Sudan University of Science and Technology

College of Graduate Studies

An Efficient Framework to Prevent Distributed Denial of Service Attack

إطار فعّال للحماية من هجوم منع الخدمة الموزع

Partial Research for Master Degree in Computer Science

(Information Security Track)

Submitted by:

Shiren Yousif Ahmed Bashir

Supervised by:

Dr. Faisal Mohammed Abdullah Ali

January 2019

Sudan University of Science & Technology

College of Graduate Studies

بسم الله الرحمن الرحيم

جامعة السودان للعلوم والتكنولوجيا

كلية الدراسات العليا

كلية الدراسات العليا

Approval Page

(To be completed after the college council approval)

Name of Candidate: SHIREN | YOUSIF | AHMED | BASHIR

Thesis title: ~~A novel Approach~~ An efficient framework to prevent Distributed Denial of Service (DDOS) Attack

Degree Examined for: M.Sc in Computer Science

Approved by:

1. External Examiner

Name: Dr. Khalid Ahmed Ibrahim

Signature: Khalid Date: 5/5/2019

2. Internal Examiner

Name: Dr. ABUAGLA BABIKER MOHAMMED

Signature: Date: 5/5/2019

3. Supervisor

Name: Dr. Faisal Mohammed Abdalla Ali

Signature: Faisal Date: 5/5/2019

الآية

قال تعالى:

(لَّيْسَ الْبِرَّ أَن تُوَلُّوا وُجُوهَكُمْ قِبَلَ الْمَشْرِقِ وَالْمَغْرِبِ وَلَٰكِنَّ الْبِرَّ مَنْ آمَنَ بِاللَّهِ وَالْيَوْمِ الْآخِرِ وَالْمَلَائِكَةِ وَالْكِتَابِ وَالنَّبِيِّينَ وَآتَى الْمَالَ عَلَىٰ حُبِّهِ ذَوِي الْقُرْبَىٰ وَالْيَتَامَىٰ وَالْمَسَاكِينَ وَابْنَ السَّبِيلِ وَالسَّائِلِينَ وَفِي الرِّقَابِ وَأَقَامَ الصَّلَاةَ وَآتَى الزَّكَاةَ وَالْمُوفُونَ بِعَهْدِهِمْ إِذَا عَاهَدُوا ۖ وَالصَّابِرِينَ فِي الْبَأْسَاءِ وَالضَّرَّاءِ وَحِينَ الْبَأْسِ ۗ أُولَٰئِكَ الَّذِينَ صَدَقُوا ۖ وَأُولَٰئِكَ هُمُ الْمُتَّقُونَ)

صدق الله العظيم

سوره البقره الآية ﴿177﴾

DEDICATION

This dissertation is dedicated to:

My Sweetheart, my dear father,

My Soul, My lovely mother,

My sisters and brothers,

My beloved friends: Islam and Noon,

My friends and colleagues,

Ask God to bless them and prolong their life

ACKNOWLEGMENTS

In the name of Allah, the Most Gracious and the Most Merciful Alhamdulillah, all praises to Allah for the strengths and His blessing in completing this thesis and I Would like to express my gratitude and thankfulness to my Supervisor **Dr. Faisal Mohammed Abdalla** his continued efforts during all phases of the research Ask God to make this work in the balance of his good deeds, I am very appreciative to thank all who provide a helping hand to achieve this study specifically my teachers at the college of computer sciences and information technology Sudan University of sciences and technology. Also I want thankfulness for everybody supports me to complete this research.

ABSTRACT

Internet and web services have become an inseparable part of our lives. Hence, ensuring continuous availability of service has become imperative to the success of any organization. But these services are often hampered by constant threats from myriad types of attacks. One such attack is called Distributed Denial of Service (DDoS) attack that results in issues ranging from temporary slowdown of servers to complete non-availability of service. The complexity of DDoS attack makes their detection and mitigation difficult.

In this research, an effective protection framework based on FNM open-source tool and iptables was proposed, FNM is use to detect DDoS-based flood attack (SYN, UDP, and ICMP) by adjusting the abnormal rate of packet data sent (threshold), FNM discovered the attack and notified the administrator of the system via e-mail and produced a report containing detailed information about the attack, it was noted that there are data packets issued by the server responding to the attack in the variable outgoing pps, which means consumption of server resources resulting in the denial of service, after that was used packet filtering in Linux kernel by used iptables script to filter attack traffic and drop, then was tested re-attack and compared to the variable

value of outgoing pps, which became zero which means there is no data packets issued by the server, the experimental result shows that when using the tools FNM and iptables it has more security and enhances safety in detecting and minimizing attack-blocking service.

المستخلص

لقد أصبحت خدمات الويب والإنترنت جزءًا لا يتجزأ من حياتنا ومن ثم ، أصبح ضمان استمرار توفر الخدمة أمراً حتمياً لنجاح أي مؤسسه. لكن هذه الخدمات غالباً ما تعيقها التهديدات المستمرة من أنواع الهجمات التي لا تعد ولا تحصى . واحد من تلك الهجمات يسمى هجوم منع الخدمة الموزع مما ينتج عنها بُطئ في تقديم هذه الخدمات وحتي حجبها من المستفيد. بسبب تعقيد هذا الهجوم مما يجعل إمكانيه إكتشافه والتقليل منه أمر في غايه الصعوبه.

في هذا البحث، تم اقتراح إطار فعّال للحماية من هجوم منع الخدمة الموزع مبني على إستخدام أداه الفاست نت مون لإكتشاف الهجوم عن طريق ضبط المعدل الغير طبيعي لحزم البيانات المرسله(حد) و ضبط الحصول على حزم البيانات عن طريق نواه اللينكس، عند إختبار هجوم منع الخدمة الموزع فإن الفاست نت مون إكتشف الهجوم و أخطر مدير النظام عبر بريده الإلكتروني وأنتج تقرير يحتوي على معلومات تفصيليه عن الهجوم، لُوحظ في التقرير أن هنالك حزم بيانات صادره من المخدم إستجابه للهجوم والمتمثله في قيمه متغير حزم البيانات الصادره من المخدم مما يعني إستهلاك موارده وبالتالي ينتج عنه منع الخدمة، بعد ذلك أُستخدم تقنيه تنقيه حزم البيانات الموجوده في نواه للينكس عن طريق إستخدام اوامره لتنقيه حركه مرور الهجوم ومسحه ثم بعد ذلك تم إعاده إختبار هجوم منع الخدمة الموزع ومقارنه قيمه متغير حزم البيانات الصادره من المخدم والتي أصبحت صفر مما يعني الحفاظ على موارد المخدم، تظهر النتيجه التجريبيه إنه عند إستخدام أداتي الفاست نت مون و تقنيه تنقيه حزم البيانات الموجوده في نواه للينكس يعطي نتيجه أكثر سريه ويعزز الأمان في إكتشاف والتقليل من هجوم منع الخدمة الموزع.

Table of Content

LIST OF FIGURES

LIST OF ABBREVIATIONS

BPS	Bit Per Second
CIA	Confidentiality, Integrity, Availability
DOS	Denial of Service
DDOS	Distributed Denial of Service
DPI	Deep Packet Inspection
FNM	FastNetMon
ICMP	Internet Control Message Protocol
MF2	More fragmentation 2
NMAP	Network mapping
OSI	Open Systems Interconnections
PPS	Packet Per Second
PCAP	Packet capture
RFC	Request For Comments
TCP	Transmission Control Protocol
TCP-SYN	Synchronize/ start
SYN-ACK	Synchronize/acknowledge

UDP	User Datagram Protocol
VM	Virtual Machine
PCE	Packet Capture Engine

CHAPTER: I

INTRODUCTION

1.1 Introduction

Contemporary society has grown increasingly reliant on information and the systems used to store, process, and communicate that information. Consequently very few aspects of modern-day life would continue to operate smoothly in the absence of functioning information and communications systems. This increasing societal dependence on information and communications technologies in general and communications networks in particular is most obvious when the delivery of services via these systems and networks is disrupted even for relatively short periods. Such situations in which access to networked services by legitimate customers or clients is deliberately disrupted are collectively categorized as 'Denial of Service' or DoS attack [1].

1.2 Denial of Service (DoS)

Denial or degradation of service may result from malicious or benign actions. These actions may originate locally or remotely from the service, or user, experiencing denial or degradation of service. The communications bandwidth, memory buffers, computational resources, or the network protocol or application processing logic of the victim, or any systems on which the victim depends for delivering service may be targeted. The ultimate goal of a DoS attacks is to compromise the availability of services [1].

1.2.1 Distributed Denial of Service (DDoS)

DDoS attack are a variant of the more generic DoS attack. The key feature of a DDoS attack are the large number of hosts used to launch such an attack. It is common to see up to hundreds of thousands (if not millions) of hosts being used to launch a DDoS attack [1]. A Distributed Denial-of-Service (DDoS) attack are carried out by simultaneously by compromised systems against targets causing system and service unavailability. Regardless of industry and size, companies worldwide are increasingly becoming target of DDoS attack. The sophistication and intensity of these attack are exponentially rising due to increase in number of compromised systems, unpatched vulnerabilities and increased business impact [2].

The complexity of DDoS attack make detection and mitigation difficult. Moreover, it also increases the overall operational costs to deploy mitigation solutions and it is not cost effective to deploy at the edge of victim networks. Quite a lot of research has been done to classify DDoS attack and suggesting techniques to detect and mitigate them. Also, there are several open source based intrusion and DDoS detection software's available online. Open source systems have increased considerable inclination because of their adaptability, support and cost-effectiveness [5].

1.3 Problem Statement

Today, internet and web services have become an inseparable part of our lives. Hence, ensuring continuous availability of service has become imperative to the success of any organization. But these services are often hampered by constant threats from myriad types of attacks.

One such attack is called Distributed Denial of Service (DDOS) attack that results in issues ranging from temporary slowdown of servers to complete non-availability of service .

1.4 Research Objectives

a) To ensuring continuous availability of internet services.
b) To production communications bandwidth, memory buffers, computational resources from DDoS attack.
c) To prevention server from DDoS attack by using open source DDoS detector tool (FNM) beside use iptables to migitaion attack.

1.5 Important Research

Help Organizations to Ensuring **continuous availability of services** and production communications bandwidth, memory buffers, computational resources from DDoS attack.

1.6 Research scope

The aim of this research how FNM open source tool can detection and mitigation DDoS flooding attack in layer three and four using anomaly detection and using iptables netfilters.

1.7 Research Methodology

This research proposes an efficient framework for detection and mitigation DDoS based flooding attacks by use FNM open source tool for very high-performance DDoS detector and use packet filtering technique in the Linux kernel to mitigate it. The experimental setup Figure (1.1) is made by use VMware workstation to create VM victim and VM attacker, use Ubuntu operating system in Web server VM victim for test FNM to capture

packet, and detection in Linux kernel also test a packet filtering in the Linux kernel to mitigate DDoS attack.

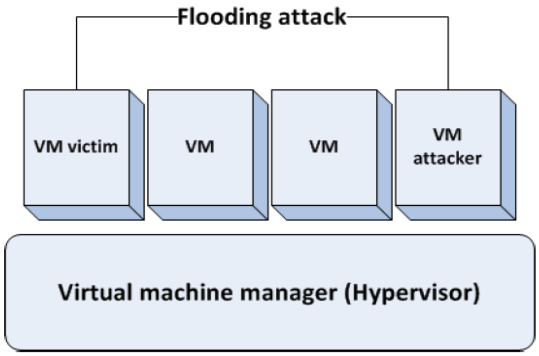


Figure (1.1) Experimental setup

Propose methodology in the Figure (1.2) show how work, in the first level detection DDoS attack of incoming traffic by use FNM within web server to show result what is incoming traffic is attacking or not, If not, forward traffic to local process of the web server, but if is attack FNM generate report attack file contains detail information about attack and notify system administrator by email then come second level mitigation web server by use packet filtering technique in Linux kernel and use iptables script to drop attack traffic.

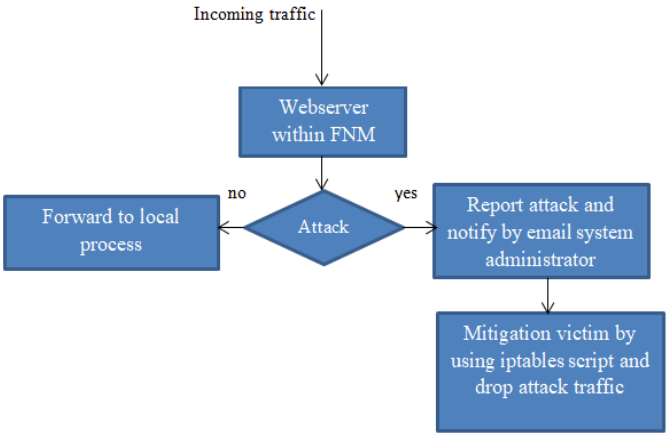


Figure (1.2) Flow chart of detection and mitigation DDoS attack

1.7 Research Organization

Chapter one gives introduction about the resaerch, defining the problem, objectives, methodology and scope. Chapter two represents literature review. Chapter three contains methodology. The chapter four contains implementation and result. The last is conclusion and future work.

CHAPTER: II

LITERATURE REVIEW

2.1 Introduction

The requirements of information security within an organization have undergone two major changes in the last several decades. Before the widespread use of data processing equipment, the security of information felt to be valuable to an organization was provided primarily by physical and administrative means. When we talk about the computer, the need for automated tools for protecting files and other information stored on the computer became evident. This is especially the case for a shared system, such as a time-sharing system, and the need is even more acute for systems that can be accessed over a public telephone network, data network, or the Internet. The generic name for the collection of tools designed to protect data and to thwart hackers is **computer security**. The second major change that affected security is the introduction of distributed systems and the use of networks and communications facilities for carrying data between terminal user and computer and between computer and computer. **Network security** measures are needed to protect data during their transmission. In fact, the term network security is somewhat misleading, because virtually all business, government, and academic organizations interconnect their data processing equipment with a collection of interconnected networks. Such a collection is often referred to as an internet, i and the term **internet security** is used [3].

2.2 Computer Security Concepts

The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability, and confidentiality of information system resources (includes hardware, software, firmware, information/data, and telecommunications).

Key objectives that are at the heart of computer security:-

a) Confidentiality: This term covers two related concepts:

-Data confidentiality: Assures that private or confidential information is not made available or disclosed to unauthorized individuals.

-Privacy: Assures that individuals control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed.

b) Integrity: This term covers two related concepts:

-Data integrity: Assures that information and programs are changed only in a specified and authorized manner.

-System integrity: Assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system.

c) Availability: Assures that systems work promptly and service is not denied to authorize users [3].

These three concepts form what is often referred to as the CIA triad Figure (2.1)

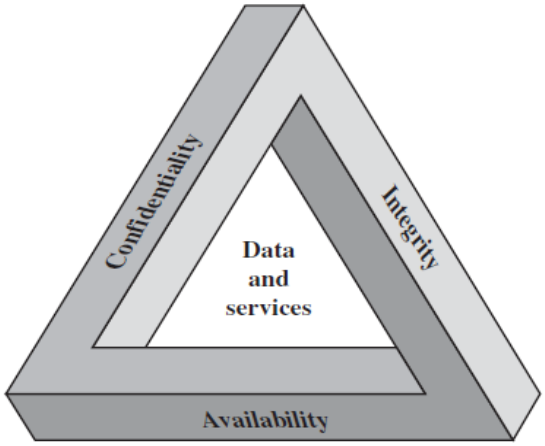


Figure (2.1) the security requirements triad

FIPS 199 provides a useful characterization of these three objectives in terms of requirements and the definition of a loss of security in each category.

a. Confidentiality: Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information. A loss of confidentiality is the unauthorized disclosure of information [3].
b. Integrity: Guarding against improper information modification or destruction, including ensuring information non repudiation and authenticity. A loss of integrity is the unauthorized modification or destruction of information.
c. Availability: Ensuring timely and reliable access to and use of information. A loss of availability is the disruption of access to or use of information or an information system.

Although the use of the CIA triad to define security objectives is well established, some in the security field feel that additional concepts are needed to present a complete picture. Two of the most commonly mentioned are

d. Authenticity: The property of being genuine and being able to be verified and trusted; confidence in the validity of a transmission, a message, or message originator. This means verifying that users are who they say they are and that each input arriving at the system came from a trusted source.

e. Accountability: The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity. This supports non repudiation, deterrence, fault isolation, intrusion detection and prevention, and after-action recovery and legal action. Because truly secure systems are not yet an achievable le goal, we must be able to trace a security breach to a responsible party. Systems must keep records of their activities to permit later forensic analysis to trace security breaches or to aid in transaction disputes.

2.3 The OSI Security Architecture

Threat: A potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm.

Attack: An assault on system security that derives from an intelligent threat. That is, an intelligent act that is a deliberate attempt (especially in the sense of a method or technique) to evade security services and violate the security policy of a system.

The OSI security architecture focuses on security attacks, mechanisms, and services. These can be defined briefly as:

• Security attack: Any action that compromises the security of information owned by an organization.

• Security mechanism: A process (or a device incorporating such a process) that is designed to detect, prevent, or recover from a security attack.

• Security service: A processing or communication service that enhances the security of the data processing systems and the information transfers of an organization. The services are intended to counter security attacks, and they make use of one or more security mechanisms to provide the service [3].

2.3.1 Security Attacks

A useful means of classifying security attacks, used both in X.800 and RFC 2828, is in terms of passive attacks and active attacks. A passive attack attempts to learn or make use of information from the system but does not affect system resources. An active attack attempts to alter system resources or affect their operation.

Passive Attacks: Passive attacks are in the nature of eavesdropping on, or monitoring of, transmissions. The goal of the opponent is to obtain information that is being transmitted. Two types of passive attacks are the release of message contents and traffic analysis.

