Sniffers
Basics and Detection

[Version 1.0-1]

Sumit Dhar
dharvsnl@yahoo.com

Information Security Management Team
Reliance Infocomm

Preface

This is the first version of my article titled “Sniffers Basics and Detection”. I have tried to explain in a very simple way what Sniffers are, how they work, methods of detecting sniffers, various sniffing tools and finally how to protect against sniffers. The reason I wrote this document was the fact when I started trying out sniffers, there was not a single document that covered this topic comprehensively.

This article is a work in progress. I keep adding material as and when requested by users. I am planning to add another section on using the various sniffing tools that are mentioned in this article.

I would love to hear from you, specifically if you want more details to be added to this document. Are there any doubts/queries of yours that this article didn’t clear up? Do you want more information on a particular topic? If yes, please mail me. I would love to hear from you and help you if possible. Your comments, suggestions and criticisms about this article are welcome.

And finally, this article is dedicated to my good friend GVS Karthik (Nange as we used to call him in IIT).

Sumit Dhar
dharvsnl@yahoo.com

Sniffers: Basics and Detection

> *“If you know the enemy and know yourself, you need not fear the result of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer a defeat. If you know neither the enemy nor yourself, you will succumb in every battle.”*
>
> --Sun Tzu in The Art of War

Introduction

A sniffer is a program or a device that eavesdrops on the network traffic by grabbing information traveling over a network. Sniffers basically are "Data Interception" technology. They work because the Ethernet was built around a principle of sharing. Most networks use broadcast technology wherein messages for one computer can be read by another computer on that network. In practice, all the other computers except the one for which the message is meant, will ignore that message. However, computers can be made to accept messages even if they are not meant for them. This is done by means of a Sniffer! Many people assume computers connected to a switch are safe from sniffing. Nothing could be further from the truth. Computers connected to switches are just as vulnerable to sniffing as those connected to a hub. This article seeks to explore the topic of sniffers, how they work, detecting and protecting your assets against the malicious use of these programs. Finally, towards the end we will talk about some commonly available sniffers.

How A Sniffer Works

A computer connected to the LAN has two addresses. One is the MAC (Media Access Control) address that uniquely identifies each node in a network and is stored on the network card itself. It is the MAC address that gets used by the Ethernet protocol while building “frames” to transfer data to and from a machine. The other is the IP address, which is used by applications. The Data Link Layer uses an Ethernet header with the MAC address of the destination machine rather than the IP Address. The Network Layer is responsible for mapping IP network addresses to the MAC address as required by the Data Link Protocol. It initially looks up the MAC address of the destination machine in a table, usually called the ARP (Address Resolution Protocol) cache. If no entry is found for the IP address, the Address Resolution Protocol broadcasts a request packet (ARP request) to all machines on the network. The machine with that address responds to the source machine with its MAC address. This MAC address then gets added to the source machine’s ARP Cache. The source machine in all its communications with the destination machine then uses this MAC address.

There are two basic types of Ethernet environments and how sniffers work in both these cases is slightly different.

- **Shared Ethernet:** In a shared Ethernet environment, all hosts are connected to the same bus and compete with one another for bandwidth. In such an environment packets meant for one machine are received by all the other machines. Thus when a machine Venus

(Comp 1) wants to talk to Cupid (Comp 2) in such an environment, it sends a packet on the network with the destination MAC address of Cupid along with its own source MAC address. All the computers on the shared Ethernet (Comp 3 and Comp 4) compare frame's destination MAC address with their own. If the two don't match, the frame is quietly discarded. A machine running a sniffer breaks this rule and accepts all frames. Such a machine is said to have been put into promiscuous mode and can effectively listen to all the traffic on the network. Sniffing in a Shared Ethernet environment is totally passive and hence extremely difficult to detect.

