# Global Information Assurance Certification Paper

**Lora Danielle**
SANS Security Essentials (GSEC) v.1.2e

## Introduction to dsniff

Network sniffing is an important tool for monitoring network activity. Conversely, it is also a tool that can be used to exploit network resources. Dug Song's dsniff and the utilities provided within the dsniff suite overcome the obstacles that previously provided security in a switched environment. In the following discussion, I will review how shared and switched network environments work and explain the manner in which sniffing can be used in each environment. Next I will discuss the countermeasures that can be taken to prevent and detect sniffing in a switched environment. In addition, I will introduce dsniff and explain the suite of utilities included with dsniff. Finally, I will give an explanation of how to install and use the dsniff tools on a switched network.

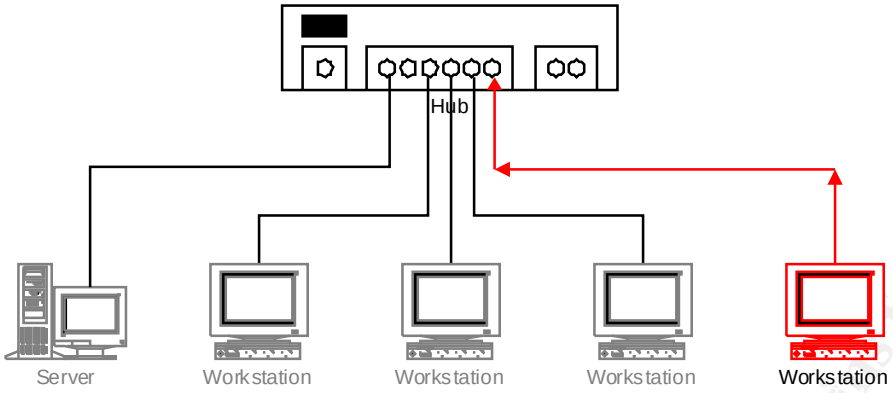
### Shared Ethernet

It is important to distinguish the difference between the two basic types of Ethernet environments. The first environment that I will discuss is shared Ethernet, which means that the LAN (Local Area Network) Ethernet cables converge into hubs. The way that traffic is passed on this type of network might be compared to the way mail is distributed during a mail call at military boot camp. One person stands at the front of the room and calls out the names on the letters as everybody stands around and listens for their own name to be called. Everyone hears whom the letter is addressed to, but only the person whom the mail is actually addressed to would (hopefully) pick up the letter. In this example, the person calling out names would represent the hub, while the people expecting letters would represent the workstations on the LAN.

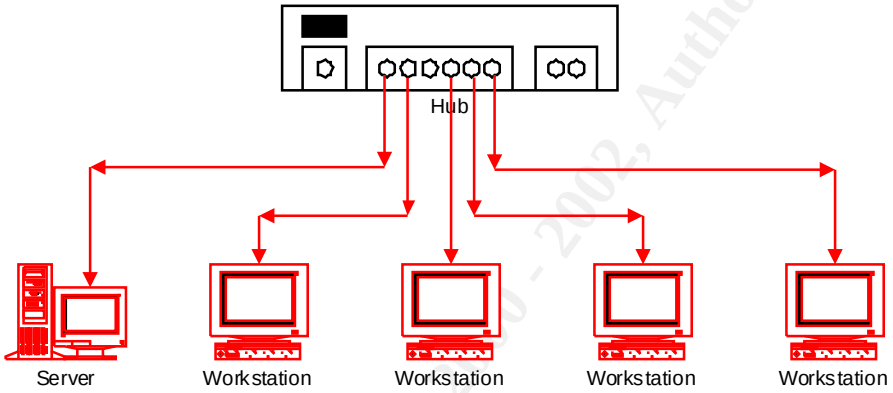
By using shared Ethernet, hubs allow for a single broadcast domain, which means if Computer A wants to send information to Computer B, the information travels to each computer on the network and says "I am from Computer A, and I am looking for Computer B." All of the computers except for Computer B ignore the information, while Computer B picks it up. Because information is transmitted to the computers on the LAN in this manner, sniffing traffic on a shared Ethernet LAN is very easily accomplished.

