Global Information Assurance Certification Paper

Packet Sniffing: An Integral Part of Network Defense
Daniel Magers
May 09, 2002

Overview

The need for network security has rapidly become a vital consideration for even the most novice of computer users. What was once an ambiguous term, Network Security has evolved into its own enterprise devoted exclusively to the development of system security software.

By definition, security provides safety, and in the context of computer network security it is there to prevent undesirable intrusion to your system. Unfortunately, the "all in one" program is non-existent making it necessary to employ a "layered" network security package to insure optimum protection. The purpose of this paper is to provide an informative synopsis of the packet sniffer product and the advantages of including one as part of a layered network defense.

A packet analyzer is an excellent Information Technology (IT) management tool that supplies a comprehensive view of your network health and performance. A properly deployed packet sniffing product assists in the early detection of unusual traffic patterns, which could be a sign of malicious activity. The seemingly never ending battle to impede hacking makes it imperative to have the ability to monitor your system for security breaches.

Background

A successful security model will include, but is not limited to:

- A written security policy endorsed by management
- A security toolkit that assists IS personnel in the performance of their duties
- Regularly scheduled and impromptu audits
- Router-based security
- Standard Operating Policy for incident response and handling
- Firewalls/Intrusion detection in real or near-real time

An Intrusion Detection System (IDS) is a viable choice when establishing a line of defense. An IDS can be application based, host based, network based, hybrid, or personal. It utilizes a number of established parameters to prevent external access: intrusion signatures for specific applications, suspicious activity, signature based activity, packet filtering (based on different values of the IP packet), and/or port number.

The proxy and application level firewalls protect against access to critical servers and segments. They initiate and then log pop up alarm windows when the system identifies a potential intrusion.

Network based intrusion detection systems were traditionally promiscuous packet sniffers with IDS filters that decoded protocols and maintained state. A packet sniffer IDS is the additional measure that broadens the perimeter and enhances the security capabilities.

Packet Sniffers as part of your Network Monitoring Solution

Packet sniffers are software applications that use a network adapter card in promiscuous mode to capture all network packets that are sent across a local area network. When placed in promiscuous mode, a network adapter card sends all packets it receives on the physical wire to an application for processing. This process enhances the monitoring capabilities of IS personnel by enabling them to view networks at a high level. They can determine what applications are running, the users who are logged on, and the source generating the high traffic. Once the baseline has been established, a type of first warning system has been created; any unusual or unidentified traffic patterns will alert of possible intrusion.

The diagnostic scope of the original packet analyzers was limited to message headers of data packets on the network; IS administrators were able to view low level information about the packets such as address detail of sent and received messages, file sizes, and transmissions. Graphs and text-based descriptions were used to evaluate and diagnose performance problems with servers, the network wire, hubs, and applications. These solutions required administrators to go off-line to analyze the data they recorded from their networks. The slow response and time-consuming effort could be quite detrimental to a network's performance. Any suspicious activity or its impact would not be known until well after it occurred.

The new and improved applications are notably more efficient allowing for real-time analysis. A system now captures packets off the network, decodes them into human readable format, runs the packet through a well-defined analysis system and then displays the information. This technology alerts and substantially increases the response time to issues before they become a significant problem.

The vendor-configured alert, or tripwire(s) is a default set by the network administrator to meet the particular needs of the organization. Examples of tripwire values are: too much bandwidth being utilized by a specific system; slow HTTP, POP3, or FTP response time; too many TCP retransmissions; or an IP header checksum error.

There are two broad varieties of packet sniffers available. The first is a stand-alone product incorporated into a portable computer that plugs into the network to gather diagnostic data. The second is part of a larger package used for local area networks (LAN), wide area networks (WAN), and Web services. The latter variety provides a centralized view of a network.

Sniffers are also capable of providing graphical representations and statistics. The peer map defines which systems are in communications with one another and the volume of traffic they are passing. The data supplies a quick and high-level account of traffic

activity. Detailed statistics such as, the exact percentage of network traffic attributed to a specific protocol (RIP, HTTP, NetBIOS…) are also supplied. The analyst is able to retrieve data from the entire network, hone in on a particular node or protocol, and receive the historical statistics to compare past and present performance.

How they work

Ethernet was conceived with the premise that all machines on a local network will share the same media (collision domain).