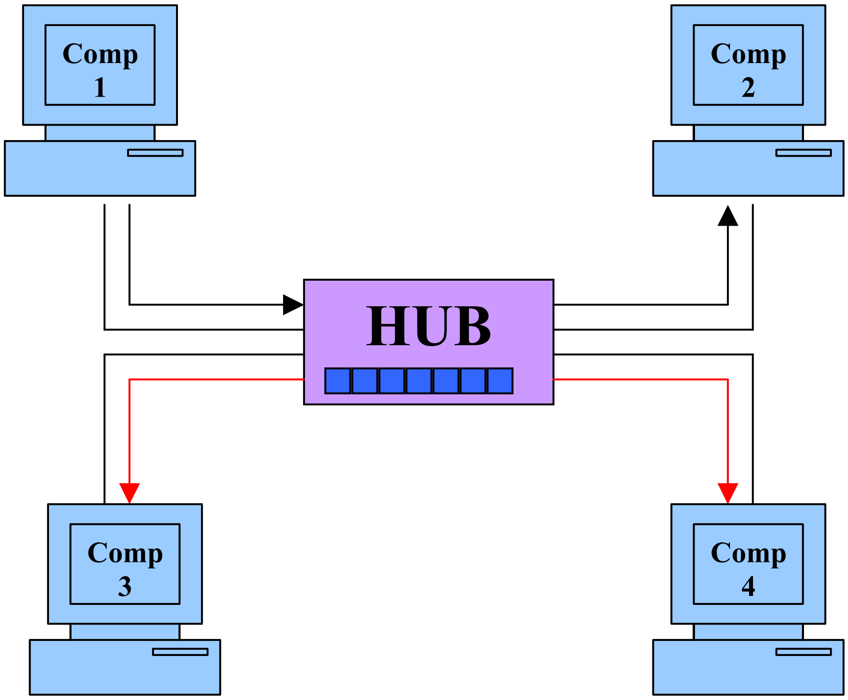

Figure 1: A Shared Ethernet Network

[Packets from Comp 1, meant for Comp 2 are received by both Comp 3 and Comp 4.But under normal circumstances they reject the packets as the MAC address in the packet does not match their MAC address. But if either of these computers is put in promiscuous mode, they can capture the entire communication between Comp 1 and Comp2]

- **Switched Ethernet:** An Ethernet environment in which the hosts are connected to switch instead of a hub is called a Switched Ethernet. The switch maintains a table keeping track of each computer's MAC address and the physical port on the switch to which that MAC address is connected and delivers packets destined for a particular machine correspondingly. The switch is an intelligent device that sends packets to the destined computer only and does not broadcast it to all the machines on the network, as in the previous case. This results in better utilization of the available bandwidth and improved security. Hence the process followed earlier, of putting the machine into promiscuous mode, to gather packets does not work. As a result of this, even many experienced Systems Administrators fall into the belief that switched networks are totally secure and immune to sniffing. Sadly, this is not really true.

