# A Proposed Framework for Integrating Stack Path Identification and Encryption Informed by Machine Learning as a Spoofing Defense Mechanism

Anne Kaluvu[1]

*School of computing, Jomo Kenyatta University of Agriculture and Technology, Kenya*

***Abstract:*** *Spoofing attacks have been terrorizing the information world for decades; so many methodologies have been formulated to attempt the eradication of these attacks. This study elaborates on a proposed framework for integrating StackPi and Encryption informed by Machine learning as spoofing defense methodologies. IP Spoofing is one of the major tools used by hackers in the internet to mount spoofing attacks and has been difficult to eradicate. Stack Pi uses Path Identification markings to differentiate between spoofed packets and the legitimate packets and in addition encryption is used to apply proper authentication measures that can enhance the speed of detection and prevention of IP spoofed packet. Machine learning incorporated in this framework to address the short comings of StackPi-IP filtering method and thereby increasing its efficiency. The integration of these three methodologies makes an ideal mechanism for eradicating spoof attacks.*

***Keywords:*** *Spoof attacks, Stack Pi, Encryption, Machine Learning*

## I. Introduction

Sending IP packets with fake source addresses is known as packet spoofing and is used by attackers for numerous purposes. These include obscuring the correct source of the attack, implicating an additional site as the attack source, pretending to be a trusted host, hijacking or interrupting network traffic, or causing replies to goal another system (Linta & Khan, 2013).

In today's Internet, Linta & Khan, (2013) noted that attackers can forge the source address of IP packets to both maintain their anonymity and redirect the blame for attacks. When attackers inject packets with spoofed source addresses into the Internet, routers forward those packets to their destination just like any other packet often without checking the validity of the packet's source addresses. IP Spoofing is one of the major tools used by hackers in the internet to mount spoofing attacks. In such attacks the attackers duplicate the Source IP of packets that are used in the attack. Instead of carrying the original source IP of the machine the packet came from, it contains an arbitrary IP address which is selected either random fashion or particularly. The ease with which such attacks are generated made them very popular (Wyld et al., 2011).

To be successful, the intruder must first find out the IP address of a trusted system, and then change the packet headers such that it appears that the packets are impending from the trusted system. In IP address spoofing Internet Protocol packets are created with forged source IP address. The Main aim of spoofing is for hiding sender identity. In this the attacker without authority access computer or network showing as if malicious message came from trusted machine by spoofing that machine Address. (Gupta & Kavyashree., 2013).

One of the most difficult challenges in defending against spoofing is that attackers often spoof the source IP address of their packets and thus evade traditional packet filters. Unfortunately, the current routing infrastructure cannot detect that a packet's source IP address has been spoofed or from where in the Internet a spoofed IP packet has originated from (Gupta & Kavyashree., 2013). The combination of these two factors makes IP spoofing easy and effective for attacks. In fact, many different types of Internet attacks utilize spoofed IP addresses for different purposes.

## II. Proposed Framework

Zargar, et al., (2010) established that combining source address authentication (to prevent IP spoofing), capabilities, and filtering would be the most effective and efficient solution because of the robustness of capabilities and the relative simplicity of a capability-based design.

The researcher proposes the following (figure 1) combinational framework as a defense mechanism against spoofing attacks.