All workstations utilizing the same segment will see all traffic passing through that segment. The Media Access Control (MAC) address burned onto your network interface card (NIC) determines if the traffic is intended for your workstation. If the message is addressed to your machine, it will accept and processes it. The other workstations will receive the broadcasted message, but ignore the traffic. If a NIC on any other machine is set to promiscuous mode, that workstation will collect and store the traffic on any available storage media.

A sniffer program is designed to perform this very task. It places the NIC in promiscuous mode, thus becoming your personal network spy.

The main components of a packet sniffer are:

1. Hardware – usually standard network adapters, though some require special hardware
2. Capture driver – the most important part, because it captures the network traffic from the wire, filters it for the particular traffic you want, then stores the data in a buffer.
3. Buffer – stores the captured data in either a set buffer (when the buffer fills, data capture stops) or round robin (where new data overwrites the oldest data).
4. Real-time analysis – this feature does minor analysis of the captured frames as they come off the wire.
5. Decode – displays the contents of network traffic with descriptive text so an analyst can observe traffic performance.
6. Packet editing/transmission – a feature that allows you to edit then transmit your own packets onto the network.

Places to deploy your network sniffer

In early days of flat networks, traffic monitoring was basically a simple process. All traffic in one part of the network was received in all other parts of the network. A traffic monitoring program could be placed anywhere in the network and capture packets for analysis.

In today's environment of switched and routed hierarchical networks, the deployment of traffic analysis tools requires careful planning. In a switched environment, multiple

packet sniffers would be required unless there is a common location in the network where all packets are routed. Deploying an individual monitor for each point-to-point link would not be cost effective. Also, it could severely degrade network performance in as much as; all of the captured data is forwarded to a central monitoring location.

In a switched environment you can use a switch port analyzer. The port configures the switch to copy transmit, receive or both from one port or virtual local area network (VLAN) to another port (the span port). The packet sniffer can be placed on the switch without modifying the core infrastructure of the switch. It also allows you to manage IDS without additional hardware. Unfortunately, you can only have one span port per switch.

Another option is to use a hub configuration. This is usually done between two switches, a router and a switch, or a server and a switch. The hub allows traffic to flow between the two devices while directing a copy of the traffic to the packet sniffer.

A final solution is to use a tap. Similar to a hub, a tap is hardwired into the device. It doesn't have an impact or impede the flow of traffic; it is protected against attacks due to its hardwiring, and allows the packet sniffer to monitor errors such as undersize and oversize packets, and bad CRCs.

Available Network Monitoring programs

The first step in choosing a program is to take into account your network configuration, IT/IS staff capabilities, and vendor support (if using a commercial product). Once these variables are considered, you can set the criteria for the requirements to look for in a program

The following is a partial list of existing programs. I've included a brief edited description of the product.

ANTISNIFF – Developed by L0PHT, it is a proactive security-monitoring tool. It has the ability to scan a network and detect whether or not any computers are in promiscuous mode. It was designed to identify compromised machines with IP stacks that a remote attacker could utilize to sniff network traffic.
http://www.l0pht.com/

ETHEREAL – UNIX-based program that also runs on Windows. It comes in both a read only (protocol analyzer) version as well as a capture (sniffing) version. The read only version is great for decoding existing packet captures (such as the traces that BlackICE generates). It avoids the hassle of installing the packet capture driver.
http://ethereal.zing.org/

INTERNET SECURITY SYSTEM INC.'s REALSECURE – A hybrid approach that combines network-based and host-based intrusion detection into a single platform. Uses

a standards-based approach, comparing network traffic and host log entries to the known and likely methods of attackers. Provides alarms and other configurable responses.
http://www.iss.net/

NAI Sniffer – A key part of an overall network management strategy to eliminate network downtime, simplify network management, facilitate network planning, and reduce the cost of operating and managing you critical networking resources. Supports all major network topologies, speeds, and connection methods.
http://www.sniffer.com/products/dssrmon-analysis/deafult.asp?A=1

PORTSENTRY – Part of the Abacus project suite of security tools. It is a program designed to detect and respond to port scans against a target host in real-time. It runs on TCP and UDP sockets and works on most UNIX systems. Advanced stealth detection modes are available, as well as detect SYN, FIN, NULL, XMAS, and Oddball packet scans. All modes support real-time blocking and reporting of violations.
http://www.psionic.com/abacus/portsentry/

SNIFF'EM – This product monitors both incoming and outgoing network traffic. It allows IS personnel to capture and retrace the steps of any network user (critical during the collection of forensics evidence of an intrusion).
http://www.sniff-em.com/