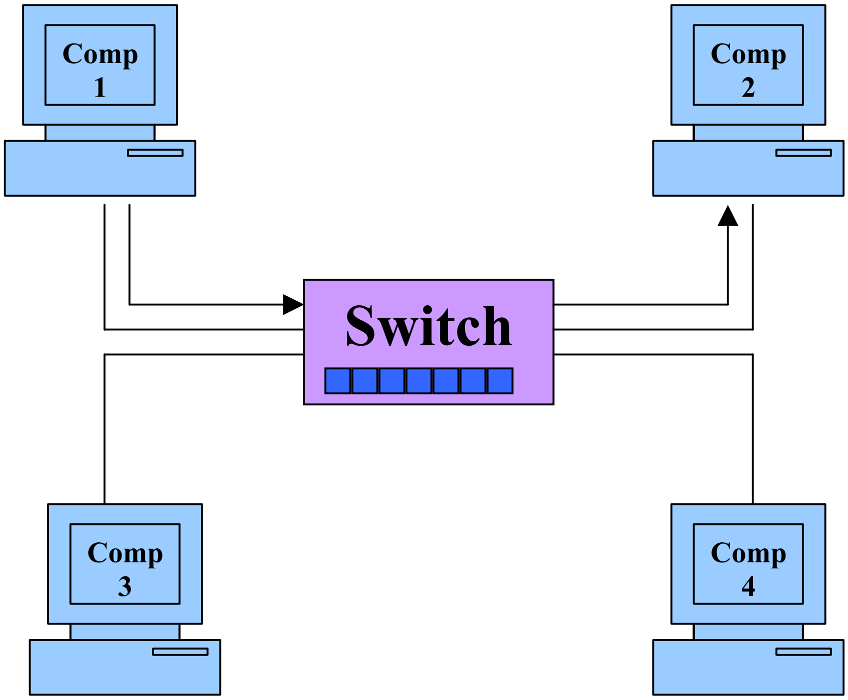

Figure 2: A Switched Network

[In a switched network, packets from Comp 1 meant for Comp 2 are not received by other terminals connected to the switch. Even if the Comp 3 and Comp 4 are in promiscuous mode, they will yet not be able to see the traffic between Comp 1 and Comp 2. For that the malicious user will either arpspoof or MAC flood the switch]

Though a switch is more secure than a hub, the following methods can still be used to sniff on a switch:

1. **ARP Spoofing:** We have explained earlier how ARP is used to obtain the MAC address of the destination machine with which we wish to communicate. The ARP is stateless, you can send an ARP reply even if one has not been asked for and such a reply will be accepted. Ideally when you want to sniff the traffic originating from machine Venus, you can ARP Spoof the gateway of the network. The ARP cache of Venus will now have a wrong entry for the gateway and is said to be poisoned. This way all the traffic destined for the gateway will pass through your machine. Another trick that can be used is to poison a hosts ARP cache by setting the gateway's MAC address to FF:FF:FF:FF:FF:FF (also known as the broadcast MAC). An excellent tool for this is the **arpspoof** utility that comes with the dsniff

suite. Using arpspoof to poison the ARP cache of a machine is accomplished by giving the command:

```
[root@ringwraith root]# arpspoof -t 203.199.66.243 203.199.66.193
0:80:ad:7c:7:3a 52:54:5:f3:95:1 0806 42: arp reply \
                                        203.199.66.193 is-at 0:80:ad:7c:7:3a
0:80:ad:7c:7:3a 52:54:5:f3:95:1 0806 42: arp reply \
                                        203.199.66.193 is-at 0:80:ad:7c:7:3a
```

The -t flag specifies the target whose ARP cache we wish to poison and the other argument is the IP address of the gateway that we wish to spoof. So now all the data destined for the gateway from the target machine will have to pass through our machine. Before you do this, it is essential you to turn on IP Forwarding on your machine. You can do this by giving the command:

```
[root@ringwraith root]# echo 1 > /proc/sys/net/ipv4/ip_forward
[root@ringwraith root]# cat /proc/sys/net/ipv4/ip_forward
1
[root@ringwraith root]#
```

If the cat command returns a value of 1, then IP Forwarding has been enabled but if it returns 0, it means IP Forwarding has not been enabled. It is important to enable IP Forwarding or else the network will die.

Note: This way you can sniff the traffic from the target machine to the gateway, but not the traffic from the gateway to the target machine. In order to do that, you will need to poison the ARP cache of the gateway too. Given the importance of the gateway machines, quite a few administrators often install programs like arp-watch to detect such malicious activities. Hence trying to poison the gateway might be risky.

