Aalto University
School of Science
and Technology

Department of Communications and Networking

Firoozeh keshvari Ghalati

DEFENDING AGAINTS DISTRIBUTED DENIAL OF SERVICE ATTACK UNDER TUNNEL BASED FORWARDING

Supervisor: Professor Raimo Kantola

Aalto University Abstract of Master's Thesis

Author:	Firoozeh Keshvari Ghalati	
Name of the Thesis:	Defending against Distributed Denial of Service Attack Under Tunnel Based Forwarding	
Date:	8.6.2011	**Pages:** 61
Department:	Department of Communications and Networking	
Professorship:	S-38 Networking Technology	
Supervisor:	Prof. Raimo Kantola	
Instructor:	Dr. Markus Peuhkuri	

Abstract:

Today, attacks are a harmful element of the computer networks. Distributed Denial of Service (DDoS) attack is one of the most harmful attacks. Many defense mechanisms have been proposed to mitigate the effect of the attacks. 2In this thesis, we study two methods for defending against DDoS attacks.

First, we identify the attack packets to detect a DDoS attack by checking the TTL value of incoming packets and monitoring the number of new source IP addresses of incoming packets. Second, we propose an algorithm to traceback the attack traffic to identify the source IP address of origin by deploying a tunneling based protocol.

The tunneling based protocol is called the Locator/Identifier Separation Protocol (LISP) and it is deployed in a domain network to encapsulate all outgoing packets decapsulate all incoming packets. As a side-effect the tunneling protocol reveals the ingress point of attack traffic.

We also analyzed the approach in a simulation environment and compare the results in the domain network when deploying the tunneling based protocol.

Keyword: Distributed Denial of Service Attack, Traceback, TTL, LISP Protocol

Aalto Korkeakoulu — Diplomityön tiivistelmä

Tekijä: Firoozeh Keshvari Ghalati

Työn nimi: Suojautuminen Distributed Denial of Service Attackia Vastaan Tunnelointiin Perustetun Uudelleen Lähettämisen Ympäristössä

Päivämäärä: 8/6/2011 **Sivumäärä:** 61

Osasto: Sähkö- ja tietoliikennetekniikan osasto
Professuuri: S-38 Tietoverkkotekniikka

Työn valvoja: Prof. Raimo Kantola
Työn ohjaaja: Dr. Markus Peuhkuri

Tiivistelmä:

Tänään hyökkäykset ovat haitallinen ilmiö tietoverkoissa. Hajau Palvelunessa hyökkäys(DDoS) on yksi haitallisista hyökkäyksistä. Monea suojamekanismeja on ehdotettu hyökkäysten vaikutuksen lieventäminen.

Tässä työssä tutkimme kahta menetelmää DDoS hyökkäyksiä vastaan. Ensin, tunnistamme hyökkäyksen paketit, havaitsemme DDoS saapuvat hyökkäyspaketit pakettien TTL arvoa tarkailemalla ja laskemme uusia lähde-IP-osoitteita saapuvissa paketteissa. Toiseksi ehdotamme algoritmia, joka jäljittää hyökkäysliikennettä ja tunnistaa lähde IP-osoitetta hyödyntämällä tunnelointiprotokollaa.

Tunnelointiprotokollaa kutsutaan Locator/ID Split Protokollaksi eli (LISP). Se on sijoitettu alueverkkoon ja sen tehtävä on lähtevien pakettien kapselointi ja saapuvien pakettien dekapselointi. Sivuvaikutuksena on, että tunnelointiprotokolla paljastaa hyökkäysliikenteen sisätulopisteen alueverkkoon. Olemme myös analysoineet tätä menetelmää simulointiympäristössä

Acknowledgement

I would like to thank my supervisor, Professor Raimo Kantola, for his suggestions, generous help and support.

I also want to thank my instructor, Dr. Markus Peuhkuri, for his support and help.

I want to thank Mrs Anita Bisi, in the international student affairs office, for her kindness and generous help during my study in the HUT University and my stay in Finland.

And finally, I want to thank my husband and my little girl for their patience during my study.

TABLE OF CONTENTS

ACKNOWLEDGEMENT....IV

LIST OF FIGURES....XII

CHAPTER 1....1

INTRODUCTION....1

1.1 MOTIVATION.... 1

1.2 THE PROBLEM.... 3

1.3 THE SCOPE AND METHODOLOGY.... 4

1.4 THE STRUCTURE OF THE THESIS....4

CHAPTER 2....5

AN OVERVIEW OF DENIAL OF SERVICE ATTACKS....5

2.1 DENIAL OF SERVICE ATTACK....5

2.2 DISTRIBUTED DENIAL OF SERVICE ATTACK....6

2.2.1 DISTRIBUTED DENIAL OF SERVICE ATTACK ARCHITECTURE....7

2.2.1.1 AGENT-HANDLER MODEL....7

2.2.1.2 DISTRIBUTED REFLECTOR DENIAL OF SERVICE ATTACK MODEL....8

2.2.1.3 IRC BASED DDOS ATTACK MODEL....9

2.3 ATTACK MECHANISMS....10

2.3.1 COORDINATION OF DDOS AGENTS....10

2.3.2 IP SPOOFING....10

2.3.3 FLOODING DDOS ATTACKS....11

2.3.3.1 ICMP FLOODING-BASED ATTACK (SMURF)....12

2.3.3.2 TCP SYN FLOODING-BASED ATTACK....12

2.3.3.3 UDP FLOODING-BASED ATTACK....13

2.4 LIMITING FACTORS TO IP SPOOFING....13

2.4.1 INGRESS FILTERING....13

2.4.2 REVERSE PATH FORWARDING (RPF)....14

2.4.3 NETWORK ADDRESS TRANSLATION (NAT) 15

2.5 SUMMARY 15

CHAPTER 3 16

DOS ATTACK DEFENSE PROPOSALS 16

3.1 INTRUSION DETECTION SYSTEMS 16

3.2 DISTRIBUTED DENIAL OF SERVICE DETECTION METHODS 18

3.2.1 DOS-ATTACK-SPECIFIC DETECTION 19

3.2.1.1. DETECTION OF SYN FLOODING ATTACKS 19

3.2.1.2 MIB 20

3.2.1.3 MULTOPS 20

3.2.1.4 DISCUSSION 21

3.2.2 ANOMALY-BASED DETECTION 21

3.2.2.1 ARTIFICIAL IMMUNE SYSTEM (AIS) 21

3.2.2.2 STATISTICALLY-BASED ANOMALY DETECTION 22

3.2.2.3 DISCUSSION 22

3.2.3 TRAFFIC-VOLUME-BASED DETECTION 23

3.2.3.1 AGGREGATE-BASED CONGESTION CONTROL (ACC) 23

3.2.3.2 DISCUSSION 23

3.2.4 IP ATTRIBUTES-BASED DDOS DETECTION 24

3.2.4.1 SOURCE IP ADDRESS MONITORING 24

3.2.4.2 TTL 24

3.2.4.3 COMBINATION OF MULTIPLE ATTRIBUTES 25

3.2.4.4 ENTROPY 25

3.2.4.5 HOP-COUNT FILTERING 26

3.2.4.6 DISCUSSION 26

3.3 ATTACK SOURCE IP IDENTIFICATION 27

3.3.1 ACTIVE IP TRACEBACK 28

3.3.2 PACKET MARKING 29

3.3.2.1 ICMP BASED MARKING 29

3.3.2.2 PACKET MARKING IN IP HEADER 29
3.3.3 LOGGING 31
3.3.4 HASH-BASED IP TRACEBACK 31
3.4 SUMMARY 32
CHAPTER 4 33
INTRODUCTION TO LISP PROTOCOL 33
4.1 INTRODUCTION LISP PROTOCOL 33
4.2 OVERVIEW OF LISP 34
4.2.1 LISP DATA-PLANE OPERATION 35
4.2.2 LISP CONTROL PLANE 36
CHAPTER 5 37
TUNNELING BASED IP TRACEBACK SYSTEM 37
5.1 INTRODUCTION 37
5.2 TUNNELING BASED NETWORK TOPOLOGY 38
5.3 TUNNELING BASED IP TRACEBACK ARCHITECTURE 39
5.3.1 DETECTOR 41
5.3.2 LOGGER 42
5.3.3 SEARCHING DEVICE 42
5.3.4 MANAGER 43
5.4 TUNNELING BASED IP TRACEBACK PROCESS 43
5.5 TUNNELING BASED IP TRACEBACK DESIGN MODEL 44
5.5.1 DESIGN OF DETECTOR 44
5.5.2 DESIGN OF PACKET LOGGER 46
5.5.3 DESIGN OF SEARCHING DEVICE 47
5.5.4 FINDING ATTACKER IN CORPORATE NETWORK 47
CHAPTER 6 49
EVALUATION AND ANALYSIS 49
6.1 DEPLOYING THE TUNNELING BASED IP TRACEBACK 50
6.2 TUNNELING BASED IP TRACEBACK SIMULATIONS 50
6.2.1 SIMULATION TOOL 50

6.2.2 THE SIMULATION OF TUNNELING BASED IP TRACEBACK PROCESS 51

6.2.3 SIMULATION SCENARIOS AND RESULTS 52

CHAPTER 7 56

CONCLUSION 56

REFERENCES 58

LIST OF FIGURES

FIGURE 2.1 AGENT-HANDLER MODEL OF DDOS ATTACK 8

FIGURE 2.2 REFLECTOR MODEL OF DDOS ATTACK 9

FIGURE 2.3 INGRESS FILTERING 14

FIGURE 3.1 THE HEADER OF THE IP PACKET 30

FIGURE 4.1 LISP HEADER FORMAT 36

FIGURE 5.1 TUNNELING BASED IP TRACEBACK NETWORK TOPOLOGY 39

FIGURE5.2 THE BIG PICTURE OF TUNNELING BASED IP TRACEBACK PROCESS *42*

FIGURE 5.3 PROCESS OF TUNNELING BASED IP TRACEBACK 45

FIGURE 6.1 THE END-TO-END DELAY IN TWO MODES.......... 54

FIGURE 6.2 TRAFFIC RECEIVED IN TWO MODES.......... 55

FIGURE 6.3 THE QUEUE SIZE OF THE ETR IN TWO MODES 55

Chapter 1

Introduction

1.1 Motivation

The Internet (known as ARPANET) was created in 1969 by Advanced Research Project Agency (ARPA) of the Department of Defense (DOD) in the United States of America [1]. Since the internet at that time was created for researchers to share their network and resources on their research, it was designed for openness and stability in a way that everyone on the Internet could send packets and the receivers should receive and process any packets. The result of this method of design is poor security. On the other hand, today, the Internet is not just a tool for researchers, it is the main infrastructure of sharing information and for this reason security and reliability of the Internet are more important than ever before and they are now one of the main concerns of national security. Unfortunately, with the growth of the Internet, attacks on the Internet can be launched anywhere and due to the openness of the Internet, they have also increased quickly. The lack of authentication helps attackers to create and send malicious traffic or fake identity. All systems connected to the Internet are potential targets for attackers since the openness of the Internet makes them accessible to attackers.

Denials of Service (DoS) attacks have become a major security threat to Internet services, as they aim to stop the service provided by a target. A Denial of Service (DoS) attack can be characterized as an attack with the purpose of preventing legitimate users from using a victim computing system or network resource [2]. A Distributed Denial of Service (DDoS) is a kind of DoS attack in which multiple sources send attack traffic to the target, while in a DoS attack the traffic to the target comes from one source.

DoS and DDoS attacks can be classified to two forms. The first one is crashing a system by sending malformed packets, while in the second form of attack the attacker sends a massive volume of useless traffic to occupy all resources of the target, making it inaccessible. DDoS attacks are more dangerous than DoS attacks and cannot be prevented easily because, by using multiple sources, the power of the attack is amplified and defense is made more complicated; also attackers in DDoS attacks often use unreal source IP addresses.

The first DDoS attack was reported in the summer 1999[3]. Numerous DDoS detection and numerous response techniques have been proposed, however, they are often unreliable or imperfect in detecting a DDoS attack for two main reasons. The first is that it is difficult to distinguish between DDoS attack traffic and normal traffic. The second is that the sources of DDoS attacks are hard to find in a distributed network.

When an attack has been detected, the first reaction can be tracing the real attacker. It is sometimes impossible to find the real attacker as so many compromised hosts are involved in the attack; also, most attackers use an unreal source IP address when they send attack traffic toward the target. This means that they put fake source addresses in the packet that they send out. In this case, the logs of all of the intermediate routers must be examined one by one to trace the attack path. The Internet was simply not designed with these vulnerabilities in mind, and a real solution would involve re-engineering the entire network architecture. This thesis presents techniques for defending against DDoS attacks by finding the real attacker or its agents.

1.2 The Problem

Finding a new solution to detect the origin of the attack traffic in DDoS attacks is the main motivation of this thesis. This leads to selecting an effective algorithm to detect the DDoS attack first and then selecting a new method to find the real attacker. Once an attack has been detected, an ideal response would be to block the attack traffic at its source. Since most attackers use source IP address spoofing, and also IP routing is stateless and in a DDoS attack a number of compromised hosts are involved, there are no simple methods to track IP traffic to its source. In order to address this limitation, numerous schemes have been proposed based on enhancing router functions to support IP traceability. In this thesis we try to improve existing methods in case when IP tunneling is used to carry packets edge by edge. For finding the source IP address of the host involved in the attack, we need to know the IP address of the edge router (ingress point). In a tunneling based network, the edge router (ingress point) inserts its IP address into the packet and encapsulates the packet by using the LISP protocol and forwards it to the victim's network. As a result, the victim or the victim's egress router sees the address of the ingress point. We first explore under what conditions we can trust that source address and how we can utilize it to carry out the traceback.

1.3 The Scope and Methodology

The object of this research is to develop mechanisms to detect and react to attacks. These mechanisms should detect the attack quickly and track the attack to the source accurately.

This research particularly studies DoS and DDoS attacks in computer networks based on IP protocol, assuming that many nodes in the network support the LISP protocol and packets in the networks are routed with this protocol. We also analyze the case that some of the Internet traffic is still routed in the traditional way.

1.4 The Structure of the Thesis

The rest of the thesis is organized as follows:

Chapter 2 gives an overview of denial of service attacks. In this chapter, DoS and DDoS attacks and attack mechanisms are briefly described.

Chapter 3 describes DDoS attack defense proposals. In this chapter, we describe DDoS attack defense mechanisms and detection methods. Then some of the mechanisms for IP source address identification are described briefly.

