**CHAPTER-1**

**INTRODUCTION**

IP spoofing is the creation of IP packets using somebody else’s IP source addresses. This technique is used for obvious reasons and is employed in several of the attacks discussed later. Examining the IP header, we can see that the first 12 bytes contain various information about the packet. The next 8 bytes contains the source and destination IP addresses. Using one of several tools, an attacker can easily modify these addresses – specifically the “source address” field. A common misconception is that IP spoofing can be used to hide our IP address while surfing the Internet, chatting online, sending e-mail, and so on. This is generally not true. Forging the source IP address causes the responses to be misdirected, meaning you cannot create a normal network connection as shown in Figure1.1.

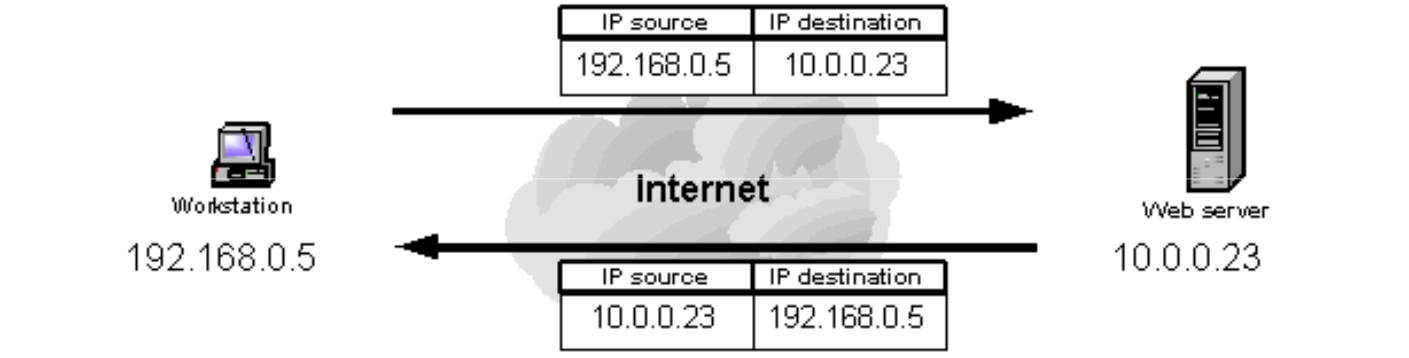


Figure 1.1 Normal network traffic

Valid source IP address, illustrates a typical interaction between a workstation with a valid source IP address requesting web pages and the web server executing the requests. When the workstation requests a page from the web server the request contains both the workstation’s IP address (i.e. source IP address 192.168.0.5) and the address of the web server executing the request (i.e. destination IP address 10.0.0.23). The web server returns the web page using the source IP address specified in the request as the destination IP address (192.168.0.59) and its own IP address as the source IP address (10.0.0.23) as shown in Figure1.2

