Volume 2, No. 3, May-June 2011
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

IP Spoofing Traceback – Recent Challenges and Techniques

Manish Kumar*
Asst. professor,
Dept. of Master of Computer Applications
M. S. Ramaiah Institute of Technology,
Bangalore, India
manishkumarjsr@yahoo.com

Dr. M. Hanumanthappa
Dept. of Computer Science and Applications,
Jnana Bharathi Campus, Bangalore University,
Bangalore .India
hanu6572@hotmail.com

Dr. T.V. Suresh Kumar
Professor & Head,
Dept. of Master of Computer Applications,
M. S. Ramaiah Institute of Technology,
Bangalore,India
hod_mca@msrit.edu

Abstract: - In current Internet communication world, validity of the source of IP packet is an important issue. The problems of IP spoofing alarm legitimate users of the Internet. IP spoofing is a technique used to gain unauthorized access to computers, whereby the intruder sends messages to a computer with an IP address indicating that the message is coming from a trusted host. This paper review recent progress of spoofing defenses by various researchers. Techniques and mechanisms proposed are categorized to better illustrate the deployment and functionality of the mechanism.

Keywords: IP Spoofing, Intrusion, Man in Middle Attack

I. INTRODUCTION

IP spoofing is a technique used to gain unauthorized access to computers, whereby the intruder sends messages to a computer with an IP address indicating that the message is coming from a trusted host. To engage in IP spoofing, a hacker must first use a variety of techniques to find an IP address of a trusted host and then modify the packet headers so that it appears that the packets are coming from that host.
After the occurrence of the infamous Internet Worm, IP spoofing has been identified as a real risk to the Internet and computer network community. Since then, the Internet has suffered a huge number of large-scale attacks. There are many variants of IP spoofing used in an attack. In this paper, we aim to examine the attack methods, and to identify counter-measures.

II. ATTACK WITH IP-SPOOFING

A. *Background*

IP is the connectionless, unreliable network protocol in the TCP/IP suite. It has two 32-bit header fields to hold address information. IP's job is to route packets around the network. It provides no mechanism for reliability or accountability. IP simply sends out the data and hopes they make it intact. If they don't, IP can try to send an ICMP (Internet Control Message Protocol) error message back to the source, however this packet can get lost as well. IP has no means to guarantee delivery. Since IP is connectionless, it does not maintain any connection state information. The fact that it is easy to modify the IP stack to allow an arbitrarily chosen IP address in the source (and destination) fields makes IP vulnerable to attacks .

A → B: SYN; my number is X
B → A: ACK; now X+1
SYN; my number is Y
A→ B: ACK; now Y+1

Figure 1 TCP/IP handshake

TCP is the connection-oriented, reliable transport protocol in the TCP/IP suite. Connection-oriented means the two hosts participating in a discussion must first establish a connection before data may change hands. Three-way handshake is used to establish a connection , as outlined in figure 2.1

Reliability is provided in a number of ways, here we are only concerned with are data sequencing and acknowledgement. TCP is layered on top of IP and provides virtual circuits by splitting up the data stream into IP packets and reassembling them at the far end. TCP assigns sequence numbers to every segment and acknowledges all data segments received from the other end. Both hosts use this number for error checking and reporting.

B. *IP spoofing*

IP spoofing uses the idea of trust relationships. The attack is a "blind" one, meaning the attacker will be assuming the identity of a "trusted" host. From the perspective of the target host, it is simply carrying on a "normal" conversation with a trusted host. In reality, the host is conversing with an attacker who is busy forging IP packets. The data that the target sends back (destined for the trusted host) will go to the trusted host, which the attacker never “sees” them. To prevent disruption

from the trusted host, he has to disable the trusted host, using DOS, so that it will not respond to the target's replies. The attacker must guess what the target sends and the type of response the server is looking for. By trial communication with the target, the attacker can predict the initial sequence number (ISN) in the target's response. He then does not need to actually "see" the response. This allows him to work in the "blind" and manipulate the system.
IP spoofing (Figure 2.1.1) consists of these steps:

- Selecting a target host (the victim).
- Identifying a host that has a "trust" relationship with the target. This can be accomplished by looking at the traffic of the target host. There cannot be an attack if the target does not trust anyone.
- The trusted host is then disabled using *SYN flooding* (Figure 2.2.2) and the target's TCP sequence numbers are sampled.
- The trusted host is impersonated and the sequence number forged. This is difficult when the attacker has to find out the target's ISN and the round trip time between the target and the attacker's host.
- A connection attempt is made to a service that only requires address-based authentication (no user id or password).
- If a successful connection is made, the attacker executes a simple command to leave a backdoor. This allows for simple re-entries in a non-interactive way for the attacker.