SNORT – A libpcap-based, cross-platform lightweight tool that can be deployed to monitor small TCP/IP networks and detect a wide variety of suspicious network traffic and attacks. It provides enough data to make informed decisions on the proper course of action when suspicious traffic is detected. Its components include: a decoder engine, a detection engine, and a logging and alert subsystem. The engines work on user defined rules that are easy to write and implement.
http://www.snort.org/

SURFCONTROL SUPERSCOUT – a CISCO solution for network monitoring. It can monitor an IP protocol as well as, all or selective sources. It will enforce network access policy using an administrator defined rules engine. As a pass-by sniffer, it inspects data flow, but does not interrupt it.
http://www.cisco.com/warp/public/cc/so/neso/sqso/csap/surf_rg.htm

TCPDUMP – A powerful tool that allows us to sniff network packets and make some statistical analysis out of those dumps.
http://www.tcpdump.org/

For every action there is a reaction

Unfortunately, we are faced with the sometimes poignant reality that for every defensive measure we install, there is an element devoted to rendering it useless. Any safeguard available in the public domain is ultimately temporary. The programmers developing the

safeguards are in a constant research mode of counter-counter measures. The best line of defense begins with knowing how we become vulnerable.

Ways the Hacker deploys packet sniffing

One of the easiest exploits is the "router redirection". ARP queries contain the designated IP-to-MAC mapping for the sender. In order to reduce ARP traffic, most machines will cache the information received from the query broadcasts. A malicious attacker can redirect nearby machines to forward traffic through it by sending out regular ARP packets containing the router's IP address mapped to its own MAC address. All the machines on the local wire will believe the hacker is the router, and therefore will pass the traffic through him/her.

A similar attack would be to Denial of Service (DoS) a target victim and force it off the network, then begin using its IP address. If a hacker does this carefully, he or she can inherit connections already established without dropping them. Windows machines are so accommodating that, when they come onto the network and see someone else using their address, they will kindly shut down their own TCP/IP stacks and allow this to continue. SMB (the Windows file sharing protocol) is also considerate enough to allow predictable identifiers, thus allowing crackers/hackers to predict enough information to keep the connection going.

So how do we provide the maximum level of protection for our network against malicious packer sniffers?

Brian Posey in his article Sniffing out Packet Sniffers suggests,

> 'Try watching for machines that are performing lots of DNS lookups. Although a high volume of DNS lookups [doesn't] necessarily indicate packet sniffing, it's [definitely a good gauge]. If you suspect that a particular machine might be packet sniffing, try setting up a bait machine. A bait machine is a PC that no one knows exists. Plug it up to the network and generate a small amount of network traffic. As you do, keep an eye on the DNS queries to see if the suspected machine ran a DNS query on the bait machine. If it did, then it's almost certainly sniffing packets.'

Robert Graham provides the following solutions on his FAQ page:

> If you are working in an organization that still employs hubs, you can replace your hub with a switch. This is a simple, yet effective defense against casual sniffing. Unfortunately, a switch still creates a ["broadcast domain"], providing an attacker the ability to spoof ARP packets.
>
> Most Ethernet adapters allow the MAC address to be manually configured. Thus a hacker can spoof MAC addresses by reassigning the address on the adapter, or by bypassing the built in stack and handcrafting frames. The hacker must maintain a

constant stream of outgoing frames in order to convince the auto-learning switch that they are the legitimate owner of the MAC address.

Many, if not most switches allow MAC addresses to be configured statically in order to prevent this sort of thing. While it may be a difficult management burden to do this for all end nodes, it may prove useful for the router, restricting the hacker to wiretapping individual end nodes instead of everyone all at once.

Some switches can be kicked out of "[“bridging”] mode into [“repeating”] mode where all frames are broadcast on all ports all the time. Overflowing the address tables with lots of false MAC addresses does this. This can be done with a simple traffic generation phase, or by sending a continual stream of random garbage through the switch.

One of the most effective measures we can employ is IPSec. IPSec is a set of extensions to the IP protocol family. IPSec is an Internet Engineering Task Force (IETF) set of protocols that enable encrypted communication between users and devices, providing data integrity, data authentication, data confidentiality and encryption on the public internetwork.

IPSec provides the following security enhancements for the Internet Protocol:

Authentication Header (AH) – provides authenticity guarantee for packets by attaching strong cryptographic checksum to packets. Authentication header covers the whole packet, from the header to the end of the packet.

