Proceedings published by International Journal of Computer Applications® (IJCA)
International Conference on Computer Communication and Networks CSI- COMNET-2011

Sniffing: A Major Threat to Secure Socket layer and its Detection

Ajay Mathur
Dept.of Computer Sc.and Engg.Govt.Poly. College,Jodhpur,Technical Education,Rajasthan

Sudhir Kr.Sharma
Dept.of Computer Sc.and Engg,Govt.Poly.College, Jodhpur,Technical Education,Rajasthan

Amit Mishra
Dept.of Computer Sc.and Engg,JECRC,Jodhpur'JNU, Jodhpur

ABSTRACT
Network sniffing was considered as a major threat to network and web application. Every device connected to the Ethernet-network receives all the data that is passed on the segment. By default the network card processes only data that is addressed to it. However listening programs turn network card in a mode of reception of all packets – called promiscuous mode. So, a *sniffer* is a special program or piece of code that put the Network Interface Card (NIC) in the promiscuous mode. When NIC works in promiscuous mode, the user of that system can steal all the data including password etc. without generating any traffic. Any network system running the sniffer can see all the data movement over the network. Many sniffers like wireshark, Cain & Abel, ethersniff etc. are available at no cost on the internet. There are many proposed solutions are available for the detection of network sniffing including antisniff [1], SnifferWall [2], Sniffer Detector [3] etc. but any solution does not guarantee full security. Due to this reason many new techniques were developed including secure socket layer (https), one time password etc. but now there are some techniques that can be used to sniff this secure data. In this paper we are discussing different aspects of sniffing, methods to sniff data over secure socket network and detection of sniffer. The paper describes all the technical details and methods to perform this task. The Address Resolution Protocol packets are used to query hardware addresses from IP addresses. We are using this fact to verify to whether the NIC's are set to promiscuous mode. When NIC receiving all packets, it will not block any packet and forwards it to the kernel for further processing. Now according to the working of the ARP, the kernel may make mistake by responding to some packets that it is not supposed to respond. So according to this mechanism we can compose fake ARP request packets and send them to every node on the network. If any node responds to this fake request, we can detect it is running in promiscuous mode.

General Terms
My general term which can be used for general classification of the submitted material is sniffing.

Keywords
Keywords are your own designated keywords which can be used for easy location of the manuscript using any search engines.

1. INTRODUCTION
Computer networks are the backbone of an organization. In most of the cases, any organization that is using network depends on the Ethernet technology. In a hub based Ethernet network, when the source wants to send a data packet to destination it broadcasts the message on to the network. Then this packet moves to all the computers connected in the network. Each machine is supposed to ignore the packet if it is not destined for the Internet Protocol (IP) address assigned to that computer/machine. The network interface card (NIC) performs this filtering operation. The packet sniffer is a program that puts the NIC in a special mode called promiscuous mode. In this mode, the NIC does not perform the filtering operation and passes all the received data to the operating system for further processing [3]. The sniffer in the network can be shown in Fig.1.

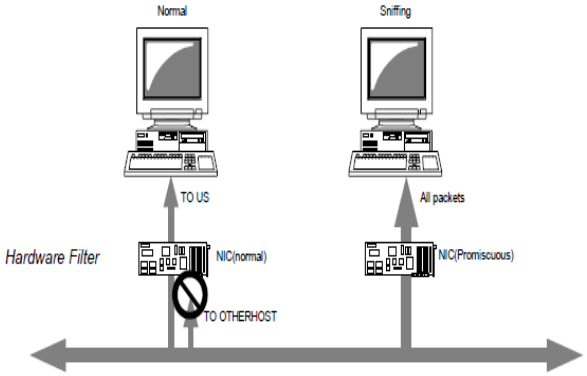

Fig.1 NIC working in Promiscuous Mode

There are many popular sniffers, which are available for free on the internet, as listed below:

- Wireshark
- Kismet
- Tcpdump
- Cain and Abel
- Ettercap
- EtherApe

For sniffing data over secure socket layer, we are considering Ettercap. It is a free sniffer tool for UNIX environment but now it is also available for windows based systems.

2. SECURE SOCKET LAYER AND SNIFFING

In this section, the method of sniffing over secure socket layer is discussed. Before going into the details of sniffing, working of Secure Socket Layer (SSL) should be discussed. Netscape designed the secure socket layer protocol for web security purpose in 1993.

