A HYBRID APPROACH TO FILTER AND TRACEBACK IP-SPOOFED PACKETS IN DDoS ATTACKS

A DISSERTATION

Submitted in partial fulfillment of the requirements for the award of the degree of

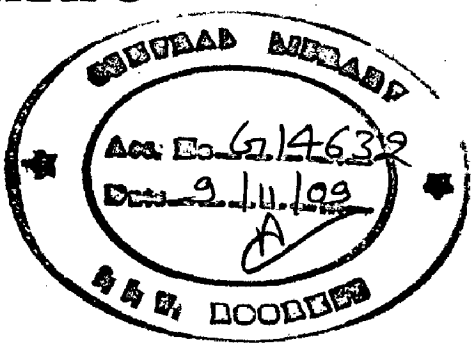
MASTER OF TECHNOLOGY

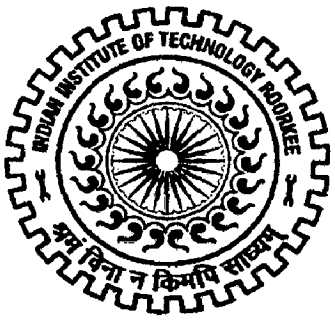
in

COMPUTER SCIENCE AND ENGINEERING

By

KRISHNA PARACHIKAPU





DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE -247 667 (INDIA)

JUNE, 2009

CANDIDATE'S DECLARATION

I hereby declare that the work, which is being presented in the dissertation entitled **"A HYBRID APPROACH TO FILTER AND TRACEBACK IP-SPOOFED PACKETS IN DDoS ATTACKS"** towards the partial fulfillment of the requirements for the award of the degree of **Master of Technology** in **Computer Science and Engineering**, submitted in the Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, Roorkee (India) is an authentic record of my own work carried out during the period from July 2008 to June 2009, under the guidance of **Dr. A. K. Sarje, Professor, Department of Electronics and Computer Engineering, IIT Roorkee.**

I have not submitted the matter embodied in this dissertation for the award of any other degree or diploma.

Date: 15-06-2009

Place: Roorkee.

P. Krishna

(KRISHNA PARACHIKAPU)

CERTIFICATE

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Date: 15/6/2009

Place: Roorkee.

(Dr. A. K. Sarje)

Professor,

Department of Electronics and Computer Engineering,

IIT Roorkee, Roorkee – 247 667.

ACKNOWLEDGEMENTS

I would like to take this opportunity to extend my heartfelt gratitude to my guide and mentor **Dr. A. K. Sarje,** Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, for his trust in my work, his able guidance, regular source of encouragement and assistance throughout this dissertation work.

I also wish to thank Indian Institute of Technology Roorkee for giving me this opportunity.

Most importantly, I would like to extend my deepest appreciation to my family for their love, encouragement and moral support. Finally I thank God for being kind to me and driving me through this journey.

KRISHNA PARACHIKAPU

ABSTRACT

Today, Internet is the prime medium for communication and is the most sought after service by innumerable amount of users across the globe. At the same time, its commercial nature is causing increasing vulnerability to cyber crimes and there has been an enormous increase in the number of DDoS attacks on the internet over the past decade. Network resources such as network bandwidth, web servers and network switches are mostly the victims of many attacks.

Current internet architecture allows the attacker to spoof the source address of the IP packet by rewriting the packet header. This gives provision to conceal the identity of the source of attack. IP spoofing is the most popular form of Distributed Denial of Service attack. A large number of schemes have been proposed and implemented for the defense against DDoS attacks. Some defend the attack by filtering and dropping packets and some defend the attack by tracing back to the source of attack after experiencing it. Both the mechanisms have their own drawbacks.

In this dissertation "A HYBRID APPROACH TO FILTER AND TRACEBACK IP-SPOOFED PACKETS IN DDoS ATTACKS", we propose a hybrid packet marking mechanism to overcome the short comings of the above methods. This mechanism filters out the IP-spoofed packets and simultaneously traces back to the source of attack. Two different mechanisms for traceback are developed, one is using Bloom filters and another is by using Probabilistic Packet Marking strategy. In the proposed strategy, packet marking can be done at intermediate routers. Each packet is marked in two areas where one mark is used in filtering process and another is used in traceback process.

The proposed scheme has been simulated using JAVA based JIST (Java In Simulation Time) Simulator. Various test cases have been thought of, for which simulations were performed by varying different parameters. The results are compared with the existing schemes and finally, conclusions are presented.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1. Introduction

Communication is the key in changing the way the world looks, thinks, and works. Hardwired telex and telephone were the only source for communication before computers. Defense Advanced Research Projects Agency (DARPA) introduced the Advanced Research Projects Agency Network (ARPANET) to provide convenient sharing of specialized computing resources across different defense institutions. A significant landmark was the introduction of TCP/IP as a set of protocols for internetworking which created the nucleus of the Internet [1]. Internet was commercialized as the TCP/IP standard which was adopted by hardware and software vendors allowing organizations to interconnect heterogeneous systems. The mass production of Internet-capable personal computers and an unprecedented growth in the number of Internet Service Providers (ISP) enabled Internet to be commonly accessible to everyone.

Today, the Internet is an essential part of our everyday life and many important and crucial services like banking, shopping, transport, health, and communication are partly or completely dependent on the Internet. Similarly it has also enhanced the criminal's ability to perform unlawful or unethical activities. Thus, any disruption in the operation of the Internet can be very inconvenient for most of us. Recent trends in the Internet [2, 3] show that, at some point the Web sites were getting up to 50,000 fake hits per second from illegitimate machines and the total amount of the DDoS attacks reached over 40 gigabit per second barrier. It also shows that the amount of DDoS attack traffic has been increasing in size (doubled) year by year. Figure 1.3 shows the pattern of attack traffic year wise.

An attempt to make a computer resource or a service unavailable to its intended users is called as Denial of Service (DoS). Denial of Service in its distributed form is called Distributed Denial of Service (DDoS). In DDoS attacks, the attacker first takes control of a large number of vulnerable hosts on the internet by compromising

them. The attacker then uses those hosts to simultaneously send a huge number of packets to the victim, thereby exhausting all of the victim's resources. During DDoS attack, massive amounts of traffic arrive at the target of attack (i.e., victim). This target is either the network service or the network itself. Due to the huge amount of traffic, the computational overhead increases on the victim and the victim services get disrupted. The sole purpose of the DDoS attacks is to reduce or eliminate the availability of a service provided over the Internet, to its legitimate users. Figure 1.1 illustrates an example of how DDoS attack happens.

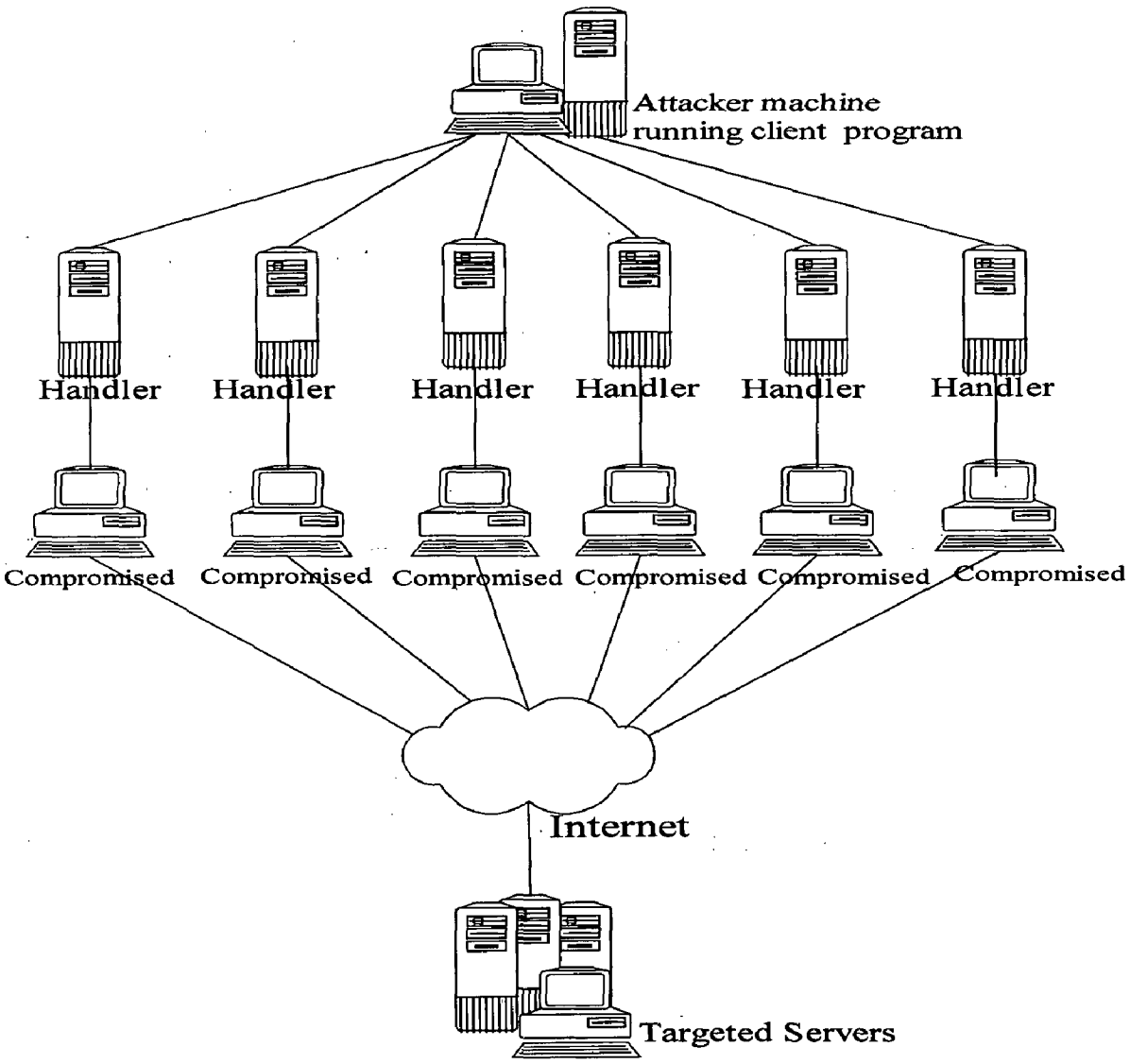


Figure 1.1: An example scenario of DDoS Attack

Current Internet architecture allows the attacker to spoof source address of the IP packet by rewriting the packet header. This process is called IP spoofing and it gives the provision to conceal the identity of the source of attack. IP spoofing is usually employed in conjunction with DDoS attacks in the Internet. In present Internet

environment DDoS Attack [4, 5] is a serious security problem and it causes severe damages on the targeted servers.

1.2. Motivation

The usage of internet is growing day by day. According to the recent sources [6, 7] the number of hosts connected to the Internet has increased to almost 550 million and there are currently more than 1.5 billion users of the internet. Also there has been an increase in the number of cyber crimes worldwide. A large number of security breach incidents are affecting many organizations and individuals. The crimes committed are also becoming more and more sophisticated. Law enforcement is in a perpetual race with the cyber criminals to ensure that they are in a level playing field.

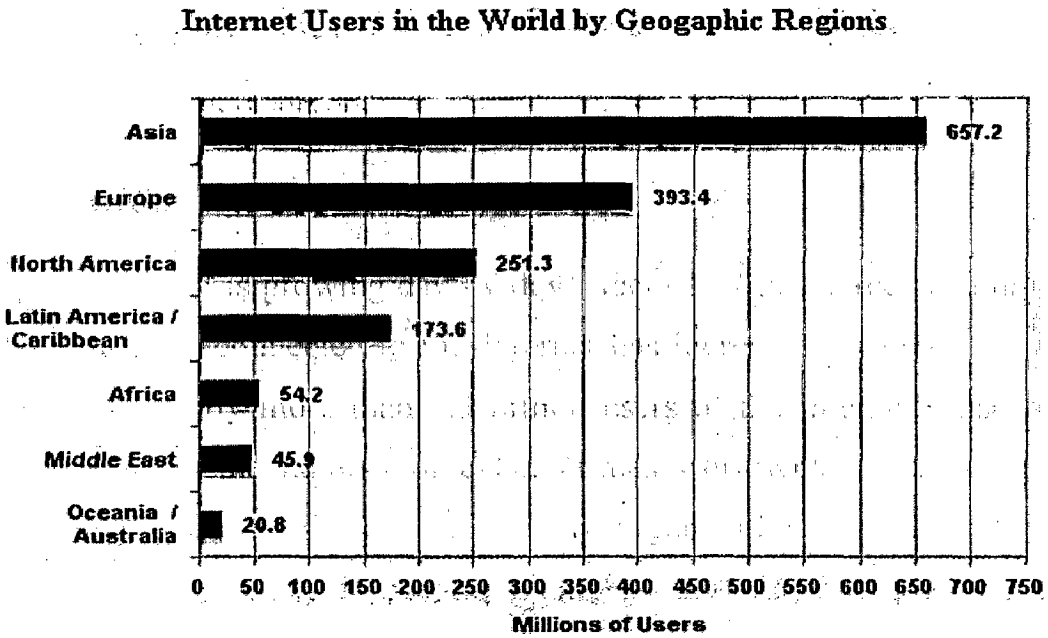


Figure1.2: Estimated Internet Users in the World by Geographic regions according the survey of Mini Watts Marketing Group

Figure 1.2 shows the Size of the Internet users in the world by various Geographic Regions. This is the recent information according to the survey of Mini Watts Marketing Group [7]. According to this survey, the estimated Internet users are 1,596,270,108 for March 31st 2009.

A recent study conducted by Arbor Networks [2] shows the year by year increase of the DDoS attack traffic on the Internet, from the year 2001 to 2007 [Figure 1.3].

This indicates that DDoS attack traffic size (in gigabits-per-second) nearly doubled in year 2007 from the year 2006.

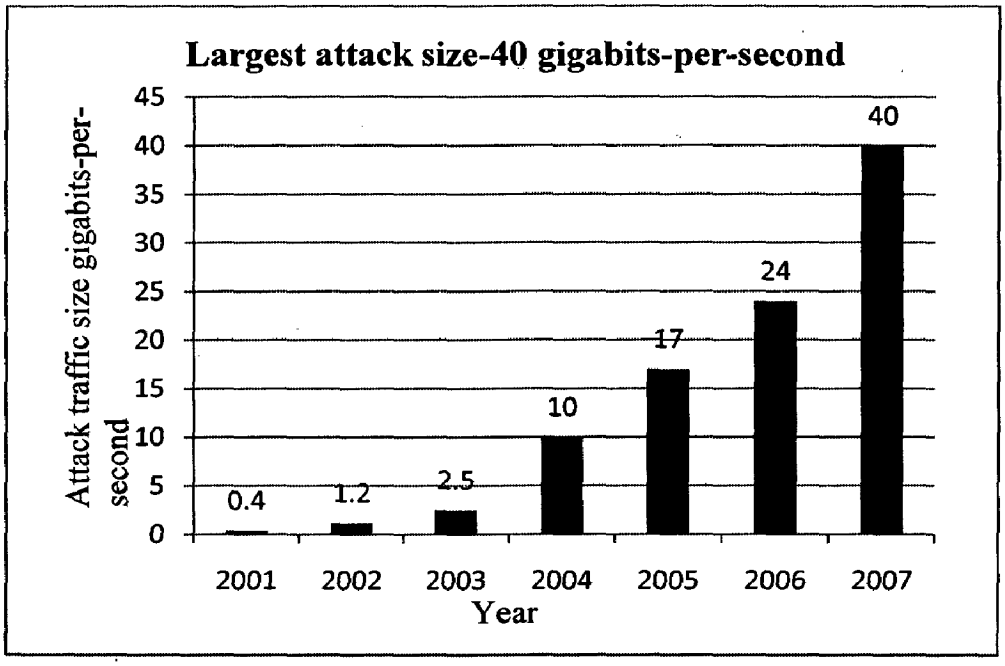


Figure1.3: Study: DDoS attacks threaten ISP infrastructure

Various methodologies and frameworks have been proposed to defend DDoS attacks. They are limited in different aspects, like computational overhead on the intermediate routers, limited resources like memory, false positives. The challenge is to improve on these aspects and give a comprehensive solution.

There are two ways of defending the DDoS attacks. One way is to find out the attack packets and filter those packets using firewalls. The other way is to find out the attack source and take punitive action against them in order to avoid further attacks. The existing methods have the features of only one type of defending mechanism, i.e., either filtering the attack packets or tracing back to the source of attack.