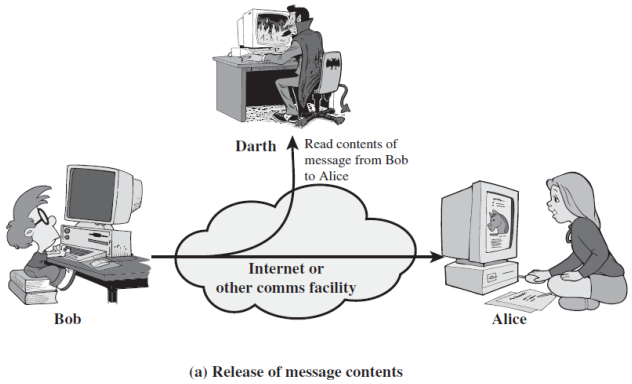


Figure (2.2) Release of message contents

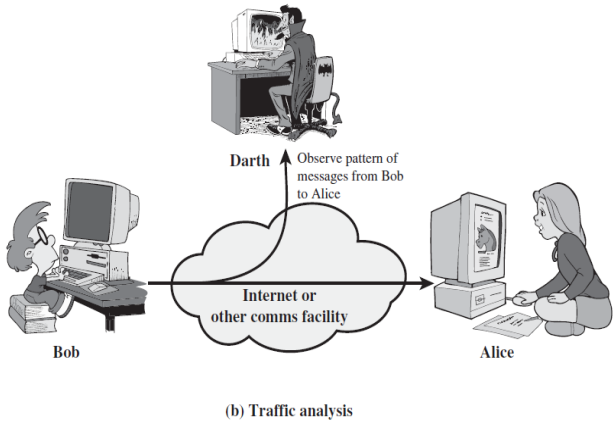


Figure (2.3) Traffic analysis

The release of message contents is easily understood Figure (2.2). A telephone conversation, an electronic mail message, and a transferred file may contain sensitive or confidential information. We would like to prevent an opponent from learning the contents of these transmissions.

Traffic analysis, is subtler Figure (2.3).Suppose that we had a way of masking the contents of messages or other information traffic so that opponents, even if they captured the message, could not extract the information from the message. The common technique for masking contents is encryption. If we had encryption protection in place, an opponent still might be able to observe the pattern of these messages. The opponent could determine the location and identity of communicating hosts and could observe the frequency and length of messages being exchanged.

This information might be useful in guessing the nature of the communication that was taking place. Passive attacks are very difficult to detect, because they do not involve any alteration of the data. Typically, the message traffic is sent and received in an apparently normal fashion, and neither the sender nor the receiver is aware that a third party has read the messages or observed the traffic pattern. However, it is feasible to prevent the success of these attacks, usually by means of encryption. Thus, the

emphasis in dealing with passive attacks is on prevention rather than detection.

Active Attacks: Active attacks involve some modification of the data stream or the creation of a false stream and can be subdivided into four categories: masquerade, replay, modification of messages, and denial of service [3].

A masquerade takes place when one entity pretends to be a different entity Figure (2.4). A masquerade attack usually includes one of the other forms of active attack. For example, authentication sequences can be captured and replayed after a valid authentication sequence has taken place, thus enabling an authorized entity with few privileges to obtain extra privileges by impersonating an entity that has those privileges [3].

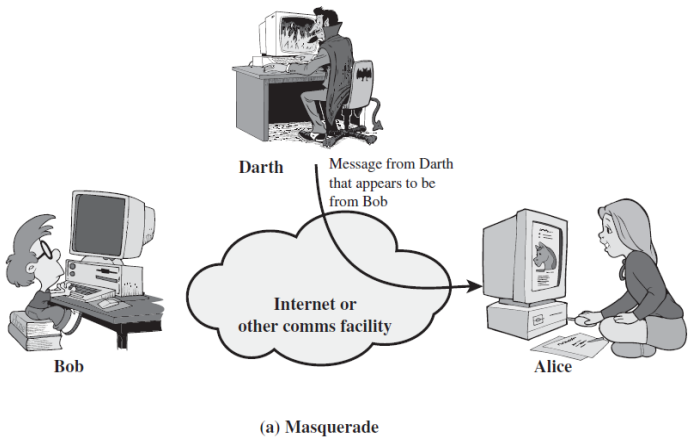


Figure (2.4) Masquerade attack

Replay involves the passive capture of a data unit and its subsequent Retransmission to produce an unauthorized effect Figure (2.5).

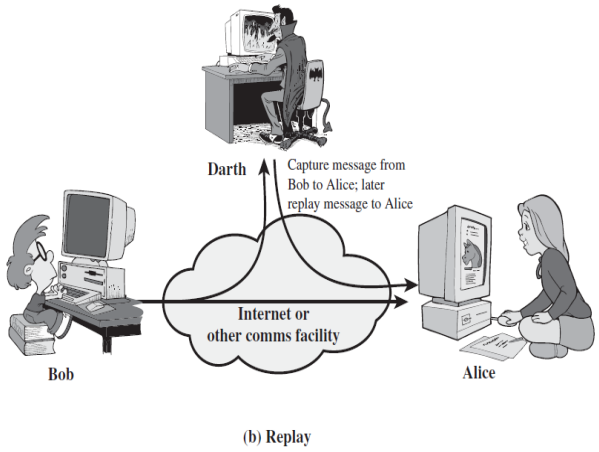


Figure (2.5) Replay attack

Modification of messages simply means that some portion of a legitimate message is altered, or that messages are delayed or reordered, to produce an unauthorized effect Figure (2.6). For example, a message meaning "Allow John Smith to read confidential file accounts" is modified to mean "Allow Fred Brown to read confidential file accounts."

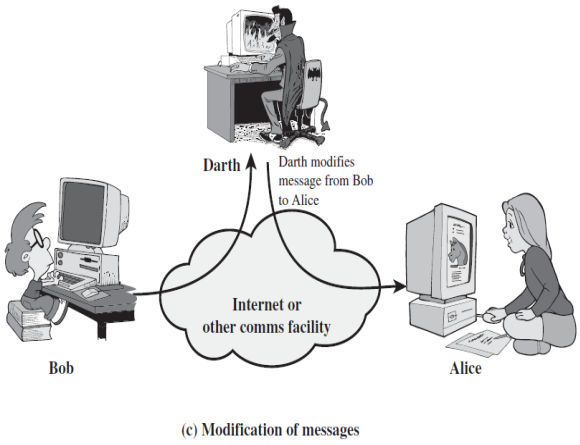


Figure (2.6) Modification of messages attack

The denial of service prevents or inhibits the normal use or management of communications facilities Figure (2.7). This attack may have a specific target; for example, an entity may suppress all messages directed to a particular destination (e.g., the security audit service). Another

form of service denial is the disruption of an entire network—either by disabling the network or by overloading it with messages so as to degrade performance [3].

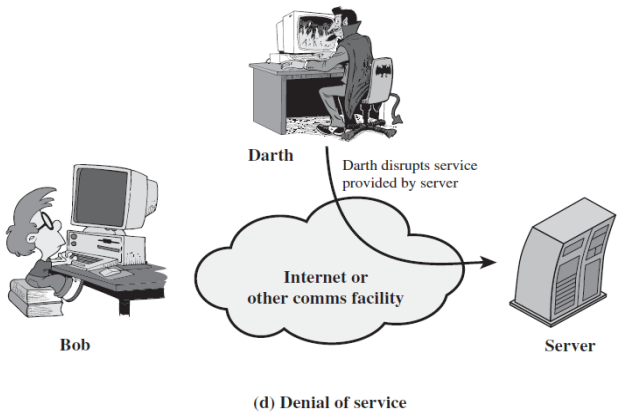


Figure (2.7) Denial of service attack

Active attacks present the opposite characteristics of passive attacks. Whereas passive attacks are difficult to detect, measures are available to prevent their success. On the other hand, it is quite difficult to prevent active attacks absolutely because of the wide variety of potential physical, software, and network vulnerabilities. Instead, the goal is to detect active attacks and to recover from any disruption or delays caused by them. If the detection has a deterrent effect, it also may contribute to prevention [3].

2.4 Denial of Service Attack

The Internet has become an important part of our society in numerous ways, such as in economics, government, business, and daily personal life. Further, an increasing amount of critical infrastructures, e.g., power grid, air traffic control, are managed and controlled via the Internet, in addition to traditional infrastructure for communication. However, today’s cyberspace

is full of attacks, such as Distributed Denial of Service (DDoS), information phishing, financial fraud, email spamming, and so on. We can see that cyberspace has become a heaven for intelligent criminals, who are motivated by significant financial or political reward. According to an annual report from the FBI's Internet Crime Complaint Centre, financial loss resulting from cyber-attack totaled US$559.7 million in 2009. Symantec identified more than 5.5 billion malicious attacks in 2011, an increase of 81% over the previous year.

Moreover, the number of unique malware variants increased to 403 million, and the number of web attacks per day increased by 36 %.Among various Internet based attacks, Denial of Service (DoS) attack is a critical and continuous threat in cyber security. In general, DoS attacks are implemented by either forcing a victim computer to reset, or consuming its resources, e.g., CPU cycles, memory or network bandwidth. As a result, the targeted computer can no longer provide its intended services to its legitimate users. When the DoS attacks are organized by multiple distributed computers, it is called distributed denial of service attack, which is a popular attack method in the cyberspace. From classical textbooks, we know security falls into three categories: confidentiality, availability and integrity. It is obvious that DDoS attacks belong to the availability category [4].

2.5 How to Launch DDoS Attacks

In general, DDoS attacks can be launched in two forms. The first one targets to crash a system by sending one or more carefully crafted packets, which are designed based the vulnerability of the victim. For example, the "ping-of death" attacks, which can cause some operating systems to crash,

freeze, or reboot. This form of DDoS can be defeated by patching the system vulnerabilities. The second form DDoS is to use a large amount of traffic to exhaust the resources of a victim, such as network bandwidth, computing power, operating system data structures, and so on. As a result, the quality of service of the victim is significantly degraded or disabled to its legitimate clients. Compared with the first form, the second form of DDoS attack is hard to deal with. In order to launch an effective DDoS attack, cyber attackers have to firstly establish a network of computers, which is known as a botnet or army. We call the people who control a botnet as botmasters or botnet owners. In order to organize a botnet, attackers take advantage of various methods to find vulnerable hosts on the Internet to gain access to them. Attackers generally use different kinds of techniques (referred to as scanning techniques) to find vulnerable machines. The next step for the attacker is to install programs (known as attack tools) on the compromised hosts. The hosts running these attack tools are known as bots or zombies the headquarter of a botnet is call command and control (C&C) server. It is necessary for a C&C server to communicate with its bots for a number of reasons, such as updating the attack tools, and issuing an attack order. In order to sustain their C&C servers from detection, botnet programmers may set up a few intermediate nodes as step stones between the C&C server and bots. They also take cryptography techniques to encrypt the messages of their communication. Moreover, in order to avoid evictions, botnet programmers are taking various techniques, such as IP flux or domain flux, to sustain their C&C servers. Consequently, they also need to design novel strategies for their bots tophone home. There are two different DDoS attack classes: typical DDoS attack and DRDoS (Distributed Reflection Denial of Service) attack. The hosts of both categories are compromised machines that have been recruited during the scanning process and are installed with malicious code [1].

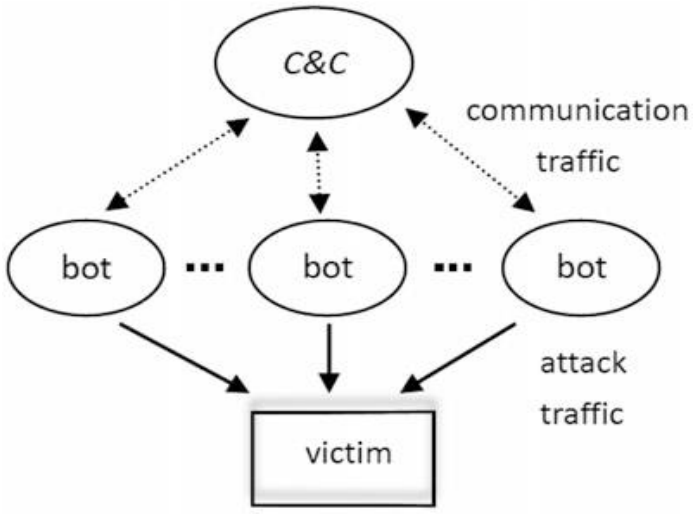


Figure 2.8 Typical distributed denial of service attack

As shown in Figure (2.8), in a typical DDoS attack, an attacker coordinates and orders the C&C server, and in turn, it coordinates and triggers bots. More specifically, the attacker sends an attack command to the C&C server who activates all attack processes on the bots, which are in hibernation, waiting for the appropriate command to wake up and start attacking. Then, C&C servers, through these processes, send attack commands to bots, ordering them to mount a DDoS attack against a victim. By doing it this way, the bots begins to send a large volume of packets to the victim, flooding its system with useless load and exhausting its resources [1].

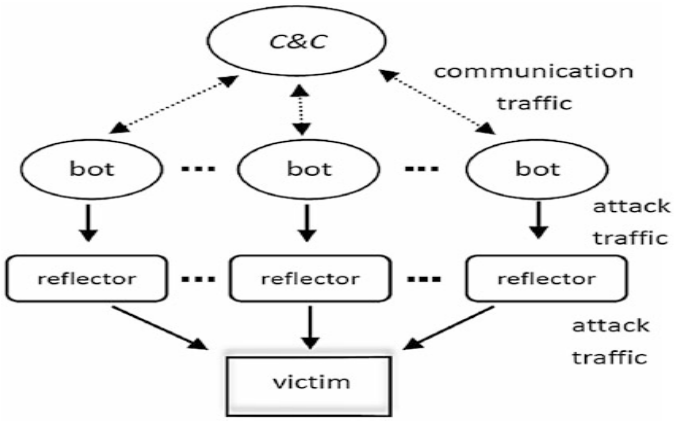


Figure (2.9) Distributed reflection denial of service attack

Unlike a typical DDoS attack, a DRDoS attack network consists of C&C servers and reflectors as shown in Figure (2.9). The scenario of this type of attack is the same as that of a typical DDoS attack up to a specific stage. The attackers have control over C&C servers, which, in turn, have control over bots. The difference with a DRDoS attack is that bots, led by C&C servers, send a stream of packets with the victim's IP address as the source IP address to other uninfected machines (known as reflectors). This exhorts these innocent machines to connect to the victim because they believe that the victim was the host that requested it. As a result, there is a large amount of traffic to the victim from the reflectors for the opening of new connections [1].

2.6 Types of DDoS attacks

Generally, DDoS attack can be classified into three main groups based on type and magnitude of traffic used: volumetric or volume based attacks, protocol attacks and application attacks. DDoS is divided in to two based on the attack target: Infrastructure layer and application layer. In this section we will discuss the infrastructure layer, which includes volumetric and protocol attacks.

2.6.1 Volumetric/ Volume based Attacks

This type of attack saturates the bandwidth of the network by sending packet storm. The magnitude of attack is measured in bits per second (bps). The attack involves bots and zombies to send a huge amount of traffic to exhaust the bandwidth capacity of the network. The effect of the attack saturates the network links and overwhelms routers, switches, firewalls and Internet Service Provider (ISP) and overall network level devices. Afterwards, the legitimate users request will be dropped from reaching to service provider end. Common attacks of this category are UDP flood, TCP flood, ICMP flood and packet flood [5].

2.6.1.1 UDP Flood Attack

The stateless, connection-less communication model, nature of UDP makes a common tool for different attacks which requires manipulating packet. UDP packet is easy to construct and generate. As it is stateless, it is easy to forge source IP so that it could be spoofed and hard to trace the right source of the sender. Therefore flooding using UDP packets become one of the most well-known and compelling methods for DoS and DDoS attack. UDP can be constructed as a very small packet, so that the attacker can easily send a high volume of small-sized UDP packets which causes forwarding issues for network level forwarding devices such as routers, firewalls, and inline traffic processing devices. The less effective UDP flood attack can cause jitter and latency in real time streaming protocols for voice and video.

Under the normal condition, a server which receives UDP request goes through two steps. First, the server checks if a requested port is open and a specific application is running to handle the requests coming through the port. Second, if there is no application is running to handle the request it

will respond with ICMP packet setting destination unreachable flag to inform the source address that a unavailability of the requested service. During UDP flood attack, the attacker uses a large flood of UDP with spoofed-IP address and saturates network resources with the request and also with the same amount of destination unreachable ICMP packets responses. As a result, the finite resource of victim network will be exhausted by the process of checking and responding for a huge volume of UDP request floods. This results in denial of service for legitimate traffic [5].

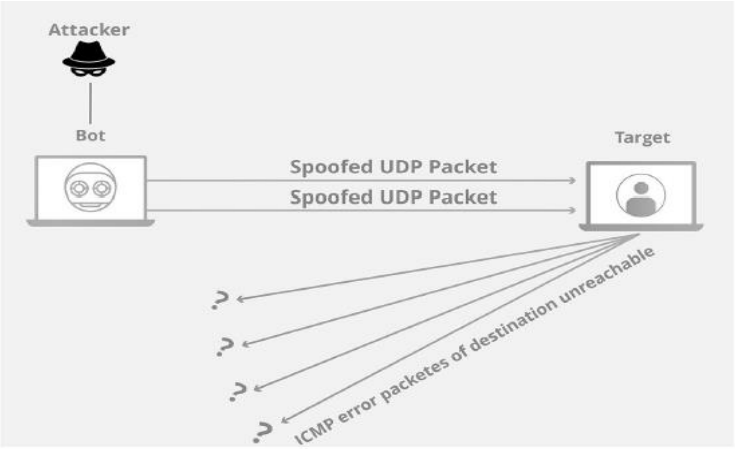


Figure (2.10) UDP flood attack

2.6.1.2 ICMP Flood Attack

ICMP Flood attacks exploit the Internet Control Message Protocol (ICMP), which enables users to send an echo packet to a remote host to check whether it's alive. More specifically during a DDoS ICMP flood attack the agents send large volumes of ICMP_ECHO_ REPLY packets ("ping") to the victim. These packets request reply from the victim and this results in saturation of the bandwidth of the victim's network connection. During an ICMP flood attack the source IP address may be spoofed [6].

