 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, 2014.

[26] L. Garber, "Denial-of-service attacks rip the internet," *Computer (Long. Beach. Calif).*, vol. 33, no. 4, pp. 12–17, 2000.

[27] 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," *Cluster Comput.*, vol. 17, no. 3, pp. 691–702, 2014.

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

[29] E. Anitha and S. Malliga, "A packet marking approach to protect cloud environment against DDoS attacks," *2013 Int. Conf. Inf. Commun. Embed. Syst. ICICES 2013*, pp. 367–370, 2013.

[30] J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 2, pp. 39–53, 2004.

[31] A. R. Kumar, P. Selvakumar, and S. Selvakumar, "Distributed denial-of-service (DDoS) threat in collaborative environment - A survey on DDoS attack tools and traceback mechanisms," *2009 IEEE Int. Adv. Comput. Conf. IACC 2009*, no. March, pp. 1275–1280, 2009.

[32] 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)*, no. Lisa, pp. 329–339, 2000.

[33] K. Scarfone and P. Mell, "Guide to Intrusion Detection and Prevention Systems (IDPS) Recommendations of the National Institute of Standards and Technology," *Nist Spec. Publ.*, vol. 800–94, p. 127, 2007.

[34] A. Bakshi and B. Yogesh, "Securing cloud from DDOS attacks using intrusion detection system in virtual machine," *2nd Int. Conf. Commun. Softw. Networks, ICCSN 2010*, pp. 260–264, 2010.

[35] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering," *Engineering*, vol. 2, p. 1051, 2007.

[36] C. Wohlin, *Experimentation in Sofware Engineering*, vol. 53, no. 9. 2013.

[37] M. Petticrew and H. Roberts, "Systematic Reviews in the Social Sciences," vol. 42, no. 5, p. 336, 2008.

[38] N. Jeyanthi and N. C. S. N. Iyengar, "Packet resonance strategy: A spoof attack detection and prevention mechanism in cloud computing environment," *Int. J. Commun. Networks Inf. Secur.*, vol. 4, no. 3, pp. 163–173, 2012.

[39] Q. Yan, F. R. Yu, S. Member, 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 Commun. Surv. Tutorials*, vol. 18, no. c, pp. 2–23, 2015.

[40] T. Vissers, "DDoS defense system for web services in a cloud environment," *Futur. Gener. Comput. Syst.*, vol. 37, pp. 37–45, 2014.

[41] B. Khadka, C. Withana, A. Alsadoon, and A. Elchouemi, “Distributed Denial of Service attack on Cloud : Detection and Prevention,” vol. 4, no. September, pp. 210–215, 2015.

[42] Y. Lanjuan, “Defense of DDoS attack for cloud computing,” in *IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, 2012, pp. 626–629.

[43] W. Alosaimi, M. Zak, and K. Al-Begain, “Denial of Service Attacks Mitigation in the Cloud,” in *Proceedings - NGMAST 2015: The 9th International Conference on Next Generation Mobile Applications, Services and Technologies*, 2015, pp. 47–53.

[44] T. Karnwal, “A Comber Approach to Protect Cloud Computing against XML DDoS and HTTP DDoS attack,” in *Electrical, Electronics and Computer Science, IEEE Students' Conference*, 2012, pp. 1–5.

[45] N. Jeyanthi and P. C. Mogankumar, “A virtual firewall mechanism using army nodes to protect cloud infrastructure from DDoS attacks,” *Cybern. Inf. Technol.*, vol. 14, no. 3, pp. 71–85, 2014.

[46] C. Borean, R. Giannantonio, M. Mamei, D. Mana, A. Sassi, and F. Z. B, “Internet and Distributed Computing Systems,” vol. 9258, pp. 143–154, 2015.

[47] H. Wang, “A moving target DDoS defense mechanism,” *Comput. Commun.*, vol. 46, pp. 10–21, 2014.

[48] S. Yu, Y. Tian, S. Guo, and D. Wu, “Can We Beat DDoS Attacks in Clouds?(Supplementary Material),” *Nsp.Org.Au*, vol. 25, no. 9, pp. 1–4, 2000.

[49] F. Guenane, M. Nogueira, and G. Pujolle, “Reducing DDoS attacks impact using a hybrid cloud-based firewalling architecture,” *2014 Glob. Inf. Infrastruct. Netw. Symp. GIIS 2014*, 2014.

[50] H. Fujinoki, “Dynamic Binary User-Splits to protect cloud servers from DDoS attacks,” *2013 2nd Int. Conf. Innov. Comput. Cloud Comput. ICCC 2013*, pp. 125–130, 2013.