SSL is a separate protocol layer just for security. It was inserted between HTTP and TCP layer of standard protocol. It can be shown in Fig.2 as:

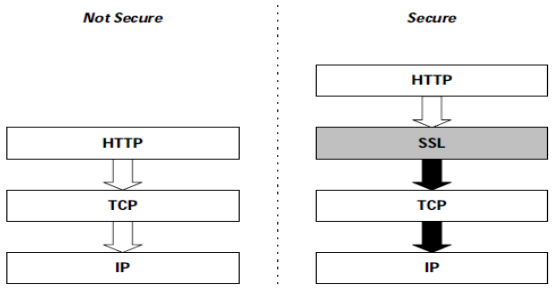

Fig.2 SSL Layer between HTTP and TCP

The SSL protocol consists of a set of messages and rule about when to send (and not to send) each one.The SSL defines two different roles for the communicating parties. One system is always a client, while the other is a server. The client is the system that initiates the secure communication; the server responds to the client's request. SSL works through a combination of programs and encryption/decryption routines that exist on the web server computer and in web browsers (like Netscape/Firefox and Internet Explorer) used by the Internet public. The process can be shown in Fig.3:

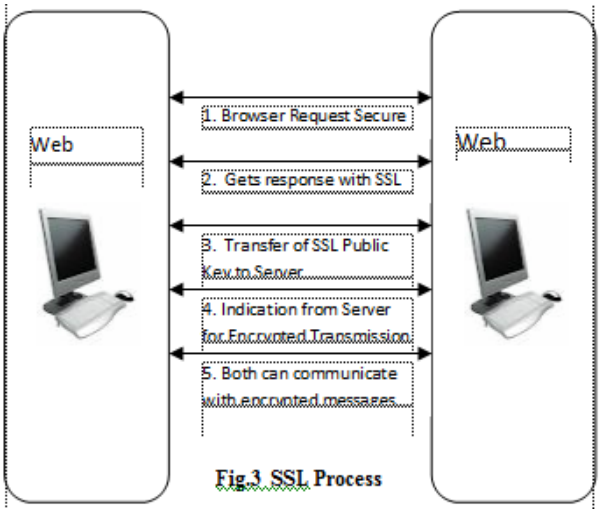

Fig.3 SSL Process

The SSL certificate is installed on a system to encrypt sensitive data such as credit card information. SSL Certificates give a website the ability to communicate securely with its web customers. Without a certificate, any information sent from a user's computer to a website can be intercepted and viewed by hackers and fraudsters. It is similarto the difference between sending a post card and a tamper proof sealed envelope [7].As discussed earlier, the server installed a certificate in client's system. The Ettercap can be used to sniff data over the secure socket layer. Ettercap is a tool made by Alberto Ornaghi (ALoR) and Marco Valleri (NaGA) and is basically a suite for man in the middle attacks on a LAN. For those who do not like the Command Like Interface (CLI), it is provided with an easy graphical interface.Ettercap is able to perform attacks against the ARP protocol by positioning itself as "man in the middle" and, once positioned as this, it is able to:

- Infect, replace, delete data in a connection
- Discover passwords for protocols such as FTP, HTTP, POP, SSH1, etc ...

Provide fake SSL certificates in HTTPS sections to the victims.

Man in the Middle Attack:-

This is an attack where a pirate put its machine in the logical way between two machines speaking together as shown in the Fig.4below.

