An Improved Packet size Entropy based DoS Attack Detection Scheme

ASWANI KUMAR T
(211CS2062)

under the guidance of
Prof. Banshidhar Majhi

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela – 769 008, India

An Improved Packet size Entropy Based DoS Attack Detection Scheme

Dissertation submitted in

May 2013

to the department of

Computer Science and Engineering

of

National Institute of Technology Rourkela

in partial fulfillment of the requirements

for the degree of

Master of Technology

by

ASWANI KUMAR T

(Roll 211CS2062)

under the supervision of

Prof. Banshidhar Majhi

Department of Computer Science and Engineering

National Institute of Technology Rourkela

Rourkela – 769 008, India

Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, India. www.nitrkl.ac.in

Dr. Banshidhar Majhi
Professor

May 31, 2013

Certificate

This is to certify that the work in the thesis entitled ***"An Improved Packet size Entropy Based DoS Attack Detection Scheme"*** by ***Aswani Kumar T***, bearing roll number 211CS2062, is a record of an original research work carried out by him under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of ***Master of Technology*** in ***Computer Science and Engineering - Information Security***. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Prof. Banshidhar Majhi

Acknowledgment

First of all, I would like to thank the Almighty God, without whose blessings I wouldn't have been writing this "thesis".

I would like to express my deep sense of respect and gratitude towards my supervisor Prof. Banshidhar Majhi, who has been the guiding force behind this work. Without his invaluable advice and assistance, it would not have been possible for me to complete this thesis. I consider it as my good fortune to have got an opportunity to work with such a wonderful personality. I am indebted to all the professors, co-researchers, batch mates and friends at National Institute of Technology Rourkela for their active or hidden cooperation.

I thank all the members of the Department of Computer Science and Engineering,and the Institute, who helped me by providing the necessary resources, and in various other ways, in the completion of my work.

When I look back at my accomplishments in life, I can see a clear trace of my family's concerns everywhere. My dearest mother, whom I owe everything I have achieved; This thesis is a dedication to her.

Aswani Kumar T

Abstract

A denial-of-service attack is an attempt by a single person or a group of people to disrupt an online service. The cost of the attack depends on the importance of the online service in the Internet world, whether it is online banking or online shopping. Shutting down some services for some hours can cost millions and millions of dollars for companies like Amazon, eBay, HSBC, etc. So a denial-of-service attack is a very serious problem in the online world. By recognizing such an attack at the beginning can reduce the damage caused by these attacks. Even so, such an attempt is extremely difficult on the networks where the traffic is very high. Furthermore, people who were determined to take down a particular network service will definitely do a lot of homework and can cause much more damage than a general denial-of-service attack can.

However, there are a lot of mechanisms available today to identify the denial-of-service attacks. One such method is entropy based detection scheme. In entropy based detection scheme, packet size entropy based scheme is much faster and easy to implement. Even so, there are some shortcomings as well to this method. This thesis introduces a new parameter to the packet size entropy based DoS attack detection scheme so that it can improve the detection accuracy. The new parameter is the entropy of the source and destination IP address combination. I.e. a concatenation of both addresses will give a hash like value, which can uniquely identify a particular path. By this parameter, even if the attacker changes the packet size using simple application programs for packets such as ICMP, the attack can be detected.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Denial-of-service attack can be simply explained with the analogy of the telephone network. A telephone number can be easily attacked by calling to that number by a number of people simultaneously, which in turn do not give access to a legitimate caller. A denial-of-service attack is a malignant attempt by a single person or a group of persons to disrupt an online service. Denial-of-service attacks have caused huge financial losses in recent years in the Internet. Denial-of-Service attacks affected businesses on websites like eBay.com, amaon.com, yahoo.com, ZDNet.com, Buy.com and a lot of other similar websites [1, 2].

Most of the attacks that come under a denial-of-service are bandwidth attacks. The attackers generate a huge traffic in the network and overload the network with unwanted or bogus Internet packets. Detection of bandwidth attack is difficult when the detector is far from the victim. But it becomes easier when the detector is placed near to the victim.

Recently, a lot of denial-of-service attack detection schemes have been proposed. Most of these schemes come under volume-based scheme or feature-based scheme. Volume-based scheme needs a detectable disruption in the traffic volume. When the attack is done gradually, then there is a possible vulnerability in some volume based scheme. On the other hand, feature-based scheme detects the attack by inspecting the header information. It checks the header, and some schemes even check the

data parts as well to detect any possible anomaly in the traffic. But the checking of every single packet is time consuming and if the traffic is very high, it becomes very difficult. Feature based-schemes are most accurate in detection, but they are notoriously processor hungry.

This thesis focuses on another approach, which takes the positive sides of both volumes-based approach and feature-based approach, detecting denial-of-service attack using packet size distribution. The method only uses the entropy of the packet size, and when there is a spike in the packet size entropy- time series, it could be a potential denial-of-service attack. This method can be slightly improved by adding a new parameter called eSD. When eSD time series gives a spike in the series, we say that it could be a possible denial-of-service attack. At the same time, the packet rate in the network should exceed the threshold value. If the packet rate is low and there is a spike in the series is not considered as a denial-of-service attack.

1.1 Thesis Organization

In Chapter 2 we are discussing about the network attacks in general, denial-of-service attacks in particular and the methods that are available to identify a denial-of-service attack. Chapter 3 discusses about the motivation and objective of the thesis. A new scheme is proposed in Chapter 4 from the lights of literature reviews. The experiments that are conducted on the new scheme and the results are discussed in chapter 5. In Chapter 6 we conclude the thesis.

Chapter 2

Overview of Network Attacks and Detection Techniques

2.1 Network Attacks

An attack is usually perpetrated by someone with bad intentions. Generally such people are known as black hat hackers. There are people who do attacks to find out the flaws in the network, such an attacker actually doing penetration testing. While there are many specific ways to launch attacks to a network, there are three general types of attacks:

Reconnaissance Attacks

Access Attacks

Denial-of-Service Attacks

These attacks are not discrete. These attacks can be used in combination to meet the goals of the malicious attacker

2.1.1 Reconnaissance Attacks

Reconnaissance attacks [3] are used to gather information about a target network or a system. Such an attack may seem harmless at the time and may be overlooked by network administrators as network noise, but it is usually the information gained through reconnaissance attack that is used in subsequent access or denial-of-service attack.

Several means may be used to gather information about an organization and could include automated and technological attacks as well as human social attacks. Examples might include ICMP ping sweeps against a network or SNMP walking technologies to gather network map and device configuration data. Likewise, application level scanners could be used to search for vulnerabilities such as web server CGI or ASP weakness.

No specific damage may be caused by reconnaissance attack, but it is analogous to burglar staking out a neighborhood, watching for times of inactivity, and occasionally testing windows and doors for access. Reconnaissance attacks are common and should be considered as a serious threat to an organization as they may give potential attackers the information required to perform access or denial of service attack.

2.1.2 Access Attacks

As the name implies, access attacks are those involving the unauthorized use of a target machine or machines. The means by which an intruder gains access to infrastructure are typically specific to the exploitable vulnerabilities present in the operating system, application software or physical protection mechanism. Often these vulnerabilities are discovered by hackers during reconnaissance attack.

Access attack can be manual or automated and may be composed of unstructured or structured threats. Access attacks are categorized into data retrieval attacks, system access and privilege escalation. The first form of access attack is the unauthorized data retrieval in which information is read, copied or moved to a

system. The data retrieval access attack is a common form of internal threats and is largely the result of poorly configured file and directory permissions. For instance, world readable Windows file shares or Unix NFS directories are relatively simple ways for unauthorized users to gain access to potentially sensitive data such as accounting or human resource information. Use of proper mounting or access permission and even encryption could prevent such access.

System access attacks occur when an attacker has Operating system level or actual log-in access to a device. Such illegal access could be achieved through weak or non-existent passwords or through known exploits against operating system vulnerabilities. Many secondary attacks could result from illegitimate system access. For example, compromised machines could be used to target other machines. Once a hacker obtains access to the system, he or she could attempt for privilege escalation.