When someone puts a network card in promiscuous mode and installs a sniffing program on a shared network, their computer is able to collect all of the traffic on the network, instead of only the traffic that is

addressed specifically to that computer. The following diagram depicts the communication method used by a hub:




Traffic is sent from the computer to the hub.



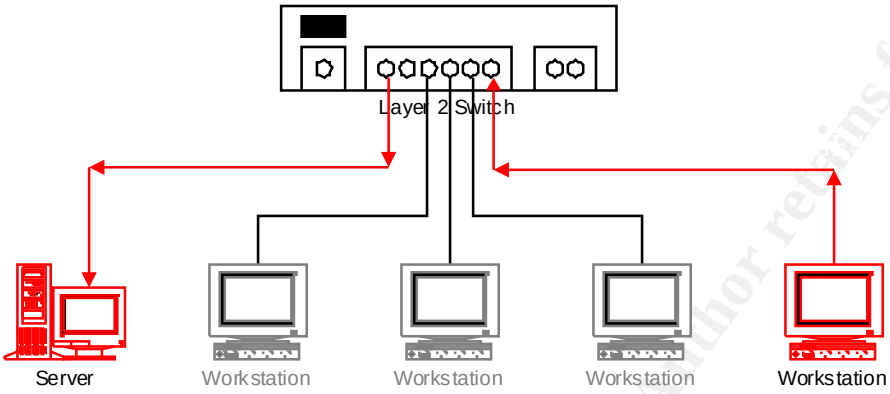

The hub forwards the traffic to the other computers on the LAN.

**Switched Ethernet**

In contrast, if you replace the hubs with switches, each Ethernet connection becomes a host-isolated connection between the switch and the computer. The switch keeps track of each computer's MAC address, and delivers the information destined for that computer only through the port that is connected to that computer. Using the mail example, this is more like a mailman, who walks around door-to-door and delivers the mail only to the person that it is addressed to. Here, the mailman represents the switch, and the different houses represent the workstations.

The switch maintains a table of information that maps the port on the switch to the MAC (Media Access Control) address of the computer that is plugged into each port of the switch. When Computer A sends information to Computer B, the switch is intelligent enough to send the information out a single port, to Computer B, rather than announce it to all of the computers that are plugged into the switch.

Because the switch can segregate traffic like this, it is no longer feasible to simply put a promiscuous card out on the network, because the switch will only forward the traffic that is meant for each computer to that individual computer's network card. Aside from broadcast traffic, which will continue to be sent to all computers, you will no longer be able to catch traffic that is not destined for your machine. The following diagram demonstrates the method by which a switch delivers traffic to and from computers on its segment:




The computer sends the information to the switch, and the switch forwards the packet to the destination computer.

For more information on security issues relating to switched networks, you can browse to the following link on the SANS website: http://www.sans.org/infosecFAQ/switchednet/switched_list.htm