Motivation of our work is to develop an efficient and hybrid mechanism to defend the DDoS attacks, in which we can simultaneously filter and traceback IP spoofed packets.

1.3. Problem Statement

In this dissertation, we propose a hybrid packet marking mechanism which promises defense against DDoS attacks in a robust way. This mechanism covers the following activities simultaneously:

(i) Filter IP spoofed packets and

(ii) Trace back to the source of attack by using Bloom filters and Probabilistic Packet Marking.

1. 4. Organization of the Report

The rest of the report is organized as follows. Chapter 2 discusses about different types of DDoS attacks and different mechanisms to defend against them. In Chapter 3, the existing mechanisms and their limitations to defend the DDoS attacks are briefly discussed. Chapter 4 briefly describes about the proposed method and discusses the implementation of our proposed mechanism. In Chapter 5, simulation results are presented which evaluate our mechanism. Finally Chapter 6 compares our mechanism with existing mechanisms, concludes our work and also give the suggestions for future work.

Chapter 2

DDoS ATTACKS AND DEFENSE APPROACHES

In this Chapter, we first list out the draw backs of TCP/IP protocol suite [8] which facilitates IP spoofing. Next we briefly look into types of IP spoofing. Finally, we move on to describing about different kinds of DDoS attacks which uses IP spoofing and the different defense mechanisms to guard against them.

2.1. Security Problems in TCP/IP Protocol

The following features of TCP/IP protocol suite facilitate various types of attacks.

- It is very easy to mask a source IP address by manipulating an IP header.
- Sequence number prediction, which leads to session hijacking or host impersonating.

In order to understand the above statements clearly, we need to first have a look into the IP and TCP protocol structure which are discussed below:

2.1.1. IP Protocol

This protocol operates at layer 3, i.e., Network layer of the OSI model. Because of its connection less property there is no information regarding its transaction state. Figure 2.1 shows the IP header of the IP packet.

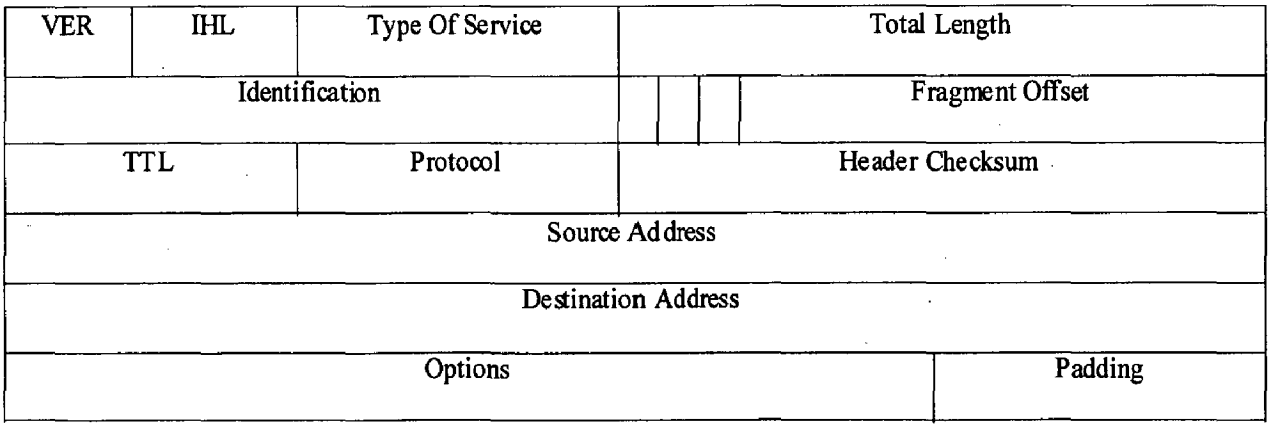


Figure 2.1: IP Packet Header

The IP header contains the fields for source address and destination address. Using one of several tools, (Trinoo [17], Tribe Flood Network (TFN) [32], Shaft [33]) an attacker can easily modify the addresses, specifically the source address field and sends the packets to victim. It is important to note that each datagram is sent independent of all others due to the stateless nature of IP.

2.1.2. TCP Protocol

IP can be thought of as a routing wrapper for transport layer, which contains the Transmission Control Protocol. Unlike IP, TCP uses a connection oriented design. This means that the participants in a TCP session must first build a connection via three-way handshake (SYN_SYN/ACK), and then update one another on progress via sequence numbers and acknowledgement numbers. Figure 2.2 shows the TCP packet header.

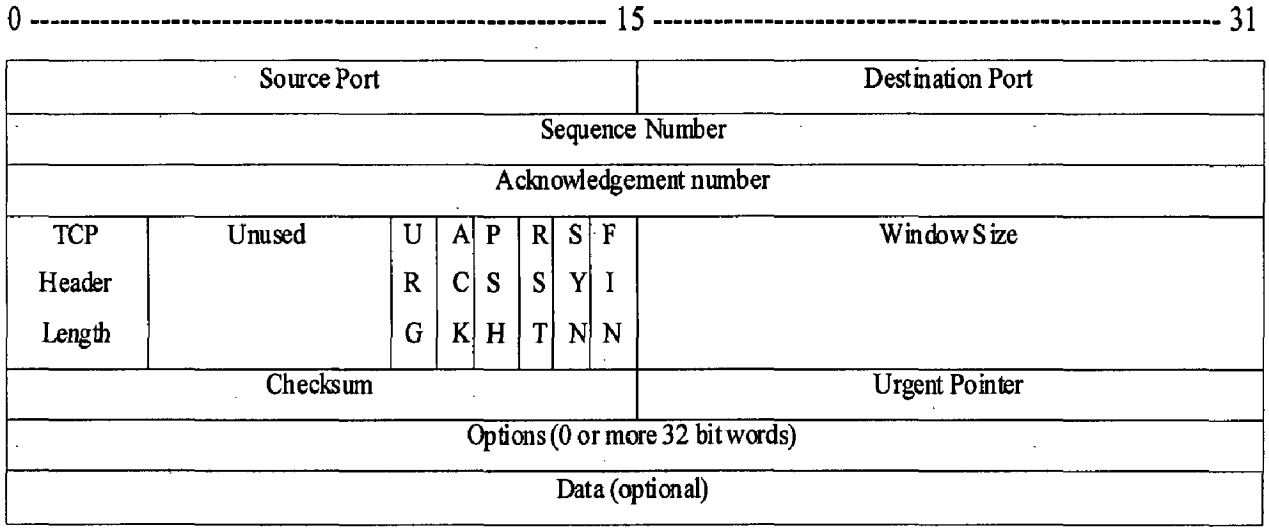


Figure 2.2: TCP Packet Header

We can observe that TCP header is different from IP header where first 12 bytes consists of port number and sequencing information. Much like an IP datagram TCP packets can be manipulated using software. The source and destination port numbers normally depend on the network application in use.

The sequence number is the number of the first byte number in the current packet which is relevant to the data stream. The acknowledgement number in turn contains the value of the next expected byte or the sequence number in the stream.

➢ TCP Sequence Number Prediction

The most fascinating hole in the TCP is the sequence number prediction [8]. The Normal TCP connection can be established by the process of three-way handshake. In this the client selects and transmits initial sequence number (ICSN) to the server, the server acknowledges it and sends back its own sequence number ISSN (Initial Server Sequence Number), and then the client acknowledges it.

The steps in Three-way handshake are shown below. First the client sends an initial sequence number by setting SYN bit. When the server receives this packet it responds back to client by sending its own initial sequence number and acknowledgment to the client's sequence number. Then the client completes the connection by acknowledging the sequence number from the server.

Client → Server: SYN (ICSN i.e. initial sequence number from client)

Server → Client: SYN (ISSN)/ACK (ICSN) i.e. initial sequence number from server

Client → Server: ACK (ISSN)

Client → Server: data

And/or

Server → Client: data

By predicting (i.e., analyzing the traffic) an intruder can find out the sequence number, constructs a packet and spoof a trusted host on a local network.

The following shows how an intruder hijacks the session of a server. Here we consider T as a trusted host, and X as intruder.

Intruder → Server: SYN (IXS), SRC = T

Server → T: SYN (ISSN), ACK (IXSN)

Intruder → Server: ACK (ISSN), SRC = T

Intruder → Server: ACK (ISSN), SRC = T, nasty – data

First intruder X sends a connection request to the server by sending an initial sequence number (IXSN) by impersonating as a trusted host T. Then the server

acknowledges the IXSN and sends an initial sequence number (ISSN) to client T. Then intruder X acknowledges to ISSN as the trusted host. In this the server actually doesn't receive any message from trusted host T.

Even though the message *Server* → T does not go to *Intruder*, *Intruder* was able to know its contents, and hence could send data.

One way of preventing the prediction of sequence numbers is to use a cryptographic algorithm for initial sequence number (ISSN) generation.

The above drawbacks in TCP/IP protocol suite allow the attacker to spoof different fields of the IP packet and TCP packet. The attacker takes the advantage of these and freely attacks the networks resources. The next section deals with the various spoofing attacks.

2.2. Types of Spoofing Attacks

The process of modifying (forging) the source IP address of the IP packet is called source IP address spoofing or simply IP spoofing. By spoofing the source address of the packet with a genuine source address, the attacker gains an unauthorized access to a computer or network.

We can categorize the spoofing attacks [9] broadly into four types, based on the way the attacker spoofs the packet header.

1. Blind spoofing
2. Non-blind spoofing
3. Man in the middle attack
4. Denial of service attack

When spoofing the packet, the attacker can spoof the packet in one of the two ways. He can either

(i) Select the source IP address specifically or randomly

(ii) He can predict the sequence numbers blindly or specifically.

2.2.1. Blind Spoofing

In this type of spoofing, the attacker sends several packets blindly to predict the sequence numbers. This is a more sophisticated attack, because the sequence numbers and acknowledgement numbers are unreachable. In order to circumvent this, several packets are sent to the target machine in order to sample sequence numbers.

2.2.2. Non-Blind Spoofing

In this the attacker predicts the sequence number of the packet accurately. This type of attack takes place when the attacker is on the same subnet as the victim. The sequence number and acknowledgement numbers can be sniffed, eliminating the potential difficulty of calculating them accurately. The biggest threat of spoofing in this instance would be session hijacking. This is accomplished by corrupting the data stream of an established connection, then re-establishing it based on correct sequence and acknowledgement numbers with the victim machine.

2.2.3. Man in the Middle Attack

In these attacks, a malicious party intercepts a legitimate communication between two friendly parties. The malicious host then controls the flow of communication and can eliminate or alter the information sent by one of the original participants without the knowledge of either the original sender or the recipient. In this way, an attacker can fool a victim into disclosing confidential information by "spoofing" the identity of the original sender, who is presumably trusted by the recipient.

2.2.4. Denial of Service Attack

In first three attacks, the attacker tries to establish the connection with the victim. But in DoS attack, the attacker doesn't worry about establishing the connection with the victim, that's why most of the times they spoof the source IP address with a random IP address.

In this type of attacks, attacker concerns only with consuming bandwidth and resources and they need not worry about properly completing handshakes or connections. Rather, they wish to flood the victim with as many packets as possible

in a short amount of time. In order to prolong the effectiveness of the attack, they spoof source IP addresses to make it difficult to trace back the packets.

A Denial-of-Service attack is characterized by an explicit attempt by attackers to prevent legitimate users of a service from using that service. The main aims of these attacks are:

1. Consumption of scarce, limited, or non-renewable resources
2. Destruction or alteration of configuration information
3. Physical destruction or alteration of network components

Now, we move on to Distributed Denial of Service (DDoS) attacks and their different types in the following section.

2.3. DDoS Attacks

A Distributed Denial of Service (DDoS) attack is one in which a multitude of compromised systems attack a single target, thereby causing denial of service for users of the targeted system. The flood of incoming messages to the target system essentially forces it to shut down, thereby denying service to its legitimate users. In DDoS attacks, the attacker always tries to modify the source address of the attack packets to conceal the identity of the attack, and makes it difficult to distinguish such packets from legitimate packets.

We can classify the DDoS attacks in to two main categories. They are:

1. Flood attacks
2. Software attacks

2.3.1. Flood Attacks

In flood attacks the victim is overwhelmed by a continuous flood of traffic designed to consume resources at the targeted server (CPU cycles and memory) or in the network (bandwidth and packet buffers). These attacks result in degraded service or a complete site shutdown. TCP SYN Flood attack, Smurf attack, and UDP flood attack fall into this category.

➢ **TCP SYN flooding attack**

Taking advantage of the flaw of TCP three-way handshaking behaviour, an attacker makes connection requests aimed at the victim server with packets with unreachable source addresses by spoofing the source address of the packets. Because of the spoofed address the server not able to complete the connection requests and, as a result the victim wastes all of its resources.

➢ **Smurf attack**

An attacker sends forged ICMP echo packets to broadcast addresses of vulnerable networks. All the systems on these networks reply to the victim with ICMP echo replies. This rapidly exhausts the bandwidth available to the target, effectively denying its services to legitimate users.

➢ **UDP flood attack**

UDP is a connectionless protocol and it does not require any connection setup procedure to transfer data. A UDP flood attack is possible when an attacker sends a UDP packet to a random port on the victim system. When the victim system receives a UDP packet, it will determine what application is waiting on the destination port. When it realizes that there is no application that is waiting on the port, it will generate an ICMP packet of destination unreachable to the forged source address. If enough UDP packets are delivered to ports on victim, the system will go down.

2.3.2. Software Attacks

These attacks are relatively easy to counter either through the installation of software patches that eliminate the vulnerabilities or by adding specialized firewall rules to filter out malformed packets before they reach the target system. Ping of death, Teardrop and Land attacks fall into this category.

➢ **Ping of death attack**

In this attack, an attacker sends an ICMP ECHO request packet that is much larger than the maximum IP packet size to victim. Since the received ICMP echo request

packet is bigger than the normal IP packet size, the victim cannot reassemble the packets. The OS may be crashed or rebooted as a result.

- **Teardrop attack**

In this attack, an attacker sends two fragments that cannot be reassembled properly by manipulating the offset value of packet and cause reboot or halt of victim system.

- **Land attack**

Here, an attacker sends a forged packet with the same source and destination IP address. When this type of packets appeared at the victim machine, it confuses and crashes or hangs out.

This completes the discussion about DDoS attacks. We now present the different defence mechanisms employed to counteract these attacks. They are described in the following section.

2.4. Defense Approaches

Several mechanisms [10-15] exist to defend against IP spoofed DDoS attacks. DDoS attacks make the computer resources unavailable to the legitimate users and give unauthorized access to a malicious user. In order to fulfill the requirements or the services of legitimate users, the victim must defend the attacks. These defense approaches are broadly divided into three categories.

2.4.1. Preventive Defense

Preventive schemes aim at improving the security level of a computer system or network, thus preventing attacks from happening, or enhancing the resistance to attacks. Proactive server roaming [10], Ingress filtering [16] and Outgress filtering fall into this category.

- **Ingress filtering**

In this filtering technique, the boarder router filters the incoming packets, which comes with the source IP address, belongs to the same autonomous system domain. To do this the boarder router is configured with an access control list (ACL) that blocks the private addresses on the downstream interface.

The principal problem with ingress filtering is that its effectiveness depends on widespread, if not universal, deployment. Unfortunately, a significant fraction of ISPs, perhaps the majority, do not implement this service either because they are uninformed or have been discouraged by the administrative burden, potential router overhead or the boarder router may be in the hands of private organizations or attackers who intentionally supports attack environment. A secondary problem is that even if ingress filtering were universally deployed at the customer-to-ISP level, attackers could still forge addresses from the hundreds or thousands of hosts within a valid customer network.

- **Outgress or Egress filtering**

In this filtering technique the border router filters out the outgoing packets which are going with a source IP address that does not belongs to the same Autonomous System domain. Here the border router is configured with an access control list that blocks the outgoing packets with the source IP address, that doesn't belongs to the same autonomous system domain. It is also having the same problem like ingress filtering as mentioned above.

2.4.2. Source Tracking

Source tracking[11, 12, 13] mechanisms aim to track down the source of attacks, so that punitive action can be taken against the attackers and further attacks can be avoided. Probabilistic Packet Marking (PPM) [13], Deterministic packet Marking (DPM) [11, 12], Message Traceback [14], and logging [20] fall into this category.

The idea behind packet marking mechanism is to encode the path information inside the IP packet. This idea was first proposed by Savage et al. in Probabilistic Packet Marking [13], in which the routers insert path information into the Identification field of IP header. In this method, each packet is marked with certain probability, such that the victim can reconstruct the attack path using these markings and thus track down the sources of offending packets.

