Cyber Attacks Explained: Packet Spoofing

Last month, we started this series to cover the important cyber attacks that impact critical IT infrastructure in organisations. The first was the denial-of-service attack, which we discussed in detail. This month, we will cover the packet-spoofing attack, which is found to be a favourite among hackers, and widely used in exploiting network vulnerabilities. We will also discuss how this type of attack affects Linux systems, and how to mitigate the risks.

Spoofing, by definition, means to imitate or trick someone. To understand the spoofing attack, we need to examine the IP packet structure in detail. Many cyber attacks stem from design flaws in the fundamental network designs; packet spoofing is no exception. Please refer to Figure 1.

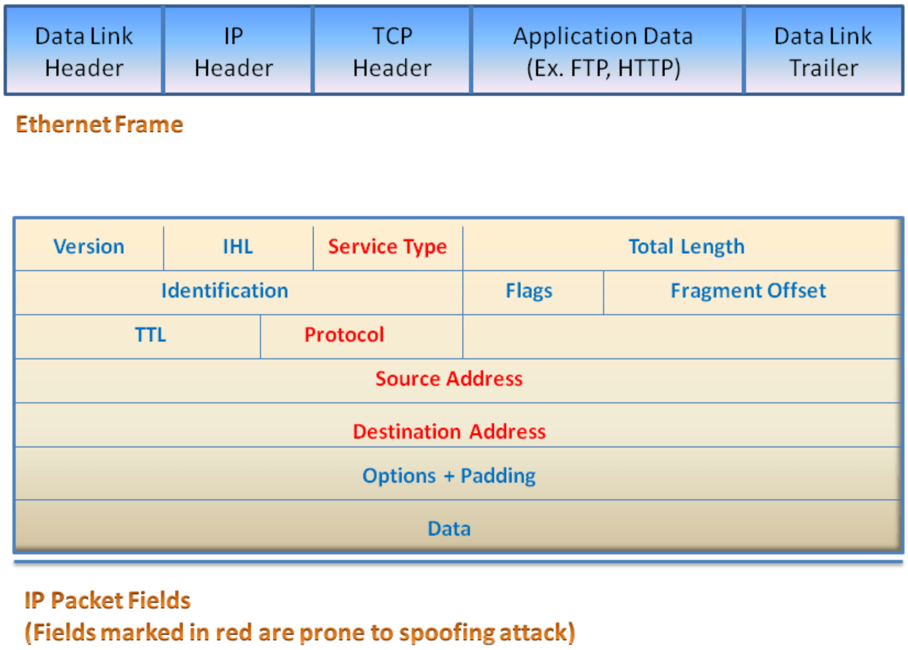

Figure 1: Ethernet network packet

As we know, a basic Ethernet network packet is essentially a data chunk with various predefined fields such as source and destination MAC addresses, frame checksum code, preambles, etc. In the case of the TCP/IP protocol, the TCP frame encapsulates the IP datagram, and both piggyback on the basic Ethernet packet. The TCP provides the connection-oriented information, while the IP packet contains source and destination addresses and ports. It is the duty of various OSI layers to contribute their bit towards the formation of a complete packet.

As we learnt last month the TCP/IP works on a three-way handshake (SYN, SYN-ACK and ACK),. This handshake establishes a connection between two different network interface cards, which then use the packet sequencing and data acknowledgements to send or receive data. The conversation is formally concluded by using a FIN/FIN-ACK handshake.

The source and IP address field values decide who is talking to whom, while the port fields decide which source application is talking to which destination application. IP address fields in the IP packet are filled up automatically by the upper layers; however, malicious users or programs can modify these fields — and this is exactly where the spoofing comes into the picture (refer to Figure 2).

