4.7. Sniffing

A sniffer is a packet-capturing or frame-capturing tool. It basically captures and displays the data as it is being transmitted from host to host on the network. Generally a sniffer intercepts traffic on the network and displays it in either a command-line or GUI format for a hacker to view. Most sniffers display both the Layer 2 (frame) or Layer 3 (packet) headers and the data payload. Some sophisticated sniffers interpret the packets and can reassemble the packet stream into the original data, such as an email or a document.

Sniffers are used to capture traffic sent between two systems, but they can also provide a lot of other information. Depending on how the sniffer is used and the security measures in place, a hacker can use a sniffer to discover usernames, passwords, and other confidential information transmitted on the network. Several hacking attacks and various hacking tools require the use of a sniffer to obtain important information sent from the target system. This chapter will describe how sniffers work and identify the most common sniffer hacking tools.

Understanding Host-to-Host Communication

All Host-to-Host network communications is based upon the TCP/IP Data Communications Model. The TCP/IP Model is a 4 layer model. The TCP/IP Model maps to the older OSI model with 7 layers of data communication. Most applications use the TCP/IP suite for host-to-host data communications.

In normal network operations, the application layer data is encapsulated and a header containing address information is added to the beginning of the data. An IP header containing source and destination IP address are added to the data as well as a MAC header containing source and destination MAC addresses. IP addresses are used to route traffic to the appropriate IP network, and the MAC addresses ensure the data is sent to the correct host on the destination IP network. In this manner, traffic is sent from source host to destination host across the Internet and delivery to the correct host is ensured. The postal system works much the same way. Mail is routed to the appropriate area using the zip code, and then the mail is delivered within the zip code to the street and house number. The IP address is similar to the zip code to deliver mail to

the regional area, and the street and house numbers are like the MAC address of that exact station on the network.

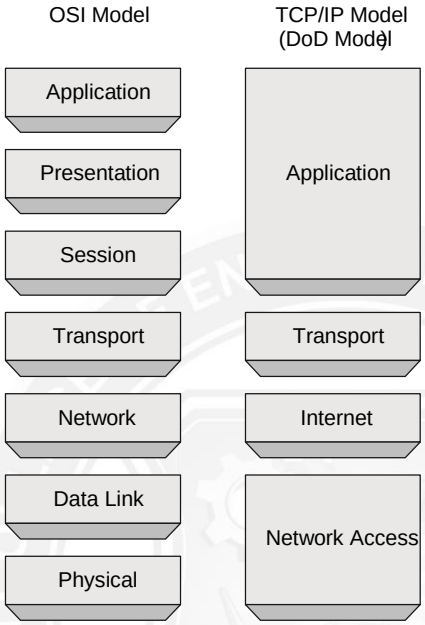
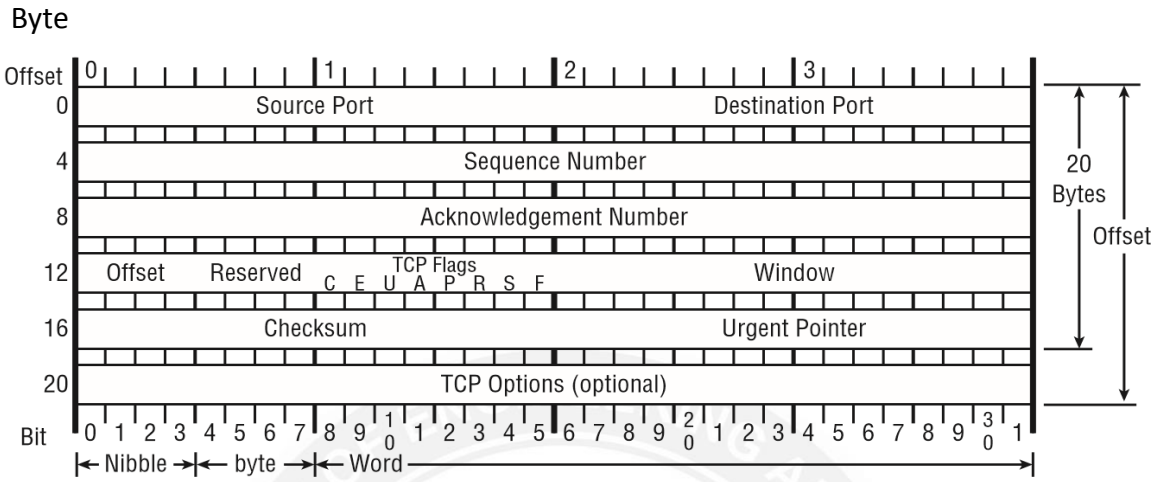


Fig: TCP/IP Model

The address system ensures accurate delivery to the receiver. In normal network operations, a host should not receive data intended for another host as the data packet should only be received by the intended receiver. Simply said, the data should only be received by the station with the correct IP and MAC address. However, we know that sniffers do receive data not intended for them.