IP spoofing works because trusted services only rely on network address-based authentication. Since IP is easily duped, address forgery is not difficult. The hardest part of the attack is in the sequence number prediction, because that is where the calculation and guesswork comes into play.

C. *Attacks*

Attacks using IP spoofing includes:

- Man–in-the-middle (MITM): packet sniffs on link between the two endpoints, and therefore can pretend to be one end of the connection.
- Routing re-direct: redirects routing information from the original host to the attacker's host (a variation on the man-in the-middle attack).
- Source routing: The attacker redirects individual packets by the hacker's host.
- Flooding: SYN flood fills up the receive queue from random source addresses.
- Smurfing: ICMP packet spoofed to originate from the victim, destined for the broadcast address, causing all hosts on the network to respond to the victim at once. This congests network bandwidth, floods the victim, and causes a loop at the victim.

With MITM attack, packets between the two ends go through the attacker and the attacker 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. Routing attack refers to redirecting the route of packets. Sender of a packet can specify the route that a packet should take through the network. As a packet travels through the network, each router will examine the "destination IP address" and choose the next hop to forward the packet to. For DOS, the attacker creates half-open connections that fill up the system and disable the system from receiving new incoming requests. Normally there is a timeout associated with a pending connection, so the half-open connections will eventually expire and the victim server system will recover. However, the attacking system can send IP spoofed requests faster than the victim system can release the pending connections. In smurfing, the attacker uses ICMP echo requesting packets directed to IP broadcast addresses from remote locations to generate a denial-of-service attack. A common implementation of this process is the "ping" command, which is included with many operating systems and network software packages.

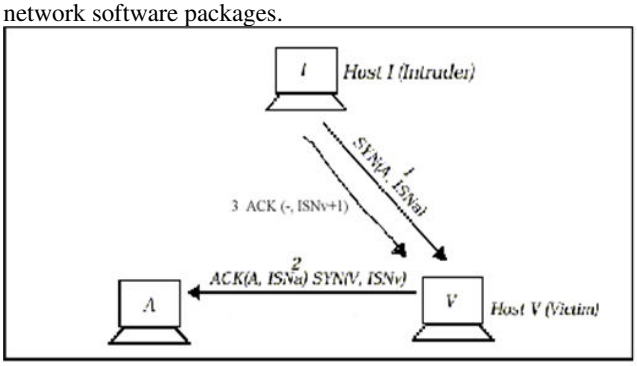

Figure 2.A typical IP spoofing

D. *Tracing Challenges*

One major problem in building an effective response to network-based attacks is the lack of source identification. Without effective source tracing, the attacked victim is blind at defending network-based attacks, and no effective intrusion countermeasures such as blocking and containing can be implemented. Network-based attacks can not be effectively repelled or eliminated until its source is known.
Masquerade attacks [23] can be produced by spoofing at the link-layer (e.g., using a different MAC address than the original), at the Internet layer (e.g., using a different source IP address than the original), at the transport layer (e.g., using a different TCP/IP port than the original one), at the application layer (e.g., using a different email address than the original). Let $C = H1 \rightarrow H2 \rightarrow \ldots \rightarrow Hi \rightarrow Hi+1 \rightarrow \ldots \rightarrow Hn$ be the connection path between hosts $H1$ to Hn. Then, the IP traceback problem is defined as: Given the IP address Hn, identify the actual IP addresses of hosts Hn-1, …, $H1$. If $H1$ is the source and Hn is the victim machine of a security attack, then C is called the attack path [23].

Reconstruction of the attack path back to the originating attacker $h1$ may not be a straightforward process because of possible spoofing at different layers of the TCP/IP protocol stack and also the intermediate hosts becoming compromised hosts, called stepping-stone, and acting as a conduit for the attacker's communication. The security functions practiced in existing networks may also preclude the capability to follow the reverse path. For example, if the attacker lies behind a firewall, then most of the traceback packets are filtered at the firewall and one may not be able to exactly reach the attacker. The link testing techniques start the traceback from the router closest to the victim and

interactively determine the upstream link that was used to carry the attack traffic. The technique is then recursively applied on the upstream routers until the source is reached. Link testing assumes the attack is in progress and cannot be used "post-mortem". There are two varieties of link testing techniques: input debugging and controlled flooding.

