International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 11 Issue 2 –NOVEMBER 2014.

PACKET SNIFFING

S.Vasu

Lecturer, Dept of Computer Science, KBN College, Vijayawada, A.P, India

Abstract— Packet sniffing, it is very commonly used by prying hackers. We will look at a few tools typically used as sniffers and also figure out how to protect IT infrastructure from such attacks. Sniffing involves capturing, decoding, inspecting and interpreting the information inside a network packet on a TCP/IP network. The purpose is to steal information, usually user IDs, passwords, network details, credit card numbers, etc. Sniffing is generally referred to as a "passive" type of attack, wherein the attackers can be silent/invisible on the network. This makes it difficult to detect, and hence it is a dangerous type of attack. As we already learnt over the previous months, the TCP/IP packet contains vital information required for two network interfaces to communicate with each other. It contains fields such as source and destination IP addresses ports, sequence numbers and the protocol type. Each of these fields is crucial for various network layers to function, and especially for the Layer 7 application that makes use of the received data. By its very nature, the TCP/IP protocol is only meant for ensuring that a packet is constructed, mounted on an Ethernet packet frame, and reliably delivered from the sender to the receiver across networks. However, it does not by default have mechanisms to ensure data security.

Key Words: Hacking, Network, Snipping

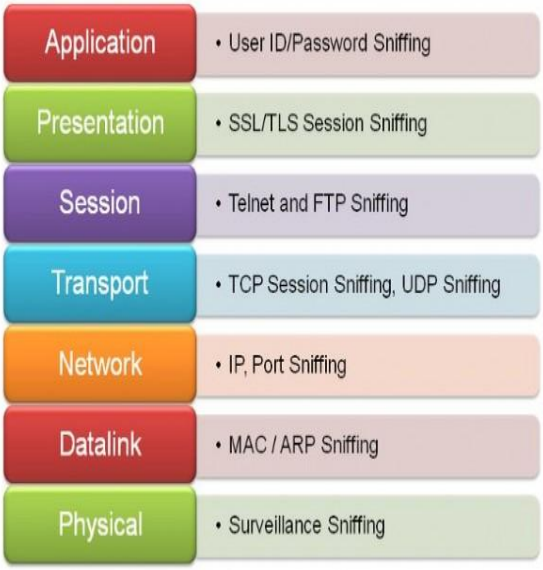

Figure1. Shows the OSI layers and the information a hacker can steal at each layer by successfully sniffing a network.

Figure1:

I. INTRODUCTION

A packet sniffer is a utility that has been used since the original release of Ethernet. Packet sniffing allows individuals to capture data as it is transmitted over a network. Packet sniffer programs are commonly used by network professionals to help diagnose network issues and are also used by malicious users to capture unencrypted data like passwords and usernames in network traffic. Once this information is captured, the user can then gain access to the system or network.If you want to keep information confidential or are concerned about packet sniffing, it is advised that you work on encrypted protocols and encrypt all sensitive data, such as e-mails, being sent over the Internet or network. A great encryption program is PGP, users who are using Telnet should consider using SSH instead.

To understand why hackers sniff, we need to know what they can get from the network.

The sniffing process is used by hackers either to get information directly or to map the technical details of the network in order to create a further attack. Hackers are always in favour of sniffing, because it can be done for a longer time without getting caught.

How do they 'sniff'?

Network sniffing uses sniffer software, either open source or commercial. Broadly, there are three ways to sniff a network, as shown in Figure 2.

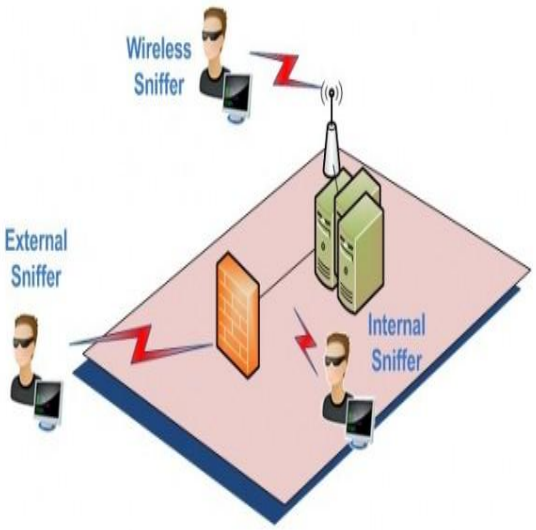

Figure2:

It is important to remember that sniffing can range from Layer 1 through Layer 7. Talking about physical connectivity, a person (who may be an employee of the firm) who is already hooked up to the internal LAN can run tools to directly capture network traffic. Using spoofing techniques, a hacker outside the target network can intercept packets at the firewall level and steal the information. In the latest form of packet sniffing, wide usage of wireless networks has made it easy to sit near the network and penetrate it to get information.

Regardless of where the hackers are located on the network being sniffed, they use packet capturing or packet sniffer software. Modern packet sniffers are supposed to be used for troubleshooting network problems, but can be used for hacking too

1. Packet capturing
2. Network traffic usage and analysis
3. Packet conversion for data analysis
4. Network troubleshooting

Unethical usage

1. User identity and password stealing
2. Email or instant message data stealing
3. Packet spoofing and data theft
4. Monetary or reputational damage

Regarding the technical details of how sniffing is done, we need to remember that packet capturing software always runs in promiscuous mode, whereby it is capable of intercepting and storing all packets on a network. This also means that, even though the packet is not meant for the network interface on which the sniffer is running, it is captured, stored and analysed.

