DETECTION AND DEFENCE AGAINST TCP SYN FLOOD ATTACKS

Project Report submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology

Information Technology

By

Chiranshu Talwar (121405)

Anisha Gupta (121424)

Under the supervision

of

Ms. Ruhi Mahajan

To

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234,

Himachal Pradesh

CERTIFICATE

We hereby certify that the work presented in this report entitled **“ Detection and Defense against TCP SYN Flood attack”** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of our own work carried out over a period from August 2015 to May 2016 under the supervision of **Ms Ruhi Mahajan** (Assistant Professor, CSE).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Signature of Candidates :

Chiranshu Talwar	Anisha Gupta
(121405)	(121424)

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

Supervisor Name : Ms Ruhi Mahajan

Designation : Assistant Professor

Department name : CSE

Dated: 30/05/2016

ACKNOWLEDGEMENT

We take this opportunity to express our deepest gratitude and appreciation to all those who have helped us directly or indirectly towards the partial completion of this project.

First and foremost, we would like to express our sincere appreciation and gratitude to our project supervisor **Ms. Ruhi Mahajan,** Assistant Professor, Department of Computer Science & Engineering and Information Technology, Jaypee University Of Information Technology, Solan, for her insightful advice, encouragement, guidance, critics and valuable suggestions throughout the course of our project work. Without her continued support and interest, this project would not have been the same as presented here.

We express our deep gratitude to **Dr S.P. Ghrera**, HOD, Computer Science & Engineering and Information Technology, Jaypee University Of Information Technology, Solan, for his constant co-operation, support and for providing necessary facilities throughout this program.

Also we would like to express our thanks to teaching and non-teaching staff in the Department of Computer Science & Engineering and Information Technology, Jaypee University Of Information Technology, Solan for their invaluable help and support during this period.

Our special thanks to our parents, supporting families and friends who continuously supported and encouraged us in every possible way for the successful completion of this project.

Last but not the least; we thank God Almighty for his blessings without which the completion of this project work would not have been possible.

CHIRANSHU TALWAR (121405)
ANISHA GUPTA (121424)

TABLE OF CONTENT

Page No.

LIST OF ABBREVIATIONS

DoS	Denial of Service
Ddos	Distributed Denial of Service
IP	Internet Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
ICMP	Internet Control Message Protocol
SYN	Synchronized
ACK	Acknowledgement
TCB	Transmission Control Block
DARB	Delay Probing Method
TTL	Time to live
NS2	Network Simulator 2
NAM	Network Animator
IPSEC	Internet Protocol Security

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Over the years, the Internet has evolved from a tool for the research community to an indispensable network connecting over a billion nodes worldwide. Nowadays network growing in a rampant manner and uses as transfer medium like data, money transaction, information etc. Even though internet plays a vital role still there is some vulnerability eg. virus, spam, hacking, DOS, DDos, etc. There are many security threats existing on the Internet, one of them is the TCP SYN flood attack. We are focusing on Distributed Denial of Service; out of plenty of Denial of Service mechanism, we took SYN Flood attacks.

In this report , we will study the effect of denial-of-service attacks arising from TCP SYN flooding. SYN flooding attack has been widely observed world-wide, and occupies about 90% of the DoS attacks. We examine the effects of the attacks on individual host with the help of some useful parameters. We will detect the attack using ns2 tool .We will also compare the results of both the scenarios ,before and after the attack. Then, we will design the tool to prevent such attacks. The tool will contain the application of different algorithms and comparison of different approaches to fina an efficient defence algorithm against the TCP SYN flood attacks.

Finally, we will deduce the results which will detect and defend such TCP SYN flood attacks.

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

The aim of DoS attacks is to make services unavailable to legitimate users by flooding the victim with legitimate-like requests and current network architectures allow easy-to-launch, hard-to-stop DoS attacks. Nowadays every one relies on online transactions. These transactions involve one of the many types of denial of service attacks is known as TCP SYN Flood attack. Defending against those types of attacks is not trivial job, mainly due to the use of IP Spoofing and the destination-based routing of the Internet. Efficient packet filtering technique using firewall, SYN cookies and SYN cache are some solutions to defend TCP SYN Flood attacks.

1.2 GENERAL TERMS

1.2.1 Transmission Control Protocol/Internet Protocol (TCP) :

TCP/IP is the basic communication language or protocol of the Internet. It can also be used as a communications protocol in a private network (either an intranet or an extranet). When we are set up with direct access to the Internet, our computer is provided with a copy of the TCP/IP program just as every other computer that you may send messages to or get information from also has a copy of TCP/IP.

TCP/IP is a two-layer program. The higher layer, Transmission Control Protocol, manages the assembling of a message or file into smaller packets that are transmitted over the Internet and received by a TCP layer that reassembles the packets into the original message. The lower layer, Internet_Protocol, handles the address part of each packet so that it gets to the right destination. Each gateway computer on the network checks this address to see where to forward the message. Even though some packets from the same message are routed differently than others, they'll be reassembled at the destination.

TCP uses ‘3-Way Handshaking’ to establish a connection between client and a server.

1.2.2 3 -Way Handshaking:

A three-way-handshake is a method used in a TCP/IP network to create a connection between a local host/client and server. It is a three-step method that requires both the client and server to exchange SYN and ACK (acknowledgment) packets before actual data communication begins.

A three-way-handshake is also known as a TCP handshake.

A three-way-handshake is primarily used to create a TCP socket connection. It works when:

- A client node sends a SYN data packet over an IP network to a server on the same or an external network. The objective of this packet is to ask/infer if the server is open for new connection.
- The target server must have open ports that can accept and initiate new connections. When the server receives the SYN packet from the client node, it responds and returns a confirmation receipt - the ACK packet or SYN/ACK packet.
- The client node receives the SYN/ACK from the server and responds with an ACK packet.
- Upon completion of this process, the connection is created and the host and server can communicate.

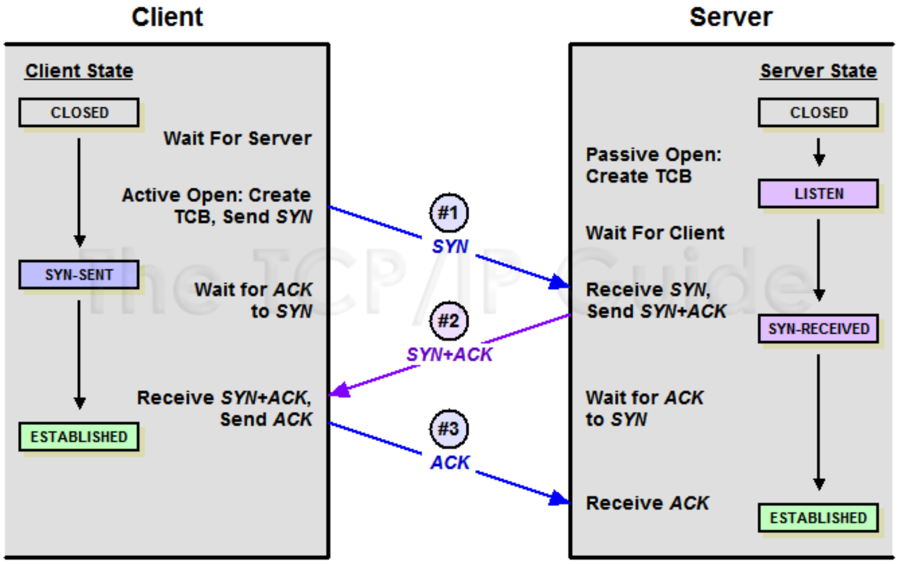


Fig. 1.1 3-way Handshaking

1.2.3 TCP SYN Flood Attack:

SYN flooding is an attack vector for conducting a denial-of-service (DoS) attack on a computer server. The attack involves having a client repeatedly send SYN (synchronization) packets to every port on a server, using fake IP addresses. When an attack begins, the server sees the equivalent of multiple attempts to establish communications. The server responds to each attempt with a SYN/ACK (synchronization acknowledged) packet from each open port, and with a RST (reset) packet from each closed port.

In a normal three-way handshake, the client would return an ACK (acknowledged) packet to confirm that the server's SYN/ACK packet was received, and communications would then commence. However, in a SYN flood, the ACK packet is never sent back by the hostile client. Instead, the client program sends repeated SYN requests to all the server's ports. A hostile client always knows a port is open when the server responds with a SYN/ACK packet.

The hostile client makes the SYN requests all appear valid, but because the IP addresses are fake ones, it is impossible for the server to close down the connection by sending RST packets back to the client. Instead, the connection stays open. Before time-out can occur, another SYN packet arrives from the hostile client. A connection of this type is called a half-open connection. Under these conditions, the server becomes completely or almost completely busy with the hostile client and communications with legitimate clients is difficult or impossible. For this reason, SYN floods are also known as half-open attacks. The transmission by a hostile client of SYN packets for the purpose of finding open ports and hacking into one or more of them, is called SYN scanning.

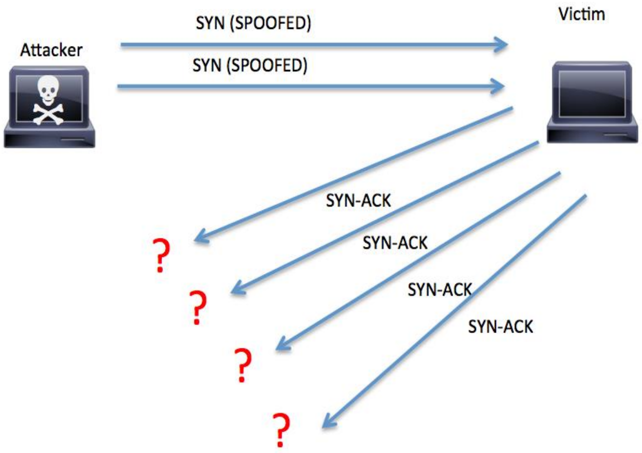


Fig. 1.2 SYN Flood Attack

1.3 PROBLEM STATEMENT

We are aiming to resolve the problem of TCP SYN flooding attack. A brief of overview of the issues, which provide the impetus of the work in this chapter is briefly as follows

. • Investigate the design of a simple network but sufficient to perform a denial-of-service attack using TCP SYN flooding.

• Investigate software and hardware entities required in the test-bed to facilitate TCP SYN flooding.

• Investigate a convenient method to generate TCP SYN packets in a number that is sufficient to cause a denial-of-service attack on the designed test-bed.

• Investigate how a host in a network can be said to experience a SYN flooding denial-of-service attack.

• Investigate how different settings affect (degrade or improve) the capability of an attacked host's operating-system to defend against TCP SYN flooding attack.

• Investigate the degradation of the services offered by the victim host during TCP SYN flooding attack.

• Investigate the algorithm that prevents such attacks.

1.4 OBJECTIVE

Distributed denial-of-service attacks on public servers have recently become a serious problem. The SYN flooding attack is frequent network based Denial of Service attack. This attack exploits the vulnerability of TCP connection known as 3 way handshaking. The SYN flooding attack sends too TCP SYN request to handle by the server. This action cause victim system responds slowly. Maximum attacks are launched through TCP and exploit the resources and bandwidth of the machine. Hence, it has become important to detect Dos Attacks and defend our system against such attacks.

1.5 METHODOLOGY

- First of all, we studied about the topic and learnt its basics.
- Then, we read various research papers and inferred various algorithms and started studying about those algorithms.
- We have to work for detection and prevention of the TCP SYN flood attacks.
- For detection, we will use ns2 simulator for attacking on two servers and gathering certain information.
- We will calculate throughput. Jitter. Delay and other related parameters to detect whether the attack has taken place or not.

- We will use xgraphs to measure the performance of the server.
- Using this approach, we will detect the TCP SYN flood attack.
- For prevention, we used IPsec and Ant Colony Algorithm.
- In IPsec, we will introduce 2 mobile nodes to deviate the path on which attack already took place.
- Then we will compare the 2 scenarios using data transmission rate and packet loss with xgraphs.
- Another technique we used is , Ant Colony Algorithm.
- This technique provides solution for routing for nodes which are static and nodes under movement.
- Then, again the performance is measured and defence mechanism is done.
- In this way, we will infer about the TCP SYN flood attacks- its detection and prevention.

1.6 ORGANIZATION

The rest of the report is organized as follows:

- Chapter 2 describes about the background and related work of the project. In this chapter we will focus on the work already done on detection and defense against TCP SYN Flood attacks. We will study different papers published and then try to implement any algorithm in our project which will help in detecting and defending the attacks.
- Chapter 3 describes the basics of ns2. Its commands and NAM. It tells the basic steps to create a simple network on the simulator and also discusses the performance metrics for the packet transfer over the network.
- Chapter 4 describes the results obtained by experimenting and calculating the values before and after the TCP SYN flood attack by which we can detect if the system is affected by TCP SYN Flood or not. Then We will apply the defence mechanism using IPsec algorithm and ant colony optimization. We will then analyze the results by comparing the two scenarios- before and after the mechanism is applied.
- Chapter 5 contains the conclusion and future work.

CHAPTER 2
LITERATURE SURVEY

Distributed denial-of-service attacks on public servers have recently become a serious problem. To assure that network services will not be interrupted and more effective defense mechanisms to protect against malicious traffic, especially SYN floods.

On the Internet, a distributed denial-of-service (DDoS) attack is one in which a multitude of compromised systems attack a single target, thereby causing denial of service for users of the targeted system. The flood of incoming messages to the target system essentially forces it to shut down, thereby denying service to the system to legitimate users. An attempt to make a computer resource unavailable to its intended users [1].

In DoS attacks the adversary mainly targets a few services like network bandwidth router or server CPU cycles system storage, operating system data structures, protocol data structures and software vulnerabilities. DoS can be a single source attack, originating at a single host, or can be a multi-source attack, where multiple hosts and networks are involved [2].

2.1 TYPES OF ATTACKS

There are 3 types of DDoS Attacks :

- **Bandwidth attacks**

Bandwidth attacks are relatively straightforward attempts to consume resources, such as network bandwidth or equipment throughput. High-data volume attacks can consume all available bandwidth between an ISP and site. The link fills up, and legitimate traffic slows down.

- **Logic Attacks**

Logic attacks exploit vulnerabilities in network software, such as a web server, or the underlying TCP/IP stack. Some vulnerability by crafting even a single malformed packet. Peer-to-peer attacks have found a way to exploit a number of bugs in peer-to-peer servers to initiate DDoS attacks. Application level floods are Various DoS-causing exploits such as buffer overflow can cause server-running software to get confused and fill the disk space or consume all available memory or CPU time.

- **Protocol Attacks**

The basic flood attack can be further refined to take advantage of the inherent design of common network protocols. These attacks do not directly exploit weaknesses in TCP/IP stacks or network applications but, instead, use the expected behavior of protocols such as TCP, UDP, and ICMP to the attacker's advantage. Examples of protocol attacks is SYN flood [1].

The basis of the SYN flooding attack lies in the design of the 3-way handshake that begins a TCP connection. In this handshake, the third packet verifies the initiator's ability to receive packets at the IP address it used as the source in its initial request, or its return reachability.

Depleting the backlog is the goal of the TCP SYN flooding attack, which attempts to send enough SYN segments to fill the entire backlog. The attacker uses source IP addresses in the SYNs that are not likely to trigger any response that would free the TCBs from the SYN-RECEIVED state. Because TCP attempts to be reliable, the target host keeps its TCBs stuck in SYN-RECEIVED for a relatively long time before giving up on the half connection and reaping them. In the meantime, service is denied to the application process on the listener for legitimate new TCP connection initiation requests [4].

2.2 DETECTION TECHNIQUES:

DoS attacks can be detected :

1) By using traffic signatures
2) By recognizing anomalies in system behaviours.

A signature-based approach uses the signatures of the well-known attacks to determine if the packet represents a suspicious activity. Anomaly-based approach will detect abnormal behaviours by monitoring network traffic and comparing it with the baseline behaviours. The baseline will identify what is “normal” for that network. The baseline activity could be identified by a combination of average packet size, number of packets per second, flows per second, and bytes per second. Then the system can trigger an alert when it finds a significant deviation from the baseline.