Given a series of computer hosts $H_1, H_2, \ldots H_n$ (n>2), when a person (or a program) sequentially connects from H_i into H_{i+1} (i=1,2,..n-1), we refer to the sequence of connections on $<H_1, H_2, \ldots H_n>$ as a connection chain, or chained connection. The tracing problem of a connection chain is, given H_n of a connection chain, to identify $H_{n-1}, \ldots H_1$.

E. Tracing Approaches

In general, tracing approaches for a connection chain can be divided into two categories: host-based and network-based, each of which can further be classified into either active or passive. The fundamental problem with the host-based tracing approach is its trust model. Host-based tracing places its trust upon the monitored hosts themselves. In specific, it depends on the correlation of connections at every host in the connection chain. If one host is compromised and is providing misleading correlation information, the whole tracing system is fooled. Because host-based tracing requires participation and trust of every host involved in the network-based intrusion, it is very difficult to be applied in the context of the public Internet.

Network-based tracing is the other category of tracing approaches. Neither does it require the participation of monitored hosts, nor does it place its trust on the monitored hosts. It is based on the property of network connections: the application level content of chained connections is invariant across the connection chain. In particular, the thumbprint [21] is a pioneering correlation technique that utilizes a small quantity of information to summarize connections. Ideally it can uniquely distinguish a connection from unrelated connections and correlate those related connections in the same connection chain. While thumb printing can be useful even when only part of the Internet implements it, it depends on clock synchronization to match thumbprints of corresponding intervals of connections. It also is vulnerable to retransmission variation. This severely limits its usefulness in real-time tracing.

One fundamental problem with passive network-based approaches is its computational complexity. Because it passively monitors and compares network traffic, it needs to record all the concurrent incoming and outgoing connections even when there is no intrusion to trace. To correlate at any host in the connection chain, it needs to match every concurrent incoming connection with every concurrent outgoing connection at that host. That is, for a host with *m* concurrent incoming connections and *n* concurrent outgoing connections, the passive network-based correlation approach would take $O(m \times n)$ comparisons, in addition to the $O(m+n)$ scanning and recording of concurrent connections.

On the other hand, the active network-based approach dynamically controls how connections are correlated through customized packet processing. It does not need to record all the concurrent incoming and outgoing connections at any host in the connection chain. It does not need to match each concurrent incoming connection with each concurrent outgoing connection. For a host with m concurrent incoming connections and n concurrent outgoing connections, the active network-based approach is able to correlate within time dependent only on the number of connections being actively traced, in addition to the O(m+n) scanning of concurrent connections.

F. Current Active Research on IP Traceback

Two network tracing problems are currently being studied: IP traceback and traceback across stepping-stones (or a connection chain). IP traceback is to identify the origins of sequences IP packets (e.g., identify the origin of DDOS packets) when the source IP addresses of these packets are spoofed. IP traceback is usually performed at the network layer, with the help of routers and gateways. Traceback across stepping-stones is to identify the origin of an attacker through a chain of connections (e.g., connections established with telnet, rlogin, or ssh), which an attacker may use to hide his/her true origin when he/she interacts with a victim host. Traceback across stepping-stones is beyond the network layer, since at each intermediate host the data is transmitted to application layer in one connection, and then resent to the network in the next connection. The problem we propose to address is the latter one.

Research on IP trace back has been rather active since the late 1999 DDOS attacks [15,16,17]. Several approaches have been proposed to trace IP packets to their origins. The IP marking approaches enable routers to probabilistically mark packets with partial path information and try to reconstruct the complete path from the packets that contain the marking [18,19,20]. DECIDUOUS uses IPSec security associations and authentication headers to deploy secure authentication tunnels dynamically and trace back to the attacks' origins [21,22]. ICMP traceback (iTrace) proposes to introduce a new message "ICMP trace back" (or an iTrace message) so that routers can generate iTrace messages to help the victim or its upstream ISP to identify the source of spoofed IP packets [23]. An intention-driven iTrace is also introduced to reduce unnecessary iTrace messages and thus improve the performance of iTrace systems [24].

An algebraic approach is proposed to transform the IP traceback problem into a polynomial reconstruction problem, and uses techniques from algebraic coding theory to recover the true origin of spoofed IP packets [25]. An IP overlay network named CenterTrack selectively reroutes interesting IP packets directly from edge routers to special tracing routers, which can easily determine the ingress edge router by observing from which tunnel the packets arrive [26]. A Source Path Isolation Engine (SPIE) has been developed; it stores the message digests of recently received IP packets and can reconstruct the attack paths of given spoofed IP packets [27,28]. There are other techniques and issues related to IP traceback (e.g., approximate traceback [29], legal and societal issues [30], vendors' solutions [31]). An archive of related papers can be found at [32].