In the message traceback method [14, 18], routers generate ICMP traceback messages for some of received packets and send with them. By combining the ICMP packets with their TTL differences, the attack path can be determined.

In logging method [20] each router records the packet information from which it has traversed. This information is used later in reconstructing the path to the attacker.

A common problem existing in all the above solutions is that the reconstruction of attack path becomes quite complex and expensive when there are a large number of attackers. These types of solutions for defending DDoS attacks are designed to traceback to the source of attack and to take corrective actions against them after an attack has happened and cannot be used to stop the ongoing DDoS attack.

2.4.3. Reactive Solutions

Reactive solutions aim at improving the security of the computer system or network by detecting an ongoing attack and react to it by controlling the flow of attack by mitigating its effects. Pushback [21] method, filtering method [22] and other method proposed by Yaar et al [15] fall into this category.

The Pushback [21] method generates an attack signature based on the pattern of dropped packets after detecting congestion, and applies a rate limit on corresponding incoming traffic. This information is then propagated to upstream routers, and the routers help to drop such packets, so that the attack flow can be pushed back.

Of the above methods, source tracking and reactive solutions basically use packet marking mechanisms. In packet marking mechanism, the packets can be marked by intermediate routers while they are traversing from source to destination. Later, this marked information can be used in detecting the attack, filtering and tracing back to the source of IP spoofed packets. Packet marking schemes give best results in defending the DDoS attacks, because these mechanisms filter out the packets or traceback the packets to the source of attack, based on the marked information.

This finishes this chapter on DDoS attacks and their defense approaches. The next chapter reviews the existing techniques introduced on defense mechanisms by various authors.

Chapter 3

LITERATURE REVIEW

This chapter gives an extensive literature survey on the existing defense mechanisms. It also presents an analysis regarding which of the methods gives the best results. The limitations of the existing techniques are also described. A concept known as Bloom Filters introduced by B.H. Bloom in [19] has been discussed briefly in Section 3.2 which helps us in understanding of our proposed method in chapter 4.

3.1. Existing methods

The existing mechanisms to defend against DDoS attacks can be broadly divided into two classes. The first class of methods defend the attack by filtering the packets where the attacker cannot be traced and punished and second class of methods defend the attack by tracing back to the source of attack and take punishable actions on the attacker in order to avoid further attacks. In the latter method the victim has to experience the attack. We briefly look in to each class of mechanisms in the following sections.

3.1.1. Filtering Mechanisms

The existing mechanisms for filtering the IP-spoofed packets use packet marking mechanisms, except Hop-count filtering.

Wang et al. [23] proposed a mechanism to filter out the spoofed packets based on the number of hops the packet traverses. This technique uses a mapping between IP address and hop-counts, and the victim uses this mapping to distinguish spoofed IP packets from legitimate ones. The drawback of this mechanism is that the hop-counts are calculated based on the value of TTL field, where the attacker may spoof this field too.

The method proposed by Yaar et al. [15] uses the idea of packet marking for filtering out the attack packets instead of trying to find the source of such packets.

This scheme uses a path identifier (called Pi) to mark the packets; the Pi field in the packet is separated into several sections and each router inserts its marking in to one of these. Once the victim has known the marking corresponding to attack packets, it can filter out all such packets coming through the same path.

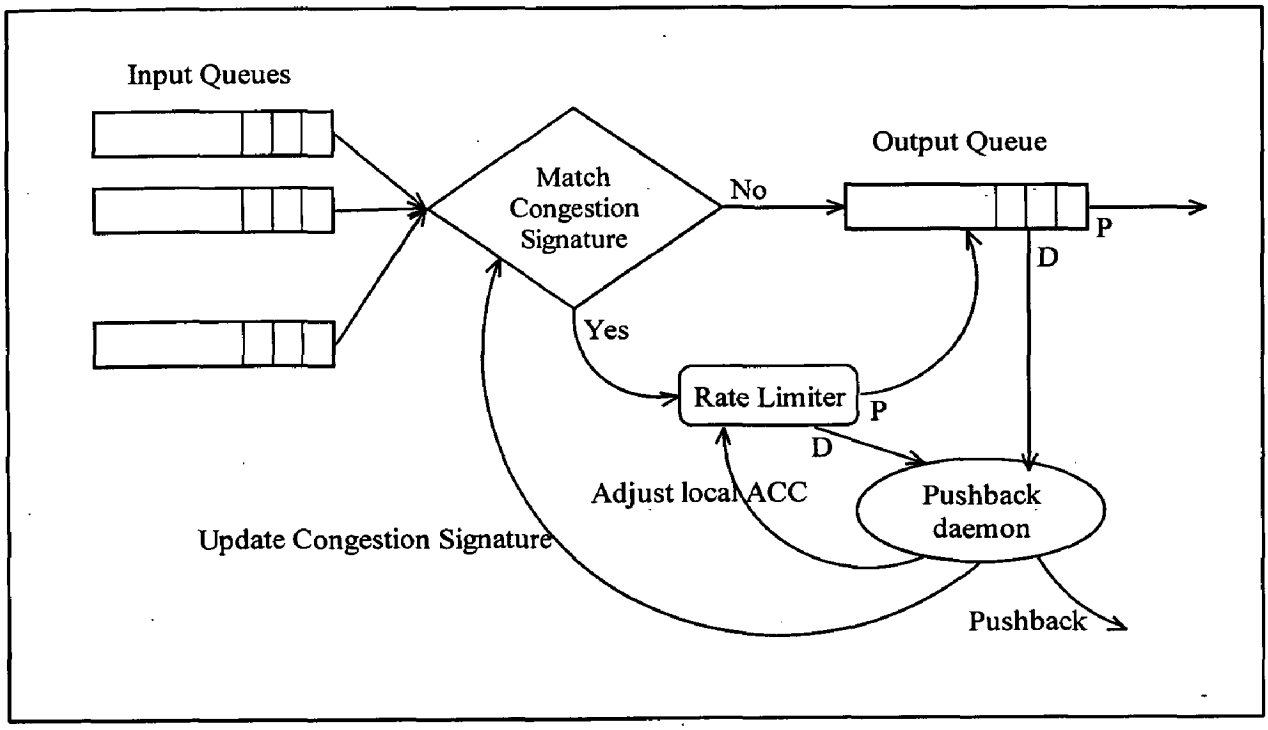


Figure 3.1: Partial view of a router which supports pushback mechanism.

John et al. [21] proposed a mechanism to filter the packets at the intermediate routers by generating the attack signatures based on the pattern of dropped packets after detecting congestion, and applies a rate limit on corresponding incoming traffic. This information is then propagated to upstream routers, and the routers help to drop such packets, so that the attack flow can be pushed back. . this functionality is added to each router to detect and preferentially drop packets that probably belong to an attack. Upstream routers are also notified to drop such packets (hence the term Pushback) in order that the router's resources be used to route legitimate traffic. There are two techniques here: local Aggregate Congestion Control (ACC) and pushback. Local ACC detects the congestion at the router level and devises an attack signature and that can be converted into a router filter. The signature defines a high-bandwidth aggregate, a subset of network traffic, and a local ACC determines an appropriate rate limit for this aggregate. Pushback propagates this rate limit for the aggregate to the immediate upstream neighbours that contribute the largest amount

of the aggregate's traffic. This mechanism works best against DDoS flooding-style attacks. Figure 3.1 shows the pushback mechanism at the router.

Aaaraj N et al. [24] proposed a mechanism called Neighbour Stranger Discrimination (NSD), in which the NSD routers perform signing and filtering functions besides routing. This approach divides the whole network into neighbors and strangers. If the packets from a network reach the NSD router directly without passing through other NSD routers, this network is a neighbour network. Two NSD routers are neighbour routers to each other if the packets sending between them do not transit other NSD routers. Therefore, a packet received by an NSD router must be either from a neighbour networks, or from a neighbour router. Each NSD router keeps an IP addresses list of its neighbour networks and a signatures list of its neighbour routers. If a packet satisfies neither of the two conditions, it is looked as illegitimate and dropped.

Chen et al. [22] extended the pushback model for detection and prevention of IP spoofed DDoS attacks by using the packet marking mechanism. In this mechanism all the intermediate routers marks the packets with their identification. When the packet reaches the destination it contains the marking which is marked by all the routers on the path that the packet traverses. Further the firewall at the victim site filters the packets based on the marking the packet contains. According to this scheme a packet traverses on the same path contains the same mark. Based on this information the firewall filters out the packets. And also it learns the signatures from the dropped packets and pushback the list to upstream routers to rate limit the traffic before reach the victim.

3.1.1. Traceback Mechanisms

The existing approaches for IP traceback can be grouped into two orthogonal dimensions: packet marking [26] and packet logging [27]. The main idea behind packet marking is to record network path information in packets. In mark based IP traceback, routers write their identification information (e.g., IP addresses) into a header field of forwarded packets. The destination node then retrieves the marking information from the received packets and determines the network path. Where as logging based methods stores the information about the packets in the network (i.e.,

at intermediate routers) and uses this information later to trace the source of the packets.

Due to the limited space of the marking field, routers probabilistically decide to mark packets so that each marked packet carries only partial path information. The idea was first put forward by Savage et al. [13], called probabilistic packet marking (PPM), in which the routers insert path information into the Identification field of IP header in each packet with certain probability, such that the victim can reconstruct the attack path using these markings and thus track down the sources of offending packets. PPM incurs little overhead at routers. However, it requires a flow of marked packets to construct the network path toward their origin.

Song and Perrig improved PPM basing on a pre established map of upstream routers, and provide authentication to the markings by encoding them using MAC functions [25]. In order to reduce the number of packets needed for the attack path reconstruction in PPM, Peng et al. [28] propose a mechanism which dynamically changes the marking probability of a router according to its location in the path. If each router marks packets with a fixed probability, the victim needs to wait for the packets marked by the routers farther away from it, which are relatively fewer. Therefore, the farther a router is to the victim, the higher the marking probability should be.

Belenky and Ansari [11, 12] proposed a deterministic marking approach (DPM), in which only the address of the first ingress interface a packet enters instead of the full path the packet passes (as used in PPM) is encoded into the packet. In this mechanism only the ingress routers marks the packets. When the spoofed packet reaches its destination it contains only the marking of Ingress (boarder) router from where the packet has generated. But this mechanism has a short coming that all the ingress routers are not in centralized control.

PPM based methods needs thousands of packets to reconstruct the path from victim to the source of attack, and is not suitable for small scale DDoS attacks. PPM based methods give better results when DDoS attack is strength is large and underperform when attack strength is small.

In order to further reduce the number of packets needed to reconstruct the path HOSOI et al. [29] proposed a packet marking mechanism using bloom filters to traceback to the source of attack. In this method k hash functions are applied on the IP address of each intermediate router and set the corresponding bits in the marking area of the packet. Further it can be forwarded to the downstream (next hop) router on the path to the packet's destination. If the bit is already set the current router overwrites the existing bit otherwise it simply set the bits and forwards. When the packet reaches its destination the victim extracts the mark from the packet and uses that to traceback to source of attack. By using this method only one packet is sufficient to traceback the attacker.

The basic idea in packet logging is to record the path information at routers. In log-based IP traceback, packets are logged by the routers on the path toward the destination. The network path is then derived based on the logged information at the routers. Compared to markbased IP traceback, the log-based approach is more powerful as it can trace attacks that use a single packet, i.e., software exploit attacks, along with flooding attacks [30]. Historically, packet logging was thought impractical due to the enormous storage space required for packet logs.

Snoeren et al. [20] proposed a hash-based IP traceback approach, called Source Path Isolation Engine (SPIE), to realize log-based IP traceback in practice. Their approach reduces the storage overhead significantly through recording packet digests in a space-efficient data structure, a Bloom filter [19]. SPIE has made a significant improvement on the practicality of log-based IP traceback. However, its deployment at high-speed networks has still been a challenging task due to the high storage overhead and access time requirement for recording packet digests. Considering the effectiveness of log-based IP traceback in tracing both flooding and software exploit attacks, there is a need to develop more scalable solutions to facilitate its deployment at high-speed networks. The next section briefly discusses about bloom-filters.

3. 2. Bloom Filters

Bloom filters were introduced in 1970 by B. H. Bloom [19] and they have been widely used in many applications such as database applications, peer to peer networks, resource routing and packet routing.

A Bloom filter is a space-efficient data structure used to test set membership of an input data. It is an array of m bits, initialized to zero, used to represent a set of n elements, $S = \{x_1, x_2, \ldots, x_n\}$. The filter uses k independent and uniform hash functions, $h_1, \ldots, h_k$ each with range in $\{1, \ldots, m\}$. To add an element $x_i \in \{x_1, \ldots, x_n\}$ to the filter, k hash functions are applied on the input x_i and the corresponding bits in the filter are set. The following is the pseudo-code for adding an element x to the filter.

```
ADD ELEMENT X
 for j = 1 to k
    do
         filter[hj(X)] ←1
```

It should be noted that when a bit is already set to "1"-the n additional settings do not change it. The existing "1" is just over written which is a simple OR operation of all hash values. To test the membership of an element y, the k hash functions are applied to y and the corresponding bits are checked. If one of the bits is "0" then clearly the element is not in the set. If all the bits are equal to "1" then we could say that the element belongs to the set. The following pseudo-code checks if y is an element of the set

```
CHECK ELEMENT Y
 for j = 1 to k
    do
        if filter[hj(Y )] = 0 return False
return True
```

If an element z has all the corresponding bits equal to "1" without the element itself belonging to the set then we can call that a false positive. The false positive rate can

be calculated as follows. When a given hash function h_i is applied to an input x_i the result is a value between *1* and *m*. Since the hash functions are uniform, the probability that this result is equal to a particular number *b* is $\frac{1}{m}$. Therefore the probability of the bit at position *b* being 1 after one hash function is $\frac{1}{m}$. The probability that it is 0 is $1-\frac{1}{m}$. The probability that it is 0 after all *k* hash functions are applied is $(1-\frac{1}{m})^k$. Since there are *n* elements in the set, the probability that the bit *b* is equal to 0 after we process all n elements is $(1-\frac{1}{m})^{kn}$. Hence $1-(1-\frac{1}{m})^{kn}$ is the probability that a given bit *b* is set to 1 after all input elements $x_1, \ldots, x_n$ are processed. Since we want the false positive rate, we need the probability that for an arbitrary input *y* the corresponding *k* bits are 1 without *y* belonging to the set. This probability is

$$f_p = \left(1-\left(1-\frac{1}{m}\right)^{kn}\right)^k \qquad (1)$$

$$\approx \left(1-e^{-\frac{nk}{m}}\right)^k \qquad (2)$$

From the above equations we can say that the false positive rate depends on *k* and the ratio $\frac{m}{n}$.

3. 3. Research Gaps

In this section we discuss about the research gaps found in different mechanisms existed in two classes of methods.

In Hop-count filtering [23] method hop-counts can be calculated based on the value of TTL field where the attacker may spoof this field too.

PI [15] uses identification field of the packet header for marking purpose and this method fails if the attacker spoof the identification field.

Pushback mechanism proposed by John et al. [21] puts a processing overhead on the routers.

PPM [13] proposed by Savage et al. incurs overhead at the routers and it requires thousands of packets to reconstruct the attack path.

DPM [11] which was proposed by Beleky et al. has a short coming that the ingress routers may be in the hands of private organizations who doesn't support this method.

SPIE [20], ingress filtering [16] had implementation issues. Partial implementation of these solutions caused loopholes in system which makes the solution ineffective.

Most importantly there is no mechanism which effectively filters the incoming IP-spoofed packets and traceback to source of attack simultaneously.

From all above gaps found in the existing schemes, we require a mechanism which fulfils the following needs.

1) Proposed solution must eliminate the processing overhead on the routers.
2) It must require only a small hardware change on routers or it must eliminate the changes in hardware.
3) It must be well scalable to a large number of attackers.
4) It must perform real time filtering of incoming spoofed packets and also traceback to the source of attack simultaneously.

Aiming to fulfill the above mentioned requirements, we propose a hybrid packet marking mechanism that enables the victim to perform real time filtering of incoming IP spoofed traffic and simultaneously traceback to the actual source of attack, which is elaborated in the next chapter.

Chapter 4

DESIGN AND IMPLEMENTATION OF PROPOSED HYBRID PACKET MARKING MECHANISM

In this chapter, we present a hybrid packet marking mechanism which incorporates two packet marking methods for traceback. One is using Bloom Filters and the other is using Probabilistic Packet Marking (PPM). The main goals of this scheme are to: i) Distinguish the IP Spoofed packets from the legitimate packets, ii) Filter IP spoofed packets and finally, iii) Traceback to the source of attack.

