Packet Sniffing:
Network Wiretapping

Packet data:
0000: 44 45 53 54 00 00 20 53 52 43 00 00 08 00 45 00 DEST.. SRC....I.
0010: 00 C2 8D 7F 40 00 FC 11 81 93 CE 0D 1D 0C CF D6 ...(8...........
0020: B4 2B 00 35 04 10 00 AE 04 94 00 01 81 80 00 01 .+.5 ..........  
0030: 00 03 00 02 00 02 04 61 69 6D 31 0A 61 64 73 6F ...........ainl.adso 
0040: 65 74 77 61 72 65 63 63 6F 6D 00 00 01 00 01 00 01 C0 ftware.com.....
0050: 0C 00 05 00 01 00 00 0D 17 00 12 04 61 69 6D 31 ..............ainl  
0060: 07 61 75 72 65 61 74 65 63 63 6F 6D 00 00 00 01 00 01 00 .aureate.com....
0070: 01 00 01 00 00 20 39 00 04 0D 08 8C C0 31 00 ..... 9........1. 
0080: 01 00 01 00 00 20 39 00 04 D8 25 0D 8C C0 36 00 ..... 9....6. 
0090: 02 00 01 00 00 10 B1 00 05 02 4E 53 C0 36 C0 36 .............NS6.6 
00A0: 02 00 01 00 00 10 B1 00 06 03 4E 53 C2 C0 36 .............NS2.6  
00B0: C0 6F 00 01 00 01 00 00 0A BD 00 04 D8 B8 AC A4 .6...............  
00C0: C0 80 00 01 00 01 00 01 42 74 00 04 D8 25 0D A4 ............Et...%..
Introduction

Imagine you are one of the network administrators for UNC Charlotte and problems continue to arise about computer science students hacking into other students’ accounts looking for valuable information and resources. Some of the students are even hacking into professors’ accounts to find exam keys and resources for large projects. How can these intruders be caught? Can you stop them from doing more malicious acts? How do these students hack into other people’s accounts? Packet sniffers could resolve many of these problems. Packet sniffers are able to peek into packets that are being sent over a network to detect hackers. How can one set up packet sniffers and allow them to look at data being sent over the network? There are several different forms of packet sniffers, ranging in price based on the complexity of the device. There are several ways to protect data from being sniffed; however, it is difficult to secure data forever.

A discussion of packet sniffing capabilities and limitations has been included in the succeeding text, from the point of view of both securing networks and hacking into them. Additional ways to secure networks, and data; which packet sniffers are compatible with which operating systems; the most commonly sniffed protocols; and different ways to packet sniff could help a network administrator in the previously discussed situation. The following information will help individuals realize what can occur on an insecure or secure network, as well as how to prevent malicious acts.

Method and Sources

The general method used for gathering data about packet sniffing involved searching for information on the Internet. Packet sniffing methods change as networks are continually evolving. Since the Internet is often the newest source of information, it is the best source to start gathering information. To avoid biased sources, we compared several websites and used the information that was agreed upon, as opposed to using information that was contradictory. Unfortunately, there were few books containing packet sniffing information that followed the approach of this paper. As a result, the
search was limited to the Internet. With today’s rapid growth in technology, books become outdated; therefore, the Internet is the best source because it is faster to post and distribute information on the web than to print a book. A book that was examined was *Computer Networks*, by Andrew S. Tanenbaum, but the information had fewer illustrations than the Internet to support the information. One website that was used was [http://www.howstuffworks.com](http://www.howstuffworks.com), which provides a detailed description of several topics, such as firewalls and proxy servers. Another crucial site, [http://www.robertgraham.com/pubs/sniffing-faq.html](http://www.robertgraham.com/pubs/sniffing-faq.html), outlined a variety of topics related to packet sniffing. This website served as a basis from which other Internet sources were used to analyze and explain the concepts better. Some of the other useful sources used were ethereal.com, surasoft.com, and infoworld.com, all focusing on technology issues.

## Findings

### What is packet sniffing?

Packet sniffing allows one to intercept data on networks. This is done through the use of packet sniffers, which are devices that can be plugged into a network and used to eavesdrop on the network traffic (Graham, 1.1 What is a “packet sniffer?”). A packet sniffer intercepts data that is being transferred and then passes it to the destination nodes as if nothing has happened. Packet sniffers can be used for both helpful and malicious purposes. Network administrators can use packet sniffers to monitor network activity, while hackers can use packet sniffers to catch user information and passwords, such as data being sent through email or an FTP site.