Sniffer software contains its own network driver and buffer memory in order to capture a large chunk of packets. Modern sniffers are capable of analysing the captured packets and converting them into sensible statistical information. Now let's discuss a few ways of sniffing a network, to understand how hackers get what they want. A LAN sniff A sniffer deployed on an internal LAN can scan the entire IP range promiscuously. This helps in providing further details such as live hosts, open ports, server inventory, etc. Once a list of open ports is gathered, a port-specific vulnerability attack is possible. A protocol sniff This method involves sniffing data related to the network protocols being used. First, a list of protocols is created based on the captured data. This is further segregated to create special sniffers for each attack. For example, in a network sniff capture, if the ICMP protocol is not seen, it is assumed to be blocked. However, if UDP packets are seen, a separate UDP sniffer is started to capture and decipher Telnet, PPP, DNS and other related application details.

An ARP sniff

In this popular method, the hacker captures a lot of data in order to create a map of IP addresses and the associated MAC addresses. Such a map is further used to create ARP poisoning attacks, packet spoofing attacks, or to dig into router-based vulnerabilities.

TCP session stealing

This method is a very basic form of sniffing, in which a network interface in promiscuous mode captures traffic between a source and a destination IP address. Details such as port numbers, service types, TCP sequence numbers and the data itself are of interest to hackers. Upon capturing enough packets, advanced hackers can create fabricated TCP sessions to fool the source and destination, and be the man in the middle to take over the TCP session.

II. APPLICATION-LEVEL SNIFFING

Usually, from the data packets sniffed and captured, a few intricate application details are found out for information stealing or to create further attacks. As an example, the capture file can be parsed to perform OS fingerprinting, SQL query analysis, reveal application specific TCP port data information, etc. In another approach, creating a mere list of applications running on a server is good enough to plan an application-specific attack on it.

Web password sniffing

As the name suggests, HTTP sessions are stolen and parsed for user ID and password stealing. While the Secure Socket Layers (SSL) are incorporated for securing HTTP sessions on the network, there are numerous internal websites that still use standard but less secure encryption. It is easy to capture Base64 or Base128 packets and run a deciphering agent against it to crack the password. In modern sniffers, SSL sessions can also be captured and parsed for information, though this method is not very easy.

Detecting sniffers

As mentioned earlier, since sniffers work silently, it is very difficult to detect them on a network. There are, however, a

few tricks that can provide a clue to a possible sniffer presence. There are two ways to detect a sniffer — host-based and network-based.

In host-based detection, you can use small utilities to detect if the NIC is running in a promiscuous mode on any host in a network. Since the basic requirement for a sniffer to work is to put the network interface in "read all" mode, disabling it can very effectively help shutting down stray sniffers.

In case of network-based detection, anti-sniffer software can be run to detect the presence of specific signature packets. In another approach, scripts can be run to check each network host for the presence of known sniffers, processes, etc. Modern anti-virus or anti-spyware software are capable of detecting sniffing software and disabling it.

III. PROTECTION FROM SNIFFERS

While the very first step should be to design a tight perimeter defense system while creating network architecture, there are a few methods that could be deployed to make the infrastructure less sniffer-prone. The following tricks help achieve that to a great extent.

Disabling promiscuous mode on network interfaces results in shutting down most sniffer software. This can be done by running an admin script as a daily job on the network, or deploying a network policy at the host level to control access to the network card configuration settings.

Using switched networks can reduce the possibility of a sniffer running on the network. Unlike in a network hub, in a switched network the packets are delivered to the destination and are not visible to all nodes — thus reducing the chances of someone sniffing it on the way. Also, for network administrators it becomes easy to detect sniffers by focusing on the switched network segments, one at a time.

Anti-sniffing tools can be used to detect the network interface mode, as well as various processes and software present on servers or network hosts. Modern intrusion detection systems have this as an integrated feature.

IPsec encryption can be used for tokenbased packet security in the network infrastructure, if the data is of a confidential nature. IPSec provides data encapsulation and encryption of high standards, and is available on modern routers, firewalls and other network components. Almost all operating systems do support IPSec, and it is widely used in serious IT infrastructure. For session layer protection, SSL and TLS can be used to encrypt traffic.

IV. PROTECTING FOSS SYSTEMS

Let us now look at a few sniffer products, in order to learn about what is used in the FOSS world today. Linux systems use the tcpdump utility, which is an excellent builtin sniffer to capture and store TCP packets. As for third-party open source tools, Wireshark (Ethereal) is very famous due to its GUI interface, and packetfiltering and viewing capabilities. Sniffit, DSniff and Ettercap are similar utilities, but meant for different purposes. DSniff is powerful in terms of capturing SSL traffic.

FOSS systems have no built-in method to protect themselves from sniffers. The methods described above could pretty much be used for various Linux distros, to make those less vulnerable to sniffer attacks. A smart utility available on Linux distros, called AntiSniff, can be used in a script to detect network interfaces in promiscuous mode.Network sniffing is difficult to detect because it is a passive and silent type of attack. There are methods to detect and disable sniffers, and network administrators should incorporate those into their network to protect their IT

V. REFERENCES

[1] http://www.computerhope.com
[2] http://www.compnetworking.about.com