In this mechanism, each intermediate router marks the incoming packets in two marking areas and forwards to the downstream router. Each packet consists of two marks; one is used in filtering the packets and the other is used in traceback procedure. The basic idea of this mechanism is to defend the attack in a full fledged manner. That is why we combine the features of filtering and traceback mechanisms into a single mechanism, which filters the packets and simultaneously traces back to the source of attack. This kind of methodology has not been attempted in earlier reported works.

This chapter is organized as follows. Section 4.1 first deals with the calculations involved in the packet marking mechanism for filtering packets. In Section 4.2, we discuss the way in which marking is done for traceback process using Bloom Filters. Section 4.3 deals with marking procedure for traceback using PPM strategy. Section 4.4 gives the details of filtering procedure. In Section 4.5, the process used to traceback using Bloom Filters is discussed. Section 4.6 presents the technique of traceback using PPM strategy.

4.1. Packet Marking Method for Filtering

In this method, we record the path information that the packet traverses from its source to destination. This helps in differentiating the packets from different sources.

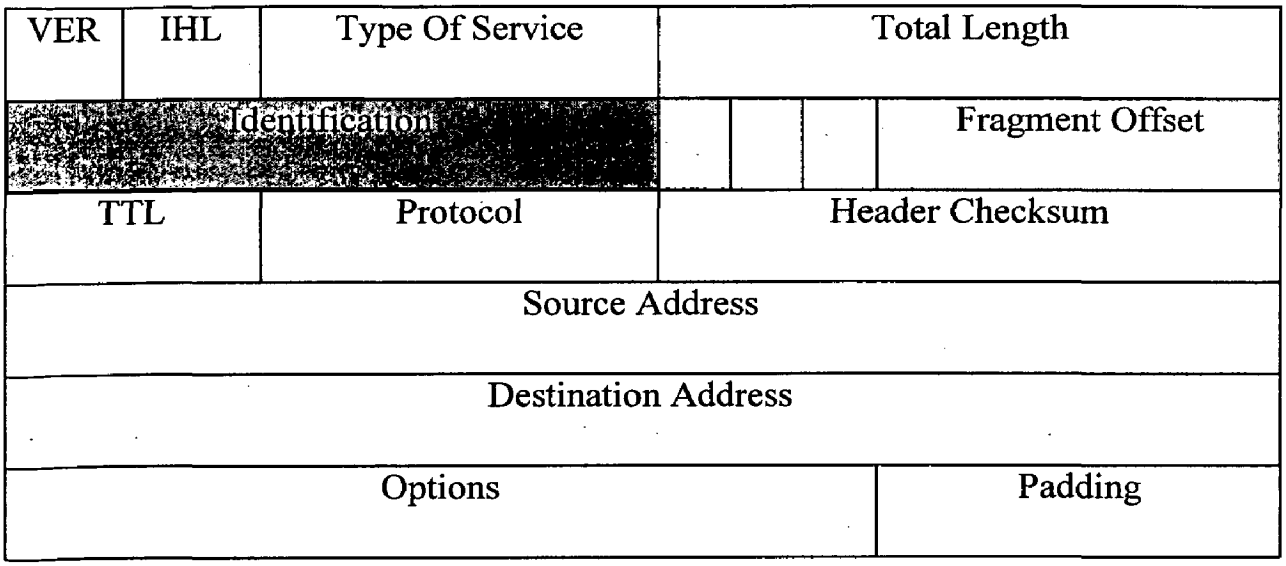


Figure 4.1: IP Packet Header

Figure 4.1 shows the various fields of an IP packet Header. Only some fields can be editable in this header at the intermediate router, like TTL, identification, fragment offset, checksum. In order to record the path information of the packet traversed, we need to add that information in to the packet at the intermediate routers. Adding data to a packet results in an increase of the packet size, and it might cause fragmentation of the packet, which in turn leads to an increase of the network traffic. In order to avoid such increase in the packet size; a possible method is to put all information into a fixed sized space. The first 20 bytes of the IPv4 packet header is fixed, so this could be the most suitable area for marking the path information of the packet. For a fast and efficient marking procedure, most marking methods utilize the identification field in the IP header as a marking area as shown in Figure 4.1. This field is used when the packet is fragmented in the network. Researchers have shown that packet fragmentation does not occur in the network so often [31]. So utilizing this field for filtering the spoofed packets is reasonably acceptable. In our mechanism too, we use this identification field as the area for packet marking.

4.1.1. Computing Packet Marking

The size (i.e., length) of the identification field is 16-bit and the size of the IP address is 32-bit. So it is not possible to put the IP address directly into the identification field. We need to convert the IP address into a 16-bit value. So the mark made by the router could be the function of its IP address. To fit a 32-bit

length IP address into 16-bit identification field we apply a hash function on the IP address of the router which converts the 32-bit length value into a 16-bit length value. Figure 4.2 illustrates the computation of the packet marking for filtering procedure.

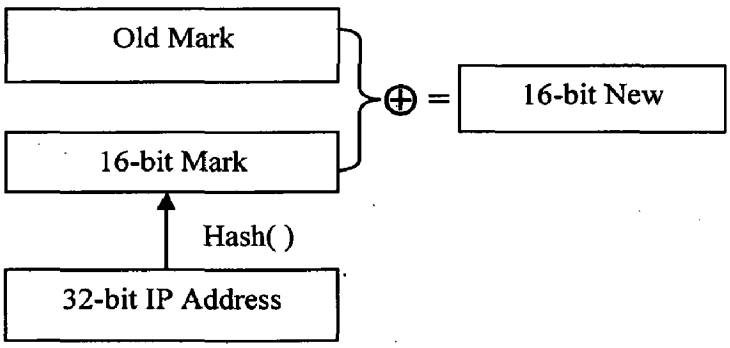


Figure 4.2: Computation of packet marking for filtering procedure

Each router pre deterministically calculates and stores the hash value of its IP address. When the packet is received by the intermediate router, it puts the 16-bit hash value (i.e., digest) into the marking area of the packet. If some information is already available in the marking area of the received packet, then the router calculates the exclusive-or (XOR) of its hashed IP address with the previous value in the marking area and puts the new value back in to the marking space of the packet. This method ensures that the marking does not change the length (i.e., size) of the marking area when a packet traverses over the internet from its source to destination, so the packet size remains unchanged.

4.2. Packet Marking Method for Traceback Procedure using Bloom-Filters

In this section, we present a packet marking mechanism which helps in tracing back the IP spoofed packets sent by the machines which are under the control of the attacker.

In marking area, information about various routers is recorded which are on the path (i.e., route) of a packet, when it is traversing from its source to destination. In this type of marking each intermediate router places its membership identity into the IP packet.

Since adding information to the packet increases packet size and further, can cause packet fragmentation, we use the 13-bit fragment offset and 3 extra bits of the IP header as shown in Figure 4.1 for traceback marking. Doing so does not affect anything in the network because we are already using identification field to carry the information for filtering purpose. Fragment offset and MF-bit can be used when the network supports fragmentation. Since we are using identification field of the packet header for filtering purpose, packet marking mechanisms do not support fragmentation. That is why we selected use these 16-bits (3 flag bits and fragment offset) for our traceback marking purpose. We call these 16 bits as a Bloom filter.

Before the marking process starts, each intermediate router calculates k (M-bit) digest values of its IP address and stores it in the router. All these k digest values are calculated by applying k hash functions on the input, i.e. on IP address. During the marking process each intermediate router writes its membership information in to the packet header by converting all the digest values in to form of bit "1" at the corresponding locations in a m $(=2^M)$ sized bit array (Bloom filter). For example if k=2, and the digest values calculated are 1010 (i.e., 10 in decimal) and 0101 (i.e., 5 in decimal), then the 11^{th} bit and the 6^{th} bit in the filter are set like this 00000100 00100000. We can observe this in Figure 4.3.

When a packet is forwarded by a router the bits "1" in the marking area are overwritten if they are already set, otherwise it just set the bits at the corresponding locations.

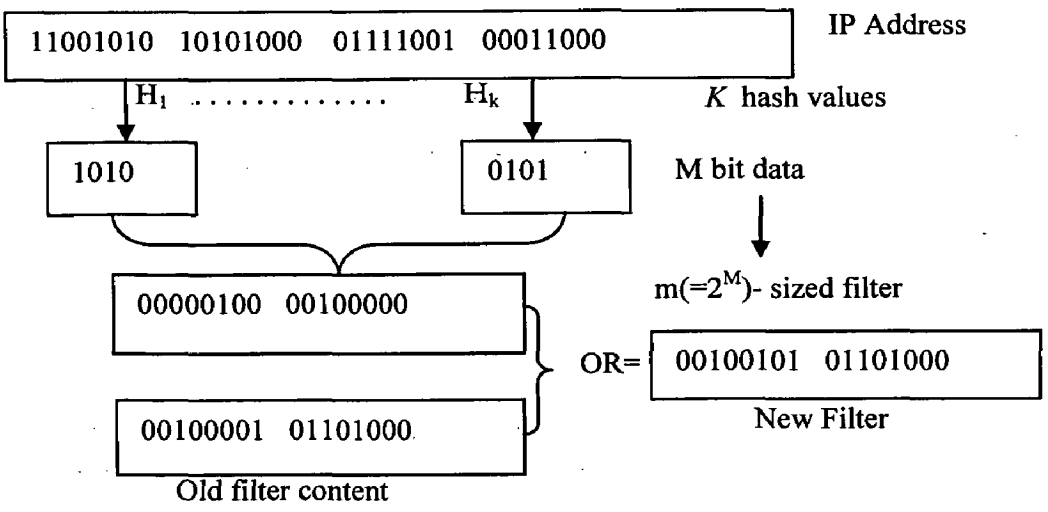


Figure 4.3: An example of bloom filter operation for K=2, M=4 (m=16).

This marking process is done at each supported intermediate router in the path the packet traversed from its source to destination.

In this mechanism each ingress router does some special task. When the packet is forwarded through this router it initializes the filter (Marking area) by setting all of the bits to "0" before marking on it.

If we take higher values for *k (k >1),* then the filter (marking) area will be exhausted for shorter distances (i.e., for less number of hops). If the filter area is exhausted (i.e., more number of bits set to "1") then there are more chances to the routers to pass the membership test even if they are not in the path and this leads to higher false positive rates.

For example in an ideal case with *k =2,* filter is fully exhausted (all the bits set to "1") only for 8 hops because the available area is 16-bits and each router sets two bits at a time. If the filter is fully exhausted we get the false positive rate of 1, in which all the router in the network passes the membership test. That is why for our method we have taken *k=1* (i.e., one hash function is applied on the IP address and only one bit will be set by each router).

4.3. Packet Marking Method for Traceback Procedure using Probabilistic Packet Marking

This section discusses about the marking procedure for traceback of spoofed packets. This procedure uses the same marking area as specified in section 4.2, i.e., 16-bits of the packet header (3 flag bits and fragment offset). We are dividing this marking area into two parts, first one is marked flag and the second one (15 bit) is traceback mark area. These two parts are initialized to "0" at the boarder router. Figure 4.4 shows the different marking areas for filtering and trace back procedures.

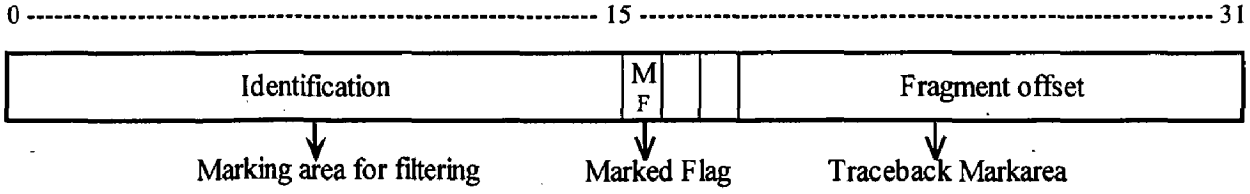


Figure 4.4: Division of 4 bytes of a packet header for filtering and traceback as different marking areas.

Initially the marked flag (MF) bit is set to "0". It will be set to "1", when the router marks the packet for traceback purpose. When the packet is received at an intermediate router, the router checks the MF bit. If the bit is set to "1" then this router forwards the packet to the downstream (next hop) router without marking the packet. If the bit is set to "0" then it indicates that the packet has not been marked by any router during its transit from its source, and the router generates a random number which is in the range between 0 and 1. (i.e., [0, 1)). If the generated value is less than the threshold (value compared against the generated value) value then the packet is marked with the router identification and sets the MF bit to "1". Here, the available area for marking is 15-bits and the router IP address is 32 bits. So we apply a hash function on IP address which converts 32-bit value to a 15-bit digest (router identification). Figure 4.5 illustrates the above procedure.

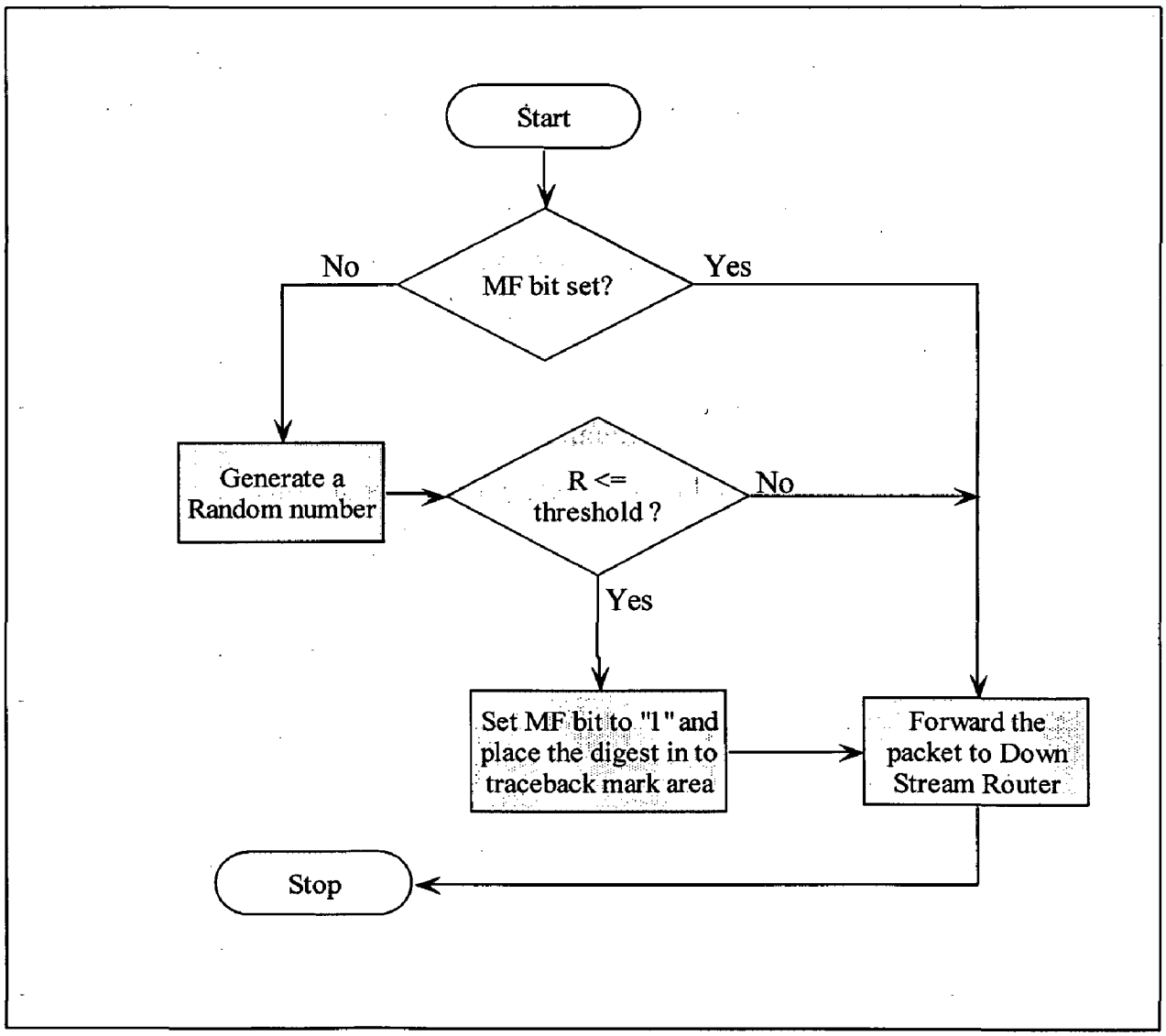


Figure 4.5: Flow chart showing marking procedure for traceback using PPM

In this way each intermediate router marks the packet during its transit from source to destination. In this procedure each packet can be marked by the intermediate routers at most once. Once the MF bit is set, no other router overwrites it.