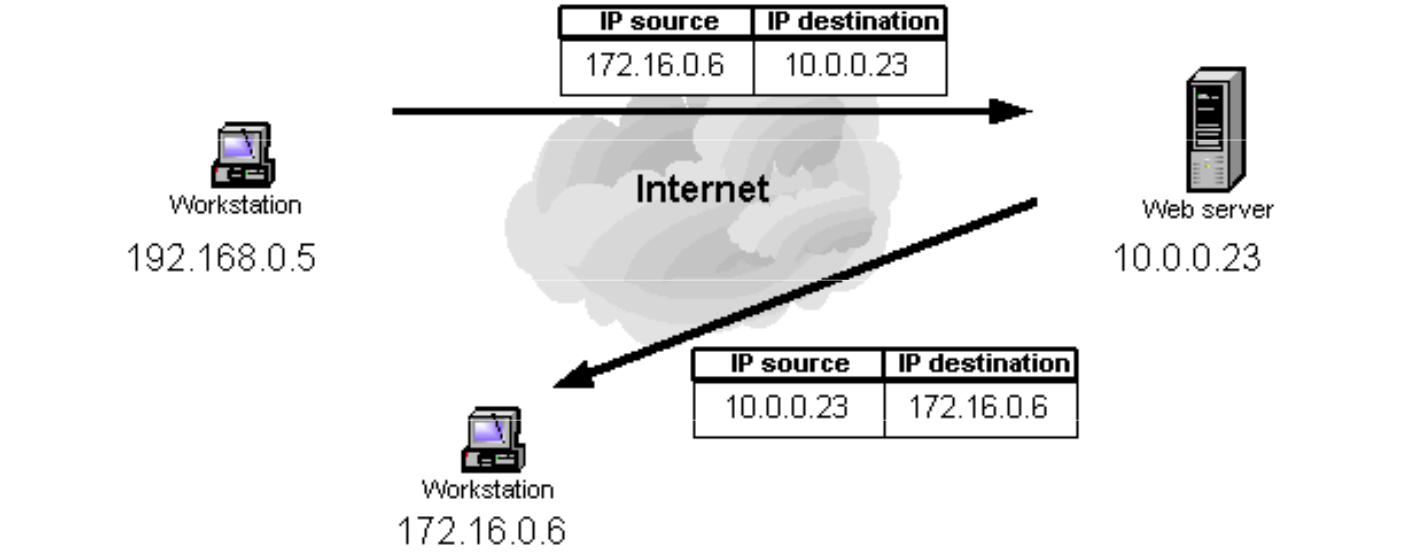


Figure 1.2 Network traffic with spoofed IP address

Spoofed source IP address illustrates the interaction between a workstation requesting web pages using a spoofed source IP address and the web server executing the requests. If a spoofed source IP address (i.e. 172.16.0.6) is used by the workstation, the web server executing the web page request will attempt to execute the request by sending information to the IP address of what it believes to be the originating system (i.e. the workstation at 172.16.0.6). The system at the spoofed IP address will receive unsolicited connection attempts from the web server that it will simply discard.

**CHAPTER-2**

**BRIEF HISTORY OF IP SPOOFING**

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, Figureured 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).

**CHAPTER-3**

**TCP/IP PROTOCOL SUITE**

IP Spoofing exploits the flaws in TCP/IP protocol suite. In order to completely understand how these attacks can take place, one must examine the structure of the TCP/IP protocol suite. A basic understanding of these headers and network exchanges is crucial to the process.

**3.1 INTERNET PROTOCOL – IP**

The Internet Protocol (or IP as it generally known), is the network layer of the Internet. IP provides a connection-less service. The job of IP is to route and send a packet to the packet's destination. IP provides no guarantee whatsoever, for the packets it tries to deliver. The IP packets are usually termed datagrams. The datagrams go through a series of routers before they reach the destination. At each node that the datagram passes through, the node determines the next hop for the datagram and routes it to the next hop. Since the network is dynamic, it is possible that two datagrams from the same source take different paths to make it to the destination. Since the network has variable delays, it is not guaranteed that the datagrams will be received in sequence. IP only tries for a best-effort delivery. It does not take care of lost packets; this is left to the higher layer protocols. There is no state maintained between two datagrams; in other words, IP is connection-less.

The IP Header is shown above. The Version is currently set to 4. In order to distinguish it from the new version IPv6, IP is also referred to as IPv4. The source address and the destination address are 4-byte Internet addresses. The Options field contains various options such as source based routing, and record route. The source based routing allows the sender to specify the path the datagram should take to reach the destination. Record route allows the sender to record the route the datagram is taking. None of the IP fields are encrypted and there no authentication. It would be extremely easy to set an arbitrary destination address (or the source address), and IP would send the datagram. The

destination has no way of ascertaining the fact that the datagram actually originated from an IP address other than the one in the source address field. It is easy to see why any authentication scheme based on IP-addresses would fail.

**3.2 TRANSMISSION CONTROL PROTOCOL – TCP**

IP can be thought of as a routing wrapper for layer 4 (transport), which contains the [Transmission Control Protocol (TCP)](http://www.faqs.org/rfcs/rfc793.html). Unlike IP, TCP uses a connection-oriented design. This means that the participants in a TCP session must first build a connection - via the 3-way handshake (SYN-SYN/ACK-ACK) then update one another on progress - via sequences and acknowledgements. This “conversation”, ensures data reliability, since the sender receives an OK from the recipient after each packet exchange.

As you can see above, a TCP header is very different from an IP header. We are concerned with the first 12 bytes of the TCP packet, which contain port and sequencing information. Much like an IP datagram, TCP packets can be manipulated using software. The source and destination ports normally depend on the network application in use (for example, HTTP via port 80). What's important for our understanding of spoofing are the sequence and acknowledgement numbers. The data contained in these fields ensures packet delivery by determining whether or not a packet needs to be resent. The sequence number is the number of the first byte in the current packet, which is relevant to the data stream. The acknowledgement number, in turn, contains the value of the next expected sequence number in the stream. This relationship confirms, on both ends, that the proper packets were received. It’s quite different than IP, since transaction state is closely monitored.

