ARIZONA STATE UNIVERSITY

CSE 434, SLN 70516 — **Computer Networks** — Fall 2022

Supplementary information on Linux Commands and Wireshark for Lab #1

The exercises in this document introduce you to useful commands used in Lab #1 and in future labs.

1 Introduction to Linux

1.1 The Manual Pages, `man`

If you are not familiar with Linux, read the manual pages for each of the following commands.

1. **`man` pages:** The machines run the Linux operating system. Manual pages (`man` pages) exist on every lab machine. See also Linux Man Pages Online. where you can type the name of the command as a search term. The search will return the associated manual page.

`man`	`mv`	`rmdir`	`pwd`	`cp`	`chmod`	`tcpdump`
`ls`	`rm`	`kill`	`more`	`mkdir`	`ping`	`ifconfig`

2. **Wireshark:** The man page for `Wireshark`, a network analyzer tool, can be found on every lab machine. You can also read more about the tool at the Wireshark network analyzer site. The manual pages of Wireshark can be found under "Command-line Manual Pages." See also §1.8 in this document for a quick introduction to Wireshark.

1.2 Using the Linux Operating System

If you are not familiar with Linux systems, try out some Linux commands by performing the following tasks on computer A:

1. Create a terminal window.
2. Change to the home directory of the root account (`/root`).
3. Create a directory `test` in that directory (unless it already exists).
4. Copy the file `/etc/hosts` to the directory `test`.
5. Change the current directory to directory `test`.
6. Change the name of file `hosts` to `hostfile`.
7. List the content of directory `test`.
8. Edit file `hostfile` with `gedit` (or some other editor). Run `gedit` in the background.
9. Switch `gedit` to run in the foreground.
10. Change the content of the `hostfile` in the editor and save the results. Quit the editor.
11. List the content of `hostfile`.
12. Remove all files in directory `test`.
13. Remove directory `test`.

1.3 Saving Your Data

Most lab exercises ask you to save data that is displayed on your monitor to a file. Familiarize yourself with some methods to save data to a file.

Note: Whenever you create a file, place the file in the directory `/root`. Since other students may purge the files in this directory, remember to save your files to a USB drive at the end of your lab session.

Here are two methods to save data to a file on a Linux system.

1. **Save data to a file with the redirection operators:** Linux provides an easy way for redirecting the output of a command to a file via the redirection `>` and append operators `>>`.

2. **Save data with a text editor (with copy and paste):** If you have experience with a Unix-like operating system, you may have your favourite text editor (*e.g.*, `vi`, `emacs`, `nano`, etc.). If you have never edited a file on a Unix-like system, we recommend the `gedit` editor. To edit a file with name `fname` using `gedit`, simply type:
   `gedit fname`
   If you use the text editor `gedit`, you can copy text by highlighting the text and pressing Ctrl-C. Then paste the text by pressing Ctrl-V. If you are copying from a terminal window, you need to highlight the text with the mouse and press Ctrl-Shift-C instead.

On computer A try each of the preceding methods to save data to a file.

Save the output of the command `ls -l /etc` to a file named `/root/etcfile_x`, where `x` refers to the method used for saving.

1.4 Copying Files to a Flash Drive

In all labs you need the data saved in the lab sessions to complete the lab report. Since the equipment in BYENG 217 is not connected to the Internet, the most convenient way to transfer your saved data is with a USB flash drive. This part of the lab acquaints you with the basic commands for accessing a flash drive on a Linux system.

A Review of Using Flash Drives in Linux

1. **Mounting:** USB flash drives are automatically mounted by the version of Linux installed on the lab machines. The mount point will be `/media/CDROM/LABEL`, where `LABEL` is either the disk label (if you previously assigned one on another computer), or a hexadecimal number of the form `####-####`. You can run the command `ls /media/CDROM/LABEL` after inserting your drive to find out its mount point. You can also issue the command `mount` to list mount points; the last is the most recently mounted, and should begin like:
   `/dev/sdb on /media/CDROM/0000-0000 type vfat`
   The part between `on` and `type` is the mount point. The `type` will be `vfat` unless you have reformatted your flash drive to use a different file system.
2. **Using the file system:** After mounting you can perform any read and write operation on the flash drive. Everything that you read from or write to the mount point will be read from or written to the flash drive. You can copy files to and from this directory, add or delete subdirectories or files, or make this directory the current directory.
3. **Unmounting:** Before you remove the flash drive, you must first "unmount" the file system on it. If you skip this step, you may lose recently-written data and may lose everything on the drive. When you unmount a drive, the current working directory should not be its mount point or any of its subdirectories. If necessary, change the current working directory with the `cd` command. The command for mounting is:
   `umount /media/CDROM/LABEL`
   where `/media/CDROM/LABEL` is the mount point you identified before. Note the spelling of the command. (It is `umount` and not `unmount`.) You can safely remove the drive after you have unmounted the file system. In the event that the system has trouble unmounting the flash drive, try using these optional arguments with the `umount` command:
   `umount -f /media/CDROM/LABEL`
   `umount -l /media/CDROM/LABEL`

Saving data to a flash drive.