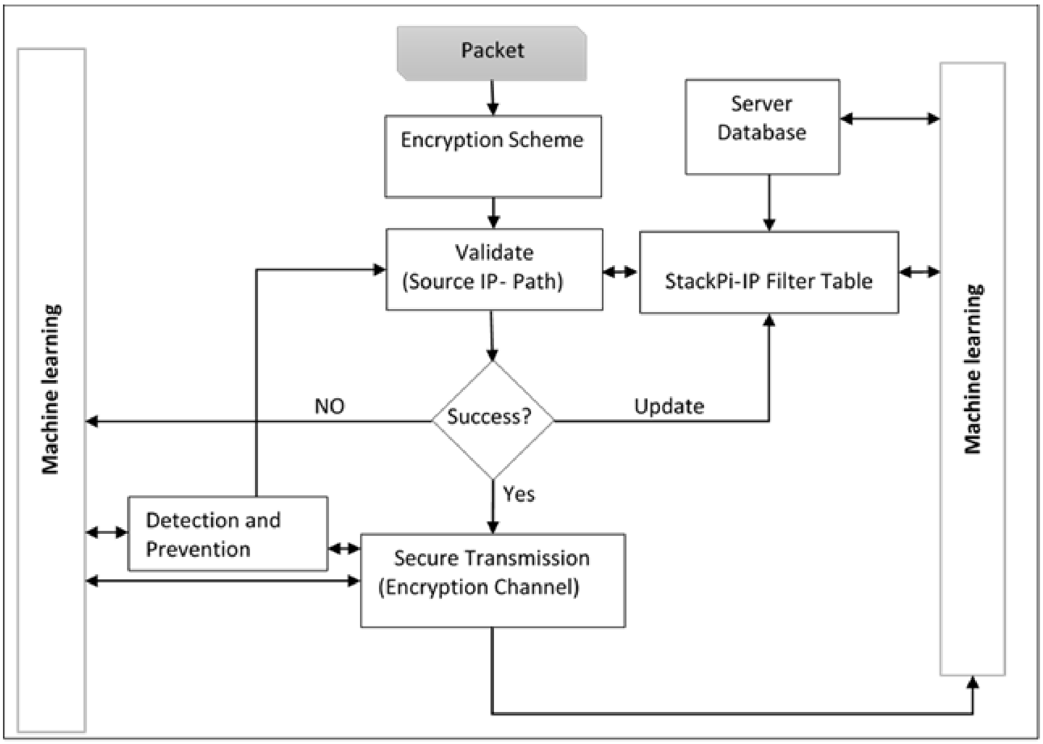

**Fig 1: Proposed Framework: Source Researcher, 2014**

### 1.1 How it Works

In this framework researcher assumes a hash code encryption scheme, however, the encryption scheme should be predetermined by the financial institution and should be enforced at both ends. This encryption is used to reduce collisions among packet-markings.

In brief, this is how the framework works; encryption is done for source IP Address into fixed-length hash code using hash function and placed into Identification field of IPv4 Header and sent packet into the network. A hash function is applied by the receiver to the source IP Address to produce hash code which is compared to the hash code available in Identification field. If both hash codes are equal then packet is authenticated. If source IP Address of packet modified in network by an attacker than hash code will not be equal and recipient discards that packet. Further, the packets path is validated using marking (stackPi-IP) and detection schemes (informed by machine learning) source is verified and packet is validated. Once packet and source address is validated then the packet is transferred for better detection and prevention of spoofed attack using machine learning, then the filter table is updated accordingly. The time required to mark each packet is saved because in this framework, once a secure transmission is established between source and destination then there is no requirement of marking and comparing process at participant routers and firewall router respectively.

If it is first time communication between sender and receiver then with the help of marking and detection schemes source is verified and packet is validated. The researcher proposes that the stackPi-IP filter table use two filter tables; filter-in and a blacklist (also regarded as drop list) table. The filter-in table consists of legitimate user's paths and IPs, the drop list on the other hand consists of known threats and their subnets whose attempted communication with the network is just dropped; by dropping it means will be silent in its denial, that is, the connection will be rejected but the initiator will assume that no service is running on the target host (or that target host does not exist). During the entire cycle of the framework it interacts with machine learning which tracks and learns packet behavior, the database filter table and checklist are consequently updated.

