

Monitoring and Analysis of Network Traffic using Snoop

Overview

In this laboratory, you will learn how to examine network traffic for analysis and troubleshooting purposes using the *snoop* utility.

There are many different utilities that can monitor network traffic and the *snoop* is just one of them. The process of monitoring traffic is usually called "sniffing" and the corresponding tools are called sniffers.

Sniffers usually work by forcing the network card to enter a "promiscuous" mode in which a network card can intercept all packets going through the network cable that attached to it. Since this operation may cause privacy problems, only root (the privileged administrator account) can access this functionality.

You can prepare for the lab by reading the manual page of the *snoop*. You can do so by login into one of the Sun workstations and issuing the command "man snoop". The most useful options for this lab are -t, -v and -p. The example section of the manual page gives the common usage of the *snoop* tool.

A Crash Tutorial on Unix Shell

This is a simple help section for those of you that are not comfortable with the unix shell.

The output of the *snoop* command is a series of text lines. Since it sometimes can be very large, you should use the *more* command to display the information in pages so that you can read a page before you advance to the next one. This is done by "feeding" the output of a command to the input of more as in the following example:

```
# snoop | more
```

The vertical bar in the command line is used to specify that the output of the first command (*snoop* in our case) will be "fed" to the input of the second command (which is *more* in our case), and what you will see on the screen will be the output of the second program. (In this case, the second command will simply print its input one page at a time and wait for the user to press a key when advancing between pages).

Another important command that you will need to use with the "|" construct is the *grep* command. The *grep* command is used to search for lines containing a particular string. For example, if you wanted to display only the lines in the output of *snoop* that contains the "ICMP Time exceeded" string, you would simply type:

```
# snoop | grep "ICMP Time exceeded"
```

The *grep* command in the above example will filter the output of *snoop* and show only the lines that contain the specified string.

You can use the "|" construct in a cascaded manner as shown in the following example:

```
# snoop -v | grep "ICMP Time exceeded" | more
```

This series of commands simply means:

Show me the lines in the output of "snoop -v" that contain the "ICMP Time exceeded" message one page at a time, and wait for me to press a key before advancing to the next pages.

PROCEDURE

1. Find the traffic trace file

Each SUN workstation in the lab room has a file `/student/snoop.cap`. This file contains packet traces that were previously captured with the *snoop* utility. The file `snoop.cap` contains TCP packets produced by 4 different network activities: a *ping* program, a *traceroute* program, a web browsing, an FTP session.

Logon one SUN workstation with the account "root" and locate the file `/student/snoop.cap`.

2. Examining ICMP traffic generated by program "ping"

Background

The *ping* command can be used to test if some machine is connected to the network. When you "ping" a remote computer, several "ICMP Echo request" packets are sent to the remote server, and the server responds with an "ICMP Echo reply" packet for each ping request.

Hint: use the commands

```
snoop -i /student/snoop.cap -ta |grep "ICMP Echo request" and  
snoop -i /student/snoop.cap -ta |grep "ICMP Echo reply"
```

to filter out all *ping* packets from *snoop* output. The option `-i` defines the input trace file; the option `-ta` let the output display the absolute (or wall-clock) capture time for each packet.

Questions

1. What is the ip address of the computer that issued the ping request?

2. What is the ip address of the computer that replied to the ping request?
3. What was the time at which the ping command was issued? (using `-v` option to see exact date/time information).

3. Examining ICMP traffic generated by program “*traceroute*”

Background

The `traceroute` command in Unix can be used to discover the path that packets follow when trying to reach some destinations. *traceroute* does this by sending UDP or ICMP packets to the destination that expire after 1 hop, 2 hops, etc. According to the specification, once a packet expires, the router (on which the packet expires) has to notify the sender with an "ICMP Time exceeded" message. The *traceroute* program is able to discover each router along the path by noting the ip addresses that notify it about expiration of the first, second, ... and so on packets.

Hint: use key word “ICMP Time exceeded” to filter out router’s responses for *traceroute* program from snoop output.

After filtering, you will get several packets at this step. Remember: The source IP addresses that sending the message “ICMP Time exceeded” are those of routers and the final destination involved in the *traceroute* command.

Questions

1. What is the ip address of the computer that issued the *traceroute* command?
2. What is the destination ip address of the *traceroute* command?
3. List the path from the source address of the *trouceroute* command to its destination.

4. Examining HTTP traffic

Background

Web browsers retrieve pages by sending HTTP requests to web servers. A web server usually listens on the port 80. An HTTP connection usually includes many TCP packets.

Hint: Use the key word “HTTP” to filter out all HTTP packets of snoop output from snoop output, and locate the packet range for the HTTP connection.

Use the command

Snoop -i /student/snoop.cap -p <starting position, ending position> -v |more
to examine the detail of HTTP packets.

Questions

1. What is the IP address of the first web site that has been browsed? At what time was it browsed?
2. What was the User-Agent (or the web browser program) that issued the HTTP request?
3. What is the number of bytes of the retrieved web page?

5. Examining FTP traffic

Background

FTP is also a TCP based protocol. There are two ports used by an FTP server during a connection. Port 20 (ftp-data) is the port used for transferring data, while port 21 (ftp) is the port to which the commands are sent. When you connect to an FTP site with a simple FTP client, all the commands that you type and most of the visual feedback that you receive on the screen are sent to or received from the port 21 of the server. The actual transmission of data files uses the port 20 of the server.

- Hint:**
- a) Use the key word “FTP” to filter out all FTP packets from snoop output.
 - b) Use the key word “RETR” to find the packets which start a data transmission.
 - c) Use the option `-v` and `-p` to examine details of particular packets.

Questions

1. What was the IP address of the first FTP server to which the user connected? Was it an anonymous FTP? (any ftp server that allows a user logins with the account “anonymous” is an anonymous FTP server)

2. What was the IP address of the second FTP server? Was it an anonymous FTP?

3. When you list the content of a directory on an FTP server, which server port is used to send the directory content to the client?