### What is packet sniffing used for?

Packet sniffing can be used commercially, for network maintenance, or by hackers, for breaking into computer networks to access the data that is transferred from one network host to another.

These sniffed data packets are used for many reasons, such as retrieving clear text passwords and usernames, deciphering problems in the network, detecting bottlenecks
and intrusion on the network from hackers, checking performance on the network, and network traffic logging (Graham, 1.2 What is it used for?).

More about packet sniffers

Packet sniffers have several components, some requiring special hardware; however, most simply require the standard network adapters. The capture driver is the part of the packet sniffer that collects the data packets from the medium (Graham, 1.4.2 Capture Driver). The capture driver may also have the ability to filter the data packets to nodes that contain only the information you desire (Howstuffworks, “How Carnivore Works”). The buffer then takes each of the data packets and stores them until it fills up or it can replace the oldest packets with the newest packets (Graham, 1.4.2 Buffer). Real-time analysis does a small amount of network performance analysis on the data as it is retrieved from the network and detects problems while capturing packets (Graham, 1.4.2 Real-time analysis). Finally, decoding is a feature of packet sniffers that translates the packets of data from binary, bits of 1’s and 0’s, into a readable form of characters that makes sense to humans (Graham, 1.4.2 Decode).

Packet Sniffing Over the Ethernet

Packet sniffers can easily sniff the Ethernet, the most commonly used LAN (Local Area Network) arrangement, because it is a shared medium. This means that all traffic on the network is sent through the same medium and must pass by every host on the network. For more information on packet sniffers, please visit the website, http://www.robertgraham.com/pubs/sniffing-faq.html.

Since Ethernet involves all hosts being able to see all of the traffic transmitted on the physical medium, this is commonly considered a logical bus topology, which makes the Ethernet an easy target for packet sniffing. Each individual machine and host on the Ethernet has its own MAC (Media Access Control) address, which is a 12-digit hexadecimal number. The MAC address is a DLL (Data Link Layer) concept that enables each machine to be uniquely identified on the network. Information on the network is not
usually sent as one large piece of information, depending on the size of the data being sent, it is broken down into data packets. Each Ethernet packet contains the following information:

<table>
<thead>
<tr>
<th>Destination MAC</th>
<th>Source MAC</th>
<th>0x0800</th>
<th>IP Packet</th>
<th>CRC</th>
</tr>
</thead>
</table>

*The MAC address is needed for both the sender and the destination node*

(Graham, 1.5.3 What is the format of the MAC address?).

When an Ethernet packet is sent through the network from one machine to another, the destination and source MAC addresses are used so that the packet goes directly to the machine with the matching destination MAC address. All other machines are supposed to either drop the packet or ignore it. However, there are ways to accept and log packets as they travel to their destinations.

**Promiscuous Mode**

A packet sniffer taps into the network and places your machine into promiscuous mode. This means that you can log the packets that pass through your machine (Packet Sniffing…, Promiscuous Mode).

Ethernet adapters normally reject incoming traffic that is not directed to a certain node. Promiscuous mode allows the packet sniffer to accept all the traffic on the physical medium. However, to sniff data, a driver is required that puts the adapter into promiscuous mode and also buffers incoming frames.

**How do I protect my network from packet sniffers?**

There are many ways to protect networks from packet sniffing. The first is to encrypt all transmissions so that when the packet sniffers copy data, the only data they can copy is encrypted. There are multiple ways of encrypting data. There is concern when browsing the Internet because hackers find out what sites have been browsed and
what data has been sent and received from those sites. As a result, a good option is to make sure that SSL (Secure Sockets Layer) is enabled in the web browser. SSL is built into all major web browsers, including Internet Explorer and Netscape Navigator, and is almost always used in e-commerce because of the use of credit card information (Graham, 2.1 SSL). When browsing is done using SSL, the web address should start with https, instead of the non-secure http. Here is an example of secure browser using Internet Explorer.

Look for the "s" after "http" in the address whenever you are about to enter sensitive information, such as a credit-card number, into a form on a Web site (Tyson, How Encryption Works).

