



**Jaypee University of Information Technology**  
**Solan (H.P.)**  
**LEARNING RESOURCE CENTER**

Acc. Num. **SP12038** Call Num:

**General Guidelines:**

- ◆ Library books should be used with great care.
- ◆ Tearing, folding, cutting of library books or making any marks on them is not permitted and shall lead to disciplinary action.
- ◆ Any defect noticed at the time of borrowing books must be brought to the library staff immediately. Otherwise the borrower may be required to replace the book by a new copy.
- ◆ The loss of LRC book(s) must be immediately brought to the notice of the Librarian in writing.

**Learning Resource Centre-JUIT**



**SP12038**



# **Prevention of IP Spoofing attack**

Project report submitted in partial fulfilment of the requirement for  
the degree of

**Bachelor of Technology**

**In**

**Computer Science and Engineering**

**Under the supervision of**

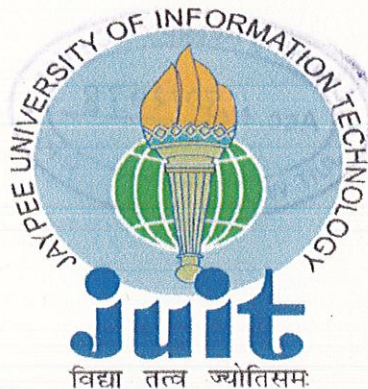
**Miss Ruhi Mahajan**

**By**

**Ankit Sharma (121250)**

**Adarsh Kaushal (121256)**

**to**



**Department of Computer Science & Engineering and Information  
Technology**

**Jaypee University of Information Technology Wagnaghat, Solan-  
173234 Himachal Pradesh**

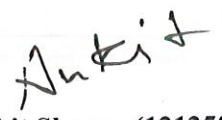


## Candidate's Declaration

I hereby declare that the work presented in this report entitled "**Prevention of IP Spoofing attack**" in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Wagnaghat is an authentic record of my own work carried out over a period from August 2015 to May 2016 under the supervision of Miss Ruhi Mahajan, Assistant Professor CSE Dept.

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

Date: 28.05.2016

  
Ankit Sharma(121250)

  
Adarsh Kaushal(121256)




## Certificate

This is to certify that project report entitled "**Prevention of IP Spoofing attack**", submitted by Ankit Sharma, 121250 and Adarsh Kaushal, 121256 in partial fulfillment for the award of degree of Bachelor of Technology in Information Technology to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Date: 28/5/16



Miss Ruhi Mahajan

Assistant Professor




## Acknowledgement

We have made a lot of efforts in completion of this project. However, it would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them.

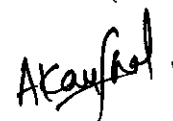
We are highly indebted to Miss Ruhi Mahajan or his guidance and constant supervision as well as providing us with all the necessary information regarding the project and also for his support in completing the project.

We would like to express our gratitude towards the faculty of Jaypee University of Information Technology, Waknaghat for their co-operation and encouragement which helped us in completion of this project.

Our thanks and appreciations also goes to every individual who have willingly helped us out in one way or the other.



Ankit Sharma (121250)



Adarsh Kaushal(121256)

## Contents

S. No.	Topic	Page No.
1.	<b>Introduction</b>	1
1.1	Introduction	1
1.2	Problem Statement	3
1.3	Objectives	10
1.4	Methodology	11
2.	<b>Literature Survey</b>	12
2.1	Overview of Research Papers/Journals	12
3.	<b>System Development</b>	23
3.1	System Design	23
3.2	Project Design	25
3.3	Algorithm	32
4.	<b>Performance Analysis</b>	35
4.1	Introduction	35
4.2	Testing	36
5.	<b>Conclusion</b>	44
5.1	Conclusion	44
5.2	Future Scope	44
5.3	Applications of IP Spoofing	44
	<b>References</b>	52



## **List of Figures**

- 1 – Simulation Scenario of IOT devices
- 2 - Simulation Scenario of IOT devices in connection
- 3 – Authentication process initiates
- 4 – Queries of Personalized aspects
- 5 – Dynamic Random key generation process starts
- 6 – Validation of key generation
- 7 – Comparison of approaches in classical and proposed way
- 8 - Satellite DSL
- 9 - NAT

## Abstract

IP Spoofing is a serious threat to the genuine use of the internet. IP spoofing attackers can overload the destination network thus preventing it from providing service to authentic user. In this project, we propose an inter domain packet filter (IDPF) architecture that can minimize the level of IP spoofing on the internet. IDPFs are constructed from the information implicit in Border Gateway Protocol (BGP) route updates and are developed in network border routers. In addition, it can help to localize the origin of an attack packet to a small number of client networks.

In computer networking, the term IP address spoofing or IP spoofing refers to the creation of Internet Protocol (IP) packets with a forged source IP address, called spoofing, with the purpose of concealing the identity of the sender or impersonating another computing system.

On January 22, 1995, in an article entitled, —New form of attack on computers linked to Internet is uncovered, John Markoff of the New York Times reported on the TCP/IP protocol suite's security weakness known as IP spoofing. The IP spoofing security weakness was published by S. M. Bellovin (1989). However, not much attention has been paid to the security weaknesses of the TCP/IP protocol by the general public. This is changing as more people and companies are connecting to the Internet to conduct business. This paper is on — “Proposed methods of IP Spoofing Detection & Prevention”. This project contains an overview of IP address and IP Spoofing and its background. It also shortly discusses various types of IP Spoofing, how they attack on communication system. This paper also describes some methods to detection and prevention methods of IP spoofing and also describes impacts on communication system by IP Spoofing. We think that our proposed methods will be very helpful to detect and stop IP spoofing and give a secured communication system.



# CHAPTER 1 –INTRODUCTION

## 1.1 Introduction

In computer networking, IP address spoofing or IP spoofing is the creation of Internet Protocol (IP) packets with a forged source IP address, with the purpose of concealing the identity of the sender or impersonating another computing system. The basic protocol for sending data over the Internet network and many other computer networks is the Internet Protocol ("IP"). The header of each IP packet contains, among other things, the numerical source and destination address of the packet. The source address is normally the address that the packet was sent from. By forging the header so it contains a different address, an attacker can make it appear that the packet was sent by a different machine. The machine that receives spoofed packets will send a response back to the forged source address, which means that this technique is mainly used when the attacker does not care about the response or the attacker has some way of guessing the response.

The Internet Protocol, or IP, is the main protocol used to route information across the Internet. The role of IP is to provide best-effort services for the delivery of information to its destination. IP depends on upper-level TCP/IP suite layers to provide accountability and reliability. The heart of IP is the IP datagram, a packet sent over the Internet in a connectionless manner. An IP datagram carries enough information about the network to get forwarded to its destination; it consists of a header followed by bytes of data. The header contains information about the type of IP datagram, how long the datagram should stay on the network (or how many hops it should be forwarded to), special flags indicating any special purpose the datagram is supposed to serve, the destination and source addresses, and several other fields.

Spoofing the source IP means replacing the source address of a packet by some other random host. It is usually (not exclusively) used in order to hide the source of this packet, to force the

target into sending network traffic in direction of the spoofed host (typical of a network traffic amplification attack like DNS amplification).

Detecting a spoofed packet can, therefore, only be done close to the source of the traffic: In a simple case, the first router on the path has the possibility to detect that the source address in the packet does not belong to any of the internal networks it knows of and therefore could drop it (that is called "egress filtering"). Unfortunately, this type of check can only be done inside or at the edge of a network and it is usually only performed by firewalls and needs to be carefully setup to avoid side effects (and therefore, not widely implemented).

Another almost identical technique, ingress filtering, tries to packet coming into the network but it must rely on some knowledge of the connected network. Typically, it only works if the filtering device knows of all the networks that are connected through it extensively (i.e. it's mostly useful between members of a peering).

Criminals have long employed the tactic of masking their true identity, from disguises to aliases to caller-id blocking. It should come as no surprise then, that criminals who conduct their nefarious activities on networks and computers should employ such techniques. IP spoofing is one of the most common forms of on-line camouflage. In IP spoofing, an attacker gains unauthorized access to a computer or a network by making it appear that a malicious message has come from a trusted machine by "spoofing" the IP address of that machine. In the subsequent pages of this report, we will examine the concepts of IP spoofing: why it is possible, how it works, what it is used for and how to defend against it.

The concept of IP spoofing was initially discussed in academic circles in the 1980's. In the April 1989 article entitled: "Security Problems in the TCP/IP Protocol Suite", author

S. M. Bellovin of AT & T Bell labs was among the first to identify IP spoofing as a real risk to computer networks. Bellovin describes how Robert Morris, creator of the now infamous Internet Worm, figured out how TCP created sequence numbers and forged a TCP packet sequence. This TCP packet included the destination address of his "victim" and using an IP spoofing attack Morris was able to obtain root access to his targeted system without a



User ID or password. Another infamous attack, Kevin Mitnick's Christmas Day crack of Tsutomu Shimomura's machine, employed the IP spoofing and TCP sequence prediction techniques. While the popularity of such cracks has decreased due to the demise of the services they exploited, spoofing can still be used and needs to be addressed by all security administrators. A common misconception is that "IP spoofing" can be used to hide your IP address while surfing the Internet, chatting on-line, sending e-mail, and so forth. This is generally not true. Forging the source IP address causes the responses to be misdirected, meaning you cannot create a normal network connection. However, IP spoofing is an integral part of many network attacks that do not need to see responses (blind spoofing).

## **1.2 Problem Statement**

There are a few variations on the types of attacks that successfully employ IP spoofing. Although some are relatively dated, others are very pertinent to current security concerns. IP-spoofing consists of several steps, which I will briefly outline here, then explain in detail. First, the target host is chosen. Next, a pattern of trust is discovered, along with a trusted host. The trusted host is then disabled, and the target's TCP sequence numbers are sampled. The trusted host is impersonated, the sequence numbers guessed, and a connection attempt is made to a service that only requires address-based authentication. If successful, the attacker executes a simple command to leave a backdoor.

