

Lab I: Using tcpdump and Wireshark

Objectives

- To get the student familiar with basic network protocol analyzer, tools and equipment used in later labs, including tcpdump and Wireshark.

Lab Readings

- Go to http://www.tcpdump.org/tcpdump_man.html and read the manual pages of tcpdump command. Filter expression can be found at <http://www.cs.ucr.edu/~marios/ethereal-tcpdump.pdf>
- Go to <http://www.wireshark.org/docs> and read about capture filters and display filters in wireshark. Alternatively, you can go to one of many Wireshark tutorial pages such as <http://www.im.ncnu.edu.tw/ycchen/bdc2008/ethereal.ppt>
- Read introduction part (pages 1-18) of the article "[Understanding IP Addressing: Everything You Ever Wanted to Know](#)"

Prelab Questions

1. Write the tcpdump command that captures packets containing ICMP messages with a source or destination IP address 10.0.1.12.
2. Write the tcpdump syntax to capture packets whose source address is 10.0.1.5 and it uses standard ssh port (either as source or destination port)
3. Write a Wireshark display filter expression that shows packets containing ICMP messages with a source or destination IP address 136.142.116.12 and frame number between 15 and 30.
4. Write a Wireshark display filter expression that shows packets containing TCP traffic with a source or destination IP address 202.44.12.99 and using ports 80, 8080.

Equipment List

| Equipment | Quantity |
|---------------------------------|----------|
| Linux PC with one Ethernet card | 3 |
| Unmanaged switch | 1 |

Introduction and Background

Since this is the first lab that you have to configure the PCs and network configuration in Linux, a brief introduction to necessary commands will be given.

Your Linux comes with graphical user interface, but in this lab (and most of other labs) you have to use console window (text mode). Most commands in Linux come with help. If you want to consult the command manual, use the command 'man'. For example, 'man ping' will show you how to use ping command. The content will be shown one page at a time. You can press space bar to scroll down or Ctrl-b to scroll up or 'q' to quit to the shell prompt.

Another useful shell feature is command/path completion to your save typing time. You can type a partial command string and then press the Tab key. If the string is a unique prefix of a command, the shell will fill out the rest for you. For example, you can type `tcpd` followed by Tab. The shell will complete `tcpdump` for you. If multiple commands match your partial string `tcpd`, the shell will list those commands. This technique also applies to path completion, which will save your time in typing the whole pathname.

Another important command is Ping, which is a basic network utility for checking whether a target host is reachable or not. Ping sends an ICMP Echo Request message to the destination host, which will then return an ICMP Echo Reply message. For list of options, issue 'man ping'; or for brief parameter listing, try 'ping -h'.

In Linux, Ethernet interfaces are named `eth0` (first port), `eth1`, and so on. You can use command `ifconfig` to display current network interface configuration. Make sure that path `/sbin/` is in your PATH environment variable by issuing the command

```
echo $PATH
```

If `/sbin` is not the list, add the path with the command

```
export PATH=/sbin:$PATH
```

You will also see interface `lo`. This is a loop back interface and will not be used in the lab.

Wireshark is a network protocol analyzer with a graphical user interface. Originally, its name was *Ethereal* but was changed to Wireshark many years ago due to some legal problem. However, the interfaces and functionalities of Wireshark have not changed much since the last version of Ethereal. Therefore, you can safely use most of the Ethereal tutorials in Wireshark. Using Wireshark, you can interactively capture and examine network traffic, view summaries, and get detailed information for each packet.