Chapter 4 introduces the LISP protocol. Chapter 5 presents a tunneling based IP traceback system. In this chapter an approach for detecting DDoS attack by checking the TTL value of incoming packets and monitoring the amount of new source IP addresses against a threshold and then tracing back to the spoofed source IP address is described. Chapter 6 evaluates and analyzes the results in different scenarios. Finally, we conclude with a summary of contributions in Chapter 6.

Chapter 2

An Overview of Denial of Service Attacks

This chapter presents an overview of denial of service attacks, types of DoS attacks and discusses the architecture of DoS and DDoS attacks.

2.1 Denial of Service Attack

A Denial of Service(DoS) attack is an attack that renders a computer or network incapable of providing service to legitimate users so the access to a network or server is blocked or degraded.

There are two types of DoS attacks. In the first type, the attacker is attempting to block or degrade the service provided by the victim by exploiting the software and protocol vulnerabilities in the system [4]. The second type of DoS attack is bandwidth attack [4].

The first type of attack can happen on the network level, and use bugs in the software or try to exhaust the hardware resources of the network devices. It may even happen in the Operating System (OS) level. Sometimes it takes advantage of bugs in the net applications that are running on a host and use the resources of the victim [5].

The second type of DoS attack known as bandwidth attack is based on sending a large volume of attack traffic to the victim in order to use the bandwidth of the victim, causing it to process a large volume of malicious traffic instead of giving service to legitimate users [6].

DoS attacks can be launched against services such as a web server, or networks such as the network connection to a server. The effect of a DDoS attack is more considerable when the targets are companies that rely on their online services to conduct business. On February 9, 2000, Yahoo, eBay, Amazon.com, E*Trade, ZDnet, Buy.com, the FBI, and several other Web sites were the target of DoS attacks and they met substantial damage and inconvenience [7]. So, it is important to deter the damage caused by DoS attacks.

2.2 Distributed Denial of Service Attack

A distributed denial-of-service (DDoS) attack is a DoS attack which relies on multiple compromised hosts in the network to attack the victim. By using multiple compromised hosts in the attack as agents, the attacker can launch a more dangerous and complicated attack. Like what was previously mentioned for DoS attack types, there are two types of DDoS attack. In the first type, the attacker is attempting to block or degrade the service for legitimate users by exploiting software [5]. In the second type, which is known as bandwidth attack, the attacker sends a large amount of attack traffic to congest the network resources of the victim [8]. Since in a DDoS attack, an attacker uses several compromised hosts to launch an attack, two kinds of victim can be categorized. The services under attack are those of the "primary victim", while the compromised systems used to launch the attack are often called the "secondary victims" [8]. The use of secondary victims makes the DDoS attack more complicated because tracking and finding the real attacker becomes more difficult.The DDos attack has two different methods for sending attack traffic to the victim. In the first, an attacker compromises a number of agents to send attack traffic to the victim, while in the second one the attacker uses reflectors which send replied packets to the victim. These two types will be explained in more detail later [9].

A DDoS attack is composed of four elements [10]:

1. The real attacker.
2. The handlers or master compromised hosts that are capable of controlling multiple agents.
3. The attack agents or zombie hosts that create a large amount of traffic toward the intended victim.
4. A victim.

In a DDoS attack the attacker chooses the agents which perform the attack. The attacker then exploits the Trojan software of the agents and plants the attack code, protecting it simultaneously from discovery and deactivation. Next the agents inform the attacker via handlers that they are ready. The attacker commands the onset of the attack. Some DDoS powerful toolkits that are a kind of DDoS application system are available to potential attackers increasing the danger of becoming a victim in DDoS attack. Some of the most well-known DDoS tools are Trinoo, TFN, Stacheldraht, TFN2K, Mstream and Shaft [10].

2.2.1 Distributed Denial of Service Attack Architecture

The attacker must cooperate with all of its DDoS agents before the traffic attack reaches the victim. Therefore, there must be control channels between the agents and the attacker [4]. This means that all agents send attack traffic after they have received commands from the attacker through the control channels. A DDoS attack network [11], which consists of an attacker, agents, and the control channels, are divided into three types: the agent-handler model, the Internet Relay Chat (IRC)-based model, and the reflector model, which are explained in more detail in the following sections.

2.2.1.1 Agent-Handler Model

An Agent-Handler DDoS attack network, as it is shown in Figure 2.1, consists of the attacker, the handlers, and the agents.

The handlers are software packages that the attacker uses to communicate with agents [6]. These software packages are located on compromised routers or servers that allow the attacker to send control messages to agents, instructing them to send massive attack traffic to the victim. Depending on how the attacker configures the DDoS attack network, agents can be instructed to communicate with a single handler or multiple handlers. The communication between the attacker and handler and between the handler and agents can be via TCP, UDP, or ICMP protocols.

As mentioned previously, this type of attack network is hard to prevent, as agents have no knowledge that their system has participated in the attack. In the Agent-Handler model each agent program that launches the attack uses only a small amount of agent resources both in memory and bandwidth; consequently they experience minimal change in performance. The agents that have been violated to run the agent software are referred to as the secondary victims, while the target of the DDoS attack is called the primary victim [6].

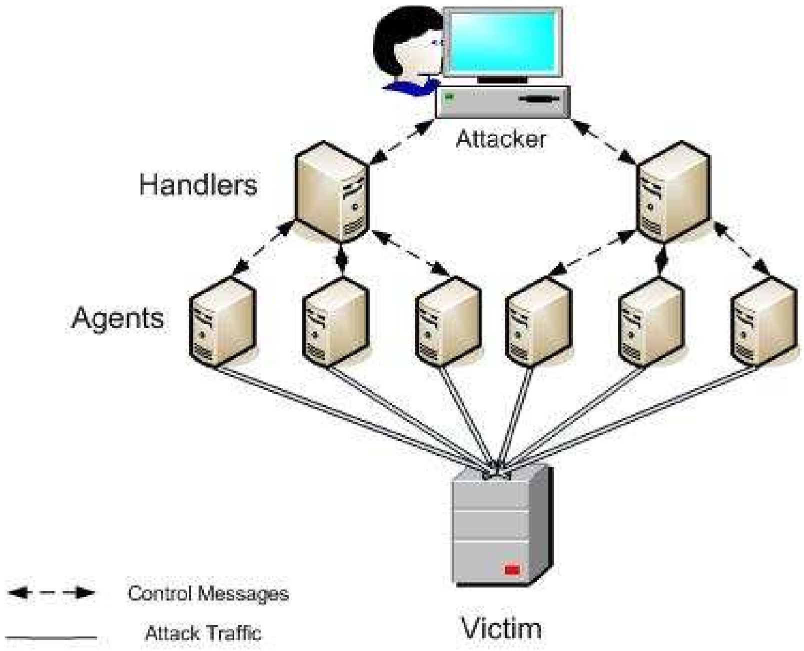


Figure 2.1: Agent-handler Model of DDoS Attack

2.2.1.2 Distributed Reflector Denial of Service (DRDoS) attack Model

The DRDoS has three phases to launch a DDoS attack. The first phase functions as DDoS attack. In the second phase, when the attacker has gained control of the agents, the agents are obliged to send traffic to the third parties that are reflectors as it is shown in the Figure 2.2, and use the victim’s IP address as the source IP address of spoofed traffic to the third parties. Finally, in the third phase, the third parties send reply traffic to the victim, which constitutes the DDoS attack.

Unlike some types of DDoS attacks, the reflector does not need to serve as an amplifier which can broadcast messages to all IP addresses in its subnet [12]. So, the reflectors can serve their legitimate users while also being members of an attack network. Since the attacker does not need to compromise reflectors, a reflector can be any host that returns a response if it receives a request. Therefore, a DRDoS attack just needs a small number of agents to compromise and a sufficient number of reflectors to send reply messages to the victim [13].

It is clear that DRDoS attacks are more dangerous than DDoS attacks because in DRDoS attacks, the attack traffic is more widely distributed by using third parties; also, the distributed reflector denial of service attack has the ability to amplify the attack by broadcasting messages to all IP addresses in its subnet.

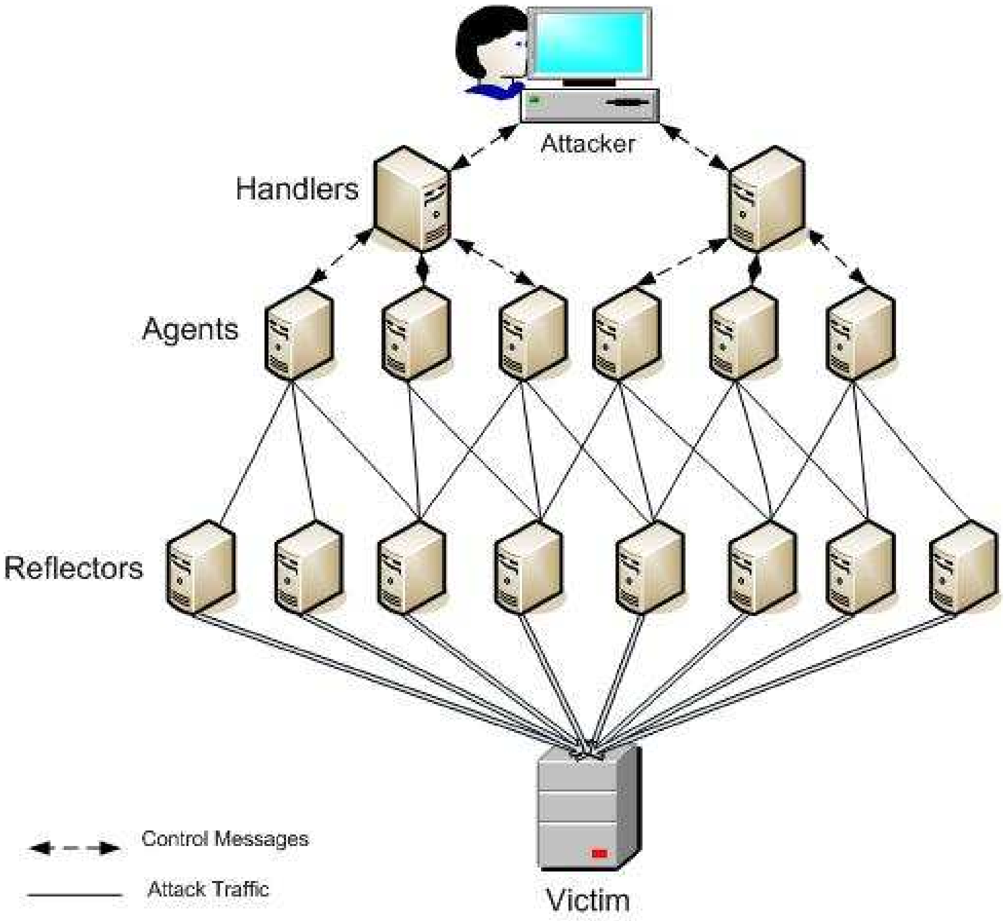


Figure 2.2: Reflector Model of DDoS Attack

2.2.1.3 IRC Based DDoS Attack Model

The architecture of an IRC-based model is not much different from that of the
Agent-handler model, except that instead of communication between an attacker and agents based on handlers, an IRC communication channel is used to connect the attacker to the agents [11].

2.3 Attack Mechanisms

An actual attack will consist of a flooding or logic attack against a single victim. An attacker uses different mechanisms to launch an attack against a particular target. Some of those are listed below.

2.3.1 Coordination of DDoS Agents

For a DDoS attack, the attacker should coordinate all DDoS agents to attack a victim [4] by sending an attack command to every agent through a control channel. There are several choices for transmitting this control channel information such as IRC channels, web-based channels, and specific peer to peer protocols [4].

2.3.2 IP Spoofing

The basic protocol for sending data over the Internet is Internet Protocol (IP). The valid IP address is a unique address assigned to a computer connected to the Internet. A valid IP address must be in the form of xxx.xxx.xxx.xxx where xxx is a number between 0-255. There are a few reserved addresses that cannot be used such as 10.x.x.x, 192.168.x.x, 172.16.0.0 that are reserved for private usage. Using theses private IP addresses in the Internet may cause invalid IP address error message. The header of each IP packet contains the source and destination IP address. The source IP address is the source that IP packet was sent from. Sometimes sender tries to hide its real source IP address by forging the header. So, the header contains a different source address and receiver will send a response to the forged source IP address. This mechanism which is called IP spoofing is a mechanism that attackers use in most DDoS attacks to hide the real source IP address of the agents and the attacker and make the tracing process more complicated. The real source address in an IP packet can be replaced by the addresses of existing hosts or even non-existing hosts. It is possible to carry out a DDoS attack without IP spoofing if an attacker has compromised enough hosts and agents or if a chain of compromised hosts is used, because in most cases compromised agents and hosts do not have enough information about the real attacker, and tracing the attacker reveals only the identity of these compromised agents and the host who is the secondary victim in a DDoS attack network. Also tracing a chain to find the origin of the attack is very hard to achieve. Also, in a reflector-based DDoS attack, agents must put the victim's address in the source address field which is a kind of IP spoofing mechanism [4].

2.3.3 Flooding DDoS Attacks

Flooding-based attacks can be defined as any activity that disables the service provider (victim) of serving legitimate users by sending a volume of useless traffic to the victim.

The effect of the bandwidth attack can be the consumption of host resources or consumption of the network bandwidth.

The first, consumption of host resources, will block the resources of the victim. Since the victim that can be a server, has limited resources, it will drop incoming packets when the traffic load becomes high to inform the sender to decrease the sending rate, while the attacker takes advantage of this and will increase the sending rate, trying to block the victim's services by sending a large volume of attack traffic. So, the legitimate users decrease their sending rates while the attacker increases its sending rate. Finally, the victim's resources such as memory will be used up and it will be unable to service legitimate users [13].

The second impact, consumption of network resources, is more effective than the first. In this type of flooding-based attack the malicious flows will dominate the communication link, consequently not only legitimate users, but also systems relying on the communication links of the attack path will be disabled [13].

Flooding-based DDoS attacks consist of two types: direct and reflector attacks [14]. In a direct attack, the agents send the Transmission Control Protocol/Internet Protocol (TCP), the Internet Control Message Protocol (ICMP), the User Datagram Protocol (UDP), and other types of packets to the victim directly. The response packets from the victim will reach the spoofed receivers due to IP spoofing.