### **Non-Blind Spoofing**

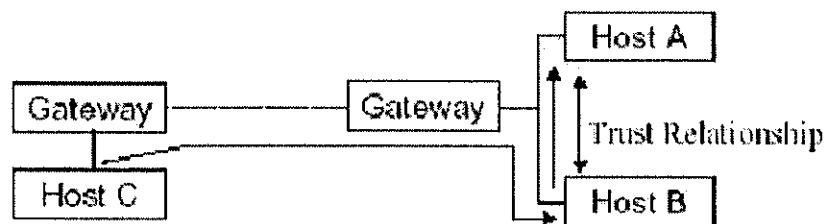
This type of attack takes place when the attacker is on the same subnet as the victim. The sequence and acknowledgement numbers can be sniffed, eliminating the potential difficulty of calculating them accurately. The biggest threat of spoofing in this instance would be session hijacking. This is accomplished by corrupting the data stream of an established

connection, then re-establishing it based on correct sequence and acknowledgement numbers with the attack machine. Using this technique, an attacker could effectively bypass any authentication measures taken place to build the connection.

### Blind Spoofing

This is a more sophisticated attack, because the sequence and acknowledgement numbers are unreachable. In order to circumvent this, several packets are sent to the target machine in order to sample sequence numbers. While not the case today, machines in the past used basic techniques for generating sequence numbers. It was relatively easy to discover the exact formula by studying packets and TCP sessions. Today, most OSes implement random sequence number generation, making it difficult to predict them accurately. If, however, the sequence number was compromised, data could be sent to the target.

Several years ago, many machines used host-based authentication services (i.e. Rlogin). A properly crafted attack could add the requisite data to a system (i.e. a new user account), blindly, enabling full access for the attacker who was impersonating a trusted host.



Usually the attacker does not have access to the reply, and abuses trust relationship between hosts. For example:



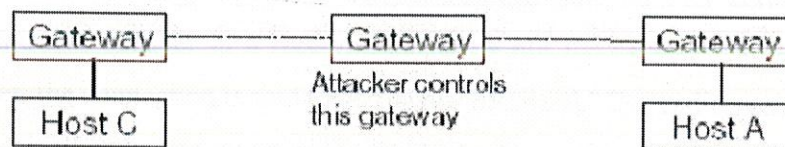
Host C sends an IP datagram with the address of some other host (Host A) as the source address to Host B. Attacked host (B) replies to the legitimate host (A).

### Man In The Middle Attack

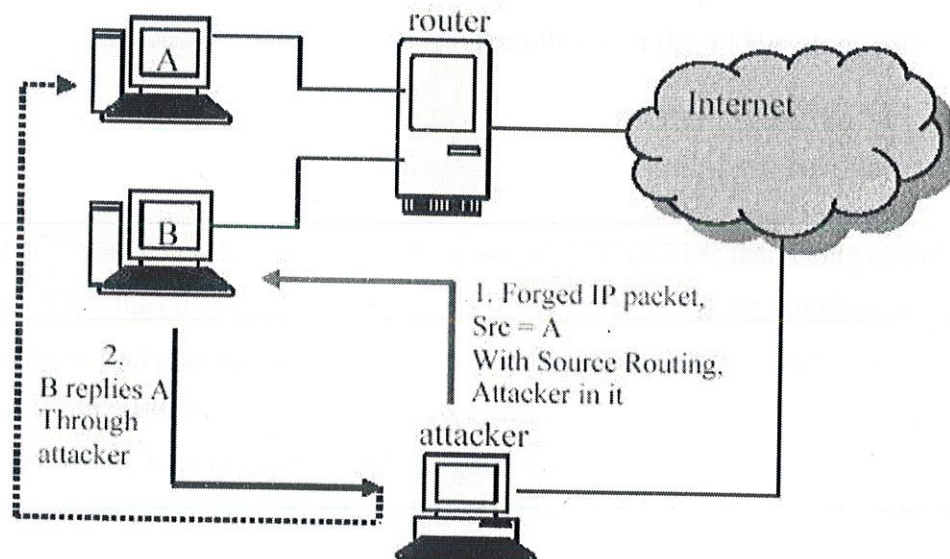
Both types of spoofing are forms of a common security violation known as a man in the middle (MITM) attack. In these attacks, a malicious party intercepts a legitimate communication between two friendly parties. The malicious host then controls the flow of communication and can eliminate or alter the information sent by one of the original participants without the knowledge of either the original sender or the recipient. In this way, an attacker can fool a victim into disclosing confidential information by “spoofing” the identity of the original sender, who is presumably trusted by the recipient.

If an attacker controls a gateway that is in the delivery route, he can

- sniff the traffic
- intercept / block / delay traffic
- modify traffic.



This is not easy in the Internet because of hop-by-hop routing, unless you control one of the backbone hosts or source routing is used. This can also be done combined with IP source routing option. IP source routing is used to specify the route in the delivery of a packet, which is independent of the normal delivery mechanisms. If the traffic can be forced through specific routes (=specific hosts), and if the reverse route is used to reply traffic, a host on the route can easily impersonate another host. The attack procedure could be:

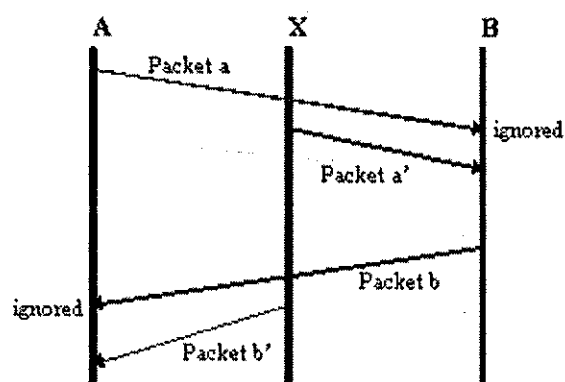


Connection hijacking exploits a "desynchronized state" in TCP communication. When the sequence number in a received packet is not the same as the expected sequence number, the connection is said to be "desynchronized." Depending on the actual value of the received sequence number, the TCP layer may either discard or buffer the packet. There is a choice, because TCP uses a sliding window protocol to allow efficient communication even in the presence of packet loss and high network latency. So, if the

received packet is not the one expected, but is within the current window, the packet will be saved on the premise that it will be expected later (various TCP mechanisms ensure that the expected packet will eventually arrive). If the received packet is outside of the current window, it will be discarded.

Thus, when two hosts are desynchronized enough, they will discard (ignore) packets from each other. An attacker can then inject forged packets with the correct sequence numbers (and potentially modify or add commands to the communication). Obviously, this requires the attacker to be located on the communication path between the two hosts so that he may eavesdrop, in order to replicate packets being sent. The key to this attack is creating the desynchronized state. Joncheray describes two possible ways to do this: one is during the three-way handshake, and the other is in the middle of an established connection.

Note that "ignored" packets may actually generate ACKs, rather than being completely ignored. When the other end receives packets with incorrect sequence numbers, it replies with an ACK packet containing the sequence number it is expecting. But the receiver of these ACK discards them, as they have the wrong sequence numbers! The receiver then sends its own ACK to notify the sender... Thus, a large number of ACKs are generated in this attack. This "signature" of the attack could be used to detect connection hijacking.





### Desynchronization during connection establishment

In this form of desynchronization, the attacker resets a connection during the three-way handshake. After host B sends the SYN&ACK packet to host A, the attacker forges new packets from B (to A) in which the connection is first closed via the RST bit, and then a new three-way handshake is initiated with A -- identical to the original, "real" handshake but with different sequence numbers. Host B now ignores messages from A (because A is using the attacker's new sequence numbers), and Host A ignores messages from B (because A is expecting messages with the attacker's sequence numbers).

The attacker then replicates new packets, with the correct sequence numbers, whenever A and B try to communicate. In doing so, the attacker may also modify the messages or inject his own.

### Desynchronization in the middle of a connection

The previous attack is limited to the initial connection. If a RST packet is sent in the middle of a connection, the connection is closed -- and the application/user is notified of this. To cause desynchronization in the middle of a connection, without closing the connection, only the sequence number counters should be altered. The Telnet protocol, in particular, provides an interesting mechanism to do this. Telnet allows special "NOP" commands to be sent. These commands do nothing, but the act of sending the bytes in the NOP command increments the expected sequence number counter on the receiver. By sending enough of these NOP commands, an attacker can cause the connection to become desynchronized. The attacker can then begin replicating new packets, with the correct sequence numbers, as before.

## Denial of Service Attack

IP spoofing is almost always used in what is currently one of the most difficult attacks to defend against – denial of service attacks, or DoS. Since crackers are concerned only with consuming bandwidth and resources, they need not worry about properly completing handshakes and transactions. Rather, they wish to flood the victim with as many packets as possible in a short amount of time. In order to prolong the effectiveness of the attack, they spoof source IP addresses to make tracing and stopping the DoS as difficult as possible. When multiple compromised hosts are participating in the attack, all sending spoofed traffic, it is very challenging to quickly block traffic.

Generally the attack is made from the root account on the attacking host against the root account on the target. If the attacker is going to all this trouble, it would be stupid not to go for root. (Since root access is needed to wage the attack, this should not be an issue.)

One often overlooked, but critical factor in IP-spoofing is the fact that the attack is blind. The attacker is going to be taking over the identity of a trusted host in order to subvert the security of the target host. The trusted host is disabled using the method described below. As far as the target knows, it is carrying on a conversation with a trusted pal. In reality, the attacker is sitting off in some dark corner of the Internet, forging packets purportedly from this trusted host while it is locked up in a denial of service battle. The IP datagrams sent with the forged IP-address reach the target fine (recall that IP is a connectionless-oriented protocol-- each datagram is sent without regard for the other end) but the datagrams the target sends back (destined for the trusted host) end up in the bit-bucket. The attacker never sees them. The intervening routers know where the datagrams are supposed to go. They are supposed to go the trusted host. As far as the network layer is concerned, this is where they originally came from, and this is where responses should go. Of course once the datagrams are routed there, and the information is demultiplexed up the protocol stack, and reaches TCP, it is discarded (the trusted host's TCP cannot respond-- see below). So the attacker has to be smart and \*know\* what was sent, and \*know\* what response the server is looking for.