Attaining higher privilege of a system allows hackers to perform far more dangerous actions. Once an intruder has system access, they often seek super-user or root privilege to install Trojan code or create a backdoor for future covert access. Privilege escalation is often acquired via operating system or application vulnerabilities such as buffer overflow attacks. Once a system has been compromised in this manner, it is completely at the control of an attacker.

2.1.3 Denial-of-Service Attacks

A third form of network attack is known as denial-of-service attack [4]. Here the attacker seeks to prevent the legitimate use of service or system. Often times, this is accomplished by overwhelming an infrastructure with bogus requests for service. Denial-of-service attacks can also be caused by corrupted data or configurations. For instance, a denial of service attack could be the result of an intentionally corrupted BGP protocol routing configuration. If the attacker changed the network advertisement, authentication attributes or anonymous system number (ASN) parameters of an organization's routing equipment, that organization could simply disappear from the internet or worse yet, traffic destined to that organization could

be routed to illegitimate remote locations on the Internet. Denial-of-service can also be dispersed so that numerous compromised machines launch a denial-of-service attack on the same target service or host known as a Distributed denial-of-service attack, such events are extremely difficult to combat since it is impossible to ascertain the difference between the legitimate and illegitimate traffic.

2.2 Overview of Denial-of-Service Attacks

2.2.1 ICMP flooding Attack or Ping to death

A Denial of Service attack that sends large amounts of ICMP packets to a victim in order to crash the TCP/IP buffer on the victims machine and cause it to stop responding to TCP/IP requests is called an ICMP flooding attack or Ping flooding attack [5, 19]. Now a days servers becomes more and more powerful. So this technique does not cause more damage to a victim nowadays. The attacker wants to magnify the amplitude of attack by increasing a number of attacking packets. The solution is a Distributed Denial-of-Service attacks (DDoS).

In a Distributed Denial-of-Service (DDoS) [17, 18] the attacker attacks a victim from multiple source systems. The attacker uses a large number of compromised machines to send bogus packets at the same time to crash a victim or Internet connection. These packets must send in masses to break down the system. The attacker needs is a botnet that contain a large number of compromised machines. These machines will be waiting for the command from the attacker. One it got a signal from the attacker, the machines will start sending requests to the server. As a result, a large number of packets, which come from many compromised computers could break down the victim system. The magnitude of the attack is based on the number of compromised computers in the botnet.

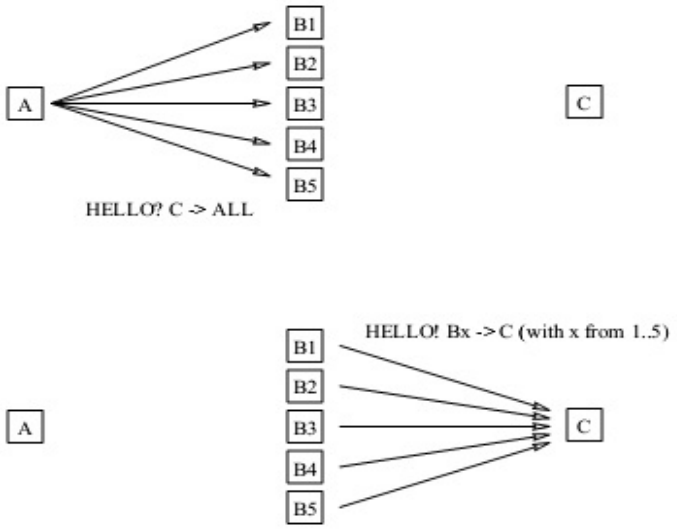

Figure 2.1: Smurf Attack

2.2.2 Smurf Attack

Smurf attack [6, 11] is similar to ICMP flooding attack. But the way in which it creates a large number of ICMP request is different. Here the attacker sends a large amount of ICMP echo to a broadcast address of the network (i.e. x.x.x.255 in class C of IPv4 type) and uses a victims IP address as the source IP address. Since the source IP is that of the victim, the reply-message from all computers in that network that respond to the broadcast address will flood the victim. There are two parties affected by this attack: the spoofed address target known as the victim and the broadcast router known as an amplifier. The victim is the target of a large amount of traffic that the amplifiers generate. The intensity of the attack depends on the number of hosts behind the router that reply to the ICMP echo packets.

The following are steps of SMURF attack;

1. The attacker forges the source IP address to be the victims IP address and sends ICMP Echo Request packets.

2. These ICMP Echo Request packets are sent to the broadcast address of the router. These packets are broadcast to all computers that are connected to the router.

3. All the machines that are alive on that network send an ICMP Echo Reply packet back to the spoofed source IP address of the victim. The amplification factor can be very large, as the number of machines alive in the network is large.

2.2.3 IP Spoofing

Spoofing of IP addresses [7] is lying about ones own IP address. An application program fills the header fields of the IP packet with any IP address it wants while writing to a raw socket. Root permission is required to do such actions which is always known to a user running Linux on a PC. If routing is purely based on the IP destination address only, it wont check the Source IP address. In Reflection attacks, attackers use one specific IP source address on all outgoing IP packets to make all returning IP packets go to the unfortunate owner of that address. The main use of IP spoofing is to hide the location of attacker in the network.

IP spoofing is now efficiently eliminated by Ingress/Egress filtering which is performed by routers. The source address of the outgoing packet is matched against a group of IP address or it checks whether the Source IP address belongs to that particular network. It is done by Routers. And if conditions are not satisfied, then the packet is dropped. For example, a router at NITRourkela will only route outgoing packets that have an IP source address from subnet 192.168.1.0/22. IP spoofing is not only a weapon to hide the identity of an attacker, it can do much more than just that. In these days instead of spoofing IP address attackers are using botnets [20,21] to get different IP addresses. So in that respect IP spoofing becomes less important in such scenarios. Ingress/egress filtering mechanism makes the life of an attacker more difficult, but it is far from being a magic bullet.

A new scheme was devised by Stefan Savage et al [8]. Their method can successfully trackback the packets. Their IP Traceback scheme assists in tracking down attackers postmortem. It requires routers to probabilistically mark packets such that the receiving end can reconstruct the route that packets followed. But it requires a minimum number of packets to be sent from the source. But this method is useful only to identify the source of the attack. But this method can be integrated to other schemes so that the scheme can identify, trace back and block the attack.

2.2.4 TCP-SYN Attacks

Normally, in a client server mode when an Internet user wants to request a service from the server the connection is set up in three steps. The user sends an establish-message to let the server know his/her request. The server ends an acknowledge-message back to the user and waits for the reply-message from the user until the connection is established or the timeout period is expired. The third step is to open a socket port by the server and waiting for a reply-message from that user. This state is known as server half-open state. The TCP-SYN attack technique [9] uses this weakness of TCP/IP by sending request-message with a large amount of data and source address from a spoofed address that does not exist.

To succeed a denial-of-service attack by using this method, the attacker launches an uncompleted three-way handshake by sending a large amount of spurious request messages to start a connection and waits for the server to reply acknowledge back to the attacker. Since the address doesn't exist, the server will not receive any messages back from the attacker. Because the three-way handshake must complete in three steps of exchanging messages between a server and client, for a while the server will be waiting for a third message from the attacker. The attacker can send a large number of bogus requests to open a connection with this method to the server and the server will waste a lot of resources to service these bogus messages and cannot

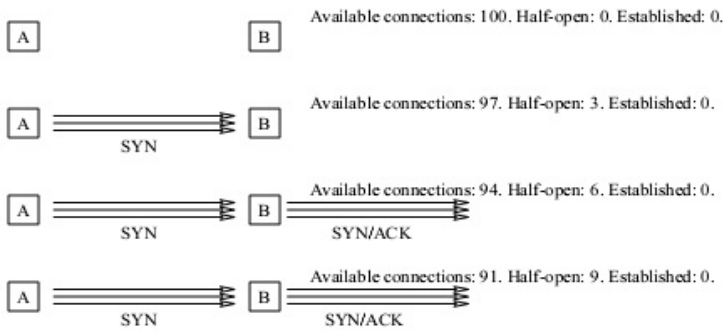

Figure 2.2: TCP-SYN Attack

serve authentic users.

