Journal of Physical Science and Innovation **Volume 7, No. 2, 2015**
ISSN: 2277-0119

THE MENACE OF IP-SPOOFING VULNERABILITY IN NETWORK ENVIRONMENTS AND MITIGATION RESPONSES

F. I. Onah

Department of Computer Science
Cross River University of Technology, Calabar, Nigeria
E-mail: ikonah80@yahoo.com

ABSTRACT

The menace of IP spoofing across the globe continues to pose challenges to administrators and stakeholders in today's networked world. Online criminals repeatedly attempt to circumvent traditional authentication safeguards through sophisticated attacks that specifically targets consumers, enterprises and citizens. The detection and prevention of malicious behaviors that can compromise the security and trust of a computer system are of primary concerns. In this paper, the challenges posed by IP-spoofing in network environments are examined. Various threat cases are identified and methods of mitigation are recommended. This enables third-party software developers to design and implement defense mechanisms against organized and systematic fraudulent attacks on computer systems in network environments.

Keywords: *Attack signatures, IP hijacking, Transmission Control Protocol (TCP), Mitigation response*

CONCEPTUAL OVERVIEW

IP Spoofing is the forging of the Internet Protocol (IP) address of a targeted computer in a network in order to gain unauthorized root access and create a backdoor entry path to the target system [1].

IP (or network address) is a software address that is unique to every PC on the Internet. It takes the form of a dot address (e.g., 146.176.151.130) and tells others where and how to find our computers on the network. It is changed when the computer is moved from one network to another. No two computers can have the same IP address at the same time [2].

In IP Spoofing, the Transport Control Protocol/ Internet Protocol (TCP/IP) packet's address is either forged (modified) to conceal the sender's (attacker's) identity or the target computer is impersonated. In this way, the attacker can send a message to computers within the network with the forged address indicating that the message is coming from a legitimate (trusted) member of the network (Fig. 1). This also implies that the attacker gains the privilege of by passing any authentication measures (user name, password, etc) taken to build the network connection [3].

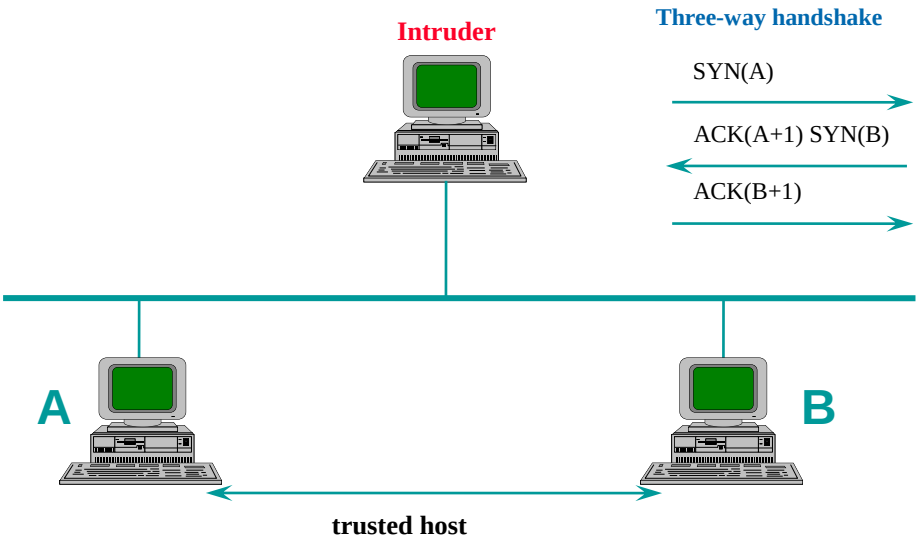

Fig. 1 Attacker impersonating trusted hosts as a legitimate member of the network

Though the attacker never sees the IP datagram (Internet protocol information from network) which the target sends back to a trusted host in the network, he uses the handshaking procedure (Fig. 2) to predict what has been sent and what the response will be. This enables him to manipulate the system in the "blind" [4].