The attacker cannot see what the target host sends, but she can *\*predict\** what it will send; that coupled with the knowledge of what it *\*will\** send, allows the attacker to work around this blindness.

After a target is chosen the attacker must determine the patterns of trust (for the sake of argument, we are going to assume the target host *\*does\** in fact trust somebody. If it didn't, the attack would end here). Figuring out who a host trusts may or may not be easy. A 'showmount -e' may show where file systems are exported, and rpcinfo can give out valuable information as well. If enough background information is known about the host, it should not be too difficult. If all else fails, trying neighboring IP addresses in a brute force effort may be a viable option.

Once the trusted host is found, it must be disabled. Since the attacker is going to impersonate it, she must make sure this host cannot receive any network traffic and foul things up. There are many ways of doing this, the one I am going to discuss is TCP SYN flooding.

### 1.3 Objectives

Spoofing an IP address has very limited use anyway. When you send data to a server from a spoofed IP address, you never see the response. Using a spoofed IP address, you'll never be able to establish any connection that requires any kind of handshake, including TCP and VPN connections.

About the best use of a spoofed IP address is to perform some kind of denial of service (DOS) attack, hiding your actual attack origin. You could perform a SYN flood, or you could flood UDP services such as DNS. However, you couldn't use it to post to a web server because HTTP requires a TCP connection.



## 1.4 Methodology

### Packet filtering

The router that connects a network to another network is known as a border router. One way to mitigate the threat of IP spoofing is by inspecting packets when they leave and enter a network looking for invalid source IP addresses. If this type of filtering were performed on all border routers, IP address spoofing would be greatly reduced. Egress filtering checks the source IP address of packets to ensure they come from a valid IP address range within the internal network. When the router receives a packet that contains an invalid source address, the packet is simply discarded and does not leave the network boundary. Ingress filtering checks the source IP address of packets that enter the network to ensure they do not come from sources that are not permitted to access the network. At a minimum, all private, reserved, and internal IP addresses should be discarded by the router and not allowed to enter the network. In Linux, packet filtering can be enabled using:

```
echo 2 > /proc/sys/net/ipv4/conf/*/rp_filter
```

### Limits of packet filtering

Packet filtering normally may not prevent a system from participating in an attack if the spoofed IP address used could fall within the valid internal address range. However it will simplify the process of tracing the packets, since the systems will have to use a source IP address within the valid IP range of the network.

## CHAPTER 2 – LITERATURE SURVEY

### 2.1 Overview of Research Papers/Journals

**Title-** Controlling IP Spoofing Through Inter Domain Packet Filters

**Author(s)-** Zhenhai Duan and Jaideep Chandrashekar

**Date of Conference-** February 2006

**Published In-** IEEE

**Location -** Japan

Distributed Denial of Service (DDoS) attacks pose an increasingly grave threat to the Internet, as evidenced by recent DDoS attacks mounted on both popular Internet sites and the Internet infrastructure [1]. Alarminglly, DDoS attacks are observed on a daily basis on most of the large backbone networks. One of the factors that complicate the mechanisms for policing such attacks is IP spoofing, the act of forging the source addresses in IP packets. By masquerading as a different host, an attacker can hide its true identity and location, rendering source-based packet filtering less effective. It has been shown that a large part of the Internet is vulnerable to IP spoofing. Recently, attackers are increasingly staging attacks via botnets. In this case, since the attacks are carried out through intermediaries, i.e., the compromised .bots, attackers may not utilize the technique of IP spoofing to hide their true identities. It is tempting to believe that the use of IP spoofing is less of a factor. However, recent studies, show that IP spoofing is still a common phenomenon: it is used in many attacks, including the high-profile DDoS attacks on root DNS servers in early February 2006.

In the response to this event, the ICANN security and stability advisory committee made three recommendations [1]. The first and long-term recommendation is to adopt source IP address verification, which confirms the importance of the IP spoofing problem. It is our contention that IP spoofing will remain popular for a number of reasons. First, IP spoofing makes it harder



to isolate attack traffic from legitimate traffic. Packets with spoofed source addresses may appear to be from all around the Internet. Second, it presents the attacker with an easy way to insert a level of indirection. As a consequence, substantial effort is required to localize the source of the attack traffic. Finally, many popular attacks, such as man-in-the-middle attacks, reflector-based attacks [10], and TCP SYN flood attacks [11], use IP spoofing and require the ability to forge source addresses. Although attackers can insert arbitrary source addresses into IP packets, they cannot, however, control the actual paths that the packets take to the destination. Based on this observation, Park and Lee [12] proposed the route-based packet filters as a way to mitigate IP spoofing. The idea is that, assuming singlepath routing, there is exactly one single path  $p(s; d)$  between source node  $s$  and destination node  $d$ . Hence, any packets with source address  $s$  and destination address  $d$  that appear to pass through a router not in  $p(s; d)$  should be discarded. The challenge is that constructing such a route-based packet filter requires the knowledge of global routing information, which is hard to reconcile on the current Internet routing infrastructure. The current Internet consists of thousands of autonomous systems (ASes), each of which is a logical collection of networks with common administrative control. Each AS communicates with its neighbors using the Border Gateway Protocol (BGP), the de-facto inter-domain routing protocol, to exchange information about its own networks and others that it can reach [13].

In this paper we proposed and studied an inter-domain packet filter (IDPF) architecture as an effective countermeasure to the IP spoofing-based DDoS attacks. IDPFs rely on BGP update messages exchanged on the Internet to infer the validity of source address of a packet forwarded by a neighbor. We showed that IDPFs can be easily deployed on the current BGP based Internet routing architecture. We studied the conditions under which the IDPF framework can work correctly without discarding any valid packets. Our simulation results showed that, even with partial deployment on the Internet, IDPFs can significantly limit the spoofing capability of attackers. Moreover, they also help pinpoint the true origin of an attack packet to be within a small number of candidate networks, therefore, simplifying the reactive IP traceback process.



**Title –** Detecting and Preventing IP Spoofed Attack By Hashed Encryption

**Author(s)-** Vimal Upadhyay and Rajeev kumar

**Date of Conference –** 2 July 2011

**Published In -** International Journal of Enterprise Computing and Business Systems

**Location -** Margaret Engineering College Neemrana

Network introduces security problems, threats, risks and other type of attacks like internal and external attacks. Wireless networks are a new type of networked systems which comprise of nodes with the physical environment and collaborate among each other to provide data to end-users. These nodes are small devices that have limited processing, communication and memory. They are placed in the environment for long periods without any assistance. This technology has a lot of potential in the areas of military, health, environmental monitoring etc. As WNs are a classification of networks, therefore, most of attacks that are applicable on networks tend to apply on the WNs. Hence Security is a difficult problem in WN. And the resource-starved nature of wireless networks poses great challenges for security. The first challenges of security in sensor network lie in the conflicting interest between minimizing resource consumption and maximizing security. Secondly the capabilities and constraints of sensor node hardware will influence the type of security mechanisms that can be hosted on a sensor node platform. Energy in the security realm is key establishment. Attacks on a WN can come from all direction and target at any node. Damage can include leaking secret information, interfering messages and impersonating nodes, thus violating the above security goals. In this paper we have explored general security threats in wireless network and made an extensive study to categorize available data gathering protocols and analyze possible security threats on them.

for openness and scalability without much concern for security, malicious users can exploit the design weaknesses of the internet to wreak havoc in its operation. Incidents of disruptive activities like e-mail viruses, computer worms and denial-of service attacks have been on the rise reports an increase of such incidents from 252 in 1990 to 137,529 in 2003). The incidents which have raised the most concern in recent years are the denial-of-service (DoS) attacks

whose sole purpose is to reduce or eliminate the availability of a service provided over the Internet, to its legitimate users.

the IP protocol include the IP address of the sending host. The recipient directs replies to the sender using this source address. However, the correctness of this address is not verified by the protocol. The IP protocol specifies no method for validating the authenticity of the packet's source. This implies that an attacker could forge the source address to be any he desires. This is a well-known problem and has been well described. In all but a few rare cases, sending spoofed packets is done for illegitimate purposes.

this paper have designed a low-cost and efficient scheme called Hemdadf, for defending against IP spoofed attacks, The Hemdadf scheme is composed of three parts: marking process, filtering process, secure transmission. The marking process requires the participation of routers in the Internet to encode path information into packets. We suggest the use of a hash function and secret key to reduce collisions among packet-markings. The scheme also includes mechanisms for detecting and reporting spoofing in a timely manner. The evaluation of the scheme under simulations would be shown that my scheme can effectively and efficiently differentiate between good and bad packets under spoofed attack. Most good packets are accepted even under the most severe attack, whose traffic is about 10 times of normal traffic. At the same time, the bad packet acceptance ratio is maintained at a low level. This scheme can be performs well even under massively IP spoofed attacks involving up to 5000 attackers. HEMDADF scheme detected the occurrence of attack precisely within 3 - 4 seconds. The quick detection is valuable to the victim so that appropriate actions can be taken to minimize the damage caused by a IP spoofed attack.



**Title-** Controlling IP Spoofing Through Packet Filtering

**Author(s) -** Yogesh Singh and Hariom Awasthi

**Date of Conference-** March 2014

**Published In-** NIET NIMS UNIVERSTY

**Location -** Jaipur

The switch that associates a system to an alternate system is known as a border switch. One approach to relieve the risk of IP spoofing is by assessing packets when they leave and enter a system searching for invalid source IP addresses. On the off chance that this kind of sifting was performed on all border switches, IP location spoofing might be significantly diminished. At any rate, all private, held, and inner IP locations ought to be tossed by the switch and not permitted to enter the system .

**Filtering Utilizing by IDPF-** To utilize the filtering by IDPF an immediate association with the Internet is needed. To begin with verify just has on your interior LAN can take an interest in trust-connections. At that point basically channel out all activity from the outside the Internet that implies to hail from within the LAN. Implementing entrance and departure shifting on border switches is an incredible spot to begin internet protection. It needed to actualize an ACL (access control rundown) that protects private IP addresses on downstream interface. On the upstream interface, we ought to limit source addresses outside of authentic extent, which will keep somebody on system from sending secured activity to the Internet.

