

# Bandwidth Starvation attack and Detection

By

**Hetuk Upadhyay**

12MCEI40



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
INSTITUTE OF TECHNOLOGY, NIRMA UNIVERSITY  
AHMEDABAD-382481

May 2014

# Bandwidth Starvation attack and Detection

## Major Project

Submitted in partial fulfillment of the requirements

For the degree of

Master of Technology in Computer Science and Engineering

By

**Hetuk Upadhyay**

(12MCEI40)

Guided By

**Prof. Sharada Valiveti**



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
INSTITUTE OF TECHNOLOGY, NIRMA UNIVERSITY  
AHMEDABAD-382481

May 2014

## Undertaking for Originality of the Work

I, **Hetuk Upadhyay**, Roll. No.**12MCEI40**, give undertaking that the Major Project entitled “**Bandwidth Starvation attack and Detection**” submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science and Engineering(INS)** of Nirma University, Ahmedabad, is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Signature of Student

Endorsed by

Hetuk Upadhyay

Prof.Sharada Valiveti

## Certificate

This is to certify that the Major Project entitled “**Bandwidth Starvation attack and Detection**” submitted by **Hetuk Upadhyay (12MCEI40)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science and Engineering(INS)** of Nirma University, Ahmedabad is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of my knowledge, haven’t been submitted to any other university or institution for award of any degree or diploma.

**Prof. Sharada Valiveti**

Guide, Associate Professor, PG-Coordinator(INS),

Department of C.S.E.,

Institute of Technology,

Nirma University, Ahmedabad.

**Dr. Sanjay Garg**

HOD [C.S.E. Dept.],

Institute of Technology,

Nirma University, Ahmedabad

**Dr K Kotecha**

Director,

Institute of Technology,

Nirma University, Ahmedabad

## Abstract

Bandwidth Starvation attack is one type of Distributed Denial of Service (DDoS) attack. Here victim is mostly some server, but some time victim could be any network also. Attacker has large number of computers acting as bots. Attacker attacks server or a network through these bots. These bots number is such large that it can create enough traffic for the server to not to able to respond to any other request of the user. Here these bots are across different network and through those networks these bots attacks the victim. Many routers comes in way to this attack and if the router is capable to stop these attacks than impact of the attack can be reduced at victim side.

All the routers are not able to protect themselves form these kind of attacks. There are some solutions to these attacks in router but they are time consuming and not the best solution. These all solutions are some security protocols or some foreign Intrusion Detection System (IDS). Here if the router has its own IDS and IDS database than the problem could be solved.

The focus is to create one attack on router and then detecting the same attack inside it. Here I have created an attack on router and then protecting it by detecting attack, type of attack and IP address from where attack is taking place.

## Acknowledgements

My deepest thanks to **Prof. Sharada Valiveti**, Associate Professor, PG-Coordinator(INS), Department of Computer Science and Engineering, Institute of Technology, Nirma University, for giving me an opportunity and guidance throughout the project. It was only due to her valuable opinion, cheerful enthusiasm and ever friendly nature that I was able to do part of my research work in a respectable manner.

It gives me an immense pleasure to thank **Dr. Sanjay Garg**, Honorable Head of Computer Science and Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr. Ketan Kotecha**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

I would also thank my Institution, all my faculty members in Department of Computer Science.

I would like to thank my colleagues for being with me and help me.

**Hetuk Upadhyay**

**12MCEI40**

# Contents

<b>Undertaking for Originality of the Work</b>	<b>iii</b>
<b>Certificate</b>	<b>iv</b>
<b>Abstract</b>	<b>v</b>
<b>Acknowledgements</b>	<b>vi</b>
<b>List of Figures</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Project Definition . . . . .	2
1.3 Project Scope . . . . .	3
<b>2 Literature Survey</b>	<b>4</b>
2.1 SYN flood attack . . . . .	4
2.1.1 SYN KILL[1] . . . . .	5
2.1.2 DelAy pRoBing (DARB)[2] . . . . .	6
2.2 Data flooding Attack . . . . .	6
2.2.1 Adaptive Bandwidth Allocation [3] . . . . .	6
2.2.2 Ingress/Egress filtering [4] . . . . .	7
2.2.3 SIFF(Stateless Internet Flow Filter) [5] . . . . .	8
2.2.4 Router based packet filtering [4] [6] . . . . .	8
2.2.5 History based IP filtering [4] . . . . .	9
2.2.6 Capability based method [4] . . . . .	9
2.2.7 Secure Overlay Service (SOS) [4] . . . . .	10
2.2.8 Secure Address Validity Enforcement(SAVE) [4] . . . . .	10
2.2.9 MUltiLevel Tree for Online Packet Statistics (MULTOPS) [7]	10
2.2.10 Shield [8] . . . . .	11
<b>3 Implementation Methodology</b>	<b>13</b>
3.1 Literature Survey Summary . . . . .	13
3.2 Implementation Tool . . . . .	13

3.3	Implementation methods . . . . .	14
<b>4</b>	<b>Bandwidth Starvation attack and Results</b>	<b>17</b>
4.1	Packet Modification . . . . .	17
4.2	Implementation . . . . .	18
4.3	Graphs . . . . .	19
<b>5</b>	<b>Packet Signature and Detecting attack Packets</b>	<b>23</b>
5.1	Packet Signature . . . . .	23
5.2	Detecting attack packet . . . . .	24
5.3	CPU and Network performance . . . . .	26
<b>6</b>	<b>Conclusion and Future Work</b>	<b>29</b>
6.1	Conclusion . . . . .	29
6.2	Future Work . . . . .	30



# List of Figures

2.1	Flow of SYN KILL [1] . . . . .	5
2.2	schematic MULTOPS in victim oriented mode [7] . . . . .	11
2.3	working of shield [8] . . . . .	12
3.1	Simple router working flowchart . . . . .	14
3.2	Simple router working flowchart with IDS . . . . .	15
3.3	Simple IDS flowchart . . . . .	16
4.1	IP and ICMP header changing code . . . . .	18
4.2	ICMP echo request flooding attack implementation . . . . .	19
4.3	bandwidth utilization . . . . .	20
4.4	ICMP packet Flooding . . . . .	20
4.5	tcp packet flow . . . . .	21
4.6	UDP packet flow . . . . .	22
4.7	combine packet flow . . . . .	22
5.1	Simple IDS flowchart . . . . .	25
5.2	IDS results . . . . .	26
5.3	CPU in normal performance . . . . .	27
5.4	CPU in attack performance . . . . .	27
5.5	network in normal performance . . . . .	28
5.6	network in attack performance . . . . .	28

# Chapter 1

## Introduction

### 1.1 Introduction

Bandwidth Starvation attack is one type of DDoS(Distributes Denial of Service) attack. Bandwidth Starvation attack is quite effective and have a devastating effect on network. In this attack attacker floods lots of packets in to network towards the target server.

In order to do so the attacker takes over many PCs that are connected to the Internet and convert them in to zombies. Zombies are the kind of PCs which are controlled by the attacker and can make them to do anything that they are not supposed to do. These group of zombies are called bot net.

Attacker controls these bot net and through commands that attacker sends to these PCs they act accordingly. If attacker sends command to create TCP flood attack on server then those PCs will attack on server by TCP flood attack. Thus attacker controls those PCs remotely. There are many computers which are under control of the attacker. Attacker can create any attack to any server and because of the large number of computers under attackers control effect of the attack can be a disaster.

To stop this kind of attack we need to stop this kind of attack. These zombies

are placed all over the world on Internet. So that they can create this attack from different places to one server because all zombies are connected to Internet. If we can detect these attacks from router from where attack packets are coming then effect of the attack at server's end would be such sort that it can be handled easily. To do this the router should have some mechanism to first detect this attack. So if the router has IDS (Intrusion Detection System) then router can detect this attacks and can protect servers from these attacks.