In addition to understanding network addresses, it is also important to understand the format of the TCP Header. Figure shows the TCP Header format.



Understanding Host-to-Host Communication

Fig: TCP Header Format

The TCP Header is comprised of the following fields:

Source Port: 16 bits The source port number.

Destination Port: 16 bits The destination port number.

Sequence Number: 32 bits The sequence number of the first data octet in this segment (except when SYN is present). If SYN is present the sequence number is the initial sequence number (ISN) and the first data octet is ISN+1.

Acknowledgment Number: 32 bits If the ACK control bit is set this field contains the value of the next sequence number the sender of the segment is expecting to receive.

Data Offset: 4 bits The number of 32 bit words in the TCP Header. This indicates where the data begins.

Reserved: 6 bits Reserved for future use. Must be zero.

Control Bits: 6 bits

- ✓URG: Urgent Pointer field significant
- ✓ACK: Acknowledgment field significant
- ✓PSH: Push Function

- ✓RST: Reset the connection
- ✓SYN: Synchronize sequence numbers
- ✓FIN: No more data from sender

Window: 16 bits The number of data octets beginning with the one indicated in the acknowledgment field which the sender of this segment is willing to accept.

Checksum: 16 bits The checksum field is a computation of all fields to ensure all data was received and the data was not modified in transit.

Urgent Pointer: 16 bits This field communicates the current value of the urgent pointer as a positive offset from the sequence number in this segment. The urgent pointer points to the sequence number of the octet following the urgent data. This field is only be interpreted in segments with the URG control bit set.

Options: variable Options may occupy space at the end of the TCP header and are a multiple of 8 bits in length.

When referring to the length of the fields in the TCP Header, 8 bits comprises a single byte. A Nibble is less than a byte and a Word is more than a byte.

How a Sniffer Works

Sniffer software works by capturing packets not destined for the sniffer system's MAC address but rather for a target's destination MAC address. This is known as *promiscuous mode*. Normally, a system on the network reads and responds only to traffic sent directly to its MAC address. However, many hacking tools change the system's NIC to promiscuous mode. In promiscuous mode, a NIC reads all traffic and sends it to the sniffer for processing. Promiscuous mode is enabled on a network card with the installation of special driver software. Many of the hacking tools for sniffing include a promiscuous-mode driver to facilitate this process. Not all Windows drivers support promiscuous mode, so when using hacking tools ensure that the driver will support the necessary mode.

Any protocols that don't encrypt data are susceptible to sniffing. Protocols such as HTTP, POP3, Simple Network Management Protocol (SNMP), and FTP are most commonly captured

using a sniffer and viewed by a hacker to gather valuable information such as usernames and passwords.

There are two different types of sniffing: passive and active. *Passive sniffing* involves listening and capturing traffic, and is useful in a network connected by hubs; *active sniffing* involves launching an Address Resolution Protocol (ARP) spoofing or traffic-flooding attack against a switch in order to capture traffic. As the names indicate, active sniffing is detectable but passive sniffing is not detectable.

In networks that use hubs or wireless media to connect systems, all hosts on the network can see all traffic; therefore, a passive packet sniffer can capture traffic going to and from all hosts connected via the hub. A switched network operates differently. The switch looks at the data sent to it and tries to forward packets to their intended recipients based on MAC address. The switch maintains a MAC table of all the systems and the port numbers to which they're connected. This enables the switch to segment the network traffic and send traffic only to the correct destination MAC addresses. A switch network has greatly improved throughput and is more secure than a shared network connected via hubs.

Another way to sniff data through a switch is to use a span port or port mirroring to enable all data sent to a physical switch port to be duplicated to another port. In many cases, span ports are used by network administrators to monitor traffic for legitimate purposes.