- **Detection using TCP flags**

Except for the initial SYN packet, every packet in a connection must have the ACK bit set. FIN ACK and ACK are used during the graceful teardown of an existing connection. PSH FIN ACK may also be seen at the beginning of a graceful teardown. RST or RST ACK can be used to immediately terminate an existing connection.

- **Detecting by using port**

There are several other characteristics of TCP traffic where abnormalities may be occurred to attackers Packets should never have a source or destination port set to 0. The acknowledgment number should never be set to 0 when the ACK flag is set. A SYN only packet, which should only occur when a new connection is being initiated, should not contain any data .

- **Detecting by tracing the route**

Another solution is we can trace the route of the corresponding message where it is started from. The ICMP message can be fixed the last router to send original IP address of the router. Now the server can check the source IP address and original IP address whether it is true or not .So the server identifies the SYN flooding attack. In the existing system, there is no time computation mechanism for defense but it is implemented in the proposed system. This defense mechanism is used identify the SYN flooding accurately with short time span and less SYN packets.

- **Firewall as Semi transparent Gateway**

The firewall monitors the traffic sent from source to destination. When it sees ACK+SYN being sent from D to S, it responds by creating an ACK message and sending to D, thus reallocating the resources and moving the connection out of the queue. If it is an attack, the firewall then sends a RST message to D and connection is dropped. If the connection request is from a proper source, he sends back ACK, which is passed by firewall. D merely sees it as the duplicated packet and discards it.

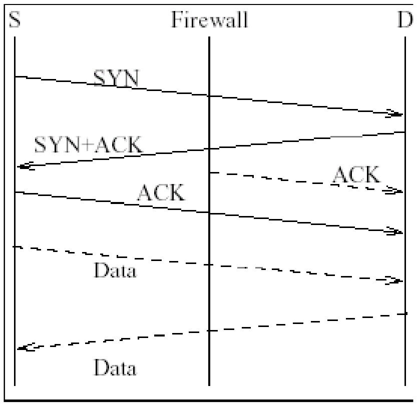


Fig. 2.1 Firewall as a semi-transparent gateway (Legitimate connection)

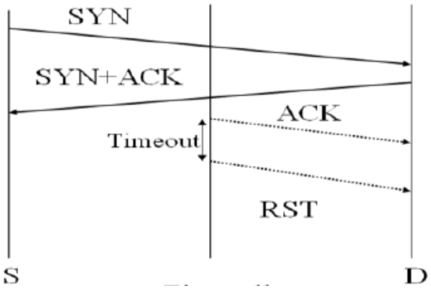


Fig. 2.2 Firewall as a semi-transparent gateway (Illegitimate connection)

- **IP Traceback**

The problem of identifying the machine that directly generates attack traffic is called IP trace back problem. IP trace back is a subtle scheme to tackle DoS attacks. If it can provide the exact attack origin, then we may apply some proper actions such as packet filtering to stop attacks completely. The mechanism must be incrementally deployable which means, it should function even when not all of the routers across the Internet use this mechanism. A trace back mechanism should not require major changes on the current infrastructure. The number of packets required to identify the attack path should be as low as possible. Also, a trace back mechanism should scale to a large number of attackers while maintaining accuracy .

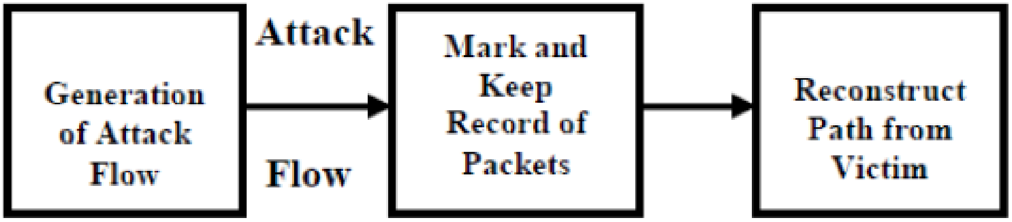

Generation of Attack Flow
Attack
Flow
Mark and Keep Record of Packets
Reconstruct Path from Victim

Fig. 2.3 Block Schematic of Proposed approach for trace back

2.3 DEFENCE TECHNIQUES:

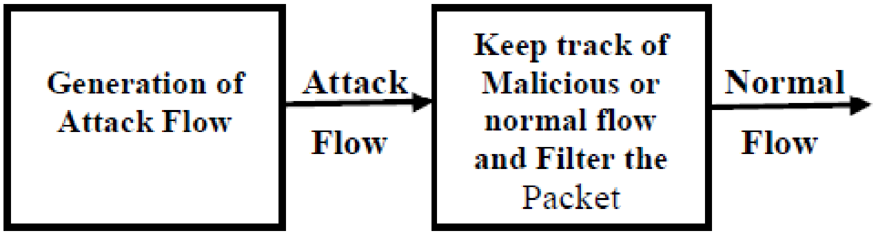

Generation of Attack Flow
Attack
Flow
Keep track of Malicious or normal flow and Filter the Packet
Normal
Flow

Fig 2.4 Block schematic of proposed approach for Defense.

- **Ingress / Egress Filtering**

Ingress Filtering is one of the source-end defense mechanisms to block spoofed packets before entering the Internet core. The purpose of ingress/egress filtering is to only allow traffic to enter or leave the network if its source addresses are within the expected IP address range. Ingress filtering is a filtering scheme that filters incoming traffic according to a specified rule. The purpose of ingress/egress filtering is to only allow traffic to enter or leave the network if its source addresses are within the expected IP address range. Suppose an attacker X resides within the leaf network .

- **SYN Cache**

In the SYN cache mechanism, the server node has a global hash table to keep half-open states of all applications, while in the original TCP these are stored in the backlog queue provided for each application. As a result, the node can have a larger number of half-open states and the impact of a SYN flood attack can be reduced [1].

- **Detecting by using port**

There are several other characteristics of TCP traffic where abnormalities may be occurred to attackers. Packets should never have a source or destination port set to 0. The acknowledgment number should never be set to 0 when the ACK flag is set. A SYN only packet, which should only occur when a new connection is being initiated, should not contain any data [1].

- **Detecting by ICMP feedback**

The hacker may send SYN packet to the server using some other IP address like 10.1.5.20. Then the server receives SYN packet and acknowledges to the corresponding IP address. But the client does not understand the SYN+ACK packet. And also, if the client is in off, the server will not receive the ACK packet and then waiting for the reply.

The implementation takes place when the server replies SYN+ACK, the ICMP messages also added with that and it is sent to the client. This approach is used to get the information like whether the client receives from SYN+ACK or not. Because, the packet is sent from the server to the client via router, hub or any other active device. The server then identifies the reply from the client and stops sending the message to the client and concluded that it is work of a hacker so the server identifies the SYN flooding attack [1].

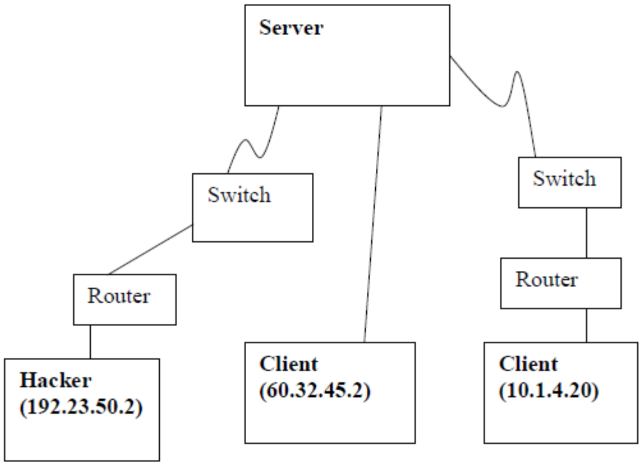


Fig. 2.6 Detecting by ICMP Feedback

2.4 DARB-DELAY PROBING METHOD

The delay between the server and the client is estimated using a delay probing method.DARB. By setting different time-to-live(TTL) values at the IP headers, the packets will die at different routers. Information about the death of packets is sent by the relevant router and this provide information for estimating the delay along the path. A score is then given to the delay value which allows an evaluation of the probability of the half-open connection being the result of a SYN fooding attack.

Information about the death of packets is sent by the relevant router and this provide information for estimating the delay along the path. A score is then given to the delay value which allows an evaluation of the probability of the half-open connection being the result of a SYN flooding attack.

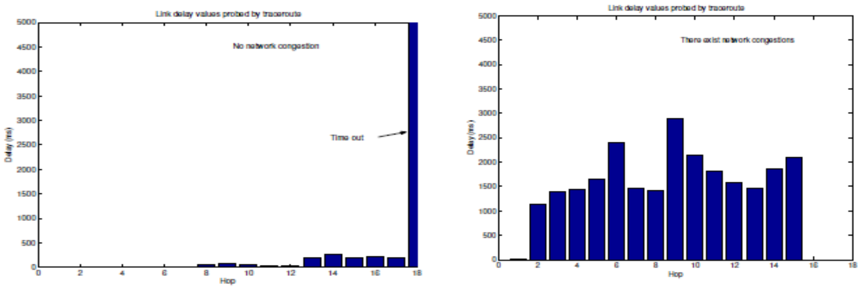
Method:

The center problem is to distinguish the abnormal half open state from the normal half-open state. A basic distinction is that most of normal half-open connections arise from network congestion whereas abnormal half-open connections have nothing to do with congestion. i.e. the delay between the network routers is probably as same as the delay under network normal status. If the half-open connection is caused by congestion, the routing path between the server and the client should show some features of congestion, for example, increased packet delay, a rising packet loss rate, and a near-capacity queue at the congested router. In our method, the path delay between the server and the client is probed by a method similar to trace route. A very long delay is regarded as a congestion feature and in this circumstance a half-open connection is considered normal. Otherwise, the half-open will be dealt with as abnormal.

Even if there exists a burst of flooding traffic at the beginning of attack, most of victims can survive for the first several minutes and serious network congestion appears gradually instead of appearing suddenly [5].

Experiment:

To verify whether the network delay is suitable to be used as the sign of traffic congestion, we probed about 100 web site from Hong Kong and Wuhan respectively hop-by-hop to obtain the delay of network. We found the average delay for a healthy network is about 100ms. Figure 2.12(a) shows the average hop-by-hop delay values of a healthy network. The delay values of first several hops are always below 20ms, which seems to be zero in the figure, and delay values of middle hops is about 100-200ms. If the probing result is time-out, we use an infinite value to substitute delay value. Time out results are always found at the last hops of delay probing because the probing packets may be fltered out by the firewall at the server side. Figure 2.12(b) depicts the delay value of a congested network, which is infected by the worm virus. The clients were informed by ISP that the service might degrade because they were being attacked by some kind of worm virus. The worm virus affected the ISP at the Hop 2 and the average delay after the Hop 2 is far above 1000ms. From such experiments, we believe the delay value is suitable to distinguish a healthy network from a congested one [5].



(a) A healthy network (b) A congested network

Fig 2.7 Shows variation between healthy and congested network.

2.5 PREVENTION OF SYN FLOOD DOS ATTACK (Using Captcha Method)

The Transmission Control Block (TCB) is a transport protocol data structure (actually a set of structures in many operations systems) that holds all the information about a connection. The memory footprint of a single TCB depends on what TCP options and other features an implementation provides and has enabled for a connection. Usually, each TCB exceeds at least 280 bytes, and in some operating systems currently takes more than 1300 bytes. The TCP SYN-RECEIVED state is used to indicate that the connection is only half open, and that the legitimacy of the request is still in question. The important aspect to note is that the TCB is allocated based on reception of the SYN packet— before the connection is fully established or the initiator's return reachability has been verified.

2.5.1 Overview

All compromised machines are automated attacks, therefore using a method that checks whether a request originated from a real human being should help to deny the attack. The system will determine whether an IP address has requested a page too many times in a given limit.

2.5.2 Algorithm For Captcha System

Step 1.Visitors sends GET request to server

Step2.Is IP allowed in the list?

Step3.If Yes

Step4. Then Process request and send website content back

Step5.Else Read client List

Step6.Has this IP sent GET request more than X times in the Last Y seconds?

Step7.Then GET recaptcha bax and request visitor confirm they are real person.

Step8.Does user get captcha input correct?

Step9.Then add user to allow list and purge other lists.

Step10.Process request and send website content back.

Step11.Else Instantiate Captcha list for IP by 1.

Step12.Check whether this IP attempted Captcha more than X times.

Step13.Then add IP address to IP Tables Deny List.

Step14.Else Get Recaptcha box and requst visitor confirm they are real person.

Step15.Else process request and send website content back.

Step16.Instantiate Client list by 1.

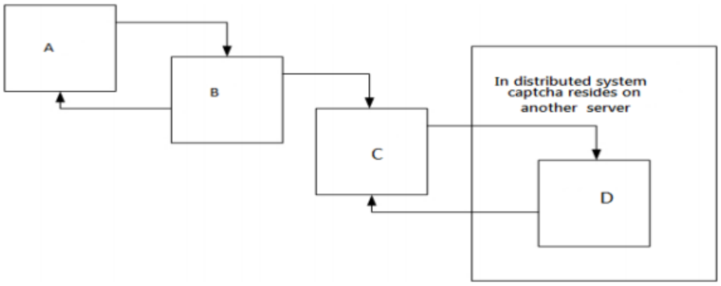


Fig 2.8 Overview of the prototype

2.5.3Apache Module

The Apache Web Server will be used to serve web page requests for the prototype. The server allows for custom modules to be included, and a basic module for detecting a possible DDoS attack and then mitigating it will be developed. The module will detect the number of times a user has requested the web page in a given time. The apache web server runs with the module compiled. Whenever a user visits a website hosted on the Apache web server, the Mitigate.c (once compiled, becomes mitigate. so) module is called .The module should have the ability to define an “allowed. Hosts” file for known IPs we do not want to “challenge” with the captcha.php module.

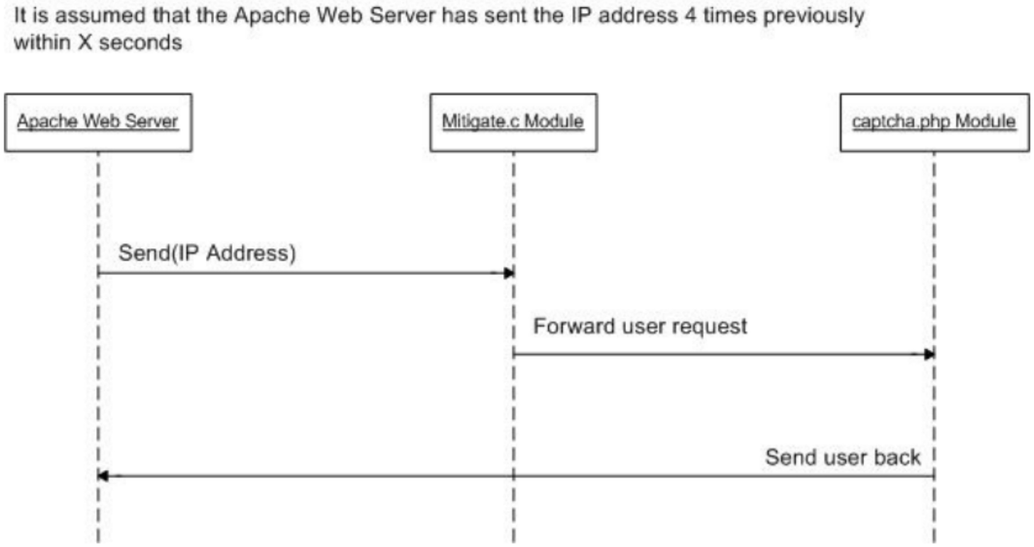


Fig 2.9 Sequence diagram for Apache module

2.5.4 PHP Module

Once the request has been handed from the mitigate.c module to the PHP module, it is the PHP modules responsibility to determine whether or not the request was made by a human or an automated robot. The PHP module makes use of the ReCAPTCHA Project, a free, hosted CAPTCHA service which the owners claim process over 30 million CAPTCHA requests a day. The PHP module should create a variable which holds the number of attempts the user has taken. It should also have a variable which holds the maximum number of attempts the user is allowed. If the user exceeds the threshold of this limit, the IP address they are using will be denied using iptables. If the request is processed correctly by the user, the Captcha.php module should send back this result to the mitigate.c Apache module so that it may add the IP address to the allowed list of hosts. As a result of this inclusion the user will never be required to verify themselves again whilst using that web server.