To let the user know that they are now sending and receiving encrypted data, a padlock, which is locked, appears at the bottom right of the web browser.

The padlock symbol lets you know that you are using encryption. (Tyson, How Encryption Works)

For email, the use of an encryption scheme will go a long way towards protecting data from packet sniffers. A popular Public Key encryption utility is PGP (Pretty Good Privacy). Available for free on the Internet for personal use, and for a fee for corporate
use, this utility enables a user to encrypt their data. For example, user A wants to send a message to user B using PGP. To do this, user A would need to encrypt the message using user B’s Public Key. When user B received the message they would use their Private Key to decrypt the message. This way the only person who can decrypt the message would be the intended receiver, user B, using their Private Key. For more information on how Public Key encryption and PGP works, refer to http://www.pgp.com. Just remember that no form of encryption is safe forever. If someone really wanted to read user’s email, they would eventually be able to decrypt your messages.

**Replacing hubs with switches**

Another form of protection from packet sniffers is to secure the network itself by replacing all the hubs in your local area network with switches. Packet sniffing is only this simple when the machines on an Ethernet network are connected via a hub and not a switch. A hub is a *dumb* device that has no idea which machines are connected on which ports; therefore, it broadcasts the data packets. Broadcasting means that data is sent everywhere, i.e., all connected nodes. It assumes that the destination host will receive the packet and all others will ignore the packet. This gives the packet sniffer the chance to copy the data and see what information is being sent across the network. However, a switch is a device that eventually knows and keeps track of which host is connected on which port; therefore, it is not as easily tricked. Switches create a point-to-point transfer every time a frame is sent. As a result, packet sniffers will have a difficult time intercepting the transmission, unless they are the intended receiver.

When the machines are connected using a switch, a packet sniffer must manipulate the switch so that it may gain access to the data packets traveling on the network. To trick the switch, packet sniffers commonly use the technique of flooding with ARP (Address Resolution Protocol) requests. This eventually causes the switch to act like a hub. (Packet Sniffing…, ARP). ARP will be explained in more detail further on.
Firewall Access

Installing a firewall between internal networks and the Internet is another way to protect networks. A firewall is software that filters the information coming through the Internet connection into your private network or computer system. If the filters flag an incoming packet of information, it will not allow it through the firewall (Tyson, “How Firewalls Work”).

The problem with firewall involves determining the level of access to provide to the Internet from the internal network, and vice versa. One option would be not to allow any outgoing or incoming traffic on any ports; however, that would defeat the purpose for having an Internet connection. One solution would be to allow only one computer on a network access to the Internet. This way all other computers on the network would make their Internet requests to only one computer.

This computer would act as a proxy server, meaning that the remote computer hosting the Web page never comes into direct contact with anything on your home network other than the proxy server itself (Tyson, “How Firewalls Work”). This allows for a much more manageable network when guarding against intruders. Each company and network administrator would need to decide the level of security to employ and then take additional measures to ensure that the ports that needed to be left open were not used for purposes outside of business use. For a more detailed discussion on firewalls and proxy servers, please visit http://www.howstuffworks.com/firewall.htm and http://www.howstuffworks.com/firewall4.htm, respectively.

Human Factor

The best option left to secure networks from packet sniffing and hacking is the Human Factor. These are the flaws that humans make. For example, making passwords easy to guess, not changing them frequently, writing passwords down on paper, or giving them to unauthorized users, are all common human mistakes. If a user would take a few minutes each month to change their passwords, it would greatly help to provide or create a secure network.
One notable program on the Internet to test your computer’s security is Gibson Research Corporation’s (GRC) Shields Up software. This software can be run directly off of their web page, https://grc.com/x/ne.dll?bh0bkyd2. This application tests the various ports used by potential hackers, reports their status, and reports options for enabling a more secure computer (Gibson). One of the actual results received when the computers at the UNC Charlotte library labs were tested is shown in Appendix A.

As evident, ports 135 and 445 reported as being Open. Although the description refers to these ports as “impossible-to-close” ports, GRC does suggest a firewall to prevent access to these ports. UNC Charlotte does use a firewall, although upon contact with the Computing Services department, we were unable to get a clear answer as to why these two ports were allowed to remain Open. Appendix B is a description of the result options: Stealth, Open, and Closed. Instead of protecting our networks, the following section covers how to sniff packets.