Though necessary to make attackers accountable (especially for DDOS attacks where there are a large amount of packets with spoofed source IP addresses), IP traceback has its own limitations. In particular, IP traceback cannot go beyond the hosts that send the spoofed IP packets. Indeed, a typical attacker will use a fair number of steppingstones before he/she finally launches, for example, a DDOS attack. Thus, only identifying the source of IP packets is not sufficient to hold the attackers responsible for their actions.

Similar to IP traceback, there have been active research efforts on tracing intruders across stepping-stones. In general, approaches for traceback across a connection chain can be, based on the source of tracing information, divided into two categories: host-based and network-based. In addition, depending on how the traffic is traced, traceback approaches can be further classified into either active or passive. Passive approaches monitor and compare all the traffic all the time, and they do not select the traffic to be traced. On the other hand, active approaches dynamically control when, where, what and how the traffic is to be correlated through customized processing. They only trace selected traffic when needed.

III. COUNTER-MEASURES

IP spoofing is dangerous and can be carried out nearly undetectably. There is generally no complete solution to prevent this type of attack. As we have mentioned earlier, the attack is contributed by the weakness inherent in the design of IP protocol [23]. Since IP packet makes no assumptions about the sender and recipient, routers along the path do not check the sender's identity. Routers will find ways to reach the destination the packet is intended for, and are not concerned with the packet's origin or its intended purpose. They only look at the destination address, that of the recipient, to decide whether they should accept the packet for their network, or forward it to one of their neighbors.

We have reviewed different type spoofing defense mechanisms proposed by various researchers. These studies have shown that most researchers try to deploy spoofing defense during packet transmission, as a credit to customer and the ISP that implement it. Network Ingress Filtering works effectively but it only prevents its own network from spoofing, rather than protecting its own network from being spoofed. On the other side, spoofing defense at the destination might introduce new problems other than anti spoofing. Deploying spoofing defense during transmission seems promising with acceptable overhead and deployment cost, but there is an obstacle ahead – the Internet itself. The architecture of the Internet consists of thousands of ASes. Each AS contains a collection of connected IP routing prefixes under the control of routers of Internet Service Provider (ISP) with a defined routing policy to the Internet. ASes of the Internet communicate with each other, maintain reachability and route traffics via various type of the routing protocol which keeps evolving. With different routing algorithm and techniques, it is hard to implement a single spoofing defense mechanism that works with everyone. With IP multicast routing, mobility network and multihomed network, it further complicates the effort to deploy the spoofing defenses effectively.

In the Spoofing Prevention Method (SPM), router that is closer to the destination of a packet verifies the authenticity of the source address of the packet [22]. Routers mark and check outgoing packet with label related to the destination. An encrypted unique temporal key is associated with each ordered pair of source and destination network. The key is known in advance by both parties, and used as a lightweight authentication mechanism to authenticate the source address of incoming packets. Keys are placed when the packet is sent out from the router and being removed after the key of the packet is authenticated (at the incoming router). When ISP detects an attack on its network, they protect themselves by allowing only packets that come from SPM member network to ensure clean traffics.A prototype testbed system for SPM in IPv6 was deployed in [23]. Their experimental result shows that SPM is able to work as proposed with the existing network architecture.

In [24] Peer-to-Peer Based Anti-Spoofing Method (APPA) is proposed. APPA works for inter AS and intra AS level. Similar to SPM, APPA tags a key on each packet at the source and verify the key at the destination. The key is only used once on a packet and will be changed for the next packet. This method addresses the problem of anti-replay which exists in several other proposed methods. They proposed The State Machine which will change the state according to the condition and create a unique key for all packets. They deploy APPA and perform strict experiments on it [26]. The result shows that APPA is very lightweight and have high efficiency. SPM and APPA have major advantage over RBF: SPM is an end-to-end protocol and requires lower deployment cost, while RBF can only work (efficiently) if all ASes implement RBF. Both SPM and APPA will work well if the edge router implements it. Spoofing detection will not work if either side of the source or the destination is not SPM or APPA router.

Distributed Packet Filtering, IP spoofing can be limited based on global routing information [27]. Route based Distributed Packet Filtering (DPF) is placed on routers at vertex cover of AS network. Every router maintains a route and filtering table. Assume a packet is sent from source S to destination D. When the packet enters the router of network from S, a set of feasible routes is being computed. Based on the routing policy, the best path is being chosen. In DPF, the shortest path is being implemented into the policy. The path from S to D is being maintained in the router's routing table. The incoming interface of the entering packet is checked when the packet arrives, by looking up into the routing table. If packet arrived from an unexpected interface, the packet will be dropped.