In a reflector attack, presented in Fig. 2.2, the response packets from reflectors truly attack the victim. So the most important aspects of a reflector attack are finding a sufficient number of reflectors and setting the victim address in the source field of the IP header.

DDoS attacks are found in several different forms and are classified based on the algorithms and methods they use to override networks. Among the most important types of DoS/DDoS attacks, ICMP flooding, TCP SYN flooding and UDP flooding can be named.

2.3.3.1 ICMP Flooding-based Attack (Smurf)

ICMP flooding-based attack is a kind of reflector attack in which the source address field of ICMP ECHO REQUEST message is set as the victim IP address. Therefore, the response messages that are an ICMP ECHO REPLY message will be sent to the victim. To make this attack more effective the ECHO REQUEST message can be sent to an amplifier that broadcasts the message to all IP addresses in its subnet. It would be better if Routers turn off the broadcast function to make the risk of Smurf attack lower [9].

2.3.3.2 TCP SYN Flooding-based Attack

When a normal TCP connection starts, a destination host receives a SYN (Synchronization/Start) packet from a source host, creates state for the new session and sends back a SYN ACK (Synchronization/Acknowledge) packet. The destination host must then hear an ACK (Acknowledge) of the SYN ACK before the connection is established. This is called a "TCP Three-Way Handshake". After a connection has been established the real data can be transmitted.

During construction a normal TCP connection server decides the number of memory blocks needed based on the number of received TCP SYN packets. If the server receives a large number of TCP SYN packets, it will run out of memory and this leads the server to be unreachable for legitimate users.

TCP SYN flooding attacks take advantage of this design by generating TCP SYN packets with random source addresses toward a victim. The victim then sends a SYN ACK back to the random source address of the received packets and adds an entry in its connection queue [9]. Since the SYN ACK is destined for an incorrect or nonexistent host, the last part of the "Three-

Way Handshake” is never completed and the victim will run out of memory and this situation will render the victim unreachable for legitimate users. Similar to the ICMP attack, the originator of the attack is difficult to trace as source IP addresses are forged [9].

2.3.3.3 UDP Flooding-based Attack

Another common DDoS scheme is the UDP flooding. The UDP (User Datagram Protocol) is a connectionless protocol, which means that it does not require an established connection to transfer data. In a UDP flooding-based attack, the attacker leads the agent host to send a UDP packet to a random port on the target machine. When the victim receives a UDP packet, it will determine the application on the destination port. If nothing is listening, it will return an ICMP unreachable packet to the forged source IP address notifying the sender that the destination port is unreachable. Therefore, the bandwidth of the victim will be filled and the connection will not be available for legitimate users. All these attacks are based on the spoofed IP address and take advantage of using the faked source IP addresses [9].

2.4 Limiting Factors to IP Spoofing

Some factors are effective on defending against IP spoofing. Some of them are listed below.

2.4.1 Ingress Filtering

Ingress filtering is filtering scheme that filters incoming packets according to a specific rules to make sure that incoming packets are from the network they claim to be from. In ingress filtering, a packet coming into the network is dropped if the network sending it should not use the given source address. Ingress filtering can be done in two levels, from ISP-to-customer and from customer-to-ISP [13]. For customer-to-ISP ingress filtering, all the IP addresses which do not belong to the ISP's network will be filtered, while for customer-to ISP filtering, any internal IP addresses and any private network IP addresses and specific IP addresses will be filtered.

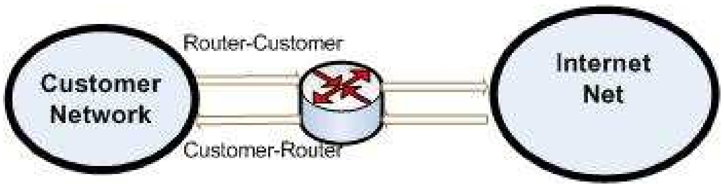


Figure 2.3: Ingress Filtering

Figure 2.3 shows a case of customer-to-ISP filtering. IP addresses other than 171.250.*.* will be filtered. Customer-to-ISP ingress filtering is effective for defending against IP source address spoofing, which is a fundamental weakness of the Internet. Unfortunately, this method is not deployed everywhere.

2.4.2 Reverse Path Forwarding (RPF)

Besides ingress filtering, a limited factor against arbitrary IP address spoofing is the Reverse Path Forwarding (RPF) that must be activated in routers of networks that support multicast. RPF is used in conjunction with multicast routing protocols such as MSDP and PIM-SM to ensure loop-free forwarding of multicast packets. In multicast routing the decision to forward traffic is based upon source address and not on destination address as is the case with unicast routing. When a multicast packet enters a router's interface it will look up the list of networks that are reachable via that input interface i.e., it checks the reverse path of the packet. If the router finds a matching routing entry for the source IP of the multicast packet, the RPF check passes and the packet is forwarded to all other interfaces that are participating in multicast for this multicast group. If the RPF check fails the packet will be dropped. This check is based on the address prefix rather than a full address because most times IP routing entries in routing tables are identified by address prefixes rather than the full 32 bits address. Although, RPF can prevent IP spoofing, but it leaves open the possibility for the attacker to choose the spoofed IP address within the routing prefixes.

2.4.3 Network Address Translation (NAT)

Network Address Translation (NAT) is a way to map an entire network to a single IP address. NAT is a technique for preserving scarce Internet IP address space in a way that networks which are not directly connected to the Internet are often given private address space. When computers on the private network want to communicate on the Internet, the NAT device quickly modifies the packets that they have sent to have a valid source IP address. These packets contain all the addressing information necessary to get them to their destination. NAT is concerned with source IP address and source TCP or UDP port. The NAT gateway will record the changes it makes in its state table to reverse the changes on return packets. When the host replies to the sent packets from the NAT, the NAT gateway will search the state table to determine if the reply packets match an already established connection. Based on the IP/port of the reply packets, a unique match will be found. The NAT gateway will then make the opposite changes it made to the outgoing packets and forward the reply packets on to the internal machine. Since NAT changes the private IP address to valid IP addresses in the Internet, it can be a limitation factor for IP spoofing method.

2.5 Summary

This chapter described what Denial of Services attacks are, how they can be carried out in IP networks and how the Architecture of DoS and DDoS attacks look like. We also described the mechanisms of attack.

Chapter 3

DOS Attack Defense Proposals

There are four steps in the defense against DDoS attacks. The first step is attack prevention, the second step is attack detection, the third step is attack source identification, and the fourth is attack reaction [13]. Attack prevention is a mechanism which stops the attacks before they actually cause damage. One effective mechanism of attack prevention is ingress filtering, which filters incoming traffic according to specified rules.

Attack prevention is difficult to deploy because attackers can easily gain control of a large number of compromised hosts known as agents and direct them to send a large amount of attack traffic with valid IP addresses. Since the communication between the attackers and the agents is encrypted, only the agents, not the attackers, can be exposed which is useful when the attack is launched by the real attacker who can launch an attack with another group of secondary victims.

Attack detection is a mechanism that detects attacks based on several algorithms. To be able to react as soon as possible, the attack should be detected as early as possible. It seems unlikely that reliable and perfect attack detection will be deployed in the near future.

The source identification mechanism helps to find the real attacker and react as quickly as possible. In order to minimize the losses caused by DoS attacks, a reaction scheme must be employed when the attack is underway.

3.1 Intrusion Detection Systems

Intrusion Detection Systems (IDS) are software or hardware or combination of both used to monitor network traffic for intrusive network or host activity. An intrusion detection system

(IDS) monitors system and network resources and activities and, using information gathered from these sources, notifies the authorities when a potential intrusion is identified. Basically, IDS systems fall into two major categories: Host-Based IDS (HIDS) and Network-Based IDS (NIDS). HIDSs are applications which analyze log files and other security related information by using a single host. NIDSs are passive nodes which have access to all traffic in a network link.

All detection systems implement one of two general detection techniques: Statistical-Anomaly-Based and Signature-Based.

A statistical-anomaly-based IDS establishes a performance baseline according to normal network traffic evaluations. It then samples current network traffic activity and evaluates it according to this baseline in order to detect whether or not it is within the baseline parameters. If the sampled traffic is outside the baseline parameters, an alarm will be triggered.

On the other hand, a signature-based IDS examines network traffic for preconfigured and predetermined attack patterns known as signatures. A great number of attacks have distinct, recognized signatures. In good security practice, a collection of these signatures must be constantly updated to prevent or mitigate emerging threats. Signature-based detection systems require a large database that contains information on every packet, and therefore causes much system overhead because the IDS must compare every packet with the signatures in the database. As a result, such systems are not appropriate for high-speed networks and are not effective against new, unrecognized attacks.

Among NIDS protocols and systems, Snort is a well-known example. Snort’s Vulnerability Research Team publishes a set of rules in a file named “ddos.rules”. This file contains a small set of signatures for detecting activity caused by attack traffic.

When Snort works as an offline, passive device, there is little it can do to stop or alleviate a bandwidth-consuming SYN flood. For instance, Snort can potentially report the detection of many SYN segments, but it would not improve the situation. The rules packaged in “ddos.rules” and “bleeding-dos.rules” are designed to either detect DoS agent command-and-control or possibly identify certain types of attacks that subvert but do not breach a target.

When deployed as an inline, active device, Snort acts as a so-called intrusion prevention system and can, in some cases, stop DoS attacks. For example, an intruder may use a malicious packet to cause a vulnerable Cisco router to reboot or freeze. An inline Snort deployment could identify and filter the malicious packet, thereby “protecting” the router. If the intruder switches to a SYN flood or other bandwidth consumption attacks against the router, however, Snort will most likely not be able to counter the attack. As a result, the need for new algorithms and techniques to detect DDoS attacks still exists.

3.2 Distributed Denial of Service Detection Methods

One of the most important steps in defending against DDoS attacks is to detect the attack as early as possible. So we need to determine and establish an optimal methodology to detect an ongoing attack. Most detection mechanisms attempt to detect an attack by observing changes in IP attributes such as traffic patterns or resource usages. These programs are typically configured to detect anomalies or deviations from normal behavior. When anomalies are detected, alerts are created so that either a system administrator or an automated program can quickly determine the type of the attack and decide which actions to take to safely minimize the effects of the attack.

The question is why do we need efficient attack detection? There are some reasons for attack detection. First of all, if a target can detect an attack early, it has more time to implement attack reaction and make the attack less effective. Secondly, it can protect the bandwidth before the attack traffic wastes the network bandwidth. In the case of a large-scale DDoS flooding attack, detecting the attack should be done throughout the network before a large amount of attack traffic crashes the target, which can be a host or a server. In these cases, to defend against a DDoS attack, a distributed defense is essential [10].

So, the most important step against DDoS attack is attack detection. There are several general groups of DoS attack detection techniques and algorithms based on T.Peng et al [13]. One major technique is referred to as DoS-attack-specific detection, which is based on the special features of different sorts of DoS attacks. Another technique is anomaly-based detection, which models the behavior of normal traffic and reports incident anomalies. Traffic-volume-based detection is based on the flooding of sent packets to the target and can be ICMP, UDP and TCP SYN flooding. Finally, IP attribute detection techniques function based on the special features of IP packets [13].

3.2.1 DoS-attack-specific Detection

Generally, the attacker tries to send as much attack traffic as possible to launch a powerful attack. If the victim is unable to reply to all packets, there will be a flow rate imbalance between the source and the victim. Also, attack traffic is created in random pattern to make an attack anonymous.

3.2.1.1. Detection of SYN Flooding Attacks

Wang et al. [15] proposed the SYN detection method to detect a SYN flood attack. When a normal TCP connection starts, a destination host receives a SYN (Synchronization/Start) packet from a source host and sends back a SYN ACK (Synchronization/Acknowledge) packet. The destination host must then hear an ACK (Acknowledge) of the SYN ACK before the connection is established. This is referred to as the “TCP Three-Way Handshake”.

Upon receiving a SYN packet, the server returns a SYN/ACK packet to the client and the connection remains half-open until the client sends the ACK. In SYN flooding attacks, the victim server will never receive the final ACK packet to complete the three-way handshake.

In normal condition, one appearance of SYN packet results in the eventual return of a FIN (RST) packet. But under a SYN flooding attack, this SYN-FIN (RST) pair’s behavior will be violated compared to a normal situation. So, the detection algorithm [15] is based on the statistical change when an attack happens. Two types of packet pairs can be used to detect SYN flooding attacks: SYN vs FIN and SYN vs ACK. where a SYN attack starts, there will be more SYN packets than FIN and ACK. So, an attack should be reported when the number of SYN vs FIN or SYN vs ACK meets a threshold.

3.2.1.2 MIB

J. Cabrera et al. [16] proposed a method to detect DDoS attacks which is called a Management Information Base. In this method, network management information is used to detect DDOS attacks. Management Information Base (MIB) is a database located in network nodes to store information about network devices. SNMP is used to access that database. Local SNMP agents update variables in MIB periodically. When the traffic is sent to these network devices, the administrators of the network can view the MIB variables. If these variables have some correlation on a sequential timeline, an attack may be indicated. For example, in ICMP ping floods, attackers send out ICMP echo requests in which the IP variable in MIB is “ipOutRequest” and, later, the receivers reply with an ICMP Echo in which the same set of variables contain “icmplnEchos”.

The detection algorithm queries the values of several specific MIB variables from local network devices periodically and correlates the relationship of these values [16].

3.2.1.3 MULTOPS

T.M Gil [21] proposed a traffic-volume-based method called MULTOPS to detect a DDoS attack. MULTOPS uses disproportional packet rates to or from the host and subnets as a heuristic to detect attacks. To collect these statistics a tree-shaped data structure keeps track of packets to or from subnets and hosts that show disproportional behavior. MULTOPS has two modes: victim-oriented and attacker-oriented. In the victim-oriented mode, MULTOPS attempts to identify the IP address of the victim and in attacker-oriented MULTOPS, tries to find the IP address of the attacker.

MULTOPS introduces a query that returns the R(P), which is the ratio of forward packets with the destination IP address prefix P to reverse packets with source IP address prefix P. In the victim-oriented mode, MULTOPS determines a victim’s IP address by looking for prefixes for which R(P) is greater than a threshold. In the attacker-oriented mode, MULTOPS determines the addresses of attackers by looking for prefixes for which R(P) is less than a certain threshold [21].

The MULTOPS method has some disadvantages. It assumes that the incoming packet rate is proportional to outgoing packet rate, which is not always the case. For example, real audio/video streams are highly disproportional. Also, MULTOPS is vulnerable to attacks with randomly spoofed IP source addresses.