**Blowfish Algorithm-** Existing Blowfish Algorithm Blowfish Algorithm is utilize for encryption and decryption. It takes a variable-length key, from 32 bits to 448 bits, making it perfect for securing information . Blowfish Algorithm is a Mobile Network, repeating a basic encryption work 16 times. The frame size is 64 bits, and the key might be any length up to 448 bits. It is altogether speedier than most encryption calculations when executed on 32-bit microchips with huge information stores. All operations are XOR and increases on 32-bit words.



**Subkeys-** Blowfish utilizes an expansive number of subkeys. These keys must be for every registered before any information coded or decoded. Each round consists of a key-based P (permutation) and it operates data encryption via 16 round F (feistelnetwork). Length of all variable like L and R ranges from 32 bits to 448 bits.

The input is a 64-bit data element, x. Divide x into two 32-bit halves: xL, xR.

Then, for i = 1 to 16:

$$xL = xL \text{ XOR } P_i$$

$$xR = F(xL) \text{ XOR } xR$$

Swap xL and xR

After the sixteenth round, swap xL and xR again to undo the last swap. Finally, recombine xL and xR to get the ciphertext.

In this paper, Detection and Removal of IP Spoofing by simulation in blowfish algorithm using extended Key Round System proposed to controls the key swapping based DDOS attacks in order to control encryption in a viable way. The development relies on BGP upgrades and this channel structure superbly works without disposing of any packets with genuine source IP address

**Title-** A Compressed Anti-IP Spoofing Mechanism using Cryptography

**Author-** S.Gavaskar, Dr.E.Ramaraj

**Date of Conference-** 4 July 2009

**Published in –** Madurai Kamraj University

**Location -** Madurai

**Compression:** Basically compression classified into two types

- **Lossy Compression** - In Computer terminology, lossy compression is a data encryption method which eliminates some of the data, in order to achieve its goal, with the result that decompressing the data yields content that is different from the original, though similar enough to be useful in some way. Lossy compression is most commonly used to compress multimedia data, audio, video, image, etc. lossless compression is required for text and data files, such as bank records, text articles, etc. In many cases it is advantageous to make a master lossless file which can then be used to produce compressed files for different purposes. We can compress many formats of digital data through that we can minimize the size of a computer file needed to store it. According to the networks the effective utilization of bandwidth needed to stream it, with no loss of the full information contained in the original file. A picture is converted to a digital file by considering it to be an array of dots, and specifying the color and brightness of each dot. If the picture contains an area of the same color, it can be compressed without loss by saying 200 red dots instead of red dot, red dot, etc red dot. The original contains a certain amount of information; there is a lower limit to the size of file that can carry all the information. For example, most people know that WinRar produce the compressed ZIP file is smaller than the original file; but repeatedly compressing the file will not reduce the size to nothing, and will in fact usually increase the size.

Lossy compression formats suffer from generation loss: repeatedly compressing and decompressing the file will cause it to progressively lose quality. This is in contrast with lossless data compression. Information-theoretical foundations for lossy data compression are

provided by rate distortion theory. Much like the use of probability in optimal coding theory, rate distortion theory heavily draws on Bayesian estimation and decision theory in order to model perceptual distortion and even aesthetic judgment.

- **Lossless Compression** -Lossless data compression is a kind of data compression algorithms that allows the exact original data to be fetched from the compressed ZIP data. The term lossless is in contrast to lossy data compression, which only allows an approximation of the original data to be re fetched, in exchange for better compression rates. Lossless data compression is used in many applications. For example, it is used in the popular ZIP file format and in the kernel OS UNIX tool gzip. It is also often used as a component within lossy data compression technologies

Lossless compression algorithms and their implementations are routinely tested in head-to-head existing methods. There are a number of better-known compression existing methods. Some existing methods cover only the compression ratio, so winners in this benchmark may be unsuitable for everyday use due to the slow speed of the top performers. Another drawback of some existing methods is that their data files are known, so some program writers may optimize their programs for best performance on a particular data set. The winners on these existing methods often come from the class of context-mixing compression software.

### **C. Cryptography:**

Cryptography has been used as a way to send secret messages between warring nations, between users, between organizations etc; as such, it became an important issue in national security and laws. With the increasing need for secure transactions for data traversing computer networks for medical, financial, and other critical applications, cryptography is now becoming a necessity for nongovernmental, nonmilitary applications. All over the globe, the laws and regulations concerning cryptography are undergoing a vast change. Legal restrictions on the import and export of cryptographic products are being debated and modified.

Cryptography has some major issues:

**Key length:** The combination of the algorithm and the key length are factors of cryptographic strength. The algorithm is usually well known. The longer key is the stronger the cryptographic



strength of a given algorithm. Some countries have export laws that limit the key length of a given cryptographic algorithm.

Key recovery: In recent years, export laws have been modified if the cryptographic algorithm includes the capability of incorporating key recovery methods. These modified laws enable governments to wire-tap for encrypted electronic data if they deem it necessary to do so.

Cryptography use: A distinction is sometimes made about whether cryptography is used for authentication and integrity purposes or for confidentiality purposes. When used for confidentiality, the export laws are typically much more stringent.

In this paper cryptography uses to enhance the security in IP compression technique In our research we tried to implement a new method in TCP/IP packet transaction. It's our faith it will increase the performance of data transformation. This method effectively improves the bandwidth utilizations and also reduces the traffic of overall network due to small size of packet. The overall performance of the network will increase while implementing compression with cryptography technique. Cryptography technique reduces the hacker intrusion and stealing of data theft. It takes control over the IP spoofing hackers.

**Title-** A Dynamic Method to Detect IP Spoofing on Data Network  
**Author(s)-** Dr.C.Venkatesh  
**Date of Conference-** 21 December 2004  
**Published In -** Anna University  
**Conference Location –** Tamil Nadu

### **Spoofed Packets Detection Methods**

A variety of methods are deployed in determining whether a received packet has spoofed source IP address or not. In Internet, when a node receiving a packet can determine whether the packet is spoofed by either an active or passive ways. The term active mean the host must perform some network action but the passive method doesn't require such action. However an active method may be used to validate cases where the passive method indicates the packet was spoofed. Among different methods this paper considers both IP trace back and hop count based detection method.

### **Traceback Techniques**

Since the late 1999 research on IP trace back has been active to detection of DDOS attacks. Several approaches have been proposed to trace IP packets to their origins. IP trace back is usually performed at the network layer, with the help of routers and gateways. The trace back techniques can trace packet paths and help in identifying the perpetrators of the DoS attacks with a high probability. These can be useful forensic tools in law enforcement but do nothing to prevent the occurrence of IP spoofing. Among the spoofing prevention techniques, many focus on shielding the destination from IP spoofing. Their shortcoming lies in the observation that they fail to protect the Internet routing fabric from being misused in forwarding spoofed packets. The rest of the spoofing prevention techniques possess the ideal goal of preventing spoofing near its source



**TTL Methods-** When IP packets are routed across the Internet, the time-to-live (TTL) field is decremented. This field in the IP packet header is used to prevent packets from being routed endlessly when the destination host cannot be located in a fixed number of hops. It is also used by some networked devices to prevent packets from being sent beyond a host's network subnet. The TTL is a useful value for detecting spoofed packets. Its use is based on several assumptions, which, from our network observations, appear to be true. When a packet is sent between two hosts, as long as the same route is taken, the number of hops will be the same. This means that the initial TTL will be decremented by the same amount. Packets sent near in time to each other will take the same route to the destination. Routes change infrequently. When routes change, they do not result in a significant change in the number of hops [12]. The objective of this work is to use both the concepts of trace back and hop count of the packet while routing from source to destination on internet. The trace back approach is used to finding out the origin of the spoofing attack using the existing traffic flow information. Furthermore, to strengthen the spoofing prevention hop count value of the packet between the source and destination are also validated. An ant-based trace back algorithm is using for finding the traffic flow information as the trace for ants finding the attack path. The hop-count information is indirectly reflected in the TTL field of the IP header, since each intermediate router decrements the TTL value by one before forwarding a packet to the next hop. The difference between the initial TTL (at the source) and the final TTL value (at the destination) is the hop-count between the source and the destination.

In this paper, The proposed IP traceback analysis method is an extended of Ant algorithm. This method examines all possible way to reach destination node during it learning stage. Among the different path to reach the destination, it is understood that the legitimate request might prepare shortest path. This shortest path is identified using the pheromone intensity. To strengthen the spoofing identification an additional metric of hop count value is also considered. Thus the legitimate request is validated and permitted to access the destination node based on the metrics hop count and flow level. The simulations results show that this approach discards almost 90% of spoofed IP request.

## Chapter 3 – SYSTEM DEVELOPMENT

### 3.1 System Design

Spoofing can take on many forms in the computer world, all of which involve some type false representation of information. There are a variety of methods and types of spoofing.

- IP Spoofing
- ARP Spoofing
- E-Mail Spoofing
- Web Spoofing
- DNS Spoofing

There are no legal or constructive uses for implementing spoofing of any type. Some of the outcomes might be sport, theft, vindication or some other malicious goal. The gravity of these attacks can be very severe, can cost us millions of dollars and should not be overlooked by the Internet security community.

#### II. IP SPOOFING

IP spoofing is used to gain unauthorized access to a computer. The attacker forwards packets to a computer with a source address indicating that the packet is coming from a trusted port or system. Attackers must go through some complicated steps to accomplish the task [1]. They must:

- Obtain a target.
- Obtain an IP address of a trusted machine.
- Disable communication of the trusted machine (e.g.SYN flooding).
- Sample a communication between the target and trusted hosts
- Guess the sequence numbers of the trusted machine.
- Modify the packet headers so that it appears that the packets are coming from the trusted host.



Attempt connection to an address authenticated service or port.