Figure (2.11) ICMP flood attack

2.6.2 Protocol Attacks

Protocol attack works by exploiting a weakness in transport layer and network layer protocols of Open Systems Interconnections (OSI) models based applications and protocols in victim network. It misuses a specific feature or implementation bug of protocols used at the victim network in order to exhaust its limited resources. Magnitude of the attack is measured in packets per second (pps). This type of attack exhausts resources of server and intermediate equipment's working in layer 4 and 5 such as load-balancer and firewalls. Common and well known attacks of this type is Transmission Connection Protocol (TCP) SYN flood [5].

2.6.2.1 TCP SYN Flood Attack

TCP is connection-oriented protocol, unlike UDP. It provides flow control, reliable, ordered and error control services for an application using TCP protocol. Before sending data using TCP it must go three steps known as the TCP three-way handshake to setup a reliable connection. First, the initiator host sends TCP-SYN (synchronize/start) and then the receiver sends SYN ACK (synchronize/acknowledge) packet back, finally the initiator sends ACK. Afterwards, the data communication carries on the established reliable connection. TCP SYN flood attack uses the first step of

TCP three-way handshake stages and sends a huge amount of TCP SYN request to exhaust the victim server. In a normal operation, the server receiving TCP SYN will send back SYN ACK flag and waits for ACK or timeout to expire the connection. Like other DDoS attack, TCP SYN flood attack sends TCP SYN packets from multiple sources with spoofed IP addresses. While trying to handle every request from the attacker which is TCP SYN flood the server become busy and it fails to respond to the legitimate users' requests. Due to the limited resources of server is exhausted by the attack traffic; it creates Denial of Services condition to the legitimate users [5].

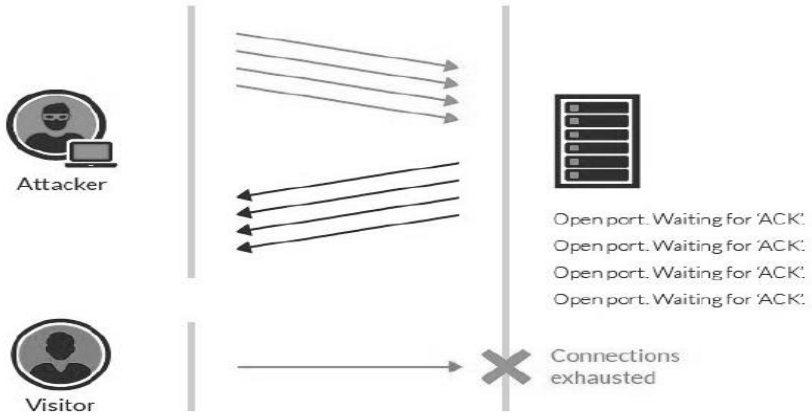


Figure (2.12) TCP SYN flood attack

2.7 DDoS Attack Detection Methods

According to [7], there are two types of network attack detection or intrusion detection methods: signature based detection or anomaly based detection.

2.7.1 Signature Detection

Signature based detection, uses a predefined sets of signatures to inspect the network traffic for the presence of attacks. The detection application employed this mechanism will compare each packet, commonly its payload, of the network traffic with a given set of patterns of DDoS attack. This method is capable of attaining high accuracy and less false positive in identifying attacks. However, it fails to identify unknown attacks which have no stored signature in the given set of patterns for an attack [5].

2.7.2 Anomaly Detection

Anomaly detection unlike signature based it identifies malicious traffic in a network by detecting anomalies network traffic pattern. The behavior of network can be analyzed in different ways, for example:

– Analyzing using packet size to check if the size is too short and violate application layer protocols.

– Rate-based detection uses a time-based profile of normal traffic volume to Detect against DDoS flooding attacks.

The advantages of the anomaly detection over signature based are that it is not limited to known attacks; it can detect previously unknown attacks based on the behavior of the attack traffic [5].

2.8 Related Works

Many of research studies discuses detection and prevention of DDOS Attacks on network layer and application layer in this research we will mention twelve related works.

Mrunali Desai et al in 2016.In this paper focuses on prevention of DDoS Attack with the help of various filters that work on application layer on the client as well as admin side web application. The filters are programmed in Java and the web services acts as an interface between client and admin. These filters can prevent attacks like flooding, phishing, buffer overflow, SYN flood attacks. These filters are designed and programmed in java using web referrals. This system aims to ensure that no authorized user is ever denied access to the web server and an illegitimate user is blocked permanently and work on application layer in web server. The limitation of this study keeps a limit on the files being uploaded even by the legitimate users and allows only a specific number of users to access the services, at a time also what happen if we use these filters on socket or transport layer I thinks is more secure or we use Secure Sockets Layer (SSL) is protocol that manages server authentication, client authentication and encrypted communication between servers, also these filters prevent SYN flood attack what about UDP and ICMP flood attacks [4].

Hrishikesh Arun Deshpande in 2015. Proposed solution in this paper to create a network of virtualized honeypots within the existing infrastructure with minimal cost and maintenance overheads. However, effective detection and deflection of attacks together with identification of attack sources is necessary. This is accomplished using honeypots. Additionally, each of these honeypots can have backup VM's that normally remain idle but can be activated the moment an existing honey VM is compromised by an attacker. This honeypot daemon, abbreviated as honey-d, works like a gateway and performs initial authentication before passing on the information to the actual server. The new solution proposes to create a virtual network or mesh of honeypot VM's and honey daemon processes to provide multiple levels of security checks and intrusion detection using behavioral analysis and challenge

response models. Also, malicious traffic is routed to honey farm there by protecting the production server and internal networks from both crashing and flooding type of DDoS attack. Honey mesh when integrated with existing security infrastructure such as firewalls, encryption, authentication services, virtual private network (VPN) etc. can protect the server network from any kind of DDoS attack. Production servers and organization's internal network (LAN) are fully protected and isn't using security mechanism to protect the organization's routers from being flooded with malicious requests [8].

Akash Naykude et al in 2016. This research paper is proposes a Traceback-based Defense against DDoS Attack (TBDA) approach, we designed one technique that is impressively filtered out the majority of DDoS attack traffic. So, our main objective or intention for this work is to improving the overall performance and quality of the appropriate traffic and also minimize the attack traffic to maintain the connection of service for better communication. There is one client or normal user, one server and many agents with one attacker. Now, this attacker is going to make a feck requests by using these agents and going to slow down and mix with normal client and server connection. But in our developed system server in more powerful to identify that which is normal client and who is attacker. After that it blocks IP address of that attacker and it's done. The main this is that attacker's request of processing is captured and does not processed, that means it is blocked without knowing them [9].

Muhammad Ahmad et al in 2017. In this paper studies the field of internet cloud computing, it has become incredible and amazing growth technology where huge amount of data and information available online. Cloud computing provide different resources on demand where the users can access these resources through the internet easily. Due to distributed nature of cloud computing technology remained untouched from the

attackers. Attackers can attack easily on cloud computing and decrease the effectiveness and resources availability to the users in the cloud. The purpose of this paper is to use Snort IPS / IDS and Wireshark to avoid DDoS attack in the cloud computing. DDoS attack try to continue busy the network by sending a huge amount of requests towards the server so that server are unable to provide and unable to respond services to the clients [10].

Karanbir Singh, et al in 2017. Most detection systems depend on some type of centralized processing to analyze the data necessary to detect an attack. In centralized defense, all modules are placed on single point. A centralized approach can be vulnerable to attack. But in distributed defense, all of the defense modules are placed at different points and do not succumb to the high volume of DDoS attack and can discover the attacks timely as well as fight the attack with more resources. These factors clearly indicate that the DDoS problem requires a distributed solution than the centralized solution. In this paper, we compare both types of defense mechanisms and identify their relative advantages and disadvantages. Later they are compared against some performance metrics to know which kind of solution is best. DDoS defense method can be put either in a centralized or distributed defense systems. These defense locations are compared against their features, advantages, and disadvantages. The comparison shows that distributed defense system are little more effective than centralized defense. But this comparison was not sufficient to prove their efficiency. We have also identified and discussed some existing defense mechanisms of each category from the literature. Later in order to prove the effectiveness of distributed defense, we compared them against some performance metrics.

The comparison clearly shows that distributed defense are better than centralized defense system. So in order to effectively control the DDoS attack, we must choose a distributed DDoS defense solution [11].

Archana .S. Pimpalkar et al in 2015. Presented a defense mechanism against source IP address spoofing that uses Hash based cryptographic technique is used for providing authentication to each packet at the client side and verifying the packet identity at the routers near the target server can efficiently identify the attack packets with fake source IP address. Attack packets are separated from normal packets and dropped before reaching the target server while normal packets are forwarded unaffected thus allowing the legitimate clients to access the server resources [12].

Soumya Suresh, Kiran V K in 2016. In this paper a software puzzle scheme with cryptographic technique is used to prevent DoS and DDoS attack. A dynamic software puzzle scheme is used so that the puzzle function used is not known in advance. Hence, malicious client cannot solve the puzzle using GPU software. The proposed system provides even more security using conventional cryptographic techniques [13].

Meklit Elfiyos Dekita in 2018. In this research study the possibilities and performance of DDoS detection and prevention on commodity hardware using open source solutions. It found commodity hardware with effective DDoS detection application like FNM and improved fast packet capturing frameworks such as netmap and PF_Ring ZC, has a potential to be used as DDoS defense mechanism in victim end [5].

Desti Mualfah, Imam Riadi in 2017.In the paper using Network forensics for detecting flooding attack on web server for doing that first use snort open source system to detect flooding attack and all activities in the network snort recorded in log files. Log files are used at this stage of the

investigation to the forensic process model method to find evidence. The results of this research scenario analysis obtained 15 IP Address recorded perform illegal actions on web server[14].

D.Deepthi Rani, T.V.Sai Krishna, et al in 2017. This paper discuss how to detection and prevention TCP SYN flood attack and it explains about efficient packet filtering technique using firewall to defend TCP SYN Flood attacks. Firewall scripts are written using command-line tool IP Tables in Linux to deny the suspicious traffic [15].

Kiattikul Treseangrat, Bahman Sarrafpour Samad S. Kolahi. This paper studied the impact of a UDP flood attack on Web Server with the new generation of Linux platform,namely, Linux Ubuntu 13, also evaluates the impact of various defense mechanisms, including Access Control Lists (ACLs), Threshold Limit, Reverse Path Forwarding (IP Verify), and Network Load Balancing.Threshold Limit is found to be the most effective defense[16].

Bahaa Qasim M. AL-Musawi in 2012. capability of iptables rules is explored to defend against this attack. To determine whether the network traffic is legitimate or not, a iptables relies on a set of rules it contains that are predefined by a network or system administrator. These rules tell the iptables whether to consider as legitimate and what to do with the network traffic coming from a certain source, going to a certain destination, or having a certain protocol type[6].

2.8.1 Summery

Most of the proposed work and research that we have covered in related work has used different techniques and it increases the overall operational costs to deploy mitigation solutions. Otherwise, there are several open source based intrusion and DDoS detection software's available online, Open source systems have increased considerable inclination because of their adaptability, support and cost-effectiveness. It is very important to have an essential requirement to choose the best open source from those available on the internet. The requirement based on the detection method (anomaly detection) against DDoS-based flooding attack and can deploy within the webserver, so choice FNM. Beside FNM, we can use packet filtering teachings in web server to prevent against DDoS-based flooding attacks.

CHAPTER: III

RESEARCH METHODOLOGY

3.1 Introduction

This chapter proposes an efficient framework for detection and mitigation DDoS based flooding attacks also discusses the steps of experimental setup.

3.2 Proposed Methodology

This research proposes an efficient framework for detection and mitigation DDoS based flooding attacks by use FNM open source tool for very high-performance DDoS detector and use packet filtering technique in the Linux kernel to mitigate it. The experimental setup Figure (3.1) is made by use VMware workstation to create VM victim and VM attacker, use Ubuntu operating system in Web server VM victim for test FNM to capture packet, and detection in Linux kernel also test a packet filtering in the Linux kernel to mitigate DDoS attack.

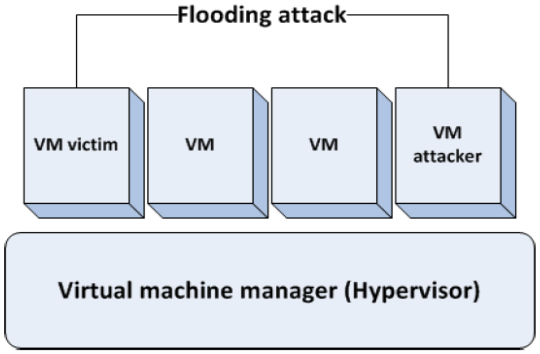


Figure (3.1) Experimental setup

Propose methodology in the Figure (3.2) show how work, in the first level detection DDoS attack of incoming traffic by use FNM within web server to show result what is incoming traffic is attacking or not, If not, forward traffic to local process of the web server, but if is attack FNM generate report attack file contains detail information about attack and notify system administrator by email then come second level mitigation web server by use packet filtering technique in Linux kernel and use iptables script to drop attack traffic.

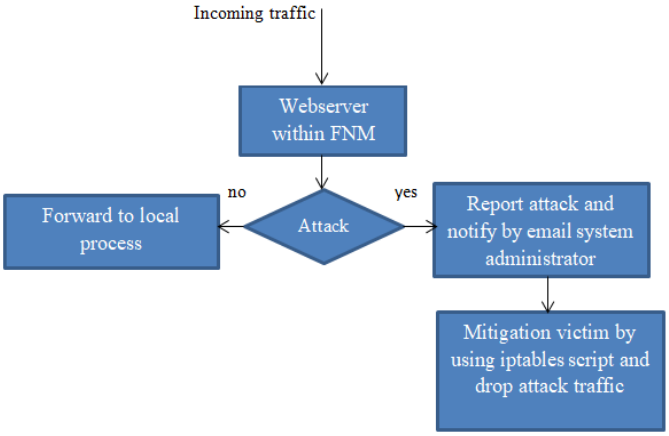


Figure (3.2) Flow chart of detection and mitigation DDoS attack

3.3 FastNetMon

FNM is a very high-performance DDoS detector built on top of multiple packets capture engines: PF_Ring, netmap, sFLOW, Netflow, PCAP. One of the interesting features of FNM is that it supports most of the network vendors and has a flexibility to be installed and modified by developer in different Linux distribution including Debian, CentOS, Ubuntu, Fedora and Gentoo. As it is designed to detect DDoS attack, it has

core algorithms that detect a pattern of different DDoS attacks. It supports anomaly detection using rate-based and protocol based to the hosts in the network. It also has additional signature-based deep packet inspection (DPI) against false positive attack detection [5].

Figure (3.3) presents FNM main software components. The main FNM software components are: Policy manager, PCE, detection engine and report manager. The policy manager is responsible for selecting one of the packets capturing modules and initializing resources (memory and CPU) based on the given hardware configuration preferences. Detection engine analysis every packet passed by the selected PCE. For some attacks, if the selected PCE provides packets with payload then advanced DPI will process the packet for false positive attack detection. Finally, report manager reports based on the detection status whether the incoming traffic is an attack or not. Afterwards, different policy enforcement devices may take an action based on the report.

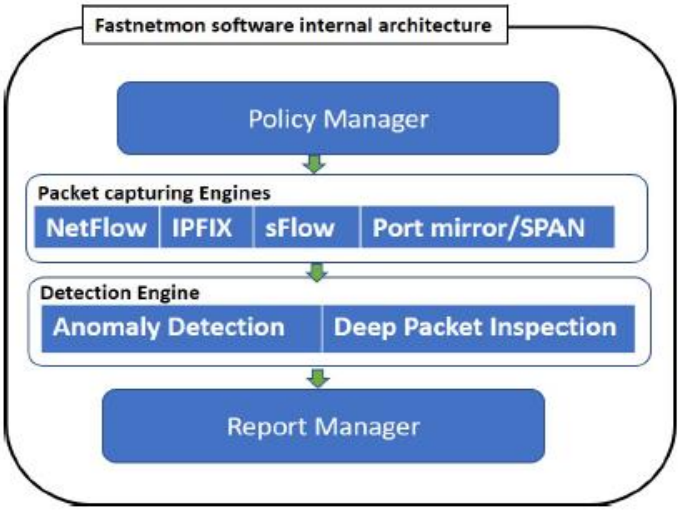


Figure (3.3) FNM software architecture

3.3.1 FastNetMon PCE

Network traffic analysis is a process used to monitor the communication pattern between hosts and towards internet in the network. This involves capturing traffic which may give limited information of a packet, which is a flow data or detailed information including packet payload. FNM supports most of current packet capturing techniques and frameworks. It supports NetFlow, sFlow and IPFIX based low data analysis for traffic collected from devices such as router or switches. This data commonly used to track key fields like: source interface, source and destination IP address, layer 4 protocols, source and destination port numbers and type of service value. FNM also supports high performance packet capturing frameworks such as netmap and PF_Ring ZC as well as common but slow packet capturing library libpcap. There is netmap-enabled version of libpcap, which enables libpcap based applications to run on top of netmap at much higher speeds. These frameworks provide packets with payload, so that FNM can apply deep packet inspection on the packet of the network traffic.

3.3.2 FastNetMon Detection Method

FNM detection logic is based on both anomaly and signature based detection methods, as can be seen from Figure (3.4). Anomaly, it detects based on the rate of the traffic incoming to or outgoing from a given networks in Classless Inter-Domain Routing (CIDR) format by policy manager. The rate is based on number of pps, mbps and flows per host. For advanced detection if the PCE provides packets with payload it uses signature based detection called nDPI. Memory consumption of FNM during detection is depends on the total number of monitored hosts. It assigns small amount of memory per host, which are data counter, current speed counter and traffic counters.