## 1.2 Project Definition

In this project we are creating an Intrusion Detection system to protect router from bandwidth starvation attack. To do so, first we create an attack that can create bandwidth starvation in our network. Here we are using smurf 4.0 C program which sends ICMP echo request to target server or any router. Here ICMP packet is modified packet. In this packet all the information from destination IP address to the checksum of packet is calculated and then given to the packet.

After creating an attack to router next task is to identify the attack packets which are ICMP echo request packets as we know. Here we have to identify the packets entering inside the router. Here to identify packets we are using signature based Intrusion Detection system(IDS) approach. Each and every packet has its own identity, But here we have changed the identity of the packet. This identity is called the signature of the packet. Here this kind of signature is already inside the IDS database. Here because the signature is with the IDS the IDS does not have to go for communication with other database and can identify the packet at same place. After identifying the attack on router IDS alerts the administrator by giving alert message.

## 1.3 Project Scope

The assumption and the scope of the project as as following.

- This project is for the wired network only. The wireless network is not included in this project.
- This project is based on the Ethernet wired cables and not any other so the result of this project is from Ethernet wired network only.
- This project is for router security and so that software router is needed for this project.
- Implementation is done for router to Analyse the packet and understand that what is the role of the packet in network management.
- Implementation is done to detect all the attack packets from the network and give alert message only.

# Chapter 2

## Literature Survey

The normal and the best way to bandwidth starvation attack is to flood and there are 2 types of flooding attack one is SYN flood and another is data flood. Both the flooding attacks are give bellow.

### 2.1 SYN flood attack

SYN flood is a TCP flooding attack technique through which attacker first sends TCP packets with SYN request. Here SYN request means that the client wants to communicate with the server and requesting connection with server. In reply server allocates the client one free connection and waits for the acknowledgement(ACK). But attacker do not give any type of ack and sends another SYN request with another IP and continue till server can not give any more connection. To protect the server from this kind of attacks the following techniques are implemented.

- Reduce the timeout period from the default to a short time
- Significantly increase the length of the backlog queue from the default
- Disable non-essential services, thus reducing the number of ports that can be attacked

But this all solutions can be bypassed and the attack can still take place. There are some effective methods so that this kind of attack can be stopped.

### 2.1.1 SYN KILL[1]

Software tool called SYN KILL, that lessens the impact of SYN flooding attacks, and in many cases defeat attacks completely. The program requires the ability to monitor and inject network traffic to and from the machines it is protecting. Ethernet is an example for a networking technology that satisfies this requirement. The program is called a monitor, because it reads and examines all TCP packets on the LAN after setting its network interface into promiscuous mode. The program is called active, because it can generate TCP packets in response to observed traffic and inject them into the network.

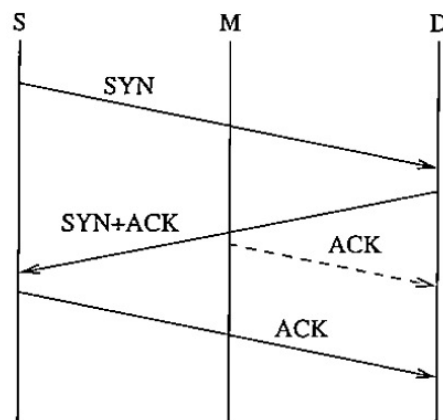


Figure 2.1: Flow of SYN KILL [1]

Here SYN request goes through SYN KILL and then to the victim. Then the victim sends the ACK to the source. Now the software sends ACK to the victim on behalf of the source and waits till the timeout. If the source does not reply in time, the connection is dropped and software sends a reset packet to the victim. If a reply comes, then communication goes on.

### 2.1.2 DelAy pRoBing (DARB)[2]

Delay is estimated using a method called DelAy pRoBing(DARB). The DARB traces outgoing paths toward network destinations by sending packets with special time-to live (TTL) lelds in the IP layer and then recording their time of deaths. The IP TTL leld limits the lifetime of packets transmitted across the Internet and is decremented by each forwarding device(routers). If the TTL leld reaches zero before the destination host is reached, the router drops the offending packets and transmits an ICMP(Internet Control Message Protocol) TTL exceeded in transit error message to the original host, informing the original host of the packet timeout. If the packet has been created appropriately, the destination host should return a nal packet to the original host when the packet reaches its destination. The time stamps of both the sent out packets and ICMP replied packets are recorded to calculate the delay between the original host and each router. The adopted DARB is similar to trace route , which works by sending packets with progressively longer TTL value.

## 2.2 Data flooding Attack

The data is flooded to the victim using zombies. The main thing for this type of attack is to identify the attacker's IP address. Here the attacker is zombie PCs. Possible solutions of the attack are listed bellow.

### 2.2.1 Adaptive Bandwidth Allocation [3]

In normal scenario inside router normal data comes. All packets have normal signature and manageable traffic is there at the router. But in attack scenario to many different IP addresses sends their data. All the packets has different signature as those are attack packets. This method propose a queuing algorithm for the network to detect attack and protecting from the attack. Here in this algorithm user profiling is done on base that which user sends his data through this path, How much data

that user is sending and when that user is sending data. In any normal day these users who send their data are listed in normal user.

In attack scenario by separating normal users from malicious users by of the Average Packet Rate (APR), and balancing bandwidths according to bandwidth flows, Quality by User (QBU) is attained to safeguard the normal users. Usually packet flows of normal users are in small amount and in short time span, whereas packet flows of malicious users are in large amount and in long time span, which might flood the network and stop network providers from providing services to users.

The drawback of this algorithm is that if the attack packets that the attacker is sending is average in number then this algorithm considers it as normal packets and also this algorithm can only protect flooding attacks and no other attack this algorithm can detect or protect.

### 2.2.2 Ingress/Egress filtering [4]

Ingress Filtering is for incoming traffic at router. Traffic coming towards the router is being analyzed in this method. In this method all analysis is done with given IP address database. Here only those IP addresses are allowed to come to the router which are given inside the domain. All the other IP addresses are blocked and not permitted to go inside router. Egress filtering is an outbound filter, which ensures that only assigned or allocated IP address space leaves the network. A key requirement for ingress or egress filtering is knowledge of the expected IP addresses at a particular port. For some networks with complicated topologies, it is not easy to obtain this knowledge.

For some networks router uses external database for the IP address. This all takes time and other process of the router gets delayed by that process. The other disadvantage is that it uses IP database. If the attacker gets to know that which IP addresses are permitted and can send data then attacker can spoof from those IP address can bypass the algorithm.



### 2.2.3 SIFF(Stateless Internet Flow Filter) [5]

SIFF (Stateless Internet Flow Filter) is other type of Ingress/Egress filtering protocol. In this protocol users get register to the server as clients. Now if the server can identify the user then IP filtering can be done easily. Here the registered user with server creates the privileged connection. The SIFF system provides a server with the ability to establish privileged communication with whatever clients. This privileged packets carry capabilities that are verified by the routers in the network, and are dropped when the verification fails. Here routers on network also uses this protocol to identify the privileged packets and the capabilities inside the packet. This means that all routers inside the network must have this protocol running so that they can identify the privileged clients from unprivileged clients. Thus SIFF is programmed to give preferential treatment to privileged packets, so that privileged packets are never dropped in favor of unprivileged ones.

As this protocol has its advantage this protocol has also disadvantage. One is that not all the router inside big or very bug network can have same set of the security protocol and configuration running. The other is that this protocol can not stop the attacker by listening inside the network. So that the attacker can analyze the privileged packet and so that attacker can copy the packet and capabilities inside the packet and carry out the attack.

### 2.2.4 Router based packet filtering [4] [6]

Route based filtering extends ingress filtering and uses the route information to filter out spoofed IP packets. Here also this protocol uses the IP database to get the spoofed IP address or the attack IP address. If an unexpected source address appears in an IP packet on a link, then it is assumed that the source address has been spoofed, and hence the packet can be filtered. RPF uses information about the BGP routing topology to filter traffic with spoofed source addresses. In this mechanism uses the BGP messages to get information about expected and unexpected IP address.