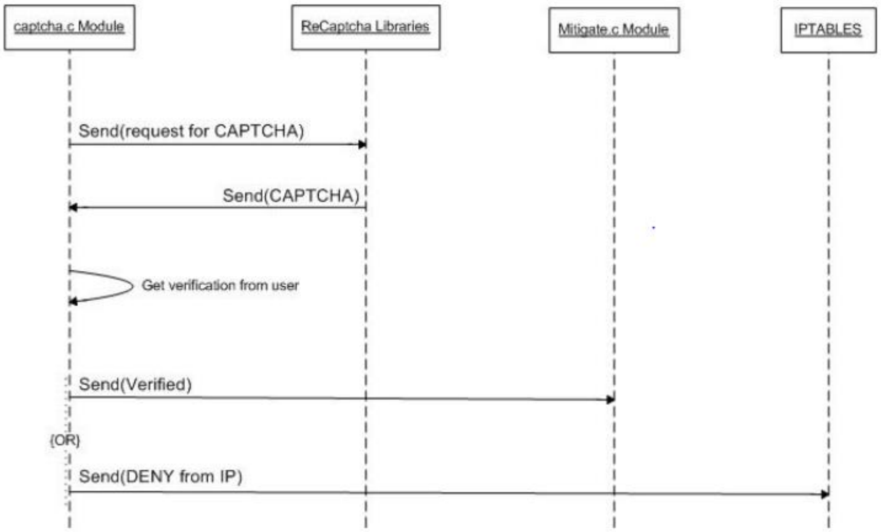


Fig 2.10 UML diagram for captcha module

To summarize, when the request is received by the server, the server should do a check to see if the associated DNS entry is in an “allowed. Hostnames” file. This way, all known search engines that comply with reverse DNS, will always be allowed through the system without being stopped by the CAPTCHA system. A downside to this is that for every request a lot of system resources are used; reading the IP address, checking the DNS, reading the “allowed. hostnames” file, etc[8].

2.6 SIMULATION STUDY OF FLOOD ATTACKING OF DDOS

In order to quantitatively study flood attacking, we have to define a measure to quantitatively describe the main performance of flood attacking. Recall the the victim usually may not be immediately overwhelmed by flood attacking but some time late after attacking is launched. Thus, we define Ta as attack time. The attack time means a time duration. The starting of the duration is the time when attacking occurs and the end of the duration is the time when the victim starts to be overwhelmed or its service performance begins to degrade significantly.

Behaviors of flood attacking under different protocols are discussed based on five experiments.

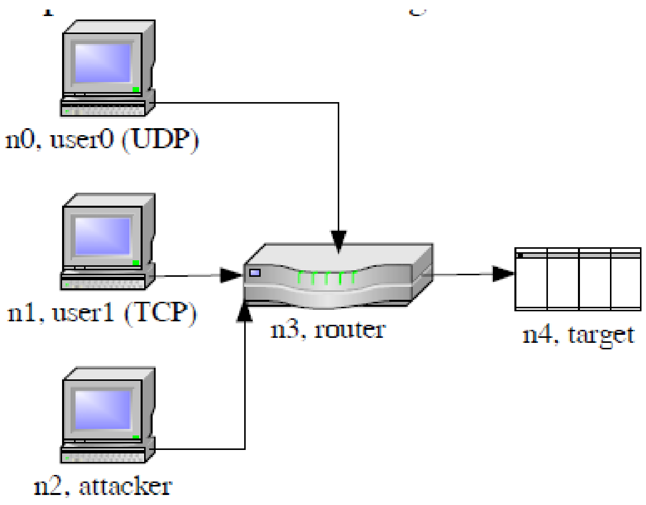


Fig. 2.11 Simplified attacking structure for simulation.

- **Expt 1: Attack-free traffic**

With the attack model in Fig., n1 sends 45% TCP data flow to n4, respectively. In this experiment, n2 sends none. Hence, attack free. Since the total data receiving at n3 only takes 65% of the total bandwidth of the link from n3 to n4, there is no traffic congestion either.

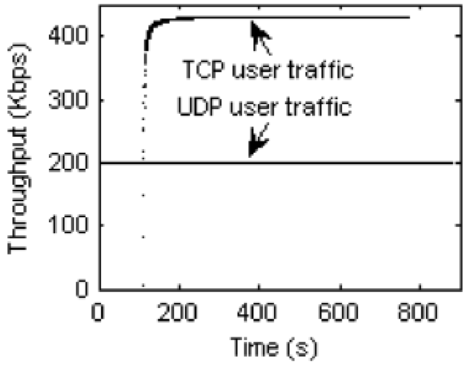


Fig. 2.12 Attack-free traffic for UDP and TCP

- **Expt 2: TCP and UDP traffic of clients under TCP-type attacking**

n0 sends 20% UDP traffic to n4, n1 sends 45% TCP traffic to n4, and n2 sends 55% TCP attack traffic to n4, concurrently. In this case, the traffic from n3 to n4 exceeds 20% of the link bandwidth. Consequently, some packets from the legitimate clients may be dropped at n3. In this case, we say that n2 produces the attack intensity of 20%. More precisely, it produces the attack intensity of 20% of the TCP type.

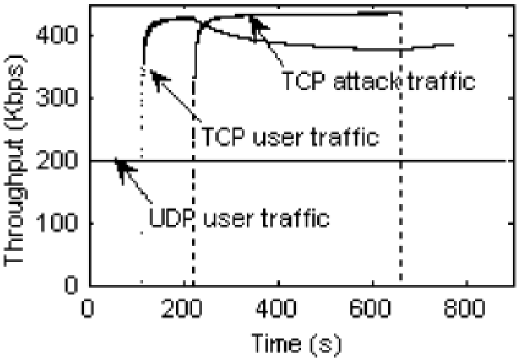


Fig. 2.13 UDP and TCP traffic under TCP-type attacking

- **Expt 3: TCP and UDP traffic of clients under UDP-type attacking**

With the attack model of Fig. 2.11, n0 sends 20% UDP traffic to n4, n1 sends 45% TCP traffic to n4, and n2 sends 55% UDP attack traffic to n4, concurrently. Since the total traffic receiving at n3 is 20% over the link bandwidth between n3 and n4, some packets from the legitimate clients may be dropped at n3. In this case, we say that n2 makes the attack intensity of 20% of the UDP.

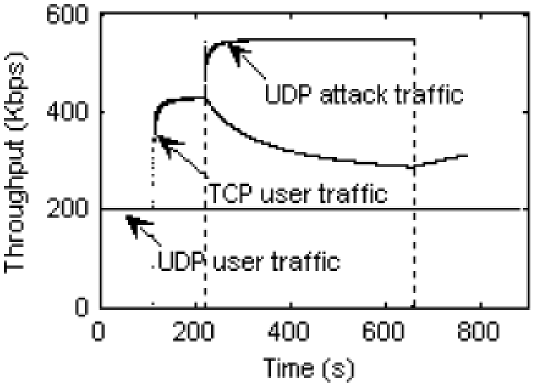


Fig. 2.14 UDP and TCP traffic under UDP-type attacking

- **Expt 4: Attacking behaviors under different attack intensity**

In this experiment, we investigate the attacking under different attack intensity. With the attack model of Fig. 2.10. n0 sends 20% TCP traffic to n4, n1 sends 80% UDP traffic to n4. In this case, n2 sends UDP attack traffic to n4 with the attack intensity of 10%, 20%, and 50%, respectively. Fig. 2.15 shows the attacking effects on the TCP client.

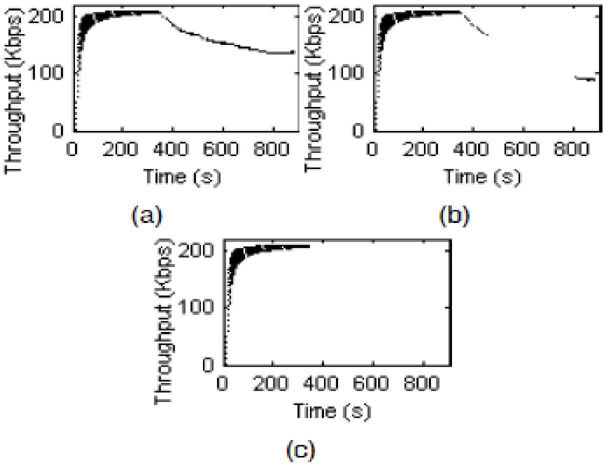


Fig. 2.15 Attack effects on TCP client under UDP type attacking under different attack intensities. (a). 10% attack intensity. (b) 20% attack intensity. (c) 50% attack intensity.

Remarks on these experiments:

The experiments 1 ~ 3 suggest the following two important points. 20% attack intensity of TCP type does not affect UDP clients but it reduces the 10% TCP traffic of the legitimate clients, see Fig. 2.9. 20% attack intensity of UDP type does not affect UDP clients either but it reduces the 20% TCP traffic of the legitimate clients as can be seen from Fig. 2.10.

From the above, we experimentally infer that TCP is connection-oriented. Its transmission rate is controlled by the receiving end. If the transmission rate exceeds the receiving ability at the receiving end, the transmission node will be informed to reduce the transmission rate. Nevertheless, UDP is connectionless-oriented. Its transmission rate is not controlled by the receiving end. Thus, TCP clients are generally easier to be attacked by the bandwidth attacking than UDP clients. From the point of view of attackers, TCP-type attacking is usually used to exhaust the resources on the target side. In this aspect, SYN flood is a typical way [3].

2.7 Vulnerability's of IPSEC: A discussion of possible weaknesses in IPSEC implementation and protocols[9]

IPSEC is not a product in itself, but simply a set of protocols developed by the Internet Engineering Task Force (IETF) as a series of Request for Comment's (RFC's). The RFC's that make up the IPSEC protocol cover the requirements of an implementation of the IPSEC protocols. When dealing with cryptography and security, there are a number of principles that will appear consistently, these principles are:-

- **Confidentiality:** Data cannot be read by anyone other than the person or destination that it was intended for.
- **Integrity:** The data intended for a destination must appear at its destination without being altered.
- **Authentication:** There must be a way for the destination of the data to verify that the source of the data is legitimate. This can also include the requirement to ensure that the source and destination cannot deny that the transaction took place (non-repudiation).

2.7.1 IPSEC Operation

The purpose of IPSEC is to provide various security services to traffic travelling between a source and destination, the destination/source may be a router, or a host. The services may be applied to all packets, or only to specific types of traffic, egtelnet or ftp.

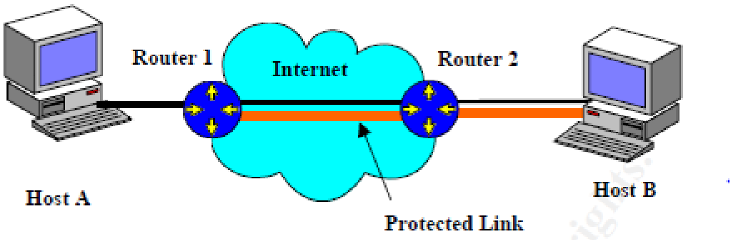


Fig. 2.16 Protection provided by IPSEC between two hosts

IPSEC has two main modes of operation, Tunnel and Transport Mode. IPSEC modifies packets in different ways depending on the Mode in which it is operating. The diagram below shows the packet modifications for the two different modes of operation.

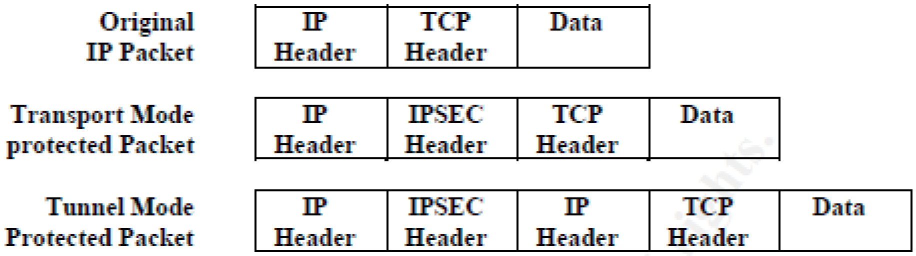


Fig. 2.17 Showing IPSEC packet modification(Doraswamy &Harkins,1999 p.44)

The IPSEC protocols define operation in two different ways, Authentication Header (AH) and Encapsulated Security Payload (ESP). AH and ESP are quite different in the security that they provide, and operate in different ways.

2.7.2 Encapsulating Security Payload (ESP)

The ESP protocol provides data integrity, authentication and confidentiality. The data to be transmitted is encrypted and the entire packet, minus the IP Header and the authentication data is authenticated. The diagram below shows a simplified version of how ESP operates.

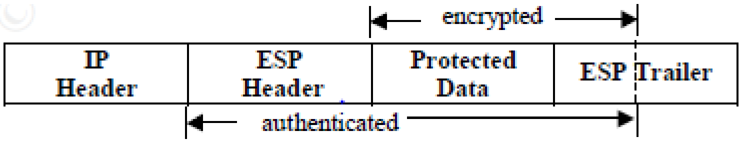


Fig. 2.18 Encapsulated Security Payload

2.7.3 Authentication Header (AH)

The Authentication Header is a much simpler protocol when compared with ESP, however it provides much less in terms of packet security. The AH header only authenticates the data and IP Header. This means that there is no confidentiality provided by AH, it only provides integrity and data source authentication. AH is useful when the only concern is protection against modification of the data while in transit. Clearly the processing overhead for AH will be significantly less than that for ESP. Encryption and decryption take much more processing than hashing.

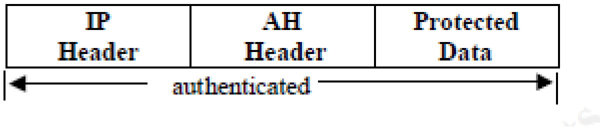


Fig. 2.19 Authentication Header

2.7.4 Security Associations

A Security Association (SA) is the term used in IPSEC to describe a negotiated set of parameters between communicating peers. With the wide range of encryption and hashing algorithms available, it is vital that both peers have a way of defining the particular algorithms that will be applied and how they will be applied to each SA. There is also a requirement for clear guidelines to define how the peers initially negotiate these parameters to set up an SA.

2.7.5 Summary & Conclusions

IPSEC is an excellent set of protocols, developed out of significant work and collaboration from within the networking security community. IPSEC can be viewed as another piece of the security arsenal, and if used correctly in conjunction with numerous other technologies it could ensure a much greater degree of security on computer networks.

One of the most important lessons that was gained from the study of IPSEC, is the requirement for customers to understand the implementation of their specific vendor or OS provider. Some vendors produce products that are not fully compliant with the IPSEC protocols, and could be seriously flawed in the security that they provide.

2.8 Intrusion Detection System based on Ant Colony System[10]

i. Intrusion Detection System

The present intrusion detection system research not only considers the accuracy judged by the network behavior, but also its extensibility in the face of the mass network flow data. Figure1 designs the framework [2] for an intrusion detection system. The framework is divided into the above several basic modules and an Audit Database, Monitored Entity module is in charge of determining the necessary monitoring network behavior; Audit Collection module is in charge of collecting the data from the host or network, its data is used for Analysis and Detection module; Audit DB is in charge of storing the mass collected data. Analysis and Detection is the core of the system. The intrusion detection system judges whether the network flow is normal or improper behavior in the module with users’ designed algorithm. The part of the research, as previously stated, can be divided into improper detection system, anomaly detection system and hybrid system. Configuration Data module is in charge of setting up the corresponding strategies for the normal or improper flow and other related setup program.