3.2.1.4 Discussion

MULTOPS assumes that the incoming packet rate is proportional to outgoing packet rate, which is not always the case. Where the packet rate from the server is much higher than from the client, false positive rates will happen. Also, MULTOPS is vulnerable to attacks with randomly spoofed IP source addresses.

SYN and Batch Detection the detection scheme is based on the fact that a SYN packet will end with a FIN or RST packet during normal TCP connection. When the SYN flood starts, there will be more SYN packets than FIN and RST packets. The attacker can avoid detection by sending the FIN or RST packet in conjunction with the SYN packets.

Accurate statistical models based on the MIB parameters from routers are still being studied to understand how accurately they can monitor DDoS attack traffic and predict when a DDoS attack is happening.

3.2.2 Anomaly-based Detection

Anomaly-based detection detects the attack if the monitored traffic behavior does not match the normal traffic profile that is built using training data. Anomaly-based detection can detect new attacks.

3.2.2.1 Artificial Immune System (AIS)

Building a normal profile is the first step for all anomaly-based detection methods. The general idea for AIS-based network intrusion detection proposed by J.L. Bebo et al. [18] includes the following four steps: First, each IP packet is reduced to a string as its identity. This string can contain the source IP address, destination IP address and destination port number. Second, during the training period, all packets that occur frequently are considered as a self string (normal profile). Third, based on the self string, detector strings are created such that they do not match any self string. Finally, when the number of detector string (the number of incoming packets with new source IP address or destination port number in a given time) reaches a certain threshold, an attack is reported.

3.2.2.2 Statistically-Based Anomaly Detection

Manikopoulos et al. [19] propose a new method for detecting DDoS attacks
by use of statistical preprocessing and neural network classification. For anomaly-based detection techniques the most important step is building a normal profile. Statistically-based anomaly detection includes two major steps. The first step is responsible for finding the effective parameters to generate similarity measures, while the second step is calculating the distance between the expected normal traffic and the monitoring traffic based on the normal profile. The effective parameters to generate a normal traffic profile can be the length of the IP packet, IP packet rate, TTL value, destination port number and etc. In the second step similarity distances are calculated. If the distance between the monitored traffic and the normal traffic profile is larger than a given threshold, a DoS attack is detected.

3.2.2.3 Discussion

The common challenge for all anomaly-based intrusion detection system
is that it is difficult or impossible for the training data to provide all types of
normal traffic behavior. To minimize the false positive rate, a larger number
of parameters are used to provide more accurate normal profiles. However,
with the increase of the number of parameters, the computational overhead to
detect intrusion increases.

3.2.3 Traffic-Volume-Based Detection

A large number of traffic volume-based anomaly detection works exist in the literature. Some of them are briefly explained.

3.2.3.1 Aggregate-based Congestion Control (ACC)

S.Floyd et al [20] proposed a new method for detecting DDoS attacks based on the amount of traffic, that is, Aggregated-based Congestion Control (ACC). ACC works based on the congestion level to detect and reduce DDOS attack traffic and flash crowds. In the ACC method, when a collection of packets from one or more flows has the same destination an attack is defined. The detection algorithm in ACC determines the destination addresses of the victim machines based on the destination network prefix of packets dropped at the observed router during a very short period. If the number of dropped packets of a certain destination address is larger than average, ACC puts the destination address on a list. The destination addresses in this list are then clustered into 24-bit or longer network prefixes. If the arrival rate of each network prefix exceeds a threshold, ACC suppose all traffic to this network prefix as DDOS attack traffic and responds to all incoming traffic sent to this network prefix.

3.2.3.2 Discussion

Some accurate prediction techniques are not suitable for real-time traffic volume prediction due to the high computational complexity. Another problem of this method is that it applies its techniques for anomaly detection of aggregate traffic. However, it is very hard to detect the trivial anomalous changes of aggregate traffic rate during the early stages of a DDoS attack because the attack traffic is actually still a small partition of the entire traffic at the victim end.

3.2.4 IP attributes-based DDoS Detection

A number of works treat anomalies as deviations in a number of IP attributes, *e.g.*,source IP address [11], TTL [22], and the combination of multiple attributes [23].

3.2.4.1 Source IP Address Monitoring

T.Peng et al [13] propose a method called Source IP Address monitoring (SIM) for detecting DDoS attack. Source IP monitoring (SIM) has two steps: online training, and detection.

In the online training phase, each IP address that has been examined to be a legitimate IP address is added to a database called IP Address Database (IAD). Some rules can be used to determine whether the IP address is legitimate or not. For example, a TCP connection with less than 3 packets can be tagged as suspicious packets. The IAD should be updated occasionally by adding new IP addresses and deleting expired IP addresses.

The second step is detection and learning. In the second step, incoming IP addresses are collected in a given time interval.
If an IP address appeared during the sampling period and it is not in the IP Address Database, it is considered to be a new IP address. By analyzing the number of new IP addresses during the sampling period compared to the size of IP Address Database, we can detect whether a DDoS attack is occurring. If an attack is detected, the online learning is suspended. Otherwise, online learning proceeds. This methodology is used to detect attacks that use a small number of source IP addresses.

3.2.4.2 TTL

A DDOS attack most often creates network congestion and changes the statistical distribution of the TTL attribute in traffic. Based on this idea, C.Jin.et al [22] proposed an approach to detect anomalies created by DDoS attacks. This method of detecting is described in detail in section 5.5.1.

3.2.4.3 Combination of Multiple Attributes

Y. Kim et al. [23] proposed a method for detecting DDoS attacks based on establishing a baseline profile during normal operation to detect deviant traffic. The key point of this method is that different sets of base profiles are needed for different applications and products. Some characteristics of packet attributes that can be useful for traffic profiling include: IP protocol type values, packet size, server port number, source and destination IP prefixes, time-to-live values, TCP/IP header length and TCP flag patterns.

Different combinations of these characteristics of packet attributes such as packet size and protocol type, server port number and protocol type, source IP prefix and TTL values and etc can be used to establish the baseline profile. During baseline profiling, the number of packets with each attribute value is counted and the corresponding ratios are calculated. The ratios of attributes are measured during multiple periods and one value that represents all periods is selected. The threshold can also be selected through either static or adaptive methods.

3.2.4.4 Entropy

L.Ling et al [24] proposed a method called Entropy. In this detection algorithm, statistical properties of specific fields in packet headers are measured at various points in the Internet. For instance, if a detector captures 1000 consecutive packets at a peering point and computes the frequency of occurrence of each unique source IP address in those 1000 packets, the detector will then have a model of source address distribution. Further computations with this distribution allow measuring the randomness or uniformity of the addresses.

Entropy can be computed for a sample of consecutive packets. Comparing the value of entropy for a sample of packet header fields with that of another sample of packet headers from the same peering point provides a mechanism for detecting changes in randomness. It has been observed through experimentation that while a network is not under attack, the entropy values for various header fields are within a narrow range. While the network that is under attack with current attack tools exhibits entropy values that quite noticeably exceed this range [24].

3.2.4.5 Hop-Count Filtering

Hop-count filtering is a mechanism proposed by C.J. Haining et al.[17]to counter spoofed IP addresses in DDoS attacks. The hop counter can be inferred from the TTL field. This mechanism classifies the packets based on their TTL value and builds an accurate IP to hop-count mapping table. Then, when the network experiences a high level of congestion, the mechanism will drop those packets whose hop number does not match the mapping table [17].

This mechanism can be tricked if an attacker spoofs the initial value of the TTL field, which is not impossible if the attacker has a greater knowledge of the network. On the other hand, it may make a false positive under high level of congestion because under a high level of congestion, congestion control mechanisms may reroute legitimate packets and that may change their normal hop numbers. They will then be dropped since they no longer match the mapping table.

It may so happen that the attacker chooses the forged IP address such that the TTL value of both the forged and real source IP address packets at the destination are the same. In this case, the hop-count filtering method doesn't work well in detecting DDoS attack and false negative results may happen.

3.2.4.6 Discussion

The source IP addresses monitoring works when the attacker deploys a large number of agents and handlers. In such a case, it doesn't need to use the IP spoofing method because a large number of agents are sending a sufficient number of attack packets. If the attacker launches the attack with limited number of agents and handlers, this method is not effective in detecting DDoS attack.

In TTL and Hop-count Filtering methods, the attacker can send a packet with any initial TTL value. If the attacker attempts to send the spoofed packets with the expected TTL value, the attacker must know the expected TTL or the number of hops from spoofed source to the target.

Sometimes the attackers are so smart and can set the initial value such that it will arrive with the expected TTL value. The attacker may trace the route to determine the number of hops between the spoofed source and the target.

3.3 Attack Source IP Identification

Once an attack has been detected; an ideal response would be to block the attack traffic at its source. Although DDoS attacks are simple to implement, they are difficult to prevent because the attacker in most DDOS attacks uses the IP spoofing method to hide its real identity. Also the attacker uses agents and handlers to launch an attack. Although, it is important to identify the handlers and the agents because these compromised hosts can be used in future attacks and must be cleaned of viruses and Trojans, a serious challenge in IP traceback is how to find the real attacker in addition to agents and handlers. Most IP traceback methods can determine the point of origin for the attack traffic at the agents, but they do not accurately reveal the identity of the real attacker behind the agents. Thus, a new method to find the real attacker is still a difficult problem to solve.

IP traceback is one of the many effective methods for restoring normal network functionality as quickly as possible, preventing reoccurrences, and, ultimately, holding the attackers accountable [25].

Common traceback methods involve packet marking, a technique where a router places a unique mark within the header of each packet that it forwards. In this method, each router will mark the incoming packets all the way to the server. Thus, when the end host receives a packet, the total mark will be used to differentiate between a client and an attacker. Therefore, with an effective packet marking scheme, a server can identify a client correctly without relying on the correct source IP address. The methods can be categorized into 4 different categories: Active IP traceback, logging, packet marking and hash-based IP traceback that are described in the following sections.

3.3.1 Active IP Traceback

The main idea for Active IP Tracback schemes is that routers interfere with the attack traffic to trace the origin of the attack.

One of these techniques is a method proposed by Backscatter et al [26] assuming that DDoS attacks generally use invalid or spoofed source IP addresses.

In the Backscatter traceback [26] method, after an attack is detected, all routers should drop all packets to the victim and send an ICMP destination unreachable error message packet to the source IP addresses. It is worth noting that the source IP addresses are spoofed IP addresses during a DoS attack, which could be invalid IP addresses. They also send all ICMP destination unreachable error message packets with invalid destination IP addresses to an analyzer to reveal the entry point of the attack packets by checking the source IP addresses of these collected ICMP packets. Since all of these packets are attack traffic, the entry point of the attack traffic can be revealed by checking the source IP address (the IP addresses of the routers) of these collected ICMP packets that can reveal the whole path from the destination to the source of attack packets. Also, a request can be sent to the upstream routers of the attack traffic entry point for further traceback.

This method has a big drawback. Since this method works based on the assumption that DoS attack traffic will always contain invalid source IP addresses , for example , 192.168.*.*, the attacker only needs to use a valid (spoofed or non-spoofed) IP address to avoid detecting the DoS attack.

Another Active IP Traceback method is link testing. Burch and Cheswick [27] proposed a link-testing traceback technique. It infers the attack path by flooding all links with large bursts of traffic and observing how this perturbs the attack traffic. This method needs powerful routers to generate huge traffic in each link, and ability to tell which link one packet comes from.

Stone [28] proposed another link testing method to overcome this limitation. During DoS attacks, attack traffic is rerouted to the overlay network which is called CenterTrack. The CenterTrack is normally equipped with routers configured for tracking. Thus, the attack packets can be easily tracked, from the routers close to the target to the attack entry point of the ISP.

3.3.2 Packet Marking

Packet marking techniques rely on routers along an attack path to mark packets with self-identifying information. Routers can do this by either generating additional ICMP based packets or by inserting their IP address into the packet header.

3.3.2.1 ICMP based marking

This approach uses ICMP traceback messages that router generates to add to the IP packets. This ICMP traceback message contains partial path information that indicates where the packet came from, when it was sent, and its authentication. The victim receives these messages in addition to information from regular network traffic.

Bellovin [31] proposed an approach called the ICMP traceback scheme that routers generate an ICMP traceback message (called an iTrace packet) to the destination containing the address of the router with a low probability. For a significant traffic flow, the destination can gradually reconstruct the route that was taken by the packets in the flow. The iTrace packets are generated with a very low probability by routers to reduce the additional traffic, which undermines the effectiveness of the scheme. Since in a DDoS attack each agent generates only a small amount of the total traffic attack, the probability of choosing an attack packet is much smaller than the sampling rate used.

3.3.2.2 Packet Marking in IP Header

In this method IP packets are marked with important information by the routers along the path. The victim uses the markings in the IP packets and tries to reconstruct the attack path. In this method, reconstruction of the attack path relies on the volume of marked packets collected at the victim. Figure 3.1 shows the header of the IP packet.

As Figure 3.1 shows, the options field and identification field are two possible fields for inscribing such marking information. The options field in the IP packet is used for adding extra information for additional processing like testing, debugging and security. Using the option field in the IP packet for adding path information might increase the packet size and cause the IP packet to be fragmented along the way.

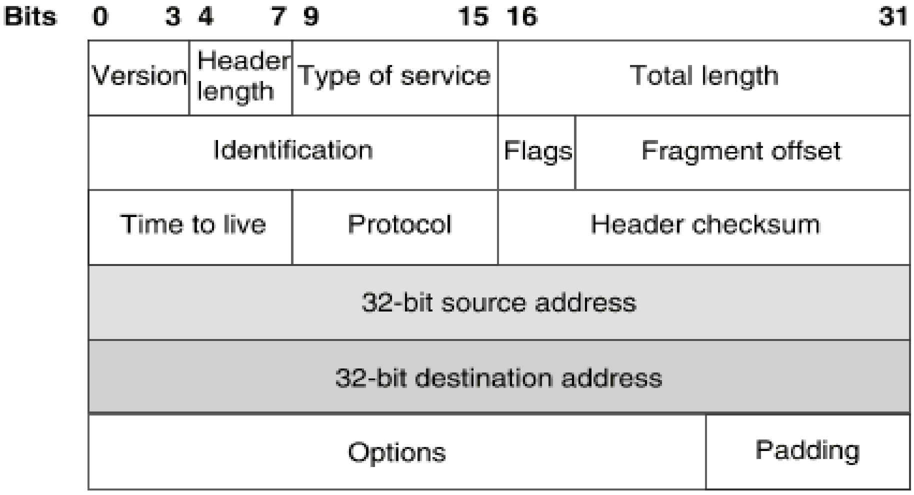


Figure3.1: The Header of the IP Packet

The identification field in the IP packet was designed to hold the fragmented packet id. Since it was seen that less than 0.25% of the IP packets on the Internet are fragmented, using the identification fields to add the packet path to the IP header is acceptable. To avoid increasing the overhead of the IP packets Savage et al. [32] proposed a new approach that describes probabilistic sampling with a probability of 1/25. The main idea behind probabilistic packet monitoring (PPM) [32] is that each router inserts its IP address (partial path information) into the incoming packets probabilistically while they travel between the source and the destination. Based on the embedded path information, a target can reconstruct the packet transmission path.