1. Use the previous commands to save a file on computer A to a flash drive.

2. On computer A, run the command `df` to obtain a list of all file systems currently mounted on your system. Save the output of the command to a file and save the file to the flash drive.

1.5 Locating Configuration Files in Linux

Linux has numerous configuration files that set the environment variables of the operating system. Studying configuration files also provides a way of learning what network configuration options are available to you.

In all labs, you will use Redhat. A list of the most important network configuration files follows:

Important: Do not modify configuration files unless asked to do so. Certain changes to the configuration files may require a reinstallation of the operating system.

Note: Configuration files are fundamentally different across different versions of Unix-like operating systems (*e.g.*, AIX, Solaris, Linux, FreeBSD). Sometimes the structure of configuration files changes between releases of the same Unix version. Furthermore, the configuration files between different versions of the same Linux distribution can have significant differences.

- `/etc/sysconfig/network`
  This file defines global parameters of the network configuration, such as the host name, domain name, and IP address of the default gateway. It also includes a line to determine whether the Linux PC acts as a router or not.
- `/etc/sysconfig/network-scripts/ifcfg-lo`
  `/etc/sysconfig/network-scripts/ifcfg-p2p1`
  `/etc/sysconfig/network-scripts/ifcfg-p2p2`
  These files define the configuration of the network interfaces. There is one configuration file for each network interface. The files `ifcfg-p2p1` and `ifcfg-p2p2` are for the two installed Ethernet interface cards. The file `ifcfg-lo` is for the loopback interface.
- `/etc/sysctl.conf`
  This file specifies many kernel options related to the network configuration.
- `/etc/hosts`
  This file specifies the mapping between host names and IP addresses for network devices. This file also determines the name of the local Linux system.

1.6 Using Ping

One of the most basic, but also most effective, tools to debug IP networks is the `ping` command. The `ping` command tests whether another host or router on the network is reachable. The `ping` command sends an `ICMP Echo Request` datagram to an interface and expects an `ICMP Echo Reply` datagram in return.

- On Linux systems, `ping` continues to send packets until you interrupt the command with Ctrl-C.
- When using `ping` on the Linux PCs, we recommend to always send at least two `ICMP Echo Request` packets. We have observed that the first `ICMP Echo Request` may often be dropped at the receiver (Time Exceeded Type 11, Code 0 or 1). This occurs when the `ICMP Echo Request` packet does not reach its destination within a certain amount of time or number of hops, *e.g.*, when waiting for an `ARP Reply` or `ICMP Redirect`. (We will learn about this later.)

Issuing `ping` commands.

1. From computer A, send five ping messages (using the `-c` option) to computer B. Save the output.
   `ping -c 5 10.0.1.12`
2. On computer B, issue a ping to the IP address of computer A. Limit the number of pings to five. Save the output.

1.7 Basics of `tcpdump`

`tcpdump` allows you to capture traffic on a network and display the packet headers of the captured traffic. `tcpdump` can be used to identify network problems or to monitor network activities.

Simple `tcpdump` exercise.

Use `tcpdump` to observe the network traffic that is generated by issuing `ping` commands.

1. Switch to computer A. Start `tcpdump` so that it monitors all packets that contain the IP address of computer B, by typing:
   `tcpdump -i p2p1 host 10.0.1.12`
2. Open a new window and execute:
   `ping -c 1 10.0.1.12`
3. Observe the output of `tcpdump`. Save the output to a file.

Another `tcpdump` traffic exercise.

1. On computer A, start capturing packets using the `tcpdump` command.
2. Issue a `ping` to the nonexistent IP address 111.111.111.111:
   `ping -c 1 111.111.111.111`
3. Issue a `ping` to the broadcast address 10.0.1.255 using the command:
   `ping -c 2 -b 10.0.1.255`
4. Save the output of `ping` and `tcpdump` to a file, how to

1.8 Basics of Wireshark

Wireshark is a network protocol analyzer with a graphical user interface. Using Wireshark, you can interactively capture and examine network traffic, view summaries, and get detailed packet information.

Running Wireshark.