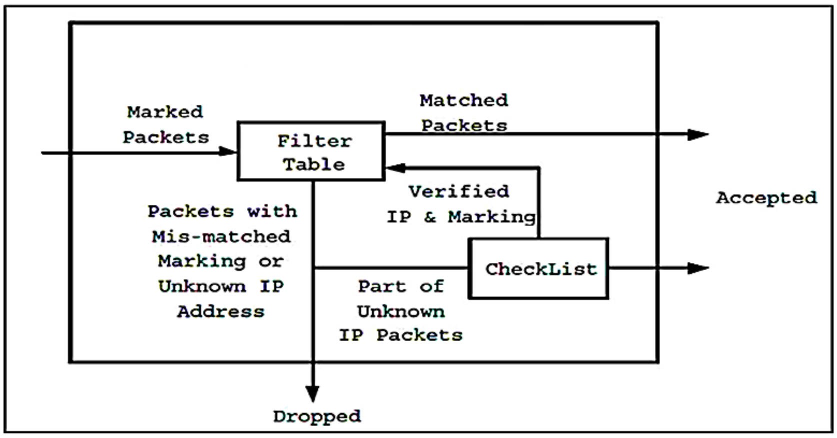

**Fig2: Marking scheme: Adopted from Bangar et al. (2012)**

The Hashed encryption scheme employs a firewall at each of the perimeter routers of the network to be protected and scans the marking field of all incoming packets to selectively filter the attack packets. Bangar, et al. (2012). On implementing the stackPi marking any packet arriving at the network is marked depending only on the path it has traversed. If the source IP address of a packet is spoofed, this packet must have a marking that is different from that of a genuine packet coming from the same address. The spoofed packets can therefore be easily identified and dropped by the filter, while the legitimate packets containing the correct markings are accepted.

The filtering sequence is such that If the (stackPi-IP) tuple is same with one of the records in the Filter Table, the packet is received; If the source IP address of the packet exists in the Filter Table, but the marking does not match, this packet is considered to be a spoofed packet and is dropped, the path and if possible subnet is blacklisted, filter table updated. If the source IP address does not appear in the Filter Table, then this packet is accepted securely for detection and prevention after which the packets will either be dropped or accepted. All echo reply messages that are received as responses to the firewall's requests are handled by the Check List verification process. They are not passed through the filter.

### 1.2 StackPI

In StackPi, as a packet traverses routers on the path towards its destination, the routers deterministically mark bits in the packet's IP Identification field. The deterministic markings guarantee that packets traveling along the same path will have the same marking. StackPi allows the victim and routers on the attack path to take a proactive role in defending against spoofing attacks by using the StackPi mark to filter out attack packets on a per packet basis. In addition, the victim can build statistics over time relating StackPi marks to IP addresses. Then if an attacker spoofs an IP address, it is likely that the StackPi mark in the spoofed packet will not match the StackPi mark corresponding to the legitimate IP address in the database, thus enabling the victim to tag packets with possibly spoofed source IP addresses. StackPi is also effective against other IP spoofing attacks such as TCP hijacking and multicast source spoofing attacks. (Das, et al., 2010; (Gupta & Kavyashree., 2013).

Pi reuses the fragmentation field of an IP packet to identify the path the packet traveled. As a packet travels the network, each router it encounters sets a bit in the fragmentation field. When the packet reaches its destination, the fragmentation field will contain a marking that is (almost) unique to the path the detection. (Soon, 2012).

### 1.3 Packet Filtering (StackPi-IP Filtering Mechanism)

StackPi allows for per-packet filter decisions and is geared to defend against spoofing attacks, it is extremely important that the filters at the endhost have a low per packet computation cost, as an endserver will need to be able to filter every packet that arrives over the network.

For the stackPi filtering design the researcher proposes the use of stackPi-IP filtering design. Numerous researches have indicated that packets from a given IP network will all arrive at the destination with a small number of distinct Pi marks, we can use this to design a powerful filter to reject packets with spoofed IP addresses.

Consider the following setup; during peace time (when a server is not under attack), the server stores the tuple <Pi mark, source IP address>, or (<Pi,IP>). When the server is under attack, it uses the <Pi,IP> database to filter out packets with spoofed source IP addresses. For each incoming packet, the server checks whether the < Pi,IP> tuple of the arriving packet matches an entry in the database; if the tuple does not match the corresponding entry in the database, it rejects the packet. A nice feature of this PiIP filter is that the server can filter out the very first malicious attacker packet. However, the forwarding path of a legitimate receiver may change and the arriving packet's <Pi,IP> tuple may not be in the database. Thus, the application writer needs to consider the output of the PiIP filter as a hint on whether the source IP address is spoofed or not. As long as the server has sufficient capacity, questionable packets may also get served, and if the packet originator turns out to be a legitimate user, the server can add the <Pi,IP> tuple to its database. Note that the PiIP filter cannot be used to detect IP spoofing attacks if the IP address in the packet is not in the database. However there are several ways to address this issue. Because packets from the same network (even if not from the same IP addresses) usually have the same Pi mark, from the Pi mark of one IP address we can derive the Pi mark of other IP addresses on the same network, this is also where machine learning comes in.