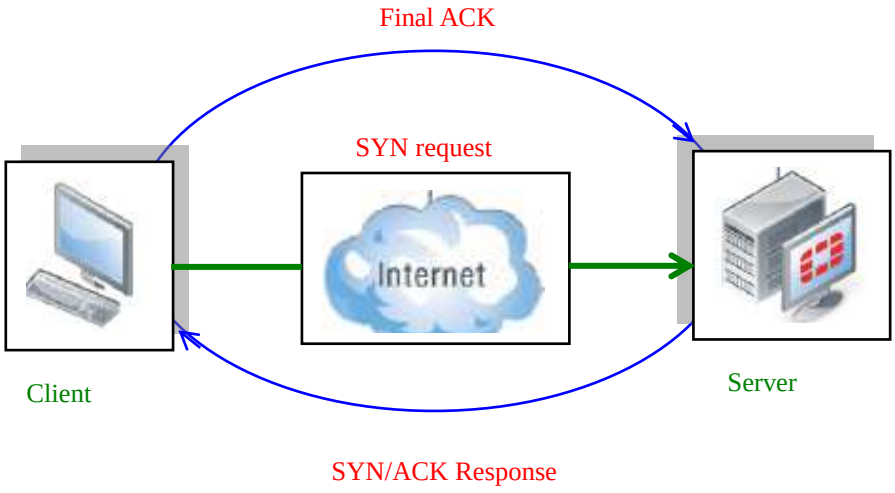

Fig. 2 Normal TCP Handshake

In the 3 way handshake, the 3 messages required before data transmission can take place include the initial sequence number sent by the client (ISNc), the initial acknowledge sequence which the server sends (ISNs), and the acknowledgement number, ACK(ISNs), which the client sends on receipt of ISNs. The normal operation from client, C to server, S is shown:

C → S:SYN(ISNc)

S → C:SYN(ISNs), ACK(ISNc)

C → S:ACK(ISNs)

C → S:data

and/or

S → C:data

Client and server exchange data [4].

If the ISNs generated by a host are predictable, the other end point need not see the SYN response to successfully establish a TCP session. An intruder, X simply finds a client machine that is off, guess the ISN of the server in regular increments, and use rsh program to login:

X (as C) → S: SYN(ISNx), ISN = a (spoofs C)

S → C: SYN(ISNs), ACK(ISNx), ISN = b, ACK = a + 1

X (as C) → S: ACK = b + 1 [spoofs C] (spoofed packet)

X (as C) → S: [echo "**" >>~/.rhosts] (spoofs C)

X (as C) → S: RESET (spoofs C)

rsh and rcp are programs that allow you to login from a remote site without a password. X can now rlogins from anywhere in the world when no one is using that computer (Fig. 3) and send nasty data (reset messages) and possibly cause the S → C message to be lost [4].