4.4. Filtering Procedure

Filtering procedure employs a firewall associated with a border router of the protected network or with the destination machine. This firewall scans every incoming packet for marking associated with it and filters out the attack packets. On employing the above marking scheme, the marking on a packet will depends on the path it traverses.

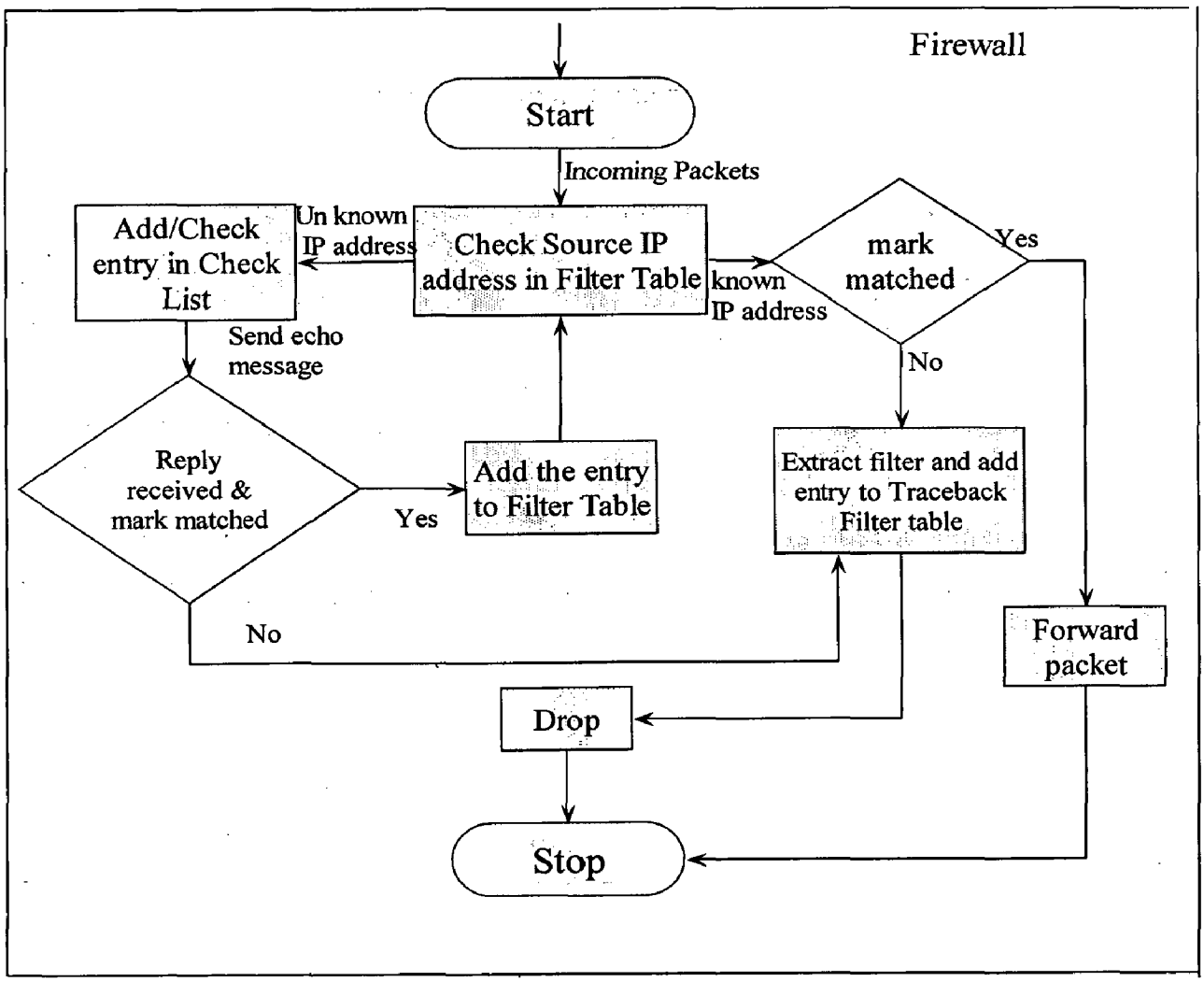


Figure 4.6: Flow chat for the Filtering procedure at the firewall

If the source IP address of the packet is spoofed then it must have a marking that is different from the marking when the packet traversed from its true source. Based on this differentiation we can easily identify and filter out the packets at the destination. Those packets having the same mark against source IP address are accepted. The implementation of our firewall is illustrated in Figure 4.6

Initially the filter table is empty. In order to start the filtering procedure the filter table must have the genuine markings, (i.e., source IP address, marking tuple). To fill up this table we, first, learn the markings for each source IP address for all incoming packets. We can set this learning period as 5 or 10 minutes whenever we assume that the destination machine is not under attack. During this learning time the firewall keeps the source IP address, marking tuples in the filter table which can be used in future to verify each incoming packet.

Normally packets follow different routes in the internet environment because of congestion and other problems like link breakdowns. With the basic filtering procedure the percentage of packets dropped would be more. So we consider more than one mark for the same source IP address during the learning phase where each mark represents the unique path that the packet traverses. Table 4.1 illustrates the structure of the filter table.

Source IP address	Markings	No of times received
0.0.0.73	987	12
	1764	9
0.0.0.38	1632	21
	750	10
	903	8
0.0.0.56	534	39
	801	30
	837	24
	2425	15

Table 4.1: Example of Filter table which is used in filtering the packets

After the learning phase the system is ready to perform the filtering procedure. If a packet is received at the destination (firewall) with the source IP address that already exists in the filter table, then it is checked against its marking and decision is made

to forward or to drop the packet. If the mark is matched then the packet will be forwarded to the destination machine for further processing; otherwise the packet is submitted to the traceback filter module for traceback process. We will see traceback process in the next section. If the packet is received with the source IP address that is not present in the filter table, then an echo message is sent to that IP address and the pair (address and mark) is saved in a checklist. If the echo reply is received with the same mark as in the checklist, the entry is added to the filter table; otherwise this address is marked as spoofed address and added to the filter table.

We can also push back the filter table to upstream routers to filter the packets before they reach the destination machine in order to reduce the strength of the attack.

4. 5. Traceback Procedure Using Bloom-Filters

Traceback procedure employs a Traceback filter in the firewall which extracts and stores the triplet (IP address, mark and filter). Here IP address is the source IP address, mark is used in differentiating the path that the packet traversed and filter is the mark (Bloom filter) received at the firewall which is marked by traceback marking procedure at the intermediate routers. This marked (filter) data is composed of an accumulation of all marks of routers on the path traversed by the packet. When a packet is submitted for traceback, the traceback filter extracts source IP address, mark and filters from the packet and stores this triplet in the traceback filter table. The table shown in Table 4.2 gives an example of traceback filter table.

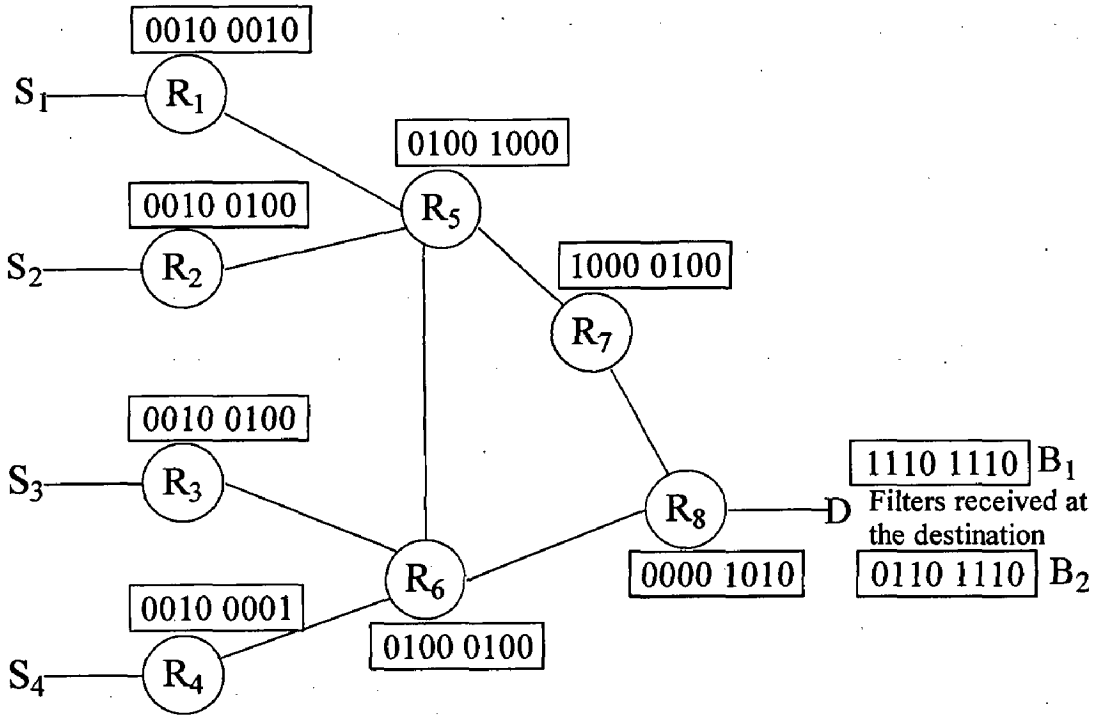
Source IP address	Markings	Filters
0.0.0.73(IP_1)	987(m_1)	10100001 01011100(B_1)
0.0.0.38(IP_2)	1632(m_2) 750(m_3)	10001001 01111000(B_2) 00000111 11101000(B_3)
0.0.0.93(IP_3)	812(m_4) 1204(m5)	01000101 10101010(B_4) 00010011 10101001(B_5)

Table 4.2: Example of traceback filter table

This table can be used at any point of time, while the attack is going on or after the completion of attack to traceback to the source of attack.

The following example illustrates the traceback procedure at the victim site.

We describe a simple example of how the proposed traceback mechanism works. Refer to the Figure 4.7. For simplicity consider a 8-bit(*m*) filter where *M* is 3-bit and k=2, here *k* represents the number of hash functions applied on the IP address of the router. So each router generates two values, within the range 0 to 7, when two hash functions are applied on the input.



S*: source D: Destination R*: Router

Each router generates hash values in the range 0 to 7 (M=3-bit)

Each router calculates and stores two hash values (k=2)

Size of marking area is m = 2^3 = 8-bits (M=3)

Marking process-Data in the marking (filter) area

{**** ****} (: S_1 at packet generation arbitrary bits)

{0010 0010} (: R_1 after initializing by 0 and write)

{0110 1010} (: R_5 does the OR operation of the filter)

{1110 1110} (: R_7 does the OR operation of the filter)

{1110 1110} (: R_8 does the OR operation of the filter)

{1110 1110} (: D received filter at the destination)

Traceback process

$(S_1 \rightarrow D)$:{ 1110 1110} (: R_1 R_5 R_7 R_8) ┐
$(S_2 \rightarrow D)$:{ 1110 1110} (: R_2 R_5 R_7 R_8) ┘ } False positive

$(S_1 \rightarrow D)$:{ 0110 1110} (: R_1 R_5 R_6 R_8) ┐
$(S_3 \rightarrow D)$:{ 0110 1110} (: R_3 R_6 R_8) ┘ } False positive

$(S_4 \rightarrow D)$:{ 0110 1111} (: R_4 R_6 R_8)

Figure 4.7: An example of a traceback process in a small network with 8-bit marking area.

Consider a packet sent from source S_1 to the destination machine D that follows the path S_1R_1 R_5 R_7 R_8D. So, the packet is marked by the routers R_1 R_5 R_7 R_8 and all the bits are accumulated in the marking area of this packet. If B_1 is the filter received by the destination machine D and we use this filter to traceback to the actual source we get two paths namely R_8 R_7 R_5 R_1, and R_8 R_7 R_5 R_2,. Therefore, tracing of a packet from Destination D to source S_1 results in two candidate paths, one is true and other one is false positive. The false positive rate of this case is 0.5. We can control this false positive rate by considering multiple paths for the same source IP address. Even if packets originate from the same source they may follow different routes while they traverse to their destination. When the packet follows different routes it carries different marks as well as different filters for the same source IP address. Suppose the packet sent from source S_1 to destination D takes two different paths, S_1R_1 R_5 R_7 R_8D and S_1R_1 R_5 R_6 R_8D. Then there are two different marks and two different filters for each source IP address, as shown in Figure 6. B1 and B2 are two filters received by destination D. If we use B1 to traceback we get two paths namely, R_8 R_7 R_5 R_1, and R_8 R_7 R_5 R_2, and using B2 we get two paths namely, R_8 R_6 R_5 R_1, and R_8 R_6 R_3. If we observe the above four paths R_8 R_7 R_5 R_1, and R_8 R_6 R_5 R_1, are converging at same point R_1 from which the packet originated. If we consider *n*

different paths (*n* marks and filters for same source IP address), then *n* traced paths converge at same point which, we can say is a source of attack.

In order to understand the false positive nature of a Bloom-filter considers the following example by referring to Figure 4.8. Figure 4.8 shows a simple network which we use as our work. Each value below the left corner shows the hash value of the IP address of that router (i.e., node). Each router in this network sets only one bit in the bloom filter when the packet received, because k=1.

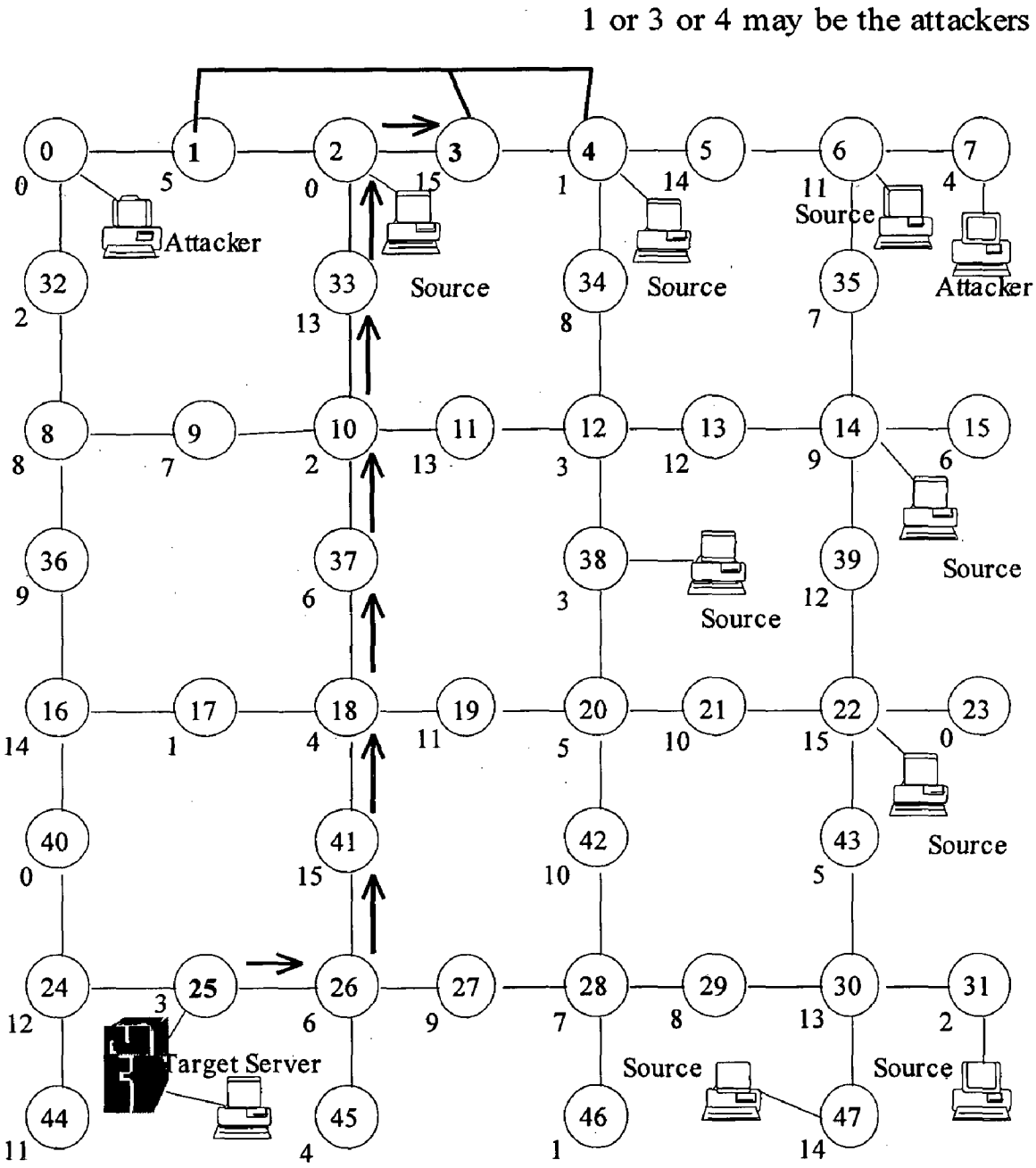


Figure 4.8: Example of a traceback process using Bloom filters with false positive nature.