**How sniffing works**

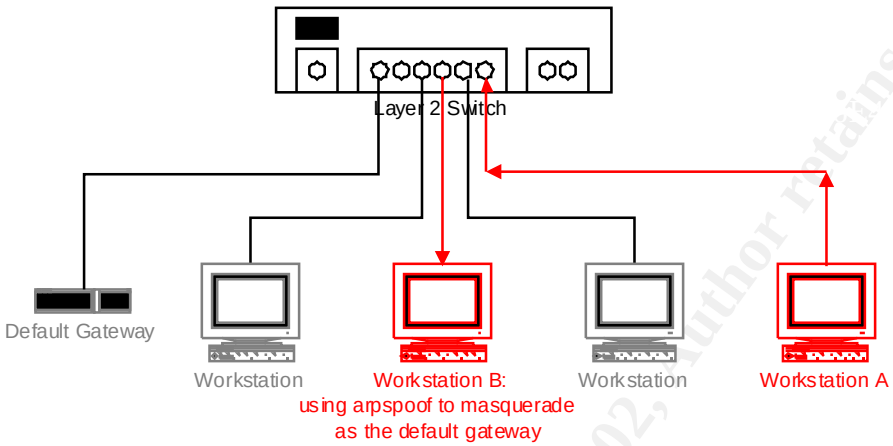
In a shared Ethernet environment, all of the traffic for all of the computers (servers and workstations) is sent out all ports on the hub. In order for someone to sniff the traffic on a shared network, they need to place a network card capable of being run in promiscuous mode into their computer. With the addition of a network-sniffing utility, the user can view any information that may be traveling on that wire. In one respect, this makes it easy for network administrators to run network-monitoring utilities, which can be very helpful for base-lining the network and diagnosing problems. On the other hand, it also makes it simple for users with rogue promiscuous network cards to pick up and analyze traffic that they should not have access to. Because of this, many companies use switched Ethernet on their networks, which creates a private session between the host and the switches on the network.

**Before you begin**

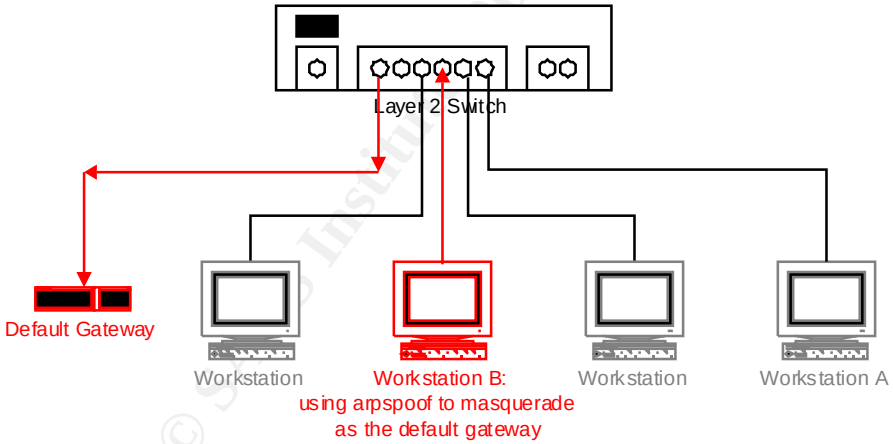
So now you may be thinking "Well, if my network is switched, then how can I sniff traffic?" The answer is: with a utility called dsniff. Dsniff is a

suite of utilities that allows a computer to intercept particular types of switched information in a variety of ways.

One way is to use an ARP (Address Resolution Protocol) redirect utility to make a computer or group of computers think that the masquerading (sniffing) computer is actually the gateway for the network, so all of the traffic from those computers is forwarded to the masquerading pc. This tool allows the false gateway to sniff the resulting traffic. This type of attack is known as a "Man (or Monkey)-in-the-Middle" attack. The following diagrams depict the traffic pattern of a Monkey-in-the-Middle attack:




Workstation A sends information to the default gateway, but Workstation B intercepts it.




Workstation B forwards the traffic on to the real default gateway.

A second option is to run another utility that is part of the dsniff suite called macof that will overload the switch, and cause it to "fail open" and begin forwarding traffic the way a hub would, rather than the way a switch should. This allows the sniffing computer to collect traffic in the same manner that it would if it were actually on a shared Ethernet network.

**Warning**

Of course, if you decide to try out dsniff, you will want to do this on your own personal network, or with the permission of the network administrators for your company. There are methods to detect sniffers, and there are legal issues to be aware of in terms of running hacking utilities without permission, particularly if you are sniffing passwords. If you are caught you may be terminated, fined, and even sent to jail. Before I began using this utility, I followed the advice given at the SANS Security Essentials track, and had three of my direct supervisors sign a "permission slip" that allowed the authorized use of dsniff to sniff network traffic and crack passwords. I distributed copies to my supervisors, and I had an extra copy filed in my HR file (you can never be too careful, especially in the technology field, where turnover is high, and your boss and your boss's boss may be gone before you are).