**How to sniff Packets - Sniffing programs for your Operating System**

The following are some of the major packet sniffers used for some of the most common operating systems.

**Windows**

*Ethereal* – although this is a UNIX based program, it is probably the best freeware sniffer for Windows. It allows users to examine data from a live network or from a capture file on disk. It has 2 versions, a protocol analyzer that is a read-only version that decodes existing packet captures, and the other that captures data. Users can also view detailed and summary data on packets by interactively browsing the captured data (Ethereal, Description). It is unnecessary to install the packet capture driver; however, if installed, it is harder to do so on Windows.

*Sniff’em* – a user-friendly and cost effective network analyzer that offers packet sniffing, protocol decoding, and USB adapters support. It also has a detection system that lets users know when there is eavesdropping on the network and can log vast
amounts of data. It supports all Windows 95 versions or higher, with the exception of Windows XP (Packet Sniffing, Sniff’em).

**UNIX**

tcpendump – this is the most common and oldest wiretap program. It is the standard form of packet capturing for UNIX. It basically dumps one decoded line of data from the packet into the command line. As mentioned, Ethereal is also used for UNIX. It is the best freeware sniffer for Linux and it provides a very good GUI (Graphical User Interface) (Packet Sniffing, Unix).

For more information on other operating systems and their packet sniffers, please see Appendix C and visit the website [http://www.robertgraham.com/pubs/sniffing-faq.html](http://www.robertgraham.com/pubs/sniffing-faq.html).

**How to sniff Cable Modem and DSL segments**

Sniffing cable modem and DSL connections are similar, what applies for cable modem also applies for DSL.

**Channels and the cable box**

Cable modems use 2 asymmetric channels: an upload and a download channel. It only receives data on a high-speed channel, the download channel, while only transmitting data on a slower channel, the upload channel. This means that the cable modem box does not receive the upload channel data, only the download data. There is a slower download speed due to congestion in the download channel; thus, increasing the likelihood that data can be lost (Graham, 3.4 The Cable Box Itself).

Since most cable modem boxes are either bridges or routers, they have separate MAC addresses and IP addresses. This means that putting the Ethernet adapter into promiscuous mode will not have an effect on the actual cable modem (Packet Sniffing…, The Cable Box).
To sniff cable modem segments, it is possible to sniff broadcasts. Examples are NetBIOS packets, which advertise user names and SNMP broadcasts, which advertise network equipment such as routers and printers.

**Redirecting Traffic**

It is possible to redirect traffic through your computer to sniff into other connections. It is possible to send out an ARP packet claiming that the machine used is the local router; therefore, this connection will be flooded because everyone else will think this machine is the router and send packets towards it. As these packets are passed, they are more likely to be sniffed.

Many operating systems support ICMP (Internet Control Message Protocol). ICMP allows the control of the redirection of packets by sending the packets through a certain host, instead of the local router.

Although the packet sniffer is sniffing all these packets redirected to it, it needs to be reconfigured to send these packets to their original destination (Graham, 3.4 Redirect).

**Eavesdropping on wireless networks – IEEE 802.11**

In theory, both Apple computers and Windows based computers (and other equipment) should be able to use the same wireless infrastructure. AirPort is an implementation of the IEEE 802.11 wireless standard.

**Spread Spectrum**

IEEE 802.11 uses a spread spectrum technology. It transmits data on a range of frequencies that only the transmitter and intended receiver know; therefore, it is extremely difficult for hackers to eavesdrop on the network. Trying to eavesdrop, or sniff packets on spread spectrum, is almost impossible because the eavesdropper only hears white noise and even detecting a signal is difficult. Security against packet sniffing is achieved through digital encryption techniques. In theory, an IEEE-802.11 device can sniff packets that are within signal range (Graham, 3.7 Spread Spectrum).
**Encryption within wireless networks**

Encryption within wireless networks can be formidable, but it is commonly insecure. This makes packet sniffing attractive for wireless LANs because of simple access. The 802.11 standard uses RC4 as its encryption protocol. RC4 can use up to 128 bits for encryption, but it usually uses only 40 bits for encryption because of export restrictions. RC4 is the basis for WEP (Wired Equivalent Privacy), which is the security standard for wireless LANs. WEP is only implemented in about half of today’s wireless LANs. Two factors that weaken the security of encryption are that everyone does not use encryption; and if WEP is used, it is not too difficult to decrypt the 40 bit encrypted messages (Graham, 3.7 Encryption).

