ECE 4110 Internetwork Programming
Lab 4: Network Traffic Analyzers and Other Tools

Group Number: _____

Member Names: _____ _____

Date Due: Tuesday Feb. 14, 2006

Last Edited: 2/1/2006

Lab Goals

- Understand network analysis and scanning tools: ping, ethereal, and nmap tools
- Understand network vulnerabilities in FTP, and TELNET
- Understand MAC address, IP address, and TCP spoofing techniques

Section I: Ping vs. Ethereal

Do a *ping* from your machine to *57.35.6.245*. Use *ethereal* to collect approximately 5 to 10 ping packets.
Q1.1 What is the total length of each IP packet?

Q1.2 What type of protocol do you see inside the protocol field of each packet?

Q1.3 List two additional types of messages, which also use the same protocol as *ping*. (Don't ask TA for the answer)

Q1.4 How much data is inside each of these protocol packets?

Q1.5 How long does it take for the echo reply to come back according to *ethereal*?

Q1.6 How long does it take for the echo reply to come back according to the statistics you see on your Linux machine? (Note you can see these statistics on the screen after you use a <*CTRL+C*> to stop the *ping* process)

Q1.7 Are *ethereal* and *ping* the same in terms of accuracy? Explain.

Section II: Network Vulnerabilities

a) Use *ethereal* to watch an *ftp* session from your machine to 'gatechftp,' which has ip address 57.35.6.245. From your PC type *$ftp 57.35.6.245* <*ENTER*> and use *linux_class* as the user, *linux_class* as the password. Then, type *$quit* <*ENTER*> to terminate the session.
Q2.1 Can you see your password in the tcp data on the analyzer?

b) Now repeat the process but use *ssh*.
Type *$ssh –l linux_class 57.35.6.245 <ENTER>* (note: it is a lower case 'L' not the number 1)
(you might get a prompt asking if you want to continue or not. Type "yes" here).
Enter *linux_class* (as the password). Then, type *$exit <ENTER>* to terminate the session.

Q2.2 Can you see your password using secure shell login?

This is the advantage of *ssh* and why soon you will no longer be able to use *ftp* in most companies. Optional: For more info on *ssh* take a look at: http://www.ssh.com/tech/whitepapers/SSH_Secure_Shell.pdf

c) Now repeat the process using telnet
From you PC type *$telnet 57.35.6.245 <ENTER>*and use *linux_class* as the user, and *linux_class* as the password. Then, type *$exit <ENTER>* to terminate the connection.
Q2.3 Do you see the individual characters of your password in the TCP packets? (Hint: Expand the field labeled 'Telnet' in the lower window and look at successive packets sent from your machine to 'gatechftp').

Section III: Network Scanning

Type *$man nmap <ENTER>* and read the man page

Type *$nmap –h <ENTER>* and look at the result

Q3.1 Explain what *nmap* would do if you were to type *$nmap –v 57.35.6.245 <ENTER>* and ran the default scan. Hint look at $*nmap –h <ENTER>* output.

Q3.2 What ports are open on *57.35.6.245*? Hint run *$nmap –v 57.35.6.245 <ENTER>*

In a second window start *ethereal* (maybe not in promiscuous mode so you do not see your neighbor's traffic) and start packet collection. Type *$nmap –v 57.35.6.245 <ENTER>* again. Stop *ethereal* packet collection. Look at the output of *ethereal*.
Q3.3 Explain in general what you see in terms of what types of packets is your machine sending to *57.35.6.245*.

Type *$nmap –v 127.0.0.1 <ENTER>*
Q3.4 What ports are open on your own machine?

Type *$nmap –O 57.35.6.245 <ENTER>* (That is a capital O not a zero).
Q3.5 Is *nmap* up to date enough/capable enough to know what operating system we are running on 'gatechftp'?

Q3.6 Can *nmap* determine exactly which kernel version is being run on 'gatechftp'? Answer yes or no. (As an afterthought try this on your own machine *$nmap –O 127.0.0.1 <ENTER>*).

Type *$nmap –sT 57.35.6.245 <ENTER>* This is another type of scan.
Q3.7 What ports does this scan find open? Why are they different or why are they the same as before (i.e., default scanning of root privileges)? Explain.