Tao Peng et al. [13] proposed a new adjusted marking scheme to increase the probability of receiving packets from distant routers. In this technique, the marking probability at every router is computed by using the distance from itself to the destination. The marking probability is computed using the formula 1/ (31-d) where d is the distance from the current router to the victim. The drawback with this scheme is that the router is dependent on the underlying protocol to compute the distance from itself to the victim. This is a router overhead, which considerably slows down the packet marking.

3.3.3 Logging

Logging is another way to traceback to the origin of the attack. All traffic is logged by key routers in the Internet and data-mining techniques are used to traceback to the attacker's source. When an attack has been detected, the victim can poll the upstream routers to check if the router has logged the attack packets. By recursively polling the upstream routers, the victim reconstructs the attack path. Logging seems to be a straightforward solution and allows accurate analysis of attack traffic even after the attack is over. But the main drawbacks of the technique are the amount of processing power involved and the amount of data needed to be logged and to be shared to the partners involved in the attack traceback.

3.3.4 Hash-based IP Traceback

Dawn Song and Adrian Perrig [33] proposed modifications to Savage's PPM method [32] to reduce the amount of information that has to be added to the IP packet header by storing a hash of each IP address instead of the IP address of routers in the whole path. The router addresses are encoded to create a hash value. The hash value created then is stored in the outgoing IP packet header.

In Snoeren et al. [34] method, routers keep a record of every packet passing through. A Bloom filter [35] is used to reduce the memory requirement to store packet records. Moreover, in order to protect privacy, only packet digests, instead of actual packets, are stored. When a traceback is needed, a target will send a traceback query for one packet to its upstream traceback routers. A router can then identify this packet by checking its records and passing the query to its neighboring routers. Eventually, the packet origin can be located. This method is arguably the most effective scheme to traceback DDoS attacks. However, the success of traceback depends on the number of tracking routers installed, and the area covered by these routers. Although an efficient scheme is used to compress the storage, it is still a huge overhead for a router to implement this scheme, especially for high speed traffic over a long period. Therefore, wide deployment is not expected in the near future, and the traceback strength is limited.

More importantly, if a router with tracking facilities is compromised by an attacker; spoofed information can be generated to mislead the traceback.

3.4 Summary

This chapter has described the existing methods for defending against DDoS attacks. It seems that it is not possible to completely prevent DDoS attacks because there will always be vulnerable hosts in the Internet to be compromised for attack purposes, and many DoS attack mechanisms are based on using ordinary features of protocols or network services. Also, all these techniques are based on one or more assumptions, which are not always reliable. Attackers can evade detection by overthrowing these assumptions.

Chapter 4

Introduction to LISP Protocol

Tunneling at the edge router is proposed for two reasons. 1) Tunneling can provide multi-homing without any effect on the size of the non-default routing table. 2) Tunneling can alleviate the problem of IPv4 address exhaustion. We note that tunneling routers insert their IP address, called routing locators, which is useful in IP traceback.

Actually, a tunneling router adds its routing locator as the source address into the outer header of IP packet. So, instead of tracing back attack packets to the host IP address, it is possible to first trace back to the Ingress Tunneling Route. It is proposed that tunneling can be deployed to connect e.g. corporate networks to the Internet. Since this approach proposed the tunneling based defense mechanism to defend against DDoS attacks, the domain network should support a kind of tunneling mechanisms. There is a number of tunneling based mechanisms to choose that each has its own strength and weakness. The LISP protocol is a tunneling-based mechanism that is an emerging IETF recommendation. This chapter gives an overview of the LISP protocol.

4.1 Introduction LISP Protocol

LISP (Locator/Identifier Separation Protocol) describes a network-based protocol that enables separation of IP addresses into two new numbering spaces: Endpoint Identifiers (EIDs) and Routing Locator (RLOC) [36]. EID are End-site addresses for hosts and routers while Routing Locators (RLOCs) are infrastructure addresses used for routing and forwarding of packets through the network. The Locator/Identifier Separation Protocol (LISP) provides a set of functions for routers to exchange information and map from non-routable Endpoint Identifiers (EIDs) to routable Routing Locators (RLOCs) [36].

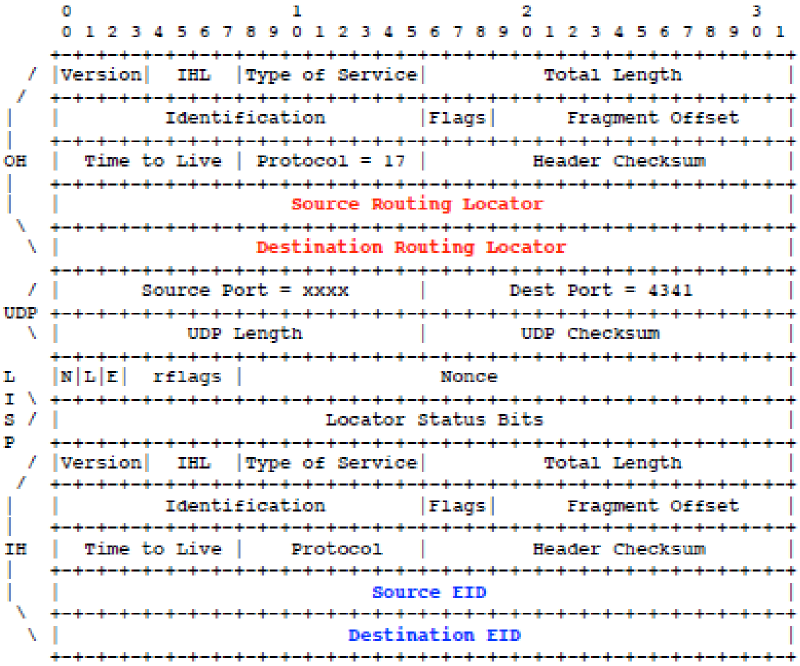
4.2 Overview of LISP

In LISP terminology, the IP addresses that hosts use for sending and receiving packets do not change. These IP addresses are called Endpoint Identifiers (EIDs). On the other hand, Routers continue to forward packets based on IP destination addresses. When a packet is LISP encapsulated, these addresses are referred to as Routing Locators (RLOCs). Most routers along a path between two hosts will not change; they continue to perform routing and forwarding lookups on the destination addresses. Two network elements in LISP protocol are the Egress Tunnel Router (ETR) and the Ingress Tunnel Router (ITR). An Egress Tunnel Router (ETR) receives LISP-encapsulated IP packets from the Internet on one side and sends decapsulated IP packets to site end systems on the other side. In fact, an ETR accepts an IP packet where the destination address in the outer IP header is one of its own RLOCs. Then the router strips the outer header and sends the packet based on the next IP header found.

An Ingress Tunnel Router (ITR) accepts IP packets from site end systems and encapsulates the packet and send the packets to the Internet. In fact, an ITR accepts the IP packet with a single IP header and treats the IP address in the packet as EIDs (inner header) and perform an EID-to-RLOC mapping lookup. Then the router prepends an outer IP header with one of its globally routable RLOCs in the source address field and the result of the mapping lookup in the Destination Address field. For routers between the source host and the ITR as well as routers from the ETR to the destination host, the destination address is an EID. For the routers between the ITR and the ETR, the destination address is an RLOC. LISP protocol has two operation modes. LISP Data-Plane Operation and LISP control Plane [36].