2.2.5 Trinoo Attacks

Trinoo attacks [10] are more sophisticated than smurf attacks. First you need a compromised system. Once you got a compromised system, the attacker will install a small daemon application. This application will wait for the master's command and stays in the compromised system. The compromised system is called a zombie machine. The communication between the master and the zombies is often encrypted so as to complicate matters for network intrusion detectors. The master can instruct all the zombies to start sending UDP packets to one destination at any point of time, which can initiate a denial of service attack.

2.2.6 Stealth Bombs

The bandwidth attacks can be simply generated by a huge amount of normal TCP/IP traffic. By running a JavaScript in a browser that pops up a few dozen windows each fetching a Web page from one server means certain death for that server if a few thousand people are willing to run this script in their browser simultaneously [12]. Self-replicating e-mail viruses can spread such kind of scripts.

Stealth bomb is different from a flash crowd. When a large number of genuine users overloads a Web server by sending genuine requests is called a flash-crowd. A website that shows the Football World Cup score could be an example of such kind.

2.3 Overview of Intrusion Detection System

Intrusion Detection system comes in many shapes and sizes. Some are small one rack unit appliances that tuck neatly into a server rack while others modulus such as Cisco ISDM, that insert directly into active network component. Some IDS are simply software applications that run on servers or workstations. Their general purpose is to monitor events on systems and network and notify security administrators of an event that the sensor determines is worthy of alert. An IDS weights these situations using a variety of means. Some IDS compare network conversations they hear to a list of known attacks sequence or signatures. When a network traffic matches a known exploit signature, they trigger an alert. These IDS are known as signature based IDS. Other IDS collects a baseline of normal network operations over time. They then continue to monitor the network for the situations that don't match what they have determined as normal. If this happens, they trigger an alert. These IDS are called anomaly based IDS. Some IDS can perform automated actions beyond simply sending alerts, such as resetting malicious connections by using a technique called TCP Reset, blocking offending source address or shunning the IP address. Some of the more advanced IDS can even reconfigure ACLs on routers and firewalls automatically.

There are several types of IDS that can be deployed to aid security administrators in their endeavors. Two types are most prevalent in modern security deployments. They are network based intrusion detection system and host based intrusion detection system. There are other types of intrusion detection systems which includes file integrity checkers and log file checkers, and decoy devices known as honeypots. Additionally there exist hybrid systems that combine some of the

different functionality mentioned above.

2.3.1 Network IDS

NIDS are devices intelligently distributed within networks that passively inspect traffic traversing the devices on which they sit. NIDS can be hardware or software based systems and depending on the manufacture of the system, can attach to various network media such as Ethernet, FDDI and others. Oftentimes NIDS have two network interfaces. One is used for listening to the network conversations in promiscuous mode and the other is used for control and reporting.

With he advent of switching, which isolates uni-cast conversations to ingress and egress switch ports, network infrastructure vendors have devised port mirroring techniques to replicate all network traffic to the NIDS. There are other means of supplying traffic to the IDS such as network taps. Cisco uses Switched Port Analyzer functionally to facilitate this capability on their devices and, in some network equipments, includes NIDS components directly within the switch.

There are many NIDS vendors, all systems tend to function in one of the two ways; signature based or anomaly based. Both are the mechanism that separates benign traffic from its malicious brethren. Potential issues with NIDS includes high speed network data overload, tuning difficulties encryption and signature development lag time.

2.3.2 Host IDS

HIDS are systems that sit at the server end point rather than in the network transit point like NIDS. The first type of IDS that is widely implemented is installed on servers and is more focused on analyzing the specific OS and application functions residing on the HIDS host. HIDS are often critical in detecting internal attacks directed towards an organization's server such as DNS, mail and web servers. HIDS can detect a variety of potential attack situations such as file permission changes

and improperly formed client-server request.

File integrity and log file checking agents are a form of HIDS that focuses on the OS binary files and the log files normally produced by OS based security mechanism such as login logs. Log file checkers run regularly as well and parse system and application logs to search for signature based alerts. For instance, multiple failed log in on a server would typically be detected and reported by log checkers.

2.3.3 Hybrid IDS

HIDS and NIDS are most commonly deployed forms of IDS, other types of IDS such as hybrid IDS and honeypots can be useful tools in detecting potential security situations. Hybrid IDS are systems that combine both HIDS and limited NIDS functionality on the same platform. A hybrid IDS can monitor systems and application events and verify a system's integrity like HIDS, yet because the monitoring of network interface runs in a non-promiscuous mode, the NIDS functionally only serves to analyze traffic destined for the device itself. A hybrid IDS is often deployed on an organization's most critical servers.

2.3.4 How IDS Works ?

All IDS are monitoring tools that serve an essential service they detect potential security events and alert administrators. Some IDS can even help perform automated actions such as issuing an Access Control List (ACL) updates to firewalls. The IDS should not be confused with perimeter security devices such as firewalls since they do not directly prevent security events through blocking and authentication mechanism themselves. NIDS are deployed where services or important traffic traverse network devices.

All network traffic that traverse these DM2 switches is inspected by the IDS via its monitoring interface. The NIDS could obtain data from a DM2 switch via a

Cisco RSPAN port or a network tap. Alternatively, the NIDS could be an internal component of the switch such as a Cisco IDS switch module.

Regardless should an attacker intimate a series of malicious actions against servers on DM2 network, it might unfold in this way:

1. A hacker working from a remote workstation on the internet begins an attack on the DM2 located target host. Perhaps the hacker is using freely available software to scan for open Windows file shares. Since the connection between the attacker and the target host traverse the DM2 switch, the NIDS hears the attacker's attempt to scan and mount unprotected Windows file shares via its monitoring interface.

2. Since the NIDS is monitoring of the network and actively comparing all the traffic against a predefined attack signature, it detects the attackers scanning attempts. Depending on how NIDS is configured, several outcomes could result at this point. He IDS could simply send an alert to the administrator via Control and Reporting Interface. Alternatively, the NIDS could automatically reset the attacker's connection or add rule set for the firewall or router to deny the attacker further access.

3. After NIDS sends an alert via its control and Reporting interface, administrators can take actions based on security policy. This may mean manually placing deny statements in the firewall rule set to deny the attacker or reporting the attacker to management and the proper authentication depending upon its severity.

Signature Based IDS

The most prevalent form of intrusion detection is through signature matching. Referred to as signature based IDS, these systems monitor the network or server and match packet traffic attributes against a set of pre-determined attack lists or

signatures. Should a particular network conversation match a signature configured on the IDS, the system alerts administrators or takes other pre-configured actions.

Signature based IDS can be quite effective in security monitoring, yet they have several drawbacks. To detect most potential attacks, the signature database on the IDS must be large. As the speed of the network increases,it is difficult for signature based IDS to keep pace with network traffic. Typically the signature based IDS must be re-tuned by removing some of the signatures from the active database before use. While this permits the IDS to function properly, it does so at the risk of missing potential attack. Similarly, because these IDS only alert administrators as to potential attacks for which it has a signature, new vulnerabilities and exploits will not be detected until the vendors or administrators develop new signatures.

IDS must be properly tuned once they are in the network environment. Because each signature within an IDS consumes system resources, it may not be advisable to load all signatures based on your network requirements and services. For instance, if you don't run a specific service or block access to the service perimeter securing devices, it might not be necessary to monitor for potential attacks against that service.

Anomaly Based IDS

Anomaly based IDS does not use static signatures to detect potential security events. Rather these IDS use network traffic base line to determine a Normal state of the network and compare current traffic to that baseline. If network anomalies occur, the IDS alerts security administrators.

Two types of anomaly based systems exist, behavior anomaly and protocol anomaly IDS. Both use the same type of statistical calculations to determine whether current traffic deviates from Normal traffic, yet they specifically track different attributes. Behavior anomaly system tends to monitor network resources using timing, volume and similar resource characteristics while protocol anomaly IDS typically monitors application level threats such as RFC complacency and other

operational protocol content attributes.

As compared to signature based IDS, an anomaly based IDS has the potential to detect new attack vectors as they occur. Anomaly IDS, however, can suffer from numerous false positive as security administrators attempt to determine the dynamic definition of Normal network operations.

