ANTI PACKET SNIFFING: A ZERO HACKING

Rupal Sinha, D.K. Mishra
Acropolis Institute of Technology and Research, Indore
rupulsinhakk@gmail.com, mishra_research@rediffmail.com

ABSTRACT
As data streams travel back and forth over the network, the sniffer captures each packet and eventually decodes and analyzes its content. So, it’s a cruel irony in information security that many of the features that make using computers easier or more efficient and the tools used to protect and secure the network can also be used to exploit and compromise the same computers and networks. This is the case with packet sniffing. Using the information captured by the packet sniffer an administrator can identify erroneous packets and use the data to pinpoint bottlenecks and help maintain efficient network data transmission. Typically, the packet sniffer would only capture packets that were intended for the machine in question. However, if placed into promiscuous mode, the packet sniffer is also capable of capturing all packets traversing the network regardless of destination.

Keywords: Packet Sniffer, Secure Socket Layer, Virtual Private Network, Hacking.

INTRODUCTION
A packet sniffer can only capture packet information within a given subnet. So, it’s not possible for a malicious attacker to place a packet sniffer on their home ISP network and capture network traffic from inside your corporate network. In order to do so, the packet sniffer needs to be running on a computer that is inside the corporate network as well. However, if one machine on the internal network becomes compromised through a Trojan or other security breach, the intruder could run a packet sniffer from that machine and use the captured username and password information in order to compromise other machines on the network. Packet sniffers are very hard to detect because they are passive, which means the sit on background reading the network stream and never send out data. So unless we know it’s there we can’t find it. One of the major drawbacks of packet sniffers is if they can’t understand data its junk. So the best way is to make data encrypted. But there are some more methods available like: Secure Socket Layer, IP Security, PGP and MIME, Virtual Private Network. These are latter discus in this paper.

Detecting rogue packet sniffers on the network is not an easy task. It simply captures the packets that are traveling to the network interface it is monitoring. That means there is generally no signature or erroneous traffic to look for that would identify a machine running a packet sniffer. There are ways to identify network interfaces on your network that are running in promiscuous mode though and this might be used as a means for locating rogue packet sniffers.

For example, the internet Small Computer System Interface (iSCSI) a means of carrying out the transfer of data over intranets, or remotely managing data storage - incorporates IPSec security, which can encrypt data as it is transferred between two devices, preventing a hacker with a sniffer from seeing the contents of that data. But this discounts the fact that a hacker would not only have to get access to the data being routed from point to point, but would have to know ahead of time which packets to capture and decrypt from the thousands of packets per second traveling over a particular network segment. For a long time, this hacking route was perceived as so risky that IPSec was almost mandatory for iSCSI traffic. However this requirement was removed just prior to the standard's ratification, when the extreme cost to implement any reasonable data rate was fully realized. Rather than trying to decode thousands of network packets from many different sources, it is a much easier course for a hacker to get to the data where it is resting in a server. Hacking a standard server is much simpler to do locating the data and uploading it to a secondary location is much simpler than trying to decode network traffic packet by packet.

A standard packet will travel from "your comp" through the network. Each computer on the network will receive that packet. Starting with "friend comp", followed by "bad guy" and ending up at "dest comp." this name because it sounds more legitimate and less threatening. The original term for capturing all packets on a network was called 'Sniffing the Ether' which sounded like something bad to people not familiar to computers and Ethernet. 'Ether' was a technology term used to describe the land of packets, made up of cables and network cards and should not be confused with the chemical ethyl oxide.

In this paper, we more discuss about packet sniffers and its prevention method uses encryption techniques, to reduce the core level of hacking. Also we prepared some new logical
techniques which were supposed to help to reduce this type of passive attacks[1].

RELATED WORK
As Technical Communication Corporation’s (TCC) cryptology expertise prepare protocol on encryption of packets based on reducing the sniffing of packets on network by encrypting the IP packets.[6]

What Type of an Attack is it?
A sniffer being used on a network to snoop passwords and anything else is considered to be a passive attack. A passive attack is one that doesn’t directly intrude onto a foreign network or computer. Using a sniffer as an example, one is set up in hopes of catching desired information including logins and passwords. On the other hand, an active attack directly interfaces with a remote machine. Remote buffer overflows, network floods and other similar attacks fall under the category of an active attack. By nature, passive attacks are not meant to be discovered by the person(s) being attacked. At no point should they have indication of your activity. This makes sniffers just as serious as any active attack.