2. **MAC Flooding:** Switches keep a translation table that maps various MAC addresses to the physical ports on the switch. As a result of this it can intelligently route packets from one host to another. The switch has a limited memory for this work. MAC flooding makes use of this limitation to bombard the switch with fake MAC addresses till the switch can't keep up. The switch then enters into what is known as a “failopen mode” wherein it starts acting as a hub by broadcasting packets to all the machines on the network. Once that happens sniffing can be performed easily. MAC flooding can be performed by using **macof**, a utility that comes with dsniff suite.

```
[root@ringwraith root]# macof
84:a4:d3:57:ef:8 12:56:52:42:dc:95 0.0.0.0.16630 > 0.0.0.0.3031: S \
                          1484147693:1484147693(0) win 512
88:f0:9:3f:18:89 d:86:53:53:d7:f8 0.0.0.0.15535 > 0.0.0.0.7466: S \
                          293820390:293820390(0) win 512
```

Warning: This method might lead to degeneration of the network services and should not be run for a long interval of time.

The cases we discussed so far, involve usage of sniffers by malicious unauthorized users. If you are the LAN administrator and need to set up a sniffer for some legitimate activity, you can connect the NIC of the sniffing machine to the SPAN port of the switch. As the SPAN port mirrors all the traffic flowing across the switch, you don't need to perform activities like ARP spoofing or MAC Flooding to get the packets destined for other machines.

Detecting Sniffers

A sniffer is usually passive, it just collects data. Hence it becomes extremely difficult to detect sniffers, especially when running on a Shared Ethernet. But it is slightly easier when the sniffer is functioning on a Switched Ethernet network segment. When installed on a computer, a sniffer does generate some small amount of traffic. Here is an overview of the detection methods:

- **Ping Method:** The trick used here is to send a ping request with the IP address of the suspect machine but not its MAC address. Ideally nobody should see this packet as each Ethernet Adapter will reject it as it does not match its MAC address. But if the suspect machine is running a sniffer it will respond, as it does not bother rejecting packets with a different Destination MAC address. This is an old method and not reliable any longer.

- **ARP Method:** A machine caches ARPs. So what we do is send a non-broadcast ARP. A machine in promiscuous mode will cache your ARP address. Next we send a broadcast ping packet with our IP, but a different MAC address. Only a machine that has our correct MAC address from the sniffed ARP frame will be able to respond to our broadcast ping request. Voila!

- **On Local Host:** Often after your machine has been compromised, hackers will leave sniffers, to compromise other machines. On a local machine run ifconfig. On a clean machine the output will be:

  ```
  [root@ringwraith root]# /sbin/ifconfig
  eth0      Link encap:Ethernet  HWaddr 52:54:05:F3:95:01
  inet addr:203.199.66.243  Bcast:203.199.  …
  UP BROADCAST RUNNING MULTICAST  MTU:1500  ...
  ```

  But on a machine running a sniffer the output will be slightly different. Specifically check the last line wherein it mentions "**RUNNING PROMISC**". That means the machine is in promiscuous mode and probably a sniffer is running on it.

  ```
  [root@ringwraith root]# /sbin/ifconfig
  eth0      Link encap:Ethernet  HWaddr 52:54:05:F3:95:01
  inet addr:203.199.66.243  Bcast:203.199.  ...
  UP BROADCAST RUNNING PROMISC MULTICAST    ...
  ```

  The output of the ifconfig command has been slightly modified to fit screen

- **Latency Method:** This method is based on the assumption that most sniffers do some parsing. Very simply put, in this method, huge amount of data is sent on the network and the suspect machine is pinged before and during the data flooding. If the machine is in promiscuous mode, it will parse the data, increasing the load on it. Therefore it will take extra time to respond to the ping packet. This difference in response times can be used as an indicator of whether a machine is in promiscuous mode or not. A point worth noting is that the packets may be delayed because of the load on the wire, resulting in false positives.

- **ARP Watch:** As described earlier, one method to sniff on a switched network is to ARP spoof the gateway. A utility called **arpwatch** can be used to monitor the ARP cache of a machine to see if there is duplication for a machine. If there is, it could trigger alarms and lead to detection of sniffers. Unfortunately on network implementing DHCP, this could trigger many false alarms. A simple change that can be made is the increase the DHCP lease time. This way even after your users come back after the weekend break, they will get the same IP address as before and the chance of a false alarm is greatly reduced.