The disadvantage of this mechanism is that if the router has recent changes or any routing information that is being changed recently or BGP message spoofing and proper selection of the IP address can bypass this. Thus this type of IP spoofing this mechanism can not detect.

### 2.2.5 History based IP filtering [4]

This protocol uses the same mechanism as the others that it uses the IP Address Database(IAD). This database contains only the IP address which are allowed inside network. Thus it doesn't have to have check for any other IP address but for the fix IP address only. Here when any packet comes in network it checks for the IP address in IAD. If the IP address is inside the database then the packet is allowed inside the network otherwise the packet will be dropped. Here the database is within the router.

But as all above this protocol has the same disadvantage as all above that proper IP selection can bypass this protocol.

### 2.2.6 Capability based method [4]

This mechanism uses request to transmit and allowed to transmit type mechanism. Source first sends request packets to its destination. Router marks (precapabilities) are added to request packet while passing through the router. If permission is granted then destination returns the capabilities, if not then it does not supply the capabilities in the returned packet. Thus all the packets are given capabilities to transit and so if any of the source that don't have this capabilities then those packets are dropped and transmitted any further inside the network.

In this system proper choosing of IP address is one of the disadvantage. Other is that all routers must have this protocol running in their system and another is that systems requires high computational complexity and space to insert the capabilities inside the packet.

### 2.2.7 Secure Overlay Service (SOS) [4]

This method uses authentication methodology to authenticate all the traffic. All traffic first sent to secure Overlay Access Point(SOAP). Authenticated traffic routed to node called beacon by consistent hash mapping. From there another node called secret servlet for further authentication. Secret servlet forwards verified traffic to the receiver.

This mechanism can protect the network from attack only until the attacker finds the way to crack the authentication and to stop that we need to keep the algorithm secret which is not possible if we want to use it for network security.

### 2.2.8 Secure Address Validity Enforcement(SAVE) [4]

SAVE protocol enables routers to update the information of expected source IP addresses on each link and block any IP packet with an unexpected source IP address. Protocol updates information rapidly but if not universally deployed then ip spoofing is possible.

As protocol needs to deploy universally which is not possible so that security of the victim is limited by that network only. The proper IP choosing is also one disadvantage of this protocol.

### 2.2.9 MultiLevel Tree for Online Packet Statistics (MULTOPS) [7]

This method uses disproportional packet rates to or from hosts and subnets as a heuristic to detect attacks. This method manages tree shaped data structure which keeps track of the packet rate from incoming and outgoing IP addresses of disproportional behavior to collect statistics.

MULTOPS stores packet statistics from inbound and outbound packets. MULTOPS works on 2 modes 1) victim oriented mode 2) source (attacker) oriented mode.

This method is anomaly based Intrusion Detection System. MULTOPS needs 2 IP stream source and destination and then it uses the anomaly data structure that keeps statistics of the IP address and then gives the result that whether or not it is attack. Here in given figure it is schematic MULTOPS in victim oriented mode.

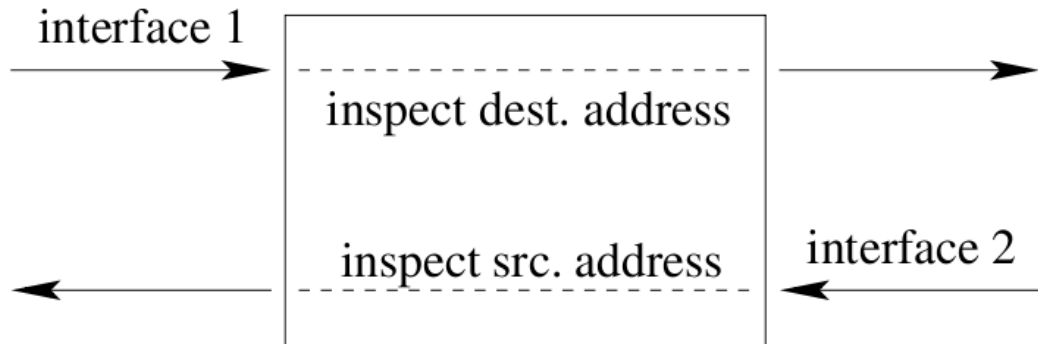


Figure 2.2: schematic MULTOPS in victim oriented mode [7]

Here advantage is also at some extend disadvantage because this method uses the anomaly based IDS it can detect some zero day attack but it can not detected spoofed IP address so if the attacker uses the user inside of the network and then attacks on victim then still attack can take place.

### 2.2.10 Shield [8]

This approach is for securing the data from source to destination. Here in this protocol all the data routes through shield router not from their real path. Thus in shield they check the packet for security and if the packet is not harmful then the packet is routed forward or the packet is dropped. Here when the source detects any attack then the route is changed the shielded router otherwise the route is the normal. In given bellow figure is of working of the shield. Here protected AS is the victim the shield is protecting.

Here the disadvantage is that if the victim can not detect the spoofed IP or the attack packets then this protocol can not be used. The other is that if victim does detect the attack and shield is invoked then the traffic at the shield will increase and

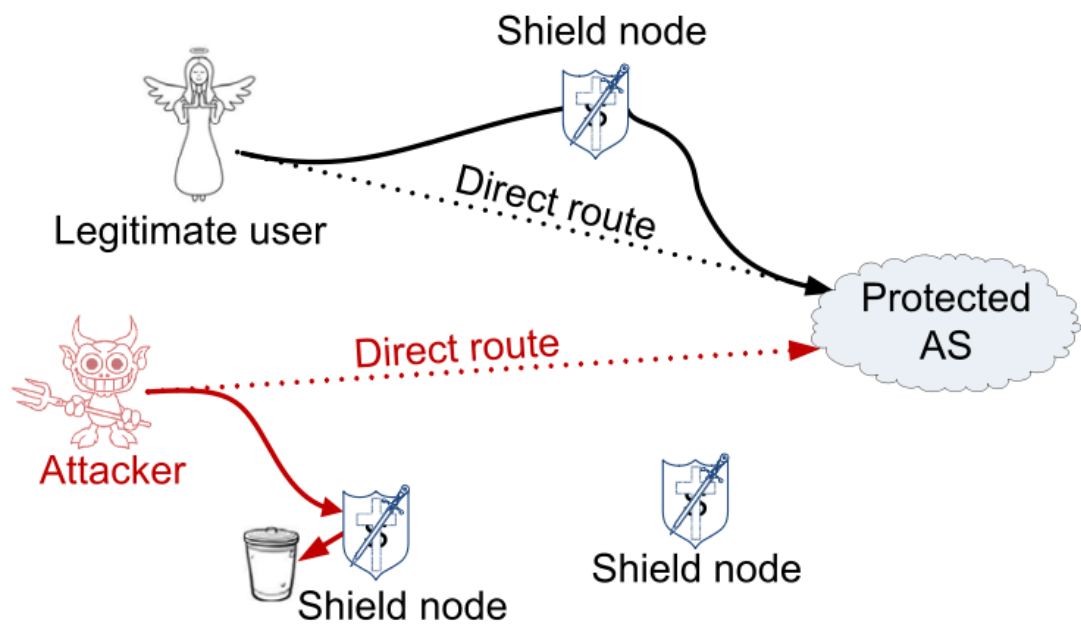


Figure 2.3: working of shield [8]

the delay will increase for the normal packets.

# Chapter 3

## Implementation Methodology

### 3.1 Literature Survey Summary

In literature survey we have seen that all the algorithms that are used in securing this attacks, either they need intrusion database or they need to coordinate with other routers for getting more information. Thus we need to have some solution such that that neither use any kind of intrusion database from other devices nor to coordinate with any other routers.

Here we propose idea of Host based Intrusion Detection System(HIDS). This IDS is signature based because the database of the malicious packets is in router/IDS and so that the router do not have to deal with any other intrusion database and do not have to coordinate with other router.