How Does a Sniffer Work?
During normal tasks such as Web surfing and messaging, computers are constantly communicating with other machines. Obviously, a user should be able to see all the traffic traveling to or from their machine. Most PCs, however, are on a Local Area Network (LAN), meaning they share a connection with several other computers. If the network is not switched (a switch is a device that filters and forwards packets between segments of the LAN), the traffic destined for any machine on a segment is broadcast to every machine on that segment. This means that a computer actually sees the data traveling to and from each of its neighbors, but ignores it, unless otherwise instructed.

We can now begin to understand the magic behind a sniffer. The sniffer program tells a computer, specifically its Network Interface Card (NIC), to stop ignoring all the traffic headed to other computers and pay attention to them. It does this by placing the NIC in a state known as promiscuous mode. Once a NIC is promiscuous, a status that requires administrative or root privileges, a machine can see all the data transmitted on its segment. The program then begins a constant read of all information entering the PC via the network card. The data traveling along the network comes as frames, or packets, bursts of bits formatted to specific protocols. Because of this strict formatting, a sniffer can peel away the layers of encapsulation and decode the relevant information stored within: source computer, destination computer, targeted port number, payload, in short - every piece of information exchanged between two computers.

Here picture shows how packets sniffer tools look likes: HTTP request by the client and the server's response. Note that the first few lines of each sniffed packet provide a summary of the transaction: timestamps, source and destination MAC addresses, source and destination IP addresses and several other bits of information. The numbered lines (0050) show the data transmitted by each packet in hexadecimal format. Additionally, an ASCII decode of the payload is located off to the left - a convenient feature for crackers and nosy neighbors watching you on the network[2].

Why Should Peoples Be Concerned?
On a normal LAN there are thousands of packets exchanged by multiple machines every minute, ample supply for any attacker. Anything transmitted in plaintext over the network will be vulnerable - passwords, web pages, database queries and messaging to name a few. A sniffer can easily be customized to capture specific traffic like telnet sessions or e-mail. Once traffic has been captured, crackers can quickly extract the information they need - logins, passwords and the text of messages. And the users will likely never know they were compromised - sniffers cause no damage or disturbance to a network environment.

Different Types of Sniffers
Most of the more popular sniffers only monitor one connection at a time. The reason for this is to make the sniffer harder to detect due to smaller logs and less use of CPU power. A small number of sniffers monitor all connections. Often times, looking at the CPU load and file system are the only ways to detect such sniffers. Intruders are often quick to backdoor systems so that normal utilities like ps and ipconfig will not provide reliable output. If you notice your CPU load is higher than normal, or that every day you lose one more Meg of disk space that can't be explained, it may point to the presence of a sniffer. This type is easier to spot because their logs will be
much larger, they will eat up much more CPU, but in return it will log much more. On large networks, these sniffers may generate up to ten megabyte logs a day if set to log all interactive traffic. Sniffers designed to monitor interactive traffic as well as mail may grow even faster. Sniffers also have different methods of logging. Some sniffers will only record the first X (X being a certain number) bytes of a packet to capture a login/password. The other method will capture the entire session, which would make it into a key logger. Some of the more versatile sniffers will support both methods. These will vary depending on the intruder and the desired end result.

**Popular Sniffers**

There are sniffers that are considered to be primarily 'hacker' tools while the rest of them are considered to be system administration tools. If you are looking for a sniffer to put into a production environment then you are going to want to find a sniffer that is actively in production and is rather mature in its evolution such as tcpdump, ethereal, and snort. The following design because it is another text all together however there are list sniffers that are open source and for free. This is not a complete list but still is a comprehensive one.