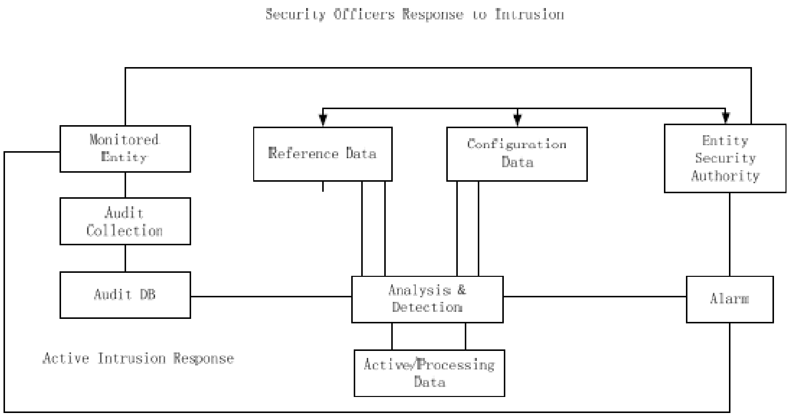
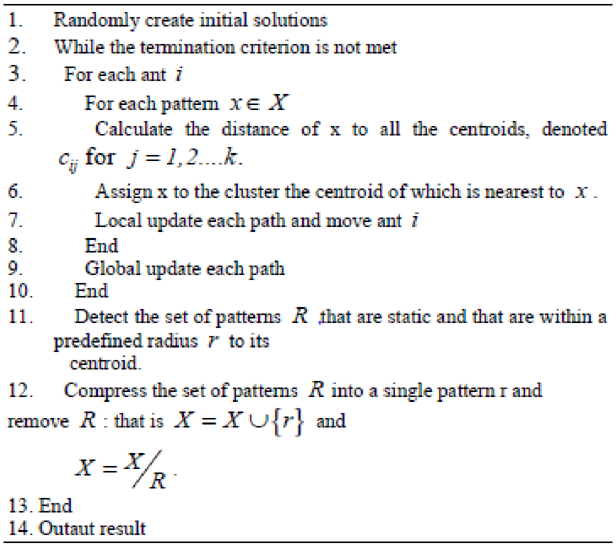


Fig. 2.20 Framework drawing of the intrusion detection system

ii. Ant Colony Optimization applied in the Intrusion Detection System

The performance of the traditional regulation judgment or the discrimination method based on the statistics is insufficient in the face of the mass network flow data. It solves these problems with the use of heuristic algorithm [4]. Many achievements have proved its feasibility and effectiveness. Many researches attempt to use it as the judging algorithm of the intrusion detection system for

the effect of the heuristic algorithm based on the ant behavior is remarkable. These researches can be divided into two kinds: ant-based clustering algorithm (ACA) [16-17] introduced by Deneubourg and ant colony optimization (ACO) [18-19] introduced by Dorigo.



```
Randomly create initial solutions
While the termination criterion is not met
  For each ant $i$
    For each pattern $x \in X$
      Calculate the distance of x to all the centroids, denoted $c_{ij}$ for $j = 1, 2 \ldots k$.
      Assign x to the cluster the centroid of which is nearest to $x$ .
      Local update each path and move ant $i$
    End
    Global update each path
  End
  Detect the set of patterns $R$ ,that are static and that are within a predefined radius $r$ to its centroid.
  Compress the set of patterns $R$ into a single pattern r and remove $R$ : that is $X = X \cup \{r\}$ and
    $X = X/R$ .
End
Outaut result
```

Fig. 2.21 Fast ant colony algorithm

iii. Network Environment

The network topology framework in the research is as the figure 3 shown, the regional network and the internetwork connect the router or network switch. All input or output network flow will pass through the network equipment (router or network switch). Take the figure as the example, the network switch remains a copy of the input and output network flow to IDS and it offers analysis flow packet information and maintains the network quality. It can collect the necessary testing data flow, analyze the later designed packet and conduct the test and analysis of the packet judging the modules through the method. It can further make the judgment module have more high accuracy with the effective training.

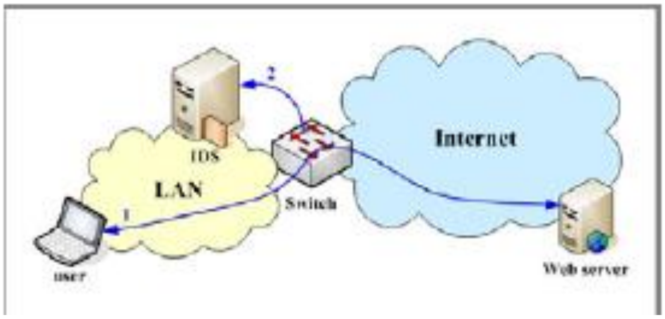


Fig. 2.22 Network Topology framework

iv. CONCLUSION

The paper speeds up the analysis and the judgment of the modules in the intrusion detection system by the use of the designed fast clustering algorithm. The intrusion detection system can analyze and manage the large network flow behavior by reducing the computation time of the modules. In the practical application, it firstly establishes a prototype of the intrusion detection system in order to test and analyze the related data. In the theoretical design, it makes several algorithms based on the heuristic computation, and then applies them in the analysis and the judgment of the modules in the intrusion detection system. As to the research on improving the intrusion detection system's performance, we design a pattern reduction algorithm on the basis of the ant optimization which can reduce the repetitive and unnecessary computations. The experimentally simulation results show that the proposed method in the research can effectively reduce the original computation time for analyzing and judging modules on the basis of the ant optimization without losing overdue correct rate. In the future research, we will regard the algorithm developed in the research and the ant optimization applied in the intrusion detection system as the foundation. It further designs and modifies the current algorithm in order to effectively detect the repetitive computation and reduce large computation time.

CHAPTER 3

SYSTEM DEVELOPMENT

3.1 TOOL USED FOR SIMULATION:

Simulation can be defined to show the eventual real behaviour of the selected system model. It is used for performance optimization on the basis of creating a model of the system in order to gain insight into their functioning. We can predict the estimation and assumption of the real system by using simulation results.

Some of the most popular simulators used to simulate the various networks are OMNET++, NS2 and OPNET. NS2 simulator was used in this study.

NS2 Simulator

NS2 is an object oriented and discrete event simulator targeted at networking research. It is written in C++/OTcl/Tcl. NS2 is mainly used for local and wide area networks. It is most easily accessible network simulator available.

- **Starting ns**

Start ns with the command 'ns <tclscript>', where '<tclscript>' is the name of a Tcl script file which defines the simulation scenario (i.e. the topology and the events). You could also just start ns without any arguments and enter the Tcl commands in the Tcl shell, but that is definitely less comfortable.

- **Starting nam**

Ns generates a nam trace file that is used to show the animation of the nodes.

On the next page we can see a screenshot of a nam window where the most important functions are being explained.

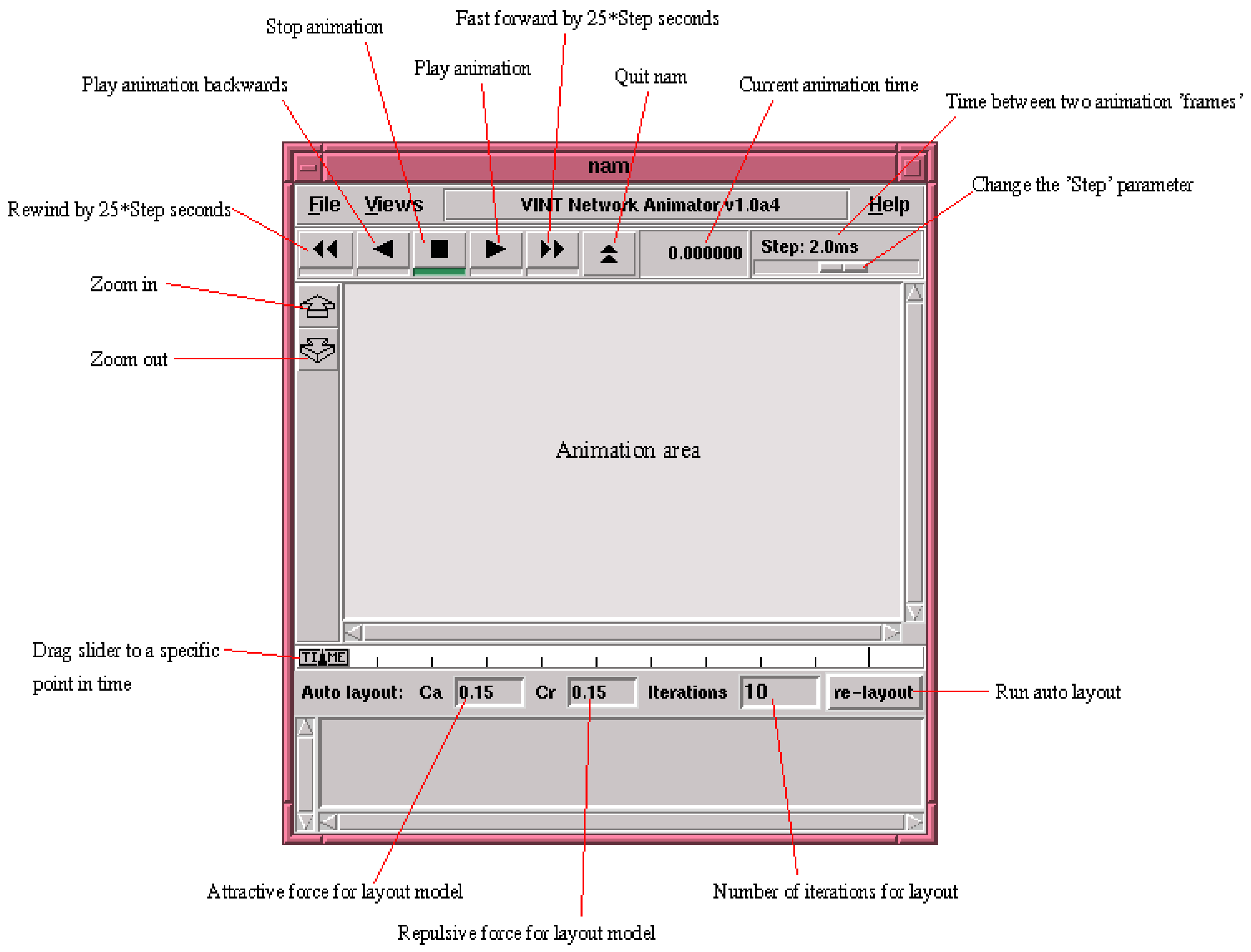

Stop animation
Fast forward by 25*Step seconds
Play animation
Play animation backwards
Quit nam
Current animation time
Time between two animation 'frames'
Change the 'Step' parameter
Rewind by 25*Step seconds
Zoom in
Zoom out
nam
File
Views
VINT Network Animator v1.0a4
Help
0.000000
Step: 2.0ms
Animation area
TIME
Drag slider to a specific point in time
Auto layout:
Ca
0.15
Cr
0.15
Iterations
10
re-layout
Run auto layout
Attractive force for layout model
Repulsive force for layout model
Number of iterations for layout

Fig. 3.1 nam window with all functions defined

3.2 APPROACH

3.2.1 Detection Mechanism

- Generate TCP-SYN attack
- Identify the various metrics for evaluating various services against TCP-SYN attacks.
- Compare impact of TCP-SYN attacks on various services.

In first phass

- 2 Attacker nodes
- 5 or 7 legitimate users
- 1 victim or server

- The topology will look like:

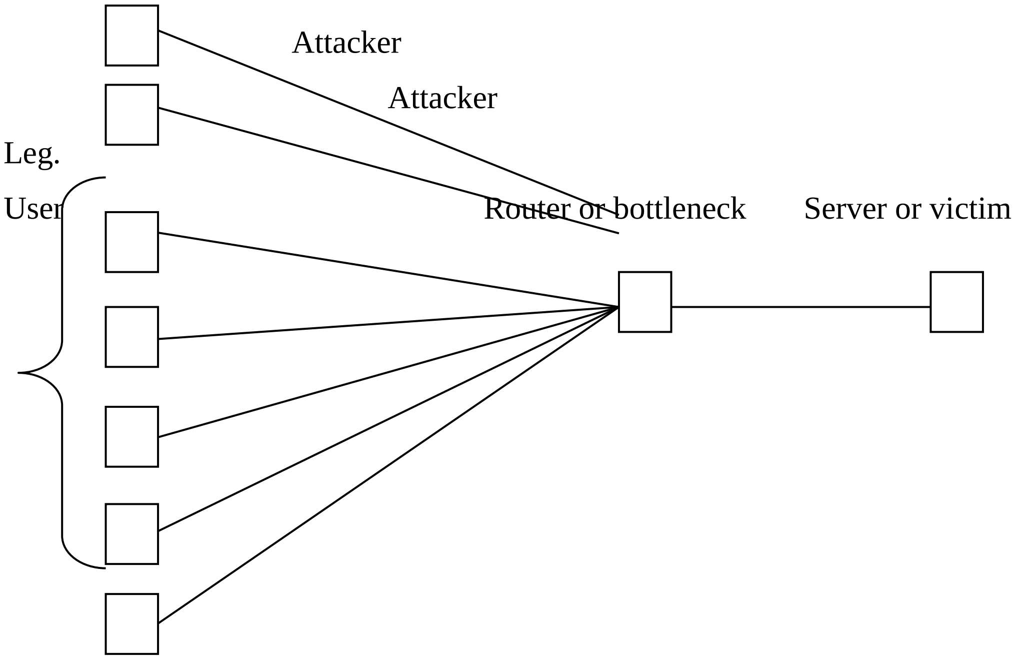


Fig 3.2 Topology for basic network

In this scenario 2 Attackers will create TCP-SYN attack and will consume the bandwidth of router or bottleneck. So the attack is bandwidth depletion attack.

After that following is done:

- Metrics calculated - packet loss, throughput, jitter, delay and response time.
- Legitimate users are using quality of services through victim or server after TCP-SYN attack the quality of service is changed means there must be variations. These variations again calculated so that the difference can be calculate.
- These metrics firstly calculated without attack on the same scenario and again calculated after attack.

Both the variations are shown through xgraphs.

3.2.2 Defence Mechanism

3.2.2.1 IPsec Algorithm[9]

Existing System:

In the existing approach nodes are static without IPsec so there is possibility for attack

Node 18 represent the base station responsible for providing connection to the underlying nodes.

S1,S2,S3,S4 represent the sentinel nodes manages some set of nodes under it for providing connection. Sentinels are interconnected with each other and provide services to the underlying nodes. Node D,R,T,N acts as the attacker blocks the connection as the bridge. so the sentinels S1 and S4 loses the connection this leads to the packet drop. When the attacker blocks the connectivity between sentinels, the mobile nodes act as the connector for the sentinels S1 and S3. So the connection is retained. Packet drop due to attacker is reduced largely.

Proposed System:

In the proposed methodology ,Mobile nodes are introduced with Ipsec configuration. In this technique two nodes are introduced mobile 1 and mobile 2. S1,S2,S3,S4 are the sentinels and node 18 acts as the base station. Sentinels are interconnected with each other and provide services to the underlying nodes as existing approach. Node D,R,T,N acts as the attacker blocks the connection as the bridge. When the attacker blocks the connectivity between sentinels, the mobile nodes act as the connector for the sentinels S1 and S3. So the connection is retained.

Packet drop due to attacker is reduced largely.

3.2.2.2 Slave-Ant Algorithm[10]

In this proposed algorithm, slave-ant technique is used to solve various abnormalities caused attacks in the network layer. This slave ant makes the protectorate activity performed by swarm intelligence, to make use of the resource that may not be provided for the single ant acting alone. The team work of the slave ant in building the task is clearly explained below.

Step 1 The slave ants move randomly in any direction but it is always make use the path created by organic substance. If there is no organic substance is present, then the ants move randomly in any direction.