**Can you sniff a switched network?**

Theoretically, it is not possible. However, there are ways to sniff a switched network. In the past, it was taken for granted that using a switch on a network would prevent hackers from sniffing its packets.

A switch eventually finds out where the nodes are on its network and thus only sends the packets to its intended destination. This is much more efficient and secure that a hub, which sends packets to all devices connected to it. This means that fewer packets are being sent and reduces the risk of packet sniffing.

With ARP redirect, let’s assume there are two users, user A and user B. If user A wants to find user B’s Ethernet MAC address, user A can use user B’s IP address to send an ARP request. User A needs to ARP the IP packet that it will send to user B, and when user B responds, it will include its MAC address. Now, user A can send the IP packet to that MAC address. Since the request was broadcasted on the local Ethernet, everyone on the network saw the information passed. This means that the possibility of sniffing increases since packets are being transmitted to more stations (Graham, 3.8.2 ARP Redirect).
In addition, *dsniff* can tap into switched networks. *dsniff* is a collection of tools that audit networks and penetrate networks for tests. Like other sniffers, it seeks for interesting data such as passwords, credit card numbers, email, files, etc. The device wanting to sniff packets sends a forged packet to the intended network it will sniff data from; this packet will tell the network that the default gateway has changed to the attacker’s system (*dsniff*, Abstract). This is similar to redirecting traffic, previously explained, where packets are redirected to the hacker’s system and then forwarded to its ultimate destination after being sniffed for interesting data and passwords. Other tools similar to *dsniff* are *filesnarf*, *mailsnarf*, *msgsnarf*, and *urlsnarf*. The data is often found in protocols such as SNMP, FTP, POP3, HTTP, IRC, and Telnet (Graham, 4.1 What protocols are vulnerable to sniffing?).

Other ways to sniff switched networks are to change a switch from *bridging* into *repeating* mode so that all frames are broadcasted on all ports at all times. This is accomplished by overflowing the address tables with continuous and random false MAC addresses (Graham, 3.8.1 Switch Jamming).

**Writing a sniffing program on Windows**

First, the packet capture driver is selected, and these can be downloaded from the Internet. High-level languages such as Visual Basic cannot be used for these drivers; therefore, other languages such as C are required. The most commonly used drivers are PCAUSA (a commercial driver) and the freeware driver of the WinDump package (Graham, 3.10 Sniffing Driver).

**Commonly Sniffed Protocols**

The most common protocols that are sniffed are SMTP, POP3, IMAP, FTP, HTTP, and Telnet. SMTP, POP3, and IMAP are email protocols. These are commonly sniffed because they can house usernames and passwords when a user joins a new entity on the Internet. FTP sites require user authentication in the form of a username and
password. Telnet, similarly to FTP, requires user authentication in the form of a username and password. HTTP uses Basic authentication, which sends passwords across a medium.

All of the above-mentioned protocols send private information in clear text. Clear text is text, or the binary representation, that has not been encrypted (Graham, 4.1 http). Because all of these protocols house user names, passwords, or both, and send this data in clear text, people who misuse packet sniffers very commonly attack them.

Protocol Analysis

Protocol analysis consists of capturing the network traffic and analyzing it to understand what is currently happening on the network. The analysis consists of reading the hex dump of the packet and interpreting the individual fields. A hexdump is a collection of hexadecimal characters. Information that is received from the interpreting of the hex dump can prove to be very meaningful to a network administrator or hacker. The analyzer pulls each field out of the packet and attempts to explain what the numbers within the field mean. Protocol Analysis is not easy since a lot of knowledge about protocols is required to analyze efficiently and correctly (Graham, 5.1 What is protocol analysis?).

Discussion

The information gathered should prompt even the most basic of users to follow the suggestions of this paper. The use of packet sniffing to serve as a watchdog on a network is a tool that all administrators can use to protect their networks. Home users likewise can use packet sniffing to determine the activities and uses of their network while they are not actively using the network.