Lab Procedures

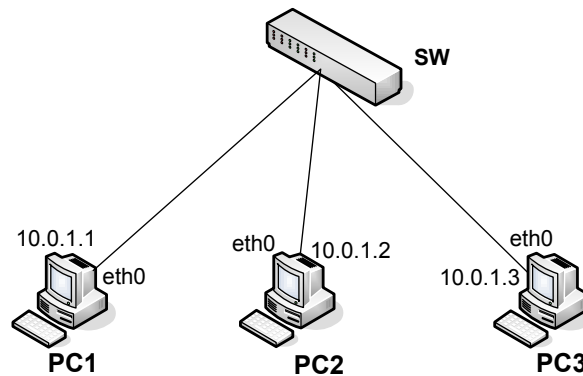


Figure 1.1 Network topology setup

Part I: tcpdump Basic

tcpdump is a console tool that allows you to capture traffic on a network and display the packet headers of the captured traffic¹. *tcpdump* can be used to identify network problems or to monitor network activities. To simultaneously view and save the output from *tcpdump*, you need to use the `-l` option of *tcpdump* together with the `tee` or `tail` commands. For example,

```
tcpdump -n -l > filename & tail -f filename
tcpdump -n -l | tee filename
```

It may be necessary to hit `Ctrl-C` to terminate the *tcpdump* session. It may sometimes be best to simply redirect the output of *tcpdump* straight to a file (e.g., `tcpdump > filename`) and view it afterward with the `more` command or a text editor.

Next we will use *tcpdump* to observe network traffic generated by issuing `ping` commands.

1. **Basic *tcpdump* usage:** Set up the network topology as shown in Figure 1.1. with default subnet mask , 255.255.255.0. For example, to configure the IP address of PC2, typing the following commands at the terminal window:

```
PC2% ifconfig eth0 10.0.1.2 netmask 255.255.255.0
PC2% ifconfig eth0 up
```
2. At PC1, start *tcpdump* to monitor all packets containing the IP address of PC2 by typing

```
PC1% tcpdump -n host 10.0.1.2
```

¹ The *tcpdump* version for Microsoft Windows is *windump*.

3. Ping a single packet from PC1 to PC2 and then terminate tcpdump (by pressing Ctrl+c). To ping a single packet, open another terminal window and type the following at prompt:

```
PC1% ping -c 1 10.0.1.2
```

Observe the tcpdump output and save it to a file.

4. At PC1, start tcpdump again by invoking command 'tcpdump -n' to capture packets.
5. Ping a single packet from PC1 to a non-existing IP address 111.111.111.111, and then ping two packets to a broadcast address 10.0.1.255.
6. Save the outputs of ping and tcpdump to a file.
7. **Using filters in tcpdump:** At PC1, ping five packets from PC1 to PC2, and use tcpdump to capture only ICMP packets that contain the IP address of PC2. Save the output of this tcpdump to a file.

Lab report:

1.1 Include the tcpdump captured output in step 3 and explain the meaning of each field.

1.2 Include the tcpdump captured outputs in step 5 and explain what happens when pinging to a non-existing IP address and pinging to a broadcast address. How many of the Linux PCs responded to the broadcast ping?

1.3 Show the tcpdump command you used in Step 7 and include the tcpdump captured output.

Part II: Wireshark Basic

This part of the lab walks you through the steps of capturing and saving network traffic with Wireshark. The exercise is conducted on PC1.