The StackPi filtering extremely light-weight and efficient, but here it presents a slightly more complex but more accurate filtering method. The filter itself is simple; examine an incoming packet's StackPi mark and source IP address and allow access based on that tuple. Ideally, a database of legitimate users' h Stack Pi, IP i tuple will be built during times when there are few or no ongoing attacks. Any packet with a StackPi marking that does not match the StackPi marking of the same IP address in the database will be flagged as a packet with a spoofed IP address. In the case of filtering based only on StackPi markings, we have to assume that our filters get feedback from some higher layer algorithm that can classify some sampled packets as legitimate packets or attack packets, and tell the filter which StackPi markings correspond to attack traffic and should be dropped. The StackPi-IP filter does not rely as strongly on this assumption, because the StackPi-IP filter does not need to be bootstrapped with attack traffic. Quite the opposite, the StackPi-IP filter is bootstrapped during non-attack periods and identifies attack periods by an increase in the incidence of packets with spoofed IP addresses. We define the set of n distinct StackPi markings recorded for address k as {m0,m1…..;mn}. For each Stack Pi mark recorded at the victim for IP address k, there is a set of other IP addresses that also map to the same StackPi mark. If the attacker were to spoof any of these, the attack packet would be accepted by the filter. Thus, the probability of an attacker with IP address k successfully spoofing is:

$$P_k = \frac{\sum_{i=0}^{n} uniqueIPs(m_i, k)}{N}$$

**(1)** Adopted from Perrig et al. (2002)

Where, the unique IPs function returns the number of unique IP addresses that map to Pi mark mi, excluding IP address k as well as any duplicates between function calls, and N represents the number of end-hosts in the topology; which is the size of the list of possible IP addresses that the attacker can spoof. Given the probability of an attacker with a specific IP address of successfully spoofing a packet, we can now calculate the probability of an attacker with a random IP address successfully spoofing:

$$P = \frac{\sum_{k=0}^{N} P_k}{N}$$

**(2)** Adopted from Perrig et al. (2002)

Numerous research in real topologies have shown that an attacker has a very small chance to successfully spoof another IP address that is not from the same network as the attacker.

### 1.4 Enabling Traceback with StackPi-IP filters

A properly bootstrapped StackPi-IP filter in conjunction with machine learning can be used to perform standard traceback, that is, complete path reconstruction from a packet's destination to its sender. When a destination receives a packet that is flagged because its source IP address does not match its StackPi marking in the StackPi-IP filter's database, the victim can consult its database to generate a list of IP addresses that correspond to the packet's StackPi mark. Once this list is compiled, the victim can determine the paths by simply executing traceroutes to the addresses on the list. Although this method does not guarantee a unique path to the sender (because there may be multiple IP addresses that map to the same StackPi mark), it does reduce the space of potential attackers and may allow the victim's administrator to cull the true attack path using external knowledge and intuition; machine learning could also be used to enhance traceroute (Dehmer & Basak,. 2012).

### 1.5 Encryption

Implementing encryption and authentication will also reduce spoofing threats this is further enhanced by ensuring that the proper authentication measures are in place and carried out over a secure (encrypted) channel. With the help of cryptosystem we can enhance the speed of detection and prevention of IP spoofed packet.

Rather than doing the marking for each packet after confirmation of source validity, if further packet transmission is required the packet is put in secure transmission with cryptosystem. It would be more reliable that source address of IP packet should be encrypted.

The researcher proposes that financial institutions select an encryption schemes that best suits them. To understand how this works better consider a hash encryption scheme; encryption is done for source IP Address into fixed-length hash code using hash function and place this hash code into Identification field (of IPv4, this is also applicable in IPV6) header and send that packet into the network. On the other side, recipient receives that packet and applies hash function to the source IP Address to produce hash code and compare this hash code to the hash code available in Identification field. If both hash code are equal then packet is authenticated. If source IP Address of packet modified in network by an attacker than hash code will not be equal and recipient discard that packet.

