

# Sniffers: What They Are and How to Protect Yourself

Matthew Tanase 2002-02-26

## Sniffers: What They Are and How to Protect Yourself

by Matthew Tanase

last updated February 26, 2002

---

### Introduction

Have you ever thought about how your computer talks with others on a network? Would you like to listen to, or “sniff”, the conversation? Network engineers, system administrators, security professionals and, unfortunately, crackers have long used a tool that allows them to do exactly that. This nifty utility, known as a *sniffer*, can be found in the arsenal of every network guru, where it’s likely used everyday for a variety of tasks. This article will offer a brief overview of sniffers, including what they do, how they work, why users need to be aware of them, and what users can do to protect themselves against the illegitimate use of sniffers.

### What is a Sniffer?

A sniffer is a piece of software that grabs all of the traffic flowing into and out of a computer attached to a network. They are available for several platforms in both commercial and open-source variations. Some of the simplest packages are actually quite easy to implement in C or Perl, use a command line interface and dump captured data to the screen. More complex projects use a GUI, graph traffic statistics, track multiple sessions and offer several configuration options. Sniffers are also the engines for other programs. Intrusion Detection Systems (IDS) use sniffers to match packets against a rule-set designed to flag anything malicious or strange. Network utilization and monitoring programs often use sniffers to gather data necessary for metrics and analysis. Law enforcement agencies that need to monitor email during investigations, likely employ a sniffer designed to capture very specific traffic. Knowing that sniffers simply grab network data, let’s see how they work.

### How Does a Sniffer Work?

Before we can explore how a sniffer operates, it may be helpful to examine what enables the tool to work. During normal tasks such as Web surfing and messaging, computers are constantly communicating with other machines. (For an introduction to the way that the Internet and networking works, please see the SecurityFocus article [A Beginner’s Guide to the Internet](#).) 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. As pointed out in [A Beginner's Guide to the Internet](#), 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.

## What Does Sniffed Data Look Like?

It is easy to grasp the concepts discussed above by watching a sniffer in action. The information in the following example was derived using [tcpdump](#), a program that has been around for quite sometime and is available for many platforms. This particular snippet is an abbreviated exchange between a machine and the SecurityFocus Web server.

```
21:06:30.786814 0:1:3:e5:46:6b 0:4:5a:d1:46:ad 0800 650: 192.168.1.3.32946 >
66.38.151.10.80: P [tcp sum ok] 1:585(584) ack 336 win 64080 <nop,nop,timestamp
608776
899338> (DF) (ttl 64, id 7468, len 636)
0x0000  4500 027c 1d2c 4000 4006 8074 c0a8 0103      E..|.,@.t....
0x0010  4226 970a 80b2 0050 54ac b070 78ef d6c3      B&.....PT..px...
0x0020  8018 fa50 c663 0000 0101 080a 0009 4a08      ...P.c.....J.
0x0030  000d b90a 4745 5420 2f63 6f72 706f 7261      ...GET./corpora
0x0040  7465 2f69 6d61 6765 732f 6275 696c 642f      te/images/build/
0x0050  626c 6c74 5f72 645f 312e 6769 6620 4854      bl1t_rd_1.gif.HT
0x0060  5450 2f31 2e31 0d0a 486f 7374 3a20 7777      TP/1.1..Host:.ww
0x0070  772e 7365 6375 7269 7479 666f 6375 732e      w.securityfocus.
0x0080  636f 6d0d 0a55 7365 722d 4167 656e 743a      com..User-Agent:
0x0090  204d 6f7a 696c 6c61 2f35 2e30 2028 5831      .Mozilla/5.0.(X1
0x00a0  313b 2055 3b20 4c69 6e75 7820 6936 3836      l;U;Linux.i686

21:06:30.886814 0:4:5a:d1:46:ad 0:1:3:e5:46:6b 0800 402: 66.38.151.10.80 >
192.168.1.3.32949: P [tcp sum ok] 2363393025:2363393361(336) ack 1437810754 win 8616
<nop,nop,timestamp 899338 608766> (ttl 61, id 10825, len 388)
0x0000  4500 0184 2a49 0000 3d06 b74f 4226 970a      E...*I..=.OB&..
0x0010  c0a8 0103 0050 80b5 8cde 8401 55b3 4042      .....P.....U.@B
0x0020  8018 21a8 0543 0000 0101 080a 000d b90a      ...!..C.....
0x0030  0009 49fe 4854 5450 2f31 2e31 2032 3030      ..I.HTTP/1.1.200
0x0040  204f 4b0d 0a41 6765 3a20 320d 0a41 6363      .OK..Age:.2..Acc
0x0050  6570 742d 5261 6e67 6573 3a20 6279 7465      ept-Ranges:.byte
0x0060  730d 0a44 6174 653a 2054 7565 2c20 3132      s..Date:.Tue,.12
0x0070  2046 6562 2032 3030 3220 3033 3a30 343a      .Feb.2002.03:04:
0x0080  3538 2047 4d54 0d0a 436f 6e74 656e 742d      58.GMT..Content-
0x0090  4c65 6e67 7468 3a20 3433 0d0a 436f 6e74      Length:.43..Cont
0x00a0  656e 742d 5479 7065 3a20 696d 6167 652f      ent-Type:.image/
0x00b0  6769 660d 0a53 6572 7665 723a 2041 7061      gif..Server:.Apa
0x00c0  6368 652f 312e 332e 3232 2028 556e 6978      che/1.3.22.(Unix
0x00d0  2920 6d6f 645f 7065 726c 2f31 2e32 360d      ).mod_perl/1.26.
```