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

1. **Starting Wireshark:** On computer A, start Wireshark by typing:
   `wireshark`
   or by double clicking on the `Wireshart Network Analyzer` icon on the Desktop (it looks like a shark fin). This displays the Wireshark main window on your desktop similar to that in Figure 1.
2. **Selecting the capture options:** Set the options of Wireshark in preparation for capturing traffic. Use the same options in other labs, whenever Wireshark is started.
   (a) From the main window, click on *p2p1* in the *Capture* column.
   (b) Click *Capture Options* beneath the interface list.
   (c) This displays the *Capture Options* window.
   (d) Select *Use promiscuous mode on all interfaces.*
   (e) Select *Update list of packets in real time.*
   (f) Select *Automatically scroll during live capture.*
   (g) Unselect *Resolve MAC addresses.*
   (h) Unselect *Resolve network-layer names.*
   (i) Unselect *Resolve transport-layer names.*
3. **Starting the traffic capture:** Start the packet capture by clicking *Start* in the *Capture Options* window.

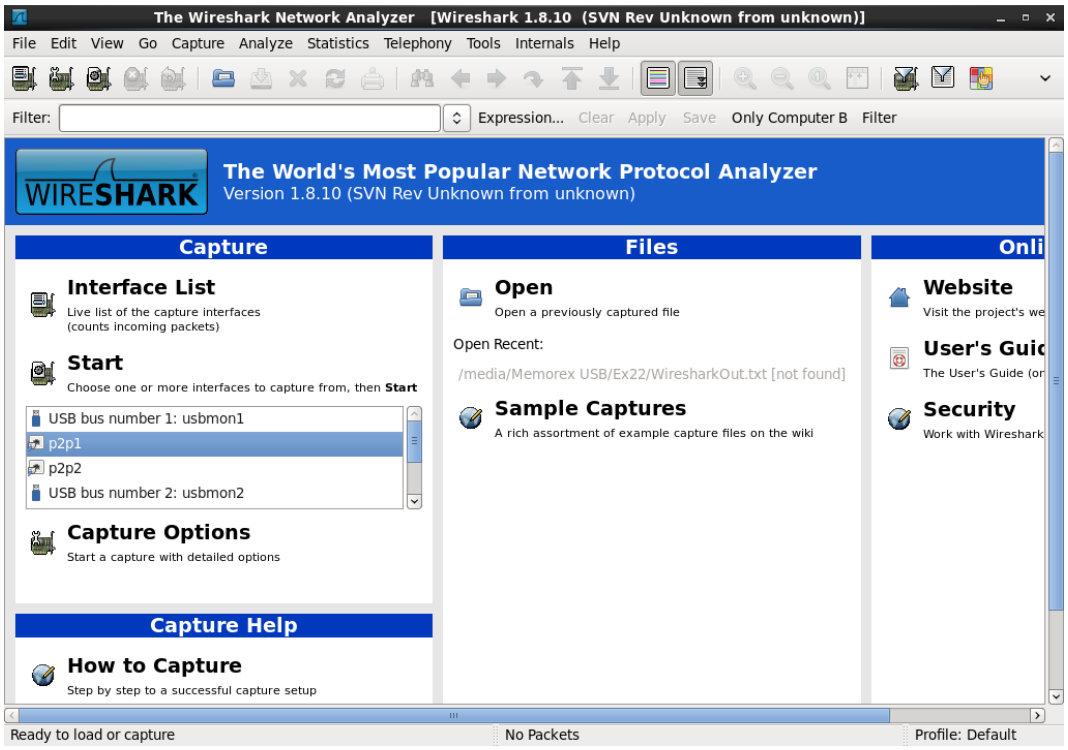

Figure 1: Wireshark main window.

4. **Generating traffic:** In a separate window on computer A, execute a `ping` command to computer C.
   `ping -c 2 10.0.1.13`
   Observe the output in the Wireshark 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 the stop button (red square) on the toolbar in the Wireshark main window.
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 a *Print* window pops up, select the options and set a filename.
   (a) Select the format *Plain text*.
   (b) Select the *Output to file* checkbox and type the filename in the field next to it.
   (c) Unselect *Packet details* if you want to save only some high-level information on each packet. This is usually sufficient. Select *Packet details* and *All expanded* if you want to save all details of all packets at all levels.
   (d) Click the *Print* button to complete the save operation.

Unless asked to do otherwise, always unselect the *Packet details* option when you include saved data in the lab report. If detailed information is required, you will be asked to save details of the captured traffic. In this case, select the *Packet details* option.

If you select *Save* from the *File* menu, the captured data is saved in the format of a `libpcap` file. This format can be interpreted by both `tcpdump` and Wireshark. Measurements saved in `libpcap` format can be analyzed at a later time. `libpcap` files are not plain text files and are not useful for preparing your report.

Unless you have `tcpdump` and/or Wireshark tools available on a system outside of the lab, which allows you to view and save captured traffic at a later time, always save captured traffic in plain text format.