If successful, the attacker will plant some kind of backdoor access for future reference

System A impersonates system B by sending B's address instead of its own. The reason for doing this is that systems tend to function within groups of other "trusted" systems. This trust is implemented in a one-to-one fashion; system A trusts system B. IP spoofing occurs in the following manner: if system A trusts system B and system C spoofs system B, then system C can gain otherwise denied access to system A. This is all made possible by means of IP address authentication, and if the packets are coming from external sources- poorly configured routers

Use cryptographic signatures to exchange authenticated email messages. Authenticated email provides a mechanism for ensuring that messages are from whom they appear to be, as well as ensuring that the message has not been altered in transit. Similarly, sites may wish to consider enabling SSL/TLS in their mail transfer software. Using certificates in this manner increases the amount of authentication performed when sending mail

Following methods that have been used in the project

Configure your mail delivery daemon to prevent someone from directly connecting to your SMTP port to send spoofed email to other sites

Ensure that your mail delivery daemon allows logging and is configured to provide sufficient logging to assist you in tracking the origin of spoofed email. Consider a single point of entry for email to your site

can implement this by configuring your firewall so that SMTP connections from outside your firewall must go through a central mail hub. This will provide you with centralized logging, which may assist in detecting the origin of mail spoofing attempts to your site

"Educate your users about your site's policies and procedures in order to prevent them from being social engineered," or tricked, into disclosing sensitive information (such as passwords). Have your users report any such activities to the appropriate

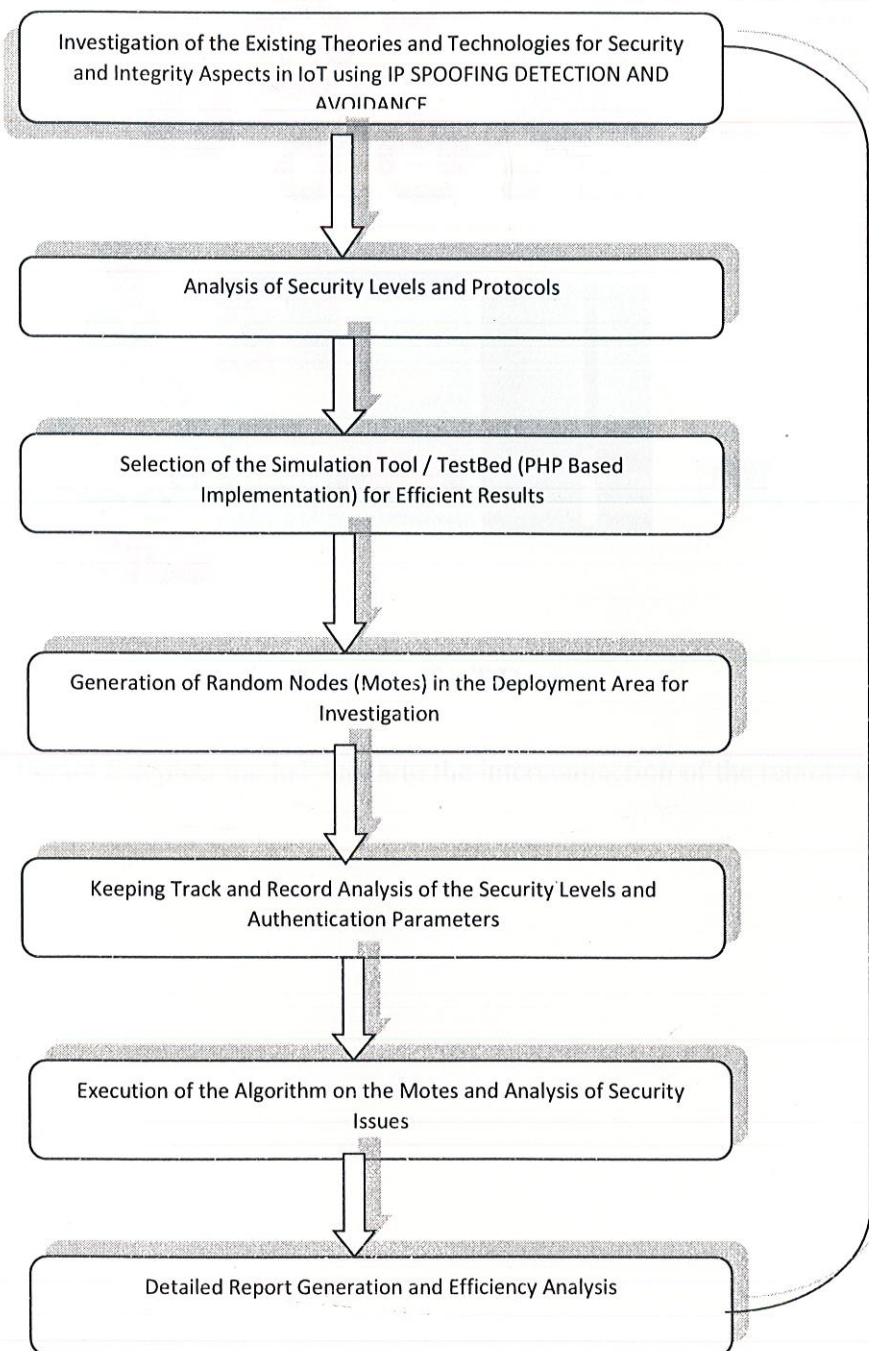
The main objective of IP compression is to avoid the overhead, which provides the bandwidth utilization. The IP header compression work initiated ten years ago but still there is some drawback and problem persists. For handling the packet transformation in effective manner we are moving to IPv6 but the header size will increase in IPv6. To increase the bandwidth utilizations, avoid the network traffic, congestion, collision, we go for compression technique. Basically compression used for minimize the size of file into half. For example if the original file size is 100mb after compression it will reduced into 50mb. While decompress your file we have to get original information without loose anything. Basic idea behind in this is remove the unwanted data's or information's.

### **3.2 Project Design**

#### **Flow of the Proposed Model and Algorithmic Approach/ Pseudo Code**

1. Remote Devices Node Matrix Activation for Data Transmission
2. Source Node Activation
3. Remote Login and RREQ by the Source Node
4. RREP by the nearby nodes for the generation and identification of path.
5. Query and Response of the Basic Questions Set
6. For Security Implementation, generate a Dynamic Key
7. Mix the Key with a set of Fuzzy Random Noise.
8. Malicious Node attempts to access the key
9. Data Packet travels through the path
10. Malicious node not able to find the actual key and failure attempts found.
11. If (Found/Match Dynamic Key) => Success => HandOver the DataPacket
12. else Terminate with the Error of Authentication and Dynamic Key Authentication Failure
13. GoTo Step 1







## Implementation of Security and Integrity in Internet of Things (IoT)

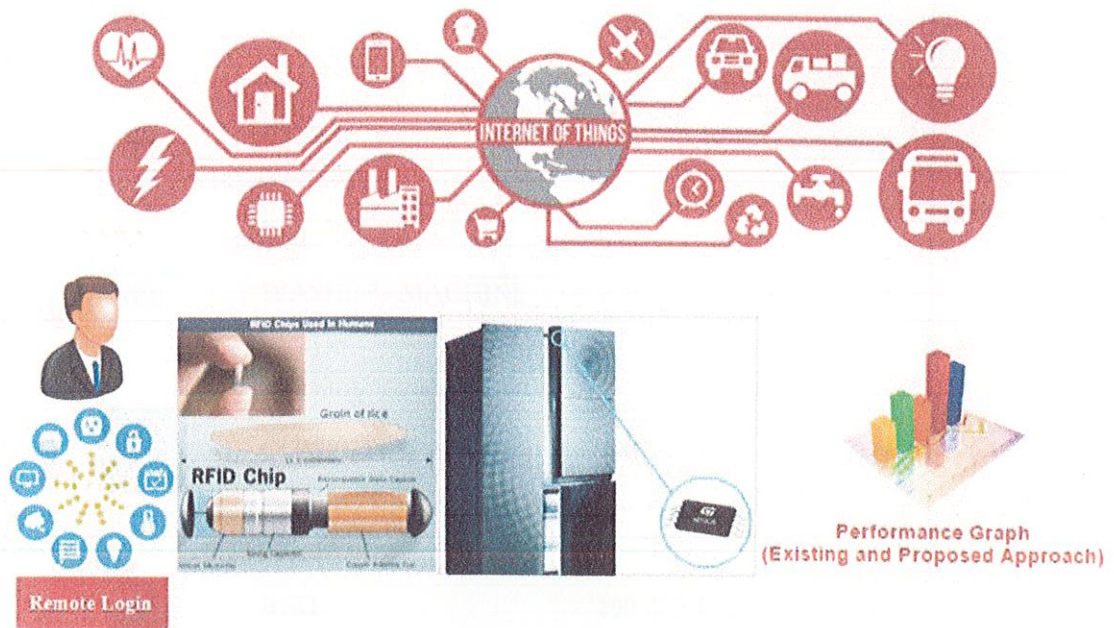


Figure 1 – Simulation Scenario of IoT Devices

Figure 1 depicts the IoT scenario the interconnection of the remote machines. Using this case,







	CAR	199.200.10.2	<a href="#">Select</a>
	CHILD	200.140.49.12	<a href="#">Select</a>
	WASHING MACHINE	230.43.52.11	<a href="#">Select</a>
	COMPUTER	100.242.4.23	<a href="#">Select</a>
	BIKE	100.200.49.89	<a href="#">Select</a>
	AC	200.98.84.13	<a href="#">Select</a>

Figure 2 – Simulation Scenario of IoT Devices in Connection

Figure 2 depicts the number of devices which are connected to the human being with their internal parameters. The parameters includes name, image and IP address of the remote machine.