### 3.2 Implementation Tool

Here we use XORP software router as a tool to implement the HIDS in project work. XORP is an open source software router which allows us to modify protocols and also provide feature to implement new protocols in it.

This tool supports CLI based approach to configure router. Here this software router supports all command which is quite smiler to the physical router but not all

the command are as same as the physical router. This tool best supports to Ubuntu 10.04.4 with kernel 2.6.x which is most stable version of Ubuntu Linux.

### 3.3 Implementation methods

As discussed we try to implement IDS inside router and for that understanding of working router is also important. Flowchart for simple working router is given bellow.

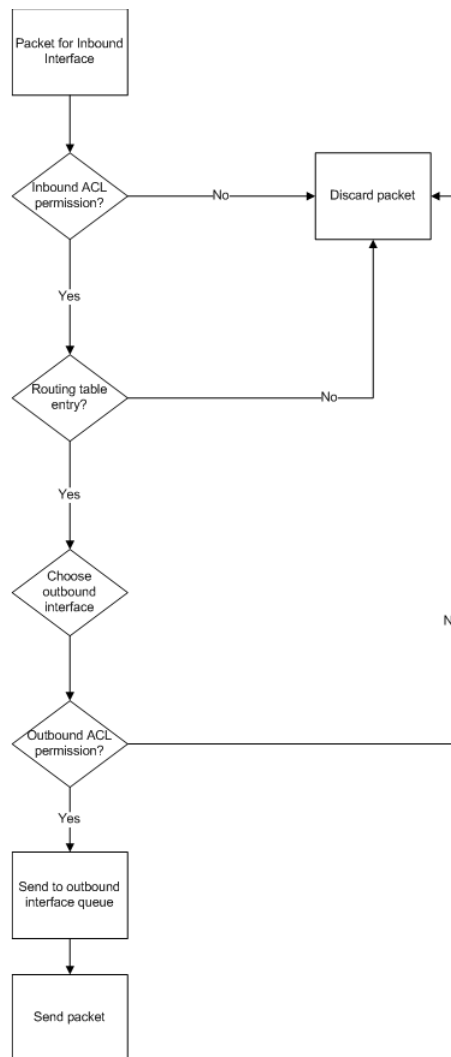


Figure 3.1: Simple router working flowchart

Here as shown in the figure first packet comes to an inbound interface of a router. First inbound ACL checks that the packet is allowed or not. If the packet is not

allowed than that packet is discarded and if it is allowed than the packet is sent forward to routing table. If the routing information is there for the packet than it is sent forward to choose outbound interface otherwise discarded. Now at the outbound interface outbound ACL also checks that this packets are allowed or not. If it allows than it sends packet to routing queue otherwise it discards the packets. Thus normal routing works with some security of ACL. Now our proposed idea of IDS is used inside the router so flowchart of the router changes at some place.

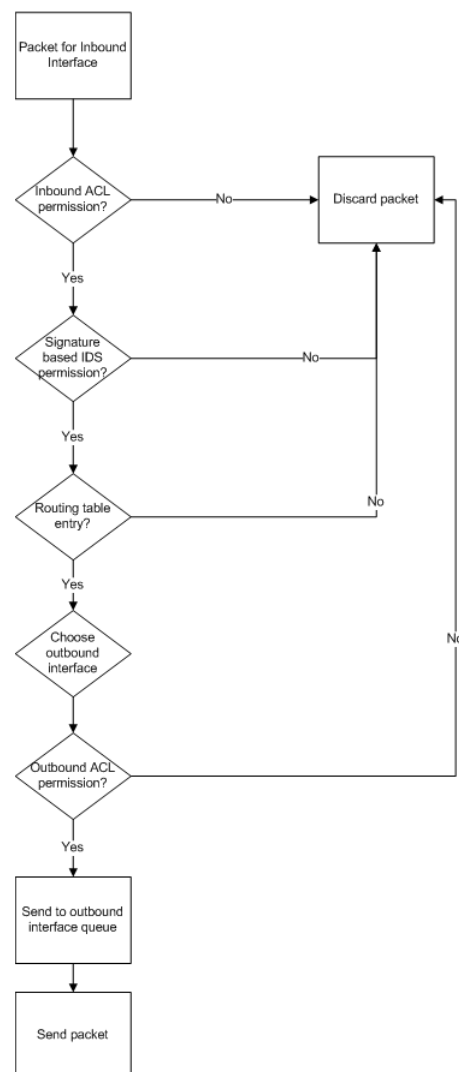


Figure 3.2: Simple router working flowchart with IDS

As shown in the second flowchart less changes is done in regard to working router.



Here we put IDS after ACL so that unnecessary checking of packet is prevented and which are already not allowed in the router. After bypassing the ACL IDS check the packet for malicious type from it's database of signature that is inside the router. If the packet signature matches the malicious type than the packet is discarded otherwise the packet is sent to the routing table.

In given bellow third figure we have a flowchart of a simple signature based IDS configured inside the router. Here first packet comes and identified inside IDS of signature database. After that it gives the result of the signature and acts accordingly. If the signature matches than the packet is malicious and it discards else it sends the packet for further processing.

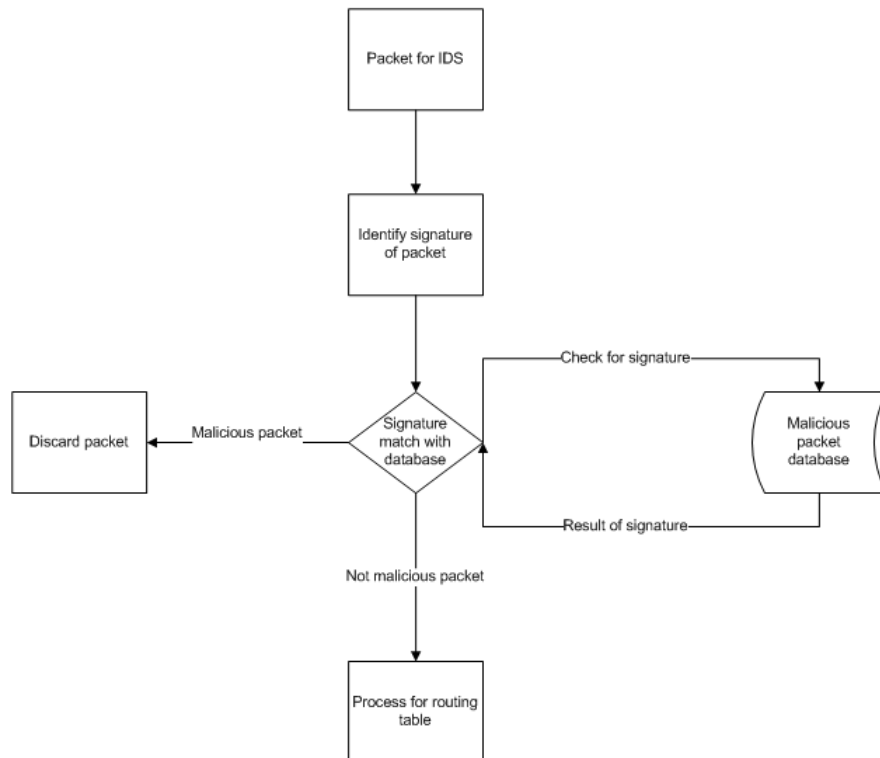


Figure 3.3: Simple IDS flowchart

## Chapter 4

# Bandwidth Starvation attack and Results

Here we performed an attack on the router in controlled environment and with systems that are configured as zombies. The attack was carried out by the systems and the results of that attack is very effective. The attack of an ICMP packet flooding is done successfully. The graph shows that how the bandwidth is utilizing and attack is carried out.