For example if the filter received at the destination is 10100000 01010101. Then the router with the hash values 0, 2, 4, 6, 13 and 15 are marked this packet. In Figure 4.8 the arrow marks represents the path to the attacker when we traceback. All the hash values are exhausted while we test the membership of the routers when we reached node 2 and node 3. So neighbors of node 2 and node 3 may be attackers. Neighbors of node 2 and 3 are 1, 3 and 4. The total number of routes existing here are 3. False positive rate is 1-1/3. i.e.,0.66.

4.6. Traceback Procedure Using PPM Strategy

This procedure employs a traceback filter which extracts and stores the triplet (IP address, mark, 15-bit digest) as mentioned in section....Here IP address is the source IP address of the received packet, mark is the one which is calculated and marked at intermediate routers using marking procedure filtering procedure. 15-bit digest is the identification value of a particular router that is there on the path the packet traversed. When a packet is detected as a spoofed IP packet, then it will be submitted to a traceback filter. After submission of such packets to traceback filter, it extracts the triplet form the packet as mentioned above and stores them in traceback filter table. Table 4.3 shows the sample of a traceback filter table using PPM. If the digest value is already present against a particular mark then it ignores it; otherwise it adds the entry in table. Like this we sample all digest values for all particular marks and for source IP addresses. These samples represent the different router identification from which the packet has traversed for that particular route.

Source IP address	Mark	15-bit Digest
0.0.0.2	987(m_1)	36-8-32-40-16-24(S_1)
0.0.0.6	1745(m_2)	20-21-18-26-41-19(S_2)
	23456(m_3)	29-43-30-28-27-26(S_3)
0.0.0.7	45623(m_4)	35-14-22-20-19-41-26(S_4)

Table 4.3: Sample of a traceback filter table.

By using these sample values we can construct the path to the source of attack from the victim. Each router has its unique identification value (digest). If the digest value

is there in the traceback filter table, then we can say that the packet has come from that router. The following example shows how to traceback to the source of attack using this filter table.

Figure 4.9 shows the example of traceback process by using the traceback filter table shown in Table 4.3.

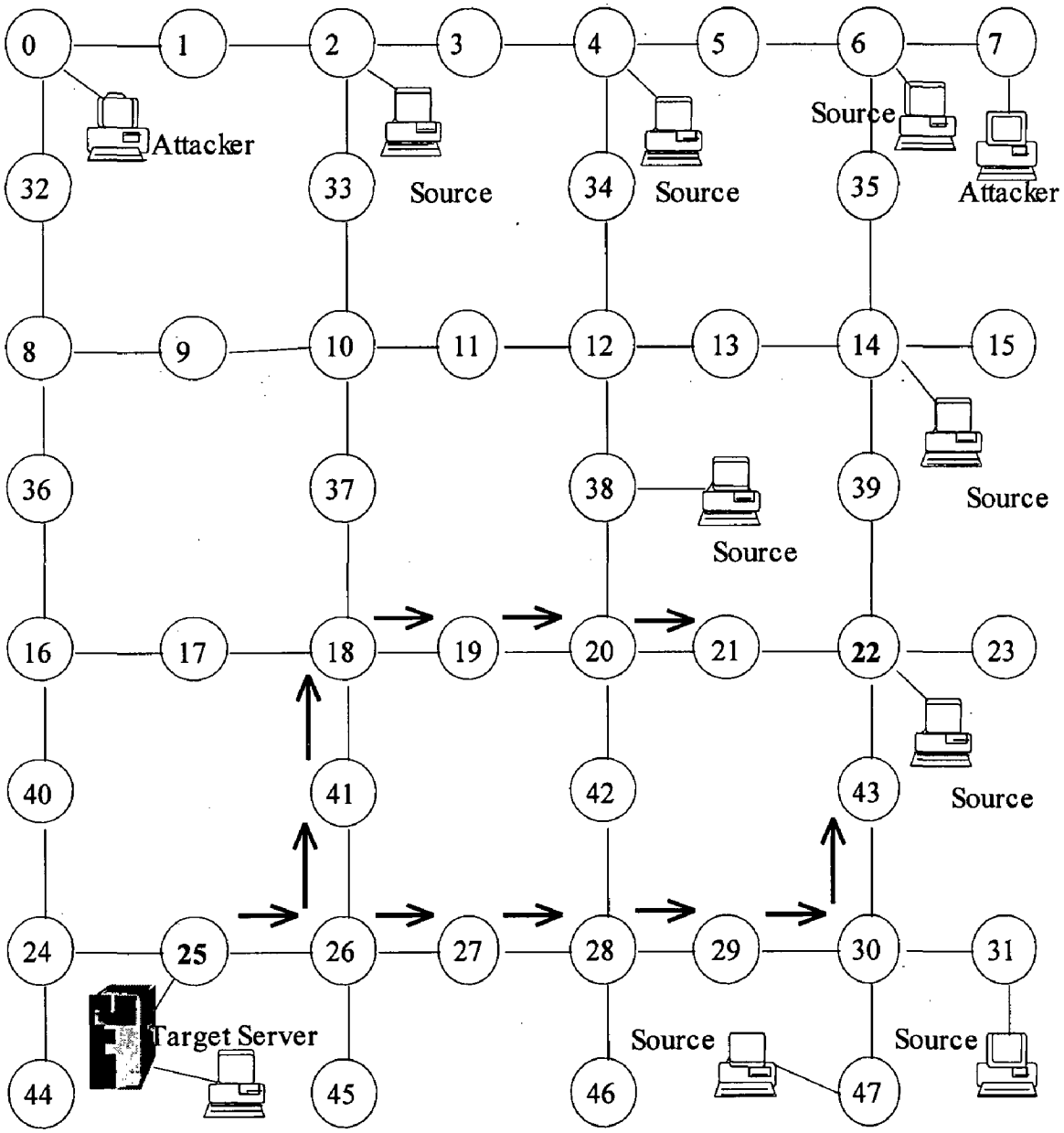


Figure 4.9: Example of a Traceback process by using traceback filter table using PPM.

Consider the marking against the 0.0.0.6 in Figure 4.9. It consists of two marks and two sample digest values. If we start traceback by using the sample S_2 we get the

path 25-26-27-28-29-30-43. And by using the sample S_3 we get 25-26-41-18-19-20-21. So the border router for first path is 43 and for the second one is 21. These two are connected to the node 22. So we can say that the source of attack is the node 22. Here the entry in the table shows that it has originated from the node 6 but when we traceback by using the samples we got to node 22. So node 22 spoofed the address with 6. One disadvantage with this mechanism is that if sufficient number of packets are not arrived from the attacker we may not get to the attacker by using the samples. Because of insufficient number of samples we may get false negatives while tracing back the attack source. If we traceback by using sample S_4 we may not get the exact path to the source of attack.

4.7. Advantages of the Proposed Mechanism

Our proposed mechanism has the advantages over the existing ones [11-13, 15-16, 21]. The existing methods can be broadly divided into two classes. The first class of methods [15, 16, 21] defends the attack by filtering the packets where the attacker cannot be traced and punished. And second class of methods [11-13] defend the attack by tracing back to the source of attack and taking punishable actions on the attacker in order to avoid further attacks. In the latter method the victim has to experience the attack. Our mechanism is embedded with both the mechanisms, so that we can defend the attack by filtering the packets, as well as traceback to the source of attack and can take punitive actions against the attacker in order to avoid further attacks.

The disadvantage with this mechanism is the slight increase in processing overhead on the intermediate routers. All the hash values of the IP address are calculated and stored in the router in 16-bit field. The processing overhead is calculation of XOR and OR operations of the old marks and the hashed IP which are readily available in the router

Chapter 5

SIMULATION AND RESULTS

The effectiveness of our scheme was evaluated by simulation using a JAVA based JiST (Java in Simulation Time) simulator.

4.7. JiST Simulator

The architecture of JiST is depicted in Figure 5.1. It consists of four distinct components: a compiler, a bytecode rewriter, a simulation kernel and a virtual machine. One writes JiST simulation programs in plain, unmodified Java and compiles them to bytecode using a regular Java language compiler. These compiled classes are then modified, via a bytecode-level rewriter, to run over a simulation kernel and to support the *simulation time* semantics described shortly. The simulation program, the rewriter and the JiST kernel are all written in pure Java. Thus, this entire process occurs within a standard, unmodified Java virtual machine (JVM).

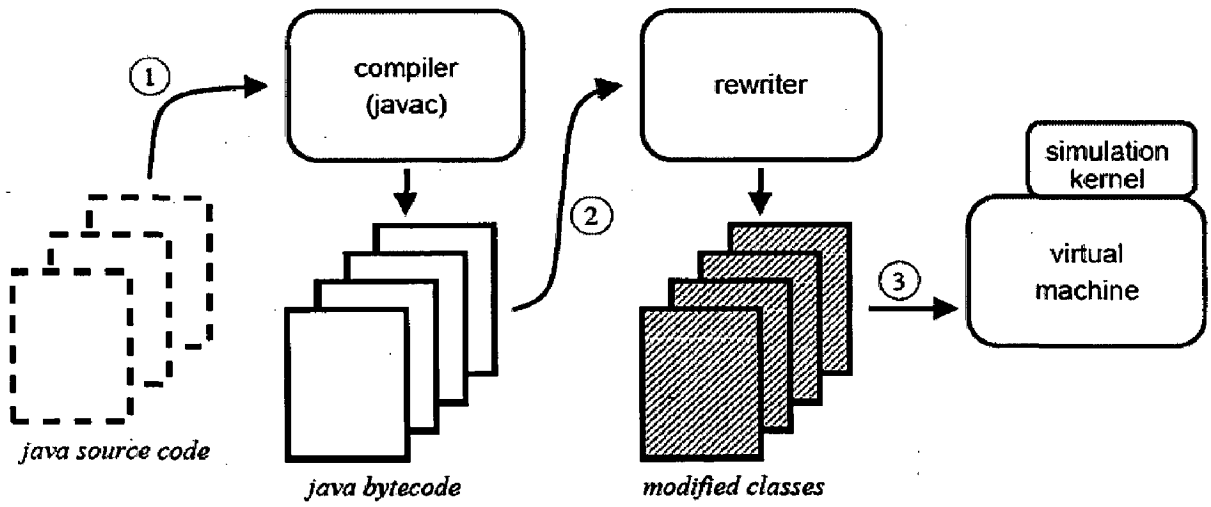


Figure 5.1: The JiST system architecture – simulations are compiled (1), then dynamically instrumented by the rewriter (2) and finally executed (3). The compiler and virtual machine are standard Java language components. Simulation time semantics are introduced by the rewriting classloader and supported at runtime by the Java-based simulation kernel.

Embedding the simulation semantics within the Java language allows us to reuse a large body of work, including the Java language itself, its standard libraries and

existing compilers. JiST benefits from the automatic garbage collection, type-safety, reflection and many other properties of the Java language. This approach also lowers the learning curve for users and facilitates the reuse of code for building simulations. The use of a standard virtual machine provides an efficient, highly-optimized and portable execution platform and allows for important crosslayer optimization between the simulation kernel and running simulation. Since the kernel and the simulation are both running within the same process space we reduce serialization and context switching overheads. In summary, a key benefit of the JiST approach is that it allows for the efficient execution of simulation programs within the context of a modern and popular language. JiST combines simulation semantics, found in custom simulation languages and simulation libraries, with modern language capabilities.

5.2. Results

The performance results of our scheme are affected by the values we choose for different parameters like the number of marks considered for a single source IP address, the number of hash functions applied (k) in traceback procedure using Bloom-filters and the marking probability of each router while marking in PPM strategy. In our simulation, we have come up with some suitable values for these parameters by trial and error and we have tested the effects of changing the values of these parameters. We have shown these effects graphically in the following sections. A network of 48 nodes was created, in which 10 nodes were sources. Among these 10 sources we have chosen some nodes as legitimate sources and rest as attackers. Figure 5.2 shows the network topology used during the simulation of our mechanism. Nodes which are connected to the desktop PCs represent the sources. Each source generates traffic with a rate of 2kbps for 5 minutes. The rate of generation of packets and the time period of traffic generation do not affect our results in a substantial way.

Our results do not depend much on the topology of the network because these mechanisms work purely based on packet marking. Normally, the denser the network, the higher the number of routes packets take to their destination. Similarly, rarer the network, lesser the number of routes. If the packets take more number of

routes, there is a chance of dropping more number of packets at the firewall. For our simulation we have taken a network that is neither dense nor rare.

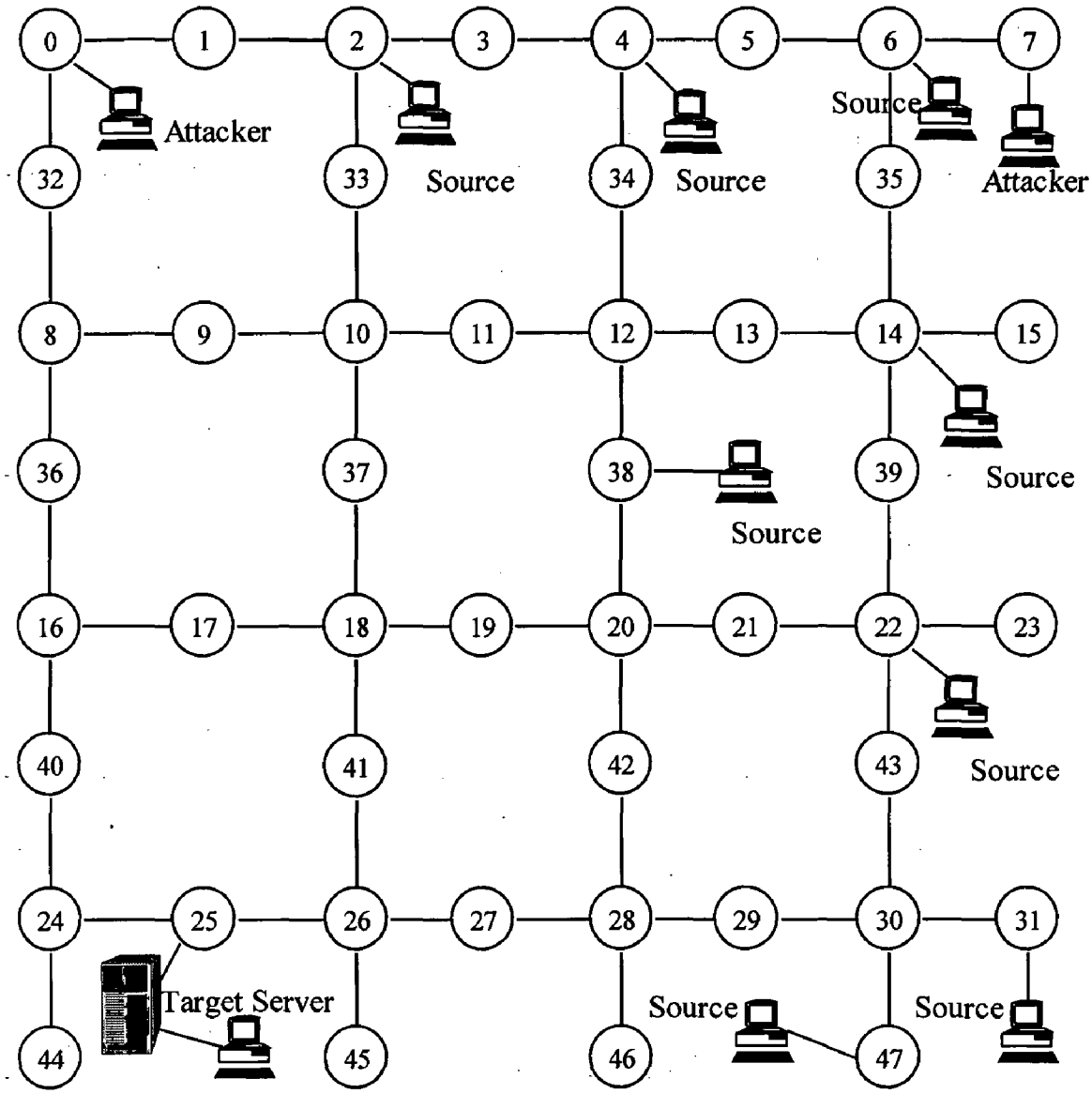


Figure 5.2: Network topology used for simulation. The nodes which are connected to the desktop PCs are the sources.

