In this exercise you will use a packet sniffer on a Linux system to show the contents of network traffic from two terminal programs.

A sniffer is a program that can capture any traffic on a network interface (for example, the Ethernet interface) and display its contents. The traffic does not have to be intended for delivery to the computer running the sniffer. The sniffer program tcpdump is widely used with UNIX/Linux.

tcpdump is a remote terminal emulator that is widely available on Internet computers including Linux and Windows systems. It provides a remote command window for a UNIX computer. You can, for example, login to osf1.gmu.edu via telnet.

SSH is a telnet-like program with better security features. You can also ssh to osf1.gmu.edu.

tcpdump, telnet and ssh are available on the Linux computers at the PW lab for IT441. telnet and ssh also are available on the Windows computers in that lab. It also has telnet and ssh servers. Normally, running tcpdump requires root privileges; however, the lab system has been set up for IT441 use to allow students, as ordinary users, to perform tcpdump†.

The general approach for this lab will be to run tcpdump in a mode where it captures only traffic to and from the computer it is running on, then to login to that computer (from another computer) using both telnet and ssh, and compare the resulting tcpdump data.

Lab Steps:
The work for this lab can be done using the IT441 lab set-up at the Prince William campus.

Alternatively, you can do this anywhere else where you have access to both Windows and Linux (ideally but not necessarily on the same network) and can perform the procedure described below. You will most likely have to have root access on the Linux system, in order to use the command tcpdump; the Linux systems in the PW lab are set up so that unprivileged users can run it. The lab requires both Linux and Windows machines to be used simultaneously: the former to run tcpdump, the latter to run a few network tools generating traffic.

If you do the work elsewhere, i.e., not in the PW labs, then include in your submission some information on the networking relationship between the Linux and Windows platforms used.

(1) The command tcpdump -n -X port 23 will capture Ethernet frames on the console. -n means show addresses in IP number format (dotted decimal); -X means list each frame in hexadecimal and ASCII; port 23 means listen only to traffic for TCP port 23 (the telnet port).

(2) Collect a trace of telnet traffic as follows:
   a. Determine the IP address of the Linux computer using the ifconfig command (see appendix).
   b. Run tcpdump -n -X port 23 on the Linux computer.
   c. Start telnet on a Windows computer in the lab, using Start → run and entering telnet x where x is the numeric IP address of the Linux system.
   d. After login, type ls (lower case ‘L’ not numeric digit ‘1’) in the telnet session to list the login directory.

† Another popular program for dumping network traffic is ethereal; this program dumps out Ethernet frames rather than TCP segments. It is preferable to do this lab using tcpdump, but appropriately filtered ethereal output is also acceptable. Since it is dumping lower-level traffic, ethereal will produce more output that you have to go through to isolate the traffic of interest.
e. Type control-C on the Linux computer to terminate the tcpdump.

(3) Review the traffic you have captured, and copy the sequence from login through the ls output to your report. There will be a lot of packets because each key stroke generates one packet; you should look carefully to find the key pressed in each. To include them in your report, you can start a text editor on Linux using Accessories → Text Editor, then copy text from the tcpdump Linux window (highlight [select] using the mouse, then Control-C to copy and Control-V to paste). The file can be written to a diskette or USB memory stick for transfer. Alternatively, you can use a web-based email client and email the file to yourself. Or, investigate the linux command “script” which creates a file of everything you type and everything that gets displayed on the screen (common mistake in using script: forgetting to stop it once the desired information has been captured).

(4) Repeat steps 2 and 3 using ssh on the Windows computer, with tcpdump -n -X port 22. Many of the frames you capture in this case will be ACKs between the Windows SSH client and Linux SSH server; you should save only one sample of the ACKs for your report. Concentrate on the login sequence and ls response.)

Lab Questions:

1. Compare the login sequences; look for your password in the tcpdump data. Can you find it in the telnet data? Can you find it in the ssh data? What do you conclude about the authentication security of the two terminal clients?

2. Compare the ls output in the tcpdump data. Can you see the output in the telnet data? In the SSH data? What do you conclude about the data security of the two terminal clients?

3. Explain why IT&E does not allow telnet login on its lab computers.

4. Hackers often install a sniffer on a compromised computer system. Suggest a likely reason for this.

Bonus Work:

In using public key encryption, the issue arises of whether to trust that a public key really is the public key for a particular individual. Certificates offer a way to provide assurance that a public key is genuine. For 10% more points on this assignment (i.e., you could earn 110% on the assignment), produce an X.509 certificate and submit it as a separate file with your uploaded assignment. You should also submit a printed text rendering of the certificate.

How do you produce this certificate? An open source package called OpenSSL comes as part of the standard release on current linux systems (and, being open source, is also available for Windows platforms where you would, if you intended to use it on Windows, need to install it). OpenSSL has a rich set of commands for key and certificate generation and manipulation. To create your certificate for this part of the lab, use the command:

```bash
openssl req -new -x509 -nodes -out cert.pem -keyout key.pem -newkey rsa:1024
```

and follow the prompts for information (used to create the Distinguished Name for the subject of the certificate).

There is documentation on the x509 command subset of OpenSSL on the course website.

Appendix:
The `ifconfig` command on Linux/UNIX platforms describes the state of a network interface (or, if used with some command line arguments, may set certain features if the user has the privilege level to allow that). We use the command with no arguments here in order to determine what IP address the Linux machine has. A typical response looks like:

```
eth0   Link encap:Ethernet  HWaddr 00:E2:3E:84:C0:43
       inet addr:192.168.0.102  Bcast:192.168.0.255  Mask:255.255.255.0
       UP BROADCAST NOTRAILERS RUNNING MULTICAST  MTU:1500  Metric:1
       RX packets:3505993  errors:0  dropped:0  overruns:0  frame:0
       TX packets:2767143  errors:0  dropped:0  overruns:0  carrier:0
       collisions:0  txqueuelen:100
       RX bytes:3508333602 (3345.8 Mb)  TX bytes:530728995 (506.1 Mb)
       Interrupt:9  Base address:0xb000
```

The text in blue (start of second line of output) is where the output identifies the Internet address tied to the interface (here, eth0).