Protection against packet sniffing became evident as the most useful portion of the paper. Business managers need to be aware of the prevention methods. Data security is a growing concern among organizations and personal use. Since the majority of students at UNC Charlotte most likely share their Internet connections; they have at least a basic network setup in their homes.
The only contradiction to the findings of this paper is that regardless of the intended use of packet sniffing by an unauthorized entity is that it is virtually impossible to stop. Turning off your computer, or disconnecting yourself completely from the Internet is effective, however not a viable solution. The best answer is to make it as difficult as possible for an outsider to sniff your data, and if sniffing does occur, to make it as difficult as possible for that sniffer to obtain any useful information from the packet sniffer.

Packet sniffing is both a controversial and interesting concept because it can potentially harm organizations and users by tapping into credit card numbers, account numbers, and other sensitive data. If better measures to prevent packet sniffing are not discovered, then the way organizations and users do business can change and the growing e-commerce acceptance can be delayed.

**Summary and Conclusions**

Eavesdropping on networks transmissions is the focus of packet sniffing. Packet sniffing is performed through the use of packet sniffers. A packet sniffer is a device that is plugged into the network to perform eavesdropping. This report contains information regarding all aspects of packet sniffing, including what components make up a packet sniffer, packet sniffing over the Ethernet, protecting your network from packet sniffers, how to sniff packets, and sniffing over a switched network. To cover how to sniff packets adequately, this section was broken up into three parts: explaining how to sniff packets on your own operating system, how to sniff packets on a DSL or cable modem, and how to sniff packets on a wireless LAN, IEEE 802.11.

Provided more time and space for the conclusion of this report, several more topics would have been explained in greater detail. These topics include, the details of a MAC address, ARP illustrations, how a packet sniffer distinguishes between different protocols, such as POP3, HTTP, and TCP, and the filtering process for a packet sniffer.

Given money to spend, it would have been a good idea to purchase the
CommView packet sniffer for Windows. Appendix C only shows a sample screen of what it looks like, but it would have been ideal to implement it and actually sniff packets.

Some related topics that would be useful to someone interested in packet sniffing would be firewalls, proxy servers, protocols, hacking, Ethernet systems, wireless LAN’s, and switched networks.
Bibliography

http://naughty.monkey.org/~dugsong//dsniff/

http://www.wildpackets.com/products/etherpeek_mac

“Ethereal, sniffing the glue that holds the Internet together.” 12 November 2002.
http://www.ethereal.com/

https://grc.com/x/ne.dll?bh0bkyd2


http://howstuffworks.lycoszone.com/carnivore2.htm

http://grc.com/oo/packetsniff.htm

http://www.surasoft.com/tut/packsniffing.htm

http://www.infoworld.com/articles/op/xml/00/05/29/000529opswatch.xml
http://www.howstuffworks.com/encryption.htm

http://www.howstuffworks.com/firewall.htm
Executive Summary

Packet sniffing is the process of reading data from packets, a form of data that can be transmitted across a network, and then analyzing this data to gather important information. Packet sniffing is commonly used by network administrators to monitor networks, and by hackers to steal usernames, passwords, and other important information from users of a system or network. Often, users do not know that they have had important information stolen from them until it is too late, if at all.

Packet sniffing can occur over several types of networks. Ways to sniff Ethernet, switched, and wireless networks, as well as cable modem and DSL segments, are discussed within this report. There are also several ways to protect from packet sniffers: replacing hubs with switches, firewalls, reducing the human factor, and using encryption are all common ways to secure a network. These additional security technologies do not promise that the data cannot be sniffed; however, they do greatly increase the task of sniffing the data and gathering information from the collected data. Secure networks will not stay secure forever. Given enough time and resources, encryptions can be decoded, and ways around other forms of security can be determined. Because of this, network security and technology must often be reevaluated to determine if the appropriate level of security is present on your network.

Telnet, Rlogin, FTP, HTTP, POP3, SMTP, and IMAP are some of the most commonly sniffed protocols. Protocol analysis is the process of interpreting the data that packet sniffers obtain. The analysis provides the information that network administrators and hackers use to do their respective jobs or functions.