Our first method, which uses Bloom-Filters for traceback process was simulated by changing the number of attackers. Each time, the percent of legitimate packets accepted and the percent of spoofed packets accepted are observed and the results are tabulated. We have tabulated these results by taking the mean value from 20 independent simulation runs. The value of maximum number (threshold) of markings (i.e. different paths) considered against a single source IP address was also changed.

Our second method which uses PPM for traceback process was simulated to estimate the minimum number of packets required to reconstruct the path to source. For this method we plot the graph for a single attacker by varying the path length d (number of hops between source and destination). In this method we have taken the marking probability of each router as $2.4/d$, where 2.4 is the proportionality constant, which is derived through a trial and error method during the simulation; here d is the number of hops between source and destination.

The results of our method may slightly vary if the topology of the network is changed. The simulation results which we have shown in the following graphs are based on the network topology which is shown in Figure 5.2.

The following discusses the results obtained during the simulation.We compare the results of our first method with the MDADF [22] method. MDADF method considers only one route for the packets, unlike our proposed method.

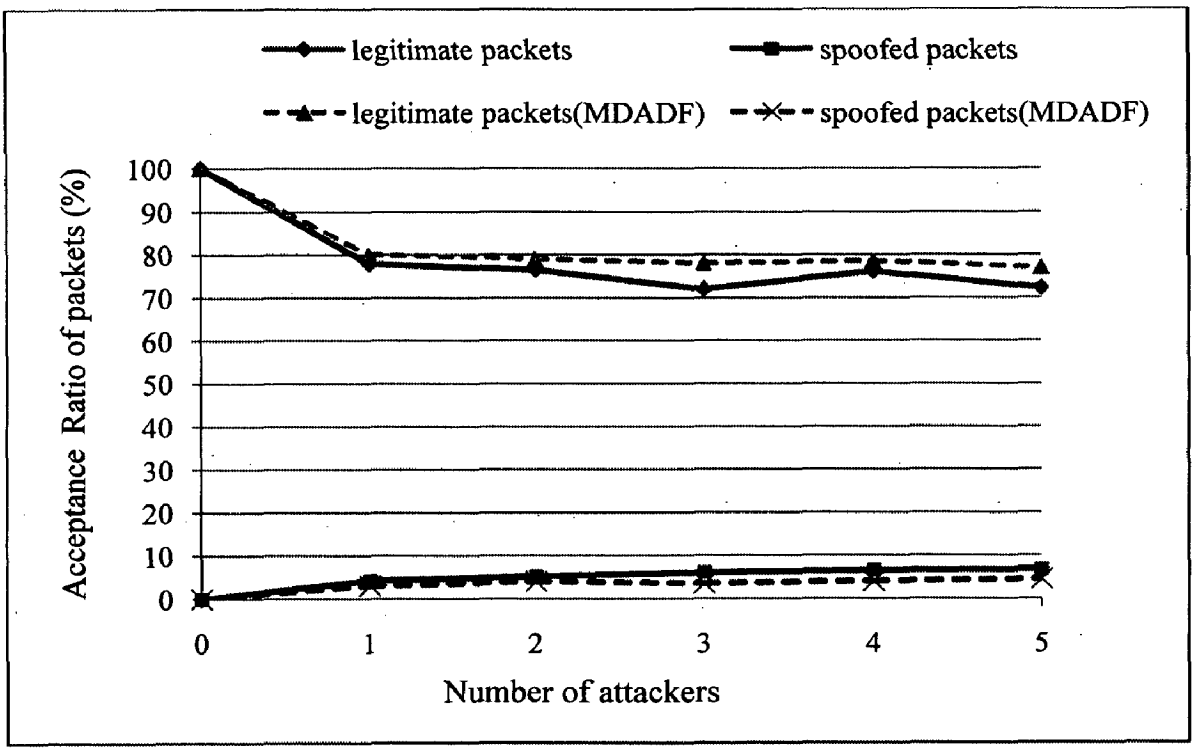


Figure 5.3: Acceptance ratio of packets vs Number of attackers with the threshold 1.

Figure 5.3 shows that the percent of packets accepted at the firewall under different magnitudes of attack. Here we observe that around 25% of the genuine packets are dropped and around 6% of the spoofed packets are accepted, when we consider maximum of one route (i.e., threshold =1) for the packets which are generated from

same source. Threshold is defined as the number of different marks considered for the packets originated from the same source. As the number of attackers is increased, there is a slight decrease in the acceptance rate of legitimate packets. The solid lines in Figure 5.2, 5.3, 5.4, 5.5 shows the acceptance ratio of packets of our proposed method and dashed lines shows the acceptance ratio of packets of MDADF method [22]. When we consider only one route (only one mark) for a single source IP address MDADF is giving better results.

Figure 5.4 shows that the ratio of packets accepted at the firewall under different magnitudes of attack. Here, we observe that around 20% of the genuine packets are dropped at the firewall and there is a negligible (1%) increase in the acceptance ratio of spoofed packets compared to threshold 1, when we consider maximum of two routes (threshold 2) for the packets which are generated from same source. With threshold value 2 we can observe that there is an increase in acceptance of the number of legitimate packets.

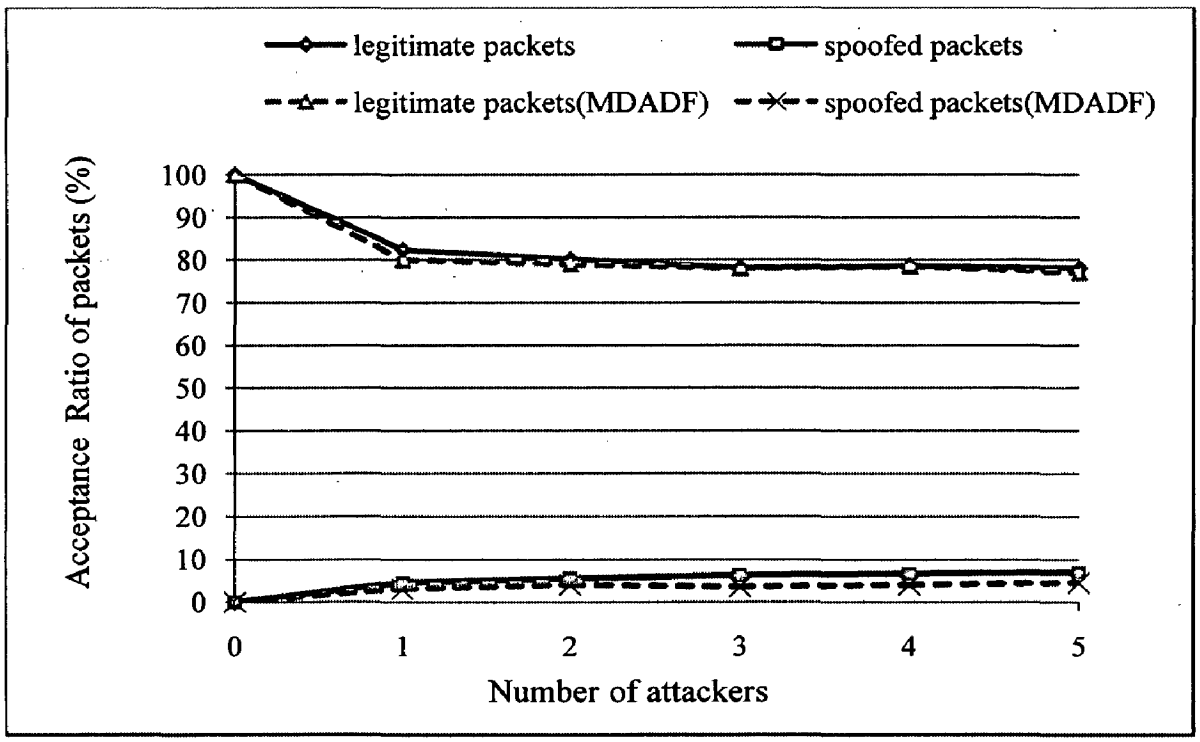


Figure 5.4: Acceptance ratio of packets vs Number of attackers with the threshold 2.

Figure 5.5 shows the acceptance ratio of packets at the firewall under different magnitudes of attack. Here we observe that around 9% of the genuine packets are dropped at the firewall with the threshold value 3. With threshold value 3 we can

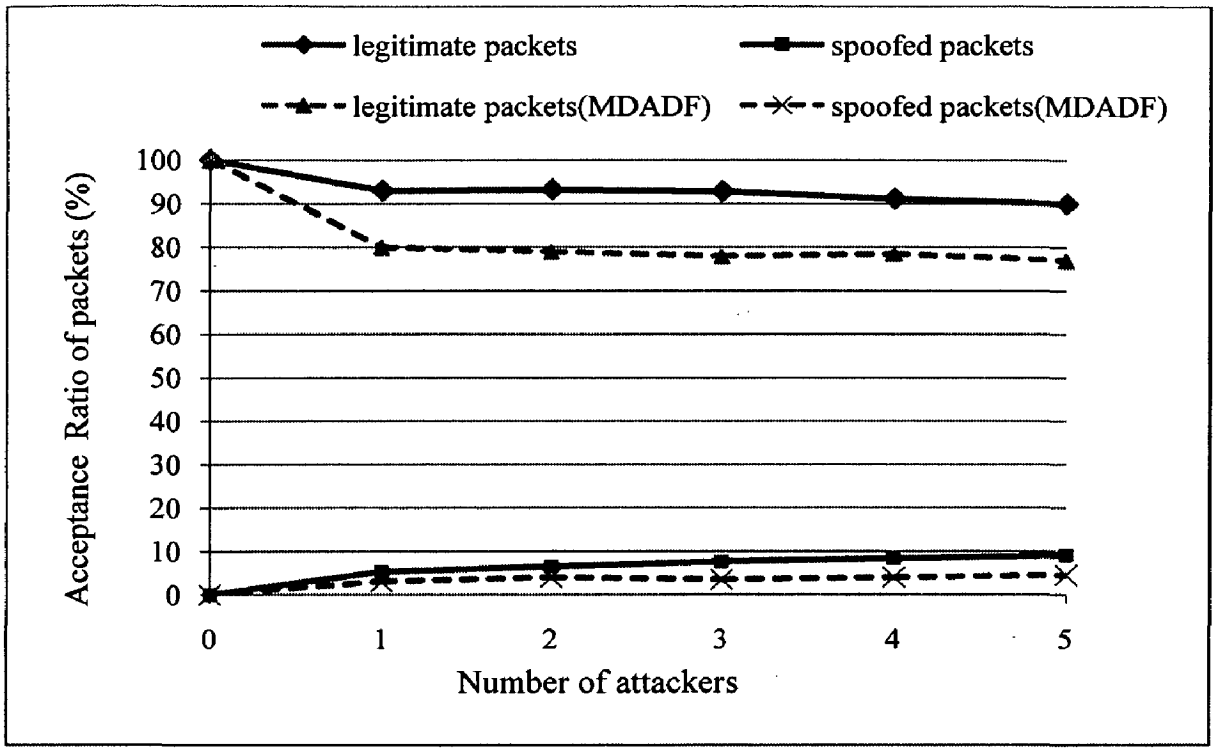


Figure 5.5: Acceptance ratio of packets vs Number of attackers with the threshold 3.

observe that there is a considerable (12%) increase in the acceptance ratio of legitimate packets and negligible (2%) increase in the acceptance ratio of spoofed packets.

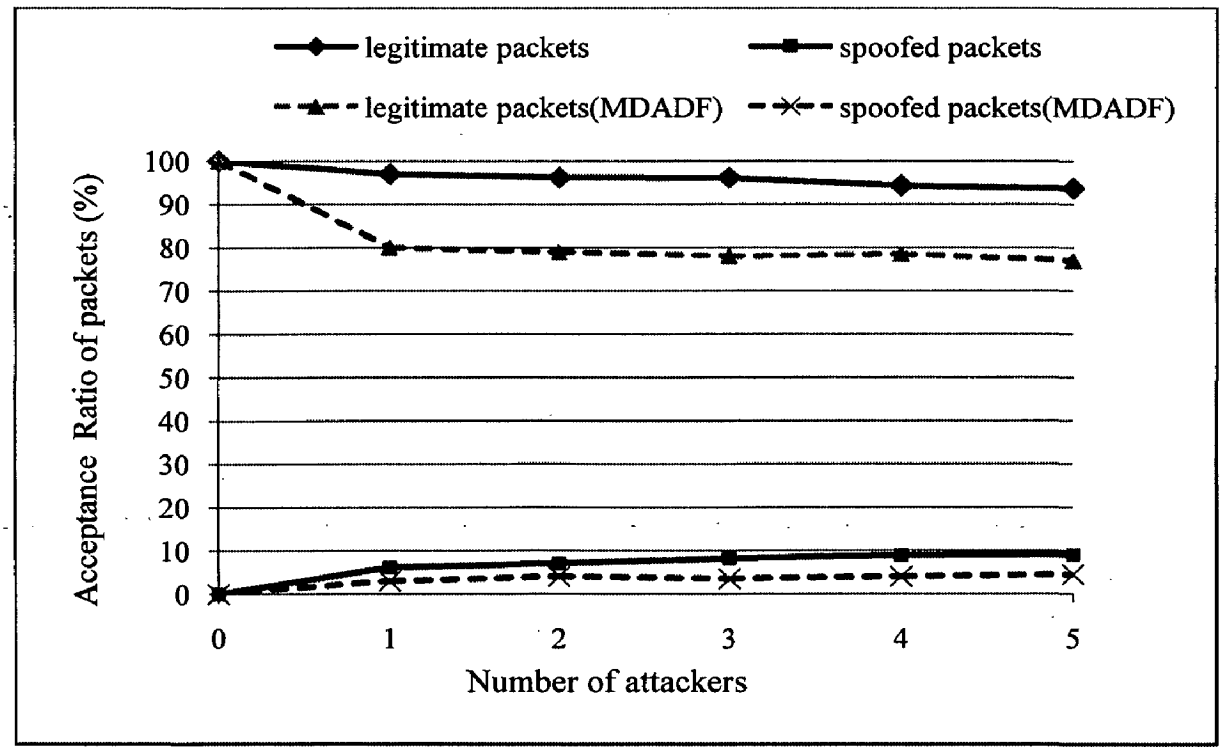


Figure 5.6: Acceptance ratio of packets vs Number of attackers with the threshold 4.

Figure 5.5 shows the acceptance ratio of packets at the firewall under different magnitudes of attack. Here we observe that around 4% of the genuine packets are dropped at the firewall with the threshold value 4.

From the above graphs we can observe that there is a considerable increase in the acceptance ratio of legitimate packets for higher threshold values. But there are some limitations on the threshold value. For higher threshold values, i) the size of the filter table and traceback table will be increased and takes more amount of storage space ii) there is a performance overhead on the firewall because the number of comparisons will become more for every incoming packet. So it is optimal to consider up to a threshold value 4.

In the above graphs we have seen the results of the filtering procedure, which is common for both of our mechanisms. The following section presents the results of the traceback procedure when Bloom-filters are used for traceback.

The following graph in Figure 5.7 shows the Mean false positive rate under different magnitudes of attack and for different threshold values.