Step 2 Each one of the slave ant carry one nugget at a time, if any ant is not carrying nugget then it will bump into some other ant to perform its task.

Step 3 The ant carrying a nugget come across another ant, then it puts the nuggets down to make another ant to carry it, the nugget is now infused with some organic substance.

Step 4 The slave ant technique preserves the most of the technique followed in the ant colony optimization. In the slave ant technique the forward route request are unicast but the route reply may not necessary to follow the forward path, it can also routed randomly.

Step 5 The slave ant technique follow the single route rather than changing route it helps in reducing the packet loss but it does not concentrates on route optimization.

Step 6 Optimized slave ant technique is proposed to provide the optimized route selection.

Organic table

The organic table follows the $M * N$ matrix where M is the destination containing organic value and N is the adjacent node containing organic value considered for the next hop during the packet transfer.

$O_{d,n}$ Provides a detail about quantity of organic adjacent node n has for the destination node d. The adjacent node has the enhanced organic, so that it has more prospect of selecting as the next bound during routing.

The organic table for the source node S is shown below

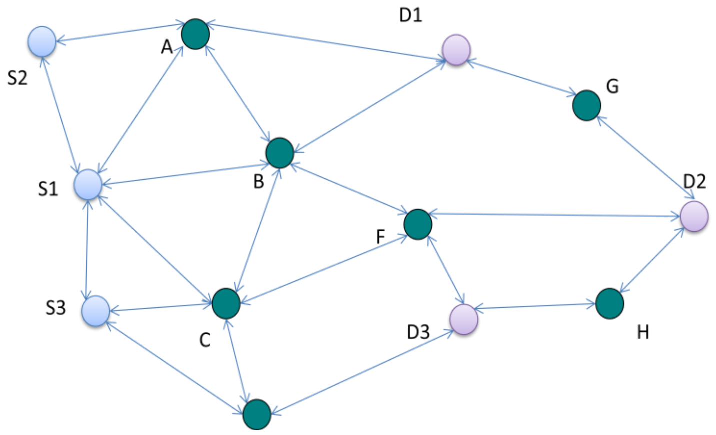


Fig 3.3 Distribute Network Topology

The sender node S1, S2, S3 has the intermediate nodes A, B, C, E, F, G, H and destination nodes D1, D2, D3 are the destination nodes participating in the topology.

Existing system:

In this existing system, they are mainly concentrated on transferring packets between the source and destination and also they are concentrated when the source and destination are under movement, nowadays the mobile devices plays major role in network topology.

Multicasting was clearly proved in the existing system by transferring packets from one source to many destinations on a single time.

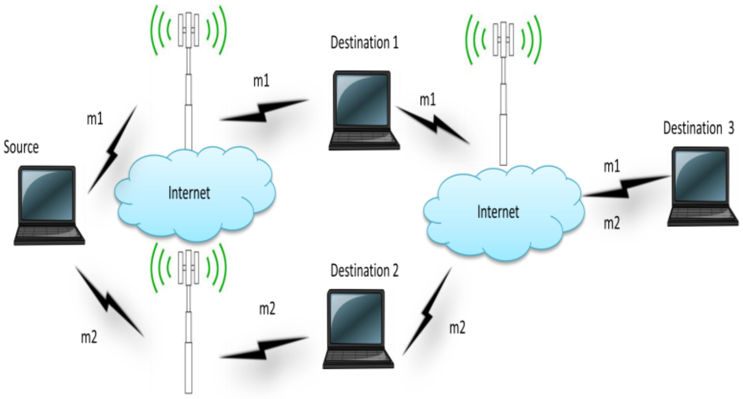


Fig 3.4 Muticasting in existing systems

Proposed System:

The proposed system is designed to overcome the various problems, arises during the multicast packet transfer to the mobile nodes.

The proposed system provides an optimized route to the various destinations using the ant algorithm as we discussed above, with the help of this path the packets are transferred to the various destinations.

Step 1 The nodes participating in the networks to access service like internet registers its identity with the server agent, the server agent replies with unique ID to the requesting node.

Step 2 The source node request route with the current access point to the destination node the current access point forwards the route request to the server agent.

Step 3 The server agent verifies the source ID, then it accepts the route request from sender then it gathers the information of receiver using destination ID from the list.

Step 4 The server agent then broadcasts the route request message using destination ID, the registered adjacent nodes that are nearer to the destination node which are ready to provide the service replies with the acknowledgement message to the server agent.

Step 5 The server agent chooses the adjacent node with the longest life time (the ability of the nodes to stay connected with the destination node) using the details collected from the ID, Such as nodes position, direction of motion and speed of the node.

Step 6 Then the server agent provides route reply message for the source node, after this authentication process, source node starts sending data packets in a secure way.

Step 7 In case any node moves away from the network, immediately the server agent replaces it with some other nodes to maintain the continuity of connection.

Step 8 In this technique, the malicious node or selfish nodes are completely eliminated from the network, as the server agent takes full control of the ad-hoc network.

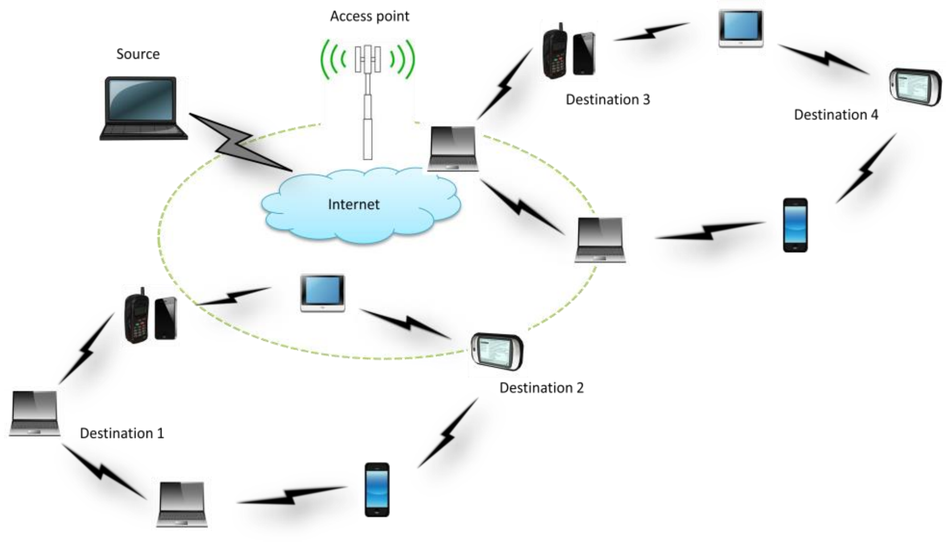


Fig 3.5 Muticasting in proposed systems

3.3 PERFORMANCE METRICES

In this section, we will discuss which parameters are used to measure the performance of the network. There are five key performance metrics such as packet loss rate, packet delivery ratio, jitter, end-to-end delay and throughput which are considerably affected by routing algorithm.

3.3.1 Packet Loss rate

Packet loss occurs when one or more packets of data travelling across a computer network fail to reach their destination. Packet loss is typically caused by network congestion. Packet loss is measured as a percentage of packets lost with respect to packets sent.

The Transmission Control Protocol (TCP) detects packet loss and performs retransmissions to ensure reliable messaging. Packet loss in a TCP connection is also used to avoid congestion and reduces throughput of the connection.

3.3.2 Jitter

Jitter is defined as a variation in the delay of received packets. At the sending sides, packets are sent in a continuous stream with the packets spaced evenly apart. Due to network congestion, improper queuing or configuration errors, this steady stream can be lumpy or the delay between each packet can vary instead of remaining constant.

3.3.3 End-to-end delay

End-to-end delay refers to the time taken for a packet to be transmitted across a network from source to destination. End-to-end delay has a critical importance when a packet arrives too late at the receiver as a consequence, the packet can be effectively lost. Lost packets due to delay have a negative effects on the received quality.

3.3.4 Throughput

The throughput is a key parameter to determine the rate at which total data packets are successfully delivered and received through the channel in the network. The throughput is usually considered in bits per second (bits/s or bps), and sometimes in data packets per second or data packets.

3.3.5 Packet Delivery Ratio

Packet delivery ratio is defined as the ratio of data packets received by the destinations to those generated by the sources. Mathematically, it can be defined as:

PDR = S1/S2 ;

Where, S1 is the sum of data packets received by the each destination and S2 is the sum of data packets generated by the each source.

3.4 HARDWARE REQUIREMENTS

The minimum requirements needed to perform operations are

- Intel Pentium Processor at 2 GHz or Higher
- RAM 256MB or more
- Hard disk capacity 10GB or more

3.5 SOFTWARE REQUIREMENTS

The software required to perform the implementation are

- Windows or Linux Operating System (Ubuntu, Fedora)
- Network Simulator(ns2)
- Gedit

CHAPTER 4

PERFORMANCE ANALYSIS

4.1 ANALYSIS OF SYSTEM DEVELOPED

The system developed is analyzed using experimental method in which we give different values and then compare the results accordingly.

4.1.1 Performance is analyzed using xgraphs experimentally

The characteristics of NS2 parameters like throughput, end to end delay, packets information, etc can be plotted using xgraph . Here we are using the xgraph to compare the result at different stages, to analyse the behavior of the network before and after the TCP attack.

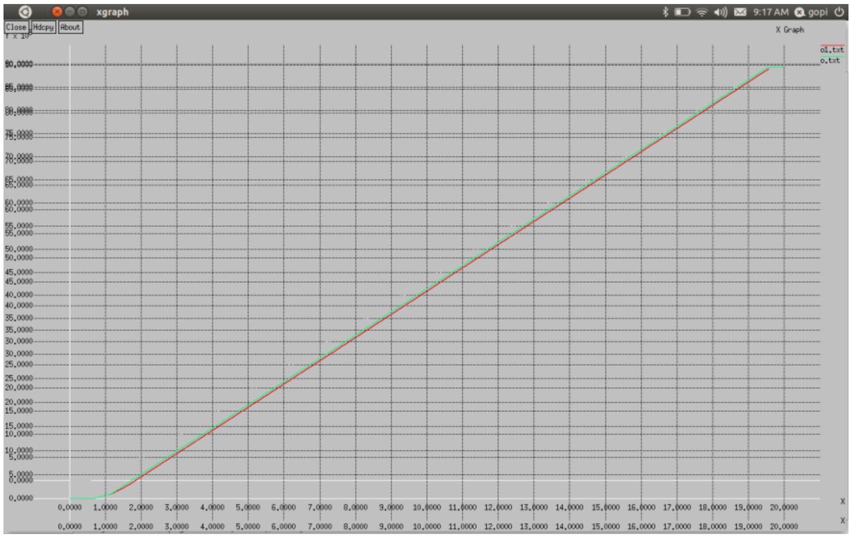


Fig. 4.1 A simple xgraph

In fig. 4.1 we see a simple xgraph with 2 lines in which the green line indicates the output before attack and can be obtained by the command:

```
Out.tr-> initial throughput
```

Here, Out is the file containing the code without attack.

The red line indicates the output obtained after the TCP attack on the same network and can obtained using the command:

Out1.tr-> Throughput after attack

where Out1 is the file that contains the attack affected code.

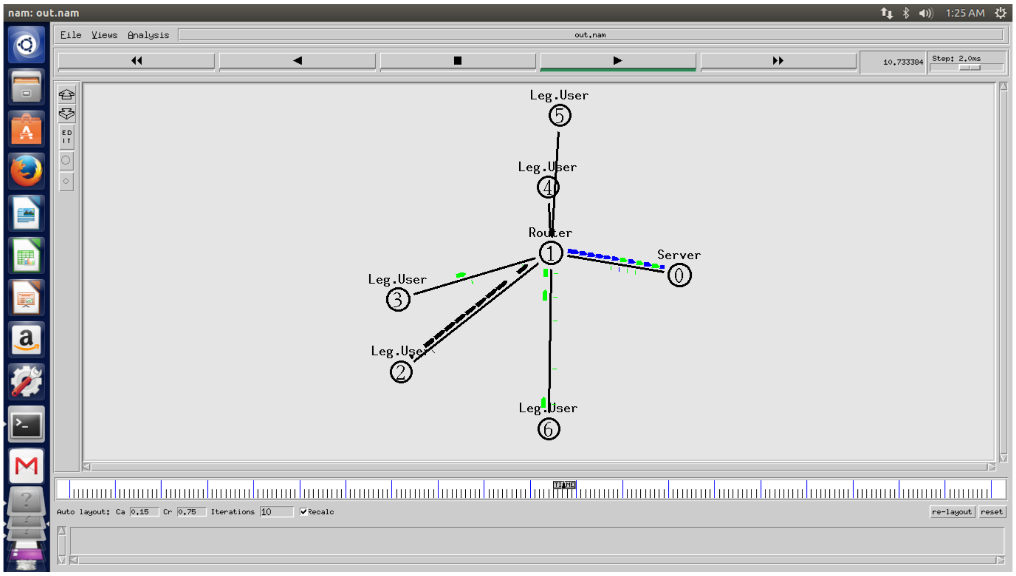
4.2 Detection Mechanism

4.2.1 Scenarios

4.2.1.1 Initial Scenario

This is the scenario of normal conditions.

There are 5 legitimate users using server and router to access internet. The 3-way handshaking is taking place between the server and the legitimate user i.e. the packets are sent from attacker to server and the server is sending syn-ack back to the users. Now to make the connection successful, acknowledgement is to be sent by user. In this way, connection is established between the server and the users. Both exchange information in the form of packets.



Auto layout: Ca 0.15 Cr 0.75 Iterations 10 ☑ Recalc

Fig 4.2 Scenario of transferring packets in a healthy network

4.2.1.2 Scenario After attack

This is the scenario of conditions when attack has taken place.

In this, apart from 5 users, 2 attackers are present who will interrupt he communication between the users and server. Here, attacker is establishing half open connections to the user and the server. As a result of which, both the server and the users are busy in acknowledging those request and the legitimate requests. The packets are dropped in this way and there is a difference in the throughput.

This tcp attack allocates the bandwidth and consumes the buffer. The client and the server will not be able to exchange packets and as a result, there will be problems in exchanging of data. The fig 4.3 shows red packets as the infected ones i.e. sent by the attacker to create half open connections.

The user and the clients will have to suffer because of the attacker.