- Aldebaran
- Anger
- Angst
- Etherape
- Ettercap
- Maxty
- Netl
- Ngrep
- Parasite
- Pylibpcap
- Smit
- Sniffit
- Snort
- Sosd
- Tcpcap
- Altiwire
- ADMsniif
- DSmiff
- Ethereal
- Ksniffer
- Nettude
- Niop
- Passfing
- Siphon
- Sniffer
- Snmpsniff
- SSLdump
- Tcpcap

**Our Proposal to how Users can Protect Themselves?**

The prevention of unauthorized sniffing, you should use strong cryptography, so even if someone does sniff you, you are not at much risk from this form of attack. When you originally designed your LAN, you should have security in mind anyway. We are not going to go into secure LAN and segmentation design because it is another text all together however there are a couple of methods to help you out.

**Anti-Sniffing Tools**

A scary aspect of these tools is who can, and will, use them. As stated earlier, sniffers can be used for both legitimate and illegitimate purposes. For instance, a network manager can use them to monitor the flow of traffic on the network to ensure that the network is operating efficiently. However, sniffers can also be used by malicious users to obtain valuable personal information. Whether it is passwords or private communication, both crackers and co-workers can benefit from reading your data. Defending against sniffers, as with any other threat, needs to start from the top and filter down to the user. As on any network, administrators need to secure individual machines and servers. A sniffer is one of the first things a cracker will load to see what is taking place on and around their newly compromised machine. Another method of protection involves tools, such as antisniff, that scan networks to determine if any NICs are running in promiscuous mode. These detection tools should run regularly, since they act as an alarm of sorts, triggered by evidence of a sniffer.

**Switched Networks**

A switched network is also a good deterrent. In the non-switched environment, packets are visible to every node on the network, in a switched environment; packets are only delivered to the target address. While more expensive than hubs, the cost of switches has fallen over time, bringing them within reach of most budgets. Unlike hubs, switches only send frames to the designated recipient; therefore a NIC in promiscuous mode on a switched network will not capture every piece of local traffic. But programs such as dsniff, allow an attacker to monitor a switched network with a technique known as arp-spoofing. Although it uses different methods, arp-spoofing can provide results similar to sniffing, i.e. compromised data. Is there anything that can truly protect your data once it reaches the network?

**Virtual Private Networks**

A VPN transparently tunnels normal LAN activities over a wider network and usually is used to support the distribution of a single organization over the Internet. Commonly supported between two firewalls, a VPN is a form of point-to-point encryption. Increasingly, this same technology also is being used to support remote users who access their organization's LAN through the Internet. Usually applied at the perimeter of a network (i.e., the Internet Gateway), a VPN is a network extension tool. It temporarily extends the boundary of a private network either to a single remote user or to another network. As implemented by most firewall vendors, a VPN session is automatically initiated when either network entity attempts to access the other. Firewall vendors usually offer a choice of authentication mechanisms for use by individual remote users (either traditional reusable passwords or one-time passwords generated by a hardware token device). Because it is configured in the transport (TCP) layer, all traffic between two entities flows through a VPN automatically without either the awareness or choice of the user or the application[4].

**Encryption**

Encryption is the best protection against any form of traffic interception. It is reasonable to assume that at some point along a path, data can always be compromised. Therefore, your best defense is to ensure that traffic is essentially unreadable to everyone but the intended receiver. This isn't difficult to do, since many organizations have deployed services that make use of Secure Socket Layers (SSL), Application Layer Encryption (ALE) and other methods that provide secure messaging, web browsing and more. Only the payloads are scrambled, ensuring...
that packets reach the correct destinations. So an attacker can see where traffic was headed and where it came from, but not what it carries[3].

**Secure Socket Layer**

SSL has become ubiquitous on the Internet. It is widely used to provide privacy for on-line storefronts and other sensitive applications. Developed and implemented by Netscape, SSL is a form of host-to-host encryption that extends encryption all the way from a server to a client workstation. Firewalls are customarily configured to allow both incoming and outgoing SSL sessions. As a transport layer service (more specifically, a service that sits directly above the transport layer), it still cannot provide integrity or non-repudiation services because it does not have direct access to the objects being transmitted through it. It has an advantage over VPN in that it can be invoked from applications which are modified to support it. Most Web browsers have been modified to invoke an SSL session when using URLs starting with http:' SSL is a convenient way to selectively provide confidentiality between a browser and a Web server. Because it provides the normal socket interface, it is possible for SSL to support virtually any application, as long as that application has been designed to invoke and use SSL instead of the generic TCP socket services. Few SSL applications are available and in practice it is used almost exclusively for Web support.