There is a machine on the same subnet as 'gatechftp' running a web server.
Use *$nmap –sT 57.35.6.245-254 <ENTER>* to find out the IP of that machine.
Q3.8 What port indicates that there is a web server running?

Go the in web browser, the icon next to the Redhat, and type the IP you just found in the address bar.
Q3.9 What web server did you find?

Aside: Nmap frontend (invoked in a terminal by typing *$nmapfe <ENTER>*) is a graphical interface that we did not use in the lab but it works the same way in general.

Section IV: MAC Address Spoofing

The MAC address for a network interface is assigned by the hardware manufacturer at the time of manufacture. Addresses are therefore completely independent of the network to which they are attached, and addresses can be spoofed with relative ease. This spoofing has the potential to undermine common security measures. OIT, for example, uses MAC addresses on ResNet and on LAWN to tie network traffic to particular students. (Students tell OIT their MAC address when they register on ResNet or log into LAWN). By forging another student's MAC address, OIT could be led to believe that your malicious activity was actually that of another student. As another example, many wireless routers only allow access from a white list of MAC addresses so that only certain computers can access the wireless network.
You may want to use ethereal and initiate some network traffic to "see" the new MAC address being used.

Exercise: Linux MAC Cloning
Inspect the current MAC and IP address:

1. Type the command *ifconfig eth0*
2. **Record your results** (either as a screen shot OR as a text file, you will need to turn it in)
3. Type the command *ifconfig eth0 down*
   or use the Linux command *ifdown eth0*
4. Type the command *ifconfig eth0 hw ether 00:30:65:24:21:30*
   (or an Ethernet address you create)
5. Type the command *ifconfig eth0 up*

6. Type *ifconfig eth0* again and **record your results** (either as a screen shot OR as a text file, you will need to turn it in)
7. Now revert back to the old MAC address by re-doing steps 3-5, but using the MAC address from step 2 in step 4. Type *ifconfig eth0* to ensure that you did this correctly. (You do not need a screenshot).

Q4.1 What would happen if two hosts on the same network had the same MAC address? Explain

Q4.2 Although it is relatively simple to forge a MAC address, it is very difficult to determine the MAC address of a host on a subnet to which you do not already have access. Why is this true?

Section V: IP Address Spoofing

For this section you will need to setup vmware and create two virtual machines. Look at the Appendix for instructions on how to do this. Once you have set up vmware and the virtual machines, continue on this section.

For here on in, host machine refers to your base Red Hat installation with IP 57.35.6.x. VM1 is the first WS4 virtual machine, with IP 57.35.6.x+1, and VM2 is the copy WS4 machine, with IP 57.35.6.x+2.

From VM1, connect to the Network Attached Storage (NAS) and copy the file *Lab4/fraggle.c*. Carefully read through the code to understand its functionality. Compile fraggle with
#gcc -o fraggle fraggle.c

The Fraggle program spoofs a stream of UDP packets seemingly from a given source to a list of destinations. Create a file (e.g. addresses.txt) and add the IP address of VM2 followed by a carriage return.

Start Ethereal on your host machine; then, from VM1 execute fraggle so that 10000 'spoofed' packets are sent from the host machine to VM2. In other words, your host machine will be listening on the wire for traffic going from one virtual machine to the other.

Q5.1. Copy or write a sample line from your Ethereal output after your answer. What machines appear to be the source and destination of this transmission?

Now edit *fraggle.c* to demonstrate how simple it is to manipulate the contents of an IP packet. In the void method *fraggle*, the entire IP packet is pointed to by *char *packet*. Add code so that the UDP payload includes the string "ECE4110 Lab 4." Recompile *fraggle*, and rerun the above test, verifying in Ethereal that you were successful in modifying the UDP datagram.
Q5.2. What code changes were necessary?

Q5.3. Take a screen shot (#1) of Ethereal to show the packets, which contain the string "ECE4110 Lab 4".

Check-off point: show the lab TA the packets captured in ethereal.

TA Signature _____ DATE _____

In the previous section we observed how to spoof a MAC address. Now, we could also spoof ARP messages in order hijack a connection. It is important that a network administrator be able to monitor the network for such attacks, known as ARP Poisoning, in order to prevent a successful hack.