[51] N. Jeyanthi, H. Shabeeb, M. A. S. Durai, and R. Thandeeswaran, “Reputation Based Service for Cloud User Environment,” *Int. J. Eng.*, vol. 27, no. 8, pp. 1179–1184, 2014.

[52] R. Anandhi and V. N. Raj, “Prevention Of DDoS Attacks On Distributed Cloud Servers By Port Lock Mechanism,” *ARPN J. Eng. Appl. Sci.*, vol. 11, no. 5, pp. 3013–3019, 2016.

[53] M. Malekzadeh, A. A. A. Ghani, and S. Subramaniam, “A new security model to prevent denial-of-service attacks and violation of availability in wireless networks,” *Int. J. Commun. Syst.*, vol. 25, no. 7, pp. 903–925, Jul. 2012.

[54] F. Guenane, M. Nogueira, and A. Serhrouchni, “DDOS Mitigation Cloud-Based Service,” *2015 IEEE Trust.*, pp. 1363–1368, 2015.

[55] S. S. Chopade, K. U. Pandey, and D. S. Bhade, “Securing cloud servers against flooding based DDOS attacks,” *Proc. - 2013 Int. Conf. Commun. Syst. Netw. Technol. CSNT 2013*, pp. 524–528, 2013.

[56] W. Dou, Q. Chen, and J. Chen, “A confidence-based filtering method for DDoS attack defense in cloud environment,” *Futur. Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1838–1850, 2013.

[57] Q. Jia, H. Wang, D. Fleck, F. Li, A. Stavrou, and W. Powell, “Catch me if you can: A cloud-enabled ddos defense,” in *Proceedings of the International Conference on Dependable Systems and Networks*, 2014, pp. 264–275.

[58] J. B. D. Joshi, H. Takabi, and S. T. Zargar, “DCDIDP: A distributed, collaborative, and data-driven intrusion detection and prevention framework for cloud computing environments,” in *7th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, 2011, pp. 332–341.

[59] B. Joshi, “Securing cloud computing environment against DDoS attacks,” in *2012 International Conference on Computer Communication and Informatics (ICCCI)*, 2012, pp. 1–5.

[60] V. . Huang, “A DDoS Mitigation System with Multi-stage Detection and Text-Based

Turing Testing in Cloud Computing," in *27th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, 2013.

[61] J. Buchanan, Bill; Flandrin, Flavien;Macfarlane, Richard;Graves, "A Methodology To Evaluate Rate-Based Intrusion Prevention System Against Distributed Denial of Service," no. August 2016, pp. 17–22, 2010.

[62] M. N. Banu, "Cloud Computing Simulation Tools - A Study," vol. 7, no. 1, pp. 13–25, 2015.

[63] M. Sauter, "' LOIC Will Tear Us Apart ' : The Impact of Tool Design and Media Portrayals in the Success of Activist DDOS Attacks," *Am. Behav. Sci.*, vol. 57, no. 7, p. 9831007, 2013.

[64] N. Hoque, M. H. Bhuyan, R. C. Baishya, D. K. Bhattacharyya, and J. K. Kalita, "Network attacks: Taxonomy, tools and systems," *J. Netw. Comput. Appl.*, vol. 40, no. 1, pp. 307–324, 2014.

[65] M. H. Bhuyan, H. J. Kashyap, D. K. Bhattacharyya, and J. K. Kalita, "Detecting distributed denial of service attacks: Methods, tools and future directions," *Comput. J.*, vol. 57, no. 4, pp. 537–556, 2014.

[66] X. Xia, Q. Pei, Y. Liu, J. Wu, and C. Liu, "Multi-level logs based web performance evaluation and analysis," *ICCASM 2010 - 2010 Int. Conf. Comput. Appl. Syst. Model. Proc.*, vol. 4, no. Iccasm, pp. 37–41, 2010.

[67] L. Hu, T. Li, N. Xie, and J. Hu, "False Positive Elimination in Intrusion Detection Based on Clustering," pp. 519–523, 2015.

[68] J. Corsini, "Analysis and Evaluation of Network Intrusion Detection Methods To Uncover Data Theft," Napier University, 2009.

[69] J. Sommers, P. Barford, and V. Yegneswaran, "Toward Comprehensive Traffic Generation for Online IDS Evaluation," *Development*, p. 12, 2005.

[70] H. Jiang and C. Dovrolis, "Passive estimation of TCP round-trip times," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 3, pp. 75–88, 2002.