This excerpt shows two packets: an 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 (0x00##) show the data transmitted by each packet in hexadecimal format. Additionally, an ASCII decode of the payload is located off to the right - a convenient feature for crackers and nosy neighbors watching you on the network.

## Why Should Users 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.

## How Can Users Protect Themselves?

### 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 have 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?

### 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), Transport Layer Security (TLS) 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.

```

21:09:04.599289 192.168.1.3.32933 > opensource-01.ee.ethz.ch.https: . [tcp
sum ok]
793:793(0) ack 7011 win 20104 (DF) (ttl 64, id 12206, len 40)
0x0000 4500 0028 2fae 4000 4006 c059 c0a8 0103 E.(/.@.@..Y....
0x0010 8184 0799 80a5 01bb 19a2 0520 be10 d77f .....
0x0020 5010 4e88 dfd0 0000 P.N.....

21:09:04.599289 opensource-01.ee.ethz.ch.https > 192.168.1.3.32933: P [tcp
sum ok]
7011:7135(124) ack 793 win 10052 (DF) (ttl 237, id 65192, len 164)
0x0000 4500 00a4 fea8 4000 ed06 43e2 8184 0799 E.....@...C.....
0x0010 c0a8 0103 01bb 80a5 be10 d77f 19a2 0520 .....
0x0020 5018 2744 8303 0000 4d3a a587 805e e2bc P.'D...M:...^..
0x0030 9a2a 8ff3 fe95 46d4 930e b2bc 74f0 a484 .*....F.....t...
0x0040 fcae 33ad 6d1f 0198 6020 aee5 0c26 908e ..3.m...`....&..
0x0050 a1b5 17b4 84b7 44bc 1b0b 434e bbae a483 .....D...CN....
0x0060 1e23 38d3 520f 687e c5e3 b62e 5225 aa2f .#8.R.h~....R%./
0x0070 f747 1a71 669c 8fd1 55bd 511c 4988 b78a .G.qf...U.Q.I...
0x0080 a08d 554e a3fe bb7d 36ca e66b fb8b 0392 ..UN...}6..k....
0x0090 a3f3 4cef 7b04 af5a 7a94 cb4c ale6 e7fa ..L.{..Zz..L....
0x00a0 9610 a5ee .....

```

Compare this sniffed sample of a web session with the [OpenSSL](#) Web server to the example earlier in the article. Notice how the header information remains readable, but the ASCII decode of the payload contains seemingly random characters - thanks to the encryption. The two participants in this exchange, however, can both decrypt and process the data once it is received. This type of safeguard can be applied to virtually any network process and should be employed whenever possible.

## Can I Use a Sniffer?

A sniffer can be an invaluable tool for administrators, security professionals, programmers and even beginners. They are excellent utilities for troubleshooting any type of network problem, since they provide a window into local traffic. I personally have used sniffers on multiple occasions for security work and once discovered a compromised machine that periodically sent updates to a cracker. For network programming, a sniffer is a necessity for debugging in the development stages. Sniffers are an outstanding resource for the curious beginner, who hopes to understand both networks and security. Nothing can bring you closer to what really happens, when computers communicate, than these tools. I still learn new things using them and often keep a copy of Richard Stevens' book [TCP/IP Illustrated Volume 1](#) nearby for quick references.

It should be noted that the casual user should be very cautious when, where and how they use these programs. Never employ sniffers on a local network without checking with an administrator. It's best to try these techniques at home, or on a network you run.

## 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.

*Matthew Tanase is President of Qaddisin a network security company based in St. Louis. He has studied computer security for 10 years and holds a dual degree in Electrical Engineering and Computer Science. Currently, he provides network and security consulting services for universities, start-ups, small businesses and large corporations.*

### Relevant Links

[A Beginner's Guide to the Internet](#)

*A good tutorial for those new to networking*

[Tcpdump](#)

*An established sniffer available for many platforms*

[Ethereal](#)

*A powerful sniffer with a GUI and additional utilities for Unix and Windows*

[Snort](#)

*A popular IDS, which can also be used as a sniffer*

[Ettercap](#)

*A sniffer designed to work on switched networks*

[Dsniff](#)

*A collection of tools which can sniff data on a switched network*

[OpenSSL](#)

*A project designed to implement SSL and TLS.*

[Privacy Statement](#)

Copyright 2006, SecurityFocus