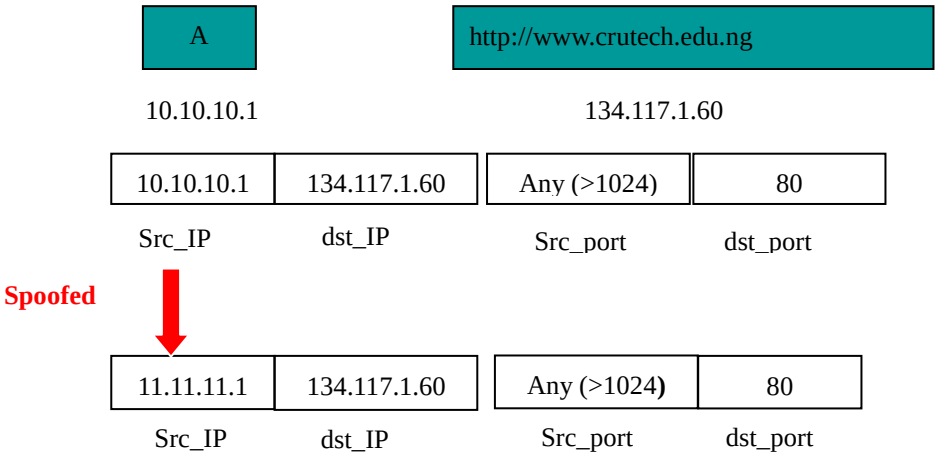

Fig. 3 IP hijacking

ATTACK SIGNATURES

An attacker sits in the middle monitoring traffic passing through a newly arranged bridge path from his computer without the victim realizing that anyone is accessing his computer (Fig. 4). This is called man in the middle attack or connection hijacking [3]. The malicious party intercepts (eavesdrops) a legitimate communication between two hosts. In principle, the attacker uses an unreachable spoofed source IP addresses to send multiple SYN requests to flood the TCP queue of the trusted host waiting for connections. The host's TCP responds with SYN/ACKS to the fake return IP address. Once the queue limit is reached, all other requests to this TCP port will be ignored, and the "trusted host" is effectively disabled [5].

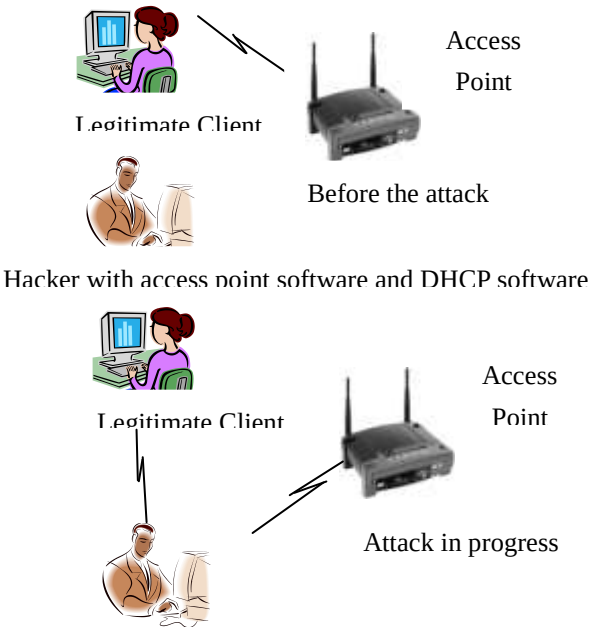

Fig. 4 Man-in-the-middle Attack

The result is that the victim's SYN/ACK is sent to no network, and the victim's system keeps waiting for the ACK from the client. The ACK never arrives, so proper handshakes required by the protocol are not completed (Fig. 3). The victim's server eventually times out waiting; creating a DoS (Denial of Service) event [5]. When attackers hijack a host's IP address in this way, they can steal personal data, load Trojan horses or malware onto the host machine, change system set-ups, delete files, change user passwords, flood the target with massive amounts of information in order to crash the entire network, or even misdirect responses so that no network traffic gets to the trusted host, etc [3]. TCP/IP protocol has no way to check if the source IP address in the packet header actually belongs to the machine sending it. Several users have been accused of accessing unauthorized material because other users have used their IP address. A login system which monitors IP addresses and the files that they are accessing over the Internet cannot be used as evidence against the user, as it is easy to steal IP addresses [3, 5]. The attacker does not get caught because the origin of the messages cannot be determined due to the bogus IP address which mirrors one of the addresses on the network. So, we must guard the privacy of our IP address carefully. Unfortunately, neither the browsers we use to surf the net nor the operating system itself allows us to hide our IP address from the outside world. Some third party software is required to provide this privacy protection [3, 5]. In practice, you are under attack if you use network monitoring software like netlog to discover a packet on your external interface that has both its source and destination IP addresses in your local domain. Another way to detect IP spoofing is to compare the process accounting logs between systems on your internal network. If the IP spoofing attack has succeeded on one of your systems, you may get a log entry on the victim machine showing a remote access whereas there is no corresponding entry for initiating that remote access on the source machine [6]. The very first successful IP-Spoofing attack was launched by Robert Morris using the infamous Internet Worm in April 1989. He used IP Spoofing to forge a TCP packet sequence (numbers) and obtain root access to his targeted system without a user ID or password [3]. Ever since this first attack, the following vulnerability samples have also been made:

SMURF Attack

Here, a broadcast ping packet is sent to a LAN and the source IP of the ping is set the same as the victim's IP address. A huge number of computers will send a reply packet to the victim leading to denial of service [7].