[71] E. McCrum-Gardner, "Which is the correct statistical test to use?," *Br. J. Oral Maxillofac. Surg.*, vol. 46, no. 1, pp. 38–41, 2008.

[72] M. Berndtsson, J. Hansson, B. Olsson, and B. Lundell, *Thesis Projects: A Guide for Students in Computer Science and Information Systems*. 2008.

[73] W. Lu, "Study on characteristic of emerging technology," in *International Conference on Management and Service Science, MASS 2011*, 2011.

[74] H. Oskar, "Mobile Phones and cloud computing," 2012.

[75] D. Rubin, "Research Guides: Quantitative Analysis Guide: Choose Statistical Test for 1 Dependent Variable."

APPENDIX A

Table 18: Comparison of DoS and DDoS attack generation tools

Tool	**Features**						
	Attack Type	***GUI***	***Programming Language***	***Operating System***	***Requirements***	***Performs attack type***	***Limitation***
Commview visual packet builder	DoS	yes	No	Works based on the juniper	Victim computer, attacker computer	Land attack, Teardrop attack	Cost will be more as there is a need to use ethernet cables and real systems
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

In the given tables SRC represents source mode, Dest represents destination mode.

Table 19: CPU load for TCP flood

CPU LOAD for TCP flood			
	PPS	Src (%)	Dest (%)
	0	10	10
	2000	19	20.1
	4000	30	32
	6000	57.4	58
	8000	54	55
	10000	53.2	54.9
	12000	53	54.6
	14000	53.5	53.8
	16000	53	52.8
	18000	52.8	52.5
	20000	52.1	52
	22000	51.6	51.8
	24000	49.2	50
	26000	49.6	49.7
	28000	49.1	49.2
	30000	48.4	48.6

Table 20: CPU load for UDP flood

CPU LOAD for UDP flood			
	PPS	Src (%)	Dest (%)
	0	10	10
	2000	12.1	14.2
	4000	16.5	16.8
	6000	55.2	55.4
	8000	58.1	58.9
	10000	57.6	58.2
	12000	55.1	56.7
	14000	52.3	52
	16000	52.9	52.5
	18000	53.5	53.2
	20000	58.9	58.5
	22000	59.2	59.8
	24000	62.9	63.1
	26000	63.2	64.2
	28000	66.1	66.5
	30000	68.1	68.9

Table 21: CPU Load for HTTP flood

CPU LOAD for HTTP flood			
	PPS	Src (%)	Dest (%)
	0	10	10
	2000	29.8	30.1
	4000	38.1	39.1
	6000	66.9	68.5
	8000	65.5	66.2
	10000	64.9	65.8
	12000	64.3	65.3
	14000	63.9	64.2
	16000	63.1	63.9
	18000	62.8	63.5
	20000	62.1	63.1
	22000	61.2	62.8
	24000	59.7	62.6
	26000	59.1	61.1
	28000	58.2	60.8
	30000	57.9	60.2

Table 22: CPU load for Mix Flood

CPU LOAD for Mix flood			
	PPS	Src (%)	Dest (%)
	0	10	10
	2000	21	25.1
	4000	38.1	50.8
	6000	63.3	66.5
	8000	62.2	64.9
	10000	62	63.2
	12000	61.8	63.9
	14000	60.7	62.9
	16000	60.1	62.1
	18000	59.8	61.7
	20000	59	61.1
	22000	58.2	59.9
	24000	60.3	60.5
	26000	63.2	64.3
	28000	66.9	67.2
	30000	69.9	70.2

Table 23: Memory Utilization for TCP flood

Memory utilization for TCP flood			
	Src	Dest	
Pps	Mbps	Mbps	
0	316	315	
2000	321	320	
4000	336	339	
6000	332	335	
8000	339	339	
10000	342	340	
12000	335	338	
14000	335	336	
16000	339	338	
18000	337	338	
20000	340	338	
22000	335	337	
24000	336	336	
26000	337	335	
28000	337	334	
30000	335	334	

Table 24: Memory Utilization for UDP flood

Memory utilization for UDP flood			
	Src	Dest	
Pps	Mbps	Mbps	
0	317	316	
2000	322	321	
4000	341	340	
6000	292	290	
8000	290	288	
10000	289	289	
12000	290	288	
14000	285	288	
16000	288	287	
18000	286	287	
20000	300	287	
22000	255	288	
24000	287	289	
26000	289	289	
28000	289	289	
30000	291	290	