An example of how to detect open and vulnerable ports on a computer is given in Appendix A. The results of the operation are recorded in Appendix B. Appendix C includes a screen sample regarding the CommView v2.0, a packet sniffer for the Windows operating system. Additional packet sniffers for MacIntosh and DOS are also discussed in Appendix C.
Quickly Check for Connectable
Listening Internet Ports

This Internet Port Probe attempts to establish standard TCP Internet connections with a handful of standard, well-known, and often vulnerable Internet service ports on YOUR computer. Since this is being done from our server, successful connections demonstrate which of your ports are "open" or visible and soliciting connections from passing Internet port scanners.

Your computer at IP:

152.15.103.117

Is being 'NanoProbed'. Please stand by. . .

<table>
<thead>
<tr>
<th>Port</th>
<th>Service</th>
<th>Status</th>
<th>Security Implications</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Your computer has responded that this</td>
</tr>
<tr>
<td>Port</td>
<td>Service</td>
<td>Status</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>21</td>
<td>FTP</td>
<td>Closed</td>
<td>port exists but is currently closed to connections.</td>
</tr>
<tr>
<td>23</td>
<td>Telnet</td>
<td>Closed</td>
<td>Your computer has responded that this port exists but is currently closed to connections.</td>
</tr>
<tr>
<td>25</td>
<td>SMTP</td>
<td>Closed</td>
<td>Your computer has responded that this port exists but is currently closed to connections.</td>
</tr>
<tr>
<td>79</td>
<td>Finger</td>
<td>Closed</td>
<td>Your computer has responded that this port exists but is currently closed to connections.</td>
</tr>
<tr>
<td>80</td>
<td>HTTP</td>
<td>Closed</td>
<td>Your computer has responded that this port exists but is currently closed to connections.</td>
</tr>
<tr>
<td>110</td>
<td>POP3</td>
<td>Closed</td>
<td>Your computer has responded that this port exists but is currently closed to connections.</td>
</tr>
<tr>
<td>113</td>
<td>IDENT</td>
<td>Closed</td>
<td>Your computer has responded that this port exists but is currently closed to connections.</td>
</tr>
<tr>
<td>135</td>
<td>RPC</td>
<td>OPEN!</td>
<td>(Remote Procedure Call) This impossible-to-close port appears in most Windows systems. Since many insecure Microsoft services use this port, it should never be left &quot;open&quot; to the outside world. Since it is impossible to close, you will need a personal firewall to block it from external access.</td>
</tr>
<tr>
<td>Port</td>
<td>Protocol</td>
<td>Status</td>
<td>Note</td>
</tr>
<tr>
<td>------</td>
<td>-------------------</td>
<td>------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>139</td>
<td>NetBIOS</td>
<td>Stealth!</td>
<td>There is NO EVIDENCE WHATSOEVER that a port (or even any computer) exists at this IP address!</td>
</tr>
<tr>
<td>143</td>
<td>IMAP</td>
<td>Closed</td>
<td>Your computer has responded that this port exists but is currently closed to connections.</td>
</tr>
<tr>
<td>443</td>
<td>HTTPS</td>
<td>Closed</td>
<td>Your computer has responded that this port exists but is currently closed to connections.</td>
</tr>
<tr>
<td>445</td>
<td>MSFT DS</td>
<td>OPEN!</td>
<td>This impossible-to-close port first appeared on Windows 2000 and was carried over to Windows XP. Since several insecure Microsoft services use this port, it should never be left &quot;open&quot; to the outside world. Since it is impossible to close you'll need a personal firewall to block it from external access. Do it soon!</td>
</tr>
<tr>
<td>5000</td>
<td>UPnP</td>
<td>Closed</td>
<td>Your computer has responded that this port exists but is currently closed to connections.</td>
</tr>
</tbody>
</table>
If all of the tested ports were shown to have stealth status, then for all intents and purposes your computer doesn't exist to scanners on the Internet!

It means that either your computer is turned off or disconnected from the Net (which seems unlikely since you must be using it right now!) or an effective stealth firewall is blocking all unauthorized external contact with your computer. This means that it is completely opaque to random scans and direct assault. Even if this machine had previously been scanned and logged by a would-be intruder, a methodical return to this IP address will lead any attacker to believe that your machine is turned off, disconnected, or no longer exists. You couldn't ask for anything better.