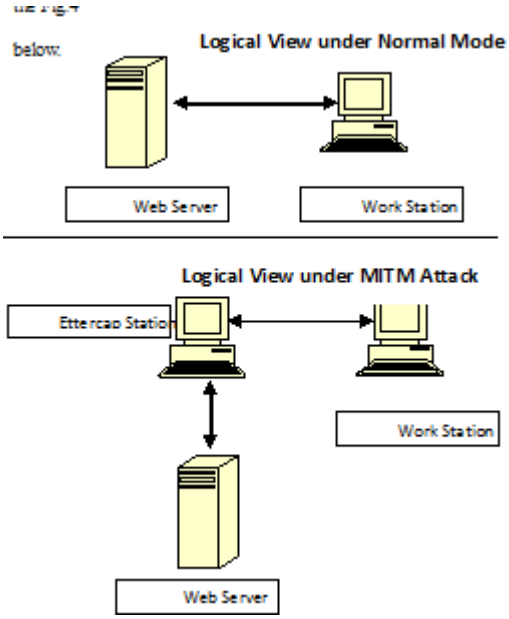

Fig.4 Normal Operation & MITM Attack

Once in this position, the pirate can launch a lot of different very dangerous attacks because he/she is in the way between to two normal machines.We'll only be able to sniff a network on the same subnet as us. The subnet is usually 255.255.255.0 so click on Options >> Set Netmask and enter the subnet of your network. Now let's start sniffing. Click Sniff >> Unified Sniffing and enter the network interface you want to use. Now we need to scout for hosts on the network. Click on Hosts >> Scan for hosts and wait for it to finish. Then click Hosts >> Host List. This will display a list of hosts. Now you need to define targets for the MITM attack. The router should be added to Target 1 and any other hosts you want to ARP poison should be added to Target 2. This is done by clicking on the host then clicking on either Target 1 or Target 2. Once you've defined your hosts, we need to ARP poison.

3. SNIFFING DETECTION

The following methods can be used to detect the sniffer present on the network.

3.1 Ping Method

In a TCP/IP (IP Version 4) network, every computer has a 32-bit IP address that is used to identify the computer uniquely. Ethernet devices have a 48-bit hardware address, and some kind of mapping between IP and Ethernet is needed when two computers needs to talk to each other. This mapping is called ARP and is short for Address Resolution Protocol. Using these facts we transmit an "ICMP Echo Request" (ping) with correct IP address and a fake MAC address. Under the normal operation, No one should reply this Request because the MAC address does not match with any computer. But if any computer/NIC working in promiscuous mode will collect this request and reply this request. In this way we can detect that any system is performing sniffing or not. The process can be shown using the following Figure. But unfortunately operating system may use virtual MAC address. In this case this technique will not work [4].

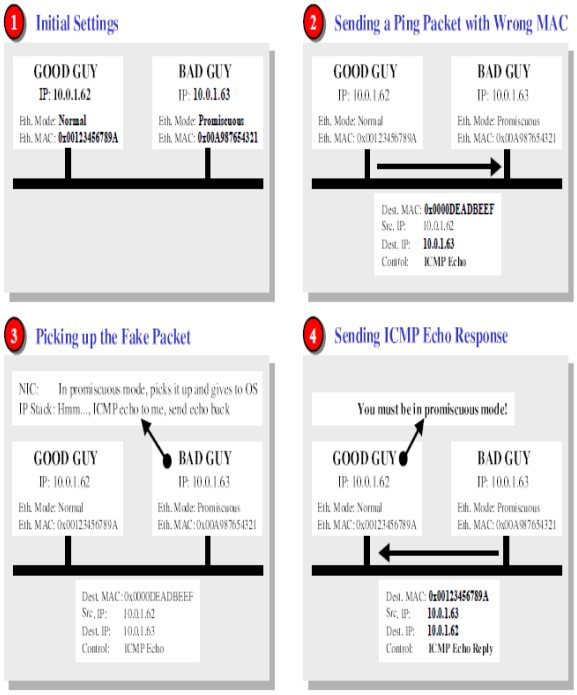

Fig.5 Sniffing Detection using Ping Method

3.2 ARP Method

Network sniffer does not send any packet to the network, so it is hard to detect sniffer. But the behavior of NIC is different from the normal mode. It forwards all the received packets to the operating system or kernel. So in this case hardware filter does not work. We can easily understand the working of this method using a real life example: Imagine a classroom with students and teacher. One student named "Mr. X" came late to class and now he is sniffing the lecture going on in the class room. He listens all the conversations going on in the class room. At the time of attendance if name of sniffer "Mr. X" is called and the "Mr. X" makes a mistake by responding "Present Sir". So NIC in promiscuous mode receives all the packets including those that are not targeting to it, it may reply to a packet which should be filtered by NIC [5] [6]. Now using this technique we can detect a sniffer present on the network. A computer system may set hardware filter in the following mode:

- Unicast
- Broadcast
- Multicast

In ARP, when a nodes wants to know the hardware address of node X, it compose an ARP request packet having (FF-FF-FF-FF-FF-FF) in destination hardware address field [8]. It shows that it is a broadcast message. So all the nodes in the network will receive this packet and only targeted node will reply in normal mode. So we can use following steps for Promiscuous Detection:-

1) We compose an ARP packet with the following format:

Ethernet address of destination	FF FF FF FF FF FF
Ethernet address of sender	00 11 22 33 44 55
Protocol type (ARP = 0806)	08 06
Hardware address space (Ethernet = 01)	00 01
Protocol address space (IPv4 = 0800)	08 00
Byte length of hardware address	06
Byte length of protocol address	04
Opcode (ARP request = 01, ARP reply = 02)	00 01
Hardware address of sender of this packet	<Own NIC's Device Address>
Protocol address of sender of this packet	<Own PC's IP Address>
Hardware address of target of this packet	00 00 00 00 00 00
Protocol address of target	<IP Address (A)>

2) After composing this packet, we sent it onto the network.