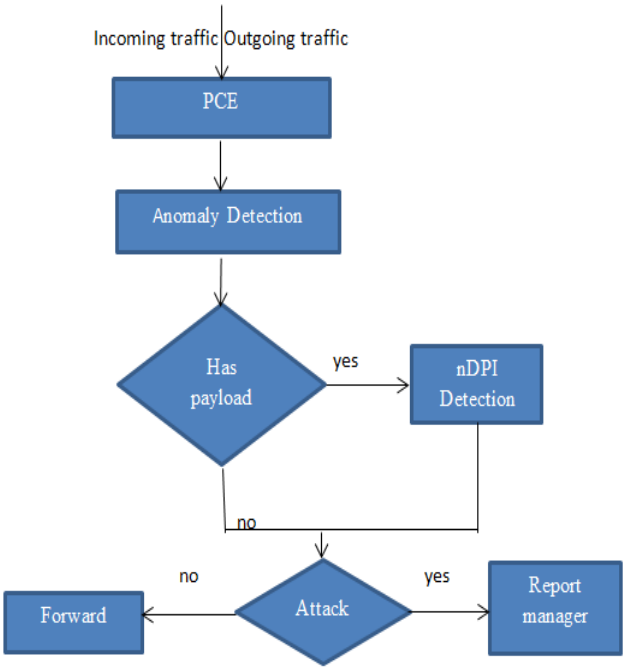


Figure (3.4) Flow diagram of detection in FNM

Using the above detection methods, FNM detects the following attack types:

– TCP-SYN flood: TCP packets with enabled SYN flag.

– UDP flood: flood with UDP packets.

– ICMP flood: flood with ICMP packets.

– IP fragmentation flood: IP packets with MF2 flag set or with nonzero fragment offset.

3.3.3 FastNetMon Report

After detecting attack FNM report module will write details of the attack in file or dumps traces in pcap for the attack traffics. If FNM is configured to take action based on the report it runs external triggers to:

– notify attack summery using custom script.

– Announce with Border Gateway Protocol (BGP) (EaxBGP) [5].

3.4 VMware Workstation

VMware Workstation is a hosted hypervisor that runs on x64 versions of Windows and Linux operating systems] (an x86 version of earlier releases was available), it enables users to set up virtual machines (VMs) on a single physical machine, and use them simultaneously along with the actual machine. Each virtual machine can execute its own operating system, including versions of Microsoft Windows, Linux, BSD, and MS-DOS. VMware Workstation is developed and sold by VMware, Inc., a division of Dell Technologies. There is a free-of-charge version, VMware Workstation Player, for non-commercial use. An operating systems license is needed to use proprietary ones such as Windows. Ready-made Linux VMs set up for different purposes are available from several sources.

VMware Workstation supports bridging existing host network adapters and sharing physical disk drives and USB devices with a virtual machine. It can simulate disk drives; an ISO image file can be mounted as a virtual optical disc drive, and virtual hard disk drives are implemented as .vmdk files.

VMware Workstation Pro can save the state of a virtual machine (a "snapshot") at any instant. These snapshots can later be restored, effectively returning the virtual machine to the saved state as it was and free from any post-snapshot damage to the VM [17].

3.5 Kali Linux

Kali Linux is a Debian-based Linux distribution aimed at advanced Penetration Testing and Security Auditing, is developed, funded and maintained by Offensive Security, a leading information security training company. Kali contains several hundred tools which are geared towards various information security tasks, such as Penetration Testing, Security research, Computer Forensics and Reverse Engineering. Kali Linux was released on the 13th March, 2013 as a complete, top-to-bottom rebuild of BackTrack Linux [18].

3.6 Ubuntu

Ubuntu 14.04 is a free and open-source operating system and Linux distribution based on Debian. Ubuntu is offered in three official editions: Ubuntu Desktop for personal computers, Ubuntu Server for servers and the cloud and Ubuntu Core for Internet of things devices and robots. New releases of Ubuntu occur every six months, while long-term support (LTS) releases occur every two years. Ubuntu is produced by Canonical and the developer community, under a meritocratic governance model. Canonical provides free guaranteed security updates and support for each Ubuntu release, starting from the release date and until the release reaches its predestinated end-of-life (EOL) date. Canonical generates revenue through the sale of premium services related to Ubuntu. Ubuntu is named after the

Southern African philosophy of ubuntu (literally, 'human-ness'), which Canonical suggests can be loosely translated as "humanity to others" or "I am what I am because of who we all are". Ubuntu is the most popular operating system for the cloud, and is the reference operating system for Open Stack [19].

3.7 DDoS attacks tools

The most common DDoS tools such as LOIC and hping3 are used based on the capability of types of DDoS attacks. Hping3 is one of the de-facto tools for security auditing and testing of firewalls and networks, and was used to exploit the Idle Scan scanning technique now implemented in the Nmap port scanner. Hping is designed to generate packets and analyses TCP/IP protocols. It is a command-line oriented with desirable parameters including:

– Flood: sending packets as fast as possible.

– S: TCP with SYN flag.

– D: data size.

– C: packet count.

– Random-source: random the source address or spoofing.

And much more parameters can be passed to hping3. It is easy to manipulate packets using hping, which makes it a best tool for DDoS attacks [5].

3.8 NMAP

Nmap is port-scanning tool was developed by Fyodor Yarochkin and is one of the most well-known port-scanning tools. Nmap is available for Windows and Linux as a GUI and command-line program. It can do many types of scans and OS identification. It also has the ability to blind scan and zombie scan, and it enables you to control the speed of the scan from slow to very fast.

3.9 IPTABLES

Iptables is part of the Netfilter project. Netfilter is a set of Linux kernel hooks that communicate with the network stack. Iptables is a command and the table structure that contains the rule sets that control the packet filtering. Iptables is complex. It filters packets by the fields in IP, TCP, UDP, and ICMP packet headers. A number of different actions can be taken on each packet, so the key to iptables happiness is simplicity. Start with the minimum necessary to get the job done, then add rules as you need them. It’s not necessary to build vast iptables edifices, and in fact, it’s a bad idea, as it makes it difficult to maintain, and will hurt performance.

There are three tables in iptables. Any rules or custom chains that you create will go into one of these tables. The filter table is the default, and is the one most used. The filter table contains these built-in chains:

INPUT: Processes incoming packets.

FORWARD: Processes packets routed through the host.

OUTPUT: Processes outgoing packets [6].

CHAPTER: IV

IMPLEMENTATION AND THE RESULT DISCUSSION

4.1 Introduction

The previous chapter the details of the proposed model were described and how detection and mitigation DDoS attack in an experimental setup. This chapter shows the implementation of the experimental setup in the term virtual network environment. A detailed description of how attacking from VM attacker to webserver VM victim and how detection and mitigation DDoS based flooding attack by using FNM and packet filtering technique.

4.2 Implementation Steps

The experimental setup is a simulated of architecture of victim end defense mechanism, as presented in Figure (4.1). VMware workstation was used to create experimental setup by creating two virtual machines as VM victim and VM attacker, IP address of victim is 192.168.237.130 and DDoS attack tools are pre-installed in generator system (Kali inux) VM attacker.

Step1: To establish a segregate network using virtualization. VMware Workstation is used to establish a segregate network and two UBUNTU 14.04 LTS and KALII 2016 operating systems are installed on it.

Step2: Installed and configured postfix mail server, Installed Apache webserver in victim machine.

Step3: Installed and configuration FNM open source tool and start services for work.

Step4: Identify open ports on victim machine used NMAP tool.

Step5: Generated attack traffic using HPING3 tool on VM attacker and flood VM victim with spoofed packets.

Step6: FNM open source tool detected attack and generated attack report, notified system administrator by email and dumped attack traffic in PCAP file

Step7: Mitigated victim by using IPTABELS script and drop flooding attack.

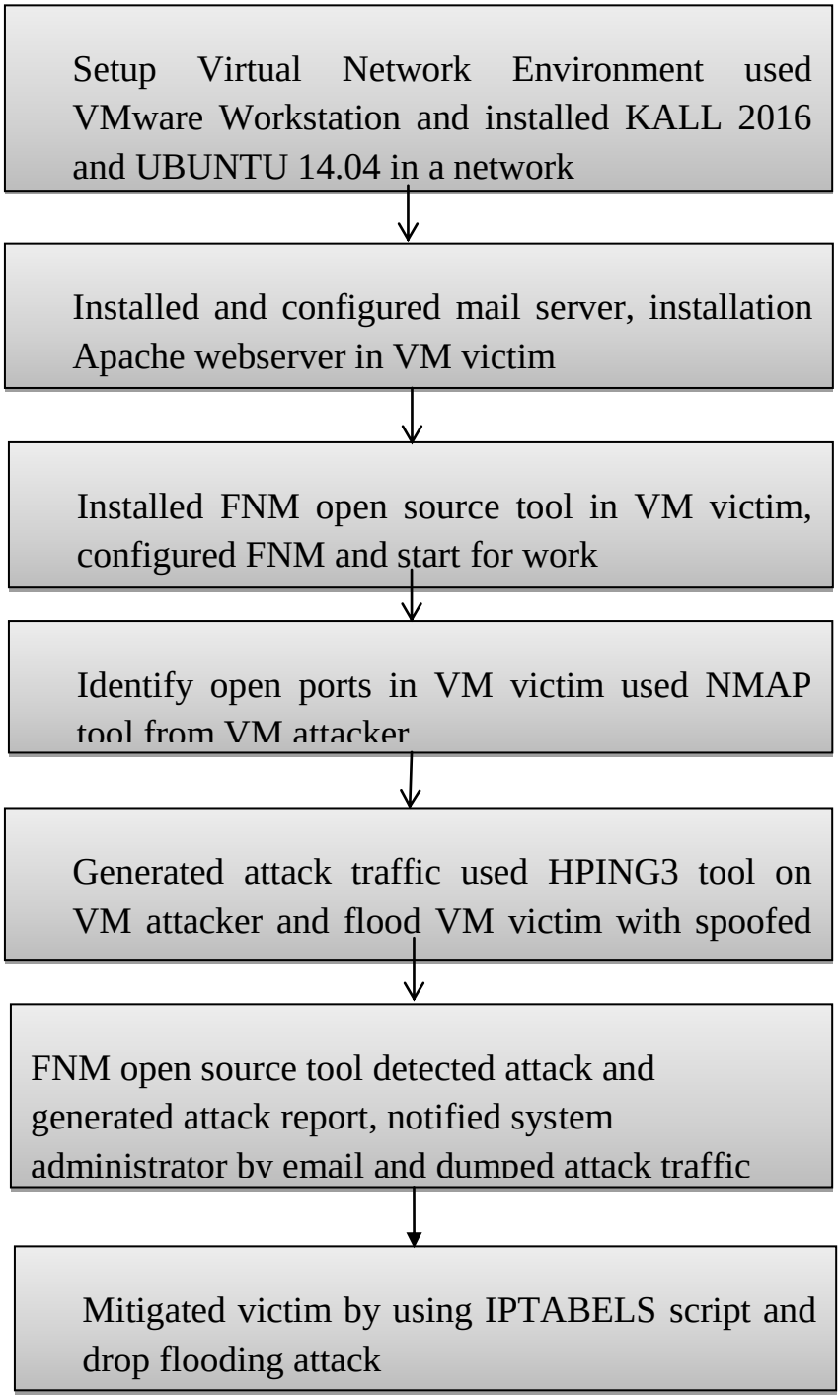


Figure (4.1) Implementation steps

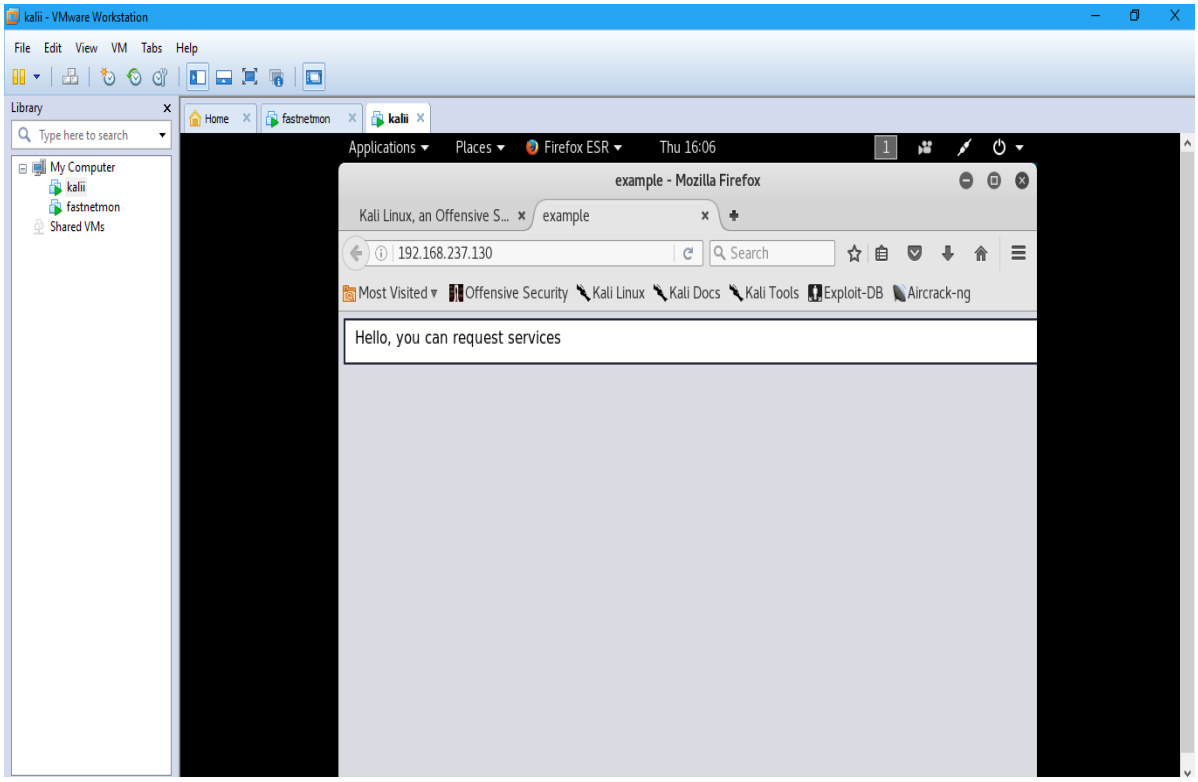


Figure (4.2) Experimental setup

4.3 FatNetMon Installation

FNM is automatically installed used the following command [14] from git repository :

#Wget https://raw.githubusercontent.com/pavelodintsov/fastnetmon/master/src/fastnetmon_install.pl Ofastnetmon_install.pl

#sudo perl fastnetmon_install.pl

In order to modify the source code of FastNetMon, we have installed the community developer version used the following command after the above automatic installation:

```
#cd /usr/src

#git clone https://github.com/pavel-odintsov/fastnetmon.git

#cd fastnetmon

#git checkout master

#cd src/build

#cmake ..

#make

#./fastnetmon
```

4.4 FastNetMon Configuration

First must be configured network list in file /etc/networks_list in CIDR form: 192.168.237.0/24 then altered in configuration file to define being detection network, specified interfaces and to configure detailed detected preference can be found in /etc/fastnetmon.conf as sample in Figure (4.3), also enable process incoming and outgoing traffic, specified 100 packets will be collected from attack traffic and specified 1900 in seconds we should keep web server in blocked state.

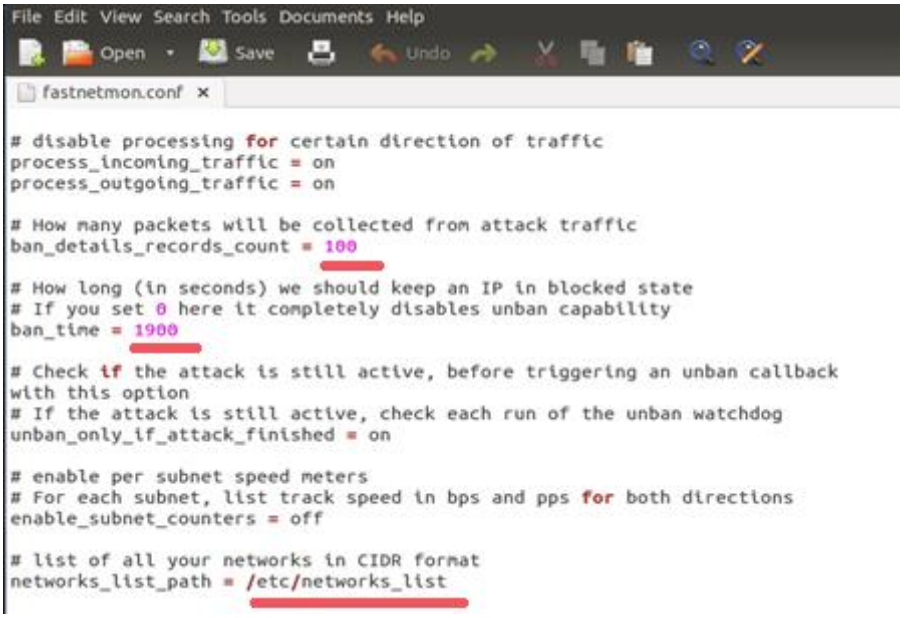


Figure (4.3) FNM sample configuration1

Second in configuration file altered packet capture engine and we used Linux kernel packet capture AF_PACKET and anomaly detection (Rate-based) by enabling ban_for_pps and specific threshold to 1000 as Figure (4.4). After configured start FNM services by command #/etc/init.d/fastnetmon start, then open FNM service by command #/opt/fastnetmon/fastnetmon in another terminal open fastnetmon client to show realistic traffic numbers by command #/opt/fastnetmon/fastnetmon_client.

```
# redraw period for client's screen
check_period = 1

# Connection tracking is very useful for attack detection because it provides
huge amounts of information,
# but it's very CPU intensive and not recommended in big networks
enable_connection_tracking = on

# Different approaches to attack detection
ban_for_pps = on
ban_for_bandwidth = off
ban_for_flows = off

# Limits for Dos/DDoS attacks threshold_pps
threshold_pps = 1000
threshold_mbps = 1000
threshold_flows = 3500

###
### Traffic capture methods
###

# PF_RING traffic capture, fast enough but the wirespeed version needs a paid
license
mirror = off

# Port mirroring sample rate
pfring_sampling_ratio = 1

# Netmap traffic capture (very fast but needs patched drivers)
mirror_netmap = off

# SnabbSwitch traffic capture
mirror_snabbswitch = off

# AF_PACKET capture engine
# Please use it only with modern Linux kernels (3.6 and more)
# And please install birq for irq ditribution over cores
mirror_afpacket = on
```

Figure (4.4) configuration detection method and packet capture method

4.5 Port Scanning in VM Attacker

Attacker to flooding VM victim must be scanned VM victim to know open ports by using map tool. Figure (4.5) shows nmap command and output, nmap command to discover well-known open ports with option p and trace target with option T4 for more information appear in nmap output, it shows port number and protocol, service name, state and service version.