### 4.1 Packet Modification

ICMP packets are very basic packets which does not support for the protocol like TCP or UDP because ICMP does not uses any port. It is comparatively very esay to change the header of the ICMP packet because it is easy to change IP header and ICMP header. The code from which the IP and the ICMP header was changing is given bellow.

We can see the code and all the parameter of the header, can changed. As we can see that total length of the ip packet is given as the sum of the IP header and ICMP header and the custom packet size(psize). The IP Internet header length(ihl) is set to 5 because we are not using more parameters. We use IPv4 for the packet.

```

ip->tot_len = htons(sizeof(struct iphdr) + sizeof(struct icmphdr) + psize);
ip->ihl = 5;
ip->version = 4;
ip->ttl = 200;
ip->tos = 0;
ip->frag_off = 0;
ip->protocol = IPPROTO_ICMP;
ip->saddr = sin.sin_addr.s_addr;
ip->daddr = dest;
ip->check = in_chksum((u_short *)ip, sizeof(struct iphdr));
icmp->type = 8;
icmp->code = 0;
icmp->checksum = in_chksum((u_short *)icmp, sizeof(struct icmphdr) + psize);

```

Figure 4.1: IP and ICMP header changing code

We set ttl to 200 for long distance. Here ttl can be set up to 255 but we set it to 200. We set tos to 0, fragment offset to 0 and protocol to ICMP because we are using ICMP echo request to perform an attack. Here we use source address as the address of the attacker pc and destination address as the given address in the argument by the zombies.

Here we calculate the checksum by function `in_chksum` for both IP and ICMP header and we use argument ICMP type as 8 and code as 0. Thus total header size of IP is 20 and ICMP is 8 and rest is datastream of 0. The capacity of the Ethernet cable is 1500 bytes and total header length is 28 so we can append 1472 bytes of datastream.

## 4.2 Implementation

Here we can see the output of code. For this we use smurf 4.0 version from smurf program which is made to send modified ICMP echo request to given destination. We can see that in the output the arguments of the programs are given as follows. The first argument to the program is the destination IP address which is given by the attacker/zombie. The second argument is the file name from which we are getting padding bits. Third argument is number of packets. After reading number of packets, that amount of packets are sent, if 0 is passed, so infinite packets get sent until user

[illegible]

Figure 4.2: ICMP echo request flooding attack implementation

This attack was performed in the lab by using multiple systems to one system in which our router is implemented. The graphs shows bandwidth utilization before attack and after attack. The graphs is of time (X) against number of packets (Y) which are being received by the router. First graph is of bandwidth utilization during the attack. As we can see that after some time when the attack has taken place the graph goes high in number of packets and bandwidth utilization also goes very high. Here bandwidth utilization graph is in black colour.

The second graph is of red colour which is of ICMP packets. We can see here that in the beginning the graph is at 0 packets but when the packets start to flood in to

system the graph goes increasingly high and reaches to the height of the bandwidth utilization graph. Here this graph has red colour to differentiate from bandwidth utilization graph. Here X is time and Y is number of packets.

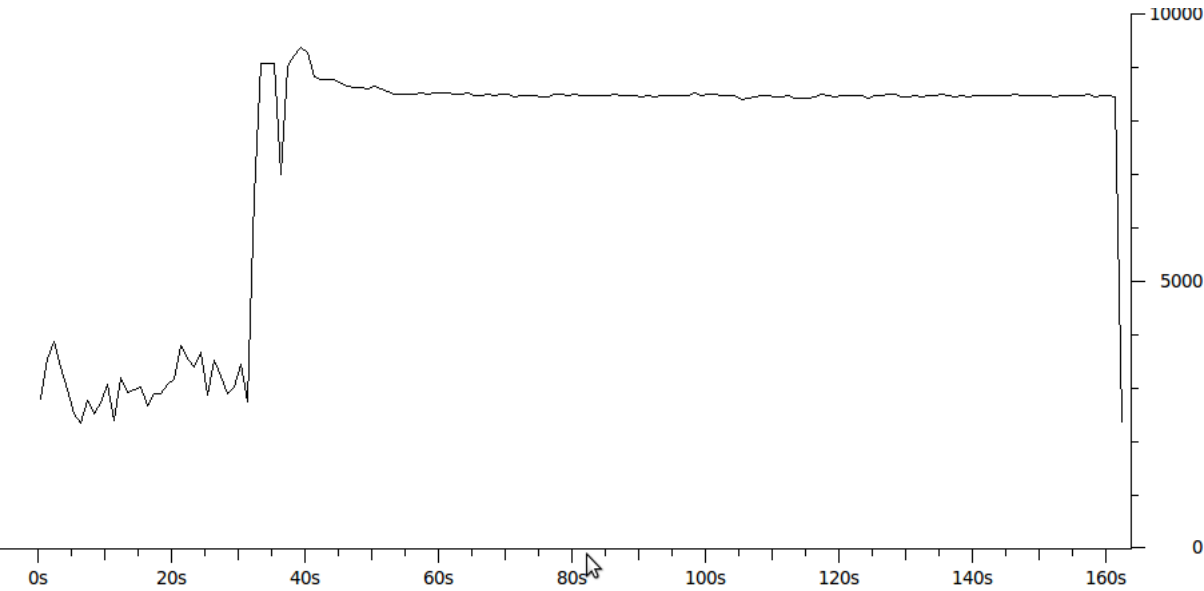


Figure 4.3: bandwidth utilization

X = time and Y = number of packets

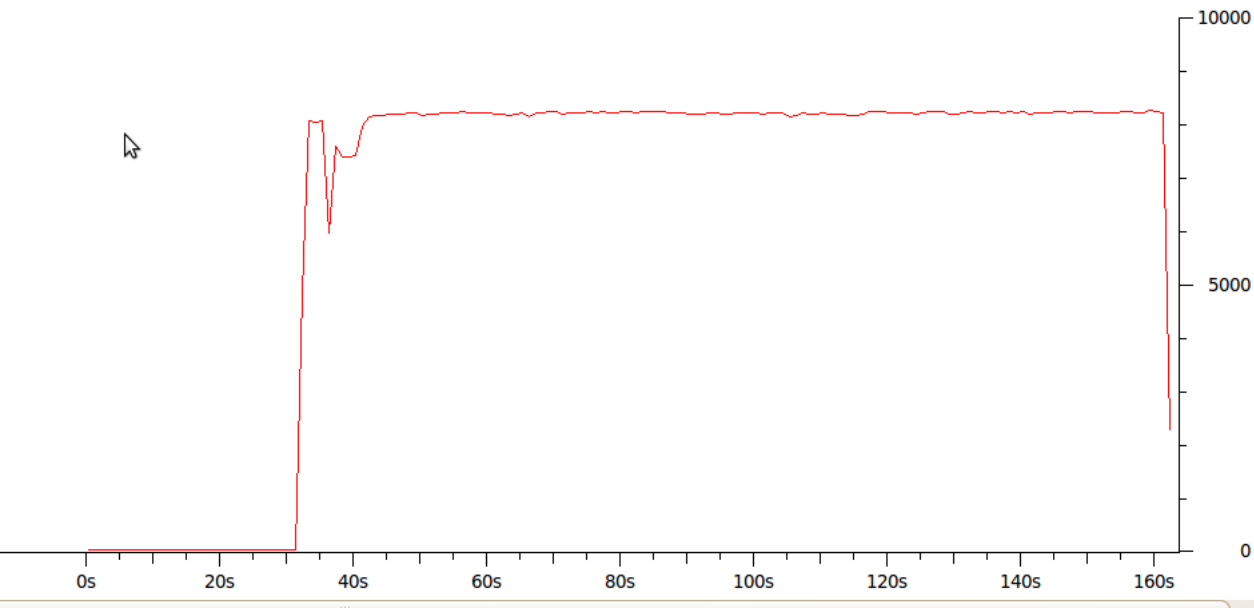


Figure 4.4: ICMP packet Flooding

As the third graph is shown it is of tcp packet data flow. As we can see that as the ICMP graph gets higher the tcp graph goes lower in graph that shows the attack was creating effectively. Router can not get tcp packets that it is receiving earlier.