**Detection and countermeasures**

The only real way to prevent the sniffing of traffic by dsniff would be to deploy and use data encryption, like IPSec. Because encryption is becoming more cost efficient and easier to implement, you may want to evaluate the benefits of using encryption on your network.

When I was using the dsniff utilities, I observed that some security measures that we already had in place were able to warn administrators of suspicious network activity. When I was using arpspoof to redirect traffic, for example BlackICE Defender, a personal firewall on my workstation, noticed that the address of my default gateway had changed by giving the following alert: "**Duplicate IP Address:** [Possible Intrusion] A duplicate IP address was detected; a system may be misconfigured, or an IP address has recently changed." This error may seem a little bit generic, but as an administrator, I recognized the IP address in question to be my default gateway, and I know that the default gateway on my network would not change without my knowledge, so I was immediately suspicious!

The corporate firewall used to protect our network noticed the ARP spoof, too. This was because we had pre-configured it to look for ARP spoofs, as it did not do this automatically.

Besides personal and corporate firewalls, you can detect dsniff or other rogue network sniffers on your LAN with a tool that L0pht Heavy Industries created called **anti-sniff**, which can be downloaded from their former web site www.l0pht.com (which will redirect you to **@stake's** web site: http://www.atstake.com/research/tools/index.html). Anti-sniff is a tool that is designed to help you find promiscuous network cards on your LAN.

**The dnsiff suite**

Now that I have discussed the different types of Ethernet environments, the concept behind sniffing on switched Ethernet, and ways to detect and

prevent unauthorized sniffing, let's get started! Christopher R. Russel has a paper that can be found on the SANS web page that goes into detail about how to use dsniff once it is installed, called "**Penetration Testing with dsniff**" (http://www.sans.org/infosecFAQ/threats/dsniff.htm). I would like to cover a different aspect of dsniff, specifically, detailed instructions on how to install dsniff. Once you have dsniff installed, please reference Mr. Russel's article for more details about using dsniff.

As described by Dug Song, dsniff contains the following utilities, which are explained in more depth in the man pages, in the dsniff readme, and in Christopher Russel's article "**Penetration Testing with dsniff.**" The following descriptions were taken from each utility's man pages:

| | |
|---|---|
| **arpspoof:** | redirects packets from a target host (or all hosts) on the LAN intended for another host on the LAN by forging ARP replies. This is an extremely effective way of sniffing traffic on a switch.<br>Kernel IP forwarding (or a userland program which accomplishes the same, e.g. fragrouter (8)) must be turned on ahead of time. |
| **dnsspoof:** | forges replies to arbitrary DNS address / pointer queries on the LAN. This is useful in bypassing hostname-based access controls, or in implementing a variety of man-in-the-middle attacks. |
| **dsniff:** | is a password sniffer which handles FTP, Telnet, SMTP, HTTP, POP, poppas, NNTP, IMAP, SNMP, LDAP, Rlogin, RIP, OSPF, PPTP MS-CHAP, NFS, VRRP, YP/NIS, SOCKS, X11, CVS, IRC, AIM, ICQ, Napster, PostgreSQL, Meeting Maker, Citrix, ICA, Symantec pcAnywhere, NAI Sniffer, Microsoft SMB, Oracle SQL*Net, Sybase and Microsoft SQL protocols. |
| **filesnarf:** | saves files sniffed from NFS traffic in the current working directory |
| **macof:** | floods the local network with random MAC addresses (causing some switches to fail open in repeating mode, facilitating sniffing. |
| **mailsnarf:** | outputs e-mail messages sniffed from SMTP and POP traffic in Berkeley mbox format, suitable for offline browsing with your favorite mail reader (mail(1), pine(1), etc.). |
| **msgsnarf:** | records selected messages from AOL Instant Messenger, ICQ 2000, IRC, MSN Messenger, or Yahoo Messenger chat sessions. |
| **tcpkill:** | kills specified in-progress TCP connections (useful for libnids-based applications which require a full TCP 3-whs for TCB creation. |
| **tcpnice:** | slows down specified TCP connections on a LAN via "active" traffic shaping. |
| **urlsnarf:** | outputs all requested URLs sniffed from HTTP traffic in CLF (Common Log Format, used by almost all web servers), suitable for offline post-processing with your favorite web log analysis tool (analog, wwwstat, etc.). |
| **webspy:** | sends URLs sniffed from a client to your local Netscape browser for display, updated in real-time (as the target surfs, your browser surfs along with them, automagically). Netscape must be running on you local X display ahead of time. |
| **sshmith:** | proxies and sniffs SSL traffic redirected by dnsspoof, captures password logins and optionally allows hijacking interactive sessions.[1] |
| **webmitm:** | proxies and sniffs HTTP/HTTPS traffic redirected by dnsspoof, capturing SSL-encrypted logins and form submissions.[1] |