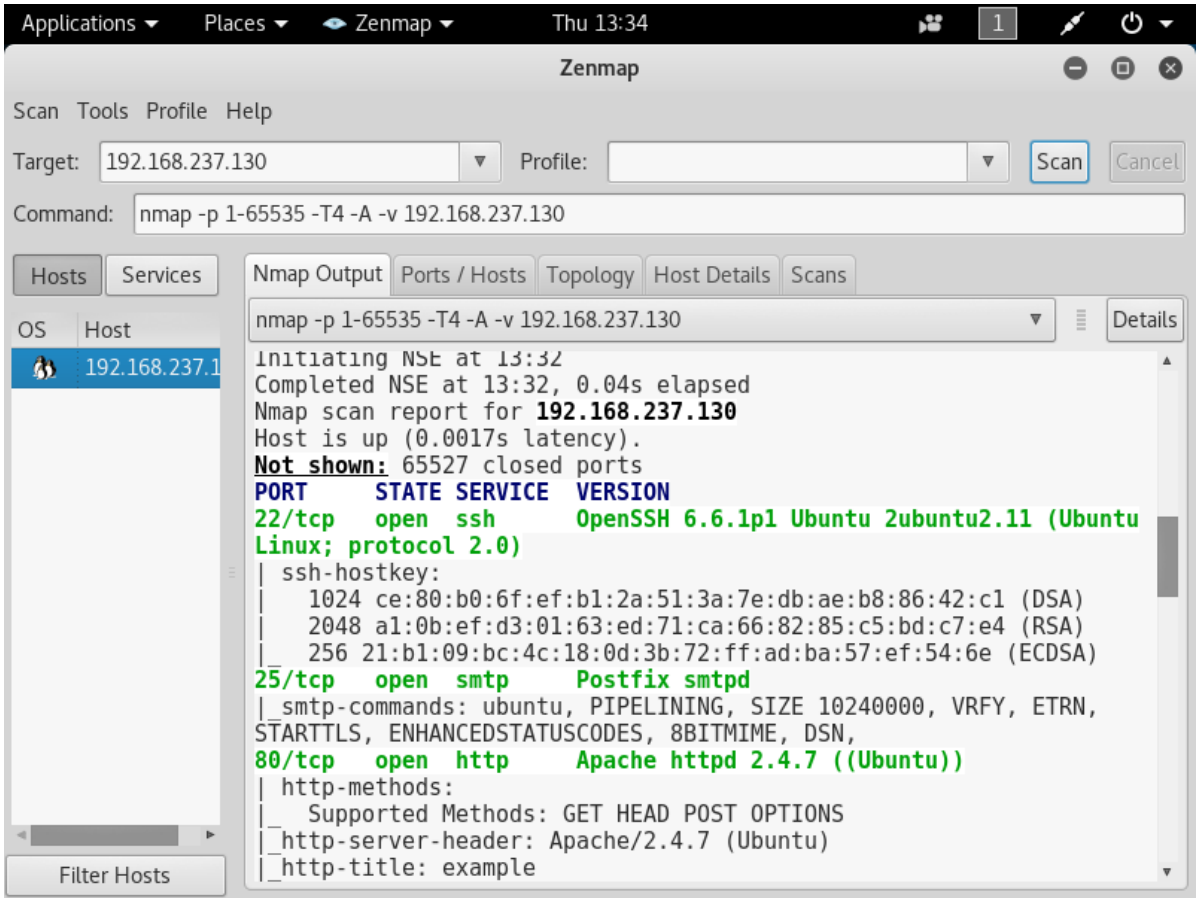


Figure (4.5) Nmap ports scanning

Above Figure show the result of port scan of the victim, its show open ports and services provided by thesis ports, aim of this research is web services so port 80 is the target port.

4.6 SYN Flood Attack

SYN flood attack was made by flooding the victim machine by running following hping command form attacker machine with parameters: **-S** flag sets the SYN flag on in TCP mode, --flood flag sends the packet as fast as possible, -d packet size of 120 bytes, -w window size of 64, --rand-source generates random spoof source IP address, -p target port 80 and target IP address as in Figure (4.6).

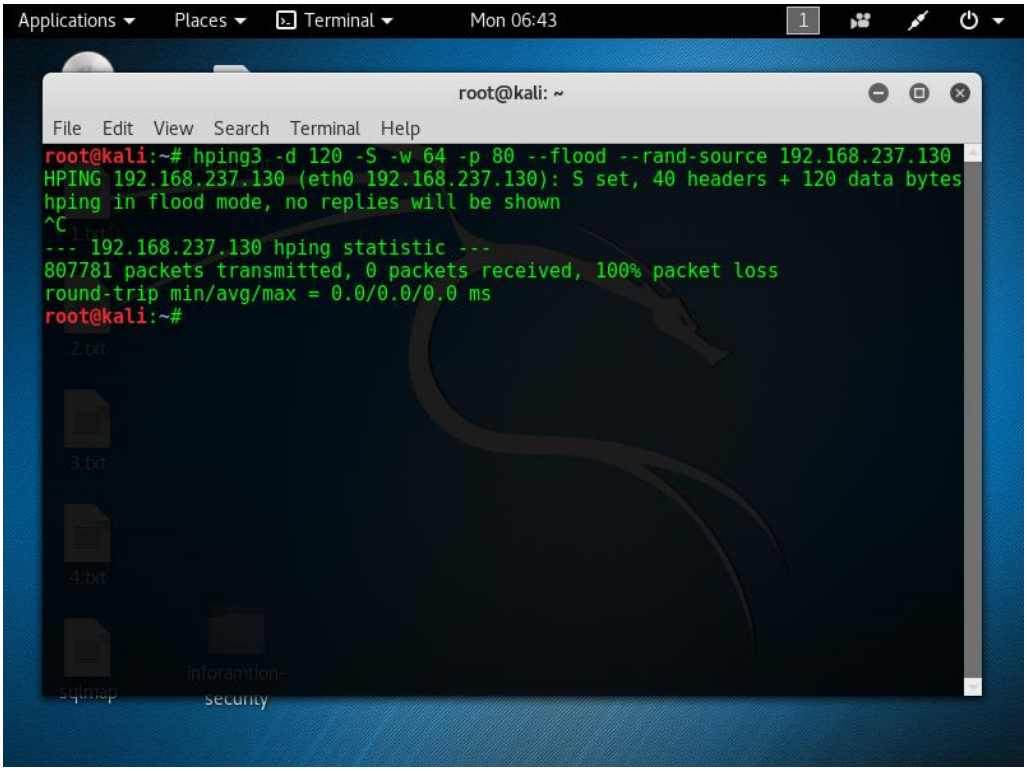
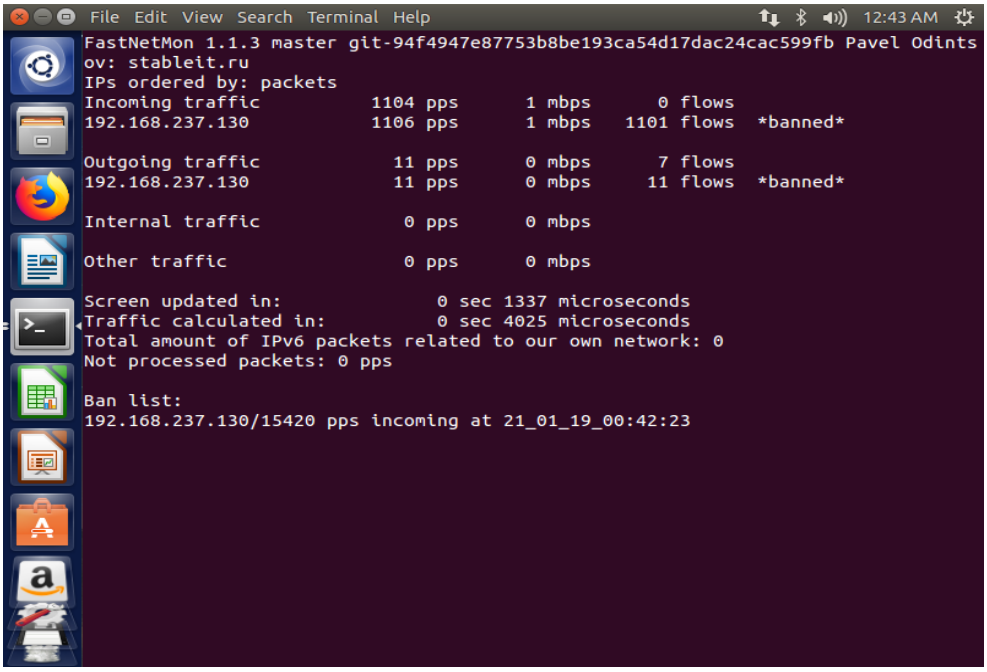


Figure (4.6) TCP-SYN flood attack

Above Figure show how send SYN flood attack by sending huge amount of SYN packets requests to the victim with spoof IP addresses that‘s mean it is not send last packet from handshake protocol to the victim, which result huge amount of the half open connection and exhaust resource victim and denial of service.

On victim machine, FNM monitoring and analysis incoming and outgoing traffic as in Figure (4.7) and a countable number of arriving packets per second If it exceeds threshold detection attack, trigger ban, generate attack report, notify to network administrator to send attack details as in Figure (4.8) and trace attack traffic to pcap file.



Figure(4.7) Statistic information of SYN attack in fastnetmon_client

Above Figure show how the FNM monitoring victim then shows statistic information about SYN attack traffic, when detected attack trigger system administrator and generated attack report file in Ban list with filename contains time of the detected attack(21_01_19_00:42:23).

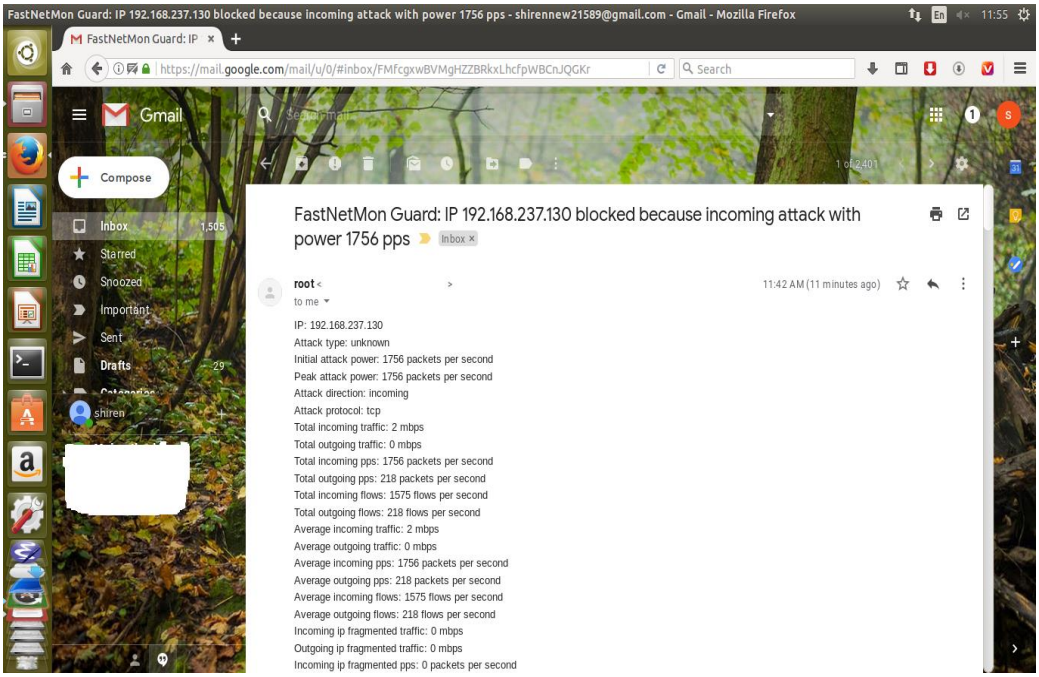


Figure (4.8) Email notify of SYN attack

Above Figure show notification message was sent by FNM when detected attack, notification message show details information's about an attack like IP of the victim, attack type, initial attack power and total outgoing pps and other information's.

FNM generates attack report as in Figure (4.9) it shows detailed information: victim IP address, attack type is unknown, but syn tcp pps explain is SYN flood attack, initial attack power, direction attack and attack protocol. Also explain total incoming and outgoing pps and tcp flaws. Incoming tcp flow for syn flag to establish connection form attacker when outgoing tcp flow for syn-ack from victim to spoof IP addresses it appears also in attack trace pcap file as in Figure (4.10).

```
IP: 192.168.237.130
Attack type: unknown
Initial attack power: 1756 packets per second
Peak attack power: 1756 packets per second
Attack direction: incoming
Attack protocol: tcp
Total incoming traffic: 2 mbps
Total outgoing traffic: 0 mbps
Total incoming pps: 1756 packets per second
Total outgoing pps: 218 packets per second
Total incoming flows: 1575 flows per second
Total outgoing flows: 218 flows per second
Average incoming traffic: 2 mbps
Average outgoing traffic: 0 mbps
Average incoming pps: 1756 packets per second
Average outgoing pps: 218 packets per second
Average incoming flows: 1575 flows per second
Average outgoing flows: 218 flows per second
Incoming ip fragmented traffic: 0 mbps
Outgoing ip fragmented traffic: 0 mbps
Incoming ip fragmented pps: 0 packets per second
Outgoing ip fragmented pps: 0 packets per second
Incoming tcp traffic: 2 mbps
Outgoing tcp traffic: 0 mbps
Incoming tcp pps: 1756 packets per second
Outgoing tcp pps: 218 packets per second
Incoming syn tcp traffic: 2 mbps
Outgoing syn tcp traffic: 0 mbps
Incoming syn tcp pps: 1569 packets per second
Outgoing syn tcp pps: 218 packets per second
Incoming udp traffic: 0 mbps
Outgoing udp traffic: 0 mbps
Incoming udp pps: 0 packets per second
Outgoing udp pps: 0 packets per second
Incoming icmp traffic: 0 mbps
Outgoing icmp traffic: 0 mbps
Average packet size for incoming traffic: 162.4 bytes
Average packet size for outgoing traffic: 58.5 bytes
Incoming

TCP flows: 6115
192.168.237.130:80 < 206.111.88.114:1553 60 bytes 1 packets
192.168.237.130:80 < 45.190.6.185:1554 60 bytes 1 packets
192.168.237.130:80 < 211.26.28.28:1555 60 bytes 1 packets
192.168.237.130:80 < 206.250.114.232:1556 60 bytes 1 packets
192.168.237.130:80 < 32.101.169.11:1557 60 bytes 1 packets
192.168.237.130:80 < 53.126.144.209:1558 60 bytes 1 packets
192.168.237.130:80 < 142.161.49.158:1560 60 bytes 1 packets
192.168.237.130:80 < 140.213.248.126:1566 60 bytes 1 packets
192.168.237.130:80 < 37.80.71.252:1567 60 bytes 1 packets
192.168.237.130:80 < 163.211.16.191:1579 60 bytes 1 packets
192.168.237.130:80 < 139.119.251.161:1580 60 bytes 1 packets
192.168.237.130:80 < 202.14.231.43:1581 60 bytes 1 packets
192.168.237.130:80 < 43.156.161.172:1597 60 bytes 1 packets
192.168.237.130:80 < 11.101.184.41:1598 60 bytes 1 packets
192.168.237.130:80 < 208.115.163.137:1604 60 bytes 1 packets
192.168.237.130:80 < 79.54.106.14:1605 60 bytes 1 packets
192.168.237.130:80 < 48.99.140.128:1606 60 bytes 1 packets
192.168.237.130:80 < 159.231.23.43:1774 60 bytes 1 packets
192.168.237.130:80 < 110.48.27.161:1790 60 bytes 1 packets
Outgoing

TCP flows: 465
192.168.237.130:80 > 251.219.32.190:1298 58 bytes 1 packets
192.168.237.130:80 > 244.200.54.88:1302 58 bytes 1 packets
192.168.237.130:80 > 248.71.47.74:1320 58 bytes 1 packets
192.168.237.130:80 > 249.233.164.129:1373 58 bytes 1 packets
192.168.237.130:80 > 245.208.253.18:1381 58 bytes 1 packets
192.168.237.130:80 > 245.140.141.192:1390 58 bytes 1 packets
192.168.237.130:80 > 253.140.237.119:1396 58 bytes 1 packets
192.168.237.130:80 > 247.4.23.61:1404 58 bytes 1 packets
192.168.237.130:80 > 244.132.231.3:1412 58 bytes 1 packets
192.168.237.130:80 > 252.137.250.39:1431 58 bytes 1 packets
192.168.237.130:80 > 253.237.53.164:1433 58 bytes 1 packets
192.168.237.130:80 > 253.173.106.126:1445 58 bytes 1 packets
192.168.237.130:80 > 245.199.140.186:1459 58 bytes 1 packets
192.168.237.130:80 > 245.149.185.80:1465 58 bytes 1 packets
192.168.237.130:80 > 245.23.234.249:1485 58 bytes 1 packets
192.168.237.130:80 > 252.156.246.45:1504 58 bytes 1 packets
```

Figure (4.9) SYN flood attack report

The above Figure show contains report file and details information about attack traffic like attack type and incoming syn tcp pps clarify it is a SYN flood attack, outgoing pps is 218 represent syn-ack response packets from victim to spoof IP addresses.