1. On PC1, start Wireshark by typing

```
PC1% wireshark &
```

This displays the *Wireshark* main window on your desktop as shown in Figure 1.2

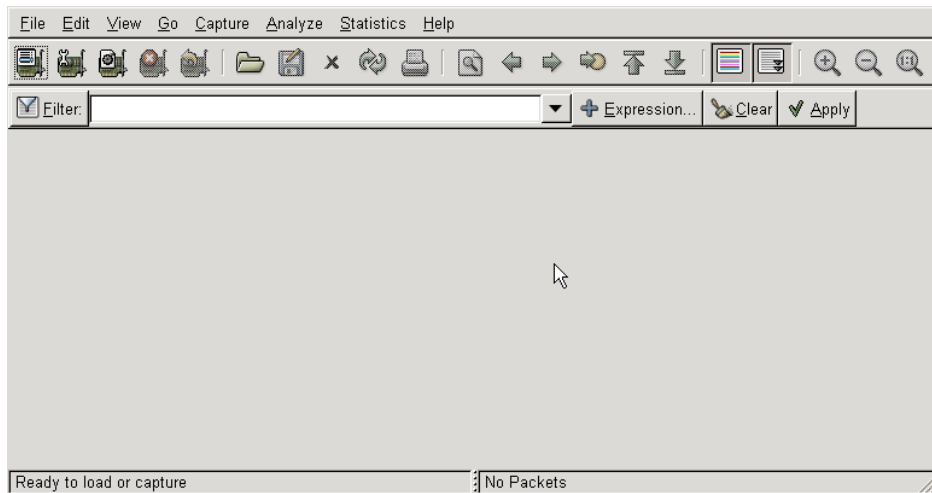


Figure 1.2 Wireshark main window

2. **Setting the capture options:** Use the instructions in Figure 1.3 to set the options of Wireshark in preparation for capturing traffic. Basically, you need to set the proper interface and make sure that you capture packets in promiscuous mode. Note that the IP address and interface in the figure differ from your actual machine.

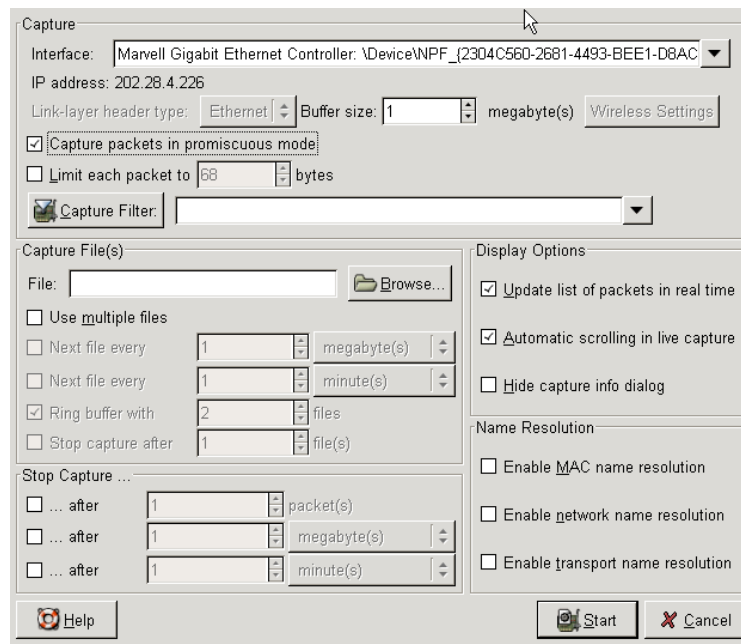


Figure 1.3 General capture settings for wireshark

3. Starting the traffic capture by clicking *Start* button in the Capture Options window. You can also start the traffic capture by clicking *Start a new live capture* button (3rd button from the left in the taskbar) once you properly set capture settings.
4. **Generating traffic:** In a separate terminal window on PC1, ping two packets to PC3. Observe the output of Wireshark in the main window. Click and highlight a captured packet in the *Wireshark* window and view the headers of the captured traffic.

5. Stopping the traffic capture: Click *Stop the running live capture* button (4th button from the left in the taskbar).
6. **Saving captured traffic:** Save the results of the captured traffic as a plain text file. This is done by selecting *Print* in the *File* menu. When the Print dialog pops up, select the Plain text-Output to file option. You must check *Packet summary line* and *Packet details* with *All expanded* options. An example of print options is shown in Figure 1.4.

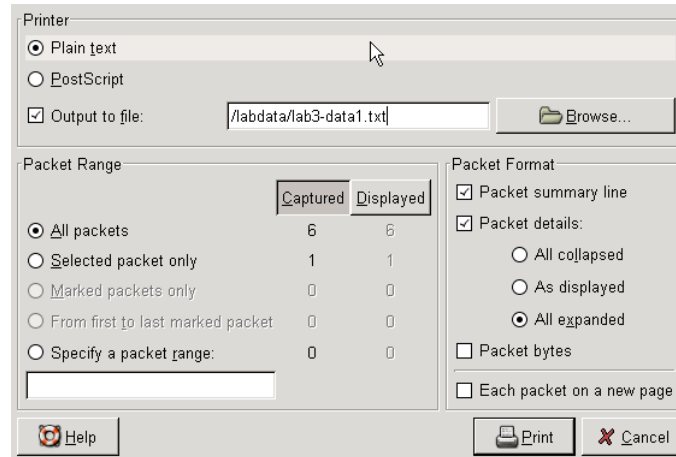


Figure 1.4 Selecting print options

Lab report:

2.1 In your lab report, include the file with the captured data and describe the differences between the files saved by *tcpdump* (in previous part) and by *Wireshark* (in this part).