### Implementation of Security and Integrity in Internet of Things (IoT)



Figure 3 – Authentication Process Initiates

Figure 3 shows the basic queries which will be asked after this scenario. The human being will be able to communicate if the attributes are known to the person in communication

Authentication Process Initialized

  
 car

Personalized Queries	
COLOR	<input type="text"/>
MODEL	<input type="text"/>
BRAND	<input type="text"/>
<input type="button" value="Authenticate"/>	

Figure 4 – Queries of Personalized Aspects

Figure 4 is the screenshot of the queries of the machine car. In this system, the color, model, brand and other details are asked to verify the integrity of the human being

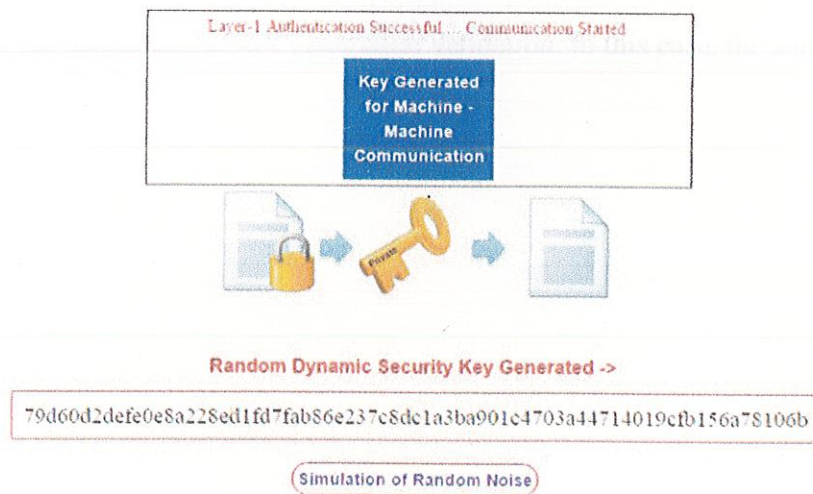


Figure 5 – Dynamic Random Key Generation Process Starts

Figure 5 is the output of dynamic key generated from the IoT scenario. In this screen, it is clearly visible that the system is giving dynamic random key and it will be passed as the private key



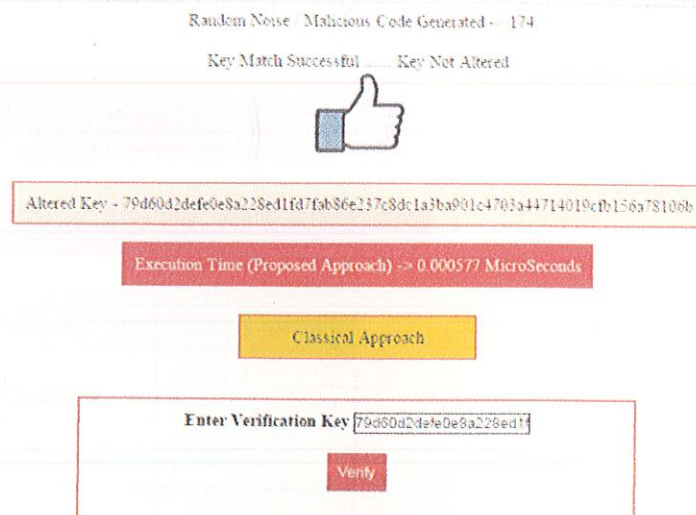


Figure 6 – Validation of Key Generation

Figure 6 depicts the screenshot of key generation validation. In this case, the authentication is verified

[Click Here to OFF](#)



The IoT Remote Device / Machine car is Activated

Execution Time (Classical Approach) -> 0.000721 MicroSeconds

Execution Time (Proposed Approach) -> 0.000577 MicroSeconds

ID	Classical Approach	Proposed Approach
15	0.013363	0.000834
14	0.012302	0.000644
13	0.005451	0.000994
12	0.005194	0.000994
11	0.005228	0.000994
10	0.004766	0.000925

Figure 7 – Comparison of Approaches in Classical and Proposed Way

Figure 7 is the comparison between classical and proposed approach in terms of execution time. This is the major parameter in interest.

### 3.3 Algorithm

Step 1:

Initialize & Activate Packet  $P_i$  at Source  $S_i$  for transmission to Destination  $D_i$

Identify and Record the IP Address



Step 2: Packet Encryption Module  $PE_k$  based on Dynamic Key  $k$  Generation, once the Packet moves from Source  $S_i$

$$C_i := PE_k(P_i)$$

Analysis and Log of the IP Addresses Spoofed or Generated

Step 3: Transmission of Encrypted Packet  $C_i$  using specified Path/Route  $R_i$

$$C_i \rightarrow D_i[R_i]$$

Step 4: Packet Authentication on Decryption

IF (  $C_i = PD_k(C_i)$  ) // Packet Decryption Module  $PD_k$  to decrypt the packet at destination

BEGIN

(a)  $DEST[i] := PD_k(C_i)$

(b) Successful Delivery of Packet

(c) ACK sent to Source  $S_i$  // Acknowledgement ACK is delivered to Source  
in case of Success

END

ELSE

BEGIN

- A record will be inserted in the Forensic Database. The Interception Table will consist of the Structure (Id, Interception Type, Timestamp of Interception). // Acknowledgement ACK is sent to Forensic Database in case of Failure Attempt

- Source  $S_i$  senses the Forensic Database.  
Select All Records from Forensic Database

IF (true)Then

print "Failure Delivery, Retransmit the packet"

(a) GOTO Step 1

(b) Update Forensic Analyzer Database for taking remedial actions.



END

**Step 5: Forensic Analyzer**

- (a) Retrieve Records for analysis of interceptions.
- (b) Analyze the type  $T_i$  of Intercept
- (c) Perform remedial stroke for avoiding the stored interception type

**Forensic Analyzer**

**Step 1:**

Create Database Connection

**Step 2:**

Analyze the Field Bytes Altered in the relation Interception Attempt

**Step 3:**

Associate a Unique Interception Type to the ID and insert a record in the Table Interception Type

**Step 4:**

if another same record of same Bytes Altered is encountered

Increment Occurrences

Keep other fields same

**Step 5:**

Full Outer Join Operation on both tables

Generate Detailed Report of the Interception Type and Number of Occurrences



## CHAPTER 4- PERFORMANCE ANALYSIS

### 4.1 Introduction

In computer science, the performance analysis is the determination of the amount of resources necessary to execute algorithms. Most algorithms are designed to work with inputs of arbitrary length. The efficiency or running time of an algorithm is stated as a function relating the input length to the number of steps which is known as time complexity or storage locations which is known as space complexity.

Algorithm analysis is an important part of computational complexity theory, which provides theoretical estimates of the resources needed by any program which solves a given computational problem. These estimates provide an insight of search for efficient algorithms into reasonable directions

In theoretical analysis it is common to estimate their complexity in the asymptotic sense which means to estimate the complexity function for arbitrarily large input. For this purpose Big O notation, Big-omega notation and Big-theta notation are used.

Exact (not asymptotic) measures of efficiency can sometimes be computed but they usually require certain assumptions concerning the particular implementation of the algorithm, called model of computation.

Time efficiency estimates depend on what we define to be a step. For the analysis to correspond usefully to the actual execution time, the time which is taken by any program to perform a step must be guaranteed to be bounded above by a constant. One must be careful here, for instance, some analysis count an addition of two numbers as one step. This assumption in certain contexts may not be warranted. For example, if the numbers involved in a computation may be arbitrarily large, the time required by a single addition can no longer be assumed to be constant.

Two cost models are generally used:

- the uniform cost model, also called uniform-cost measurement assigns a constant cost to every machine operation, regardless of the size of the numbers involved
- the logarithmic cost model, also called logarithmic-cost measurement assigns a cost to every machine operation proportional to the number of bits involved

Run-time analysis is a theoretical classification that estimates and anticipates the increase in running time of an algorithm as its input size increases. Run-time efficiency is a topic of great interest in computer science. A program can take seconds, hours or even years to finish executing, depending on which algorithm it implements.

## **4.2 Testing**

### **4.2.1 Black Box testing:**

The technique of testing without having any knowledge of the interior workings of the application is called black-box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black-box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.



Advantages	Disadvantages
<ul style="list-style-type: none"> <li>• Well suited and efficient for large code segments.</li> <li>• Limited coverage, since only a selected number of test scenarios is actually performed.</li> <li>• Well suited and efficient for large code segments.</li> <li>• Limited coverage, since only a selected number of test scenarios is actually performed.</li> </ul>	<ul style="list-style-type: none"> <li>• Limited coverage, since only a selected number of test scenarios is actually performed.</li> <li>• Inefficient testing, due to the fact that the tester only has limited knowledge about an application.</li> <li>• Blind coverage, since the tester cannot target specific code segments or error-prone areas.</li> <li>• The test cases are difficult to design.</li> </ul>

Table 4.2.1 Black box testing

#### 4.2.2 White-box testing:

White-box testing is the detailed investigation of internal logic and structure of the code. White-box testing is also called glass testing or open-box testing. In order to perform white-box testing on an application, a tester needs to know the internal workings of the code.

The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately.

The following table lists the advantages and disadvantages of white-box testing.

Advantages	Disadvantages
<ul style="list-style-type: none"> <li>• As the tester has knowledge of the source code, it becomes very easy to find out which type of data can help in testing the application effectively.</li> <li>• It helps in optimizing the code.</li> <li>• Extra lines of code can be removed which can bring in hidden defects.</li> <li>• Due to the tester's knowledge about the code, maximum coverage is attained during test scenario writing.</li> </ul>	<ul style="list-style-type: none"> <li>• Due to the fact that a skilled tester is needed to perform white-box testing, the costs are increased.</li> <li>• Sometimes it is impossible to look into every nook and corner to find out hidden errors that may create problems, as many paths will go untested.</li> <li>• It is difficult to maintain white-box testing, as it requires specialized tools like code analysers and debugging tools.</li> </ul>

Table 4.2.2 White box testing

#### 4.2.3 Grey Box Testing:

Grey-box testing is a technique to test the application with having a limited knowledge of the internal workings of an application. In software testing, the phrase the more you know, the better carries a lot of weight while testing an application.

Mastering the domain of a system always gives the tester an edge over someone with limited domain knowledge. Unlike black-box testing, where the tester only tests the application's user interface; in grey-box testing, the tester has access to design documents and the database. Having this knowledge, a tester can prepare better test data and test scenarios while making a test plan.