**Application Layer Encryption**

Only a service that can operate on discrete objects can sign them or verify them. S-HTTP is a standard set of security services that operates between Web browsers and Web servers. Careful application of the OSI Model would probably place S-HTTP at the presentation layer, but it offers the same capabilities as application layer encryption, if not the same level of flexibility, because it can directly operate on the objects being served through the Web. S-HTTP is a very useful protocol because it can provide object integrity and digital signature without requiring programmatic support, but unfortunately it is not widely implemented.

**IP Security**

In this mechanism providing cryptographic authentication for IPv4 and IPv6 datagram's. The Authentication Header is a mechanism for providing strong integrity and authentication for IP datagram's. It might also provide non-repudiation, depending on which cryptographic algorithm is used and how keying is performed. For example, use of an asymmetric digital signature algorithm, such as RSA, could provide non-repudiation. Confidentiality and protection from traffic analysis are not provided by the Authentication Header. Users desiring confidentiality should consider using the IP Encapsulating Security Protocol (ESP) either in lieu of or in conjunction with the Authentication Header. This document assumes the reader has previously read the related IP Security Architecture document which defines the overall security architecture for IP and provides important background information for this specification.

The IP Authentication Header seeks to provide security by adding authentication information to an IP datagram. This authentication information is calculated using all of the fields in the IP datagram which do not change in transit.

**MIME Security with Pretty Good Privacy (PGP)**

PGP can generate either ASCII armor or 8-bit binary output when encrypting data, generating a digital signature, or extracting public key data. The ASCII armor output is the REQUIRED method for data transfer. This allows those users who do not have the means to interpret the formats to be able extract and use the PGP information in the message. When the amount of data to be transmitted requires that it be sent in many parts, the MIME (Multipurpose Internet Mail Extensions) message/partial mechanism should be used rather than the multipart ASCII armor PGP format.[5]

Before encryption with PGP, the data should be written in MIME canonical format (body and headers). PGP encrypted data is denoted by the "multipart/encrypted" content type, and MUST have a "protocol" parameter value of "application/pgp-encrypted". Note that the value of the parameter MUST be enclosed in quotes. The multipart/encrypted MUST consist of exactly two parts. The first MIME body part must have a content type of "application/pgp-encrypted". This body contains the control information. A message complying with this standard MUST contain a "Version: 1" field in this body. Since the PGP packet format contains all other information necessary for decrypting, no other information is required here. The second MIME body part MUST contain the actual encrypted data. It must be labeled with a content type of "application/octet-stream".

**Example message:**

From: Enrique petar petar@cents.com
To: Enrique petar petar@cents.com
Mime-Version: 1.0
Content-Type: multipart/encrypted;boundary=foo;protocol="application/pgp-encrypted"
--foo
Content-Type: application/pgp-encrypted
Version: 1
--foo
Content-Type: application/octet-stream
CONCLUSION

Having looked at what they are, why they work and how they are used, it is easy to view sniffers as both dangerous threats and powerful tools. Every user should understand they are vulnerable to these types of attacks and their best defense lies in encryption. Administrators and professionals need to know that these programs are superb diagnostic utilities that can, unfortunately, be used with malicious intent on any network.

As we have studied and proposed an optimal solution for packet sniffing and encryption techniques we recommend that the protocol should be use in these fashions.

REFERENCES

[1] Rueda, Jose A. (Manitoba, CA, US), Bourlas, Yair (Manitoba, CA, US), Diamond, Jeffrey E. (Manitoba, CA, US), McLeod, Robert D. (Manitoba, CA, US) has posted article on "Method for real-time traffic analysis on packet networks"
http://www.freepatentsonline.com/6597660.html

[2] Firat kocak posted paper on 23 may 2003 on "Packet Capture and Analyazer"
http://www.codeproject.com/csharp/pacanal.asp

http://www.freepatentsonline.com/7006628.html


Copy Right © INDIACom – 2008