Table 25:Memory utilization for HTTP flood

Memory utilization for HTTP flood		
	Src	Dest
Pps	Mbps	Mbps
0	313	312
2000	319	318
4000	325	322
6000	330	328
8000	325	328
10000	333	331
12000	336	335
14000	335	338
16000	341	340
18000	338	339
20000	345	342
22000	335	340
24000	338	339
26000	338	335
28000	339	339
30000	337	336

Table 26:Memory utilization for MIX flood

Memory utilization for MIX flood		
	Src (Mbps)	Dest (Mbps)
Pps	Mbps	Mbps
0	287	286
2000	289	288
4000	296	295
6000	319	300
8000	300	305
10000	320	318
12000	320	316
14000	316	315
16000	315	314
18000	314	315
20000	315	315
22000	315	316
24000	317	318
26000	318	318
28000	319	319
30000	318	320

Table 27: Bandwidth availability for TCP flood

Bandwidth availability for TCP flood		
	Src	Dest
Pps	Mbps	Mbps
0	60.3	60.3
2000	36.2	37.2
4000	19.7	20.95
6000	11.9	12.7
8000	8.89	9.89
10000	7.08	7.08
12000	4.168	4.168
14000	2.15	1.256
16000	1.16	1.076
18000	0.896	0.896
20000	0.705	0.507
22000	0.507	0.119
24000	0.129	0.102
26000	0.105	0.086
28000	0.08	0.07
30000	0.054	0.054

Table 28: Bandwidth availability for UDP flood

Bandwidth availability for UDP flood		
	Src	dest
Pps	Mbps	Mbps
0	60.3	60.3
2000	41.25	41.25
4000	21.35	21.35
6000	11.63	11.63
8000	7.32	7.32
10000	4.861	4.861
12000	4.19	4.19
14000	1.23	1.23
16000	1.06	1.06
18000	0.589	0.589
20000	0.109	0.109
22000	0	0.04
24000	0	0
26000	0	0
28000	0	0
30000	0	0

Table 29: Bandwidth availability for HTTP flood

Bandwidth availability for HTTP flood		
	Src	dest
Pps	Mbps	Mbps
0	60.3	60.3
2000	38.861	40.2
4000	25.63	24.4
6000	11.5	11.5
8000	9.89	8.98
10000	6.46	9.89
12000	5.39	5.28
14000	2.62	2.125
16000	1.35	1.18
18000	0.795	1.076
20000	0.599	0.557
22000	0.365	0.1
24000	0.1	0.076
26000	0.076	0.066
28000	0.052	0.0369
30000	0.013	0.0296

Table 30: Bandwidth availability for MIX flood

Bandwidth availability for MIX flood		
	Src	dest
Pps	Mbps	Mbps
0	60.3	60.3
2000	34.9	35.5
4000	24.21	22.53
6000	14.9	13.5
8000	10.6	6.23
10000	5.62	8.61
12000	4.92	4.35
14000	2.98	2.31
16000	1.22	1.62
18000	0.44	0.49
20000	0.42	0.47
22000	0.39	0.106
24000	0.106	0.079
26000	0.059	0.053
28000	0.079	0.01
30000	0.01	0

Table 31: Latency availability for TCP flood

Latency availability for TCP flood		
	Src	dest
Pps	Milli seconds (ms)	Milli seconds (ms)
0	1.876	1.876
2000	1.994	2.125
4000	2.215	2.993
6000	2.425	3.632
8000	37.765	40.765
10000	32.453	35.765
12000	32.899	33.899
14000	30.993	31.993
16000	29.899	30.889
18000	30.523	31.523
20000	31.626	32.626
22000	31.898	32.898
24000	32.628	33.628
26000	32.958	33.958
28000	33.485	34.485
30000	33.412	34.995

Table 32: Latency availability for UDP flood

Latency availability for UDP flood		
	Src	dest
Pps	Milli seconds (ms)	Milli seconds (ms)
0	1.876	1.876
2000	2.113	2.659
4000	2.659	3.123
6000	3.123	3.985
8000	38.657	39.657
10000	38.898	40.895
12000	38.834	39.834
14000	35.357	36.357
16000	34.924	35.924
18000	35.254	36.245
20000	36.215	36.954
22000	36.751	37.751
24000	36.148	37.148
26000	35.851	36.851
28000	35.519	36.519
30000	36.112	36.425