Route based DPF is able to trace the attacker's source AS with only one spoofed packet arrival at the victim. For tracing back the attacker's location, the route based DPF is able to minimize the possible attacker's origin network up to a very small range of network. Their work shows that they can limit IP spoofing but it has some implementation issue; the scope of the work is too big. It is impossible to get all ISP around the world to implement it. Updating and maintaining the routing table (precisely) will also be a problem.

Duan, Yuan and Chandrashekar proposed an interdomain packet filter (IDPF) architecture based on locally exchanged Border Gateway Protocol (BGP) updates only [28], as an extension of DPF.

Source Address Validation Enforcement (SAVE) is a new protocol proposed to provide information needed to validate the source address of incoming packet [32]. Each router that the packet traverse build correct incoming table with incoming interface. With this incoming table, each router can verify the packet and filter packets with mismatching source address. SAVE provides end-to-end anti spoofing mechanism. Each router sends updates to neighbor router from time to time to update each other's incoming table like BGP and Routing Information Protocol (RIP). SAVE update records the path the update had traversed and ensures that the update message traverse through the correct path. RBF limits the range of IP addresses for possible spoofing attacks but a spoofing attack is still possible. IDPF and SAVE further

improve RBF by forwarding packets only if they came from the correct interface. Packet forwarding with source verification was proposed in [33] to address spoofing prevention via two approaches. In the first approach, definitive packet tagging, routers tag packet that originate from their domain. Along the path the packets traverse, the tag of packet will be verified. Once verified, the valid packet will be re-tagged with the tag of the forwarding router. This hop-wise tagging process will keep the number of tags each implementing router has. Packet with insufficient tagor incorrectly tagged is dropped.The second approach, deductive packet tagging, routers can verify and tag packets from nearby domain. Implementing routers involved in TCP handshake process from random routers to verify the tags.

In [15] BGP Anti-Spoofing Extension (BASE) was proposed. BASE is similar to source verification method. It combines the mechanism of DPF and Path Identifier [16]. BASE filters packet based on their path tags. Packet is tagged by a hashed marking value of their BGP path that is distributed using BGP updates. Every packet from the same source address will have the same tag regarding the path they traverse and interface they arrive from. When a packet arrives at a BASE deployed router, the router will tag outgoing packet and drop incoming packet without proper tag.

Unicast Reverse Path Forwarding (uRPF) requires that the traffic is forwarded only if the traffic arrives at the same interface as the one that is used by the router to reach the source in the forwarding table [17, 18]. Although the mechanism is simple, the effectiveness of uRPF is limited. With current architecture of the Internet, many multihomed networks have different interfaces for incoming and outgoing traffics. Traffics might traverse different path and uRPF requires extra lookup at the router's forwarding table for each packet that arrive at the router.

The efficiency of RPF depends on BGP routing information. RPF will drop valid packet if the router does not receive routing information BGP updates for the source prefix. In Spoofing Prevention based on Hierarchical Coordination Model (SP-HCM), each ordered pair of source and destination network have a unique temporary signature [19]. Similar to SPM, routers in ASes mark outgoing packets with the signature.

Upon arrival of the packet at the border router, the signature is being examined and verifies the authenticity of its source address. Source address information is transmitted by Hierarchical Coordination Model (HCM) using dynamic bloom filter. In SP-HCM, the nodes of AS have sensor that continuously perform tasks by querying routers' Management Information Base (MIB) through Simple Network Management Protocol (SNMP) to gather information about managed entities. Actuator at border routers will poll for information from sensor and process it. The network address space signature is exchanged this way. Similar problem as SPM appears. SP-HCM will only work if all ASes deploy SP-HCM mechanism.

IV. SPOOFING DETECTION AT DESTINATION

Wang, Jin and Shin proposed defense against spoofed traffic based on the value of Time To Live (TTL) on packet and compute the total hop the packet traveled from the source (attacker) to the destination [20]. This value is very accurate as the value of TTL on a packet is not forgeable by the attacker. TTL field of an IP header specifies the maximum lifetime of an IP packet. Routers perform decrement by 1 on TTL when forwarding the packet to the next router. When a packet arrives at destination, TTL is subtracted with the initial value of TTL to get the total number of hop the packet traversed. The authors built Hop Count Filtering (HCF) at the end host, an accurate IP to hop-count (IP2HC) map by grouping IP prefixes based on hop count. In this case, TTL plays the same role as the temporal key in SPM, to authenticate packets that arrived at the destination.