2.4 Overview of Detection Techniques

2.4.1 Activity Profiling

Observing a network packet's header information offers an activity profile. This activity profile is the average packet rate for a network flow, which consists of consecutive packets with similar packet fields (such as address, port,and protocol) [13]. The elapsed time between consecutive matching packets determines the flow's average packet rate or activity level. Total network activity can be measured as the sum over the average packet rates of all inbound and outbound flows.

To analyze individual flows for all possible UDP services it is required to monitor in the order of 2^{64} flows, and including other protocols, such as TCP, ICMP, and Simple Network Management Protocol (SNMP) greatly compounds the number of possible flows. To avoid high conventionality issues, individual flows with similar characteristics can be clustered. Each cluster's activity level is the summation of constituent flows.An attack is indicated by increasing activity levels among clusters, which can indicate a few attacking agents increasing their attack-generation rate. Attacks that use uniform address distributions will maximize the entropy statistic, whereas one large voluminous flow will minimize the entropy. Thresholding an entropy deviation from the expected traffic's source address profile can suggest anomalous activity.

2.4.2 Sequential Change Point Detection

Change-point detection algorithms isolate a traffic statistic's change caused by attacks. These approaches initially filter the target traffic data by address, port, or protocol and store the resultant flow as a time series. The time series can be considered a time-domain representation of a cluster's activity. If a DoS flooding attack begins at time t,the time series will show a statistical change either around or at a time greater than t. One class of change-point detection algorithms operates on continuously sampled data and requires only low amounts of memory and computational resources. An example here is cumulative sum (Cusum) algorithms.

2.4.3 Wavelet Analysis

Wavelet analysis describes an input signal in terms of spectral components. Although Fourier analysis is more common, it provides a global frequency description and no time localization.Wavelets provide for concurrent time and frequency description, and can thus determine the time at which certain frequency components are present. For detection applications, wavelets separate out time-localized anomalous signals from background noise; the input signal contains both. Ideally, the signal and noise components will dominate in separate spectral windows. Analyzing each spectral windows energy determines the presence of anomalies [14].

Chapter 3

Motivation and Objective

The Internet consists of hundreds of millions of computers connected across the world running on multiple hardware and software platforms. It assists in personal and professional needs of the people and corporations. Nevertheless, attackers use this inter-connectivity between computers to misuse resources and mount denial-of-service (DoS) attacks against arbitrary sites. A malicious user exploits the connectivity of the Internet to sideline the services offered by a victim site, mostly by flooding a victim with many requests. There are sophisticated attack tools that automate the procedure of compromising hosts and launching attacks are readily available on the Internet, and detailed instructions allow even an amateur to use them effectively.

Denial-of-service attacks cause substantial financial impairment every year, which makes it necessary to formulate techniques to detect and respond to attacks quickly. Development of effective response techniques requires intimate knowledge of attack dynamics; yet little information about attacks in the wild is published in the research community. Moore et al [15] provides insight into the dominance of DoS activity on the Internet, but their analysis is based on backscatter packets and lacks the level of detail required to study attack dynamics or generate high-fidelity models needed for DoS research. Monitoring tools today can detect an attack and identify basic properties such as traffic rates and packet types.

We had mainly volume based approach and feature based approach to detect the Denial-of-Service attack. Feature based approaches work fine only with low traffic networks. But Denial-of-Service attacks itself are heavy on traffic volume. So the performance of this approach is a main concern. It requires a real time examination of each and every packet passing through that network. The detection schemes are now moving to the entropy based approaches. Entropy based approaches are faster and easier to implement. Ping Du and Abe introduced a method to detect Denial-of-service attack efficiently using entropy based approach. Many applications have their typical packet size. For example, FTP has 1500 bytes data packets and 40 bytes of acknowledgment packet. Du and Abe observed this fact and proposed an IP packet size entropy for detecting DoS attacks. The concept here was to investigate a similarity of IP packets and uses it as a packet size entropy. By examining the time series of packet size entropy, any changes to cause some spikes in the observed time slot will be identified as a denial-of-service attack. This method is able to detect both long term and short term attacks.

Abe and Du implemented their approach with an assumption that most of the application packets will be having same packet size. That assumption is true in most of the cases. But when we were observing the ICMP attacks it is found that for ICMP packets, we cannot take this assumption. The packet size of ICMP packet can be vary from 32 byte to the MTU (Maximum Transfer Unit)of the network. So when the attacker uses this method to launch an attack, the IPSE scheme becomes vulnerable. ICMP packets are generally used for network error checking and troubleshooting. At the same time, this ICMP packets are weapons of the hackers. They use ICMP packets to flood the network. ICMP packets are also used by hackers to trace the network path and prepare network maps.

The major purposes of this thesis are,

1. To propose a new entropy based scheme for detecting Denial-of-Service attacks.

2. To show that the new scheme is an improved version of the predecessor scheme proposed by Abe and Du.

3. To check the working of the scheme on real time network traffic.

Chapter 4

Proposed Scheme

Most of the time the attacker launches a DoS attack by sending a large amount of bogus data to interfere or disrupt the service on the server. Using a volume-based scheme to detect such attacks would not be able to inspect short-term denial-of-service attacks, as well as cannot distinguish between heavy load of legitimate users and huge number of bogus messages from attackers. Enabling early detection of Denial-of-service (DoS) attacks in network traffic is an important and challenging task because Denial-of-Service attacks have become one of the most serious threats to the Internet. There are methods based on packet size entropy detection. Here what we are introducing is a hybrid approach which will use address distribution as well as packet size entropy.

Abe and Du, in their paper *"Detecting DoS Attacks using Packet Size Distribution"* [16], proposed a scheme based on packet size distribution and the packet rate. Their assumption is that when there is an attack on the network, the entropy of the system reduces suddenly. In their approach when the traffic rate and entropy of the packet size exceeds a threshold value, then the scheme identifies an attack. The equation to calculate the packet size entropy of the traffic is [16, 25],

$$H_s(t) = -\sum_{l}(\frac{n_l}{S})\log(\frac{n_l}{S}) \tag{4.1}$$

Where n_l is the number of packets having similar size l in the observation time frame and the observation window contains S packets at time t.

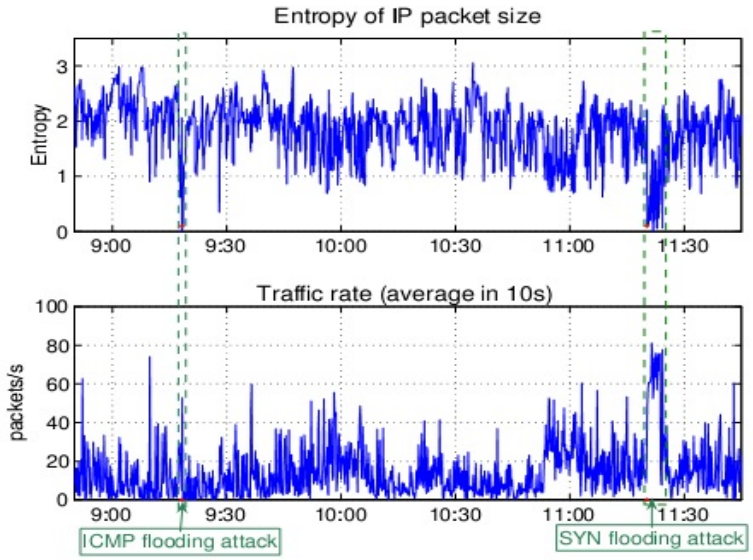

Figure 4.1: Du and Abe Detection technique viewed in terms of entropy (top plot) and volume (bottom plot)

When there is an attack on the network, the IPSE scheme shows spikes in both the packet size entropy series and the packet rate. Since this scheme takes into consideration only the packet size from the IP header. This scheme is very faster since there is no other calculations or processing is required to identify the attack. But the design flaws in the ICMP packets can be exploited by the attacker to evade from the detection. The ICMP is a troubleshooting and error-reporting tool. The payload of ICMP can be of any size depending on the MTU of the network, and it can be of variable size. This exploit can be used to launch an attack on the network using an application that using an application that generates variable size payload in an ICMP packet and send to the victim in different packet rates. Such an attack will be invisible to the Abe and Dus scheme.