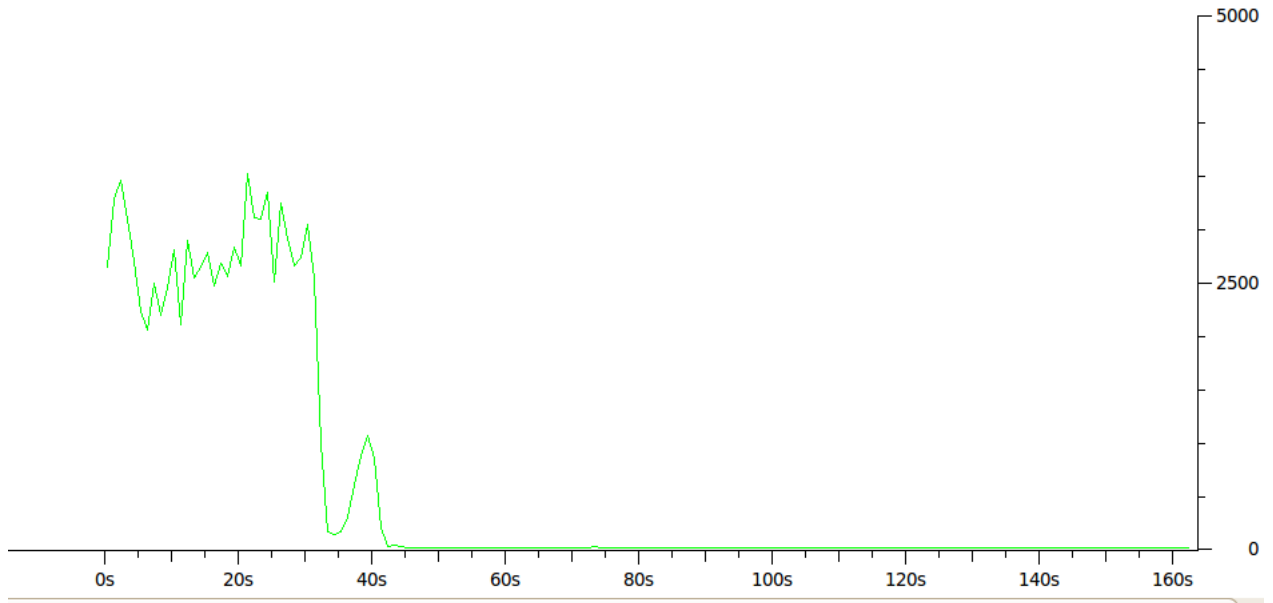


Figure 4.5: tcp packet flow

Here we have UDP packet data flow during the attack and as we can see that udp data also lower int the graph.

Combination of the different analyzed graphs are merged here in the bellow mentioned graph.

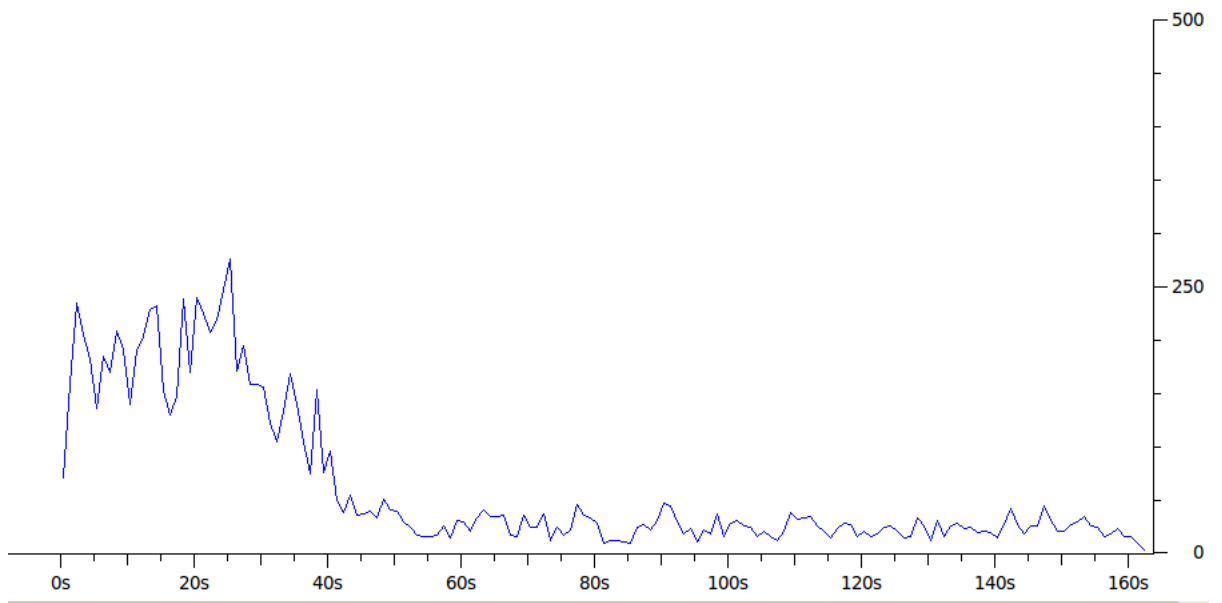


Figure 4.6: UDP packet flow

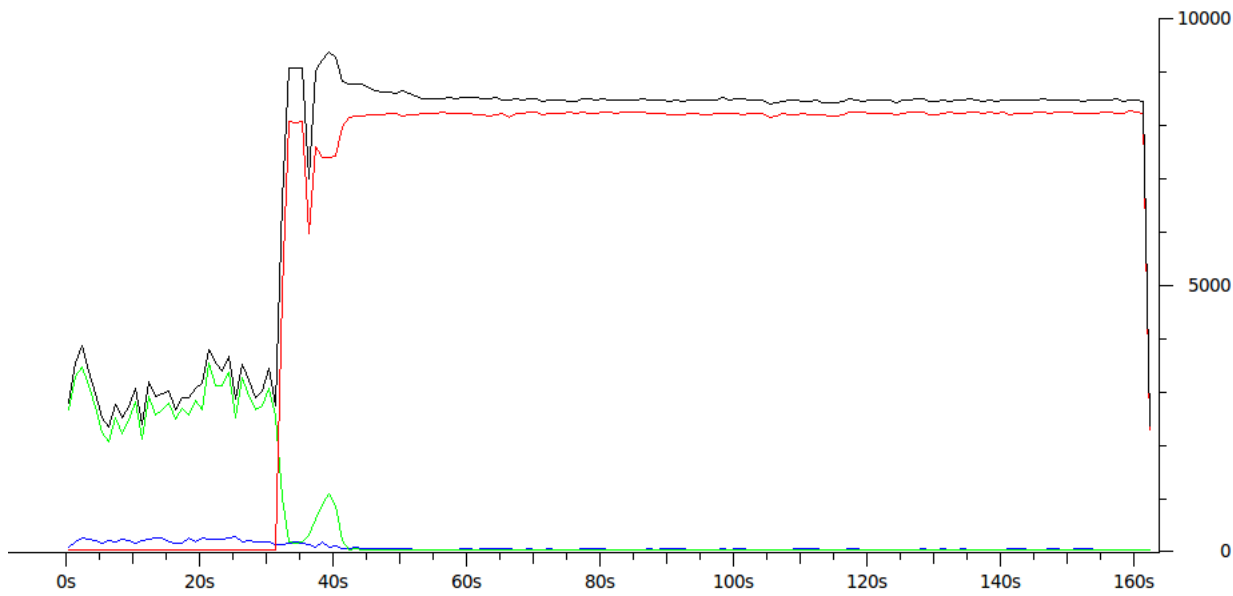


Figure 4.7: combine packet flow

# Chapter 5

## Packet Signature and Detecting attack Packets

### 5.1 Packet Signature

Signature of packet is very important to determine whether packet is attack packet or simple packet. All packets have their own signatures like ICMP packets has specific type of header and payload same as UDP packets has their own signature and as the TCP packets has their own signature. In normal scenario all packets has their own regular signature like normal ttl, normal payload size and normal flags. But at time of attack these parameters changes and with abnormal payload and flags and much more. If any attacker wants to attack any system with any of the protocol than the attacker must change these all parameters and make the packet more like of his use than the normal packets.