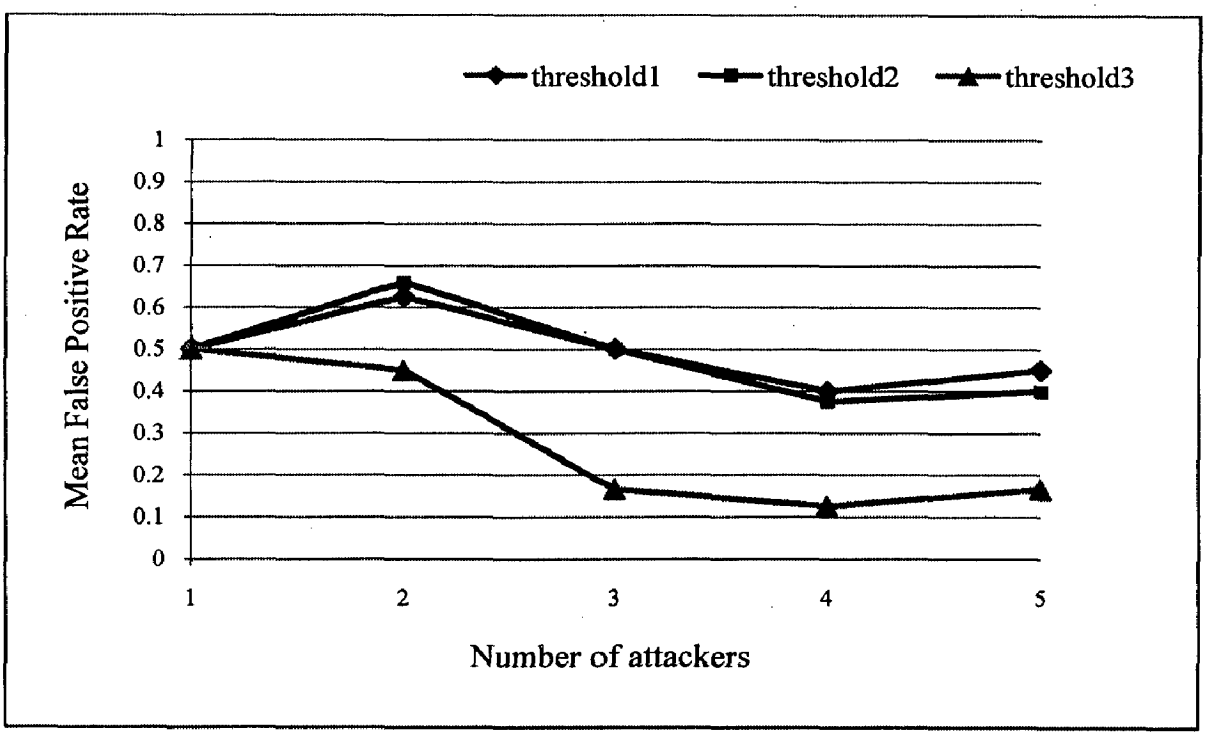


Figure 5.7: Mean false positive rate vs Number of attackers for different threshold values when Bloom-Filters are used for traceback process.

By observing figure 5.7 we say that the false positive rate of the trace routes decreases when threshold value increases.

The following section discusses the results of the traceback procedure when we use PPM strategy for traceback. Unlike the first method, this method for traceback procedure doesn't suffer from false positives, but it suffers from false negatives. This mechanism works well for large scale DDoS attacks (where attacker sends thousands of packets to attack the victim). Figure 5.8 shows the graph between the minimum numbers of packets required to reconstruct the path to source and different number of path lengths.

In PPM strategy each router marks the packet independently with certain probability p. Because of independent nature of marking of each router we are considering equal marking probability p for all the routers. This marking probably of each router is inversely proportional to the length of the path.

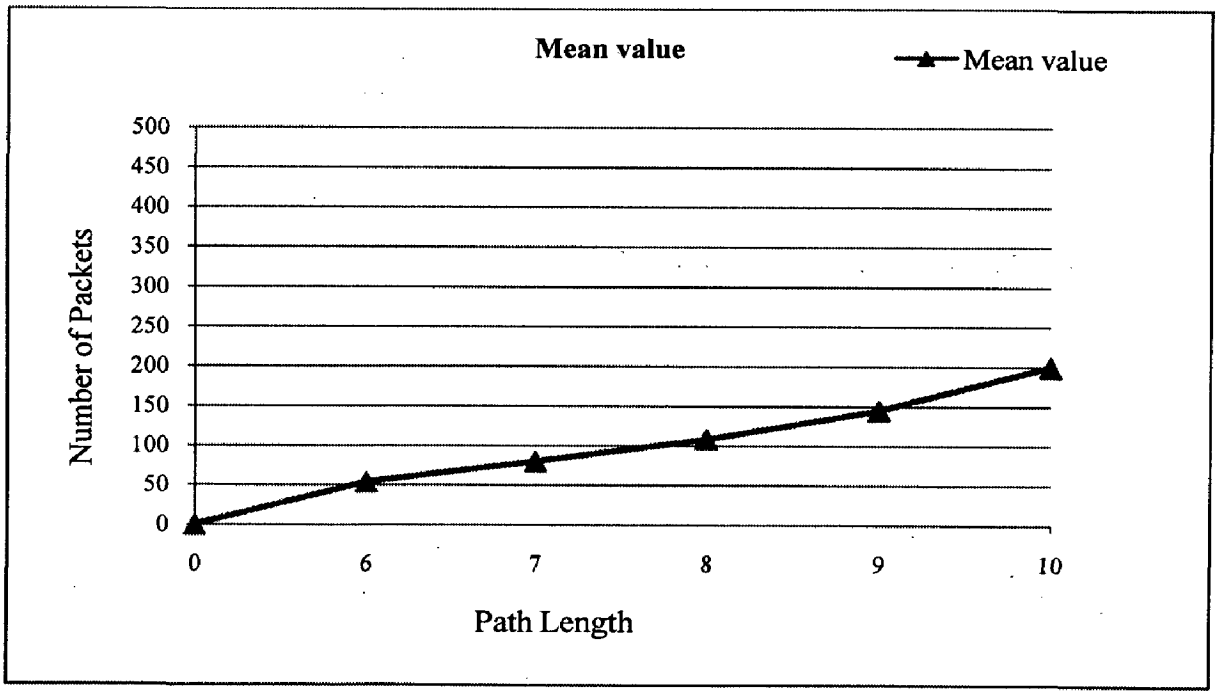


Figure 5.8: Number of packets needed to reconstruct the path vs different path lengths when Probabilistic Packet Marking (PPM) strategy is used for traceback process with k=2.4.

If marking probability is p and distance between the attacker and victim is d hops then

p is inversely proportional to d, which is given by,

$$p \propto \frac{1}{d} \qquad (1)$$

Adding a proportionality constant k to the above relation, we get,

$$p = k \times \frac{1}{d} \qquad (2)$$

From experimental results by using trial and error method it was observed that for k = 2.4, we get minimum number of packets that are needed to reconstruct the path to source for different path lengths.

Figure 5.7 shows the minimum number of packets needed to reconstruct the path when the path lengths are varied between 6 and 10 hops. We have taken the equal probability for all the routers to mark the packets depending on the distance with proportionality constant k=*2.4*.

Chapter 6

CONCLUSIONS AND FUTURE WORK

6.1. Conclusions

The scheme proposed as part of our dissertation provides a solution for defence against flooding-based IP spoofed DDoS attacks. The effectiveness of the proposed scheme is illustrated by appropriate simulation testbed in Chapter 5.

Our proposed mechanism has overcome the following limitations which were found in existing techniques.

- ❖ It combines both the features of filtering and traceback mechanisms.
- ❖ It greatly reduces the processing overhead on the intermediate routers by using simple calculations.
- ❖ By considering multiple routes (paths i.e., marks) for the packets that originate from the same source, we minimized the drop rate of legitimate packets to 4% while filtering the IP spoofed packets and also minimized the false positive rate to 2.5 when tracing back to the source of attack.

By using this mechanism we can detect the attack, filter out the IP spoofed packets and traceback to the attack source.

Our simulation results show that the higher the threshold value (max no of marks considered against same source IP address) the lower is the drop rate of legitimate packets, false positives and the higher is the acceptance rate of spoofed packets.

Our first method, which uses Bloom-Filters for traceback procedure is best suitable for small scale as well as large scale DDoS attacks because each router on the path participates in the marking process. In this method, by using a single packet we can traceback to the source of attack. The second method, which uses PPM strategy for traceback procedure is best suitable for large scale DDoS attacks and it under performs small scale DDoS attacks because at most one router on the path will mark the packet during its journey from source to destination. So we need to sample more

number of packets to get the information about all the routes that are there on the path in order to reconstruct the path.

Table 6.1 shows the comparison between the proposed methods and the existing mechanisms based on different parameters.

The following are the different parameters we have taken to compare the proposed and existing mechanisms.

I. Supports Filtering **II.** Supports Traceback **III.** Supports Pushback **IV.** Works well with Small scale DDoS attacks **V.** Works well with Large scale DDoS attacks **VI.** Works fine for any path length

Method	I	II	III	IV	V	VI
SPIE [20]		X		X	X	
PI [15]	X				X	X
MDADF [22]	X		X	X	X	X
Proposed scheme using Bloom-Filters for traceback procedure	X	X	X	X	X	
Proposed scheme using PPM strategy for traceback procedure	X	X	X		X	X

Table 6.1: Comparison of various existing schemes with the proposed schemes

From Table 6.1 we can conclude that our proposed mechanisms have overcome some of the major limitations of the existing mechanisms.

6.2. Suggestions for Future Work

The assumptions that the ingress router initializes the filter area of the packet can be discarded and an effective strategy can be implemented to initialize the marking area of the packet when bloom-filters are used for traceback process.

The packet marking mechanisms existed and proposed till now do not support packet fragmentation in the Internet because these mechanisms normally use identification field of the packet header for marking purpose. So the effective strategy can be implemented which supports packet fragmentation in the network.

REFERENCES

[1] S. Moore, "Evolution of the Internet," Electro International 1994, Hynes Convention Center, Boston, MA, pp. 263–265. May. 1994.

[2] Robert Vamosi, "Study: DDoS attacks threaten ISP infrastructure," Online at http://news.cnet.com/8301-1009_3-10093699-83.html, CNET News, Nov. 2008.

[3] Elinor Mills, "Radio Free Europe DDOS attack latest by hactivists," Online at http://news.cnet.com/8301-10784_3-9933746-7.html, CNET News, May. 2008.

[4] Christos Douligeris and Aikaterini Mitrokotsa, "DDoS Attacks And Defencemechanisms: A Classification," in Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology, (ISSPIT'03), pp. 190-193, Dec 2003.

[5] Rocky K. C. Chang," Defending against Flooding-based Distributed Denial-of-service Attacks: A Tutorial," IEEE Communications Magazine, pp. 42-51, Oct. 2002.

[6] Internet System Consortium, "ISC Domain Survey: Number of Internet Hosts," http://ftp.isc.org/www/survey/reports/2008/01/.

[7] Internet World Stats, Internet User Statistics – The Big Picture: World Internet Users and Population Stats, http://www.internetworldstats.com/stats.htm

[8] Steven M. Bellovin, "A Look Back at "Security Problems in the TCP/IP Protocol Suite," Proceedings of the 20th annual Computer Security Applications conference(ACSA'04).

[9] Matthew Tanase, "IP-spoofing: an Introduction," Online at http://www.securityfocus.com/infocus/1674 , Mar. 2003.

[10] S. M. Khattab, C. Sangpachatanaruk, R. Melhem, D. Mosse, and T. Znati, "Proactive Server Roaming for Mitigating Denial-of-Service Attacks," in Proceedings of the 1st International Conference on International Technology: Research and Education (ITRE'03), pp. 286-290, Aug. 2003.

[11] A. Belenky and N. Ansari, "IP Traceback With Deterministic Packet Marking," IEEE Communications Letters, vol. 7, no. 4, pp. 162-164, Apr. 2003.

[12] A. Belenky and N. Ansari, "Tracing Multiple Attackers With Deterministic Packet Marking (DPM)," in IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM'03), vol. 1, pp. 49-52, Aug. 2003.

[13] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Practical Network Support for IP Traceback," in Proceedings of ACM SIGCOMM'00, Aug. 2000.

[14] A. zaddoost, M. Othman, M. F. A. Rasid, "Accurate ICMP Traceback Model Under DoS/DDoS Attack," 15th International Conference on Advanced Computing and Communications, (ADCOM'07), pp. 441-446, Dec. 2007.

[15] A. Yaar, A. Perrig, and D. Song, "PI: A path identification mechanism to defend against DDoS attacks," in proceedings of the IEEE symposium on Security and Privacy, pp. 93-109, May 2003.

[16] P. Feruson and D. Seine, "Network Ingress Filtering: Defeating Denial Of Service Attacks Which Employ IP Source Address Spoofing," RFC2827, May 2000.

[17] D. Dittrich. "The Dos Project's Trinoo Distributed Denial of Service Attack Tool". http://staff.washington.edu/dittrich/misc/trinoo.analysis.

[18] A. Mankin, D. Massey, C. L. Wu, S. F. Wu, and L. Zhang, "On Design And Evaluation of "Intention-Driven" ICMP Traceback," Proceedings in 10th International Conference on Computer Communication and Networks (ICCCN'01), Scottsdale, AZ, USA, pp. 159-165, Oct. 2001.

[19] B. Bloom, "Space/Time Trade-Offs in Hash Coding with Allowable Errors," Comm. ACM, vol. 13, no. 7, pp. 422-426, 1970.

[20] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, B. Schwartz, S. T. Kent, and W. T. Strayer, "Single-Packet IP Traceback,"

IEEE/ACM Transactions on Networking, Vol. 10, No. 6, pp. 721-734, Dec. 2002.

[21] J.Ioannidis and S. M. Bellovin, "Implementing Pushback: Router-Based Defense Against DDoS Attacks," in Proceedings of the Network and Distributed System Security Symposium (NDSS'02), pp. 6-8, Feb. 2002.

[22] Yao Chen1, Shantanu Das, Pulak Dhar, Abdul-motaleb El Saddik, and Amiya Nayak, "Detecting and Preventing IP-spoofed Distributed DoS Attacks," International Journal of Network Security, Vol.7, No.1, pp.70–81, Jul. 2008.

[23] Haining Wang, Cheng Jin, Kang G. Shin, "Defense Against Spoofed IP Traffic Using Hop-Count Filtering," IEEE/ACM Transactions on Networking, Vol.15, No.1, pp. 40-53, Feb. 2007.

[24] S. Itani, N. Aaraj, D. Abdelahad, A. Kayssi, "Neighbor Stranger Discrimination: A New Defense Mechanism Against DDoS Attacks," in Proceedings of the 3rd ACS/IEEE International Conference on Computer Systems and Applications, (ICCSA'05), pp. 95-100, May 2005.

[25] D. X. Song and A. Perrig, "Advanced And Authenticated Marking Schemes for IP Traceback," in Proceedings of 20th Annual joint conference of the IEEE Computer and Communications scientists, (INFOCOM'01), Anchorage, AK, USA, pp. 878-886, Apr. 2001.

[26] H. Burch and B. Cheswick, "Tracing Anonymous Packets to Their Approximate Source," Proceedings of the 14th USENIX conference on System Administration Conf. (LISA '00), pp. 319-328, Dec. 2000.

[27] G. Sager, "Security Fun with OCxmon and cflowd," Presentation at the Internet2 Working Group Meeting, http://www.caida.org/funding/ngi1998/content/ security/ 1198/, Nov. 1998.

[28] T. Peng, C. Leckie, and K. Ramamohanarao, "Adjusted Probabilistic Packet Marking For IP Traceback," in Networking 2002, pp. 697-708, May 2002.

[29] Takurou. H, Matsuura. K, Imai. H, "Traceback by Packet Marking Method with Bloom Filters," 41st Annual International Conference on Security technology, pp. 255-263, Oct. 2007.

[30] H. Lipson, "Tracking and Tracing Cyber-Attacks: Technical Challenges and Global Policy," technical report, Software Eng. Inst., Carnegie Mellon Univ., Nov. 2002.

[31] S McCreary and K. C. Claffy, "Trends in Wide Area IP Traffic Patterns: A View from Ames Internet Exchange," CAIDA, La Jolla, CA, USA, tech. Rep. AIX005, [Online]. Available: http://www.caida.org/publications/papers/2000/AIX0005/

[32] D. Dittrich, "The Tribe Flood Network distributed denial of service attack tool". Online at http://staff.washington.edu/dittrich/misc/tfn.analysis.txt.

[33] N. Long, S. Dietrich and D. Dittrich, "An Analysis Of The "Shaft" Distributed Denial of Service Attack Tool," In Proceedings of LISA'00, 2000. Online at http://home.adelphi.edu/~spock/shaft_analysis.txt

LIST OF PUBLICATIONS

[1] Krishna Parachikapu, Sarje A. K, "A Hybrid Approach To Filter And Traceback IP-Spoofed Packets In DDoS Attacks," In the Proceedings of 11th International Conference on Distributed Computing and Networking (ICDCN'10), Kolkata, India, Jan. 2010. (communicated).