One way to prevent ARP poisoning is to monitor the network and check for consistency in ARP messages. If an IP address is claimed by one MAC address, and later that address is claimed by a different address, it is possible that ARP messages are being spoofed, sending a victim to the wrong MAC address. Similarly, if a MAC address is assigned one IP and then is changed to another, it is possible that a hacker has changed his or her MAC address, possibly for malicious purposes.

From the host machine, copy Lab4/arpwatch.tar.Z from the NAS server to your local drive. Type the commands:
#tar –zxvf arpwatch.tar.Z
#cd arpwatch-2.1a4
#./configure && make && make install

Start *arpwatch* so that we can monitor for changes. Under the default configuration, *arpwatch* will email results to the system administrator. We would rather simply print the output to the screen so type:
#./arpwatch –d
(To terminate arpwatch, hit "contraol c".)
From VM1, ping your host machine.
Q5.4 Copy or write the *arpwatch* output. (A Screen shot is OK too.)

Return to VM1 and change your machine's MAC address as we did in Section V. Ping your host machine again.
Q5.5 Copy or write the new *arpwatch* output. (A Screen shot is OK too.)

arpwatch, as shown, serves only as a detective tool, telling a system administrator when there may be a network vulnerability. The network administrator is then responsible for interpreting the warning and deciding to act or not. On Resnet, OIT uses an automated approach. If two computers appear on the network with the same MAC address, the MAC address will be automatically banned until a Resnet employee overrides the ban.

Section VI: TCP Spoofing

In our final spoofing section we will discuss sending malicious TCP packets in order to break existing TCP sessions. TCP is a connection-based protocol. Before communication can take place, a connection must be established between the source and the destination. This is done with what is known as a three-way handshake: the initiator sends a SYN message, the destination replies with acknowledgement ACK and another SYN, then the original initiator sends an ACK for the second SYN. When the communication is complete, the TCP session is terminated with a similar handshake, this time with the FIN flag.

FTP uses TCP to communicate and we will use this for our attack. On your host machine, copy the wu-ftpd-2.6.1-18.-i386.rpm file from /mnt/nas/Lab4 to your /root directory:
 # cp /mnt/nas/Lab4/wu-ftpd-2.6.1-i386.rpm /root/

Now:
 # rpm –i wu-ftpd-2.6.1-i386.rpm

Now, click on Red Hat->System Settings->Server Settings->Services and place a check mark next to wu-ftpd and click Save. Restart xinetd:
 # /etc/init.d/xinetd restart

Type *nmap localhost* and make sure ftp is running. Use the *useradd* and *passwd* commands to create a non-root user, since you cannot ftp in as root:
 # useradd <name>
 # passwd <name>
Then type in a new password, and confirm.

Open Ethereal on your host machine to observe the connection. Log into the host machine from VM2 by typing *ftp <Host IP>* . Type a command, such as *ls* to make sure that your connection is active. Type *bye* to quit.

Observe and **take a screenshot (#2)** of your Ethereal output by pressing the *print screen* button on your keyboard (you will need to turn in this screen shot). Take note of the three-way handshake initiating the connection and the four-way handshake terminating the connection.

We will now use *tcpkill* application, which is installed along with *dsnif.* Copy the files *dsniff-2.3.tar.gz*, *libnet-1.0.2a.tar.gz*, and *libnids-1.20.tar.gz* from the Network Attached Storage to VM1.

Install the tools by typing:
#tar zxvf libnet-1.0.2a.tar.gz
#cd Libnet-1.0.2a
#./configure && make && make install
#cd ..
#tar zxvf libnids-1.20.tar.gz
#cd libnids-1.20
#./configure && make && make install
#cd ..
#tar zxvf dsniff-2.3.tar.gz
#cd dsniff-2.3
#./configure && make
#make install

tcpkill monitors the network for TCP applications and will spoof *RST* messages to terminate open connections. **As we do not want to completely destroy TCP connectivity for everyone on our network, disconnect the Ethernet cable from the back of your computer.** Install *tcpkill* by typing
#make tcpkill

Start Ethereal, then open an FTP session into your host machine from VM2. In VM1, type
#./tcpkill tcp port ftp
This will make tcpkill look for ftp connections to spoof and kill. Return to VM2 and type *ls* . Notice that your connection has been terminated.