- **Using IDS:** Certain Intrusion Detection Systems, monitor for ARP Spoofing on the network. The Open Source IDS **Snort** for instance has an arp-spoof preprocessor that allows it to record packets on the network with spoofed ARP addresses. Typically it compares the IP/MAC pairing it is given in the **snort.conf** file, against the pairing in the packet flowing across the network. Whenever there is a mismatch, it generates an alert.

To be honest, it is not easy to detect sniffers. Often you have to depend on intuition to realize you have a sniffer running. If your network performance suddenly takes a hit, it is possible someone has caused the switch to go into the failopen mode or if users suddenly claim that their passwords have been changed with out their knowledge, you can suspect a sniffer on the network.

The old adage, "**Prevention is better than cure**" is very true here. Every sniffer needs to be run as root (on Linux boxes) to be useful. Locking your network so that none of the users have administrative privileges is certainly an easy way to ensure the purity of your network. Having access to the root account on every machine on the network, you can periodically check if the Network Interface has been put in promiscuous mode by using the ifconfig command. This can even be automated using scripts and run periodically every hour or so.

Preventing Sniffing

The best way to secure yourself against sniffing is to use encryption. While this won't prevent a sniffer from functioning, it will ensure that what a sniffer reads is pure junk.

If you are on a Switched network, the chances are that arp spoofing will be used for sniffing purposes. The machine that the malicious user will most probably try to arp-spoof is the gateway. To prevent this from happening, you can add the MAC address of the gateway permanently to your ARP cache. This can be done by placing the MAC address of your gateway and other important machines in the /etc/ethers file.

Switch to SSH. SSH is fast becoming the de facto standard method of connecting to a Unix/Linux Machine. For more information on SSH, check out http://www.ssh.fi. You might want to check out the open-source implementation OpenSSH at http://www.openssh.org/

Instead of using http, use https if the site supports it. In case you are really bothered about the privacy of your mail, then you should give https://www.hushmail.com/ a try. Hushmail uses SSL to ensure that the data is not read in transit. You might also want to try out Pretty Good Privacy (http://web.mit.edu/network/pgp.html) or GnuPG (http://www.gnupg.org/download.html) for encrypting and signing your mails to prevent others from reading them.

If you don't want others to be able to sniff details of the websites you visit, check out http://anon.inf.tu-dresden.de/index_en.html. Others like https://www.anonymizer.com/ also offer similar services for a pay.

On the Instant Messaging front, none of the main IM programs (Yahoo, MSN, AOL, ICQ Messengers) yet support end-to-end encryption. As a result of that, IM conversations can (and often are) logged. Users might want to check out http://www.trillian.cc/, a messenger that supports encryption. Jabber (http://www.jabber.org/) is an open source messenger that supports both end-to-end encryption as well as communication via SSL, making it practically immune to sniffing.

[Note: I am not affiliated with any of the commercial vendors and do not vouch for them either. I am just providing this information on an as-is basis.]

Sniffing Tools

Since I have been a Linux man through out, I will list some of the commonly available sniffers for Linux.

- **tcpdump:** The granddaddy of packet sniffers. Ships by default on many Linux distros! It captures the headers of packets that match a Boolean expression. The captured packed data can be saved to a file for later analysis. Available at: http://www.tcpdump.org/daily/tcpdump-current.tar.gz

- **sniffit:** Robust packet sniffer with good filtering. Available at: http://sniffit.rug.ac.be/ coder/sniffit/sniffit.html.

- **ethereal:** A free network protocol analyzer for Unix and Windows. It allows you to examine data from a live network or from a capture file on disk. Captured data can be browsed via a GUI. Available for both Unix and Windows at: http://www.ethereal.com/download.html.