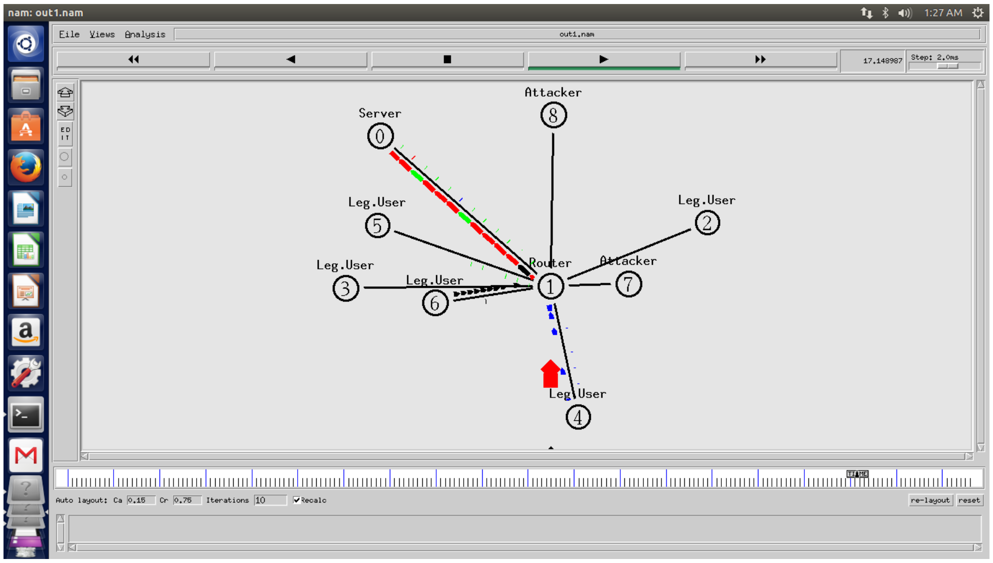


Fig. 4.3 Transfer of packets in an TCP SYN Flood affected network

4.2.2 DETECTION USING DIFFERENT PARAMETERS

4.2.2.1 Packet Loss

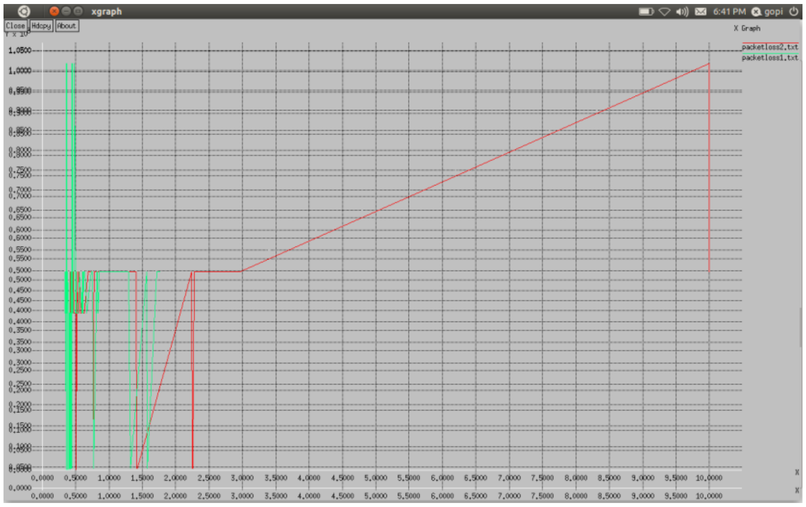


Fig. 4.4 Packet loss analysis

We observe that packet loss of data is high in the beginning for the network without attack, this is because for a healthy network the no. of packets sent are high as compared to the affected network hence greater chances of loss due to congestion. With time the packet loss eventually stops in normal network as all packets are successfully sent and acknowledgement is received but increases for the network affected with attack as the server keeps on sending acknowledgement but no response is received from the client side.

To understand better we can refer the following table:

	No attack	With attack
Number of sending packets	18206	121267
Number of lost packets	92	2024
Packet Loss rate (%)	0.51	1.67

Table 1 Packet loss comparison

4.2.2.2 Jitter

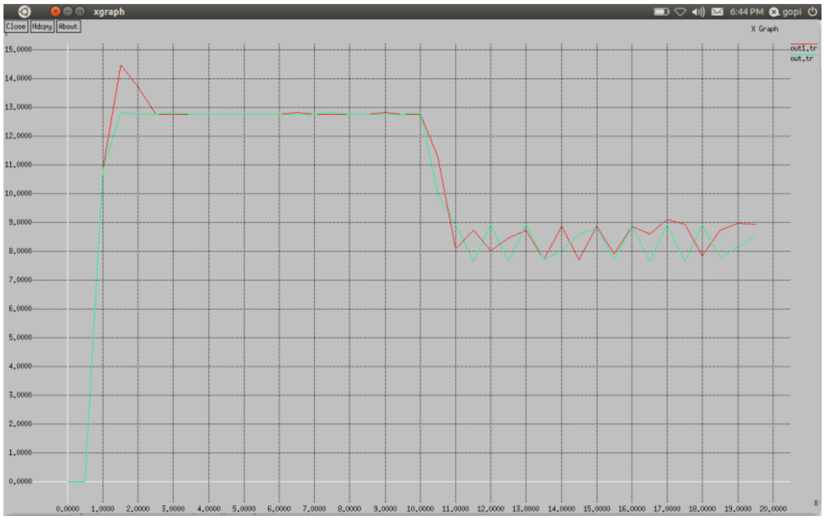


Fig. 4.5 Jitter variation

We observe that the delay variations in healthy network increase eventually and then are constant for a certain value and decreases but for an affected it takes a sudden jump before being constant. This is because delay increases for the affected packets. Though there is delay in both the networks but there is a variation in affected network from the normal one due to the SYN Flood.

4.2.2.3 End- to-end Delay

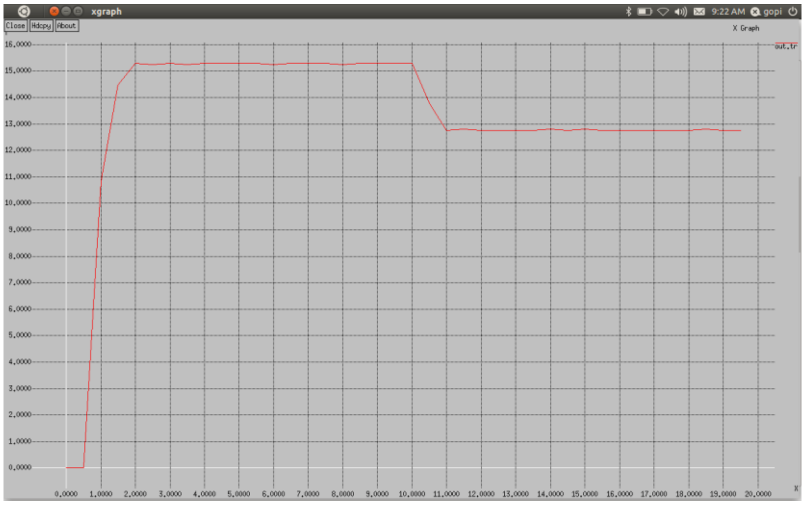


Fig 4.6 End-to-end delay before Attack

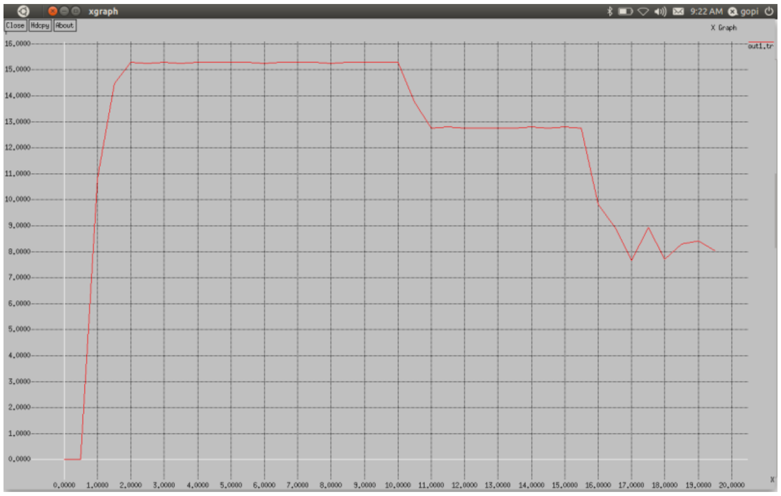


Fig. 4.7 End-to-end delay after attack

End to end delay is calculated to see the impact of TCP Syn flood attack on a network. Normally, the end to end delay should increase for an affected network but here we observe that end to end delay in a healthy network is constant after a certain period of time but fluctuates in affected network.
Initially, in affected network also the attacker behaves as a legitimate user and the delay obtained is same as that of healthy network. But later with increase in time the affected network start misbehaving for a defined period of time in different scenarios.

4.2.2.4 Throughput

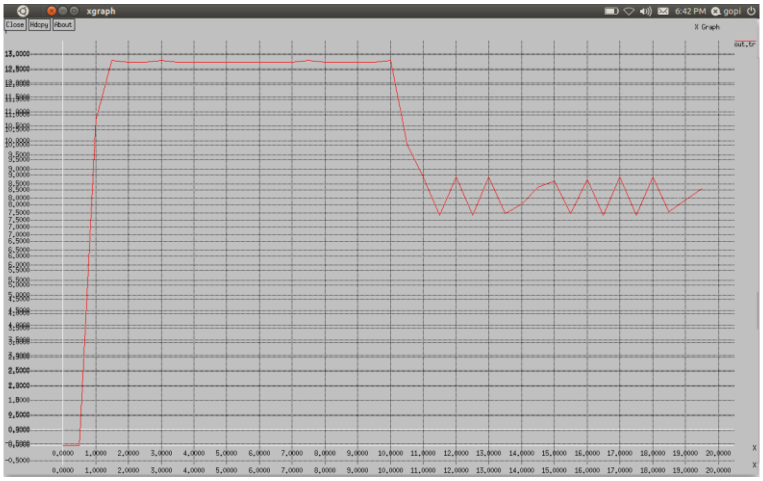


Fig. 4.8 Throughput of an healthy network

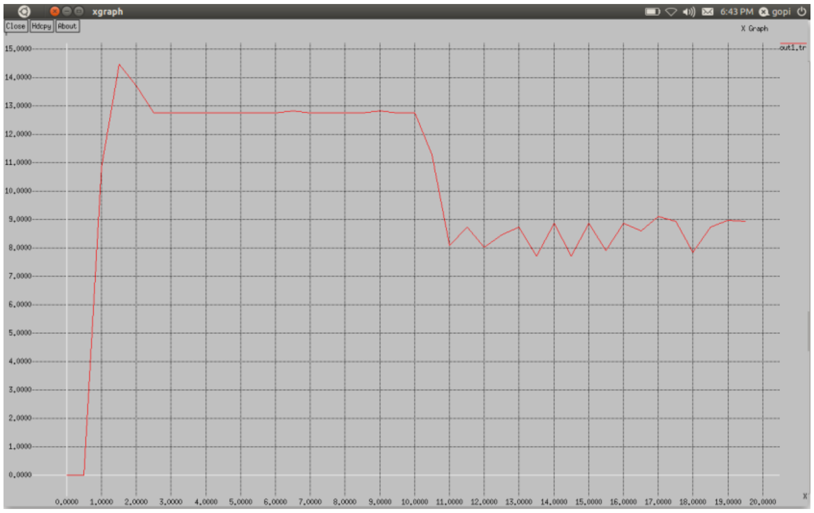


Fig. 4.9 Throughput of an affected network

We observe that throughput of the network decreases when affected by the attack. The increase in delay of an affected network resulted in decrease in throughput of the network. The number of packets and the size of the packets is calculated to find the throughput of the network. As the number of packets decrease when the network is affected so with this decrease in number the throughput also decreases.

4.3 Prevention Mechanism

4.3.1 Using IPsec

4.3.1.1 Existing System Before Attack

This is the scenario without the IPsec protocol before attack. In this, there is a base station and several nodes which will receive some packets from base station. Node 18 represent the base station responsible for providing connection to the underlying nodes. This is the normal packet flow where the attack has not been initialized. We can see the pckets flowing in red colour. Some packets are dropped in the path too.

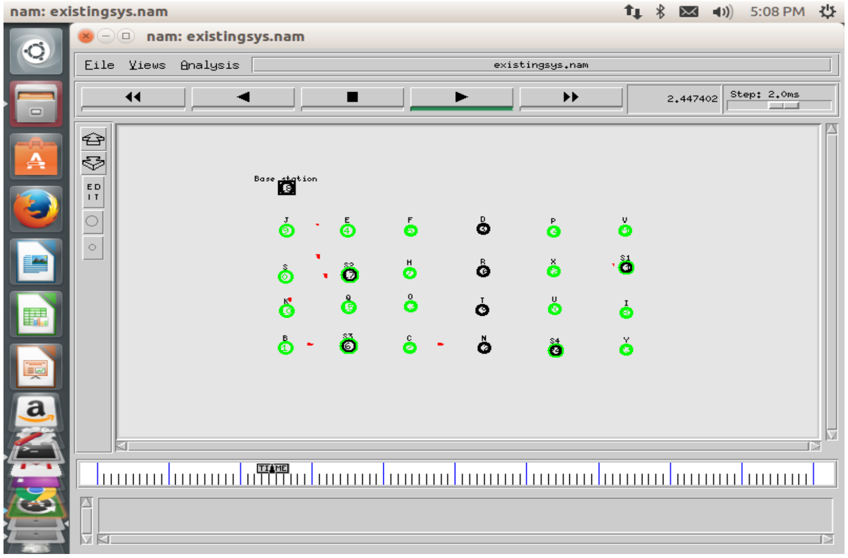


Fig. 4.10 Existing System before attack

4.3.1.2 Existing System After Attack

This is the scenario without the IPsec protocol after attack. node 18 represent the base station responsible for providing connection to the underlying nodes. S1,S2,S3,S4 represent the sentinel nodes manages some set of nodes under it for providing connection. Sentinels are interconnected with each other and provide services to the underlying nodes. Node D,R,T,N acts as the attacker blocks the connection as the bridge.So the sentinels S1 and S4 loses the connection this leads to the packet drop.

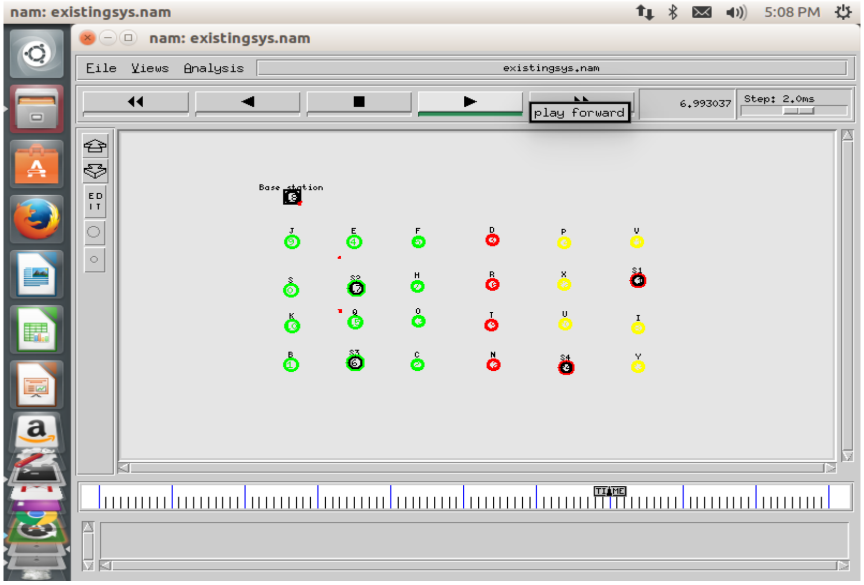
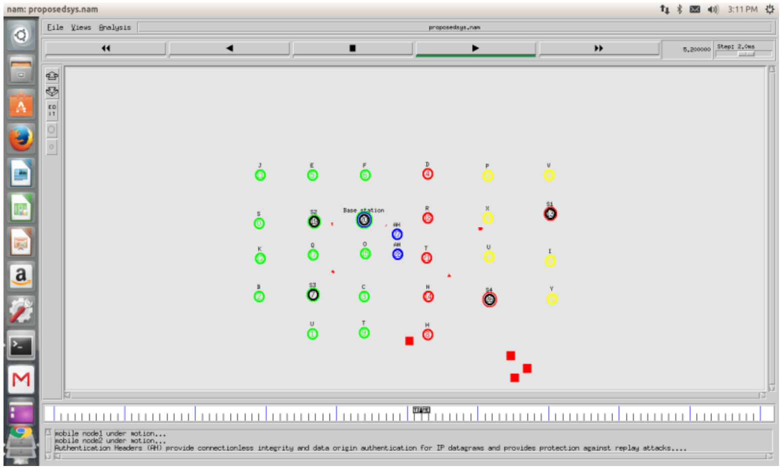


Fig. 4.10 Existing System after attack

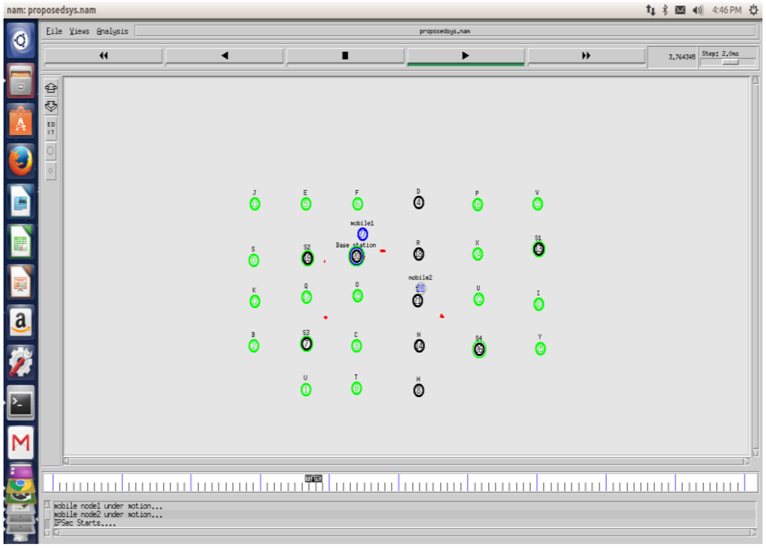
4.3.1.3 Proposed System using IPsec

In the proposed methodology Mobile nodes are introduced with Ipsec configuration. In this technique two nodes are introduced mobile 1 and mobile 2. S1,S2,S3,S4 are the sentinels and node 18 acts as the base station. Sentinels are interconnected with each other and provide services to the underlying nodes as existing approach. Node D,R,T,N acts as the attacker blocks the connection as the bridge. When the attacker blocks the connectivity between sentinels, the mobile nodes act as the connector for the sentinels S1 and S3. So the connection is retained. Packet drop due to attacker is reduced largely.



```
mobile node1 under motion...
mobile node2 under motion...
Authentication Headers (AH) provide connectionless integrity and data origin authentication for IP datagrams and provides protection against replay attacks....
```

Fig. 4.11 Proposed system1



```
mobile node1 under motion...
mobile node2 under motion...
IPSec Starts....
```

Fig. 4.12 Proposed system2

4.3.1.4 Xgraphs showing Packet Loss

This is the xgraph showing the packet loss in both the scenarios. The red graph shows the packet loss of the scenario without the IPsec algorithm. The green graph represents the packet loss when IPsec protocol is applied to the existing scenario. We can observe that the no. of packets lost in the proposed approach is lesser than the no. of packets lost in the existing approach. Thus, the proposed algorithm is highly efficient in terms of the packet loss.