At sender side source address of sender inside generated packet is used to generate the hash code with the help of any known hashed algorithm. This hash code is written in to the identification field of the packet, then the IP packet is transferred by usual method. Whenever IP packet is received at receiver side if it is first time communication between sender and receiver then with the help of marking and detection schemes source is verified and packet is validated. Once packet and source address is validated then the packet is transferred for better detection and prevention of IP spoofed attack using machine learning then the filter table and checklist is updated accordingly. All these measures are carried out over a secure (encrypted) channel.

The time required to mark each packet is saved because in this framework once a secure transmission is established between source and destination then there is no requirement of marking and comparing process at participant routers and firewall router respectively.

### 1.6 Machine learning

Machine learning incorporated in this framework to address the short comings of StackPi-IP filtering method and thereby increasing its efficiency. In machine learning this is done via three approaches:

Firstly, by inferring StackPi markings of previously unseen IP addresses. We observe that for a given destination, all the packets originating from the same network region (sometimes from the same CIDR block) will usually be routed along the same path and have the same StackPi marking. If we have seen a StackPi mark from a given network, we could infer the StackPi marks of other hosts within the same network. To ensure we reliably derive information about hosts that are on the same network we will consider using the CIDR block information from BGP routing, and using machine learning techniques in conjunction with longest prefix matching of IP addresses with their associated StackPi marks.

Secondly, by infering multiple StackPi markings in case of multi-path or short path. For a given destination, if a region has multiple paths to the destination, then the StackPi marking of any host within that region may have multiple values. For example, given two hosts, A and B, from the same region, and whose StackPi marks are X and Y, respectively, then it is likely that the StackPi marking of A could also be Y, and the StackPi marking of B could also be X. We could use machine learning techniques to automatically detect this case and infer the multiple StackPi markings from observed data. In the case where some bits in the StackPi mark still contain the original bits of the IP Identification field, research has shown that the StackPi markings have the same low order bits (from router markings pushed onto the stack) and only vary in the high-order bits (those bits that were not overwritten by router markings). Using machine learning techniques, we could automatically detect this case and filter only based on those bits that were not originally in the IP Identification field.

Thirdly, by inferring StackPi marking change caused by route change. When routes change, StackPi markings will change for some end-hosts. Because packets from the same region will have the same StackPi markings, the change of StackPi marking for one IP address will have a similar change for another IP address from the same region. Using machine learning techniques we could infer the StackPi marking change caused by a route change with a small number of packets. Also, with machine learning techniques, we may be able to infer how route changes affect the StackPi markings and hence infer the StackPi marking change of one network region by observing the StackPi marking change of another network region.

## III. Proposed Framework Evaluation

The proposed frame work presents advantages as the integrated approach ensures the different methodologies complement each other and presents several advantages including; ensuring high speed filtering of spoofed packet, enhancement in packet transmission, and once secure transmission is established no role of participating router in filtering process.

The analysis indicates the expected values of the proposed framework performance effectiveness against random strategy selection (indicated by stackPi Non-users). The figures indicate a significant difference in performance with the proposed framework being better. For instance, even before the introduction of encryption and machine learning in figure 3, when attack traffic is 160 times user traffic (in the attack scenario of 500 users and 8000 attackers), 50% of the server's capacity is utilized for servicing legitimate user's packets when using the StackPi filter, while only 0.6% of the server's capacity is used to serve legitimate user's packets when the server uses a random selection strategy. Also worth noting is the significant improved performance with the introduction of encryption and machine learning figure 4.