4.2.1 LISP Data-Plane Operation

When a host in a LISP domain emits a packet, it puts its EID in the packet source address and the EID of the correspondent host in its destination address (hosts look up EIDs in the DNS). If the destination of the packet is in another domain, the packet will be sent to one of the ITRs in its domain [37]. The ITR maps the destination EID to a RLOC that corresponds to an ETR. The ITR then encapsulates the packet, setting the destination address to the RLOC of the ETR returned by the mapping lookup or by static configuration. Figure 4.1 shows the LISP IPv4 encapsulation. When a packet arrives at the destination ETR, it decapsulates the packet and sends it to the destination. LISP packet are categorized to three types.1) Data Probe 2) Map-Reply message 3) Map-Request message [37]. The Data Probe is a data packet that an ITR sends to the mapping system to probe for mapping. Map-Request is a packet that an ITR sends to query the mapping system to request a particular EID_RLOC mapping. Map-Reply message is a packet that an ETR sends to an ITR when it receives the Map-Request or a Data-Probe packet if it receives such a packet in which the outer header destination address is the same as that of the inner header [37].



```
        0                   1                   2                   3
        0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     / |Version|  IHL  |Type of Service|          Total Length         |
    /  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   |         Identification        |Flags|      Fragment Offset    |
   |   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   OH  |  Time to Live | Protocol = 17 |         Header Checksum       |
   |   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   |                    Source Routing Locator                     |
    \  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     \ |                 Destination Routing Locator                   |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     / |       Source Port = xxxx      |       Dest Port = 4341        |
   UDP +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     \ |           UDP Length          |        UDP Checksum           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   L   |N|L|E|  rflags |                 Nonce                         |
   I \ +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   S / |                     Locator Status Bits                       |
   P   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     / |Version|  IHL  |Type of Service|          Total Length         |
    /  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   |         Identification        |Flags|      Fragment Offset    |
   |   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   IH  |  Time to Live |    Protocol   |         Header Checksum       |
   |   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   |                           Source EID                          |
    \  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     \ |                         Destination EID                       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4.1: LISP Header Format

4.2.2 LISP Control Plane

The LISP Control-Plane describes possible mechanisms for the mapping systems. Some mapping systems are NERD (A Not-so-novel EID to RLOC Database), ALT, EMACS, and CONS.

NERD is a model that has a signed compact database of EID to RLOC mappings and each ITR contains an entire mapping database. A CDN (Content Distribution Network) is used to distribute the signed database and updates to it and successive incremental updates are used to keep the databases up to date without having to retrieve entire copies [39].

LISP-ALT is a model that uses a logical topology of LISP-ALT routers that connects them to each other via GRE tunnels. EID-prefixes are advertised and aggregated along this topology and ITR sends Probes and Map-Requests over this topology to find the destination ETR and the destination ETR replies with Map-Reply [37].
LISP-EMACS uses BGP over GRE and finds ETR routers by multicast Data Probes. ETRs hash their EID-PREFIXES to join a multicast group [40].
LISP-CONS is a mapping system for LISP 3. It is a hybrid approach that pushes EID-prefixes at upper levels of hierarchy and pulls mapping from lower levels of hierarchy. Requests get to where the mappings are stored and replies are returned [38].

Chapter 5

Tunneling Based IP Traceback System

5.1 Introduction

When an attack has been detected**,** a proper response would be sending a command to block the attack. An ideal response includes source IP traceback. Unfortunately, there are no simple methods to track IP traffic to its source owing to two aspects of the IP protocol. First, it is quite simple to forge the source IP address of each packet and second, IP routing is stateless in nature.

In order to address these limitations, numerous schemes have been proposed based on enhancing router functions to support IP traceability. One of these schemes, as mentioned earlier, is Probabilistic Packet Marking (PPM) [32]. In the PPM method, routers insert partial path information into the incoming traffic probabilistically and the victim reconstructs the packet path using the partial path information. In this method, all routers should insert their IP addresses into the packets while the victim just needs the IP addresses of edge routers to find the real attacker.

For finding the source IP address of the attacker or its agent, we first need the IP address of the ingress point. So, in our new approach, only the edge routers (ingress point) insert their IP addresses into the packet encapsulating the packet by using the LISP protocol and forwarding it to the victim's network through IP tunneling.

This method is more efficient than the existing ones as it specializes to the task of identifying the entry points (i.e. Edge Routers) instead of the whole path traversed by the attack packets. On the other hand, the offending packets may be detected by the egress router or by the victim.

5.2 Tunneling Based Network Topology

In our approach, we make use of partitioning of the Internet into different domains as shown in Figure 5.1 (ISP Networks) and deploy Tunneling based IP traceback schemes in the networks instead of tracing the whole path from destination to the source. In our breakdown, Corporate and customer networks are stub networks where the hosts of the users are connected. ISP networks are non stub networks that transport the user traffic.

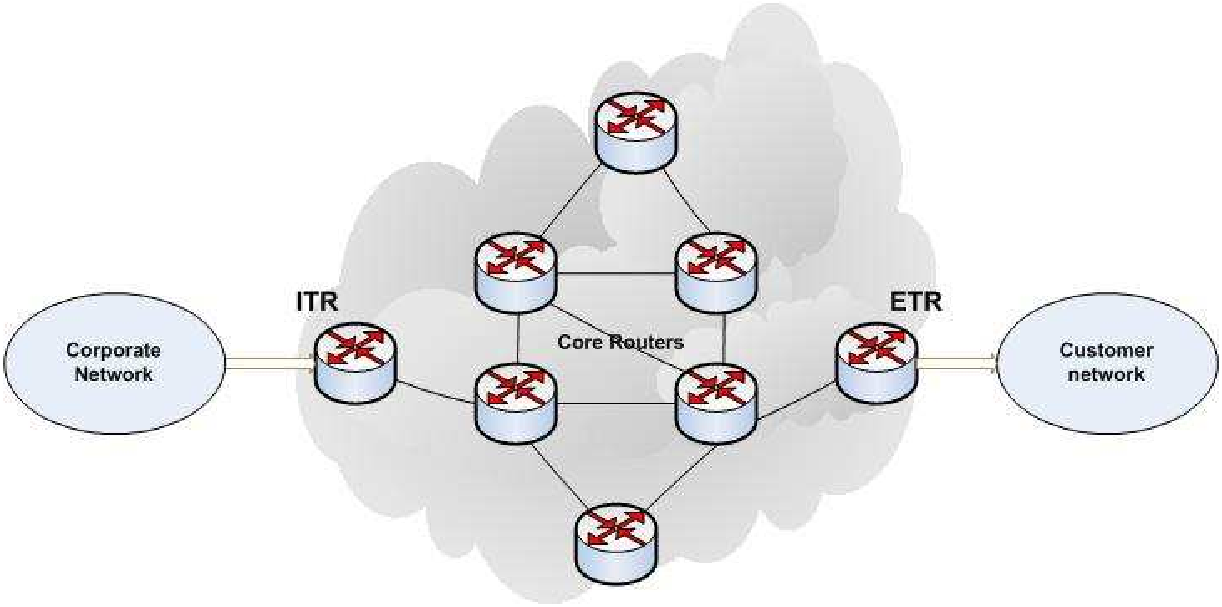


Figure 5.1 Tunneling Based IP Traceback Network Topology

We have classified the routers of the IP network into two different categories based on their functionality and position (1) edge routers that handle tunneling (2) regular routers.

An edge router has at least one direct connection with a customer or a corporate network and connects the customer network to the Internet. Regular routers may appear as core routers inside the perimeter of an ISP network or inside a customer or corporate network and route the traffic of edge routers to or from edge routers. The customer and corporate networks can be either IP networks themselves, where IP routing is used for example to connect multiple sites of a corporation or they can be flat Ethernet networks that organize their connection to the Internet through one or several tunnel routers.

To minimize the complexity of IP traceback and push it towards the edge router, the best solution is to force the edge router to insert its IP address to packets and make the core network free of state information about traffic flows. This is precisely what LISP forces the ingress router to do. In LISP, the ingress router puts its routing locator into the source address field of the outer IP header.

5.3 Tunneling Based IP Traceback Architecture

In this section, we describe the architecture of Tunneling based IP traceback system that identifies the source of the attack traffic.

In the network, we use the LISP protocol and suggest that the network and all routers in the domain network support the LISP protocol and use this protocol instead of marking packets and thus will insert the IP address of ingress points to the packets.

As shown in the Figure 5.2, we divided the IP network into two different categories: customer/corporate networks and the core networks. We assume in this thesis that the attackers or their agents have no direct access to the core routers. It means that the core routers don't generate attack traffic.

Also we assume that the victim is behind an ETR (Egress Tunnel Router) and edge routers are LISP routers. The attacker or its agent may reside either behind an ITR or it may so happen that attack traffic comes from a router that is not a tunnel router. For this case, the existing methods of IP trace back and DoS attack detection and protection applies. In this work, we will concentrate on the case that also the attacker or its agent resides behind an ITR.

We assume that an ITR is always connected to the ISP network through a Provider Edge (PE) router. Since LISP requires that routing locators and End point Identifiers that both are just IP addresses shall not be mixed, it is reasonable to assume that the ISPs take care of protecting the routing locator addressing space. To do so, the PEs must as a matter of policy apply ingress filtering to the source routing locator coming from the connected ITR. The result is that in the LISP network, source routing locators cannot be spoofed. If the ISPs further agree to carry all LISP traffic in a VLAN that is different from the VLAN that carries the non-LISP Internet traffic, an ETR can immediately spot on attacker that pretends to be an ITR.

Each ETR in this IP network has a detector to detect the attack traffic and it sends an alarm whenever the number of spoofed packets (based on wrong TTLs) or new source IP addresses meets a certain threshold. Also, each edge tunnel router should encapsulate incoming packets based on the LISP protocol; therefore we can read the ingress point address of packets whenever we need from the LISP header.

The main components of Tunneling Based IP Traceback architecture are as follows: 1) Detector, 2) Logger and 3) Searching Device 4) Manager

A Detector should be installed to detect attack traffic and control it passing through the network. In the ETR, a packet logger component is installed which is used to log the packets in a local database. The ETR also has a searching device component installed to search the entry point of attack packets. The detail of each component is described in the next section. Administration manager is used to send messages between these elements and find the attacker.

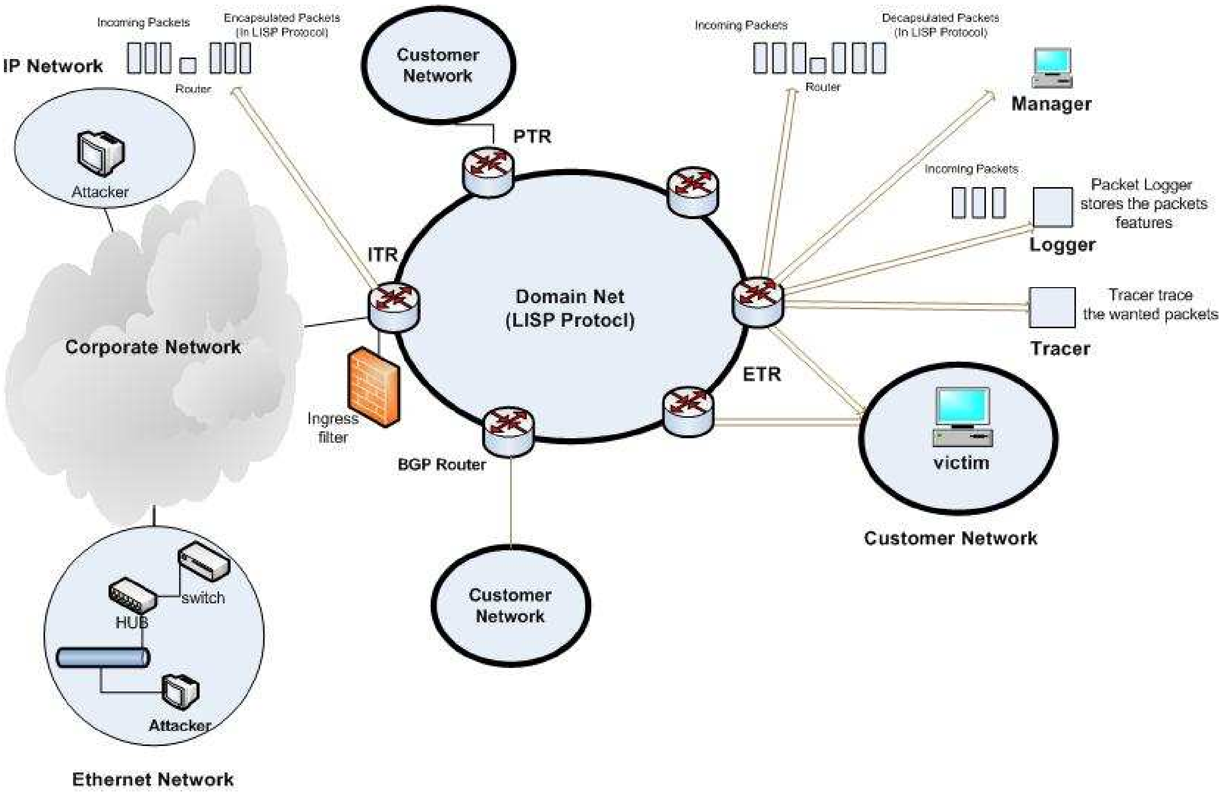


Figure 5.2: A big Picture of Tunneling Based IP Traceback Process

5.3.1 Detector

As discussed before, we need a detector to detect the attack traffic in the ETR (Egress Tunneling Router) before it is received by the victim.

Choosing an algorithm to detect attack traffic is the most important part of defending against DDoS attack. There are many schemes and theories on detection against DDoS attack. Each one has its own advantages and disadvantages.

Monitoring the number of new source IP addresses is an effective attack detection method when the attacker deploys a large number of agents and handlers to launch an attack. In such a case, it doesn't need to use the IP spoofing method because a large number of agents are sending a sufficient number of attack packets. It may so happen that the attacker uses IP spoofing with a limited number of agents to launch a DDoS attack. In these cases, another algorithm to detect spoofed packets can help to defend against DDoS attack. To detect an attack when the attacker uses the IP spoofing method, we can check the TTL value of the IP packets. Detecting spoofed packets by checking the TTL value of the packets is based on several assumptions.

1. When a packet is sent between two hosts, as long as the same route is taken, the number of hops will be the same.
2. Packets sent near in time to each other will take the same route to the destination.
3. Routes change infrequently.
4. When the route changes, significant changes in the number of hops do not happen.

If these assumptions do not hold, this method results in false positives that means that valid packets may be tagged as spoofed packets.

So, to detect both models of attacks, we can combine two methods. First, check the TTL value to detect spoofed packets. Second, check the IP addresses to find the number of new IP addresses. If the number of new IP addresses or the number of spoofed IP packets meets a certain threshold, the detector should send an alarm to the manager to warn it.

5.3.2 Logger

This component is used to log the packet information and information of ingress points associated with each packet in a well defined data structure. When a packet is received by a router, a logger extracts the packet features and additional information from the incoming packet that is being decapsulated and creates a record for each packet containing the IP address of ITR (ingress point IP address, or RLOC using LISP terms) and source IP address of packet (EID in LISP terms). In such cases that the packets are coming form non-tunnel router (as shown in Figure 5.2), the logger should insert a specific number such as 101 instead of the ingress point of the packet. So, the searching device can trace incoming packets that are routed form non-tunnel routers.

5.3.3 Searching Device

This component is used to find the information of the ingress point that is sending the spoofed packets. Searching device starts the trace process when it receives an order from the manager and searches for the target packet feature in the local database written by the logger. If a record matches with the trace packet(s), the searching device retrieves the information containing the ingress point address and source IP address and sends this information to the manager. The manager should send an ICMP message to the ITR stating that it has received offending traffic that is suspected to be a DoS attack and give the parameters of the traffic, such as the timestamp, the source EID, the destination EID, protocol used, number of packets, etc to warn the ITR of the attack that is coming from its corporate or customer network.

5.3.4 Manager

This component is used to manage the whole IP traceback process. It sends the traceback request message to the searching device after receiving it from the detector.