Encapsulating Security Payload (ESP) – provides confidentiality guarantee for packets by encrypting packets with encryption algorithms.

IP payload compression (Ipcomp) – provides encryption service to the packets.

Internet Key Exchange (IKE) – Provides the keys to ensure data authentication, confidentiality, and integrity.

IPSec protocols are only as good as the integrity of the keys. If secret keys are compromised, IPSec protocols can no longer be considered secure.

Numerous Requests for Comments (RFC) that provide detailed descriptions of the IPSec protocols are available. They can be reviewed at the following URL: http://www.ietf.org/html.charters/ipsec-charter.html

Tools for detecting packet sniffers:

ANTISNIFF – A network card promiscuous mode detector. It sends a series of carefully crafted packets in a certain order to a target machine, sniffing the results, and performing

timing tests against the target. By measuring timing results and monitoring the target's responses, it can determine if a target is in promiscuous mode.
http://www.l0pht.com/antisniff/

CPM (Check Promiscuous Mode) – A tool from Carnegie-Mellon that check to see if promiscuous mode is enable on a UNIX machine.
ftp://coast.cs.purdue.edu/pub/tools/unix/cpm/

IFSTATUS – Checks all Network interfaces on the system, and reports any that are in debug or promiscuous mode, which may be a sign of unauthorized access to the system.
http://www.cymru.com/~robt/Tools/

NEPED – Detects network cards on the network that are in promiscuous mode by exploiting a flaw in the ARP protocol as implemented on LINUX machines.
http://www.apostols.org/projectz/neped/

SENTINEL – A portable, accurate implementation of all publicly known promiscuous detection techniques.
http://www.packetfactory.net/Projects/sentinel/

Conclusion

Network packet sniffers are an integral part of the layered defense model. When deployed with other active security tools and countermeasures, they can help deter a hacker who is looking for an easy target. They may very well be the foremost of our tools to indicate that a compromise has occurred.

The importance of network packet monitoring is sometimes overlooked because consumers falsely assume a firewall is inclusive. Conversely, the sophisticated hacker is well aware of the attributes of a packet sniffer. This fact makes it all the more imperative to include them in your security package.

The hacker's mission is to devise the means to circumvent barriers set by the firewall; it behooves us to employ the means to impede their quest. Packet sniffers and protocol analyzers enhance your ability to identify intrusion by detecting network nuances, enabling the IT/IS to initiate preventive measures. To accomplish this, we must establish solid and verifiable baselines of critical segments of our network.

No single tool is equipped to eradicate all security breaches. A comprehensive defense offers network managers an unsurpassed level of protection. A secure environment requires implementing a concise security policy supported by management to include: multiple layers of defense, continuous monitoring without establishing a pattern, and a documented company response strategy to intrusions – this should clearly outline responsibilities.

List of References

Andress, Mandy, "Get to know your Network", 29 November 2001, URL: http://www.infoworld,com/articles/tc/xml/01/12/03/011203tcpackets.xml

Biggs, Maggie, "Spotting Mischief.", 01 Oct 2001 URL: http://www.fcw.com/fcw/articles/2001/1001/tec-intrusions-10-01-01.asp

Cliff, A., "IDS Terminology Part Two: H-Z, 19 July 2001, URL: http://online.securityfocus.com/infocus/1214

Graham, Robert, "Sniffing (network wiretap, sniffer) FAQ", 2000, URL: http://www.robertgraham.com/pubs/sniffing-faq.html

Joch, Alan, "Network Sniffers", 23 July 2001, URL: http://www.nettest.com/pdf/Computerworld2_0701.pdf

Liang, Brian, "How to Guide – Implementing a Network Based Intrusion Detection System", Copyright 2000, URL: http://www.snort.org/docs/iss-placement.pdf

Posey, Brian, "Sniffing out Packet Sniffers", 20 July 2001, URL: http://www.isp-planet.com/technology/2001/sniff_packet_sniff.html

Cisco Security Associate Design Guide for surfCONTROL SuperScout http://www.cisco.com/warp/public/cc/so/neso/sqso/csap/surf_rg.htm

Dolphin WorldNet Security, "Security Issues When Connecting to the Internet, URL: http://www.dolfin.com/services/firewall/security.htm

Frequently asked questions, URL: http://www.openbsd.org/faq/

LINUX Administrator's Group, URL: http://www.uiuc.edu/ro/lag/security.html

NETBSD Documentation, URL: http://www.netbsd.org/documentation/network/ipsec