If attacker wants to attack with normal protocol than it is not possible but if he changes this protocol's parameters than the attack can take place. Now by doing this the packet gets some kind of unique parameter that in normal scenario that signature are not shown by any of the packet but at the same time it is not completely different than the normal packets. As this packet shows not much different signature so that



packets which has this signature can be treated as a normal packets and also allowed in side network.

## 5.2 Detecting attack packet

After knowing the signature of a packet detecting of attack packet is the next thing that any IDS must do. To detect any attack packet IDS needs to have a database of the attack packet signature which are identified as attack packet signature. Here we have one database of packet signature in which we are detecting only ICMP packets and not any other packets.

Now here in ICMP packet one signature which is very important and which is if the ICMP packet is echo request packet than maximum size of the packet is 64 bytes and if the packet is echo reply than maximum packet size is 76 bytes. As it is defined in normal scenario ICMP packet size is no more than 76 bytes but in attack scenario packet size is more than normal scenario which is working as the signature of the packet and the attack packet can be identified from all normal packets. Normally this kind of packet packet is allowed in side the network but if this packet stays in network than it can create DDoS attack on the destination of the ICMP packet.

Here in this case of ICMP we can determine whether the packet is attack packet or normal packet from looking just by its payload. But not all the packets has the same signature. TCP packets can have big size of data like 1300 or 1400 bytes so for detecting TCP attack detection other signature like fragmentation and other things needs to be checked.

Given here figure 5.1 shows that how IDS detects the attack packets from normal packets. Here first packet entered in to the IDS. After getting the packet type of packet and the signature of the packet is determined. After that IDS request to the database for signature matching. Here the database contains the attack packet signature so if the signature do not match than the packet is normal packet but if the signature do match than the packet is an attack packet and the administrator

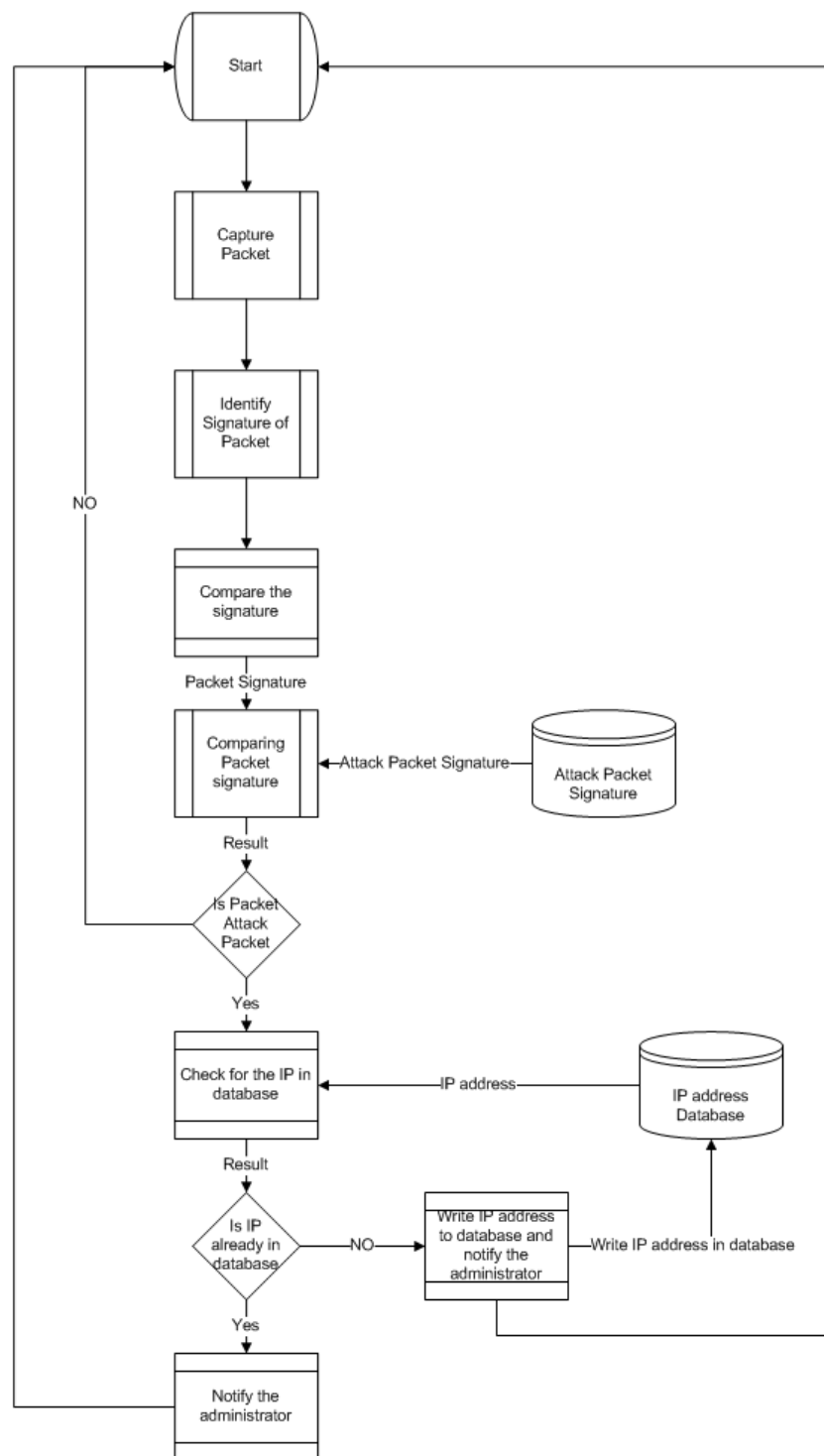
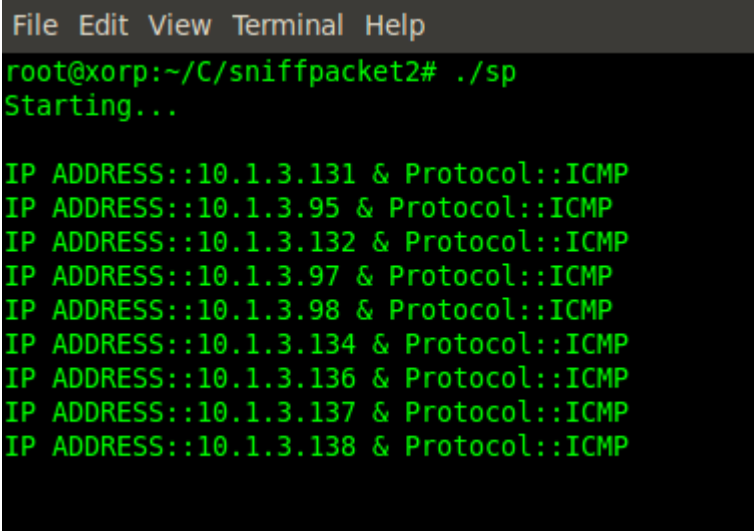


Figure 5.1: Simple IDS flowchart

is informed about the attack. Here the IDS contains the database of IP address the attack packets, so if the IP address is in the database than it do nothing but inform the administrator but if the IP address is not in the database than it writes the IP address in the database and than notify the administrator. Here IDS do not stop till it is stopped by any other person by ending it's process. In figure 5.2 result of working of IDS is given.



```
File Edit View Terminal Help
root@xorp:~/C/sniffpacket2# ./sp
Starting...

IP ADDRESS::10.1.3.131 & Protocol::ICMP
IP ADDRESS::10.1.3.95 & Protocol::ICMP
IP ADDRESS::10.1.3.132 & Protocol::ICMP
IP ADDRESS::10.1.3.97 & Protocol::ICMP
IP ADDRESS::10.1.3.98 & Protocol::ICMP
IP ADDRESS::10.1.3.134 & Protocol::ICMP
IP ADDRESS::10.1.3.136 & Protocol::ICMP
IP ADDRESS::10.1.3.137 & Protocol::ICMP
IP ADDRESS::10.1.3.138 & Protocol::ICMP
```

Figure 5.2: IDS results

### 5.3 CPU and Network performance

The CPU and Network performance is very important here. If the CPU process goes very high during attack then it is not good for router performance and if the network performance goes down then it is also a problem. But we can not control network performance because all the traffic is coming from outside not from inside. So that network performance would be low. While we can not control the network incoming traffic we can control network outgoing traffic where network performance is good.