**3.3 CONSEQUENCES OF THE TCP/IP DESIGN**

Now that we have an overview of the TCP/IP formats, let's examine the consequences. Obviously, it's very easy to mask a source address by manipulating an IP header. This technique is used for obvious reasons and is employed in several of the attacks discussed

below. Another consequence, specific to TCP, is sequence number prediction, which can lead to [session](http://www.owasp.org/asac/auth-session/hijack.shtml) [hijacking](http://www.owasp.org/asac/auth-session/hijack.shtml) or host impersonating. This method builds on IP spoofing, since a session, albeit a false one, is built. We will examine the ramifications of this in the attacks discussed below.

**CHAPTER-4**

**SPOOFING ATTACKS**

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.

**4.1 MITNICK ATTACK**

Mitnick Attack Merry X-mas! Mitnick hacks a Diskless Workstation on December 25th, 1994 The victim – Tsutomu Shinomura The attack – IP spoofing and abuse of trust relationships between a diskless terminal and login serve.

**4.2 SESSION HIJACK**

Session Hijack Alice Bob Eve I’m Bob! I’m Alice!

1. Eve assumes a man-in-the-middle position through some mechanism. For example, Eve could use Arc Poisoning, social engineering, router hacking etc...

2. Eve can monitor traffic between Alice and Bob without altering the packets or sequence numbers.

3. At any point, Eve can assume the identity of either Bob or Alice through the Spoofed IP address. This breaks the pseudo connection as Eve will start modifying the sequence numbers.

**4.3 DOS ATTACK**

DoS Attack Denial of Service (DoS) attack aimed at preventing clients from accessing a service. IP Spoofing can be used to create DoS attacks.

DoS Attack Server Attacker Legitimate Users Interweb Fake IPs Service Requests Flood of Requests from Attacker Server queue full, legitimate requests get dropped Service Requests DoS Attack The attacker spoofs a large number of requests from various IP addresses to fill Services queue. With the services queue filled, legitimate users cannot use the service. DoS becomes more dangerous if spread to multiple computers.

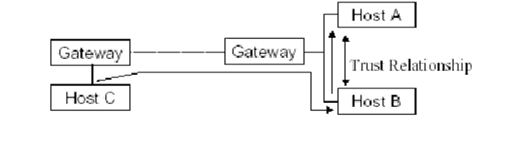
**4.4 NON-BLIND SPOOFING**

This type of attack takes place when the attacker is on the same [subnet](http://www.webopedia.com/TERM/s/subnet.html) 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.

**4.5 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.



**Figure 4.1 Blind Spoofing**

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).

**CHAPTER-5**

**MECHANISM OF THE ATTACK**

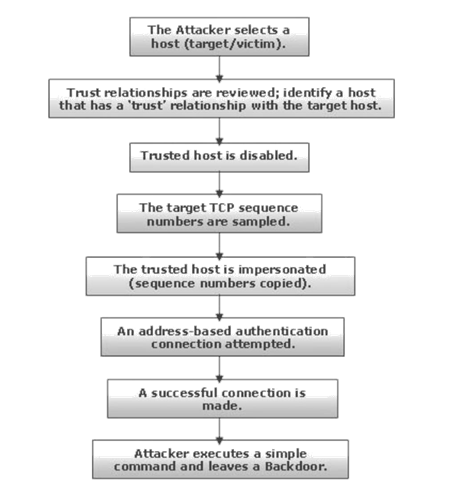


Figure 5.1 Mechanism of the attack

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 DE multiplexed 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). Figureuring out who a host trusts may or may not be easy. A 'show mount 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.

**CHAPTER-6**

**METHODS TO PREVENT IP SPOOFING ATTACK**

**6.1 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 the 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.

**6.2 FILTERING AT THE ROUTER**

If your site has a direct connection to the Internet, you can use your router to help you out. First make sure only hosts on your internal LAN can participate in trust-relationships (no internal host should trust a host outside the LAN). Then simply filter out all traffic from the outside (the Internet) that purports to come from the inside (the LAN).