The effectiveness of HCF lies on the hop-count values of the packet. HCF cannot detect spoofed and legitimate packets with same hop-count. Based on authors' work, they suggests that spoofed IP packets have mismatched IP address and hop-count (based on IP2HC). By performing a lookup in IP2HC map HCF is able to drop spoofed traffics. HCF is believed to work well as an attacker is not able to falsify the value of TTL, but intermediate attackers will be able to try to launch an attack from location with matching hop-count values. HCF causes delays to transmission. To overcome this problem, HCF operates under alert mode to detect spoofed traffic and action mode to drop packets when spoofed traffic is detected. Action mode will perform per-packet hop-count computation and compare with values in IP2HC. HCF is deployed at end host, hence easier to deploy compared to RBF.

In [21] Probabilistic Packet Marking (PPM) was proposed to mark packet with partial path information at routers. Each router marks their IP address onto the packet with the probability of P along the way the packet traversed. When DDOS attack is detected, the victim can reconstruct the whole path after collecting certain amount of packet by using the information of the mark, despite the source address in the IP header. PPM has very low overhead as it only mark by the probability of P, but it has a high computation overhead and this method is not effective. In [22] PPM was modified and reduce the computation overhead to an acceptable level. In [23] authors combine PPM and the concept of winding number. Their work shows that they are able to correctly trace the attacker's router IP address using integral equation. Deterministic Packet Marking (DPM) marks all incoming packet at the ingress router interface [24]. End host maintains a table that contains all the source addresses and their incoming interface addresses. Incoming interface address is split into two and is used as a marker to mark on packets and it will require at least two packets to obtain the interface's address. When marked packets arrive at end host, end host will check if the match of ingress address and the source address is found. If not found the match will be inserted into the table. This method seems promising but the authors' benchmarking shows that it has 50% false positive rate. They further improve DPM and apply Single-digest modification which marks packets with hash value of the interface address and segment number of the hash value [25]. The newly improved technique successfully reduces the false positive rate to 1%. DPM was improved and introduced Flexible DPM (FDPM) in [26]. Adopting a flexible mark length strategy, FDPM marks packet with flexible lengths depending on the network protocol used. The authors performed simulation and real implementation on FDPM and achieved a better result than DPM.

Path Identifier (Pi) proposed a packet marking algorithm to mark each packet that traverses through Pi enabled routers onto the packet's header in IP Identification field [16]. The IP Identification field is broken into 16/n sections. When ever a

packet enters a Pi enabled router, the router compute the value of the current packet's TTL modulo 16/n and insert it into the IP Identification field before the packet is to be forwarded.

Pi acts as a fingerprint of the packet. Packets traveling the same path will have the same Pi value. Since it is a per-packet marking mechanism, the victim will be able to defend himself from DDoS attack by filtering packets that carry the same Pi as the attacker's packet. Pi works well under the network where all routers deploy the Pi marking scheme. Unfortunately, it is rather impossible to have all routers in ASes from different ISP to deploy Pi. Furthermore, the performance of Pi degrades when there are non-Pi enabled routers in between the path. These legacy routers will forward packets without marking them. Also the authors identified the problem of Pi where TTL is vulnerable to attacks. Dynamic Pi proposed a method to mark packet dynamically with 1 or 2 bit [27]. Their result shows that Pi could obtain a better result if appropriate marking scheme is used.

StackPi improved Pi's performance by proposing two new marking schemes – Stack-based marking and Write-ahead marking [28]. StackPi treats IP Identification field as a stack. When a packet enters a StackPi router, it left shift the value of IP Identification field for n bits and mark (push) its own marking bit into the stack. For packets that arrive on legacy router, the packet will have no interaction with the marking and will be forwarded. For routers that have the IP address of the next-hop, the router computes the marking bit for the next router and push into the stack. This Write-ahead marking increases the performance of StackPi against legacy routers. StackPi's mechanism also increases the performance of HCF is being implemented together. In [29] authors enhanced Pi's idea and introduced AS-based Edge Marking (ASEM) to marks packet at AS level. ASEM only marks incoming packets on edge routers. All incoming packets are marked with the AS number (ASN) of the edge router it enters. AS path is claimed to be shorter than the IP path, hence address the problem of Pi. Pi limits the number of Pi mark to be store in IPv4 header to 16 bits, while the estimated size of the Internet requires 28 bits to store Pi for end to end hosts. AS level marking is also more stable compared to IP level marking.