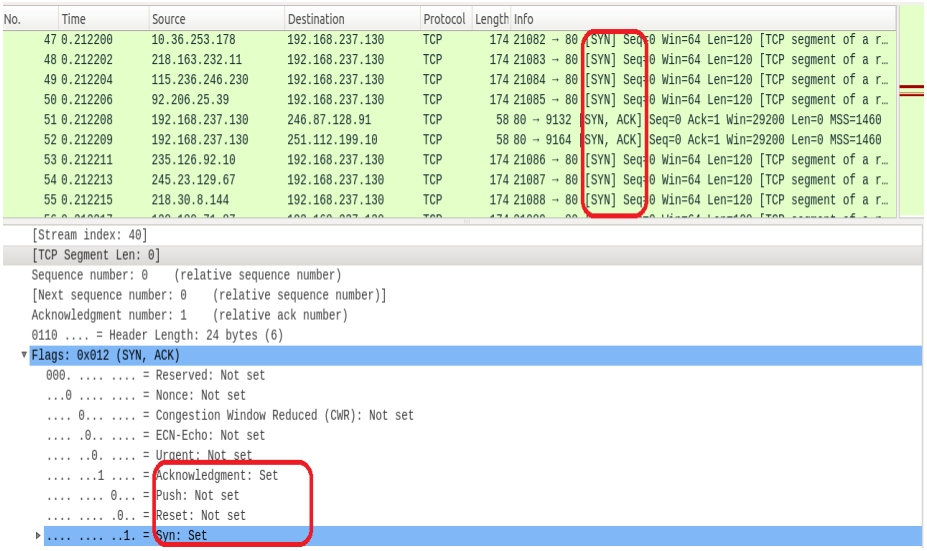


Figure (4.10) Sample trace SYN flood attack

Above Figure show trace attack traffic, red boxes show how the SYN flood attack was happened when saw a huge amount of syn packets request with spoof IP addresses and outgoing syn-ack packet response represents by Acknowledgment and syn flag is set, that is result half-open connection in webserver then denial of service.

To defend against SYN Flood Attack, iptables script is writing as bellow.

```
# iptables -N syn_flood

# iptables -A INPUT -p tcp --syn -j syn_flood

# iptables -A syn_flood -m limit --limit 1/s --limit-burst 3 -j RETURN

# iptables -A syn_flood -j DROP
```

All incoming connection are allowed till limit is reached, --limit 1/s is Maximum average matching rate in seconds, --limit-burst 3 is Maximum initial number of packets to match, FNM work again after applied script and generate attack report as in Figure (4.11) note for outgoing tcp flow is 1 and outgoing pps is 0, also in attack pcap trace file in Figure (4.12) that because all the attack packets are being dropped by the firewall according to the iptables rules.

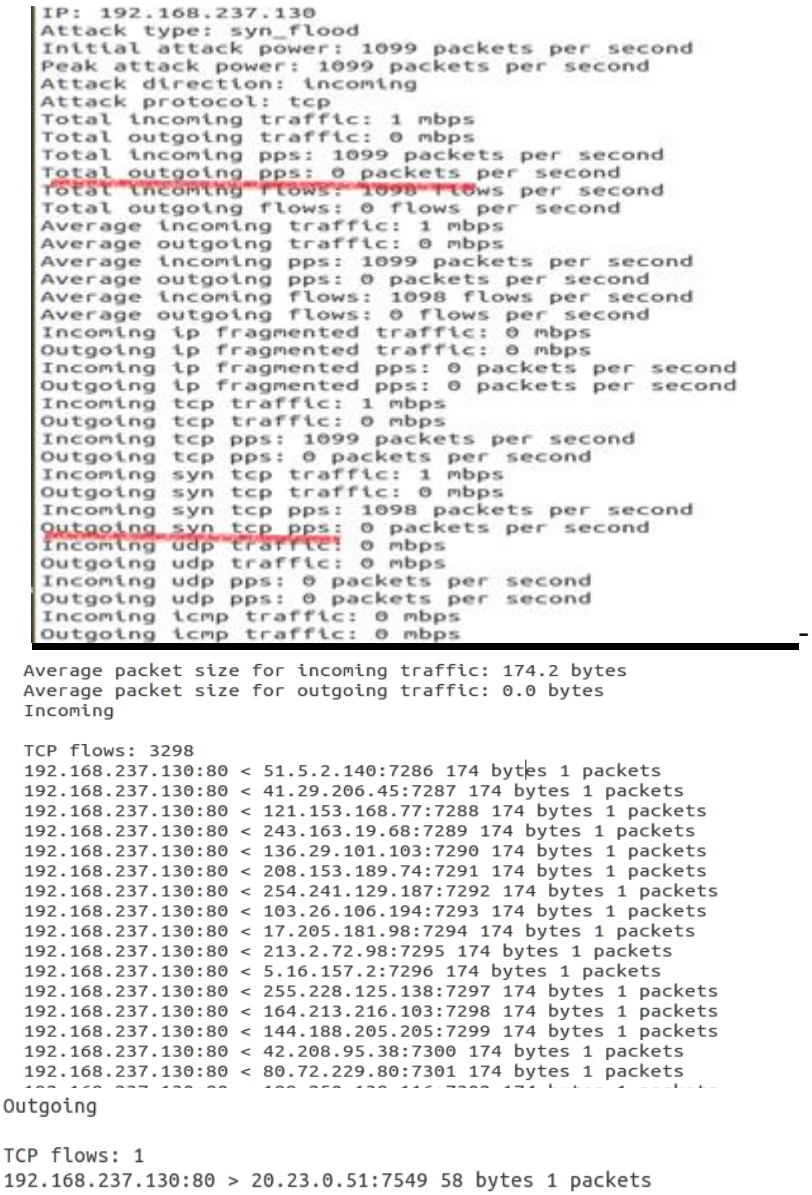


Figure (4.11) SYN flood attack report after applied iptables

Above Figure show attack report after applying iptables script to defend SYN flood attack, iptables command limit connection and drop attack traffic that result no syn-ack packet response was generated from server to spoofed IP address so outgoing pps attribute in report file is 0 that is contributed production server resource.

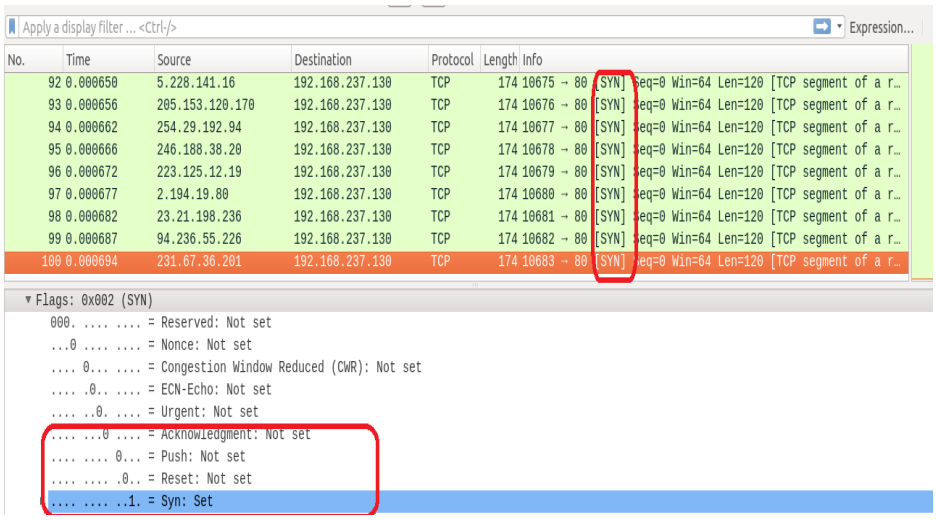


Figure (4.12) Sample trace SYN flood attack after applied iptables

Above Figure show trace attack traffic after applied iptables, red box just show syn packet request and not found outgoing syn-ack packet response from server that represent in the red boxes because drop attack traffic by iptables.

4.7 UDP Flood Attack

The attack was made by Flooding the victim's machine by running following Hping command from attacker machine with parameters: data size 32 + header 28 total of 60 bytes, destination port 80, random IP sources and ip address target. –udp flag sets to udp mode as in Figure (4.13).

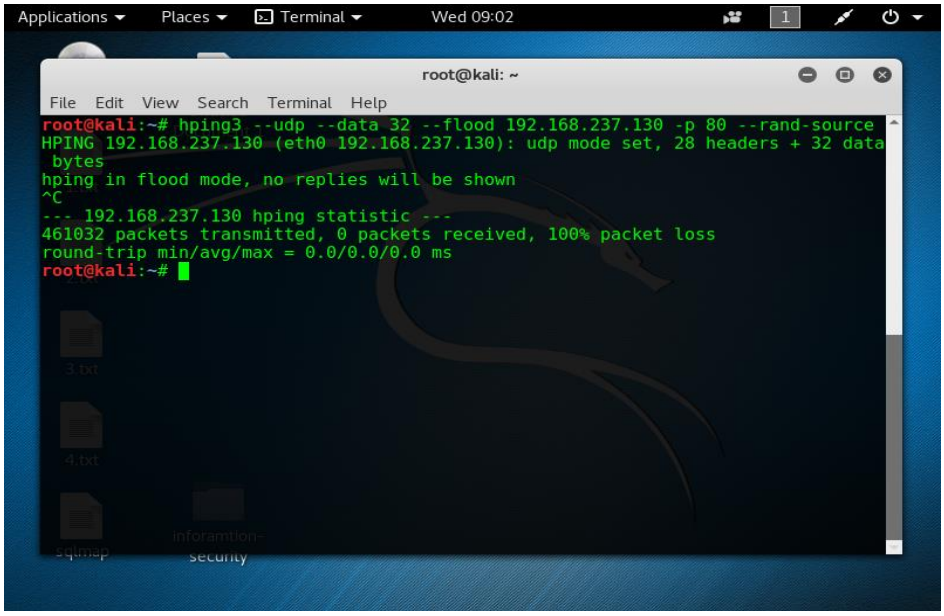


Figure (4.13) UDP flood attack

Above Figure show how send UDP flood attack by sending huge amount of UDP packets requests to the victim with spoof IP addresses as a result, the finite resource of victim network will be exhausted by the process of checking and responding for a huge volume of UDP request floods. This results in denial of service for legitimate traffic.

On victim machine, FNM monitoring and analysis incoming and outgoing traffic as in Figure (4.14) and a countable number of arriving packets per second If it exceeds threshold detection attack, trigger ban, generate attack report, notify to network administrator to send attack details as in Figure (4.15) and trace attack traffic to pcap file.

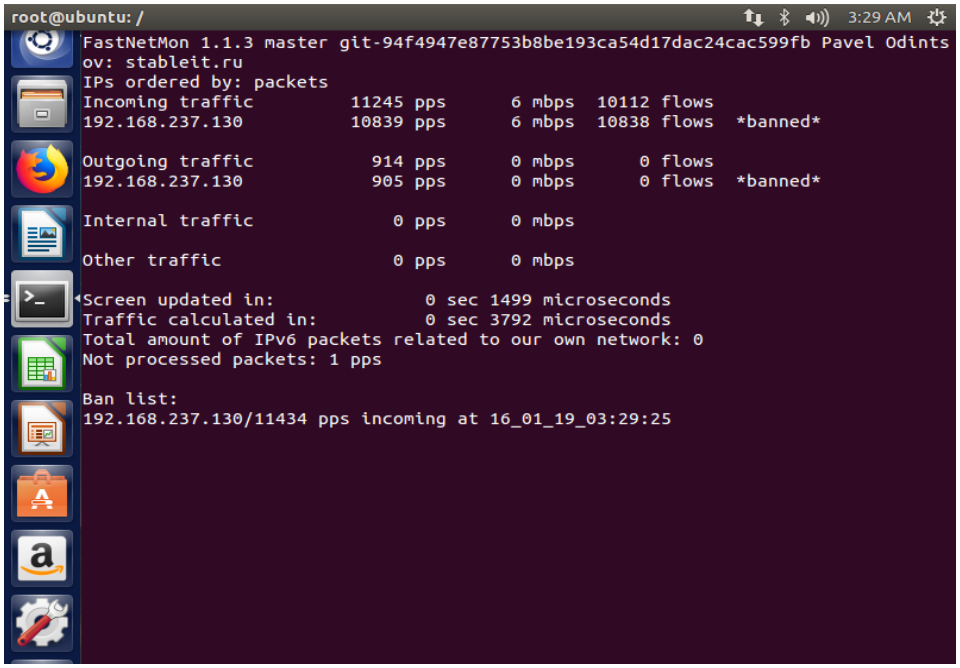


Figure (4.14) Statistic information of UDP attack in fastnetmon_client

Above Figure show how the FNM monitoring victim then shows statistic information about UDP attack traffic, when detected attack trigger system administrator and generated attack report file in Ban list with filename contains time of the detected attack (16_01_19_03:29:25).

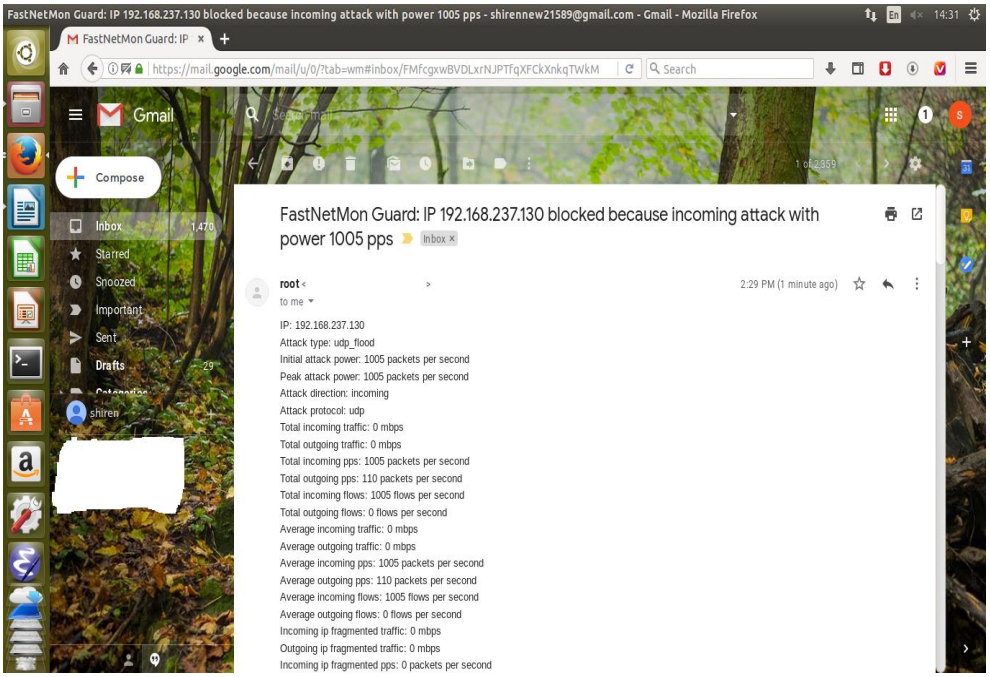


Figure (4.15) Email notify of UDP attack

FNM generates attack report as in Figure (4.16) it shows detailed information: victim IP address, attack type is udp_flood, initial attack power, direction attack and attack protocol. Also explain total incoming pps and udp flows. Incoming udp flows for send UDP packet to port when no application that is waiting on the port is, it will generate an ICMP packet of "destination unreachable" to forged source address show it in outgoing icmp pps 110 in report and attack trace pcap file as in Figure (4.17).

```
IP: 192.168.237.130
Attack type: udp_flood
Initial attack power: 1005 packets per second
Peak attack power: 1005 packets per second
Attack direction: incoming
Attack protocol: udp
Total incoming traffic: 0 mbps
Total outgoing traffic: 0 mbps
Total incoming pps: 1005 packets per second
Total outgoing pps: 110 packets per second
Total incoming flows: 1005 flows per second
Total outgoing flows: 0 flows per second
Average incoming traffic: 0 mbps
Average outgoing traffic: 0 mbps
Average incoming pps: 1005 packets per second
Average outgoing pps: 110 packets per second
Average incoming flows: 1005 flows per second
Average outgoing flows: 0 flows per second
Incoming ip fragmented traffic: 0 mbps
Outgoing ip fragmented traffic: 0 mbps
Incoming ip fragmented pps: 0 packets per second
Outgoing ip fragmented pps: 0 packets per second
Incoming tcp traffic: 0 mbps
Outgoing tcp traffic: 0 mbps
Incoming tcp pps: 0 packets per second
Outgoing tcp pps: 0 packets per second
Incoming syn tcp traffic: 0 mbps
Outgoing syn tcp traffic: 0 mbps
Incoming syn tcp pps: 0 packets per second
Outgoing syn tcp pps: 0 packets per second
Incoming udp traffic: 0 mbps
Outgoing udp traffic: 0 mbps
Incoming udp pps: 1005 packets per second
Outgoing udp pps: 0 packets per second
Incoming icmp traffic: 0 mbps
Outgoing icmp traffic: 0 mbps
Incoming icmp pps: 0 packets per second
Outgoing icmp pps: 110 packets per second

Average packet size for incoming traffic: 74.1 bytes
Average packet size for outgoing traffic: 102.9 bytes
Incoming

UDP flows: 5548
192.168.237.130:80 < 90.253.62.236:2493 74 bytes 1 packets
192.168.237.130:80 < 125.133.92.51:2494 74 bytes 1 packets
192.168.237.130:80 < 121.79.102.1:2495 74 bytes 1 packets
192.168.237.130:80 < 101.99.132.242:2496 74 bytes 1 packets
192.168.237.130:80 < 61.22.144.171:2497 74 bytes 1 packets
192.168.237.130:80 < 235.58.133.211:2498 74 bytes 1 packets
192.168.237.130:80 < 89.133.243.0:2499 74 bytes 1 packets
192.168.237.130:80 < 86.211.5.173:2500 74 bytes 1 packets
192.168.237.130:80 < 4.101.119.148:2501 74 bytes 1 packets
192.168.237.130:80 < 195.91.131.181:2502 74 bytes 1 packets
192.168.237.130:80 < 170.182.110.87:2503 74 bytes 1 packets
```

Figure (4.16) UDP flood attack report

The above Figure show contains report file and details information about attack traffic like attack type and incoming udp pps is 1005 and outgoing icmp pps is 110, outgoing icmp it is represent data packets issued by the server responding to the attack.