- **hunt:** According to Pavel Krauz, the main goal of the HUNT project is to develop tool for exploiting well known weaknesses in the TCP/IP protocol suite. Well I think he comes pretty close to it. An added advantage of using hunt is that it allows you to hijack active connections and take over their control. As far as I know, no other sniffer allows you to do that. Available at: ftp://ftp.gncz.cz/pub/linux/hunt/hunt-1.5.tgz

- **ettercap:** Ettercap is a sniffer specifically designed for switched LANs. It allows you to perform man-in-the-middle attacks against SSH and SSL. It has password collector for telnet, ftp, POP, rlogin, ssh1, icq, smb, mysql, http, NNTP, X11, napster, IRC, rip, bgp, socks 5, IMAP4, VNC, LDAP, NFS etc. Available at: http://ettercap.sourceforge.net/

- **dsniff:** I won't say much about dsniff except point you to an article by Kurt Seifried titled ``The End of SSL and SSH''. As Mark Joseph Edwards puts in an article, ``Dsniff is the Swiss army knife of privacy invasion''. The package ships with a handful of nasties: urlsnarf (to keep track of websites your network users are visiting), msgsnarf (to keep track of the instant messenger sessions of users on your LAN), mailsnarf (to keep track of the mails that users of your network are receiving), webspy (to follow a users web-surfing in real time), dsniff (to capture user passwords for quite a few protocols), filesnarf (to capture NFS files), sshmitm (to launch a man-in-the-middle attack against SSH) etc. In my opinion it is one of the most comprehensive sniffer packages available anywhere. It can wreck havoc when used for illegitimate purposes, but it is a valuable tool in hands of a capable systems administrator. Available at:
  http://monkey.org/~dugsong/dsniff/

- **lcrzoex:** It is a network toolbox for administrators that supports spoofing, sniffing, client and server creation. Over 400 possible examples are included in the package. This is another **incredible** package that I feel every systems administrator should try out. It is under active development and the author (Laurent Constantin) is a very friendly and an amazingly helpful person. Available at:
  http://www.laurentconstantin.com/

Often one or more of these programs need to be used in conjunction, to get results. Often on a switched LAN, you will first use arpspoof (which comes with Dsniff) along with hunt (in case you are planning to hijack the session) or maybe with lcrzoex (in case you are planning to capture the data to a file for later analysis). Ideally, a systems administrator should try all these packages and finally use whatever he is comfortable with.

Programs to Detect Sniffers

- Anti Sniff: From the L0pht Heavy Industries comes the new program Anti Sniff. It has the ability to monitor a network and detect if a computer is in promiscuous mode. Available at:
  http://www.securitysoftwaretech.com/antisniff/download.html

- Neped: It detects network cards on the network that are in promiscuous mode by exploiting a flaw in the ARP protocol as implemented on Linux machines. Outdated. Available at:
  ftp://apostols.org/AposTools/snapshots/neped/neped.c

- ARP Watch: ARPWatch keeps track of Ethernet/IP address pairings. This is useful when you suspect you are being arp-spoofed. Available at:
  ftp://ftp.ee.lbl.gov/arpwatch.tar.Z

- Snort: Snort is an excellent Intrusion Detection System and its arp-spoof preprocessor can be used to detect instances of ARP Spoofing, which might be an indication that someone on the network is Sniffing. Available at:
  http://www.snort.org/

None of these programs are foolproof. I speak from personal experience. I deliberately ran a sniffer on a machine and tried using these tools to detect the presence of the sniffer. Unfortunately none of the programs detected the sniffer. The reason why sniffers are called a network administrator's worst nightmare is because they are practically impossible to detect. Having blind faith in these programs to tell you if your network in under attack would be foolish. These can be aids in detection; please do not use them as the sole means of detection. And remember, with sniffers, **Prevention in better than cure**.

References

- Antisniff Technical Details: http://www.securitysoftwaretech.com/antisniff/tech-paper.html

- Robert Graham's Sniffing FAQ: http://www.robertgraham.com/pubs/sniffing-faq.html

- Dsniff Frequently Asked Questions: http://www.monkey.org/~dugsong/dsniff/faq.html