Advantages	Disadvantages
<ul style="list-style-type: none"> <li>• Offers combined benefits of black-box and white-box testing wherever possible.</li> <li>• Grey box testers don't rely on the source code; instead they rely on interface definition and functional specifications.</li> <li>• Based on the limited information available, a grey-box tester can design excellent test scenarios especially around communication protocols and data type handling.</li> <li>• The test is done from the point of view of the user and not the designer.</li> </ul>	<ul style="list-style-type: none"> <li>• Since the access to source code is not available, the ability to go over the code and test coverage is limited.</li> <li>• The tests can be redundant if the software designer has already run a test case.</li> <li>• Testing every possible input stream is unrealistic because it would take an unreasonable amount of time; therefore, many program paths will go untested.</li> </ul>

Table 4.2.3 Grey box testing

#### 4.2.4 Functionality Testing:

Test for all the links in applet, database connection, forms used in the applets for submitting or getting information from user, Cookie testing.

##### ➤ Check all the links:

- Test the outgoing links from all the pages from specific domain under test.
- Test all internal links.

- Test links jumping on the same pages.
- Test links used to send the email to admin or other users from applet pages.
- Test link checking, check for broken links in all above-mentioned links.

➤ **Test forms in all pages:**

Forms are the integral part of any applet. Forms are used to get information from users and to keep interaction with them. So what should be checked on these forms-

- First check all the validations on each field.
- Wrong inputs to the fields in the forms.
- Check for the default values of fields.
- Options to create, delete, view or modify the forms

➤ **Cookie testing:**

Cookies are small files stored on user machine. These are basically used to maintain the session mainly login sessions. Test of the application will be done by enabling or disabling the cookies in our browser options. Test to check if the cookies are encrypted before writing to user machine. If we are testing the session cookies check for login sessions and user stats after session end. Check effect on application security by deleting the cookies.

➤ **Validate your HTML/CSS:**

If you are optimizing your site for Search engines then HTML/CSS validation is very important. Mainly validate the site for HTML syntax errors. Check if site is crawl able to different search engines.

➤ **Database testing:**

Data consistency is very important in applet application. Check for data integrity and errors while you edit, delete, modify the forms or do any DB related functionality.



Check if all the database queries are executing correctly, data is retrieved correctly and also updated correctly. More on database testing could be load on DB, we will address this in applet load or performance testing below.

#### **4.2.5 Usability Testing:**

➤ **Test for navigation:**

Navigation means how the user surfs the applet pages, different controls like buttons, boxes or how user using the links on the pages to surf different pages.

➤ **Usability testing includes:**

Applet should be easy to use. Instructions should be provided clearly. Check if the provided instructions are correct means whether they satisfy purpose. Main menu should be provided on each page. It should be consistent.

➤ **Content checking:**

Content should be logical and easy to understand. Check for spelling errors. Use of dark colours annoys users and will not be used in site theme. We can follow some standards that are used for applet and content building. These are common accepted standards like as I mentioned above about annoying colours, fonts, frames etc. All the anchor text links should be working properly. Images should be placed properly with proper sizes. Content should be meaningful. These are some basic standards that should be followed in applet development. Our task is to validate all for UI testing.

➤ **Other user information for user help:**

Like search option, sitemap, help files etc. Sitemap should be present with all the links in applet with proper tree view of navigation. Checking for all links will be done. These are all optional items and if present should be validated.

#### 4.2.6 Interface Testing:

The main interfaces are:

- Applet server and application server interface.
- Application server and Database server interface.

Checks if all the interactions between these servers are executed properly. Errors are handled properly. If database or applet server returns any error message for any query by application server then application server should catch and display these error messages appropriately to users. Checks what happens if user interrupts any transaction in-between.

#### 4.2.7 Compatibility Testing:

Compatibility of our applet in which applet is embedded is very important testing aspect. See which compatibility test to be executed:

- Browser compatibility
- Operating system compatibility
- Mobile browsing
- Printing options

##### ➤ **Browser compatibility:**

Some applications are very dependent on browsers. Different browsers have different configurations and settings that our applet page should be compatible with. Our applet coding should be cross browser platform compatible. If you are using java scripts or AJAX calls for UI functionality, performing security checks or validations then more stress will be given on browser compatibility testing of your applet. Test applet on different browsers like Internet explorer, Firefox, Netscape navigator, AOL, Safari, Opera browsers with different versions.

##### ➤ **OS compatibility:**



Some functionality in our applet may not be compatible with all operating systems. All new technologies used in applet like graphics designs, interface calls like different API's may not be available in all Operating Systems. Testing our applet on different operating systems like Windows, UNIX, MAC, Linux, and Solaris with different OS flavours.

➤ **Mobile browsing:**

This is new technology age. So in future Mobile browsing will rock. Test your applet pages on mobile browsers. Compatibility issues may be there on mobile.

➤ **Printing options:**

If we are giving page-printing options then make sure fonts, page alignment, and page graphics getting printed properly. Pages should be fit to paper size or as per the size mentioned in printing option.

#### **4.2.8 Performance testing:**

Applet should sustain to heavy load. Applet performance testing should include:

- applet Load Testing
- applet Stress Testing

Test application performance on different internet connection speed. In this testing test if many users are accessing or requesting the same page. Can system sustain in peak load times? Applet should handle many simultaneous user requests, large input data from users, Simultaneous connection to DB, heavy load on specific pages etc.

Generally stress means stretching the system beyond its specification limits. Applet stress testing is performed to break the applet by giving stress and checked how system reacts to stress and how system recovers from crashes.

Stress is generally given on input fields, login and sign up areas.

## **Chapter-5 CONCLUSION**

### **5.1 Conclusion**

IP spoofing is less of a threat today due to the patches to the Unix Operating system and the widespread use of random sequence numbering. Many security experts are predicting a shift from IP spoofing attacks to application-related spoofing in which hackers can exploit a weakness in a particular service to send and receive information under false identities. As Security professionals, we must remain current with the Operating Systems that we use in our day to day activities. A steady stream of changes and new challenges is assured as the hacker community continues to seek out vulnerabilities and weaknesses in our systems and our networks.

### **5.2 Future Scope**

As we have seen that there no much reliable and effective technique to prevent from IP spoofing. So, there still need of a lot of work that could be done. There are many tools available to perform the attack but none to ensure complete security from such attacks. We could purpose some changes in the existing algorithms for IP spoofing poisoning prevention .We can use more security constarints so that hacker is not able to control any device which has ip address

### **5.3 Applications of IP Spoofing**

#### **Asymmetric routing (Splitting routing)**

Asymmetric routing means traffic goes over different interfaces for directions in and out. In other words, asymmetric routing is when response to a packet follows a different path from one host to another than the original packet did. The more correct and more general answer is, for any source IP address 'A' and destination 'B', the path followed by any packet (request or response) from 'A' to 'B' is different than the path taken by a packet from 'B' to 'A'.



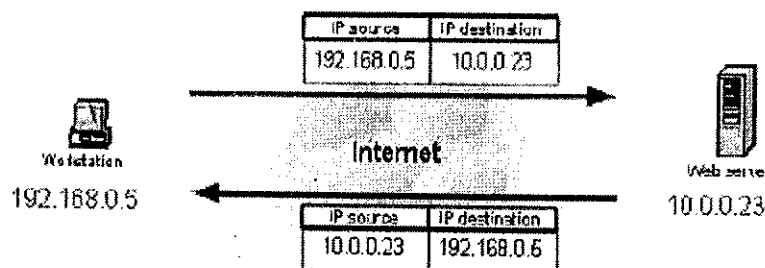


Fig 3.8 Valid Source IP Address

#### Implementation of asymmetric routing

Modern O.S. allows us to receive packets from an input interface, different from the output interface.

In Linux, we can implement asymmetric routing using iptables (linux 2.4): `iptables -A POSTROUTING -t nat -j SNAT -to 192.168.0.5 -oeth0`

This means, for all the packets going out via eth0, their source IP address will be changed to 192.168.0.5. We also have to "disable" reverse path filtering: `echo "0" > /proc/sys/net/ipv4/conf/all/rp_filter`.

## SAT DSL

Satellite DSL (SAT DSL) makes use of asymmetric routing.

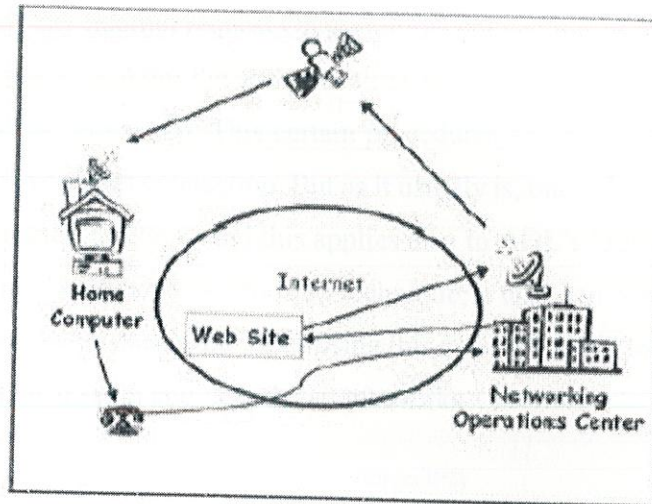


Fig 3.9 satellite DSL

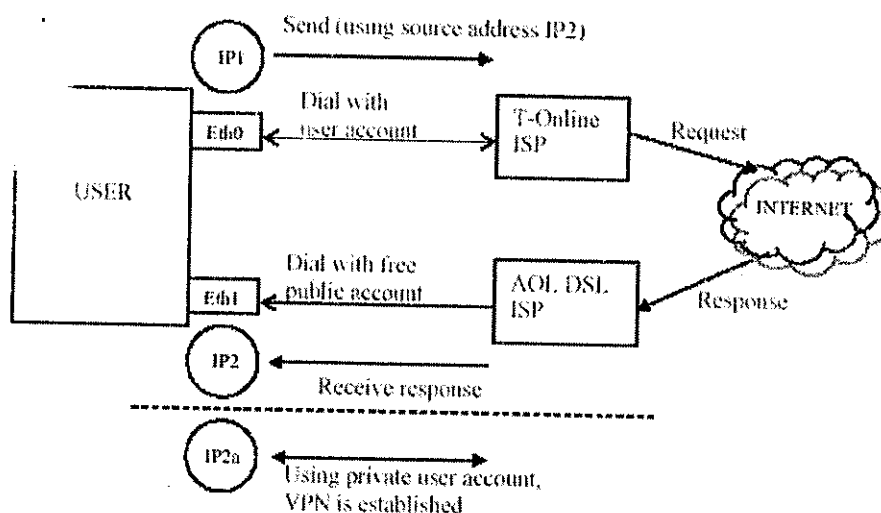
### Satellite DSL

The advantage of a satellite network is to provide high bandwidth services independent of the users location over a wide geographical area. A satellite network consists of two types of stations: feeds and receivers. Every receiver has a satellite dish connected to a user station. The user station has an extra interface, DSL modem connected to the ISP, this is called return channel. All requests to Internet are sent via DSL connection, and responses from Internet should be routed by a feed on the satellite network. After the information is sent from the feed to a satellite, it will be broadcast to all the receivers that belong to the satellite coverage. Installing feeds in strategic positions over the Internet will create shorter paths and higher bandwidth provided by the satellite network. The user host has therefore two IP addresses, one for the satellite subnetwork and the other for the regular connection subnetwork (return channel).