Once you get the utilities installed and running, you will want to experiment with the different tools in order to determine how they can help you audit the security of your network.

**Installation instructions**

When I first tried to get dsniff installed and running, I had difficulty finding a step-by-step guide. I decided to document the install procedure and explain how to install and use dsniff, so that anyone else who is having this problem might find this discussion helpful. Although dsniff has been ported to Windows NT systems, dsniff on the Linux platform has additional functionality. Additionally, there are many security tools that will run on Linux but not on Windows platforms, so I wanted to use this as a means of becoming more familiar with Linux. (Network sniffers are not operating system dependent: you can use a sniffing utility that is installed on a Linux computer to sniff traffic between computers on a Windows platform, and vice-versa). The Red Hat Linux 7.1 install was very straightforward and I did not have any problems. During the setup, the operating system can be configured to boot automatically to the X-Windows System. Because I was using DHCP for my network settings, the Linux computer came right up onto the network, and I was able to get to the Internet immediately and start downloading! If you run into problems with the install, please visit the Red Hat homepage http://www.redhat.com/ for troubleshooting advice.

**To Install dsniff:**

Once you have the Linux operating system operational, you need to download the following libraries. Based on the problems that are listed in the dsniff FAQ (http://www.monkey.org/~dugsong/dsniff/faq.html), it may be easiest to download everything in a tar.gz format. You will need to download the following files:

Libpcap v-0.5.2 or (http://www.tcpdump.org)
Libnet v-1.0.2.a or (http://www.packetfactory.net/Projects/libnet)
Libnids v-1.16 or (http://www.packetfactory.net/Projects/libnids)
Libdb v-db-3.1.17 (Berkeley DB) or (http://www.sleepycat.com)
OpenSSL or (http://www.openSSL.org)
Dsniff-2.3b4 or (http://www.monkey.org/~dugsong/dsniff/)

You can use the above hot-links, or you can go to the dsniff Frequently Asked Questions web page (http://www.monkey.org/~dugsong/dsniff/faq.html) to find hot-links to all of these utilities under section "**1.4 What else is required.**" The version numbers are very important, and using the wrong version may result in the application not working properly.

If you are going to use arpspoof to capture network traffic, you will probably want to download fragrouter (which you can get here: http://www.anzen.com/research/nidsbench/), since you may want to use it for IP forwarding.

Again, make sure that you download these packages as in a tar.gz format. Download them to the **/bin** directory.

After you have downloaded the libraries, you will need to compile and install these files. The following guidelines describe the process:

To unpack the .tar.gz file in a single step type:
**Tar -xzf filename.tar.gz**

These commands will unpack the file into a directory of the same name (i.e. **dsniff-2.3b4.tar.gz** creates a directory called **dsniff-2.3b4**).