It also sends the ICMP message to the ITR to warn it offending packets are coming from its corporate/customer network.

5.4 Tunneling Based IP Traceback Process

The main aim of the IP traceback is identifying the true IP address of the host originating attack packets. If we can identify the true IP address of the attacker's host, we can also get information about the organization involved in the attack of the attacking host.

As mentioned earlier, the tunneling-based IP traceback process uses the LISP protocol and follows LISP protocol encapsulation and decausulation methods to trace the attack.

The basic function required to trace the attack back to their source can be as follows:

-Detection of attack packets by the detector installed at the ETR.

- Sending the traceback request to the searching device by the manager or even victim through its IDS.
- Tracing of the attack packets in the logger and sending the result to the manager.

In tunneling-based IP traceback, when a packet is received by an ETR of the domain, first of all for detecting the attack, the packet is analyzed by the attack detector installed on the ETR. Figure 5.3 shows the process explained before.

5.5 Tunneling Based IP Traceback Design Model

In this section the design of the Tunneling Based IP Traceback components is discussed.

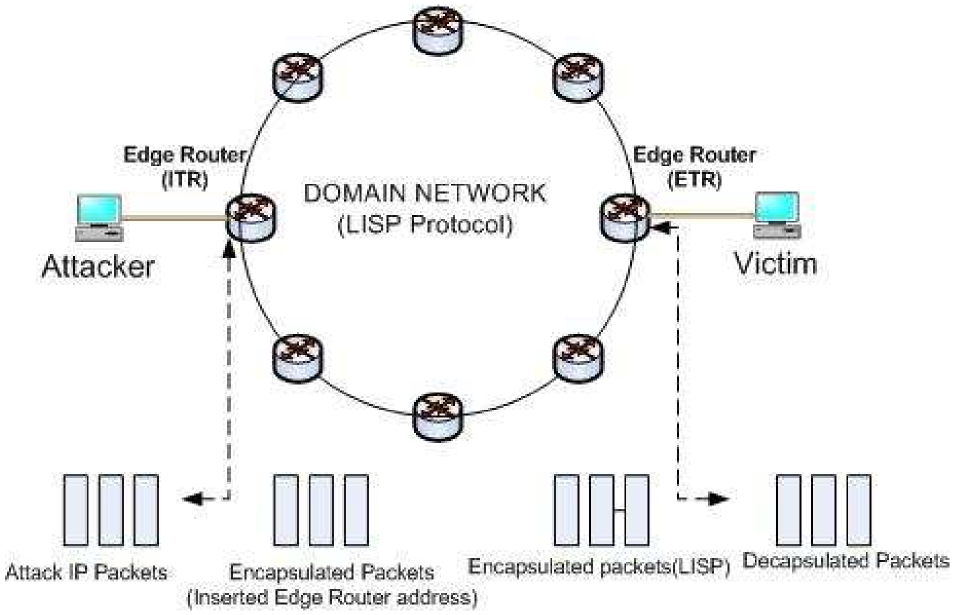


Figure 5.3: Process of Tunneling Based IP Traceback

5.5.1 Design of Detector

In this section, we describe how the detector component can be designed for the ETR to detect the DDoS attack.

As described in section 5.3.1, for the purpose of this thesis, two methods are combined to detect DDoS attack.

If the number of spoofed packets or the number of new IP addresses meets a certain threshold, the detector should detect an attack and sends an alarm to the manager. So, the detector should work in two phases. In first phase, it checks the TTL value of incoming packets to find the spoofed packets and in the phase two, it checks the source IP address of incoming packets to monitor the number of new source IP addresses.

In phase one, the detector detects the spoofed packets by checking the TTL value of incoming packets in packet header [17]. The time-to-live (TTL) is the number of hops that a packet is permitted to travel before being discarded by a router. The TTL value is an eight bits field in the IP packet header that a host used before sending a packet to the destination to prevent the packet from endlessly circulating on the Internet or other network. Each router that receives an IP packet decreases the TTL value by at least one before forwarding the packet to the next hop. If a packet's TTL field reached zero, the router detecting it discards the packet and sends an ICMP (Internet control message protocol) message back to the originating host. If the destination knows the initial value of TTL, by calculating the differences between the initial value (at the source) and the final TTL value (at the destination) the hop-count between the source and destination can be computed.

So, by checking the hop count of packets from source to the destination, a detector can detect that a DDoS attack is happening. Although, an attacker can forge any field in the IP header, he cannot change the number of hops that an IP packet takes to reach its destination.

The problem now is finding the initial TTL value by the destination. Most modern OS use only a few selected initial TTL values, 30, 32, 60, 64, 128, and 255 [41]. These initial TTL values are used by most common OS, such as Microsoft Windows, Linux, and UNIX systems. The destination only sees the final TTL. One can determine the initial TTL value of a packet by selecting the smallest initial value in the set that is larger than its final TTL. For example, if the TTL value of received packet is 112, the initial TTL value is 128, the smaller of the two possible initial values, 128 and 255. To resolve ambiguities in the cases of 30 and 32, and 60 and 64, we will compute a hop-count value for each of the two possible initial TTL values, and accept the packet if either hop-count matches [9].

Y.You. [9] proposed a method for detecting spoofed packets based on the TTL value that we explain briefly. In this method, destination builds an accurate table that maps source IP address to hop-count value and updates the mapping table for a period of time to capture hop-count changes. When a packet is received in the edge router of the victim side (ETR), the detector extract the source IP address and the TTL value and subtract it from the initial TTL value to compute the hop count. Then it searches in the table to retrieve the correct hop count based on the source IP address. If the computed hop count and the hop count of that table matches, the packet is not spoofed otherwise the packet is labeled as a spoofed packet.

Since monitoring all packets may cause delay in the work of the edge router, we can set the detector to monitor sampled packets. (For example 1% of incoming packets can be selected to be checked by detector).

If the number of spoofed packet reaches a threshold (that we can set for the detector), the detector changes its mode and checks each incoming packets and starts to log incoming packets in the logger. After receiving a certain number of suspicious packets, the detector sends an alarm to the manager to warn it attack traffic is coming.

In phase two, when the attacker uses a large number of agents with new source IP addresses to carry out the attack, the attack can be detected by a method proposed by Tao [11] that is described briefly in section 5.3.1. It has two phases: off-line training phase and detection phase.

In off-line training phase, the entire set of source IP addresses of the previous successful network connection, should be stored to compile an IP address database. A learning engine keeps the database up-to-date by inserting new legitimate IP addresses and deleting expired IP addresses.

The aim of the detection phase is to measure the percentages of new IP addresses during a sampling period. In this phase, if the source IP address of an incoming packet appeared during the sampling period and it is not in the IP address database, it is considered to be a new IP address. When the number of new IP addresses during the sampling period compared to the size of IP address database meets a certain threshold, the detector should send an attack alarm to the manager to warn it that offending traffic is coming.

5.5.2 Design of Packet Logger

In this section, the Logger design method is described.

The logger stores the information of the incoming packets in a database that can be used by the searching device to retrieve the information of the attacker. After receiving the packet by the ETR in the victim side, it inserts a record in the database containing the IP address of the ingress point (ITR) of the packet and source IP address of the packet. In such a case that packets come from a non-TR (a router that is not a tunnel router), the IP address of ingress point doesn't exist in the capsulated packets and the logger stores the source IP address and a number (for example number 101) instead of ingress point of packets to show that theses packets come from a non-TR.

This information is useful when an attack happened and helps the attack searching device to retrieve useful information of spoofed packets.

5.5.3 Design of Searching Device

When an attack happened, detector detects it and manager sends a request to trace the attack source IP address.

The searching device accepts the trace request and searches for related information in the database of the Logger. If a record of the spoofed packet(s) exists in the database, the searching device sends the result to the manager. In such a case that the attack traffic has been sent from a non-TR, the logger writes the number 101 instead of ingress point. So, the searching device warns the manager to trace back the origin source IP address through the traditional IP traceback methods. One of these methods can be active IP traceback method.

Otherwise, if the searching device finds the ingress point IP address of attack traffic, it should send an alarm to manager. Manager sends an ICMP message to the ingress point address(s) that is retrieved by searching device to warn offending packets are coming for its corporate / customer network.

5.5.4 Finding Attacker in Corporate Network

In this thesis, we assume that a corporate network is under a single administrator.

So, it is easier to find the offender in a corporate network with a single administrator than finding an arbitrary offender in the whole Internet. When the ITR receives the ICMP message from the manager claiming that attack packets are coming from its corporate network, it should try to block the attack traffic. There are different cases based on the type of the customer/corporate network where the agent/handler or attacker resides. 1) The network is a corporate Ethernet. The agent/handler or attacker can be identified by source MAC address. 2) The network is a regular routed IP network.

If the attacker belongs to an Ethernet network behind the ITR, the ITR should send an SNMP trap message to the manager of the corporate network and ask to find the interface binded to the specific IP/MAC address and block the interface. It means that the source IP and MAC address of spoofed packets are needed.

So, the manager that is a SNMP manager can send a SNMP GET message to the switch or bridge in the Ethernet network to find the interfaces binded to the IP/MAC address of spoofed packet(s) and block it.

Naturally, the corporate network administrator should take care not to react to false claims of DoS attacks against its own hosts. One way of exercising care is waiting for several claims from different ETRs. The validity of the claims can be examined by checking the RLOCs of the ETR.

Chapter 6

Evaluation and Analysis

Since all previous IP traceback methods perform the whole path traceback they have to collect information of the traces from all routers between the source and the destination which may result in considerable overhead and false positives.

In this method, we don’t need to reconstruct the whole path from the destination to the source. We just need to collect the information of the ITRs to find the real attacker or an agent of the attacker. So, the number of packets that is needed to find the source IP address of the attacker or its agents is low and even one packet is enough.

In the previous methods such as Savage et al [32] thousands of packets are needed to reconstruct the attack path and identify the origin of the attack traffic while in the improved methods such as Song et al [33] fewer packets is needed. In this method just one attack packet is enough to identify the source of the attack traffic.

In this scheme, only the edge router is involved in IP traceback process and no overhead is increased on the core routers. So, the processing overhead is very low comparing with the previous PPM methods.

Compare with the previous PPM methods, the memory that this scheme needs is the same. Also, more memory is needed in the ETR on the victim side to store the TTL values and source IP addresses of incoming packets in the training mode and log the IP packet information, but the memory requirement is the same as traditional methods that insert the IP address of each router into the IP packets's header from the source to the destination. In this chapter we evaluate the framework that was proposed.

6.1 Deploying the Tunneling Based IP Traceback

To deploy the tunneling based IP traceback, we assume that edge routers are tunnel routers that support the LISP protocol.

6.2 Tunneling Based IP Traceback Simulations

In this section, we simulate the Tunneling Based IP traceback method to evaluate the effectiveness of this method. Then we present several scenarios and their results and conclude on simulation results.

6.2.1 Simulation Tool

The simulation tool that we used for simulation is OPNET Modeler. OPNET Modeler is an industry solution for modeling and simulation of communications networks, devices, and protocols. It is an object-oriented modeling approach and graphical editors mirror the structure of actual networks and network components. Originally OPNET was developed at MIT, and introduced in 1987 as the first commercial network simulator. OPNET Modeler supports many network types and technologies [42].

OPNET Modeler is based on a series of hierarchically related editors that directly parallel the structure of actual networks.

The first editor is the Network Editor, which graphically represents the topology of a communication network. Networks consist of nodes (switch/router, server etc.) and links models (Ethernet, ATM, FDDI etc.). It is possible to manage complex networks with unlimited subnetwork nesting such as country, city, building, floor etc.. Network editor provides geographical context, with physical characteristic of the networks.

The second editor is Node Editor, which describes internal architecture of the nodes by depicting the flow of data between functional elements, called ”modules”.

The modules can generate, send, and receive packets from other modules to perform the function within the nodes. The modules represent applications, protocol layers, physical resources such as buffers, ports etc.

Behavior and functionality of the modules are described in Process Editor,

the third editor. The Process Editor uses a finite state machine (FSM) to describe
the protocols at any detail. State and transition graphically represent the process
behavior where active state is changed in relation to incoming events. Each state
of process contains C/C++ code for control. Many libraries can be used for protocol programming. To make own specific libraries, variables and statistics are possible too [42].

6.2.2 The Simulation of tunneling Based IP Traceback Process

In our simulation, we implement two customer networks and each one has several host machines. Among these host machines, one can be an attacker that generates spoofed IP packets and one is a victim that receives IP packets (spoofed and non spoofed IP packets).

We also simulate the function of detector, searching device and logger in the simulated network. To perform the detector function, a module is deployed in the ETR. Also, a packet logger logs the incoming packet information consisting of the source IP address and ingress point IP address in a database that is a text file. Searching device uses this text file to search the entry points of attack packets.