There's one additional benefit: scanners are actually hurt by probing this machine! You may have noticed how slowly the probing proceeded. This was caused by your firewall! It was required, since your firewall is discarding the connection-attempt messages sent to your ports. A non-firewalled PC responds immediately that a connection is either refused or accepted, telling a scanner that it's found a live one ... and allowing it to get on with its scanning. But your firewall is acting like a black hole for TCP/IP packets! This means that it's necessary for a scanner to sit around and wait for the maximum round-trip time possible — across the entire Net, into your machine, and back again — before it can safely conclude that there's no computer at the other end. That's very cool.

NOTE: If your system did NOT show up as Stealth! but you wish that it could, you'll need to use one of the inexpensive (or FREE in the case of ZoneAlarm 2!) personal firewalls I've discovered. I will also be creating my own firewall which you can monitor and be informed of, by adding yourself to my eMailing System. But in the meantime ... I'd advise you not to wait!
"Closed" is the best you can hope for without a stealth firewall in place.

Anyone scanning past your IP address will immediately detect your PC, but "closed" ports will quickly refuse connection attempts. Your computer might still be crashed or compromised through a number of known TCP/IP stack vulnerabilities. Also, since it's much faster for a scanner to re-scan a machine that's known to exist, the presence of your machine might be logged for further scrutiny at a later time — for example, when a new TCP/IP stack vulnerability is discovered.

You should stay current with updates from your operating system vendor since new "exploits" are being continually discovered and they are first applied upon known-to-exist machines . . . like this one!

**AS NOTED ABOVE:** If your system did NOT show up as Stealth! but you wish that it could, you will need to use one of the inexpensive [personal firewalls](#) I’ve discovered. If your system's security is a concern (as I'm afraid it needs to be in this day and age), I would advise you not to wait!

If one or more of your ports are shown as OPEN then one of the following two situations must be true:

**You have servers running on those open ports:**

If your system is running Internet servers on the ports shown as OPEN, you should stay current with PC industry security bulletins. New security vulnerabilities are being found continually. When crackers learn of a new vulnerability, they quickly grab their scanner logs to search for systems that have been scanned in the past and are of the
known-to-be-vulnerable type. This allows them to be attacking logged systems within moments of learning of a newly located security hole. It is therefore important for you to respond to any news of new vulnerabilities in your systems as quickly as possible. The crackers are hoping you'll take your time.

You DO NOT have servers running on those open ports:

If you are not actively offering Internet services through the ports shown as OPEN, something is very wrong with your system:

   It is actively advertising its presence on the Internet and soliciting the attention of ALL PASSING PORT SCANNERS!

Logs of open ports are maintained by crackers and used as points of attack.

Either a server has been started without your knowledge — as is done by Trojan horse programs like Back Orifice — or you may be running one of the many "Evil Port Monitors" which has altered your system's "open port profile" in order to monitor TCP/IP connections. Evil Port Monitors will tell you that a passing scanner has just successfully probed into your system . . . but the problem (for you) is that it was a successful scan probe and the existence of your system's wide open ports will have been noticed and logged!

Your system may be monitored for Internet attacks without alerting crackers to your presence by using a real personal firewall product — instead of one of the many evil port monitors. For the best monitoring and protection I recommend ZoneLab's FREE firewall: ZoneAlarm 2.x.
Appendix C

Other Operating System packet sniffers

Windows

CommView – can capture both Dial up and Ethernet connections on Win9x, ME, NT, 2000 and XP. It is not free and is offered by TamoSoft (Packet Sniffing…, CommView). Below is a sample screen capture:

The CommView v2.0 Sniffer

(OptOut).
Anasil – a powerful network analyzer for Win9x/NT/2000/XP specifically designed for Ethernet networks. Its functions include sniffing packets, protocol decoding, network utilization, and providing detailed network statistics. It is ideal for network administrators, installers, and software developers (Packet Sniffing…, Anasil).

Macintosh

EtherPeek – Its task is to facilitate complex tasks such as troubleshooting and debugging mixed-platform and multi-protocol networks. It has won awards for its network traffic and protocol analyzing while setting the standard for ease of use (EtherPeek).

DOS

Since DOS is not a true Operating System, it is very flexible for packet sniffing compatibility.

The Gobbler and Beholder – a simple packet sniffer that has advanced packet-filtering capabilities. It is very old, but it still used and it is from the Netherlands.

Sniffer(r) Network Analyzer – also an old packet sniffer but still used. It is a commercial product that has been replaced by the Windows Sniffer (Graham, 3.1 DOS).