TCP Sequence Number Prediction

A TCP connection is assigned a sequence number for the client and for the server. If the sequence number is predictable, intruders can create packets with forged IP address and guess the sequence number to hijack TCP connections, scan open TCP ports on a target machine to find exploitable services, and even poison Domain Name System (DNS) servers [7]. A number of enhancements for TCP/IP involving heavy use of encryption have been made [6].

OS Fingerprinting

Attackers identify a target Operating System (OS) by sending illegal packets or by analyzing responses by services running on the victim server. Although protocol definitions (RFCs) usually define how a machine should reply to data that it's expecting, these same standards do not always take illegal packets into consideration. So, each OS responds uniquely to invalid inputs and make it possible for hackers to use normal system logging to guess the remote OS without being caught [7]. Major OS have made improvements in their implementations of the protocol stack that mitigate or disable many of the identified attacks [6].

Server Sniffing

If a router is compromised, attackers can usually still sniff network data since a lot of packets flow through routers [4]. Source routing exploitation can be used to "take over" the existing connection, confuse the routing tables on a host or gateway and cause massive denial of service [7].

Password Sniffing

Passwords flowing through the network without using any encryption can be "sniffed" off clear text protocols like Pop3, FTP and Telnet with Protocol analyzers (called sniffers) which set the network card to promiscuous mode meaning that it is able to pass all data on the network to the operating system without filtering. Encryption makes the task of sniffing passwords more difficult but it is still possible to get the passwords from the encrypted data using Dictionary and Brute force attacks. Sniffing is a very effective method for hackers and attackers since it is usually a passive attack and therefore more stealthy and difficult to detect [7].

Packet Sniffing

This is where the hacker listens to TCP/IP packets which come out of the network and steals from the packets such information as user logins, e-mail messages, or credit card number. The hacker listens to a conversation between a server and a client (Fig.5) [1]. The intruder can send dangerous commands to the far end, and all the while translate sequence numbers on packets passed through so as to disguise the intrusion [2].

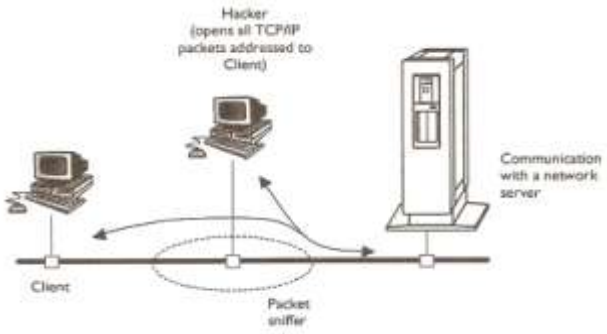

Fig. 5 Packet sniffing

Media Access Control (MAC) Spoofing

The MAC address, a 12 digit hexadecimal number, identifies the physical address of network interface card (NIC). It takes the form: 0000.0E64.5432 or 00-00-0E-64-54-32. It is physically setup in the NIC when it is manufactured and cannot be changed. Like network (IP) address, no two computers on a network can have the same MAC address [1]. Filtering MAC addresses implies including the list of MAC addresses that are allowed (and those that are not allowed) to authenticate with the access point (AP). Attackers can then determine the MAC address of users currently associated with your AP with simple eavesdropping (traffic inspection) tools like CommView for Wi-Fi or Omni Peek personal. This is why MAC filtering is not considered a reliable security solution [7].