Sniffing Countermeasures

The best security defense against a sniffer on the network is encryption. Although encryption won't prevent sniffing, it renders any data captured during the sniffing attack useless because hackers can't interpret the information. Encryption such as AES and RC4 or RC5 can be utilized in VPN technologies and is commonly used to prevent sniffing on a network.

Bypassing the Limitations of Switches

Because of the way Ethernet switches operate, it is more difficult to gather useful information when sniffing on a switched network. Since most modern networks have been upgraded from hub to switches, it takes a little more effort to sniff on a switched network. One of the ways to do that is to trick the switch into sending the data to the hackers' computer using ARP poisoning.

How ARP Works

ARP allows the network to translate IP addresses into MAC addresses. When one host using TCP/IP on a LAN tries to contact another, it needs the MAC address or hardware address of the host it's trying to reach. It first looks in its ARP cache to see if it already has the MAC address; if it doesn't, it broadcasts an ARP request asking, "Who has the IP address I'm looking for?" If the host that has that IP address hears the ARP query, it responds with its own MAC address, and a conversation can begin using TCP/IP.

ARP poisoning is a technique that's used to attack an Ethernet network and that may let an attacker sniff data frames on a switched LAN or stop the traffic altogether. ARP poisoning utilizes ARP spoofing, where the purpose is to send fake, or spoofed, ARP messages to an Ethernet LAN. These frames contain false MAC addresses that confuse network devices such as network switches. As a result, frames intended for one machine can be mistakenly sent to another (allowing the packets to be sniffed) or to an unreachable host (a denial-ofservice, or DoS, attack). ARP spoofing can also be used in a man-in-the-middle attack, in which all traffic is forwarded through a host by means of ARP spoofing and analyzed for passwords and other information.

ARP Spoofing and Poisoning Countermeasures

To prevent ARP spoofing, permanently add the MAC address of the gateway to the ARP cache on a system. You can do this on a Windows system by using the ARP -s command at the command line and appending the gateway's IP and MAC addresses. Doing so prevents a hacker from overwriting the ARP cache to perform ARP spoofing on the system but can be difficult to manage in a large environment because of the number of systems. In an enterprise environment, port-based security can be enabled on a switch to allow only one MAC address per switch port.

Understanding MAC Flooding and DNS Spoofing

A packet sniffer on a switched network can't capture all traffic as it can on a hub network; instead, it captures traffic either coming from or going to the system. It's necessary to use an additional tool to capture all traffic on a switched network. There are essentially two ways to perform active sniffing and make the switch send traffic to the system running the sniffer:

ARP Spoofing This method involves using the MAC address of the network gateway and consequently receiving all traffic intended for the gateway on the sniffer system. A hacker can also *flood* a switch with so much traffic that it stops operating as a switch and instead reverts to acting as a hub, sending all traffic to all ports. This active sniffing attack allows the system with the sniffer to capture all traffic on the network.

DNS Spoofing (or DNS Poisoning) This is a technique that tricks a DNS server into believing it has received authentic information when in reality it hasn't. Once the DNS server has been poisoned, the information is generally cached for a while, spreading the effect of the attack to the users of the server. When a user requests a certain website URL, the address is looked up on a DNS server to find the corresponding IP address. If the DNS server has been compromised, the user is redirected to a website other than the one that was requested, such as a fake website.

To perform a DNS attack, the attacker exploits a flaw in the DNS server software that can make it accept incorrect information. If the server doesn't correctly validate DNS responses to ensure that they come from an authoritative source, the server ends up caching the incorrect entries locally and serving them to users that make subsequent requests.

This technique can be used to replace arbitrary content for a set of victims with content of an attacker's choosing. For example, an attacker poisons the IP address's DNS entries for a target website on a given DNS server, replacing them with the IP address of a server the hacker controls. The hacker then creates fake entries for files on this server with names matching those on the target server. These files may contain malicious content, such as a worm or a virus. A user whose computer has referenced the poisoned DNS server is tricked into thinking the content comes from the target server and unknowingly downloads malicious content.

The types of DNS spoofing techniques are as follows:

Intranet Spoofing Acting as a device on the same internal network

Internet Spoofing Acting as a device on the Internet

Proxy Server DNS Poisoning Modifying the DNS entries on a proxy server so the user is redirected to a different host system

DNS Cache Poisoning Modifying the DNS entries on any system so the user is redirected to a different host