Inspired by Pi, Farhat proposed Implicit Tokens Scheme (ITS), which works similarly [30]. In ITS, client is required to perform a TCP three-way handshake with the token database. Once the three-way handshake is completed, Pi to the client will be dynamically added to the database. For each legitimate client IP, a token consisting of the source IP and Pi will be composed and attached to all incoming packets at the router. The packet will be examined by routers along the way it traversed. Packet without a token or without a valid token will be dropped. ITS is able to obtain good results but it creates significant resource overhead at the router. The author further improved ITS by adapting Bloom Filters into the token database [31]. Rather than maintaining the token in a database, the tokens are added to Bloom Filters every time it is composed. With this the storage and computation overhead are greatly reduced. In [32] authors also proposed a packet marking method with Bloom Filter. All routers calculate the hash value of its own IP address and convert the digest value to a bit array. Routers mark incoming packets with the value of the bits via Bloom Filter. Whenever a packet arrives at a router, the mark is examined with the digest value of routers. If the packet is legitimate, OR gate operation is performed using the mark in the packet and the current router's digest bits as input. The output of the OR gate operation is then marked on the packet before forwarding it to the next router. All packets that traverse the same path will have the same mark when they arrive at the destination. A packet can be backtraced by comparing the mark of the packet with the previous router's digest bits hop by hop in the possible path. The Attack Diagnosis (AD) method [33] applies divide-and-conquer strategy in tracing packet source. AD is divided into two paradigms. Attack detection on near victim's host is performed. Once an attack is detected, it will notify upstream routers to start marking incoming packets with an interface port identifier (PID) for traceback. Based on the marking of packets, victim can separate an attacker's traffic from other clients' traffic and notify the upstream router to filter the packets. AD has lower processing overhead compared to other proposed methods as routers mark packet only after an attack is detected. This is also the downside of AD: if an attack reached the end-host, then the damage is done.

V. CONCLUSION

P spoofing is an attack that is unavoidable. The attack exploits trust relationships in a world that everything wants to be connected to everything else. If a system is connected to the Internet and provides services, it is vulnerable to the attack. By studying the attack methods, we learn how IP spoofing works and can then identify the weaknesses of a system. By examining the counter-measures, we learn what we need to do for defense, and what we do not need to have, in terms of services and applications. By implementing the counter methods above, an administrator can guarantee that he has a low risk of being attacked. It should be remembered that there is no full proof mechanism to prevent this kind of attack. Thus, even a low risk can provide a considerable level of network security.

VI. REFERENCES

[1] A. Yaar, A. Perrig, and D. Song, "Pi: A path identification mechanism to defend against DDoS attacks," in *Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy*, 2003, pp. 93-107.

[2] H. Wang, C. Jin, and K. G. Shin, "Defense against spoofed IP traffic using hop-count filtering," *IEEE/ACM Transactions on Networking,* vol. 15, pp. 40-53, 2007.

[3] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Practical network support for IP traceback," *Computer Communication Review,* vol. 30, pp. 295-306, 28 August 2000 through 1 September 2000 2000.

[4] D. Q. Li, P. R. Su, and D. G. Feng, "Notes on packet marking for IP traceback," *Ruan Jian Xue Bao/Journal of Software,* vol. 15, pp. 250-258, 2004.

[5] M. M. Viana, R. Rios, R. M. De Castro Andrade, and J. N. De Souza, "An innovative approach to identify the IP address in denial-of-service (DoS) attacks based on Cauchy's integral theorem," *International Journal of Network Management,* vol. 19, pp. 339-354, 2009.

[6] A. Belenky and N. Ansari, "IP traceback with deterministic packet marking," *IEEE Communications Letters,* vol. 7, pp. 162-164, 2003. [25] A. Belenky and N. Ansari, "On deterministic packet marking," *Computer Networks,* vol. 51, pp. 2677-2700, 2007.

[7] Y. Xiang, W. Zhou, and M. Guo, "Flexible deterministic packet marking: An IP traceback system to find the real source of attacks," *IEEE Transactions on Parallel and Distributed Systems,* vol. 20, pp. 567-580, 2009.

[8] G. Lee, H. Lim, M. Hong, and D. H. Lee, "A dynamic path identification mechanism to defend against DDoS attacks," in *Lecture Notes in Computer Science*, 2005, pp. 806-813.