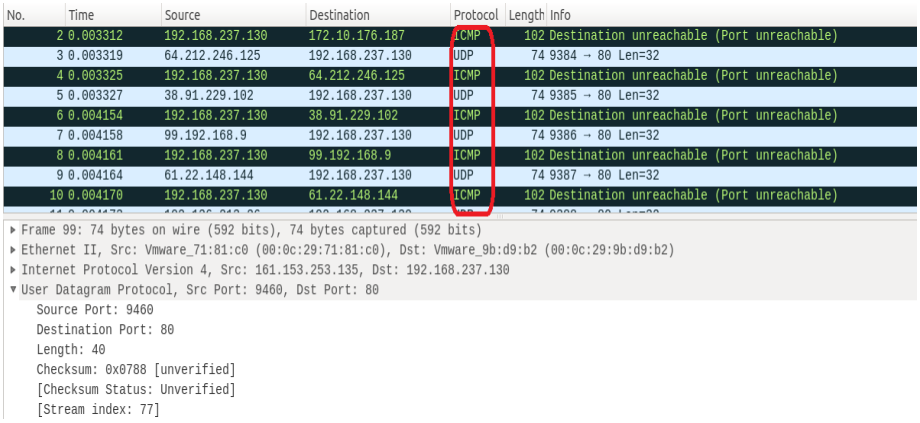


Figure (4.17) Sample trace UDP flood traffic

Above Figure show trace attack traffic, red boxes show how the UDP flood attack was happened when saw a huge amount of UDP packets request with spoof IP addresses and outgoing ICMP packet of "destination unreachable" packet response.

To defend against UDP Flood Attack, iptables script is writing as bellow:

iptables -N udp_flood

iptables -A INPUT -p udp -j udp_flood

iptables -A udp_flood -m state –state NEW –m recent –update –seconds 1 –hitcount 10 –j RETURN

iptables -A udp_flood -j DROP

drop all incoming connection are more than 10 connection in 1 second according to the iptables so victim machine is not responding or send any packet, FNM work again after applied script and generate attack report as in Figure (4.18) note for outgoing icmp pps is 0 ,also in attack pcap trace file in Figure (4.19).

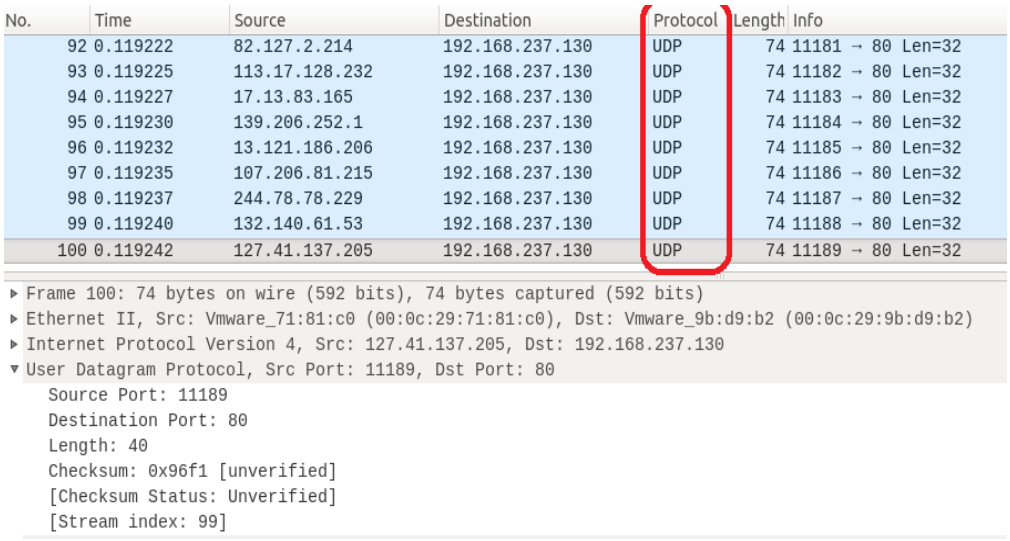
```
IP: 192.168.237.130
Attack type: udp_flood
Initial attack power: 1594 packets per second
Peak attack power: 1594 packets per second
Attack direction: incoming
Attack protocol: udp
Total incoming traffic: 0 mbps
Total outgoing traffic: 0 mbps
Total incoming pps: 1594 packets per second
Total outgoing pps: 0 packets per second
Total incoming flows: 1594 flows per second
Total outgoing flows: 0 flows per second
Average incoming traffic: 0 mbps
Average outgoing traffic: 0 mbps
Average incoming pps: 1594 packets per second
Average outgoing pps: 0 packets per second
Average incoming flows: 1594 flows per second
Average outgoing flows: 0 flows per second
Incoming ip fragmented traffic: 0 mbps
Outgoing ip fragmented traffic: 0 mbps
Incoming ip fragmented pps: 0 packets per second
Outgoing ip fragmented pps: 0 packets per second
Incoming tcp traffic: 0 mbps
Outgoing tcp traffic: 0 mbps
Incoming tcp pps: 0 packets per second
Outgoing tcp pps: 0 packets per second
Incoming syn tcp traffic: 0 mbps
Outgoing syn tcp traffic: 0 mbps
Incoming syn tcp pps: 0 packets per second
Outgoing syn tcp pps: 0 packets per second
Incoming udp traffic: 0 mbps
Outgoing udp traffic: 0 mbps
Incoming udp pps: 1594 packets per second
Outgoing udp pps: 0 packets per second
Incoming icmp traffic: 0 mbps
Outgoing icmp traffic: 0 mbps
Incoming icmp pps: 0 packets per second
Outgoing icmp pps: 0 packets per second

Average packet size for incoming traffic: 74.1 bytes
Average packet size for outgoing traffic: 0.0 bytes
Incoming

UDP flows: 5497
192.168.237.130:80 < 134.196.72.112:5593 74 bytes 1 packets
192.168.237.130:80 < 31.250.109.157:5594 74 bytes 1 packets
192.168.237.130:80 < 4.79.251.40:5595 74 bytes 1 packets
192.168.237.130:80 < 131.122.173.22:5596 74 bytes 1 packets
192.168.237.130:80 < 69.55.215.80:5597 74 bytes 1 packets
192.168.237.130:80 < 78.32.95.17:5598 74 bytes 1 packets
192.168.237.130:80 < 237.189.227.249:5599 74 bytes 1 packets
192.168.237.130:80 < 37.124.224.22:5600 74 bytes 1 packets
192.168.237.130:80 < 227.229.20.205:5601 74 bytes 1 packets
192.168.237.130:80 < 34.166.78.195:5602 74 bytes 1 packets
192.168.237.130:80 < 131.1.148.247:5603 74 bytes 1 packets
192.168.237.130:80 < 46.19.51.57:5604 74 bytes 1 packets
192.168.237.130:80 < 234.224.150.27:5605 74 bytes 1 packets
```

Figure (4.18) UDP flood attack after applied iptables

Above Figure show attack report after applying iptables script to defend UDP flood attack, iptables command limit connection and drop attack traffic that result no outgoing ICMP "destination unreachable" packet so outgoing icmp pps attribute in report file is 0 that is contributed production server resource.



No.	Time	Source	Destination	Protocol	Length	Info
92	0.119222	82.127.2.214	192.168.237.130	UDP	74	11181 → 80 Len=32
93	0.119225	113.17.128.232	192.168.237.130	UDP	74	11182 → 80 Len=32
94	0.119227	17.13.83.165	192.168.237.130	UDP	74	11183 → 80 Len=32
95	0.119230	139.206.252.1	192.168.237.130	UDP	74	11184 → 80 Len=32
96	0.119232	13.121.186.206	192.168.237.130	UDP	74	11185 → 80 Len=32
97	0.119235	107.206.81.215	192.168.237.130	UDP	74	11186 → 80 Len=32
98	0.119237	244.78.78.229	192.168.237.130	UDP	74	11187 → 80 Len=32
99	0.119240	132.140.61.53	192.168.237.130	UDP	74	11188 → 80 Len=32
100	0.119242	127.41.137.205	192.168.237.130	UDP	74	11189 → 80 Len=32

```
▸ Frame 100: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
▸ Ethernet II, Src: Vmware_71:81:c0 (00:0c:29:71:81:c0), Dst: Vmware_9b:d9:b2 (00:0c:29:9b:d9:b2)
▸ Internet Protocol Version 4, Src: 127.41.137.205, Dst: 192.168.237.130
▾ User Datagram Protocol, Src Port: 11189, Dst Port: 80
    Source Port: 11189
    Destination Port: 80
    Length: 40
    Checksum: 0x96f1 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 99]
```

Figure (4.19) Sample trace UDP flood traffic after applied iptables

Above Figure show trace attack traffic after applied iptables, red box just show UDP packet request and not found outgoing ICMP "destination unreachable".

4.8 ICMP Flood Attack

The attack was made by flooding the victim's machine by running following Hping command from attacker machine as in figure (4.20) with parameters: -p 80 sends the packet to port 80 on victim machine, --flood flag sends the packet as fast as possible, --icmp flag sets the icmp mode.

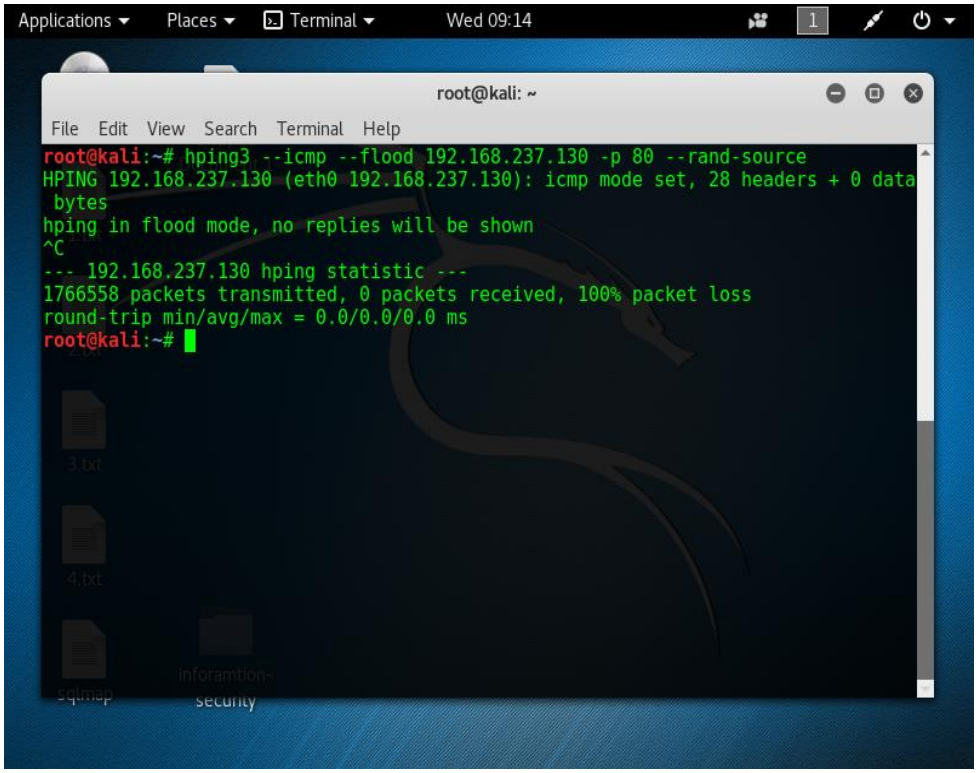


Figure (4.20) ICMP flood attack

Above Figure show how send ICMP flood attack by sending huge amount of ICMP echo requests packets to the victim with spoof IP addresses that‘s mean these packets request reply from the victim and this results in saturation of the bandwidth of the victim’s network connection and denial of service.

On victim machine, FNM monitoring and analysis incoming and outgoing traffic as in Figure (4.21) and a countable number of arriving packets per second If it exceeds threshold detection attack, trigger ban, generate attack report, notify to network administrator to send attack details as in Figure (4.22) and trace attack traffic to pcap file.

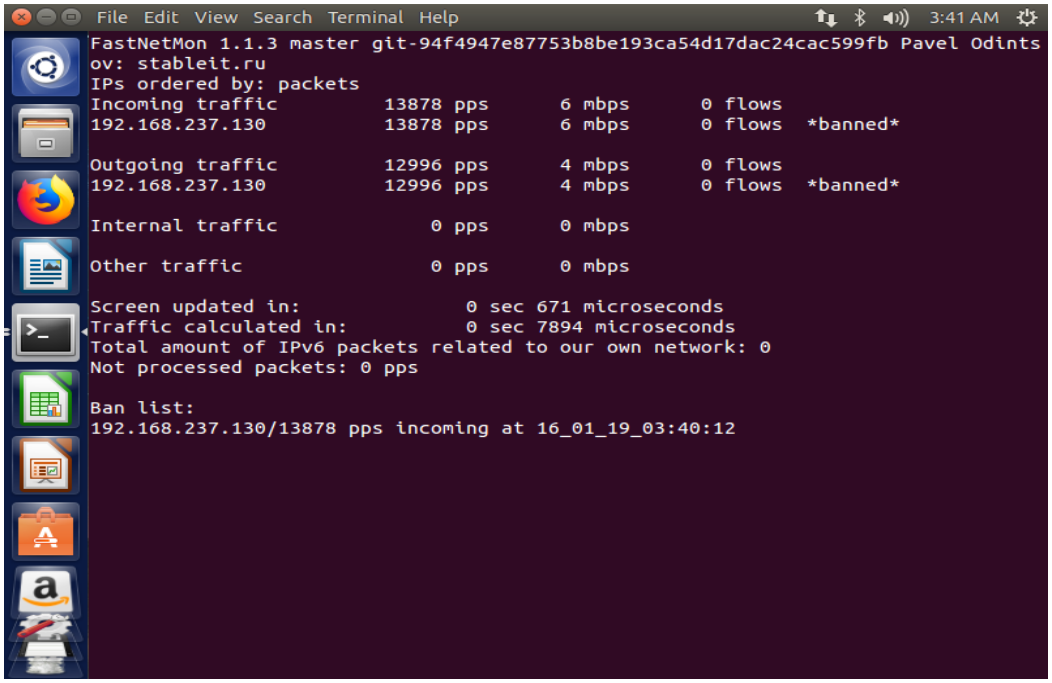


Figure (4.21) Statistic information of ICMP attack in fastnetmon_client

Above Figure show how the FNM monitoring victim then shows statistic information about ICMP attack traffic, when detected attack trigger system administrator and generated attack report file in Ban list with filename contains time of the detected attack (16_01_19_03:40:12).

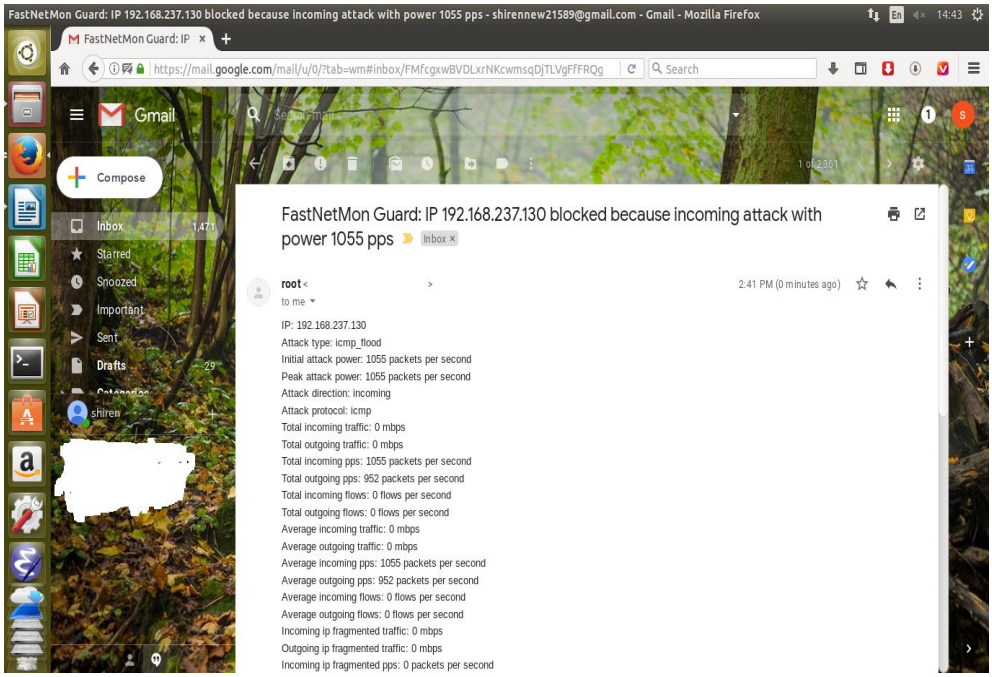


Figure (4.22) Email notify of ICMP attack

Above Figure show notification message was sent by FNM when detected attack, notification message show details information's about an attack like IP of the victim, attack type, initial attack power and total outgoing pps and other information's.

FNM generate attack report as in Figure (4.22), it shows details information: victim IP address, attack type is icmp_flood, initial attack power, direction attack and attack protocol. Also explain total incoming pps and icmp pps. Incoming icmp pps for send icmp echo request packet and victim replay send icmp echo response to forget source IP address for outgoing icmp pps also appear in attack trace pcap file as in Figure (4.23).