VULNERABILITY PROTECTION

A multifaceted distributed approach is recommended to distribute the problem so that it may be more of a real time solution that is effectively managed. Common methods of preventing IP spoofing in network environments are:

Cryptographic Authentication

IP source address authentication is worthless for determined penetrators. Some form of cryptographic authentication is needed. There are several possible approaches. Perhaps the best-known is the Needham Schroeder algorithm. It relies on each host sharing a key with an authentication server; a host wishing to establish a connection obtains a session key from the authentication server and passes a sealed version along to the destination. At the conclusion of the dialog, each side is convinced of the identity of the other. Versions of the algorithm exist for both private key and public key cryptosystems [4].

Some implementations use the Authentication Server as an alternative to address-based authentication. A server can contact a client host's Authentication Server to know the identity of the client. This method uses a second TCP connection not under control of the attacker and is inherently more secure than simple address based authentication. It thus can defeat sequence number attacks and source routing attacks. However, not all hosts are competent to run authentication servers. Again, the authentication message itself can be compromised by routing table attacks. Finally, TCP sequence number spoofing may be used if a target host is down. After the server sends out a TCP open request to the presumed authentication server, the attacker can complete the open sequence and send a false reply. If the target runs a netstat server, this is even easier; as netstat will often supply the necessary sequence numbers with no need to guess. A less obvious risk is that a fake authentication server can always reply "no". This constitutes a denial of service attack [4].

Encryption

Link level encryption, encrypting each packet as it leaves the host computer is an excellent method of guarding against disclosure of information and provides a strong assurance that one can trust the source host's IP address for identification. It also prevents physical intrusions. An attacker who tapped in to an Ethernet cable, for example, would not be able to inject spurious packets. Similarly, an intruder who cut the line to a name server would not be able to impersonate it. However, fast public key cryptosystems need to be employed to secure encrypted broadcast packets from host impersonation [4].

Packet Filtering

Packets flowing through the router or firewall at various stages of processing are examined and their attributes are compared with a set of packet classification rules, called access control lists (ACL). The goal is to permit Hypertext Transport Protocol (HTTP) traffic from networks identified in the access

control entries (e.g. source address = 10.10.10.1, destination address = 134.117.1.60, protocol = TCP and port = 80); and deny all other types of traffic. The access control list (ACL) is then associated with interface Ethernet 0 to perform traffic filtering on inbound packets [8, 9]. The ACL is configured in a consistent manner to act as the intrusion detection system (IDS) and thus enforce corporate security policy. All routers must employ proper IP filtering rules.

Verifying Source Address of Packets from a Forwarding Information Base

An algorithm to discard IP packets that lack verifiable source addresses based on a routing (entry) table is implemented using Unicast Reverse Path Forwarding; so this can be applied only on the input interface of a router at the upstream end of a connection. Cisco Express Forwarding (CEF) switching is enabled on the router and generates the Forwarding Information Base (FIB). The algorithm then checks to see if any packets received at the router interface matches the route in the FIB. It also checks if the source addresses at the receiving interface matches the routing entry for the interface. The router (or switch) forwards the packet as normal if the path is valid but discards the packet if the path is invalid [8]. Routers should only route packets from source that could legitimately come from the interface the packet arrives on. Most routers now have options to turn off the ability to spoof IP source address by checking the source address of a packet against the routing table to ensure the return path of the packet is through the interface it was received on [8, 9].

Filtering IP Traffic on Untrusted Layer 2 Ports

Another preventive measure is to match untrusted interfaces connected to the end user and trusted interfaces connected to the DHCP servers (or any other switch) using dynamic DHCP snooping and static IP source binding. It is supported only in hardware, and is not applied to any traffic processed in software. It is not also supported on private VLANs. Dynamic Host Configuration Protocol (DHCP) or static configuration are used to assign IP addresses of clients from a centralized server [8, 9].