NOTE:

In general, unless asked to do otherwise, always select *Packet summary line* option when you include saved data in the lab report. This will help keep the length of the lab report reasonably small. If detailed information is required, you will be asked to save details of the captured traffic. In this case, select the *Packet details* with *All expanded* option.

If you select *Save* in the *File* menu, the captured data is saved in libpcap format. Both *tcpdump* and *Wireshark* can interpret this format. However, libpcap files are not plain text files and are not useful for preparing your report.

7. **Setting display filters:** In the *Wireshark* main window on PC1, type the desired display filter in the field next to the *Filter* button, as shown in Figure 1.5. Click the *Clear* button to clear any existing filter. Set a display filter so that it shows only datagrams with destination IP address 10.0.1.3. The filter box background is green if

the syntax of the expression is correct and it turns to red if error occurs in the expression.

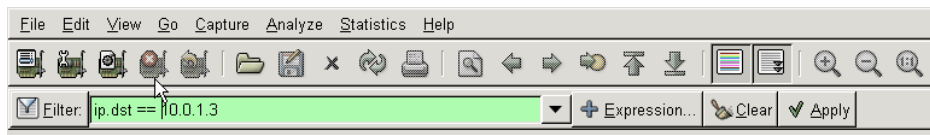


Figure 1.5 Filter box for setting display filters

Alternatively, you can click on *Expression...* button, Wireshark will pop up Filter expression dialog, as shown in Figure 1.6, for assisting expression construction.

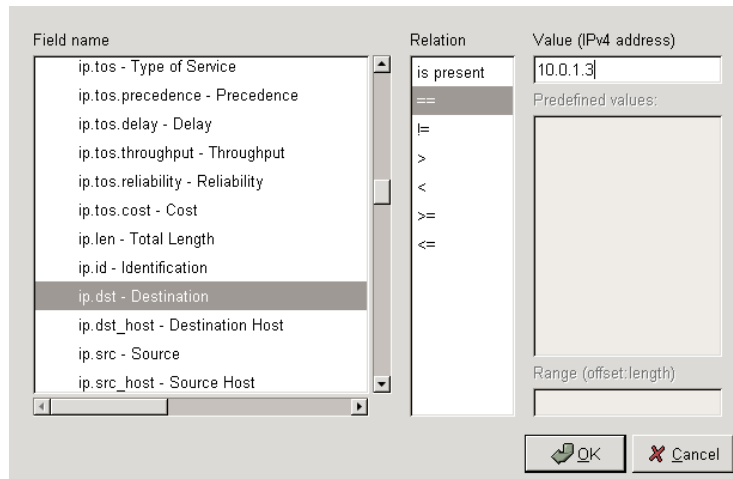


Figure 1.6 Filter expression dialog

8. Save the displayed data in the lab report by selecting *File:Print*. However, only packets that are currently being displayed are saved this time. (You have to select *Displayed* button, instead of default *Capture* button in Figure 1.4)
9. Repeat Step 7 and 8 by changing the display filter to only datagrams with source IP 10.0.1.3.
10. **More complex capture and display filters:** At PC1, clear the display filter and start to capture traffic.
11. Start a telnet session from PC1 to PC2 and log in as root. While, logged in, ping five packets from PC1 to PC2 using another terminal window and ping one packet from PC2 to PC1 using the telnetted session. Then logout after logging in successfully.
12. Stop the traffic capture of Wireshark.

Lab report:

2.2 Set the display filter to ICMP messages with source or destination address of PC2 and include the data and filter expression to the report.

- 2.3 Set the display filter to TCP traffic with source address of PC2 with source port number 23 and include the data and filter expression to the report.
- 2.4 Include the statistic summary of captured session in the report.