To counter such an attack, we introduce a new parameter called eSD. This is the entropy of source address and destination address combined. Unlike DDoS attack,

in DoS attack the attack comes from a single source address. So the source address of the attacker remains the same. And the address pool that the attacker can use for a spoofed attack will be small because of the ingress/egress routing. Only the packets that belong to the source network are routed to the network, otherwise the packets are omitted by the router. So by taking the eSD value as a new parameter to identify the denial-of-service attack will help in types of attacks that change the value of the packet size.

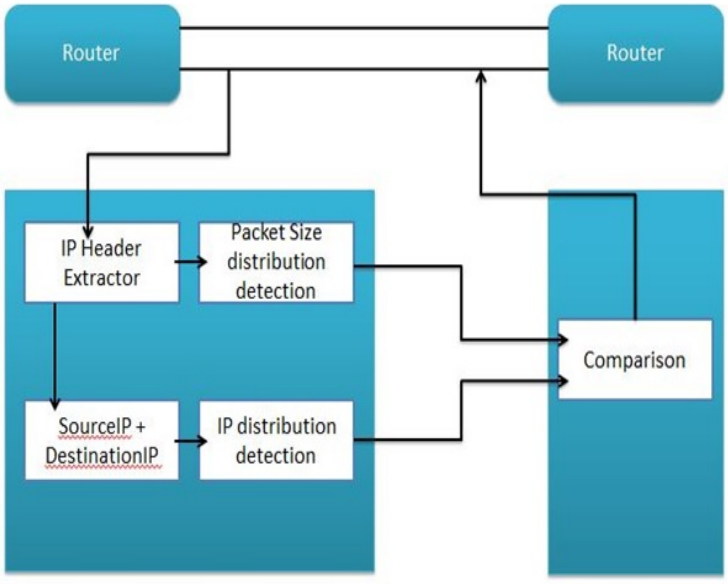

Figure 4.2: Block Diagram of the improved detection scheme

The proposed scheme works in the same way as that of the existing scheme. The additional component is the IP address extractor and IP distribution calculator. From the captured packet, the detector will extract the IP header. From the IP header, to find the packet size distribution, we will extract the length bytes and calculate the distribution of packet length for a time frame suppose 10sec. In the same time frame, we will extract the source and destination address from the IP packet and make it into a hash value so that for each source destination combination, we will get a unique value. This can be done by extracting the source and destination IP address bytes from the packets, which will be of 32 bits each. Since both are

stored in a sequential format, we can extract 64 bits together and convert it into a BigInterger value. This will reduce time and processing cycle required to concatenate and create hash values. Once the hash values are generated, we can calculate the entropy of this address distribution. This can be calculated using the equation 4.2. The entropy for the source destination address hash value is called eSD value.

$$eSD = -\sum_{i=1}^{N}(P_i)\log(P_i) \tag{4.2}$$

Here P is an array of normalized frequencies of hashed address bin distribution, collected over the time window.

$$P_i = \frac{n_i}{\sum n_i}, i = 1, 2, 3..... \tag{4.3}$$

The existing scheme works in such a way that it checks the packet rate and packet size entropy value together and if both values exceed a threshold value, it detects an attack. In the proposed scheme to alarm an attack situation, we need either Packet rate and packet size entropy or packet rate and eSD value to exceed a threshold value. In both situation, the new scheme alarms an attack. In general, the new parameter we introduced acts as an add-on to the existing scheme.

Chapter 5

Experiments and Results

The Experiment is done in two steps. The first step is to check the correctness of the implementation. For that purpose, we have tested the new scheme with DARPA/MIT Lincoln Laboratory data-set, which is a well known data-set for denial-of-service attack detection testing. The second step is to prove that our method is superior to the existing scheme and also to the check the working of the existing scheme and new scheme in real-time environment. For real-time traffic testing, we have checked our scheme and the existing scheme with our experiment setup and did manual attacks using freely available tools from the Internet. The number of attacks detected by both the methods are compared.

5.1 Checking the Correctness of Implementation

Before conducting a real-time testing using real-time traffic data and coming to the conclusions, we need to check that our implementation is correct. For that purpose, we have taken a well known data set, which is used for checking the denial-of-service attack detection schemes and conducted denial-of-service detection. The DARPA/MIT Lincoln data-set [23] contains five weeks of captured traffic, which is filtered and prepared for DoS attack detection scheme testings. It contains normal traffic in the week 1 and week 3, and attacks in the week 2,week 4 and week 5.

An intrusion detection system works in two stages. One is to find the normal traffic baseline. The second stage is to check for attacks in the ongoing traffic. To find the normal traffic baseline, we need traffic data that do not contain any attack. This baseline should represent the normal characteristic of the network traffic. From the offline data-set, we can take traffic details from week 1 and week 3, since these weeks are marked as attack free. By using equations 4.1 and 4.2, we can calculate the values of Packet Size Entropy and eSD of the network traffic when the network is functioning normally. The time frame that we are using here is 10 sec. so that we will get the traffic details more efficiently. In Abe and Du scheme, they observed that as the time window is increased, the performance of the detection scheme is improved. We also took packet rate of the network in normal traffic. These calculated values are plotted against a time series. This series will represent the network profile in the normal conditions.

We analyzed the network traffic of selected five days from the DARPA data-set and plotted the time series. A portion of the time series is shown in Figure 5.1. From the normal traffic values, we calculated the mean, median and standard deviation. These values represent the normal characteristic of the network. The calculated values are shown in Table 5.1. From these values, we can calculate the threshold values for Packet size Entropy and eSD using equations 5.2 and 5.3 respectively. The threshold value for packet rate can be calculated using equation 5.1. Calculated threshold values are given in Table 5.2.

$$PacketRate = Mean_{PR} + Median_{PR} + SD_{PR} \tag{5.1}$$

$$PacketsizeEntropy = \frac{(Mean_{EN} + Median_{EN}) - 2(SD_{EN})}{4} \tag{5.2}$$

$$eSD = \frac{(Mean_{eSD} + Median_{eSD}) - 2(SD_{eSD})}{4} \tag{5.3}$$

Table 5.1: MIT Lincoln Laboratory off-line traffic analysis

DARPA/MIT	Mean	Median	SD
Packet Rate	295	215	261
Packet Size Entropy	1.859529	1.818097	0.438062
eSD	1.612145	1.605362	0.362558

The DARPA data-set is filtered and tuned for testing of denial-of-service attacks detection schemes. DARPA Laboratory does not want to disclose all the network traffic details to outside world. So the data set may not behave exactly same as that of the real traffic.

Table 5.2: Threshold values for attack detection

Data-set	Packet Rate	Packet size Entropy	eSD
DARPA/MIT	871	0.7003755	0.6230978

The second stage we need to check the attack traffic for detecting attacks. We have three weeks of data that contain attacks. In the Lincoln Laboratory data-set documentation they have specifically written the locations of the attacks and the duration of the attacks. So to check the correctness of our implementation, we need to identify these mentioned attacks accurately. We have taken sufficient data from the data-set. Our selected traffic sequence contains 56 individual attacks.

The same procedure for calculating the normal traffic baseline is carried out for the attack detection as well. We checked for the attacks by checking whether the packet rate exceeds the normal threshold value. Once the normal threshold value for the packet rate is exceeded, we can check for the entropy values of both the packet size entropy and eSD value. In the existing scheme, a threat is identified when there is a spike in both the packet rate as well as the packet size entropy value. However, in our new improved scheme, we need excess packet rate along with either packet size entropy or the eSD value to exceed the threshold value. When either of the entropy is exceeded the threshold value, our new scheme will alarm a threat.