Guarding against Trust Relationships

The best strategy to provide security is to give access to information to the smallest set of people who need it. External machines should not be allowed to query machines in the network to determine what services can be exploited on a given machine. Access to trusted machines should be provided only to the limited set of people. External measures must be taken to verify and/or validate all information that could potentially compromise any protected resource. A routing update that arrives at your doorstep may not just come from another router. It is not safe to assume that a src IP is unspoofable. Good log generation would help, but it is hard to distinguish a genuine intrusion from the routing instability that can accompany a gateway crash. This is a hard problem in general and the focus of modern IDS systems. The use of ssh or VPN/IPSec between hosts and routers can guard against most of these attacks.

CONCLUSION

This paper has examined the challenges posed by deliberate alteration of the source address field in IP headers in today's network environments. Active attacks against the Transport Control Protocol allows the cracker to re-direct the TCP stream through his machine thereby permitting him to bypass the protection offered by such a system. Current intruder activity in spoofing source IP addresses can lead to unauthorized remote root access to systems behind a filtering-router firewall. After gaining root access and taking over existing terminal and login connections, intruders can gain access to remote hosts. They can then steal personal data, load viruses onto the host machine, change system set-ups, delete files, change user passwords, send massive amounts of information to the target machine in order to consume bandwidth and resources, misdirect responses so that no network traffic gets to the trusted host, etc.

The paper further examined various attack scenarios and provided mechanisms for timely mitigation responses. This enables third-party software developers to design and implement defense mechanisms against organized and systematic fraudulent attacks on computer systems in network environments. A multi-track distributed approach is recommended to distribute the problem so that it may be more of a real-time solution that is effectively managed.

REFERENCES

[1] Nkeki, J. I. (2007), "Dictionary of Information Technology", The Inaugural Edition, August28Media Ltd, P. O. Box 4201, Surulere, Lagos, Nigeria, pp. 110, 112.

[2] Buchanan, William (2000), "Distributed Systems and Networks", McGraw-Hill International (UK) Limited, Soppenhangers Road, Mainenhead, Berkshire, SL6 2QL, England, pp 492.

[3] Velasco, Victor (2003), "Introduction to IP Spoofing", SANS Institute, P. O. Box 124, Swansea, SA3 9BB, UK.

[4] Bellovin, S. M. (April 1989), "Security Problems in the TCP/IP Protocol Suite", Computer Communication Review, Vol. 19, No. 2, pp. 32 – 48.

[5] Onah, Ikechukwu F. C. (2006), "Digital Security in Network Environments", Ph.D. Seminar research work, Department of Computer Science, Ebonyi State University, Abakaliki, Nigeria.

[6] Karnati Hemanth, Talluri Ravikiran, Maddipati Venkat Naveen, Thumati Ravi (2012), "Security Problems and Their Defenses in TCP/IP Protocol Suite", International Journal of Scientific and Research Publications, Volume 2, Issue 12, URL: www.ijsrp.org)

[7] Onah, Ikechukwu F. C. and Inyiama, H. C. (2011), "A Survey of Detectable Network Intrusion Signatures", Journal of Science, Engineering and Technology (JSET), 18(3): 10325 – 10339, October 2011.

[8] Cisco ACL Configuration Guide, Release 12.4, Access Control Lists Overview and Guidelines, 2012 Cisco Systems, Inc., URL: http://www.cisco.com/c/en/us/td/docs/ios/12_2/security/configuration/guide/fsecur_c/scfacls.pdf

[9] Cisco URPF Configuration Guide, Release 12.4, Configuring Unicast Reverse Path Forwarding, 2012 Cisco Systems, Inc., URL: http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_data_urpf/configuration/12-4t/sec-data-urpf-12-4t-book.html

Reference to this paper should be made as follows: F. I. Onah *(2015), The Menace of IP-Spoofing Vulnerability in Network Environments and Mitigation Responses. J. of Physical Science and Innovation, Vol. 7, No. 2, Pp. 6 – 13.*