Once you unpack the files, you need to go through three more steps per file to compile and install the applications:

Starting with the first application, for example libpcap, change into the **libpcap** directory. Check that there is a file called **configure.** Run the following command: **./configure.** Once that runs, type the command: **make** to compile the code. Once it has compiled, type the command: **make install** to install the libraries and executables. Continue in this manner for all of the packages **except libdb.** When you run the **./configure** command with **libdb**, you need to add the following syntax: **./configure --enable-compat185.** Compile dsniff last, as it will probably give you errors if the libraries are not installed first. If you encounter any errors while you are trying to install these libraries, you can search for help on the Internet. I found that google (http://www.google.com) and deja (http://www.deja.com) had two of the best search engines for finding information about dsniff.

**How to get started**

Once you get all of these packages installed on your computer, you are ready to run compile and install fragrouter. If you downloaded the .tar.gz, then follow the same steps listed above to unpack and compile the program. Once you have done that, you are ready to dsniff!

It is very important to enable IP forwarding on your computer before you use utilities like arpspoof! To start fragrouter, open a terminal window and type man fragrouter. This will show you the man pages, which will tell you the different switches you can use with the applications. To run fragrouter, just type **fragrouter -I *interface* B1** (where *interface* would be eth0 or eth1, etc.). If you only have one network card, you can get away with just typing **fragrouter B1**, and it will work like that. Be sure to run fragrouter, or to otherwise enable IP forwarding on your computer. If you don't, then all of the traffic that you are redirecting from other computers will go to your computer but will not continue on to its intended destination, thus causing noticeable interference on your network.

After you enable fragrouter, you can begin to use the other utilities, like arpspoof. At this point, you should continue your dsniff education with Christopher Russel's article, as the section **"Setting up the Attack"** demonstrates exactly how to use arpspoof and verify that it is working correctly.

**Summary**

Dsniff is a valuable tool for sniffing traffic, especially in a switched Ethernet environment. Because these tools exist, it is important to determine the damage that could result from their malicious use when considering the level of security that you need to implement for your network. You cannot assume that your network is safe just because it is switched. Always remember to consider the value of the data that travels across your network, and weigh that value against the cost of protecting the data. If your network has information that must remain confidential, consider using encryption on your LAN. Ensure that you have some sort of detection mechanism on your firewall or LAN that will help you find computers with promiscuous network cards, or tool that will detect and report ARP spoofs and other suspicious network activity.

**References:**

[1] Russel, p.1.

**Works Cited:**


Edwards, Mark Joseph. "Think You're Safe from Sniffing?" 1 June 2000. http://www.ntsecurity.net/Articles/Index.cfm?ArticleID=8878 (30 April 2001).

"Ethernet Switching: An Anixter Technology White Paper." http://www.anixter.com/techlib/whiteppr/network/d0504p06.htm (23 May 2001).

"Intel Express 10 Switch+ and 100FX Switch: IP Layer 3 Switching Concepts and Configuration." http://support.intel.com/support/express/switches/10/23364.htm (23 May 2001).

Loeb, Larry. "On the lookout for dsniff: Part 1." January 2001. http://www-106.ibm.com/developerworks/library/s-sniff.html (30 April 2001).

Loeb, Larry. "On the lookout for dsniff: Part 2." February 2001. http://www-106.ibm.com/developerworks/security/library/s-sniff2.html?dwzone=security (2 May 2001).

McClure, Stuart and Joel Scambray. "Switched networks lose their security advantage due to packet-capturing tool." 26 May 2000. http://www.infoworld.com/articles/op/xml/00/05/29/000529opswatch.xml (2 May 2001).

Prosise, Chris and Saumil Udayan Shah. "Performing Pattern Matching." 13 December 2000. http://builder.cnet.com/webbuilding/0-7532-8-4011019-4.html (2 May 2001).

Russel, Christopher R. "Penetration Testing with dsniff." 18 February 2001. http://www.sans.org/infosecFAQ/threats/dsniff.htm (30 April 2001).

Song, Dug. "dsniff." http://www.monkey.org/~dugsong/dsniff/ (26 April 2001).

Song, Dug. "dsniff Frequently Asked Questions." 1 February 2001. http://www.monkey.org/~dugsong/dsniff/faq.html (26 April 2001).