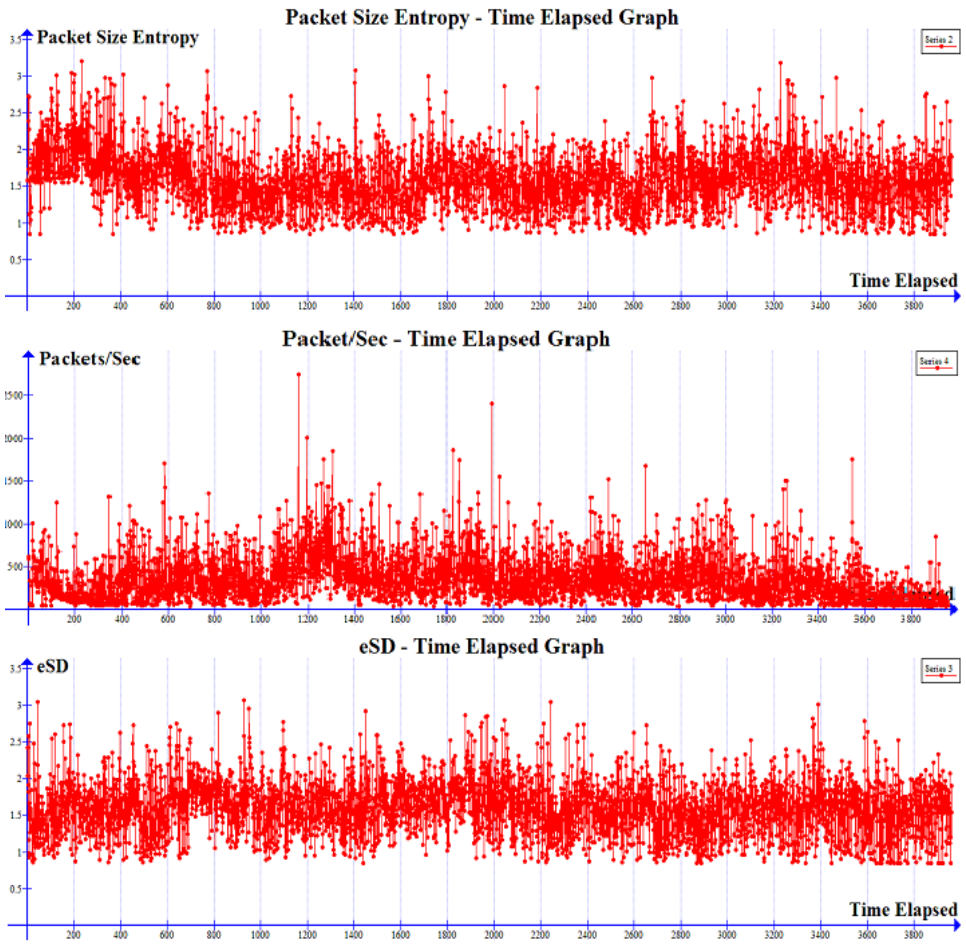

Figure 5.1: Analysis of Normal Traffic in DARPA/MIT Dataset

Our assumption is that a denial-of-service attack decreases the entropy of the overall system immediately. From the attack traffic, using existing scheme as well as the new improved scheme, we could detect all the 56 attack incidents that we were monitoring. A portion of the time series that contains attacks are shown in Figure 5.2. Our implementation of both the schemes proved to be correct. Now we can proceed with the same implementation for real-time traffic analysis. The only change required to change to real-time monitoring is that we have to change the source from the offline file to the NIC interface. The rest of the implementation remains the same. So with this program we will check the real-time traffic in an

environment that also contains attacks that are difficult to detect using Packet size entropy alone.

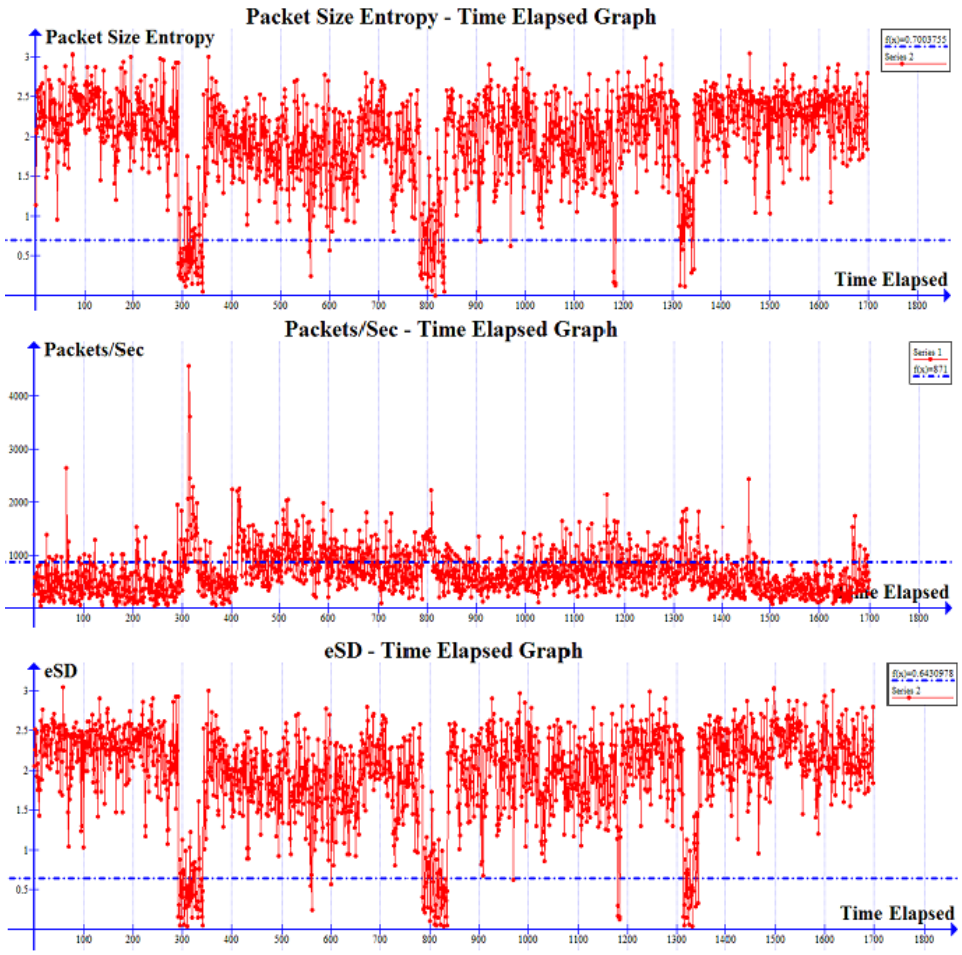

Figure 5.2: Analysis of Attack Traffic in DARPA/MIT offline dataset

5.2 Verification of the Improved Scheme

In the first part, we have checked the correctness of our implementation. The next step of the experiment is to prove that our new improved scheme is superior to the existing scheme. The offline data-set do not contain an attack that is invisible to the existing scheme and visible to the new scheme. So in the real-time environment

we need to generate such attacks as well to prove what we are claiming is right.

We have created a setup in the hostel network to generate denial-of-service attacks to a victim's machine using freely available tools from the Internet. And the IDS is installed on the victim machine. The experiment is carried out in the same way as that of the correctness checking. First, we need to generate the normal traffic characteristic of the network that we are monitoring. We monitored four-day traffic in the network that is going through the victim machine. Each day two hours of data is processed. Using these values, we generated the normal characteristics of the network that is being monitored.