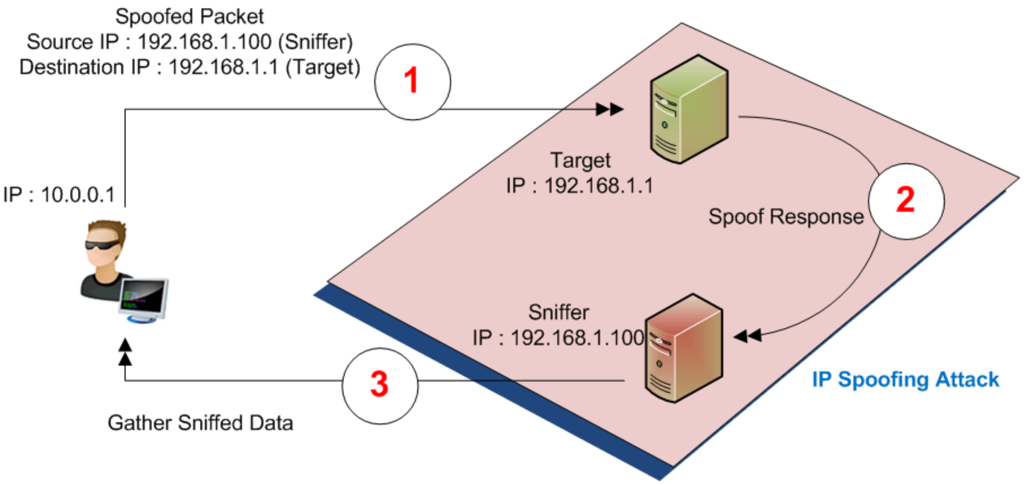

Figure 2: Packet-spoofing attack

In a very simple packet-spoofing attack, the hacker creates IP packets targeting a destination, but the source IP field is modified so that it does not have the IP address of the hackers' computer, but in fact, of some other computer, which can be used as a data collector or a sniffer by the hackers.

By its design, the TCP/IP stack running on the destination computer is supposed to respond to the source IP of the received packet. On doing so, the sniffer computer receives the packet. Having said that, if a packet-monitoring software is run on the destination computer, it will show that the packets are coming from this sniffer computer, which in reality is not the case. Thus, the hackers are successful in hiding themselves, and at the same time collecting packet information from the sniffer computer, because their own real source IP is nowhere revealed in this process.

Now let’s understand why hackers prefer spoofing. While hiding on the network is the very first step, a hacker’s intention is always beyond that. By collecting packets on the sniffer computer, hackers can gain a lot of information about the target machine. Vital information, such as open ports, type of operating system, layer 7 applications, cryptography types used, etc, can be collected, while not revealing one’s identity.

Just as an example, a hacker can send a spoofed packet to find out if the target is running a Web server. Once the port information is revealed on the sniffer, another chunk of spoofed packets can be sent to the target to establish a telnet session, and check Web server headers, which would reveal OS information as well as the type of Web server and security support information.

Please note that the hackers need not always have access to the sniffer machine; in fact, using an advanced packet-sniffing tool can help the attackers use only their own machine, send spoofed packets, collect data on the same machine, and still be invisible on the network from an attack viewpoint.

Advanced hackers can use spoofing to find out the state of a firewall too. As we know, the firewall is a stateful packet-filtering process, which works on a set of configurable rules, to decide which packet should be allowed, and which dropped. When a source machine tries to connect to another machine that is protected behind a firewall, the source machine has to first establish a TCP handshake with the firewall.

If the connection is allowed, the firewall itself establishes another connection with the destination machine, and the data transfer occurs with the firewall as a catalyst in this whole process. Also, when the destination machine tries to send an acknowledgement to the source, the firewall is bound to intercept and check whether or not a corresponding request from the source is pending. If it is not, the packet is dropped. All this is true for modern stateful firewalls, but not for old or cheap firewalls, which tend to allow ACK packets, thus exposing spoofing possibilities.

There are three reasons worth mentioning here, which typically make a network vulnerable to spoofing attacks. The first is, the IP packets can be modified very easily using free utilities available. The second reason is that even today, many applications still use source and destination IP address combinations as a secure way of authenticating a packet.

Just as an example, there are Web servers that allow or disallow HTTP requests from a list of configurable IP addresses. Such systems prove to be useless if the packet is spoofed to reflect an IP address which is in the list of allowed IPs. The third reason is that the routers forward traffic based on the destination address, and by default do not care much about the source address.

Packet spoofing types

With this basic knowledge of how spoofing works, let us now dig a bit deeper. At a broad technical level, there are two basic types of spoofing: “blind” spoofing and “non-blind” spoofing. As we discussed earlier, the packet sequence number and acknowledgements help data transfer, so sensing those fields is an important requirement of a spoofing-based attack.