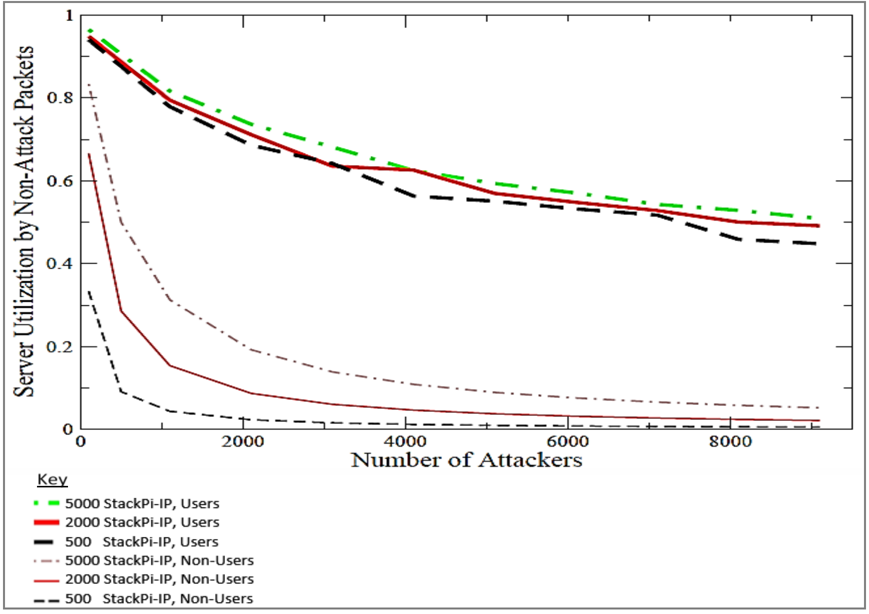

**Fig 3:** StackPi-IP number of attackers against server utilization.

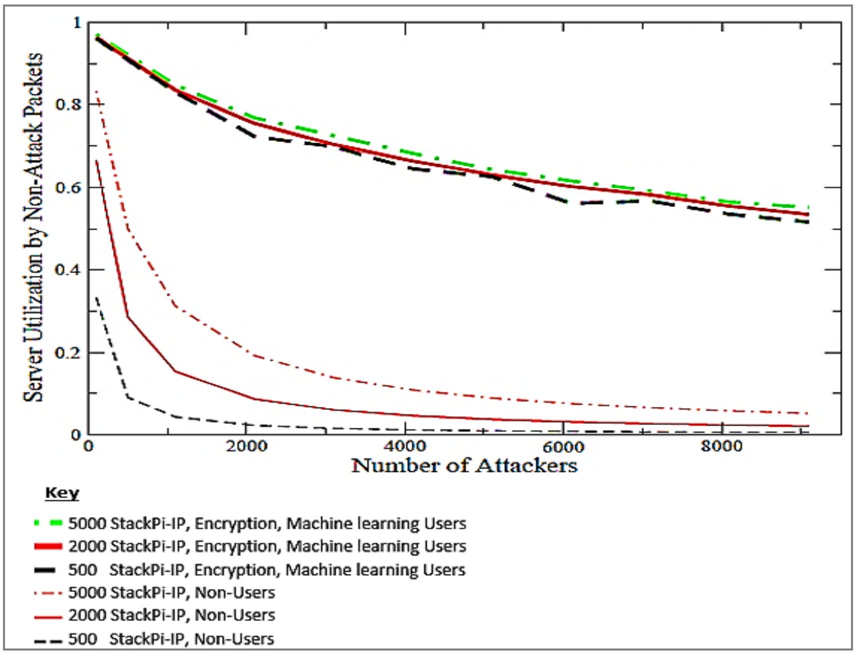

**Fig 4:** Combined approach number of attackers against server utilization.

Sensitivity analysis is the study of how uncertainty in the output of a model can be attributed to different sources of uncertainty in the model input. The sensitive variable is modeled as uncertain value while all other variables are held at baseline values (stable), (Pannell, 2010).

A sensitivity analysis was done to determine correlation between the factors that are desired in defense mechanism against spoofing attacks and our approach. As indicated in the table 3.1, if properly implemented the proposed frame has all the desired properties; with filtering (both per packet and victim being its major strength) with expected values of 11.45 and 10.03 respectively, the ability for scalability in deployment and efficiency are also some of the characteristics portrayed by the framework. Privacy ranks low but at acceptable level, this could be attributed to the fact that privacy is a result of various factors and is very relative, however this can be improved.