On the other hand, the routers in the simulated networks support the LISP protocol by implementing a module in routers to encapsulate and decapsulate IP packets to simulate the LISP protocol.

The simulation process is executed in two phases. In the first phase that is the learning phase, two databases must be manintained. First, the router maintains a data structre that gives the TTL values of distinct source IP addresses. This data structure is filled by recording , for a period of time, the TTL value of distinct source IP addresses. Second, in another database the entire source IP addresses of the previous successful network connections should be stored to compile an IP address database.

In the second phase, when a packet is recieved, the router compares the TTL value of the packet to the expected TTL value of the packet. If they are not the same, the packet would be flagged as suspecious. The router then sends a packet to that source address that will cause a reply to check the TTL value of the reply packet , if they are not the same again, it means that the packet was malcious. After receiving a number of malicious packets (we can initialize the threshold), the detector sends an alarm to the manager to warn it that an attack traffic is coming.

On the other hand, the source IP address of incoming packet should be compared with the source IP address database. If there is not a record in the source IP address database that matches the received packet source IP address, it will be tagged as a packet with new source IP address. When the number of packets with new source IP addresses within a given time meets a threshold, detector sends an alarm to the manager to start the process of tracing.

Also, a logger is installed in the ETR to store the received packet information. A record consists of source IP address and ingress point IP address of the packets. Searching device searches the information stored associated to malicious packets and retrives the ingress point IP address of malicious packets whenever it recieves a trace request from a manger to search for a specific packet(s). So, the origin of the attacker can be identified with very little number of attack packets or single packet recieved by the victim.

6.2.3 Simulation Scenarios and Results

To measure the effectiveness of the detector and other components, we have implemented different scenarios in our simulation.

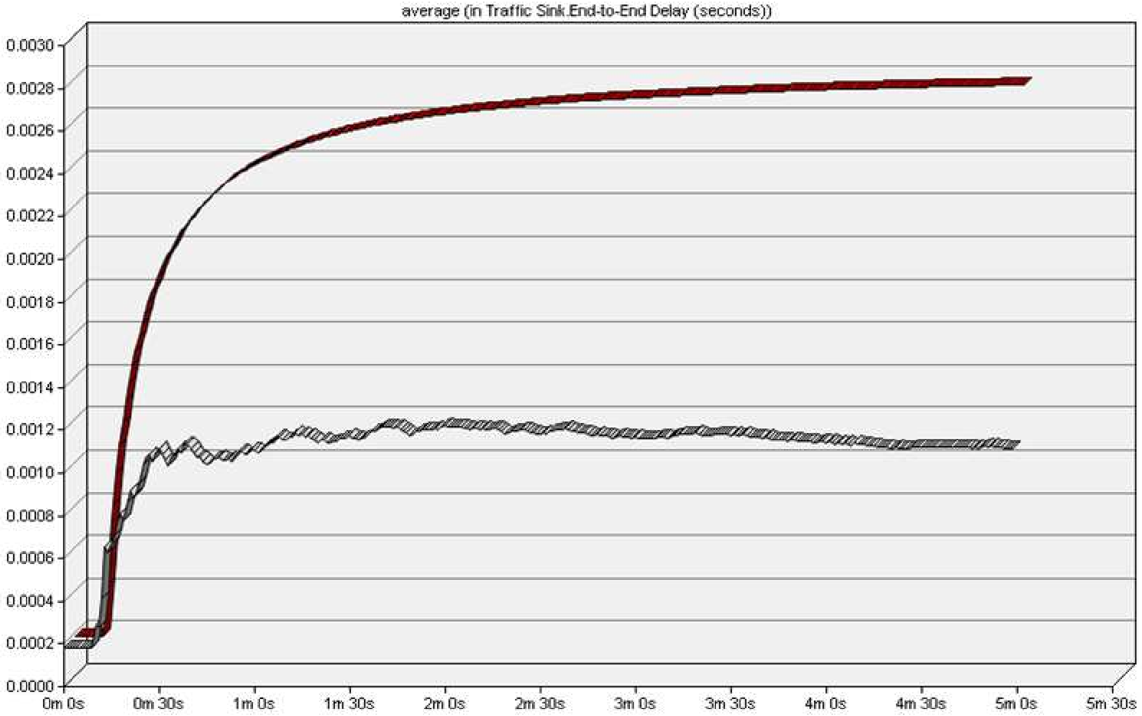
Sneario1: the goal of this scenario is to measure the overhead of the ETR when a decetor is installed on it. When a detector is installed on the edge router processing overhead is increasing and the router effeciency is decreasing. The queue size of the router as a effeciency measure can be observed. Results of this scenario are shown in Figure 6.1.

As it is shown in the Figure 6.1, the attack has happened 15 seconds after running the program, and the End-to-End delay shows that the delay is increasing after the attack happened when the detector started the detection. The Figure compares the ETR in two phases 1) with detector, 2) without detector. As the figure shows the difference between delays in two phases is not considerable.

The Figure 6.2 compares the received packets in two states. First a host is sending normal traffic. After 15 seconds, the attacker launches an attack with sending spoofed IP packets to the victim. The gray graph shows the received packet rate before starting attack while the red graph shows the rate of the received packets in the victim after launching the attack.

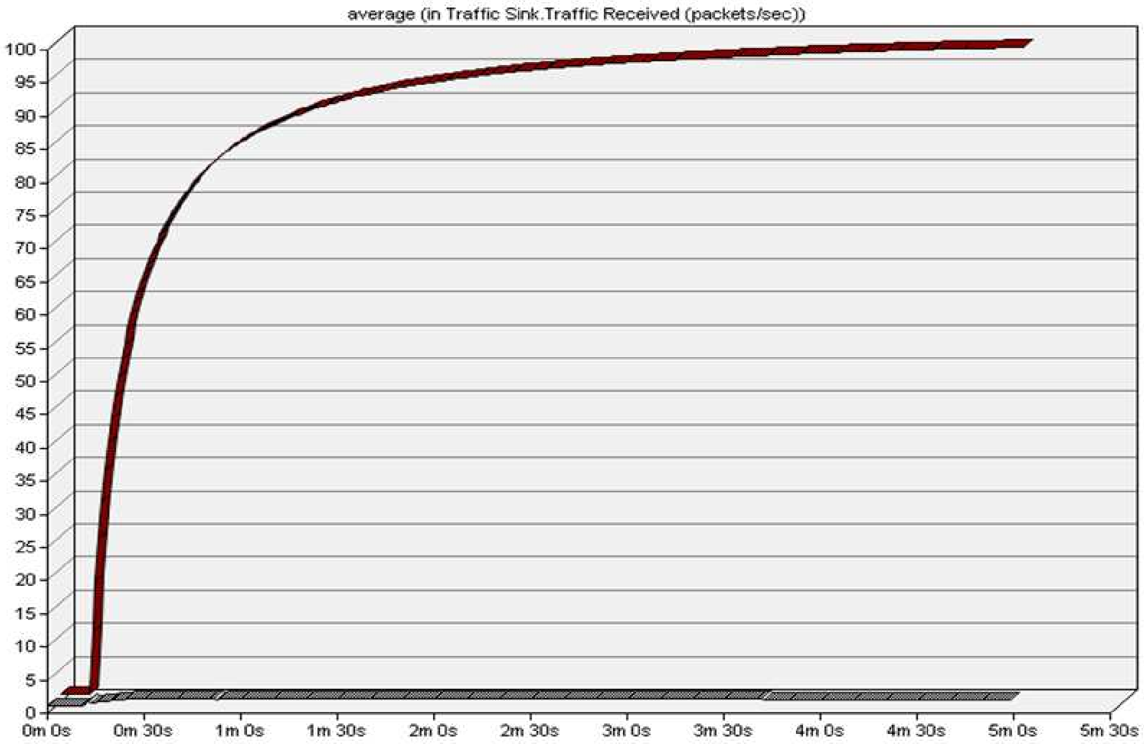
A sign of measuring the efficiency of this method is the router queue size. The Figure 6.3 shows the queue size of the ETR in two modes, with detector and without detector. As the figure shows, the queue size of the ETR is increased after installing a detector to detect spoofed packets but the differences in two phases are not considerable.

As a result a router needs little more processing power to detect spoofed packets.



⬜ End-to-END delay withiout detection
🟥
End-to-End delay with detection

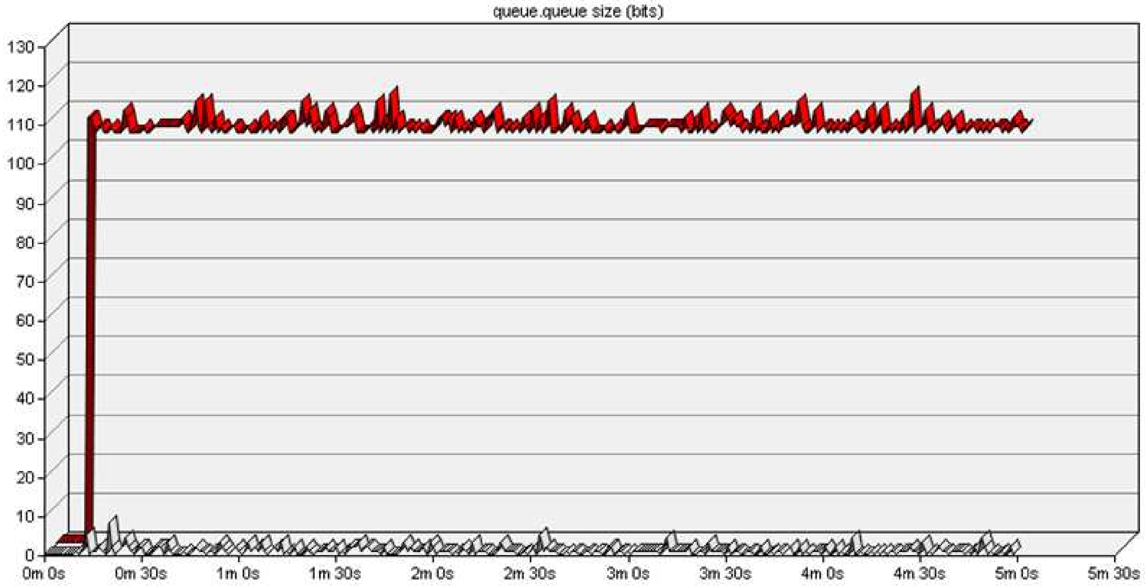
Figure 6.1: the End-to-End Delay in two Modes



⬜ Traffic Recieved (packet/seconds) withiout Attack

🟥 Traffic Recieved (packet/sesonds) with Attack

Figure 6.2: Traffic Received in two Modes.



⬜The Queue Size without Detection
🟥 The Queue Size with Detection
Figure 6.3: The Queue Size of the ETR in two Modes

Scenario2: in this scenario, the rate of the malicious packet generated by the attacker is increased and the end-to-end delay, queue size and detction time is compared in different rate of the sending malicious packets. We have set the threshold to 6. It means that after recieving 6 malicious packet, the detector send s an larm to the manager to sratr the tracing process.

As the tables 6.1 shows when the number of genarated malicious . As the table shows the differences between the delay and the queue size of the ETR in three different sending rate of attack packets are not considerable.

Sending rate(packet/sec)	**Delay(ms)**	**Queue size(bits)**	**Detection time(ms)**
100	2.7	103	50
200	2.75	106	25
400	2.78	107	12

Chapter 7

Conclusion

In this thesis, we revisited the IP traceback problems and used the previous IP traceback methods to design a new method that applies to a tunneling based core network.

In this approch, we have introduced LISP (a tunneling based protocol) for capsulating and decapsulating IP packets. We assume that in the LISP protocol , RLOCs can not be spoofed but in some cases RLOCs can be spoofed. For example the attacker can send the LISP encapsulated packets directly. To solve this problem, the tunneling protocol should support a return routability check that does not exist in LISP protocol yet. However, if the LISP is deployed in s separate VLAN and it is agreed that RLOC poofing is eliminated using ingress filtering traceback is eased.

There are several disadvantage when a detector and searching device are installed on an individual router to detect attack packets to trace back a DDoS attack to the source.

When a detector is installed on a router, the amount of processing power at the router should be increased and the memory that needs to store the table of the TTL values of incoming packets and source IP address of successful conections must be increased. As mentioned in Chapter 6, the processing power that the router needs to detect attacks is little. On the other hand, these days with powerful routers that can provide functionalites such as firewalls, increasing the amount of memory of the router is not a big concern. After , adding a detector on the router to prevent serious damage to the customer network is not impossible.

As mentioned in the previous chapter, false negatives may happen by the detector because of some reasons. First, in some cases we do not have an entry in the TTL value table for recieved packets from a host not previously encounterd in the learning phase and they may not respond to the probe messages. Without any information we are not able to know if the packet is suspicious or not.

For solving this problem , we can suppose that similar IP adrresses have the same number of hops to the target and we can predict values for recievd packets without an entry in the TTL table.

Second, the attacker can send a packet with any initial TTL value. If the attacker attemps to send the spoofed packets with the expected TTL value, the attcker must know the expected TTL or the number of hops from spoofed source to the target. The attacker should be able to set the initial value such that it will arrive with the expected TTL value. The attacker may trace the route to determine the number of hops between the spoofed source and the target, this results in the false nagatives in the detector.

Our approach is simple and easy to implement in a large scale network. In this approach, we have tried to find the attacker or its agents. Finding the real attacker who launch the attack is still a big concern of network researchers and an effective solution should be found to solve this porblem.

References

[1] H. F.Lipson. \Tracking and tracing cyber-attacks: Technical challenges and global policy issues". Special Report CMU/SEI-2002-SR-009, CERT Coordination Center (2002).

[2] S. Specht and R. Lee. Taxonomies of distributed denial of service networks, attacks, tools, and countermeasures. *Technical Report CE-L2003-03*, 164, May 2003.

[3] L. Garber, Denial-of-service attacks rip the Internet." *IEEE Computer*, vol. 33, no. 4, April 2000, pp. 12{17}.

[4] J. MÄolsÄa, \Mitigating denial of service attacks in computer networks". PhD thesis, Helsinki University of Technology, Espoo, Finland, June 2006.

[5]Christos Douligeris and Aikaterini Mitrokotsa. Ddos attacks and defense mechanisms: A classifications. *Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology*, pages 190{193, December 2003.

[6]B.R.Swain, \Mitigation of DDoS attack using a probabilistic approach and end System based technology". Msc thesis, Department of Computer Science and Engineering National Institute of TechnologyRourkela-769 008, Orissa, India,May 2009

[7] Valon Sejdini Li Xiaoming and Hasan Chowdhury. Denial of service (dos) attack with udp flood. School of Computer Science, University of Windsor, Windsor, Ontario, Canada.

[8] A. Kulkarni, S. Bush, and S. Evans. \Detecting distributed denialof-service attacks using Kolmogorov complexity metrics".

[9] Y.You, \A defense framework for flooding-based ddos attacks . Msc thesis, Queen's University Kingston, Ontario, Canada ,August 2007

[10] Christos Douligeris and Aikaterini Mitrokotsa. Ddos attacks and defense mechanisms: A classifications. *Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology*, pages 190{193, December 2003.

[11] S. M. Specht and R. B. Lee, \Distributed denial of service: taxonomies of attacks, tools and countermeasures." in *Proceedings of the 17th International Conference on Parallel and Distributed Computing Systems*, September 2004, pp. 543{550.

[12] V. Paxson, \An analysis of using re°ectors for distributed denial-of-service attacks." ACM SIGCOMM Computer Communication Review, vol. 31, no. 3, July 2001.

[13] T.peng, \defending against distributed denial of service attacks, Phd thesis, Department of Electrical and Electronic Engineering, University of Melbourne,april 2004.

[14] R. K. Chang, \Defending against flooding-based distributed denial-of-service attacks: A tutorial." *IEEE Commun. Mag.*, vol. 40, no. 10, October 2002, pp. 42{51.

[15] H. Wang, D. Zhang, and K. G. Shin. \Detecting SYN flooding attacks". In Proceedings of IEEE INFOCOM 2002, pp. 1530{1539 (2002).

[16] J. B. D. Cabrera, L. Lewis, X. Qin, W. Lee, R. K. Prasanth, B. Ravichandran, and R. K. Mehra. \Proactive detection of distributed denial of service attacks using MIB traffic variables- a feasibility study".

[17] Cheng Jin Haining Wang Kang G. Shin /An Effective Defense Against Spoofed DDoS Traffic URL : http://www.cs.wm.edu/~hnw/paper/filter.pdf

[18] J. L. Bebo, G. H. Gunsch, G. D. Lamont, P. D. Williams, and K. P. Anchor. \CDIS: Towards a computer immune system for detecting network intrusions".

[19] C. Manikopoulos and S. Papavassiliou. \Network intrusion and fault detection: A statistical anomlay approach". IEEE Communications Magazine 40(10), 76-82 (2002).

[20] R. Mahajan, S. M. Bellovin, S. Floyd, J. Inannidis, V. Paxson, and S. Shenker, \Controlling high bandwidth aggregates in the network," *Technical Report, AT&T Center for Internet Research at ICSI*, July 2001.

[21] T. M. Gil and M. Poletto. Multops: a data-structure for bandwidth attack detection". In Proceedings of the 10th USENIX Security Symposium (2001).

[22] Cheng Jin and Haining Wang and Kary G.Shin,\Hop-Count Filtering: An Effective Defense Against Spoofed DDos Traffic.

[23]Yoohwan Kim, Ju-Yeon Jo, Kyunghee Kim Suh, "Baseline Profile Stability for Network Anomaly Detection, Third International Conference on Information Technology: New Generations (ITNG'06), 2006

[24] Liying Li Jianying Zhou and Ning Xiao' \DDoS Attack Detection Algorithms Based on Entropy Computing.

[25] S.C. Lee and C. Shields, “Tracing the Source of Network Attack: A Technical, Legal and Societal Problem,” *Proc.2001 IEEE Workshop on Information Assurance and Security*, IEEE Press, 2001, pp. 239–246.

[26] B. Gemberling, C. Morrow, and B. Greene. ISP Security-Real World Techniques. Presentation, NANOG (2001). URL http://www.nanog.org/mtg-0110/greene.html.

[27] H. Burch and B. Cheswick. \Tracing anonymous packets to their approximate source". In Proceedings of the 14th Systems Administration Conference (New Orleans, Louisiana, USA, 2000).

[28] R. Stone. \Centertrack: An IP overlay network for tracking DoS floods". In Proceedings of the 9th USENIX Security Symposium (Denver, Colorado, USA,1999).

[29]S. Savage, D. Wetherall, A. Karlin, and T. Anderson. \Practical network support for IP traceback". In Proceedings of the 2000 ACM SIGCOMM Conference,pp. 295{306 (2000).

[31] S. Bellovin. The ICMP traceback message. IETF Internet Draft (2000). URL www.research.att.com/~smb/papers/draft-bellovin-itrace-00.txt.

[32] S. Savage et al., “Network Support for IP Traceback,” *IEEE/ACM Trans. Networking*, vol. 9, no. 3, 2001, pp. 226–237.

[33] Song, D., and Perrig, A. Advanced and Authenticated Marking Schemes for IP Traceback. Proc. of IEEE INFOCOM, Vol. 2, April 2001, pp. 878-886.

[34] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, S. T. Kent, and W. T. Strayer. \Hash-based IP traceback".

[35] B. H. Bloom. \Space/time tradeo_s in hash coding with allowable errors". Communications of the ACM 13(7), 422- 426 (1970).

[41] Swiss Academy and Research Network. http://www.map.meteoswiss.ch/map-doc/ftp-probleme.htm.

[36] Network Working Group,D. Farinacci, V. Fuller ,D. Meyer, D. Lewis \Locator/ID Separation Protocol (LISP)draft-ietf-lisp-12,,Internet-Draft, cisco Systems,April 26, 2011

[37] The Internet Protocol Journal ,Volume 11, No.1 (LISP) http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_11-1/111_lisp.html

[38]Network Working Group, S.Brim,N,Cippa,D.Farinacci,V.Fuller,D.lewis ,\LISP-CONS: A Content distribution Overlay Network Service for LISP,draft-meyer-lisp-cons-03.txt,November,14,2007

[39] Network Working Group E.Lear, \A Not-so-Novel EID to RLOC database,
draft-lear-lisp-nerd-02.txt, September,19,2007

[40]S.Brim ,\EID mapping multicast access cooprating system for LISP ,
draft-curran-lisp-emacs-00.txt,September,11,2007

[42]OPNET Technologies INC, OPNET modeler accelerating Network R&D,
http://etidweb.tamu.edu/ftp/entc489rf/Opnet/opnet_modeler.pdf