Stop Ethereal and observe your output. Stop *tcpkill* and observe its output.
Take a **screenshot (#3)** of your Ethereal output that includes the spoofed RST packets (you will need to turn in this screen shot).
Copy your *tcpkill* output to a text file (This will also have to be turned in).

Q6.1: Describe the *tcpkill* output. What are the numbers represented in the output? (Hint: you may want to correlate these messages with the Ethereal output)

Q6.2: Although this is a simple attack, it cannot be used by a hacker to arbitrarily break Internet connections. Why not?

Reconnect the Ethernet cable in the back of your computer. You're done.

Turn-in List

1. Questions and answers.
2. Section IV: two screen shots (or text files).
3. Section V: one screen shot.
4. Section VI: two screen shots.

Grading Notes

1. Make sure you handwriting is neat and readable, and **your answers are well supported**.

APPENDIX: SETTING UP VMWARE AND VIRTUAL MACHINES

1. Installing VMware on your RedHat Workstation 4 Host

Note: You will need a TA during this part to type in a VMware license number for you.

Now you need to install a program called VMWare. This software allows one machine to run multiple virtual machines. So, you'll have your own virtual mini-net to do your experiments.

Copy the *VMware-workstation-5.0.0-13124.i386.rpm* installation file from the */mnt/nas/VMWare* directory to */root.*

On your Red Hat Enterprise 4.0 host, open a terminal window. You can do this by right clicking on the mouse and selecting new terminal.

```
# rpm –Uhv VMware-workstation-5.0.0-13124.i386.rpm
# /usr/bin/vmware-config.pl
```

Accept the license
Accept the default /usr/share/icons by hitting enter
Accept the default /usr/share/pixmaps by hitting enter
Yes to build vmmon
Accept the default location for C header files

Answer yes to do you want networking for your virtual machine?
Answer no to do you want to be able to use NAT networking in your virtual machines
Answer no to do you want to use host only networking in your virtual machine

...
What this has done is set up a bridged network We are only going to use the bridged network which will act like a hub for all virtual machines that we wish to put on top of our Linux host. Each of these virtual machines will be just like plugging another machine into a hub.

Launch vmware (*# vmware*) and goto Help, Enter Serial Number. Have the TA come in and enter a serial number for your VMware license.

Answer no to would you like to register now.

2. Installing Red Hat Enterprise 4 virtual machines

Now you will be installing virtual machines on your host machine. One of the ways of doing this is to create a new virtual machine in VMware and then installing the OS on it, just as you would on a normal machine. If you already have one of these machines, you can make another machine from it by just copying the right directories. This cuts the installation time by a huge amount.
Copies of the virtual machines, created by the TAs, are available on the NAS server. You will be creating virtual machines out of them. Follow the steps below to do this.

- Copy the "*/mnt/nas/VMWare/RedHatWS4*" directory to your "*/root/vmware*" directory by using the command
  # cp -r /mnt/nas/VMWare/RedHatWS4 /root/vmware/
  This will take a long time to copy since the images are 4-6GB.
- Go to the VMWare directory
  # cd ~/vmware
- Now copy the all of the files you just got into a second directory "*RedHatWS4Copy"* by using the command

 # cp –r RedHatWS4 RedHatWS4Copy

Again, this will take a long time to copy since the images are 4-6GB.

- Type 'vmware &' in a terminal window to start VMWare. The '&' sign allows a process to run in the background, leaving the terminal open for use.