Probable problem with AOLs DSL connection setup AOL DSL service implements a certain connection setup procedure in order — to apply VPN (Virtual Private Network) for its users. When a user dials in to the AOL DSL ISP, these procedures are taken place:



User is connected to the ISP using a public account and so a network connection between user and the ISP is established. But user can only receive data using this connection, thus is not able to send any internet request. On top of this connection, A VPN is established using user's private account. After the authentication succeeds, a user can send and receive data through this VPN connection. This certain procedures are AOL's attempt to create secure internet traffic over DNS connection. But as it usually is, one solution to a security problem may lead to another problem. And this applies also to AOL's DSL connection setup. With certain setup and an IP address spoofing technique, a user can connect to AOL DSL ISP, and download as much data as he wants using this connection without paying any cent. This picture depicts such setup and how the attack works



### Problem in AOL DSL

On first network interface, the user dials for a DSL connection to T-Online or other ISPs using his account. The user can send and receive data with this connection.

On second network interface, the user dials to AOL DSL ISP using a free public account to establish a DSL connection that goes one way from ISP to user.

Before the user sends packet through T-Online connection, he spoofs the source IP address of the packet into the IP address of the second network interface (which is connected to AOL DSL)

And so he sends requests through T-Online connection, and receives response through AOL DSL connection. This way the user only needs to pay for every bits he sends to T-Online, and get for free every bits he receives from AOL DSL, which would have cost a lot more than the cost for sending bits, because people usually spend more time downloading from the internet instead of sending data to the internet.

**Nat** -NAT is network address translation. Normally, packets on a network travel from their source to their destination through many different links. None of these links really alter your packet, they just send it onward. If one of these links were to do NAT, then they would alter the source or destinations of the packet as it passes through. Usually the link doing NAT will remember how it mangled a packet, and when a reply packet passes through the other way, it will do the reverse mangling on that reply packet, so everything works

NAT have several applications:

- **Modem Connections to The Internet**

Most ISPs give you a single IP address when you dial up to them. You can send out packets with any source address you want, but only replies to packets with this source IP address will return to you. If you want to use multiple different machines (such as a home network) to connect to the Internet through this one link, you'll need NAT.

- **Multiple Servers**

Sometimes you want to change where packets heading into your network will go. Frequently this is because (as above) you have only one IP address, but you want people to be able to get into the boxes behind the one with the 'real' IP address. If you rewrite the destination of incoming packets, you can manage this. This type of NAT was called port-forwarding. A common variation of this is load-sharing, where the mapping ranges over a set of machines, fanning packets out to them.



## • Transparent Proxying

Sometimes you want to pretend that each packet which passes through your Linux box is destined for a program on the Linux box itself. This is used to make transparent proxies: a proxy is a program which stands between your network and the outside world, shuffling communication between the two. The transparent part is because your network won't even know it's talking to a proxy, unless of course, the proxy doesn't work. NAT has two different types: Source NAT (SNAT) and Destination NAT (DNAT). Source NAT is when you alter the source address of the first packet: i.e. you are changing where the connection is coming from. Source NAT is always done post-routing, just before the packet goes out onto the wire. Masquerading is a specialized form of SNAT. Destination NAT is when you alter the destination address of the first packet: i.e. you are changing where the connection is going to. Destination NAT is always done before routing, when the packet first comes off the wire. Port forwarding, load sharing, and transparent proxying are all forms of DNAT.

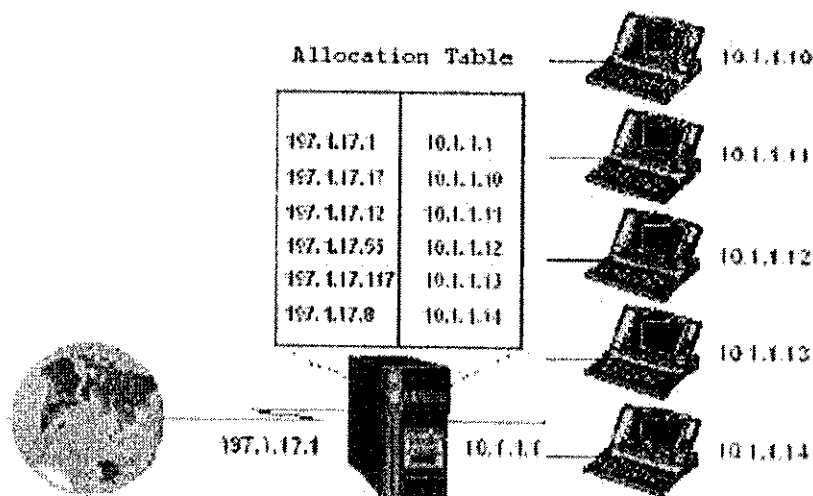


Fig 4.0 NAT

## **IP Masquerade**

IP Masquerade, is a specific form of Network Address Translation (NAT) which allows internally connected computers that do not have registered Internet IP addresses to communicate to the Internet via the Linux server's Internet IP address. IP masquerading lets you use a single Internet-connected computer running Linux with a real IP address as a gateway for non-connected machines with "fake" IP addresses. The Linux box with a real address handles mapping packets from your intranet out to the Internet, and when responses come back, it maps them back to your intranet. This lets you browse the web and use other Internet functions from multiple machines without having a special network setup from your ISP.

IP Masquerade is a networking function in Linux similar to the one-to-many (1:Many) NAT (Network Address Translation) servers found in many commercial firewalls and network routers. For example, if a Linux host is connected to the Internet via PPP, Ethernet, etc., the masquerade feature allows other "internal" computers connected to this Linux box (via PPP, Ethernet, etc.) to also reach the Internet as well. Linux IP Masquerading allows for this functionality even though these internal machines don't have an officially assigned IP address. IP masquerading is different from NAT. While IP masquerading implements a specific many-to-one NAT, IP NAT allows complex many-to-many translations. For static real IP address we use NAT, while for dynamic real IP address (via PPP) we use IP masquerading.

## **Services vulnerable to IP Spoofing**

Configuration and services that are vulnerable to IP spoofing

- RPC (Remote Procedure Call services)
- Any service that uses IP address authentication
- The X Window system
- The R services suite (rlogin, rsh, etc.)



## TCP and IP spoofing Tools

- Mendax for Linux -Mendax is an easy-to-use tool for TCP sequence number prediction and rshd spoofing.
- spoofit.h - spoofit.h is a nicely commented library for including IP spoofing functionality into your programs.
- ipspoof - ipspoof is a TCP and IP spoofing utility.
- hunt - hunt is a sniffer which also offers many spoofing functions.
- dsniff - dsniff is a collection of tools for network auditing and penetration testing. dsniff, filesnarf, mailsnarf, msgsnarf, urlsnarf, and webspay passively monitor a network for

## REFERENCES

- [1] Controlling IP Spoofing Through Packet Filtering Using Simulations In Blowfish Algorithm Yogesh Singh1, Hariom Awasthi M.Tech. Scholar, NIET NIMS UNIVERSITY Jaipur, A.P. dept. of IT/CSE, NITE, NIMS UNIVERSITY Jaipur, Rajasthan-303121, India
- [2] P. Watson, .Slipping in the window: TCP reset attacks,. In Cansecwest Conference,2004
- [3] J. Stewart, .DNS cache poisoning - the next generation,. LURHQ, Technical Report, Jan. 2003
- [4] A Comprehensive Analysis of Spoofing P. Ramesh Babu Dept of Information Technology Rajamahendri Inst. of Engg & Technology Rajahmundry
- [5] D.Lalitha Bhaskari Dept of C.S & S.E AU College of Engineering Visakhapatnam INDIA CH.Satyanarayana Dept of C.S.E JNTUK College of Engineering Kakinada –
- [6] Analyze and Determine the IP Spoofing Attacks Using Stackpath Identification Marking and Filtering Mechanism V. Shyamaladevi1 , Dr. R.S.D Wahidabanu Research Scholar, K.S.Rangasamy College of Technology Tiruchengode, Tamilnadu, India
- [7] A Compressed Anti IP Spoofing Mechanism using Cryptography S.Gavaskar, Dr.E.Ramaraj Research Scholar , Technology Adviser Madurai Kamraj University, Madurai.

### Websites-

<http://searchsecurity.techtarget.com/definition/IP-spoofing>  
<http://www.symantec.com/connect/articles/ip-spoofing-introduction>  
[https://en.wikipedia.org/wiki/IP\\_address\\_spoofing](https://en.wikipedia.org/wiki/IP_address_spoofing)  
<http://www.veracode.com/security/spoofing-attack>  
<https://www.techopedia.com/definition/3993/ip-spoofing>  
<http://www.computerworld.com/article/2546050/network-security/the-top-five-ways-to-prevent-ip-spoofing.html>  
<https://www.internetsociety.org/blog/2013/06/can-we-stop-ip-spoofing-internet>