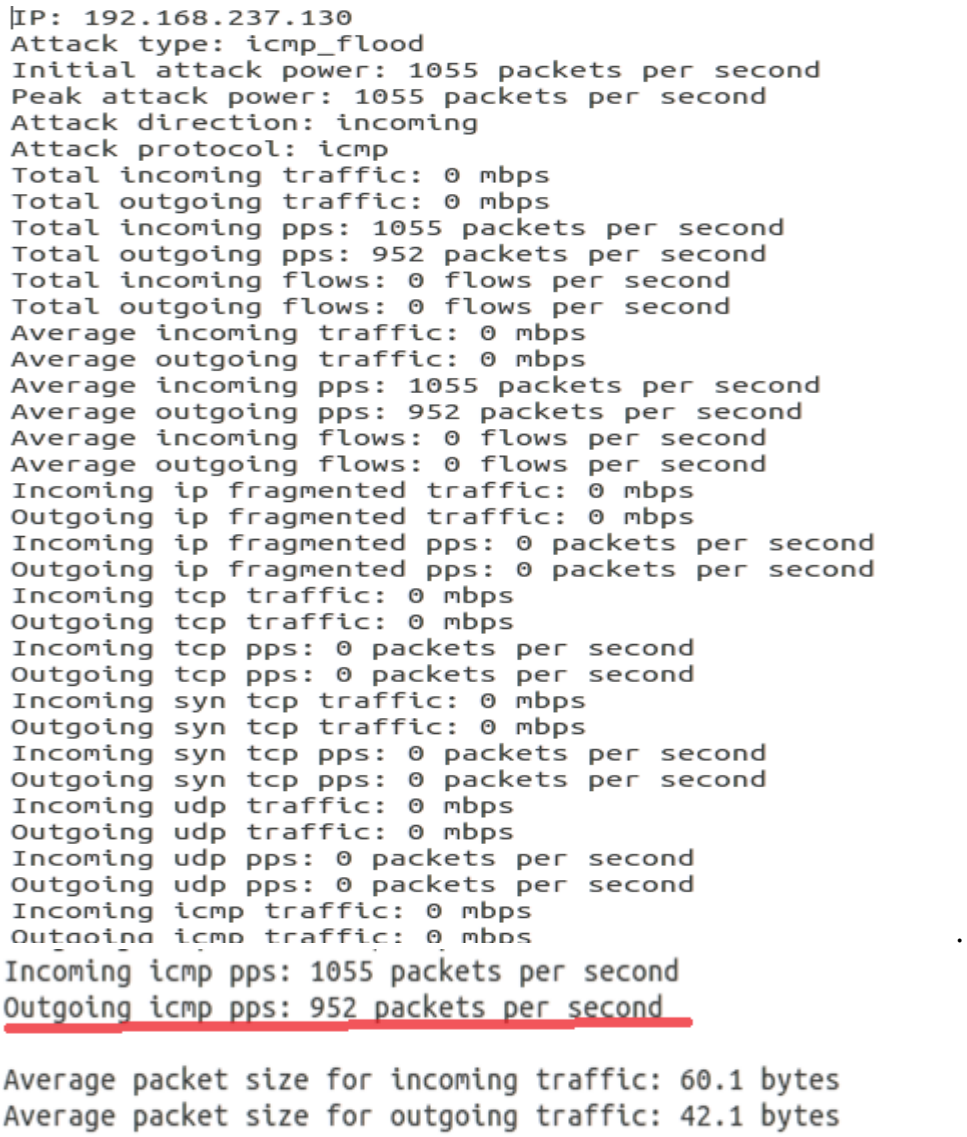


Figure (4.23) ICMP flood attack report

The above Figure show contains report file and details information about attack traffic like attack type and incoming icmp pps, outgoing pps is 952 represent ICMP echo reply response packets from victim to spoof IP addresses.

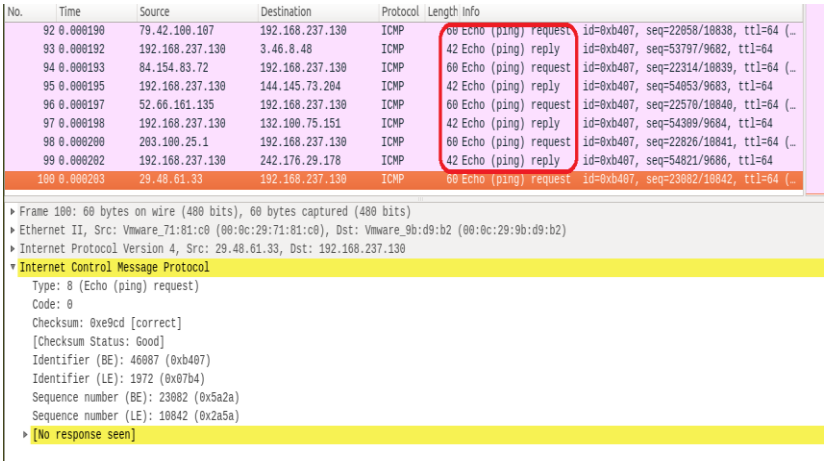


Figure (4.24) Sample trace ICMP flood traffic

Above Figure show trace attack traffic, red boxes show how the ICMP flood attack was happened when saw a huge amount of ICMP echo request packets with spoof IP addresses and outgoing ICMP echo replay packet.

To defend against ICMP Flood Attack, iptables script is writing as bellow:

iptables -N icmp_flood

iptables -A INPUT -p icmp -j icmp_flood

iptables -A icmp_flood -m limit --limit 1/s --limit-burst 3 -j RETURN

iptables -A icmp_flood -j DROP

All incoming connection are allowed till limit is reached, --limit 1/s is Maximum average matching rate in seconds, --limit-burst 3 is Maximum initial number of packets to match, FNM work again after applied script and generate attack report as in Figure 4.24 note for outgoing icmp pps and outgoing pps is 0, also in attack pcap trace file in Figure (4.25) attacker is sending ICMP Echo Request packets continuously but victim machine is

not responding by sending ICMP Echo Reply packets as all the packets are being dropped by the firewall according to the iptables rules.

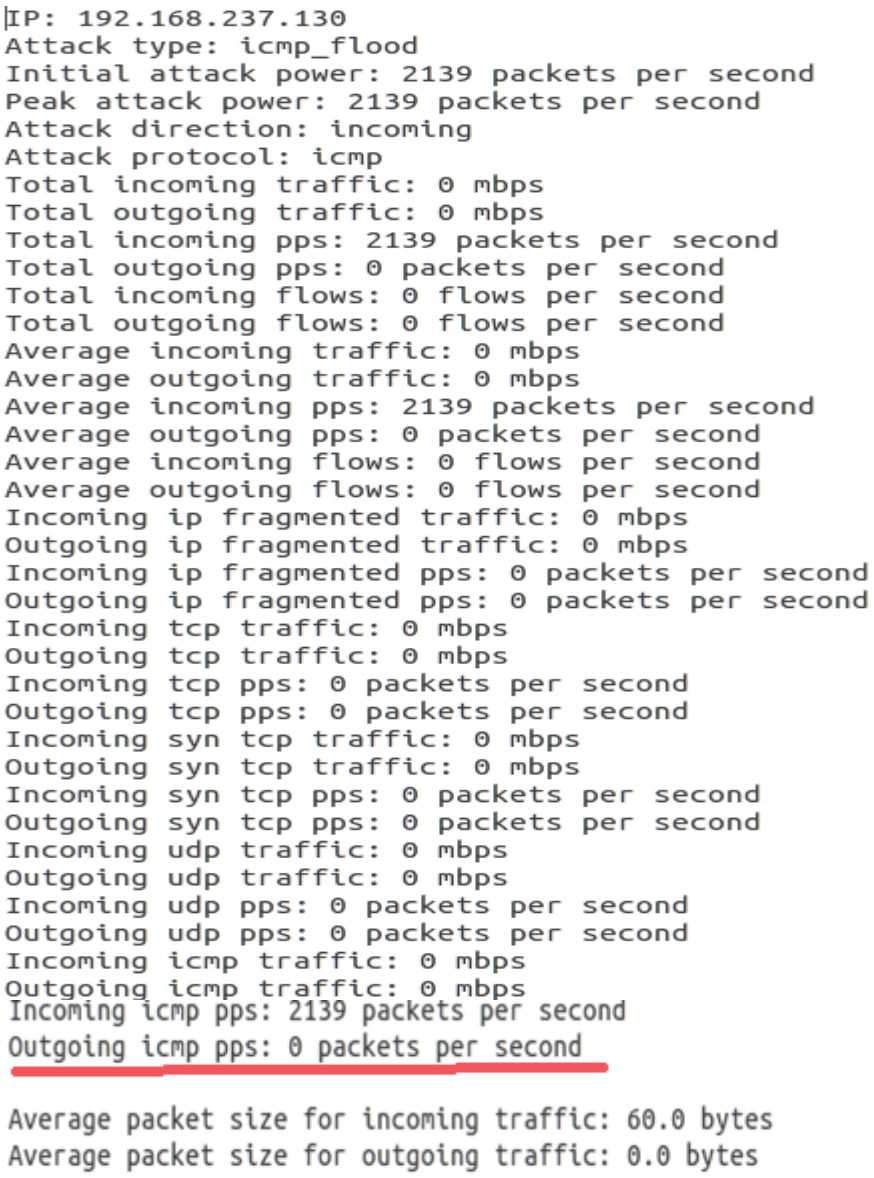


Figure (4.25) ICMP flood attack report after applied iptables

Above Figure show attack report after applying iptables script to defend ICMP flood attack, iptables command limit connection and drop attack traffic that result no ICMP echo reply packet response was generated from server to spoofed IP address so outgoing pps attribute in report file is 0 that is contributed production server resource.

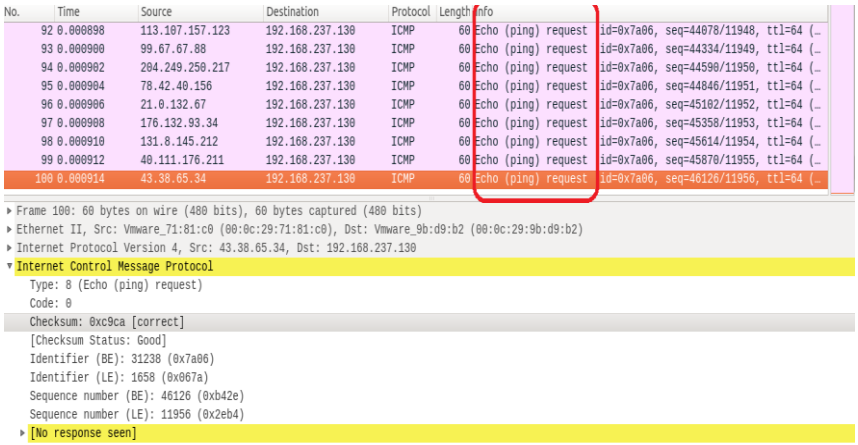


Figure (4.26) Sample track ICMP attack traffic after applied iptables

Above Figure show trace attack traffic after applied iptables just found ICMP echo request packets and not found outgoing ICMP echo reply packets response from server that represent in the red boxes because drop attack traffic by iptable.

4.9 Results Discussion

In a previous implementation work show how FNM detection DDoS based flooding Attack, generate reports and notify by email, also we using iptables to mitigate it and Contribute to prevention webserver victim resource and **continuous availability of Service**, when tested DDoS-based flood attack FNM detected attack and trigger system Administer email and generated attack report which contains detailed information about Attack, notice response packets from server to spoof IP addresses in outgoing pps Attribute which mean exhaust resource server, after that was used packet

filtering in Linux kernel by using an iptable script to filter attack traffic and drop and again tested DDoS-based flood attack and comparison outgoing PPS attribute, it is found zero so no Response packets will generate, the result of the tests shows when use FNM and iptable is More security to detection and mitigation DDoS-based flood attack in web server.

The point must be discussed is detection speed and generate report in FNM, Network Forensics For Detecting Flooding Attack On Web Server in this paper using Intrusion Detection System (IDS) Snort for detection flooding attack and record the Activities of the network in the form of log files with the extension pcap, Log files are Used at this stage of the investigation to the forensic process model method to find Evidence and work for generating report [14], Compare with this paper, we achieve to FNM provide forensic evidence of the flooding attack with high-performance to detect attacks with the extension pcap by capture attack traffic to pcap file and generate attack report contains detailed information about attacker and flow between them And send report to email notifying of a network administrator.

The other point must be discussed is Threshold, Analysis of UDP DDoS Flood Cyber Attack and Defense Mechanisms on Web Server with Linux Ubuntu 13, in this paper studied the impact of a UDP flood attack on the Web Server with the new generation Linux platform, namely, Linux Ubuntu 13, also evaluates the impact of various Defense mechanisms, including Access Control Lists (ACLs), Threshold Limit, Reverse Path Forwarding (IP Verify), and Network Load Balancing. Threshold Limit is found in be the most effective defense and FNM uses this defense mechanism So FNM is the best open source software tool for detection and mitigation DDoS based flooding attack with more features it is flexible and more security [16].

5. THE CONCLUSION ANDD FUTURE WORK

5.1 The Conclustion

Ensuring continuous availability of service has become imperative to the success Of any organization and prevent network from DDoS attacks is very important. The Complexity of DDoS attack makes detection and mitigation difficult, in this research, Efficient framework based on FNM open-source tool and iptable is proposed, FNM is use to detection DDoS-based flood attack (SYN, UDP, and ICMP) by configuration Anomaly detection (threshold) and packet capture in Linux kernel, when tested DDoS- Based flood attack FNM detected attack and trigger system administrator email and A generated attack report which contains detailed information about the attack, notice the response Packets from server to spoof IP addresses in outgoing pps attribute which mean exhaust Resource server, after that was used packet filtering in Linux kernel by using iptable script To filter attack traffic and drop and again tested DDoS-based flood attack and Comparison outgoing pps attribute, it is found zero, so no response packets will generate, The result of the tests shows when use FNM and iptable is more security to detect and Mitigating DDoS-based flood attack in web server.

5.2 Future Work

FNM has two versions: Community version, which is open to everyone and with Limited detection capability, while advanced or the commercial version supports advanced Detection and mitigation features, in this research using FNM community for how Detection and mitigation deals based flooding attack for future work development FNM Community version of enhancement actions after detection attack. For future work Development FNM community version to detection, remote attack and export attack report in xml format for web server.

REFERENCES

[1] C. Chellapan and H. Narasimhan and V. Varadarajan, An Investigation into the Detection and Mitigation of Denial of Service (DoS) Attacks, 9788132202769th ed., S.V. Raghavan and E. Dawson, Ed. India: Springer New Delhi Dordrecht Heidelberg London New York, 2011.

[2] Gayathri Gopalakrishnan, and Raj Sharman Manish Gupta, "Countermeasures against Distributed Denial of Service," ASIA'16, p. 7, June 8-9 2016.

[3] William Stallings, NETWORK SECURITY ESSENTIALS: APPLICATIONS AND STANDARDS, 4th ed., 2011.

[4] Sawan Patel2, Parag Somaiya3, Vishal Vishwanathan4 Mrunali Desai1, "Prevention of Distributed Denial of Service Attack using Web Referrals: A Review," International Research Journal of Engineering and Technology (IRJET), vol. 03, no. 04, p. 3, Apr 2016.

[5] Meklit Elfiyos Dekita. (2018) Experimental evaluation of DDoS detection and prevention using open source and commodity hardware.

[6] Bahaa Qasim M. AL-Musawi, "MITIGATING DoS/DDoS ATTACKS USING IPTABLES," International Journal of Engineering & Technology IJET-IJENS, vol. 12, no. 03, pp. 101-111, 2012.

[7] Mohammed Alenezi and M Reed, "Methodologies for detecting dos/ddos attacks against network servers," In Proceedings of the Seventh International Conference on Systems and Networks Communications—ICSNC, 2012.

[8] Hrishikesh Arun Deshpande, "HoneyMesh: Preventing Distributed Denial of Service Attacks using Virtualized Honeypots," International Journal of Engineering Research & Technology (IJERT), vol. 4, no. 08, p. 5, August 2015.

[9] Sagar Jadhav, Krushna Kudale, Sumaiyya Shaikh, Yogendra Patil Akash Naykude, "TBDA-Traceback Based Defence Against DDoS Attack," International Research Journal of Engineering and Technology (IRJET), vol. 03, no. 04, p. 5, Apr 2016.

[10] Kareem Ullah Muhammad Ahmad, "SNORT IMPLEMENTATION WITH WIRESHARK HELPING AVOIDING DDOS ATTACK IN THE CLOUD COMPUTING," International Journal of Computer Science and Information Security (IJCSIS), vol. 15, no. 1, p. 1, January 2017.

[11] Karanbir Singh and Kanwalvir Singh Dhindsa and Bharat Bhushan, "Distributed Defense: An Edge over Centralized Defense against DDos Attacks," Computer Network and Information Security, p. 9, march 2017.

[12] A. R. Bhagat Patil Archana.S. Pimpalkar, "Defense against DDoS Attacks Using IP Address Spoofing," International Journal of Innovative Research in Computer, vol. 3, no. 3, p. 8, March 2015.

[13] Kiran V K Soumya Suresh, "Prevention of Dos and DDoS Attack Using Cryptographic Techniques," International Journal of Advanced Research in Computer and Communication Engineering IJARCCE, vol. 5, no. 1, p. 4, Feburary 2016.

[14] DESTI MUALFAH and IMAM RIADI, "Network Forensics For Detecting Flooding Attack On Web Server," (IJCSIS) International Journal of Computer Science and Information Security, vol. 15, no. 02, pp. 326-331, February 2017.

[15] D.Deepthi Rani and T.V.Sai Krishna and G.Dayanandam and Dr.T.V.Rao, "TCP Syn Flood Attack Detection And Prevention," International Journal of Computer Trends and Technology (IJCTT), vol. 4, no. 10, pp. 3412-3417, Octber 2013.

[16] Kiattikul Treseangrat, Bahman Sarrafpour Samad S. Kolahi, "Analysis of UDP DDoS Flood Cyber Attack and Defense Mechanisms on Web Server with Linux Ubuntu 13," p. 5.

[17] Wikipedia. [Online]. https://en.wikipedia.org/wiki/VMware_Workstation

[18] wikipedia. [Online]. https://en.wikipedia.org/wiki/Kali_Linux

[19] wikipedia. [Online]. https://en.wikipedia.org/wiki/Ubuntu