Implementing ingress and egress filtering on your border routers is a great place to start your spoofing defense. You will need to implement an ACL (access control list) that blocks private IP addresses on your downstream interface. Additionally, this interface should not accept addresses with your internal range as the source, as this is a common spoofing technique used to circumvent firewalls. On the upstream interface, you should

restrict source addresses outside of your valid range, which will prevent someone on your network from sending spoofed traffic to the Internet.

**6.3 ENCRYPTION AND AUTHENTICATION**

Implementing encryption and authentication will also reduce spoofing threats. Both of these features are included in [Ipv6](http://www.webopedia.com/TERM/I/IPng.html), which will eliminate current spoofing threats. Additionally, you should eliminate all host-based authentication measures, which are sometimes common for machines on the same subnet. Ensure that the proper authentication measures are in place and carried out over a secure (encrypted) channel.

**6.4 CRYPTOGRAPHIC METHODS**

An obvious method to deter IP-spoofing is to require all network traffic to be encrypted and/or authenticated. While several solutions exist, it will be a while before such measures are deployed as defacto standards.

**CHAPTER-7**

**APPLICATIONS OF IP SPOOFING**

**7.1 ASYMMETRIC ROUTING (SPLITTING ROUTING)**

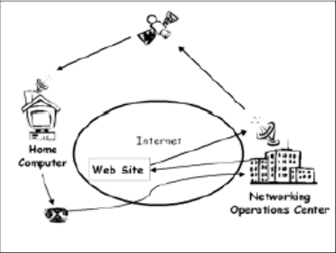
Asymmetric routing means traffic goes over different interfaces for directions in and out. In other words, asymmetric routing is when the response to a packet follows a different path from one host to another than the original packet did. The more correct and more general answer is, for any source IP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



Figure 7.1 Valid source IP address

**7.2 SAT DSL**

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



**Figure7.1. 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).

The traffic path of satellite DSL is:

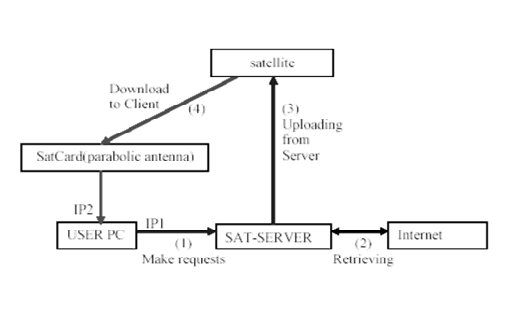


Figure7.2. Traffic path of satellite DSL

**7.3 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.

**7.4 TCP AND IP SPOOFING TOOLS**

|  |  |  |
| --- | --- | --- |
| 1) | Mendax | Mendax for Linux |
| is an easy-to-use tool for TCP sequence number prediction |
| 2) | spoofit.h | spoofit.h |
| is a nicely commented library for including IP spoofing |
|  |  | functionality into your programs. |
| 3) | Ipspoof | Ipspoof |
| is a TCP and IP spoofing utility. |
| 4) | hunt | Hunt |
| is a sniffer which also offers many spoofing functions. |
| 5) | Dsniff | Dsniff |
| is a collection of tools for network auditing |
| and penetration testing. |

**CHAPTER-8**

**ADVANTAGES & DISADVANTAGES**

**ADVANTAGES**

Freedom of spoofing. The attacker is not bounded by a specific range of IPs. No wasted or unneeded initiated packets. The attacker sends one TCP/UDP packet per port.No tracing of the original scanner. Detection of the scanning machine isimpossible at the IP layer.

**DISADVANTAGES**

No replies. There will be no reply packets arriving at the scanning machine.

No results. Since replies are not received, the attacker won’t know port status.

**CHAPTER 9: 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.

**REFERENCE**

1. http://www.scs.carleton.ca/~dlwhyte/whytepapers/ipspoof.htm
2. http://www.suse.de/~mha/linux-ip-nat/diplom/node4.html
3. http://keskus.hut.fi/tutkimus/ipana/paperit/qosr/s130-qosr-asymmetric.pdf
4. http://www.infosys.tuwien.ac.at/teaching/courses/inetsec/slides/sli.pdf
5. http://www.research.att.com/~smb/papers/ipext.pdf
6. http://www.firewall.cx/nat-intro.php
7. [http://zork.net/~phil/cracking/internet.html](http://zork.net/~phil/Cracking/Internet.html)
8. http://zork.net/~phil/cracking/internet.html ip spoofing
9. http://bear.cba.ufl.edu/teets/projects/ism6222f102/perryna/index.html