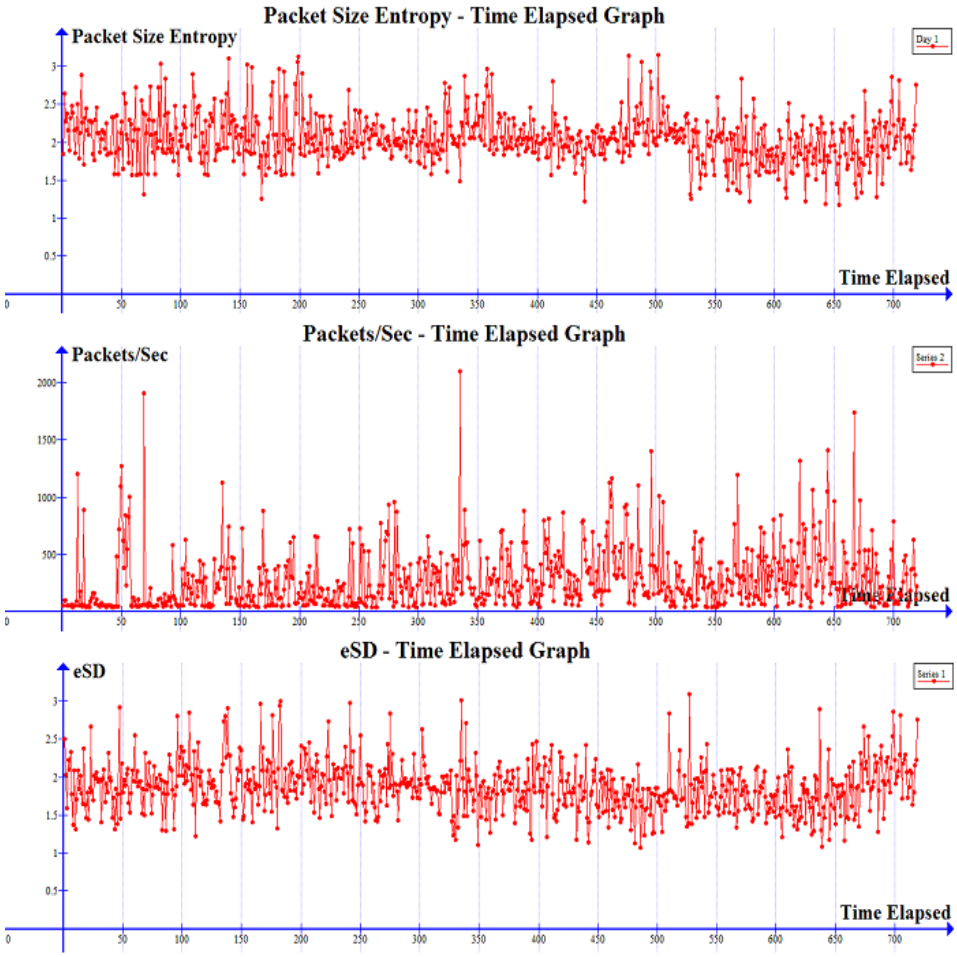

Figure 5.3: Analysis of Normal Traffic in real-time setup

As we have done in the first stage, here also we generated a time series of packet rate, packet size entropy and eSD values. The equations used are 4.1 and 4.2 to find the entropy. A portion of the time series is given in Figure 5.3. The mean, Median and Standard Deviation of these traffic data are calculated. This is the normal behavior of the network we are monitoring. These values are shown in Table 5.3.

Table 5.3: Real-time traffic analysis

NITR	Mean	Median	SD
Packet Rate	361	286	306
Packet Size Entropy	1.939656	1.925946	0.326357
eSD	1.787853	1.770041	0.323497

For this traffic, we calculated the threshold values to detect the attack using the equations 5.1, 5.2 and 5.3. These calculated threshold are given in Table 5.4.

Table 5.4: Threshold values for attack detection

Data-set	Packet Rate	Packet size Entropy	eSD
Real-Time Data	953	0.803222	0.727725

Here from the real-time traffic it is observed that eSD and the Packet size Entropy, both are different. But, in the case of offline traffic analysis, both the values remained the same.

Now it is time to analyze the new scheme in the real-time traffic. We monitored the network for 2 hrs. and calculated the entropy in real-time. Meanwhile, we generated a total of 7 attacks to the victim machine. Out of the seven attacks 5 of them are normal ICMP flooding attack and 2 of them were ICMP flooding with change in payload values. Abe and Du's scheme is expecting all the packets with same packet size. However, the attack is a different type. So it must be invisible to the existing Abe and Du's scheme.

As per our assumption, by plotting the values in the time series, it is found that these two attacks were invisible to the Abe and Du's scheme. At the same time, our new parameter responded positively all to the attacks that we have carried out. So as we have said, by using the eSD parameter, we can improve the detection accuracy of the existing scheme. The time series is given in Figure 5.4.

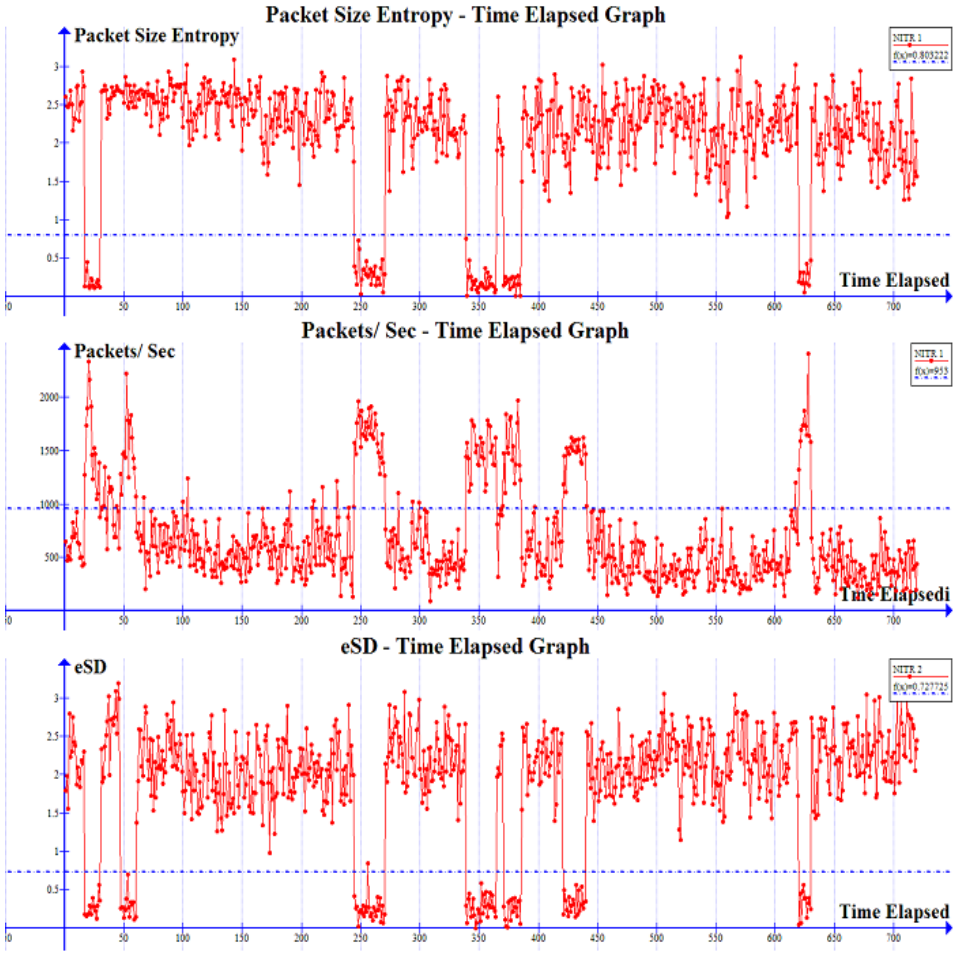

Figure 5.4: Analysis of Attack Traffic in real-time data

Even though eSD values alone identified all the attacks in the experiments we have conducted, we are not introducing the eSD parameter as a standalone parameter. We do not believe it can exist alone to detect all the attacks. Together with Packet size entropy eSD can identify accurately more attacks. Also Abes scheme

uses another equation for calculating the threshold to identify the attacks, and we are using another equation. So definitely this is a new scheme in comparison with the Abe and Du's scheme. And we need either Packet rate and packet size entropy or Packet rate and eSD values to exceed the threshold value. In such a situation, our scheme will alarm an attack situation.

Table 5.5: Comparative Analysis on Attack Detection

	MIT Lincoln Laboratory Data-set		Real-time Traffic Simulation Environment	
	monitoring	identified	monitoring	identified
Abe and Du's scheme	56	56	7	5
Improved Scheme	56	56	7	7

From the analysis of both the scheme for detecting the denial-of-service attack, it is proven that the new scheme is an improved version of the existing scheme. In our attack simulation environment, the existing scheme detected only five attacks out of 7. Whereas our new improved scheme detected all the 7 attacks simulated in the experiment environment. We have checked the correctness of the implementation by doing a correctness checking using well know denial-of-service attack data-set containing 56 attacks.