**Table 1:** Sensitivity analysis: StackPi | Desired properties

| | | **StackPi-IP** | | |
|---|---|---|---|---|
| | | Expected Value | Variance | S.D |
| **Desired Properties** | Per-packet filtering | 11.45 | 324.211 | 18.006 |
| | Victim filtering | 10.036 | 178.663 | 13.366 |
| | Incremental deployment | 8.801 | 138.77 | 11.78 |
| | Efficient | 6.802 | 77.904 | 8.826 |
| | Fast response | 7.126 | 85.125 | 9.226 |
| | Scalable | 5.128 | 46.983 | 6.854 |
| | Privacy | 4.252 | 26.457 | 5.144 |

## IV. Summary

A synergy between machine learning and filtering methods, and encryption provides an optimum defense mechanism against spoofing attacks. Once an IP and subsequent subnet is detected to have initiated a spoofing attack and fails to authenticate it is blacklisted and all packets from this source are dropped.

## References

[1]. Linta S. R. and Khan, R. . Today's Impact on Communication System by IP Spoofing and Its Detection and Prevention. Santa Cruz, CA. GRIN Verlag. 2013.

[2]. Linta S. R. and Khan, R. . Today's Impact on Communication System by IP Spoofing and Its Detection and Prevention. Santa Cruz, CA. GRIN Verlag. 2013.

[3]. Wyld D. C, Wozniak M and Chaki. N, Trends in Network and Communications: International Conferences. In Proceedings Volume 197 of Communications in Computer and Information Science, ISSN 1865-0937. Chennai, India. Springer Science & Business Media. 2011.

[4]. Gupta, V & Kavyashree, H. Comparative Study of IP Address Spoofing: Attacks and Their Defense Mechanism. International Journal of Innovative Research and Studies. Vol 2 Issue 5. ISSN 2319-9725. 2013.

[5]. Gupta, V & Kavyashree, H. Comparative Study of IP Address Spoofing: Attacks and Their Defense Mechanism. International Journal of Innovative Research and Studies. Vol 2 Issue 5. ISSN 2319-9725. 2013.

[6]. Zargar, S. T, Joshi, J and Tipper, D. A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. IEEE communications surveys & tutorials, accepted for publication. 2010.

[7]. Bangar. P. C, Mulani. J.A, Ekad, A. B, Ganjewar, P.D, Shinde, P. Study of IP-Spoofed Distributed DoS Attacks Using Hashed Encryption Scheme. International Journal of Computer Science, Information Technology and Management.( 1). 1-2. Pune University.2012.

[8]. Bangar. P. C, Mulani. J.A, Ekad, A. B, Ganjewar, P.D, Shinde, P. Study of IP-Spoofed Distributed DoS Attacks Using Hashed Encryption Scheme. International Journal of Computer Science, Information Technology and Management.( 1). 1-2. Pune University.2012.

[9]. Das, V.V, Stephen, J and Chaba, Y. Computer Networks and Information Technologies: Second International Conference on Advances in Communication, Network, and Computing, CNC 2011, Bangalore, India. Proceedings. Springer Science & Business Media. 2011.

[10]. Gupta, V and Kavyashree, H. Comparative Study of IP Address Spoofing: Attacks and Their Defense Mechanism. International Journal of Innovative Research and Studies. Vol 2 Issue 5. ISSN 2319-9725.2013.

[11]. Soon, L. IP Spoofing Defense: An Introduction. Universiti Putra Malaysia, 43400 Serdang, Selangor.2012.

[12]. Perrig, A,Song, D and Yaar,A. Stack Pi: A new Defense Mechanism against IP spoofing and DDOS attacks. Carnegie Mellon University. Pittsburgh, PA. 2002

[13]. Perrig, A,Song, D and Yaar,A. Stack Pi: A new Defense Mechanism against IP spoofing and DDOS attacks. Carnegie Mellon University. Pittsburgh, PA. 2002.

[14]. Dehmer M. & Basak, S. Statistical and Machine Learning Approaches for Network Analysis. New York City. John Wiley & Son.2012.

[15]. Pannell, D.J. Sensitivity analysis of normative economic models: Theoretical framework and Practical strategies.University of Western Australia, Nedlands, W.A. 6907, Australia. 2010.