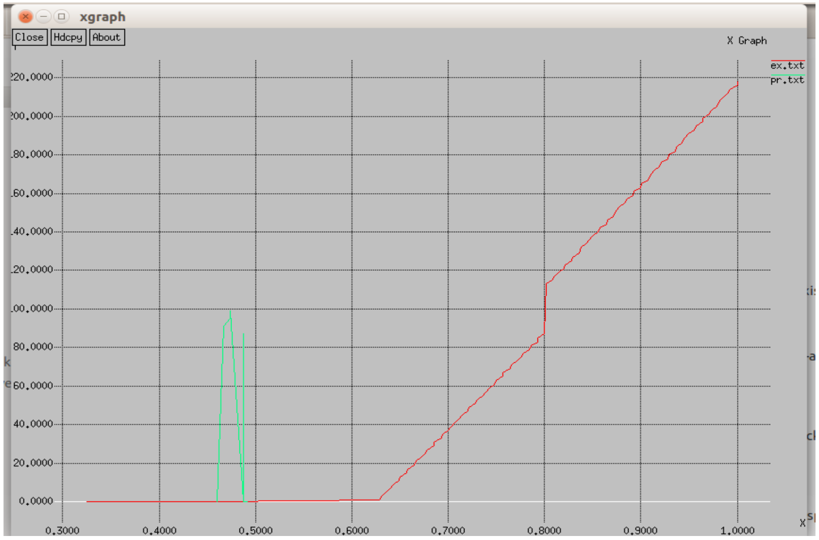


Fig. 4.13 Xgraph showing packet loss

4.3.1.5 Xgraphs showing Data Transmission Rate

This is the xgraph showing the data transmission in both the scenarios. The red graph shows the data transmission rate of the scenario without the IPsec algorithm. The green graph represents the data transmission rate when IPsec protocol is applied to the existing scenario. We can observe that in proposed approach, data is transmitted at a much higher rate than in the existing one. More no. of packets are transferred in lesser time efficiently in scenario where we applied Ipsec algorithm.

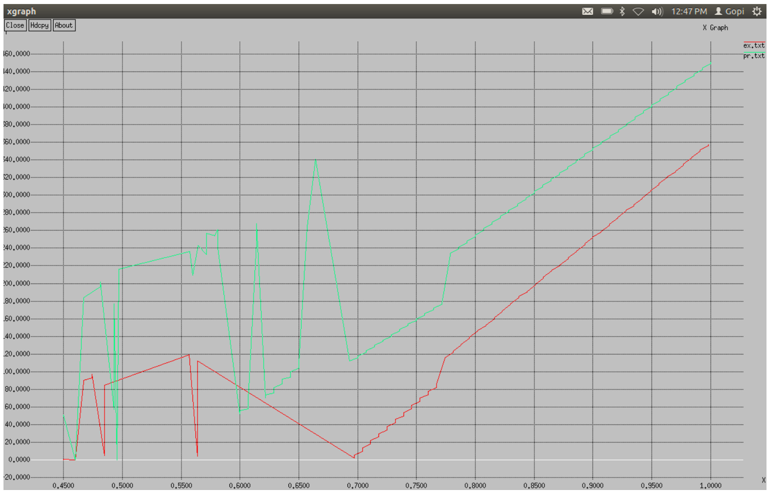


Fig. 4.14 Xgraph showing data transmission rate

4.3.1.6 Comparative Analysis of Packet Loss Rate in Existing and Proposed Approach

Here we can see the packet loss rate clearly, which is much higher in the existing approach

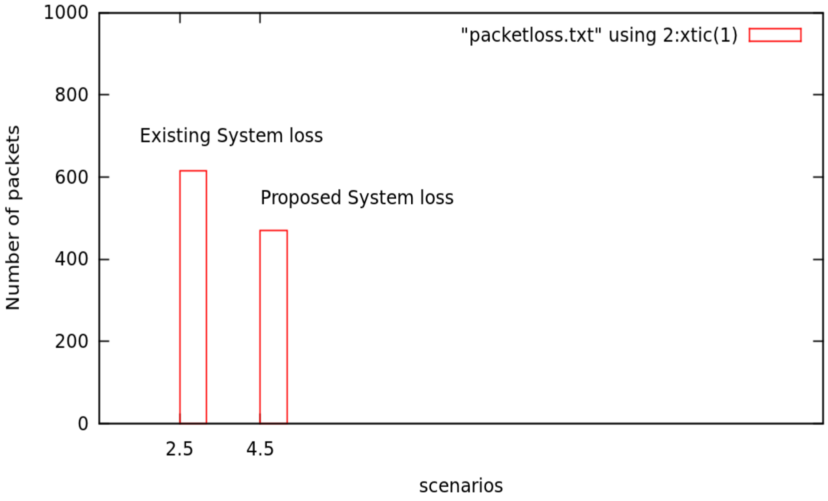


Fig. 4.15 Xgraph showing packet loss rate

4.3.1.7 Comparative Analysis of Jitter Parameter in Existing and Proposed Approach

This is the xgraph showing jitter in both the scenarios. The red graph shows the jitter of the scenario without the IPsec algorithm. The green graph represents jitter after IPsec protocol is applied to the existing scenario. We can observe that in proposed approach, delay is lesser than in existing approach.

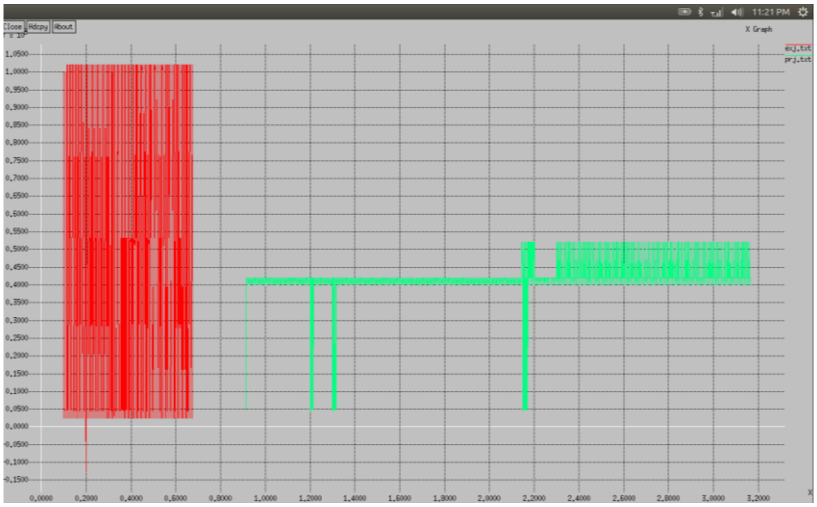


Fig. 4.16 Xgraph showing jitter

4.3.1.8 Comparative Analysis of Throughput Parameter in Existing and Proposed Approach

This is the xgraph showing throughput parameter in both the scenarios. The red graph shows the throughput parameter of the scenario without the IPsec algorithm. The green graph represents throughput parameter after IPsec protocol is applied to the existing scenario. We can observe that in proposed approach, throughput is much more than in the existing approach.

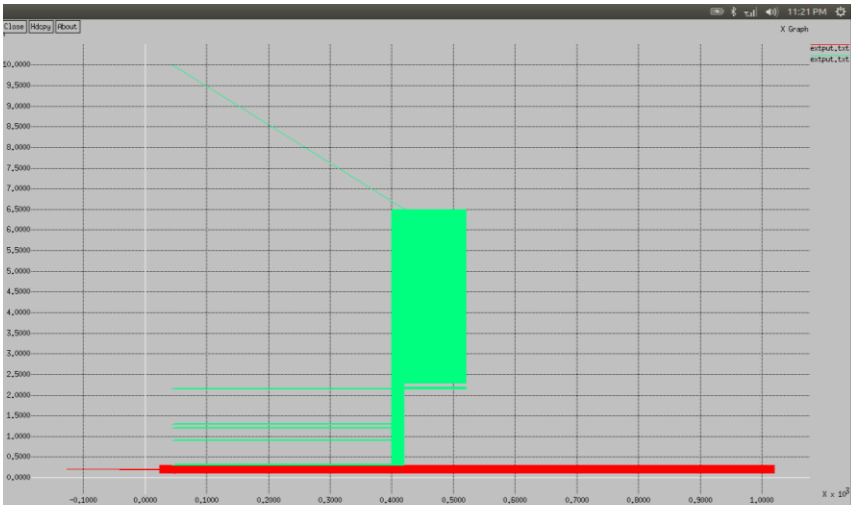


Fig. 4.17 Xgraph showing throughput

4.3.2 Using Ant Colony Optimisation

4.3.2.1 Existing System Before Attack

In the existing system, there are no. of nodes .The data flows from source to destination. The node Node 0(source) needs to send packets to the Node 4(destination). So it transfers the packet through the route via node1, node3 (attacker). So the route is like Node 0->Node 1->Node 3->Node 4.

Node 3 is the attacker joins the network as the trusted user with RSS frequency normal as other nodes. So it causes TCP SYN attack.

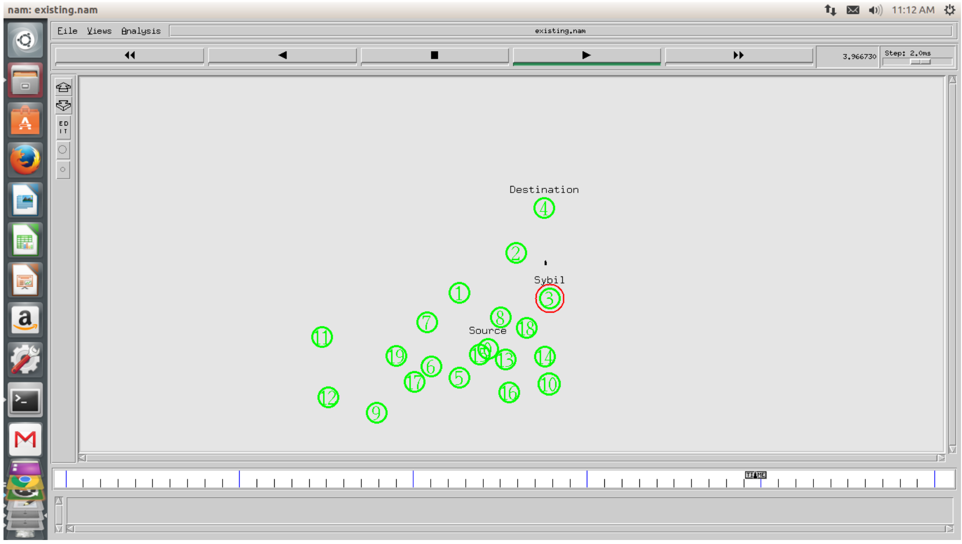


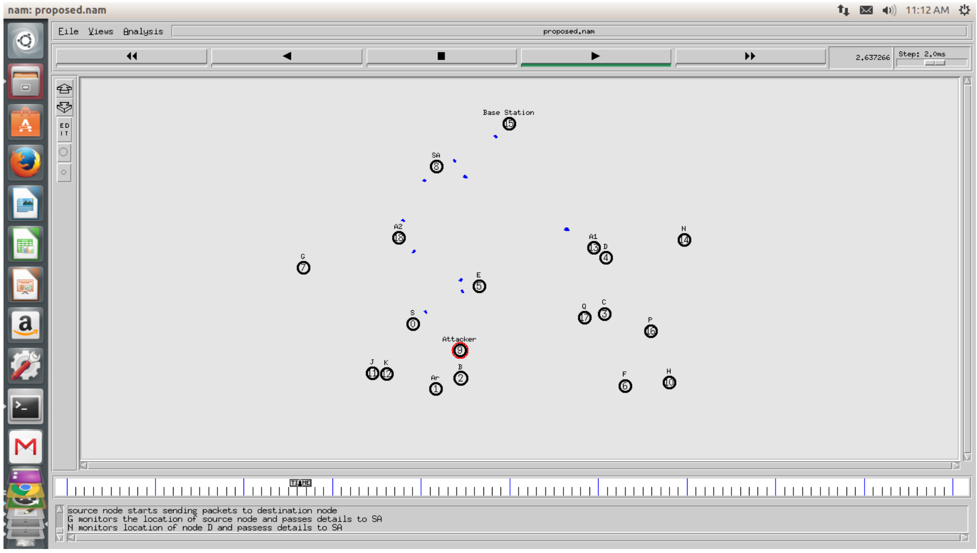
Fig. 4.18 Existing System without ant colony algorithm

4.3.2.2 Proposed System

In the proposed system, the Node 0 needs to send packet to Node 4. Node 5 initially get service from base station so Node 0 broadcasts route request to Node 5->Node 18->Node 18-> Node 8-> Node 15-> Node 8-> Node 13-> Node 4. Once the route request reaches the destination Node 4 reply for the route request in similar fashion of the route request.

Node 0 finds the destination node id along with base station id. So it starts sending packets in that path. Node 7 and Node 14 monitors the position of source and destination nodes under movement.

Attacker node Node 9 remains unaccessible because it is not registered under base station to connect to network. The proposed system follows the trusted communication so there is no possibility of accessing the resources.



```
source node starts sending packets to destination node
G monitors the location of source node and passes details to SA
N monitors location of node D and passess details to SA
```

Fig. 4.19 Proposed System with ant colony algorithm

Some points regarding xgraphs

Label X axis denotes the time in milliseconds and Y axis denotes the packets size.

The spots in the graphs denotes the packet loss occurred in the particular time for existing system and green represent the packet loss for proposed system.

The thick red represents the place where more packet drop happened.

4.3.2.3 Xgraph showing Jitter

Jitter is the no. of packets arriving in a particular node/time duration. We can see that the jitter of the proposed approach is much higher than the jitter in the existing approach. Higher no. of packets are transferred in a lesser time in the proposed approach. Thus, ant colony algorithm proves to be efficient in terms of jitter and delay.