Here in these given bellow figures CPU and Network performance at incoming traffic is given. In figure 5.3 it is normal scenario where CPU process is between 10 percent to 60 percent. This performance is normal. But in figure 5.4 CPU process is between 30 percent to 90 percent. This figure shows CPU performance during the attack scenario. As we can see that during attack CPU process goes high because all the operation of IDS takes place. But this performance can be tolerated for some time as it is not constantly very high. Here in both figure it is CPU process against time graph.

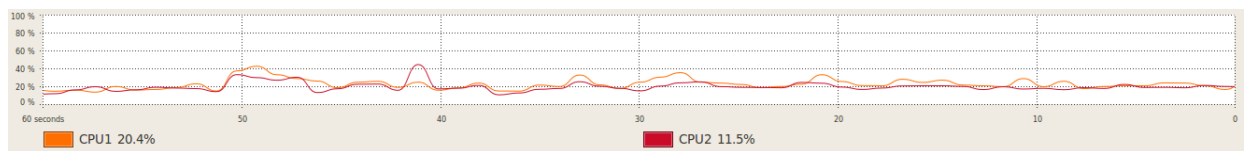


Figure 5.3: CPU in normal performance

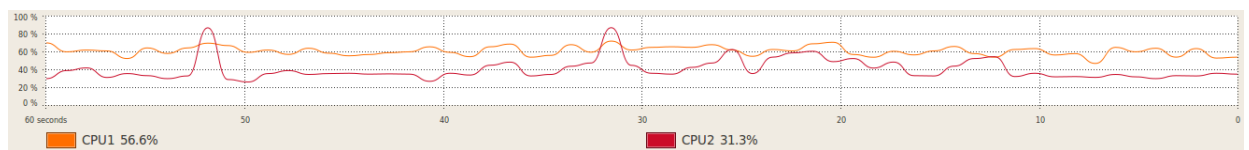


Figure 5.4: CPU in attack performance

In figure 5.5 it is of network performance against time. In this figure we can see that it is a normal scenario graph where traffic is normal and everything is fine. But in the other end in figure 5.4 it is of attack scenario graph in which network bandwidth is occupied at its pick and no other data can be transmitted to that network as there is no more bandwidth for transmitting data. Here this end is where all traffic is coming in so this end of the network can not be handled and bandwidth utilization can not be stopped because all the traffic is coming to that end of the router.

Thus this is the performance of the CPU and network during the attack and in normal scenario. Here we can see that both network and CPU performance decreases when ever attack is taking place. Here network performance is from incoming side so network performance can not be handled but the CPU performance can be improved.

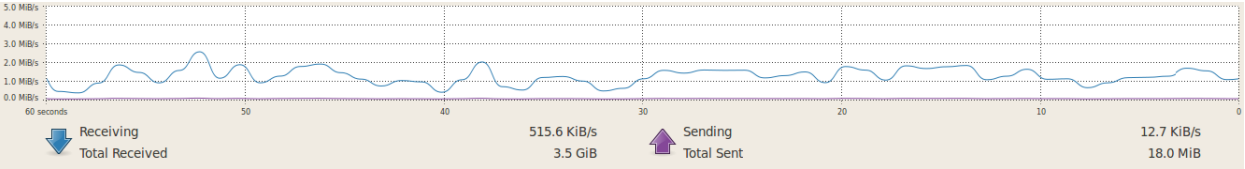


Figure 5.5: network in normal performance

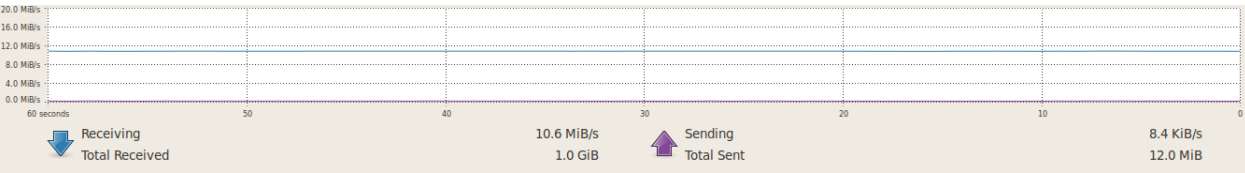


Figure 5.6: network in attack performance

# Chapter 6

## Conclusion and Future Work

### 6.1 Conclusion

Here we can conclude that when we created the ICMP flood attack the network performance is decreased. This attack sends to many attack packets to the victim that bandwidth of the victim is utilize by the attacker only and no other user can access the victim or communicate with it. The effect of this attack is very devastating.

In the other hand we created signature based IDS to detect this type of packets and generate alert message to administrator of the router. Here as we can see that CPU and network performance decreases when attack takes place with IDS. IDS detects this all packets and gives the alert message to administrator that this attack is taking place and the IP address if that attack source.

Here thus we can detect the attack from its packet signature and give the message to administrator about this attack to do as the administrator wants to do.

## 6.2 Future Work

This project is of only detecting the attack packets and give the alert message which only IDS. This project could be further more extended and can be created Intrusion Detection and Prevention System (IDPS).

Here CPU processing is also one problem that when attack is taking place the IDS needs more CPU power in detecting and giving the alert message to the user. In this project we can also create an IDS that needs less CPU processing then it is using in this project.

# Bibliography

- [1] 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," IEEE Computer Society Washington, no. 208, 1997.
- [2] B. Xiao, W. Chen, Y. He, and E. H.-M. Sha, "An active detecting method against syn flooding attack," Academic Press, Inc. Orlando, FL, USA, vol. 68, pp.56,470, Apr. 2008.
- [3] C.-H. Lin, J.-C. Liu, H.-C. Huang, and T.-C. Yang, "Using adaptive bandwidth allocation approach to defend ddos attacks," in MUE, pp. 176-181, IEEE Computer Society, 2008.
- [4] B. B. Gupta, R. C. Joshi, and M. Misra, "Distributed denial of service prevention techniques," CoRR, vol. abs/1208.3557, 2012.
- [5] A. Yaar, A. Perrig, and D. Song, "Siff: A stateless internet flow filter to mitigate ddos flooding attacks," in In IEEE Symposium on Security and Privacy, pp. 130-143, 2004.
- [6] Chang, R. K.C., "Defending Against Flooding-based Distributed Denial-of-service Attacks: A Tutorial", Comm. Mag., IEEE Press, volume 40, oct, 202
- [7] Thomer M. Gil, Massimiliano Poletto, "MULTOPS: a data-structure for bandwidth attack detection", Vrije Universiteit, Amsterdam, The Netherlands and



M.I.T., Cambridge, MA, USA,SSYM'01 Proceedings of the 10th conference on  
USENIX Security Symposium - Volume 10, 2011

- [8] Erik Kline Alexander Afanasyev Peter Reiher, "Shield: DoS Filtering Using Traffic Deflecting", Laboratory for Advanced Systems Research, UCLA Computer Science Department, 19th IEEE International Conference on Network Protocols, 2011