In the blind spoofing type of attack, the attacker sends multiple packets to the target to sample the sequencing numbers. Once the pattern is found out, it becomes easy for the attacker to create another set of spoofed packets, to collect the data being transferred. Whereas, in the non-blind type, the attackers must be on the same subnet as the target, to be able to easily see the sequencing numbers and acknowledgements. Once they get their hands on it, they can break the established connection between the target and the other computer to which it is talking, and re-establish the connection by modifying the sequence numbers to that of the attackers’ own machine.

An extended version of this type of attack is the man-in-the-middle attack, whereby an entire session is stolen to decipher and steal data.

Now let’s take a look at how the basic concept of spoofing could be used by hackers to impact various TCP-based application services.

IP port spoofing

In this type, the source port is modified in order to cheat the NAT devices and firewalls, and also to hide deep in the network. A firewall that is not very well managed can end up leaving stale rules in action, which can potentially lead certain outgoing ports on certain IP addresses to be open. IP port spoofing attacks take advantage of this situation.

This is a serious attack, because the hacker can be outside the network, and still be able to look at internal traffic.

ARP spoofing

Since it is all about modifying or forging packets, modern attackers don't restrict themselves to IP fields. In this type of attack, which is also known as ARP poisoning, the attackers send spoofed ARP packets on the local area network, to associate the attackers' own machine's MAC address with the IP address of another host, which is the target.

Due to this, traffic for the target now reaches the attackers' machine, due to the binding between IP and MAC addresses. Thus this becomes a local man-in-the-middle attack. The attackers can further hide by forwarding received data to the actual destination, and since there is no data lost in this process, it is almost impossible to find the man-in-the-middle stealing the data.

DNS spoofing

Also known as a DNS poisoning attack, this is more serious. In this case, the domain name system server is spoofed to alter entries of domain names to reflect the attackers' IP address. This results in sending Web or email traffic to the attackers' machine. This attack is achieved by creating multiple forged packets wherein the IP, port and service type entries are modified to serve the purpose.

The repercussions of such an attack are very dangerous, whereby a website can be defaced, or email data can be stolen.

Email and Web spoofing

Speaking about Layer 7, the same technique of impersonation can be stretched to crafting fake email addresses, Web requests and hyperlinks. This is usually done to plant a Trojan or a virus and spread it across. Please remember that spoofing is a basic component in planting a denial-of-service attack, mainly because the attacker would always want to hide behind a curtain.

Protecting FOSS systems

While the spoofing attack is a difficult one to tackle, there are a few preventive mechanisms that all network administrators should adopt in their infrastructure. Since the spoofing starts at Layer 2, the real protection has to be implemented in the critical network components such as routers, firewalls, switches, etc.

Deploying the latest stateful firewall, and enabling features in it that fight against source packet spoofing, can be a first level of defense. As a daily chore, network logs from firewalls, routers and switches can be parsed using a script, to see if multiple duplicate ACK packets are being generated.

Also, if an unusual number of SYN packets are being observed, which are not being responded to, that could be a clue to spoofing. The real spoofing detection is to check a bunch of packets for a given set of source and destination IP addresses, and see that there is no other IP address involved in that communication session.

This is a difficult task for a script program to achieve, and for complex networks, deploying an intrusion detection device usually helps to a great extent. There is a thumb rule while inspecting packets, which says that in a network packet log, if an internal LAN IP address is being shown in the log file capturing data for an external interface, that indicates there is a problem.

Linux/FOSS systems come with a built-in, but usually less-known feature called source address verification. It is a kernel feature which, when turned on, starts dropping packets that appear to be arriving from the internal network, but in reality are not. Most of the latest kernels, such as Ubuntu and CentOS, do support it, but if your Linux distro does not, it is time to upgrade. Modifying the hosts.conf file to add nospoof on is another level of defense to try.

In terms of detection, for smaller Linux networks, a nice utility called arpwatch is very useful. This keeps track of MAC and IP addresses, records all changes, and can be scripted to alert administrators about a possible attack. Scripting can also be done to go through network interface logs and look for anomalies in terms of source address forging.

Summary

Packet spoofing is a difficult type of attack to tackle. It can result in serious data loss, and there are ways to detect it and stop it. Configuring firewalls, switches and routers is an important step to prevent networks from spoofing, and network administrators should know about it.

Finance firms are found to be common victims to this kind of attack, and their IT management teams should take the necessary steps to prevent financial losses or damage to their reputation. Implementing IPS devices certainly helps in getting control over the IT network infrastructure security.