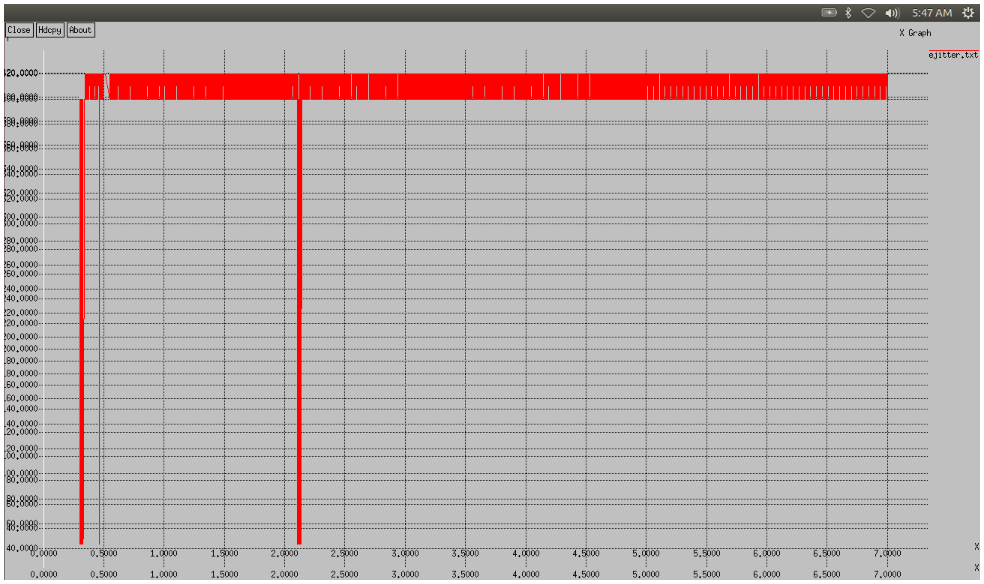


Fig. 4.20 Jitter Parameter Analysis for Existing Approach

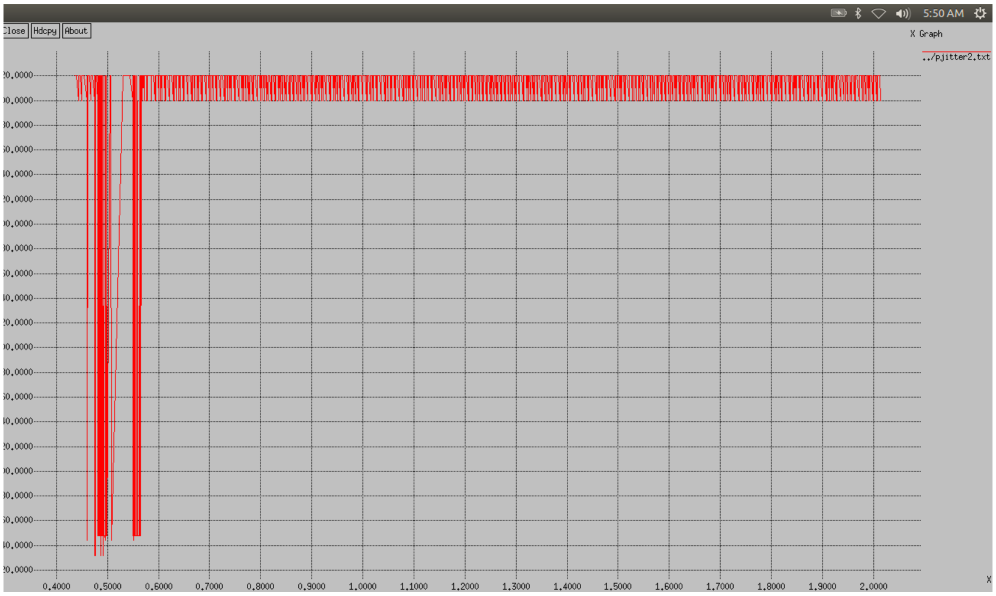


Fig. 4.21 Jitter Parameter Analysis for Proposed Approach

4.3.2.4 Xgraph showing Throughput parameter

In this xgraph, the green graph represents the existing algorithm and the red graoh represents the proposed slave ant algorithm. It could be clearly seen that the throughput of the green graph is much higher than the throughput of the red graph. Thus, slave ant gives more throughput than the old existing approach.

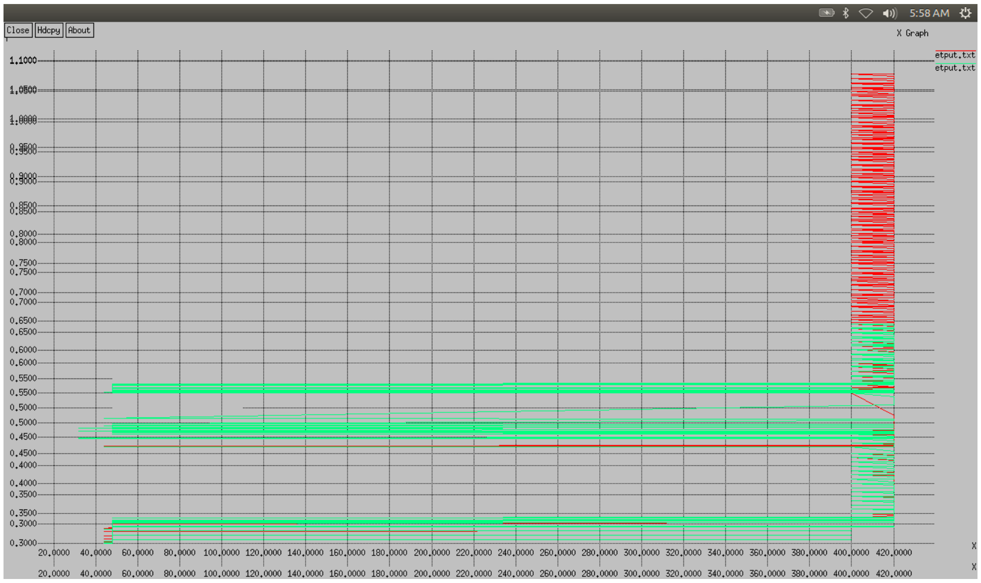


Fig. 4.22 Throughput Parameter Analysis

4.3.2.5 Xgraphs showing Data transfer parameter

In this xgraph, green graph represents the proposed algorithm and the red graph represents the existing approach. In the proposed approach, data transfer rate is higher and more efficient than in the existing approach.

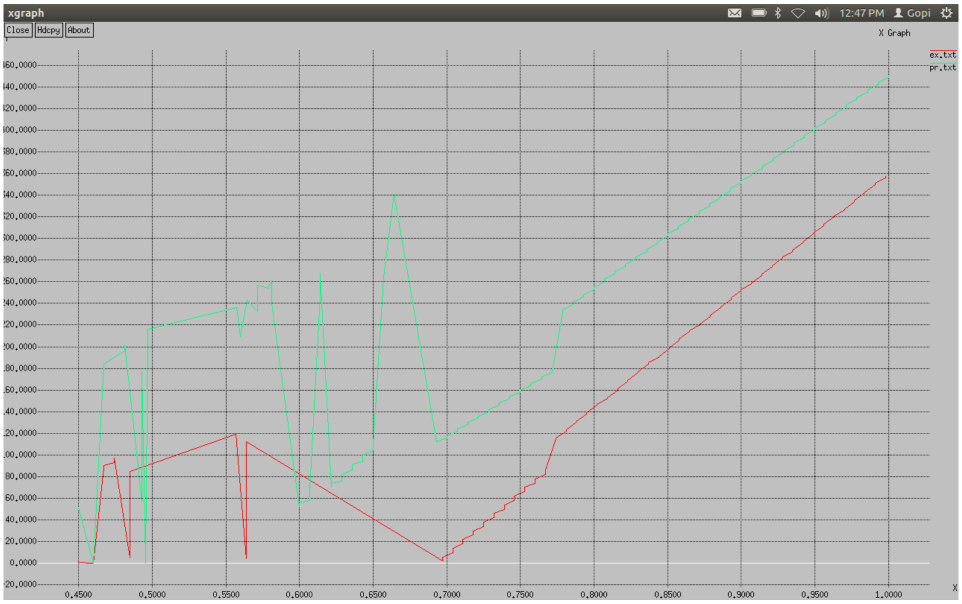


Fig. 4.23 Data Transfer Parameter Analysis

4.3.2.6 Xgraphs showing packet loss

In our approached we compared with packet drop in the existing system there were 4024 packets dropped due to TCP SYN attack but in our proposed technique only 4 packets dropped

- Existing technique packet loss=4024/9.1*100=44,219.78
- Proposed technique Packet Loss=4/9.1*100=43.95

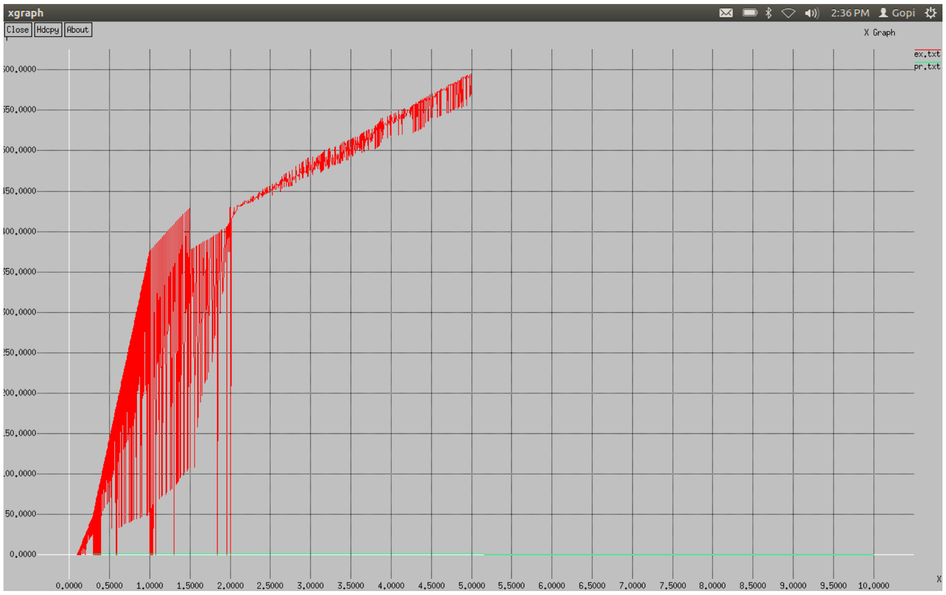


Fig. 4.24 Packet Loss Analysis

In this chapter we saw that how the performance parameters change when the network is affected by an attack. By analysing these factors time to time we can detect if the network is affected by any attack or not. Then we applied the algorithms to defend the attacks. Hence, using above techniques we can detect as well as defend the network from a tcp syn flood attacks.

CHAPTER 5

CONCLUSION

TCP SYN Flood Attack is a foremost hazard to the Internet these days. Valid users have a hard time connecting to the servers that are open to TCP attacks. These attacks impersonate normal clients and consume the system resources at extensive scale, in this manner considerably refusing service to the normal nodes. This work proposed a TCP defense solution which consigns distrust evaluation to a timestamp in fraction to its divergence from normal behavior and uses a TCP resistance approach to choose whether and at what time the timestamp is utilized. Using an experimental simulation in ns2, we verified the strength of TCP attack as well as the effectiveness of TCP defense capability to prevent it.

We have developed a mechanism to detect the TCP SYN flood attacks and an active defense mechanism to protect against TCP SYN Flood attacks using ns2. Our mechanism is based on different parameters that will help to detect whether the attack has taken place or not. SYN attack is detected by combined approach of using TCP flags, port, and by tracing the route of the source. By certain parameters, we detected that attack took place. Our defence mechanism is robust and efficient to detect various SYN flooding attacks that includes IPsec protocol and Ant Colony Algorithm. It achieves high detection accuracy and short detection time.

This work focus specifically on detection and avoidance of TCP SYNC attacks on the network infrastructure whether it is concerned with the sniffing of packets or stealing the actual identify of the genuine node. However, there are number of techniques to address and remove this issue, but our approach is efficient enough because of the fact that this proposed approach is relying on the security trust architecture. It means there is a trust between the data transmission channel. The concept of mobile node and server agents are integrated to remove any probability of the attack. In the proposed approach implementation, better results in terms of jitter, packet loss and throughput are obtained in different simulation scenarios. In the proposed work, the current proposed approach can be further enhanced using simulated annealing, an excellent technique for optimization on multiple parameters.

FUTURE WORK

As future work, it is planned to improve our routing approach to be effective in proper WSN settings, including nodes having high mobility. The improved approach can be carried out considering more factors that can be applied in a routing problems such a delay factor, congestion control etc. Other approaches may include Filtering, Increasing Backlog, Reducing SYN-RECEIVED Timer, Recycling the Oldest Half-Open TCB, SYN Cache, SYN Cookies, Hybrid Approaches, Firewalls and Proxies that could defend the TCP attacks. Also some effective algorithms can be applied that requires no modification or configuration of end-host software.

Thus, Algorithms with better efficiency could be applied to prevent such attacks on a larger network.

REFERENCES

[1] Saravanan , Gowri Shankar ,“An active defense mechanism for tcp syn flooding attacks”

[2] Monal r., Toomey, “New integrated defense and traceback approach for denial of service attacks.”

[3] Ming Li, Jun Li, Wei Zhao, “Simulation Study of Flood Attacking of DDOS”, 2008 International Conference on Internet Computing in Science and Engineering, 28-29 Jan. 2008,page no. 286 - 293

[4] Sanjyoti Tarai, Khaleel Ahmad, Jayant Sekhar, “Prevention of SYN Flood DOS Attack”, Volume-2,Issue3, May-June,2013

[5] Hung Hom, Kowloon , “An Active Defense Mechanism for TCP SYN flooding attacks”, Proceedings of the 2005 11th International Conference on Parallel and Distributed Systems (ICPADS'05)

[6] Deepak Singh Rana, Sushil Kumar Chamoli ,Naveen Garg ,“A Study and Detection of TCP SYN Flood Attacks with IP spoofing and its Mitigations”, Deepak Singh Rana et al ,Int.J.Computer Technology & Applications,Vol 3 (4)

[7] Xi’an, “Applied informatics and communication”, International Conference , ICAIC 2011 , China, August 20-21,2011,Proceedings Part 3

[8] C. L. Schuba, I. V. Krsul, M. G. Kuhn, E. H. Spafford, A. Sundaram and D. Zamboni, “Analysis of a Denial of Service Attack on TCP”, Proceedings of IEEE Symposium on Security and Privacy, May 1997.

[9] SANS Institute InfoSec Reading Room,“Vulnerability of IPSEC: A Discussion of Possible Weaknesses in IPSEC Implementation and Pro”, Daniel Clark, Version 1.3, 14 March 2002

[10] Chuan Cai, Liang Yuan, “Intrusion Detection System based on Ant Colony System”, JOURNAL OF NETWORKS, VOL. 8, NO. 4, APRIL 2013

[11] Wei Chen, Dit-Yan Yeung, “Defending Against TCP SYN Flooding Attacks Under Different Types of IP Spoofing”, volume 3, pages 1353— 1357, Dec 2003.

[12] Samad S. Kolahi, Amro A. Alghalbi , Abdulmohsen F. Alotaibi, Saarim S. Ahmed, and Divyesh Lad, “Performance Comparison of Defense Mechanisms Against TCP SYN Flood DDoS Attack”,
Unitec Institute of Technology, Auckland, New Zealand

[13] Kanika, Renuka Goyal, Gurmeet Kaur , “Monitoring of traffic over the victim under TCP SYN flood in a lan”, IJRET, Volume 03 , Issued on 04Apr-2014

[14] S.Gavaskar ,Madurai Kamaraj, “Three Counter Defense Mechanism for TCP SYN Flooding Attacks”, International Journal of Computer Applications (0975 – 8887) Volume 6– No.6, September 2010

[15] Vasilios A. Siris and Fotini Papagalou,2004, “Application of Anomaly Detection Algorithms for Detecting SYN Flooding Attacks”, IEEE Communications Society Globecom.