- Follow the steps below to create the first Workstation 4 virtual machine
  - Select File->New->New Virtual machine to create a new virtual machine
  - Choose Custom machine and click Next
  - Select Legacy and click Next
  - Select Linux for operating system
  - Change the name of the machine to RedHatWS4 and change the directory to /root/vmware/RedHatWS4 and click Next
  - Click Yes on the pop-up window.
  - Leave the virtual memory setting as it is. If it gives you problems, you can increase or decrease the amount of memory for each machine, later.
  - Select Bridged networking and click Next.
  - Leave the I/O adapter types as they are.
  - Choose "Use an existing virtual disk" and click Next.
  - Click Browse, go into the */root/vmware/RedHatWS4* directory and choose the file called "Red Hat Linux WS4.vmdk"
  - Click Finish. This will create a Red Hat Workstation 4 virtual machine

- Follow the steps below to create a second Workstation 4 virtual machine
  - Select File->New->New Virtual machine to create a new virtual machine
  - Choose Custom machine and click Next
  - Select Legacy and click Next
  - Select Linux for operating system
  - Change the name of the machine to RedHatWS4Copy and change the directory to /root/vmware/RedHatWS4Copy and click Next
  - Click Yes on the pop-up window.
  - Leave the virtual memory setting as it is. If it gives you problems, you can increase or decrease the amount of memory for each machine, later.
  - Select Bridged networking and click Next.
  - Leave the I/O adapter types as they are.
  - Choose "Use an existing virtual disk" and click Next.
  - Click Browse, go into the */rootvmware/RedHatWS4Copy* directory and choose the file called "Red Hat Linux WS4.vmdk"
  - Click Finish. This will create another Red Hat Workstation 4 virtual machine

3. Configuring the Workstation 4 virtual machines

Start one of the virtual machines by clicking on it in the shortcuts bar and clicking "Start this virtual machine." It will boot up like a normal system. Login with username "root" with password "password".

Note: You need to hit control and alt keys at the same time to release your mouse from VMWare.

The IP address for this machine has not been set yet. It will depend on what group you are in. Using the IP table given before, assign an IP address that is one more than your base WS4 installation. (e.g. If your range is 57.35.6.131 – 57.35.6.135 assign an IP of 57.35.6.132). Follow the steps below to do this.

- Click on the terminal icon to start a terminal.
- Type "*ifconfig*" and press Enter. It will show you the current network setup. If an IP was assigned to the original machine, of which yours is a copy, it'll be shown on the screen. eth0 means the Ethernet interface on your computer. If you had two network cards, then it would be eth0 and eth1.
- Now type "*ifconfig eth0 57.35.6.x+1 netmask 255.255.255.0*" where x is the ip of your base installation.

- Now type “route” to see the routing configuration. You need to have the default gateway set as 57.35.6.1. If there is any other entry for default, delete it by typing
  # route del gw x.y.z.a netmask 255.255.255.0 (where x.y.z.a is the wrong entry)
- You can now add the new gateway by typing
  # route add default gw 57.35.6.1
- Check your configuration by pinging your host computer (ping 57.35.6.x) and also the default gateway (ping 57.35.6.1). Press control+c to stop the ping.
- Set up the other virtual machine in exactly the same way, but set its IP address to *57.35.6.x+2,* where x is the IP of your host machine. Try pinging one virtual machine from the other virtual machine to make sure everything is set up correctly.
- Now check if any unnecessary services are running. You can turn them off by typing ‘*ntsysv*’ in a terminal window. Scroll down to sendmail and turn it off by pressing the space bar at that line. Do this for syslog also. Tab to OK, press return bar when OK highlighted.

Note: When you turn the machine off, the IP settings might revert back to the original IP. In this case you'll have to set the IP again. So, check the IP first, if you have problems with networking.

You can “pause” the virtual machines so that next time you start them, they'll start from the state you left them in. This is a bit fast than rebooting every time.

4. NAS Configuration on Virtual machines

To be able to access the NAS server from these virtual machine you will need to the NAS configuration. This should already be done for you. Check your */etc/fstab* file to make sure that it has the line:

//NAS/linux_class /mnt/nas smbfs noauto,username=linux_class,uid=500,gid=500 1 0

Make sure there is a directory called */mnt/nas*

The */etc/hosts* file should have the line

57.35.6.10 nas.storagerus.com nas

Now to mount the NAS, every time, you can just type
 # mount /mnt/nas
The password is “linux_class”.
After this, the NAS will be loaded just like a floppy or cd. It even has a graphical folder in */mnt*.

Summary:

At this point you have set up your RedHat Workstation 4 host machine with a base address, and two RedHat Workstation 4 virtual machines. You now have three computers in one box all connected together in a bridged network which can communicate through the single network interface card out to any network to which you connect your host machine's physical interface card.