Chapter 6

Conclusion

This thesis introduces an alternative technique to detect denial-of-service attack by using packet size distribution and Address distribution. The major strength of the new scheme over IPSE-based Denial-of-Service detection is that it can detect denial-of-service attacks that come with altered packet size. We proved it using real-time traffic data. We are not claiming that our method is superior to all other methods. There are shortcoming as well in our approach. Experiments using DARPA/MIT Lincoln Laboratory offline-data set and our relatively small data-set won't cover all the attacks in the world. There are a lot of new attacking methodologies introduced by the attackers nowadays. All the methods are not available to the public due to security reasons, so it is difficult to study about the attack schemes and prevention mechanisms. Still a lot amount of data is available for academic purposes. Our method does not consider the distributed denial-of-service attack. Still we can identify Distributed Denial-of-Service attacks from the packet size distribution feature in our scheme. The main hurdle is a distributed denial of service attack that comes with different source addresses and comes with altered packets. Still a lot of research possibilities are available in this area.

Bibliography

[1] Sandoval, G. ; Wolverton, T.(2000, February 9). *"Leading Web sites under attack"* [online].Available: http://news.cnet.com/2100-1017-236683.html

[2] Wikipedia, Free Encyclopedia.(2012, July 11). *"Timeline of Internet conflicts"* [online].Available: http://en.wikipedia.org/wiki/Timeline_ of_ Internet_ conflicts#2000

[3] Udhayan, J. ; Prabu, M.M. ; Krishnan, V.A. ; Anitha, R. *"Reconnaissance Scan Detection Heuristics to disrupt the pre-attack information gathering "*. In International Conference on Network and Service Security, N2S '09, pages 1 - 5, June 24-26 2009.

[4] Carl, G. Kesidis, G. ; Brooks, R.R. ; Rai, S. *"Denial-of-service attack-detection techniques"*. Internet Computing, IEEE (Volume:10 , Issue: 1), pages 82 - 89, Jan.-Feb. 2006

[5] Udhayan, J. ; Anitha, R. *"Demystifying and Rate Limiting ICMP hosted DoS/DDoS Flooding Attacks with Attack Productivity Analysis"*. IEEE International Advance Computing Conference, pages 558 - 564, March 6-7 2009.

[6] Kumar, S. ; Azad, M. ; Gomez, O. ; Valdez, R. *"Can Microsofts Service Pack2 (SP2) Security Software Prevent SMURF Attacks?"*. International Conference on Internet and Web Applications and Services/Advanced International Conference on Telecommunications, AICT-ICIW '06, page 89 February 19 - 25 2006.

[7] Manusankar, C. ; Karthik, S. ; Rajendran, T. *"Intrusion Detection System with packet filtering for IP Spoofing"*. International Conference on Communication and Computational Intelligence (INCOCCI), pages 563 - 567, December 27-29 2010.

[8] Savage, S. ; Wetherall, D. ; Karlin, A. ;Anderson, T. *"Practical Network Support for IP Traceback."* Technical report, Department of Computer Science and Engineering, University of Washington, 2000.

[9] Ohsita, Y. ; Ata, S. ; Murata, M. *"Detecting distributed denial-of-service attacks by analyzing TCP SYN packets statistically "*. In IEEE Global Telecommunications Conference, GLOBECOM '04, pages 2043 - 2059, November 2004.

[10] Dittrich, D. *"The DoS Project's "trinoo" distributed denial of service attack tool"*. Technical report, University of Washington, October 21, 1999. [Online] Available : http://staff.washington.edu/dittrich/misc/trinoo.analysis.txt

[11] CERT Coordination Center. *"CERT Advisory CA-1998-01 Smurf IP Denial-of-Service Attacks"*. March 13, 2000. [Online] Available : http://www.cert.org/advisories/CA-1998-01.html

[12] Electrohippies Collective. *"Client-Side Distributed Denial-of-Service: Valid Campaign Tactic or Terrorist Act?"*. Leonardo Fights Back, MIT Press, Volume 34(3), pages 269-274, March 2001. [Online] Available : http://www.jstor.org/stable/1576948.

[13] Karamcheti, V. ; Geiger, D. ; Kedem, Z. ; Muthukrishnan, S. *"Detecting Malicious Network Traffic Using Inverse Distribution of Packet Content"*. Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data, MineNet '05, August 2005.

[14] Dainotti, A. ; Pescape, A. ; Ventre, Giorgio . *"Wavelet-based Detection of DoS Attacks"*. IEEE Global Telecommunications Conference, Volume 25(8), pages 1-6, 2006.

[15] Moore, D. ; Voelker, G.M. ; Savage. S. *"Inferring Internet Denial-of-Service Activity"*. ACM Transactions on Computer Systems (TOCS), volume 24(2), pages 115-139, May 2006.

[16] Ping Du ; Abe, S. *"Detecting DoS Attacks Using Packet Size Distribution"*. In 2nd Bio-Inspired Models of Network, Information and Computing Systems (Bionetics 2007), pages 93 - 96, December 10-12 2007.

[17] Honda, S. ; Nakashima, T. ; Oshima, S. *"Entropy Based Analysis of Anomaly Access of IP Packets"*. In 3rd International Conference on Innovative Computing Information and Control, page 101, 2008.

[18] Jin Wang ; Xiaolong Yang ; Keping Long. *"A new relative entropy based app-DDoS detection method"*. IEEE Symposium on Computers and Communications (ISCC), pages 966-968, December 2010.

[19] Rahmani, H. ; Sahli, N. ; Kamoun, F. *"DDoS flooding attack detection scheme based on F-divergence"*. Elsevier Computer Communications, Volume 35(11), pages 1380-1391, June 2012.

[20] Yu Chen ; Kai Hwang. *"Collaborative Change Detection of DDoS Attacks on Community and ISP Networks"*. IEEE International Symposium on Collaborative Techniques and Systems, pages 401-410, May 2006.

[21] Kumar, K. ; Joshil, R.C. ; Singh, K. *"A Distributed Approach using Entropy to Detect DDoS Attacks in ISP Domain"*. International Conference on Signal Processing, Communications and Networking, pages 331-337, February 2007.

[22] Hussain, A. ; Heidemann, J. ; Papadpoulos, C. *"A Framework for Classifying Denial of Service Attacks"*. Proceedings of ACM Conference on Applications,

Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM '03, August 2003.

[23] Lippman, R. ; Haines, J.W. ; Fried, D.J. ; Korba, J. ; Das, K. *"The 1999 DARPA Offline Intrusion Detection Evaluation"*. Computer Networks: The International Journal of Computer and Telecommunications Networking, volume 34(4), pages 579-595, October 2000.

[24] Yu Chen ; Kai Hwang ; Wei-Shinn Ku . *"Collaborative Detection of DDoS Attacks over Multiple Network Domains"*. IEEE Transaction on Parallel and Distributed Syatems, volume 18(12), pages 1649-1662, December 2007.

[25] Tritilanunt, S. ; Sivakorn, S. ; Juengjincharoen, C. ; Siripornpisan, A. *"Entropy-based Input-Output Traffic Mode Detection Scheme for DoS/DDoS Attacks"*. International Symposium on Communications and Information Technologies (ISCIT), pages 804-809, 2010.

[26] Al-Haidari, F. ; Sqalli, M. ; Salah, K. ; Hamodi, J. *"An Entropy-Based Countermeasure against Intelligent DoS Attacks Targeting Firewalls"*.IEEE International Symposium on Policies for Distributed Systems and Networks, pages 41-44, 2009.

[27] Sqalli, M.H. ; Firdous, S.N. ; Baig, Z. ; Azzedin, F. *"An Approach for Identifying Malicious Activities in Honeynet Traffic"*. 2011 International Conference on Cyberworlds (CW), pages 23-30, 2011.

[28] Kind, A. ; Stoecklin, M.P. ; Dimitropoulos, X. *"Histogram-based traffic anomaly detection"*. IEEE Transactions on Network and Service Management, Volume 6(2), pages 110-121, 2009.