3) Now, this packet is supposed to be blocked by the hardware filter (in the NIC) of the target machine. But if the NIC is working in promiscuous mode, this packet will pass the hardware filter and reaches to kernel for processing. If we receive a response from that machine, the machine is in promiscuous mode.

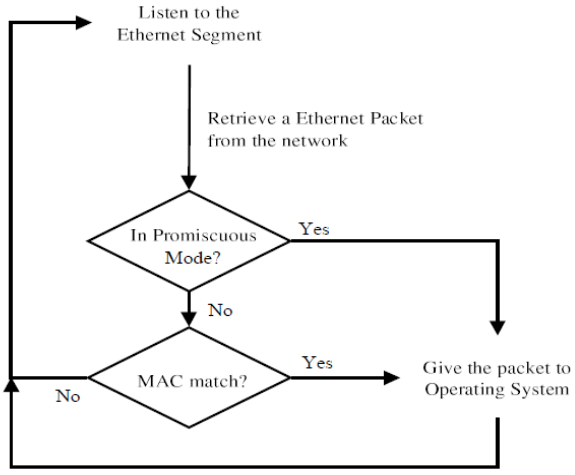

Fig.6 Flow diagram of Packet to the Operating System

For sniffer detection we set destination or Target hardware address different from the broadcast address. Suppose we set it to 00-00-00-00-00-02. Now in normal mode every node will discard this packet due to hardware filter. In promiscuous mode, the system kernel assumes that it is an ARP request for system so it responds back to the requesting node. In this way we can detect a node for sniffing [2].

3.3 Decoy Method

As we know many protocols allow plain text passwords and these passwords may be hacked by hacker, who is running the sniffer. The decoy method uses this activity for detecting the sniffer. We set a client and a server using POP, Telnet or any other plain text protocol. We configure some special accounts or virtual accounts on this server. When hacker gets username and password of this account then he tries to log in using this information. We can use standard intrusion detection system to track or log this activity. We can also identify the hacker's system when he tries to log in using that fake username and password. So the decoy method basically works on the principle of Honeypots in which we attract the hacker or intruder, so that we can identify them when they perform any action.

4. CONCLUSION

In this way it can be concluded that network sniffing is a major threat for computer security because sniffer is a passive component and it does not send any packet to the network. So it is difficult to detect the sniffer. The one solution to this problem is secure socket layer. But data can be hacked over SSL networks using sniffing tools like Ettrrcap etc. Similarly sniffer detection methods can be used to detect the sniffers present on the network. We have developed and Implemented ARP Sniffer Detector that runs on Linux System. All the methods described here may not work with 100% efficiency because the whole paradigm is changing very frequently and the hackers and intruders are discovering new methods for the intrusion. In the similar way new methods should be discovered for security.

5. REFERENCES

[1] Antisniffing: http://www.securitysoftwaretech.com/antisniffing, (2004).

[2] H. M. Kortebi AbdelallahElhadj, H. M. Khelalfa, An experimental sniffer detector: Snifferwall, (2002).

[3] Thawatchai Chomsiri, Sniffng packets on lan without arp spooffing, Third 2008 International Conference on Convergence and Hybrid Information Technology(2008).

[4] D. Wu and F. Wong, Remote sni_er detection, Computer Science Division, University of California, Berkeley (1998).

[5] Daiji Sanai, Detection of promiscuous node using arp packets, www.securityfriday.com (2001). 50-51

[6] Detection and Prevention of Active Sniffing on Routing Protocol, Pathmenanthan ramakrishna' and mohd aizaini maarof, Student Conference on Research and Development Proceedings, Shah Alam, Malaysia (2002).

[7] www.evsslcertificate.com/ssl/description-ssl.html

[8] http://www.tcpdump.org.

[9] http://reptile.rug.ac.be/˜coder/sniffit/sniffit.html

[10] www.scribd.com/doc/29844162/Ettercap-Tutorial

[11] S. Grundschober, Sni_er detector report, IBM Research Division, Zurich Research Laboratory, Global Security Analysis Lab (1998).

[12] B. Issac S. Kamal, Analysis of network communication attacks, The 5th Student Conference on Research and Development (2007).