[9] A. Yaar, A. Perrig, and D. Song, "StackPi: New packet marking and filtering mechanisms for DDoS and IP spoofing defense," *IEEE Journal on Selected Areas in Communications,* vol. 24, pp. 1853-1863, 2006.

[10] Z. Gao and N. Ansari, "A practical and robust inter-domain marking scheme for IP traceback," *Computer Networks,* vol. 51, pp. 732-750, 2007.

[11] H. Farhat, "Protecting TCP services from denial of service attacks," in *Proceedings of the 2006 SIGCOMM workshop on Large-scale attack defense* Pisa, Italy: ACM, 2006.

[12] H. Farhat, "A scalable method to protect from IP spoofing," in *1st International Conference on the Applications of Digital Information and Web Technologies, ICADIWT 2008*, Ostrava, 2008, pp. 569-572.

[13] T. Hosoi, K. Matsuura, and H. Imai, "IP traceback by packet marking method with bloom filters," in *Proceedings -International Carnahan Conference on Security Technology*, Ottawa, ON, 2007, pp. 255-263.

[14] R. Chen, J. M. Park, and R. Marchany, "A divide-and-conquer strategy for thwarting distributed denial-of-service attacks," *IEEE Transactions on Parallel and Distributed Systems,* vol. 18, pp. 577-588, 2007.

[15] S. M. Bellovin. (2000) ICMP Traceback Messages. Internet Draft: draft-bellovin-itrace-00.txt.

[16] W. Bender, D. Gruhl, N. Morimoto and A. Lu. (1996) Technique for Data Hiding. IBM Systems Journal, Vol. 35, Nos. 3&4

[17] S. Bhattacharjee, K. L. Calvert and E. W. Zegura. (1997) An Architecture for Active Networking. High Performance Networking (HPN'97), White Plans, NY.

[18] K. L. Calvert, S. Bhattacharjee and E. Zegura. (1998) Directions in Active Networks. IEEE Communication Magazine.

[19] R. H. Campbell, Z. Liu, M. D. Mickunas, P. Naldurg and S. Yi. (2000) Seraphim: Dynamic Interoperable Security Architecture for Active Networks. In Proceedings of IEEE OPENARCH'2000.

[20] H. Y. Chang, R. Narayan, S.F. Wu, B.M. Vetter, X. Y. Wang et al. (1999) DecIdUouS: Decentralized Source Identification for Network-Based Intrusions, In Proceedings of 6th IFIP/IEEE International Symposium on Integrated Network Management.

[21] S. Staniford-Chen, L. T. Heberlein. (1995) Holding Intruders Accountable on the Internet. In Proceedings of IEEE Symposium on Security and Privacy.

[22] Computer Emergency Response Team. (2000) CERT Advisory CA-2000-01 Denial-of Service Development. http:// www.cert.org/advisories/CA-2000-01.html.

[23] Computer Emergency Response Team. (1999) Results of the Distributed-Systems Intruder Tools Workshop. http://www.cert.org/reports/dsit_workshop.pdf.

[24] Computer Security Institute. Annual CSI/FBI Computer Crime and Security Survey. (2001) http://www.gocsi.com/prelea_000321.htm.

[25] [11] N.G. Duffield and M. Grossglauser. (2000) Trajectory Sampling for Direct Traffic Observation. Proceedings of the ACM SIGCOMM '2000.

[26] M. B. Greenwald, S. K. Singhal, J. R. Stone and D. R. Cheriton. (1996) Design an Academic Firewall: Policy, Practice and Experience with SURF. Internet Society Symposium on Network and Distributed System Security (NDSS '96).

[27] L. T. Heberlein, K. Levitt and B. Mukherjee. (1992) Internetwork Security Monitor: An Intrusion-Detection System for Large-Scale Networks. In Proceedings of 15th National Computer Security Conference.

[28] J. D. Howard. (1997) An Analysis of Security Incidents on The Internet 1989 - 1995, PhD Thesis, http://www.cert.org/research/JHThesis/Start.html.

[29] J. Ioannidis and M Blaze. (1993) The Architecture and Implementation of Network-Layer Security under Unix. In Proceedings of 4th USENIX Security Symposium.

[30] W. Jansen, P. Mell, T. Karygiannis, D. Marks. (1999) Applying Mobile Agents to Intrusion Detection and Response. NIST Interim Report (IR) – 6416.

[31] H. Jung, et al. Caller Identification System in the Internet Environment. (1993) In Proceedings of 4th USENIX Security Symposium.

[32] S. Kent, R. Atkinson. (1998) Security Architecture for the Internet Protocol. IETF RFC 2401.