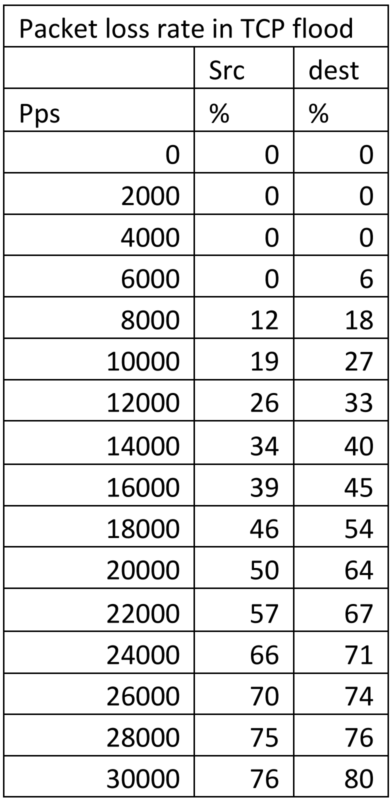
Table 33: Latency availability for HTTP flood

Latency availability for HTTP flood		
	Src	dest
Pps	Milli seconds (ms)	Milli seconds (ms)
0	1.876	1.876
2000	1.958	1.958
4000	2.324	2.324
6000	3.245	2.955
8000	36.324	37.324
10000	37.965	38.965
12000	38.335	39.335
14000	38.992	39.992
16000	37.125	38.125
18000	36.895	37.895
20000	36.325	37.325
22000	36.795	37.759
24000	36.982	37.982
26000	36.743	37.743
28000	36.143	37.143
30000	35.124	36.941

Table 34: Latency availability for MIX flood

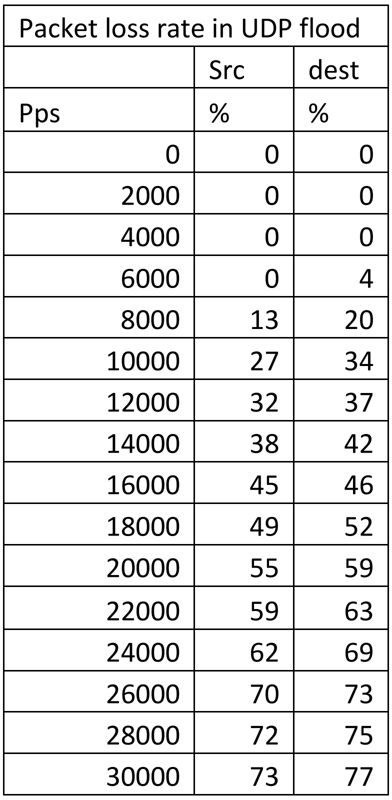
Latency availability for MIX flood		
	Src	dest
Pps	Milli seconds (ms)	Milli seconds (ms)
0	1.876	1.876
2000	2.548	3.548
4000	2.245	4.245
6000	1.127	2.127
8000	31.548	32.548
10000	33.731	34.731
12000	34.234	35.234
14000	34.992	36.129
16000	34.212	35.421
18000	34.324	35.324
20000	33.752	34.752
22000	34.987	35.987
24000	35.578	36.578
26000	36.245	37.245
28000	37.147	38.147
30000	39.125	39.357

Table 35:Packet loss rate in TCP flood



Packet loss rate in TCP flood		
	Src	dest
Pps	%	%
0	0	0
2000	0	0
4000	0	0
6000	0	6
8000	12	18
10000	19	27
12000	26	33
14000	34	40
16000	39	45
18000	46	54
20000	50	64
22000	57	67
24000	66	71
26000	70	74
28000	75	76
30000	76	80

Table 36:Packet loss rate in UDP flood



Packet loss rate in UDP flood		
	Src	dest
Pps	%	%
0	0	0
2000	0	0
4000	0	0
6000	0	4
8000	13	20
10000	27	34
12000	32	37
14000	38	42
16000	45	46
18000	49	52
20000	55	59
22000	59	63
24000	62	69
26000	70	73
28000	72	75
30000	73	77

Table 37:Packet loss rate in HTTP flood

Packet loss rate in HTTP flood		
	Src	dest
Pps	%	%
0	0	0
2000	0	0
4000	0	0
6000	1	3
8000	13	19
10000	21	30
12000	30	36
14000	36	43
16000	47	52
18000	51	56
20000	56	59
22000	58	61
24000	61	65
26000	64	69
28000	69	73
30000	70	75

Table 38:Packet loss rate in MIX flood

Packet loss rate in MIX flood		
	Src	dest
Pps	%	%
0	0	0
2000	0	0
4000	0	0
6000	0	1
8000	14	20
10000	27	32
12000	31	36
14000	36	45
16000	46	48
18000	50	55
20000	54	58
22000	55	60
24000	60	63
26000	63	67
28000	66	71
30000	67	72